Time Series Assignment

1. Load the data

```
```{r load_data, echo=FALSE}
elec_train <- read_excel('2023-11-Elec-train.xlsx')
colnames(elec_train) <- c("Date", "Power_kw", "Temp_c")

head(elec_train)
```
```

A tibble: 6 × 3

Date <chr>	Power_kw <dbl>	Temp_c <dbl>
40179.052083333336	165.1	10.55556
1/1/2010 1:30	151.6	10.55556
1/1/2010 1:45	146.9	10.55556
1/1/2010 2:00	153.7	10.55556
1/1/2010 2:15	153.8	10.55556
1/1/2010 2:30	159.0	10.55556

6 rows

2. Add the explanatory variables

Convert the timestamp to Date format and add the hour and Day variables extracted from Date

```{r convert_timestamp}

elec_train$Date <- as.POSIXct(elec_train$Date, format="%m/%d/%Y %H:%M", tz="UTC")
# Add time features
elec_train$Hour <- as.numeric(format(elec_train$Date, "%H"))
elec_train$Day <- as.numeric(format(elec_train$Date, "%u"))


head(elec_train)
```

A tibble: 6 × 5

Date <S3: POSIXct>	Power_kw <dbl>	Temp_c <dbl>	Hour <dbl>	Day <dbl>
<NA>	165.1	10.55556	NA	NA
2010-01-01 01:30:00	151.6	10.55556	1	5
2010-01-01 01:45:00	146.9	10.55556	1	5
2010-01-01 02:00:00	153.7	10.55556	2	5
2010-01-01 02:15:00	153.8	10.55556	2	5
2010-01-01 02:30:00	159.0	10.55556	2	5

We see that one row of Data is null, it is at beginning and we remove it.

3. Check for the missing values

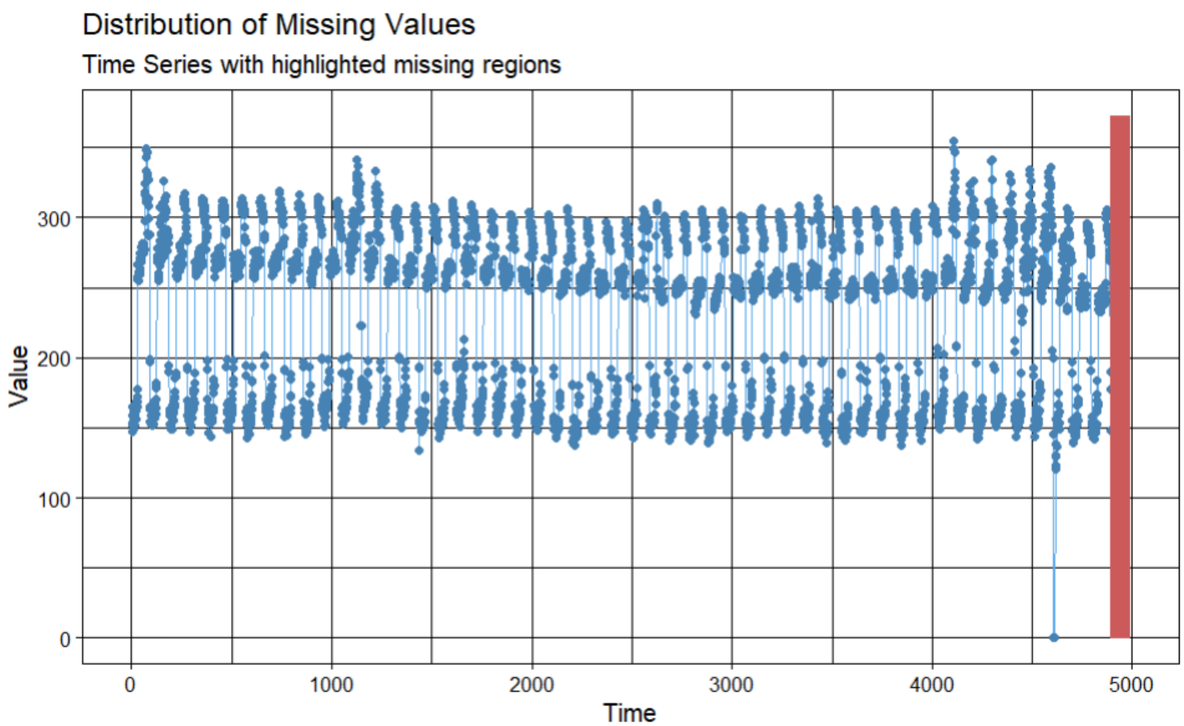
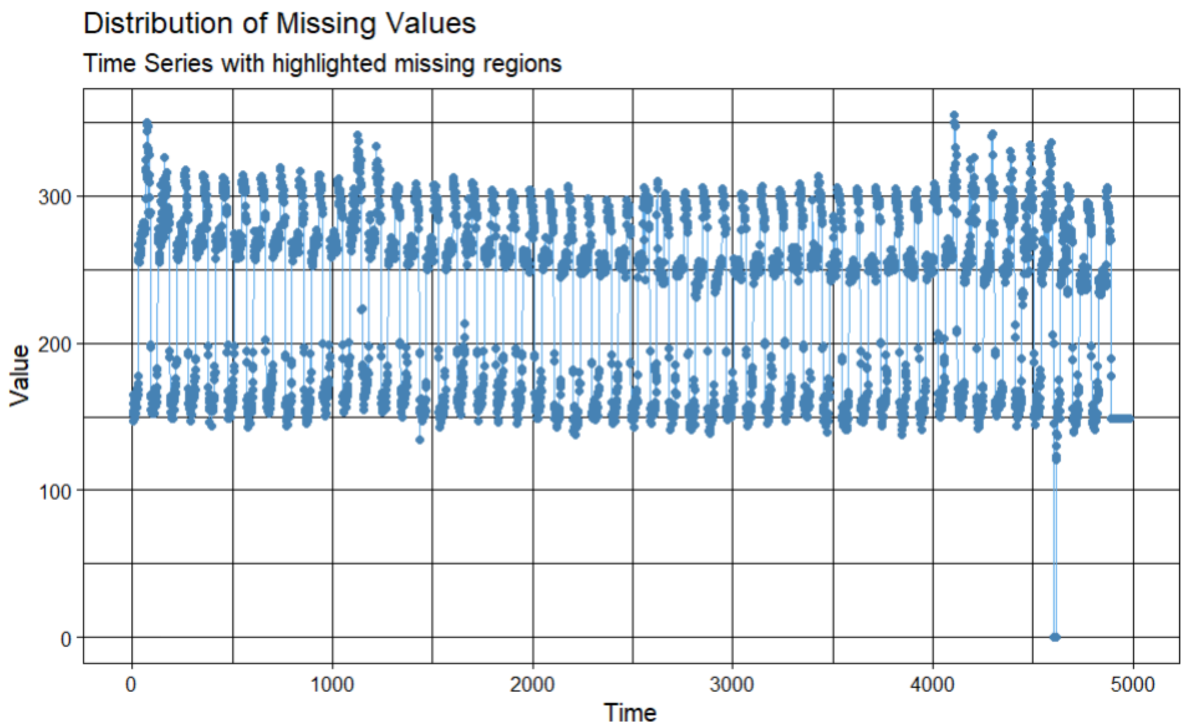
```{r check_missing_value, echo=FALSE}

colSums(is.na(elec_train))
```

```
    Date Power_kw   Temp_c     Hour      Day
       1       96        0        1        1
```

We see that Power has 96 missing rows, we fix it using **interpolation**

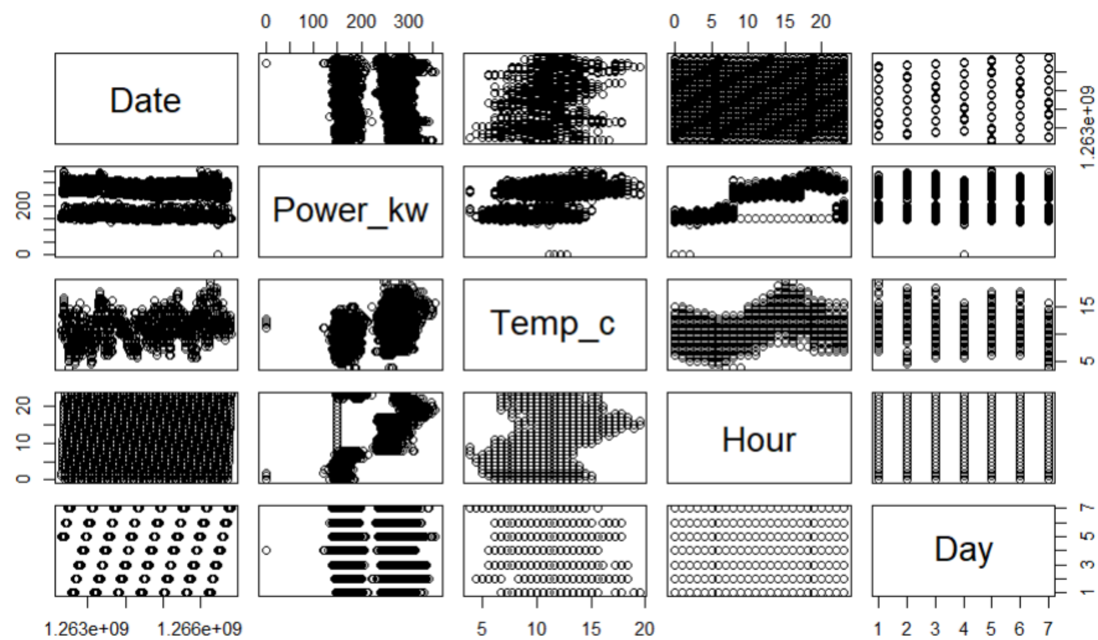
Plot Missing Data for power

```
```{r missing_data_power, echo= FALSE}

library(imputeTS)

ggplot_na_distribution(elec_train$Power_kw)


elec_train$Power_kw = na_interpolation(elec_train$Power_kw)

ggplot_na_distribution(elec_train$Power_kw)

```
```

Before interpolation



After Interpolation



After converting to Date format, the first row of data does not have the date.

```
```{r check_missing_value_3, echo=FALSE}

colSums(is.na(elec_train))
```
```

```
     Date Power_kw    Temp_c      Hour       Day
        1        0         0         1         1
```

```
```{r missing_value_after_dateformat, echo=FALSE}
# Remove rows where Date is NA
elec_train <- elec_train[!is.na(elec_train$Date), ]

# Verify that no missing values remain in the Date column
colSums(is.na(elec_train))
```
```

```
     Date Power_kw    Temp_c      Hour       Day
        0        0         0         0         0
```

4. Summary of the Data

```
```{r summary, echo=FALSE}
summary(elec_train)
```
```

```
      Date                          Power_kw        Temp_c           Hour      
 Min.   :2010-01-01 01:30:00   Min.   :  0.0   Min.   : 3.889   Min.   : 0.00  
 1st Qu.:2010-01-14 01:03:45   1st Qu.:161.7   1st Qu.: 9.444   1st Qu.: 6.00  
 Median :2010-01-27 00:37:30   Median :252.5   Median :11.111   Median :12.00  
 Mean   :2010-01-27 00:37:30   Mean   :229.2   Mean   :10.947   Mean   :11.51  
 3rd Qu.:2010-02-09 00:11:15   3rd Qu.:276.2   3rd Qu.:12.778   3rd Qu.:18.00  
 Max.   :2010-02-21 23:45:00   Max.   :355.1   Max.   :19.444   Max.   :23.00  
      Day       
 Min.   :1.000  
 1st Qu.:2.000  
 Median :4.000  
 Mean   :4.114  
 3rd Qu.:6.000  
 Max.   :7.000  
```

5. Plot the Data

```
plot(elec_train)
```



6. ACF and PACF Plotting

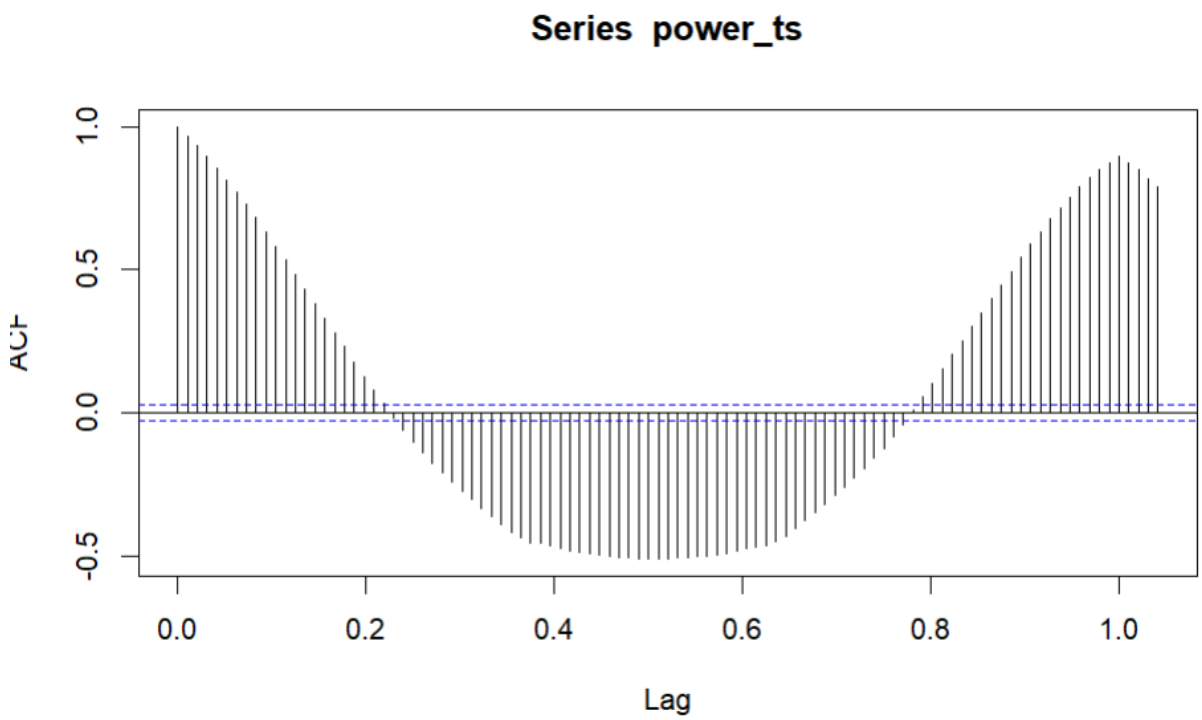
Since the data is for every 15 mins from 01-01-2010 01:30:00 to 21-02-2010 23:45:00.

24 hours * 60 mins /hour = 1440
1440 / 15 = 96

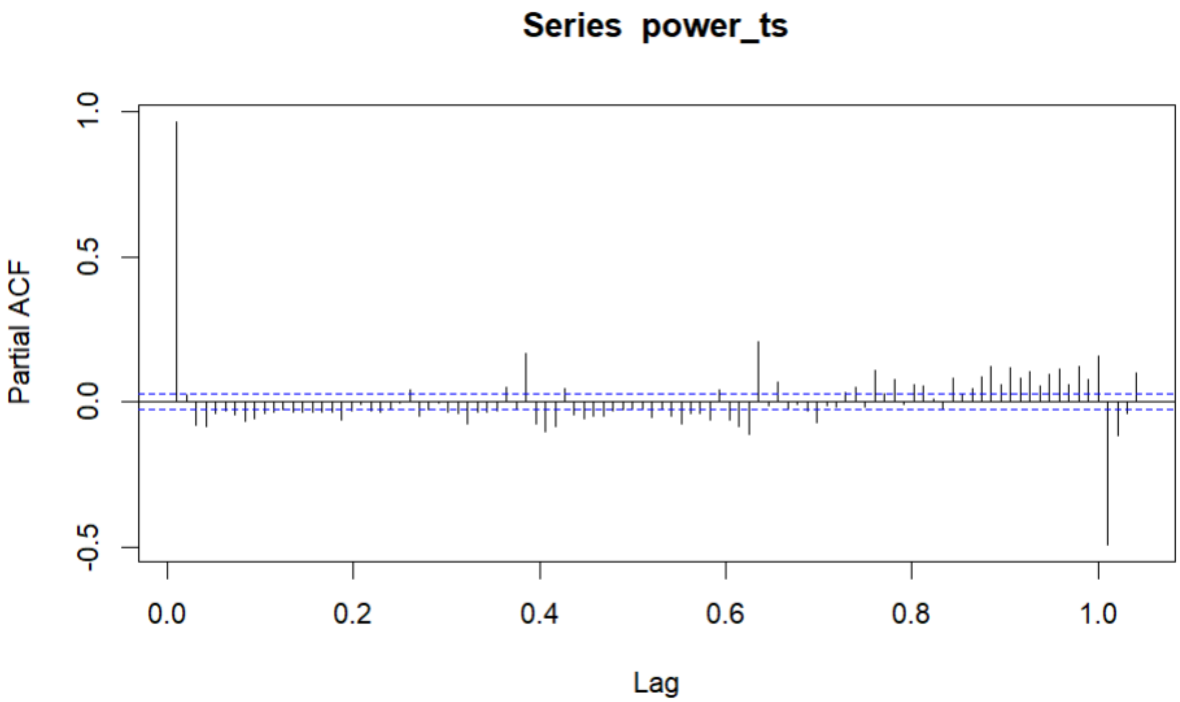
For seasonal cycle is 1 day with 96 frequency of observation

```
# Convert to time series
power_ts <- ts(elec_train$Power_kw, frequency = 96)

# ACF and PACF to check seasonality for power
acf(power_ts, lag.max = 100)
pacf(power_ts, lag.max = 100)
```



The sinusoidal pattern in the ACF suggests **seasonality** in the data.
SARMIA model might work well in this data

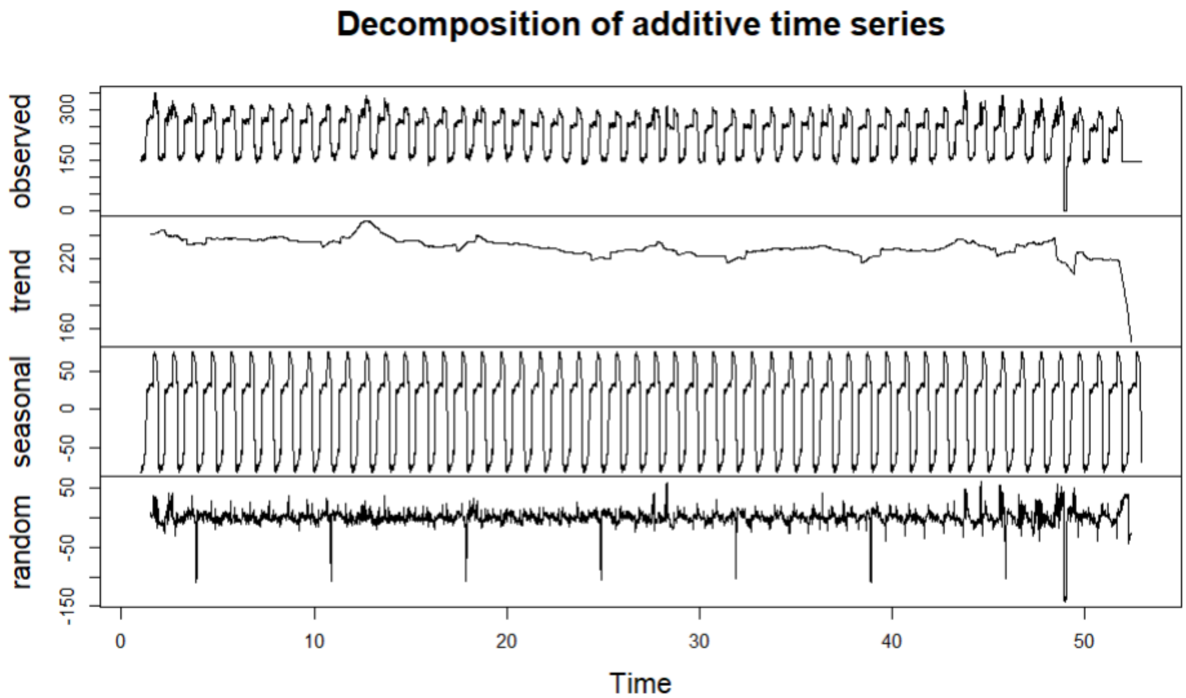


The strong spike at lag 1 suggests that **an AR(1) term** may be useful.

7. Seasonality Check

```
print(frequency(power_ts))

decomposition <- decompose(power_ts, type = "additive")
plot(decomposition)
```



This confirms the seasonality exists

8. **Stationarity Check**:

Use the Augmented Dickey-Fuller (ADF) test to confirm whether differencing is needed

```
library(tseries)

adf_test <- adf.test(elec_train$Power_kw, alternative = "stationary")
print(adf_test)

```

```
Warning: p-value smaller than printed p-value
        Augmented Dickey-Fuller Test

data:  elec_train$Power_kw
Dickey-Fuller = -15.072, Lag order = 17, p-value = 0.01
alternative hypothesis: stationary
```

This confirms that differencing is not required/

Same for Temp_c

```
library(tseries)

adf_test <- adf.test(elec_train$Temp_c, alternative = "stationary")
print(adf_test)

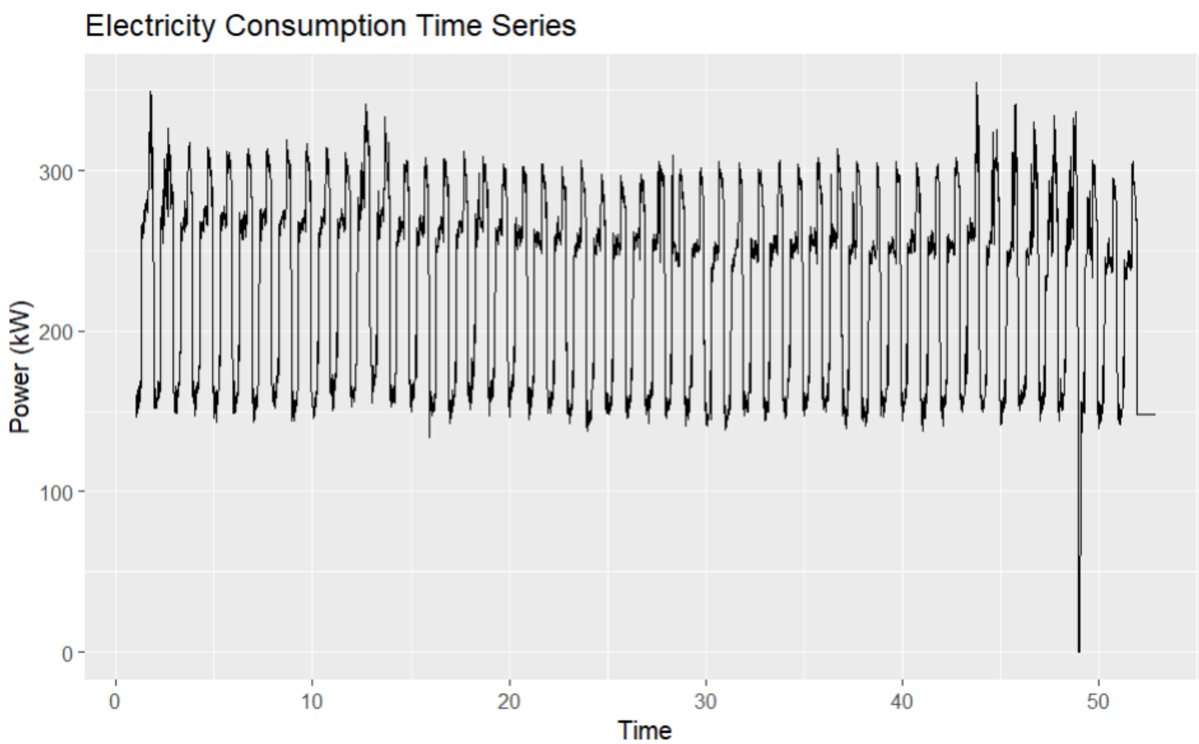
```
```
Warning: p-value smaller than printed p-value
	Augmented Dickey-Fuller Test

data:  elec_train$Temp_c
Dickey-Fuller = -11.899, Lag order = 17, p-value = 0.01
alternative hypothesis: stationary
```

No differencing is required.

9. Plotting target variable

```
autoplot(power_ts) +
  ggtitle("Electricity Consumption Time Series") +
  xlab("Time") +
  ylab("Power (kW)")
```



10. tslm check

Below we checked the relationship with the independent variable
and Temp_c plays a significant role

Also the R square value increases from 0.61(data) to 0.64(data+trend) to 0.90 (data+trend+season)
Here seasonality play a major role

```
fit=tslm(power_ts~Temp_c+Hour+Day,data=elec_train)
summary(fit)
```

```
Call:
tslm(formula = power_ts ~ Temp_c + Hour + Day, data = elec_train)

Residuals:
     Min       1Q   Median       3Q      Max 
-176.274  -16.966    1.733   23.645  116.141 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 125.86083    2.51917  49.961  < 2e-16 ***
Temp_c        4.02030    0.21159  19.000  < 2e-16 ***
Hour          5.89209    0.08153  72.273  < 2e-16 ***
Day          -2.06859    0.26008  -7.954 2.23e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 36.72 on 4982 degrees of freedom
Multiple R-squared:  0.6133,    Adjusted R-squared:  0.6131 
F-statistic:  2634 on 3 and 4982 DF,  p-value: < 2.2e-16
```

```
fit=tslm(power_ts~Temp_c+Hour+Day+trend,data=elec_train)
summary(fit)
```

```
Call:
tslm(formula = power_ts ~ Temp_c + Hour + Day + trend, data = elec_train)

Residuals:
     Min       1Q   Median       3Q      Max
-163.257  -16.308    1.264   22.680  120.817

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.318e+02  2.444e+00  53.920  < 2e-16 ***
Temp_c       5.157e+00  2.116e-01  24.366  < 2e-16 ***
Hour         5.747e+00  7.884e-02  72.895  < 2e-16 ***
Day         -1.774e+00  2.509e-01  -7.072 1.74e-12 ***
trend       -7.171e-03  3.616e-04 -19.829  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 35.36 on 4981 degrees of freedom
Multiple R-squared:  0.6416,	Adjusted R-squared:  0.6413
F-statistic:  2229 on 4 and 4981 DF,  p-value: < 2.2e-16
```

With season

```
fit=tslm(power_ts~Temp_c+Hour+Day+trend+season,data=elec_train)
summary(fit)
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.34 on 4887 degrees of freedom
Multiple R-squared:  0.9054,	Adjusted R-squared:  0.9035
F-statistic: 477.4 on 98 and 4887 DF,  p-value: < 2.2e-16
```

11. Without and With Temperature Testing different Models - HoltWinter , RandomForest, SARIMA, VAR

```r
cat("Final Results:\n")
cat("Holt-Winters RMSE (without temperature):", rmse_hw_add, "\n")
cat("SARIMA RMSE (without temperature):", rmse_sarima, "\n")

cat("SARIMA RMSE (with temperature):", rmse_sarima_temp, "\n")
cat("Random Forest RMSE (with temperature):", rmse_rf, "\n")
cat("VAR RMSE (with temperature):", rmse_var, "\n")

cat("XGBoost RMSE ( with temperature):", rmse_xgb, "\n")

```

```
Warning: longer object length is not a multiple of shorter object lengthFinal Results:
Holt-Winters RMSE (without temperature): 87.85393
SARIMA RMSE (without temperature): 88.64746
SARIMA RMSE (with temperature): 21.82396
Random Forest RMSE (with temperature): 8.033563
VAR RMSE (with temperature): 65.26381
XGBoost RMSE ( with temperature): 38.04243
```

Conclusion

We see WITHOUT temperature Holt winter and SARIMA perform close.
WITH temperature into consideration we see Random Forest and SARIMA performs well.

Final output is exported in xlsx sheet and .Rmd file is attached to the same.