From Weather to Wattage: Predictive Modelling of Power Consumption in Tetouan, Morocco

DASC 6510_02/ STAT 4990

Final Project Report

Tianle Zhong, Kai Sun, Srinidhi Ramakrishnan

Abstract

The project aims to analyze and predict the electric power consumption for Zone 1 in Tetouan, Morocco, based on time series data between January 2017 and January 2018, with sampling every 10 minutes. The analysis commences with visual mining, structural decomposition and a plot analysis to discover trends and seasonality. The data is divided into training (December 24 - 28, 2017) and test (December 29 -30, 2017) subsets. Residual checks indicate daily seasonality for benchmark models (NAIVE, Mean and Seasonal NAIVE). ARIMA models are fitted and ARIMA(2,1,1) and its seasonal counterpart ARIMA pdq(2,1,1) PDQ(0,1,1) has been selected on AICc. The forecasts are examined by Ljung-Box tests and explore residuals. Lastly, we employ a dynamic regression model with exogenous variables (temperature, humidity and date) to further improve our forecasts. The Seasonal ARIMA model has the lowest RMSE (1741) as seen with cross-validation across the models, which shows the models' energy utilization prediction quality on power consumption and demonstrates the model's effectiveness for short-term energy consumption forecasting.

1. Introduction

Urban areas experience rapidly increasing electricity demand due to population growth, industrialization, and urbanization, providing an increasing need for efficient forecasting to ensure energy sustainability and reliability. Tetouan, a city in northern Morocco with over 550,000 people, is a useful case study given its increasing energy demands and reliance on imported electricity. Amendis, a public service operator that is in charge of drinking water and electricity distribution across Morocco, plays a major role in managing the city's electricity distribution using the Supervisory Control and Data Acquisition (SCADA) system, where consumption patterns are monitored in three separate zones: Quads, Smir, and Boussafou. With ten-minute interval zone-wise samples, the data allows us to understand consumption patterns, predict the energy requirements, and adjust and manage the energy supply accordingly.

In this report, we are concerned instead with Zone 1 of Tetouan and the electric power consumption data collected there between January 2017 and January 2018. Hence, the dataset is indeed enriched with environmental factors (i.e., temperature, humidity, wind speed, and diffuse flow metrics), allowing the exogenization of variables in forecasting mode. Indeed, such short-term prediction models need to be precise and interpretable, making them suitable for decision-making in energy distribution and load balancing.

Forecasting energy consumption is crucial for the operation of a city's electricity system, and many studies have been conducted on this topic[1][2][3].

In this report, we try to use time series models to forecast the energy consumption in this area. The objective of this report is to evaluate five time series models: Naive, Mean, Seasonal Naive, Seasonal ARIMA, and Dynamic Regression on the energy consumption forecast.

2. Data

We obtained the data from Kaggle (https://www.kaggle.com/datasets/fedesoriano/electric-power-consumption/data). The dataset has a time series structure that records power consumption at 10-minute intervals over a year (January 2017 to December 2017), and allows for a stable basis for analysis and forecasts of

electricity consumption in Tetouan. Its high temporal resolution provides granular usage patterns, enabling us to detect short-term fluctuations (hourly/daily cycles), in addition to long-term trends and seasonal impacts. Such lack of familiarity will ultimately be key to understanding the nature of energy demand, and thus predictive modeling.

One of the main components to take advantage of this dataset is forecasting. That makes it reasonable to experiment with different time series predictive models, ranging from simple baseline models like Naive and Seasonal Naive to complicated ARIMA and dynamic regression-based approaches. Because of the periodicity in the data (e.g., daily seasonality), these methods are more effective as they can leverage such patterns to forecast the future values. The additional predictors like temperature, humidity, and wind speed makes it a more realistic approach in terms of the forecasting process where external factors are also considered for the modeling, which makes sense for electricity usage.

Focusing on the power consumed by Zone 1, the study illustrates how time series and forecasting methods can be utilized to discover actionable insights. This information enables energy management and planning, keeping the balance of supply and demand in a city with growing energy needs. The dataset is rich in detail and is also structured as a time series, perfect for creating accurate and scalable forecasting models.

3. Methodology

The approach for power consumption analysis and forecasting at Tetouan is decoupled into multiple stages of exploratory analysis, model development, and performance evaluation. The main objective is to develop accurate predictive models for Zone 1 power consumption, both using time series techniques as well as dynamic regression with exogenous variables.

3.1 Data Preparation and Exploration

This analysis first set up the dataset as a time series for modeling purposes. We began with exploratory data analysis to figure out the trends, seasonality, and patterns from the power consumption data. The original time series was then subjected to a structural decomposition to separate the trend, seasonal, and residual components, thereby allowing potential insights into daily and weekly seasonality. However, since the ARIMA model is not capable of handling data with multiple seasonal periods, we adopted the approach described by Forcone [4] to focus our analysis only on the data from the last week in 2017.

3.2 Benchmark Models

Naive, Mean, and Seasonal Naive baseline models were developed on the training data to create a benchmark. The simplicity of these models allowed for understandable predictions that can serve as benchmarks to compare to more complex methods. The Seasonal Naive was specifically able to capture daily trends in the data. The Ljung-Box test was used for residual diagnostics to verify whether residuals were uncorrelated.

3.3 ARIMA Modeling

Zone 1's power consumption was modeled by using the Autoregressive Integrated Moving Average (ARIMA) model[5]. After testing several ARIMA parameters, those that resulted in the lowest Akaike Information Criterion corrected (AICc) were chosen, which leads us to the selected model ARIMA pdq(2,1,1) seasonal PDQ(0,1,1). All of the forecasting plots and diagnostics were used to confirm the adequacy of the model.

3.4 Dynamic Regression

A dynamic regression model was constructed based on the power consumption time series together with its exogenous variables (temperature and humidity). Dynamic ARIMA modeling used external predictors for improved forecasting. Residual diagnostics, including the Ljung-Box test, were employed to evaluate the forecasts and ensure that the model captured relevant information without leaving significant correlations in the residuals.

3.5 Cross-Validation and Performance Evaluation

Cross-validation was performed to compare the benchmark models, ARIMA, and dynamic regression performance. Forecast accuracy was assessed using metrics such as Root Mean Square Error (RMSE). In this case, the SARIMA model reached the minimum RMSE of 1741, therefore it was capable of predicting the power consumption in the short term for Zone 1 better than other approaches.

This systematic approach facilitates comprehensive scrutiny of the data, utilizing advanced modeling techniques to produce accurate forecasts and provide meaningful insights for urban energy management.

4. Result

4.1 Time series cross-validation results on training set

Table 1 is the RMSE of five models evaluated by time series cross-validation. We can see that the Naive model achieved the highest RMSE of 7047, and the Seasonal ARIMA model obtained the lowest RMSE of 1741.

Model	Naive	Mean	Seasonal Naive	Seasonal ARIMA	Dynamic Regression
RMSE	7047	6829	6792	1741	3889

Table 1. RMSE Comparison for Models

4.2 Model result on test set

Figure 1 shows the next-day predictions of Zone 1 power consumption using three benchmark models. It is evident from the figure that none of these models capture the underlying patterns or fluctuations in power consumption. The p-value of 0 for all three models from the Ljung-Box test also indicates significant autocorrelation in the residuals, further confirming the limitations of these models.

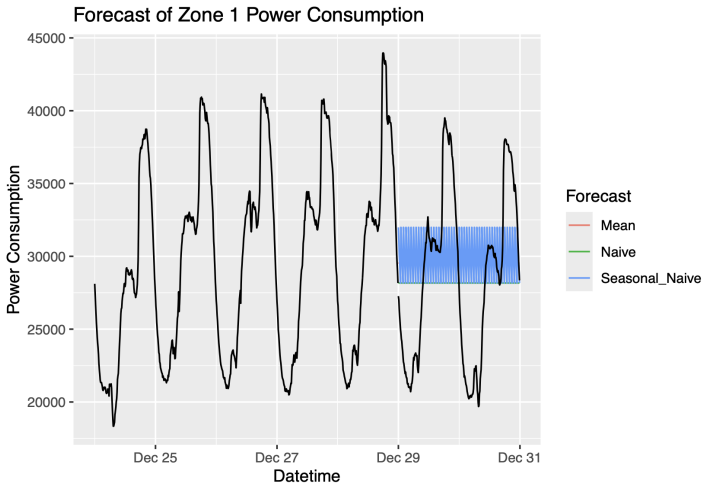


Fig 1. Actual vs. Predicted Zone 1 Power Consumption using three benchmark models

Figure 2 shows a comparison of the actual (black line) and predicted (blue line) zone 1 electric power consumption using the seasonal ARIMA model. Unlike the benchmark models, the seasonal ARIMA effectively captures the observed patterns and seasonal fluctuations in power consumption. In addition, the ljung-box test yields a p-value of 0.11, indicating that the residuals are white noise and confirming that the model captures the data's structure well.

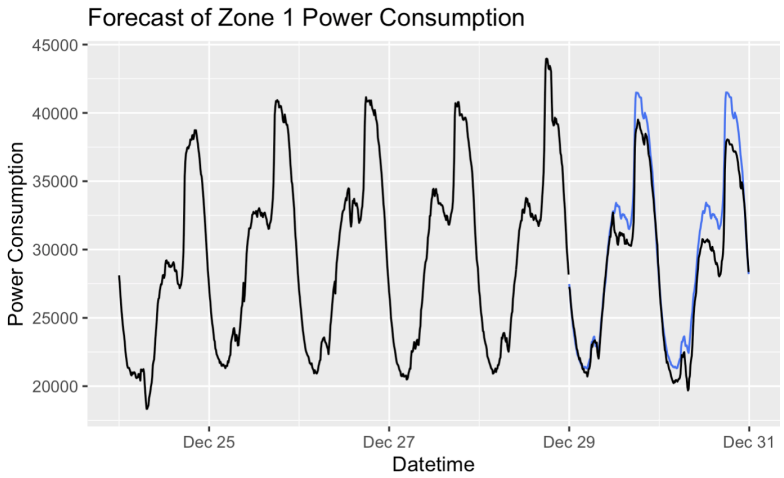


Fig 2. Actual vs. Predicted Zone 1 Power Consumption using SARIMA model

Figure 3 illustrates the comparison between actual zone 1 electric power consumption (red line) and the predicted values (blue line) generated by the Dynamic Regression model. Overall, the model has captured the pattern and variation of the data well. Although we can observe overestimation of the peak values in the last two days. Additionally, the Ljung-Box test returns a p-value of 0.53, indicating that the residuals exhibit white noise, confirming that the model successfully captures the underlying structure of the data.

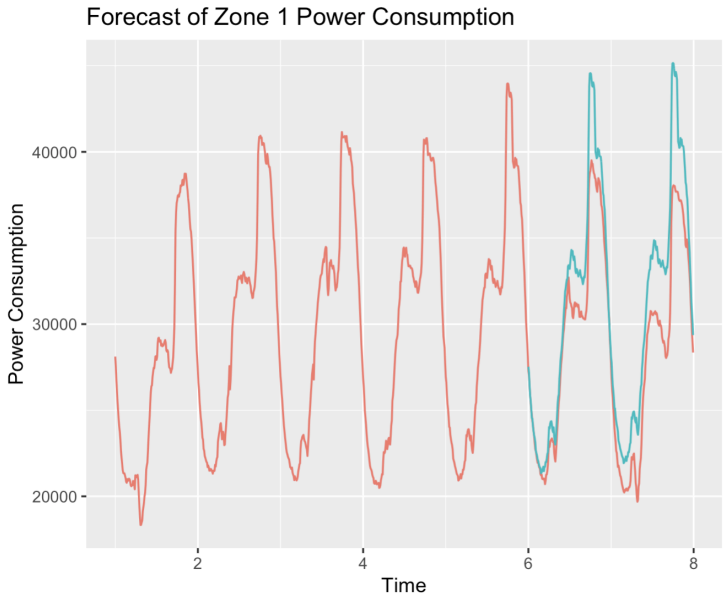


Fig 3. Actual vs. Predicted Zone 1 Power Consumption using Dynamic Regression model

Overall, the Seasonal ARIMA model showed the best performance for the forecast.

5. Conclusion

This study aimed to exploit time series analysis and forecasting techniques to forecast electric power consumption data for Zone 1 of Tetouan (Morocco), with a high-resolution dataset (10-min interval) collected during one year. Using ARIMA and dynamic regression, the analysis adjusts for natural seasonality, trends, and external environmental covariates (temperature and humidity) using the data until December 2017. The study-based findings have real-world implications and offer an intuition for understanding the aspects affecting energy demand across various external factors in an urban environment.

The research started with exploratory data analysis and identification of important seasonality and trends through a systematic workflow. The structural decomposition showed daily and weekly periodicities, and these seasonal components can help to improve the results of forecasting models. Basic time series models (Naive, Mean, and Seasonal Naive) provided a basic understanding of the dataset but did not demonstrate the capability to capture the complexity of the time series data, as indicated by high residual autocorrelation and poor performance metrics.

ARIMA modeling is key to enhance the forecast, with the Seasonal ARIMA (SARIMA) model achieving the best result, having RMSE as low as 1741 in cross-validation compared to other approaches. The residuals of the SARIMA model forecasts resembled white noise closely, indicating that the complex seasonal and trend components of the data were captured well. This result highlights the suitability of SARIMA for energy forecasts in heterogeneous environments with pronounced seasonal behavior, which we anticipate to occur in short-term forecasts.

When exogenous predictors of usage were included, such as temperature and humidity, dynamic regression models added even more value and depth to the analysis by incorporating these factors from the real world that affect electricity demand. While the dynamic regression model was able to capture broad consumption patterns and adjust for outer influences, such models had

a tendency to underestimate the peak. Despite the boundaries of training data, the model did an excellent job with an RMSE of 3889 and showed the opportunity for further optimization.
The results of this study highlight the need for the assessment of energy consumption data to choose suitable models for its specific characteristics. SARIMA illustrates the advantage of harnessing seasonality in the data for accurate forecasting, whereas dynamic regression shows the value of incorporating external factors to improve prediction. This dual method provides a powerful framework for urban energy planning, integrating the merits of classical time series methods with the adaptability of regression-based forecasting.
Future scope for this work could be to explore combined models, such as an ensemble of SARIMA and dynamic regression, or the introduction of additional exogenous factors, such as macroeconomic indicators or social activities that may impact consumption behavior. Moreover, as renewable energy sources become increasingly integrated into energy systems, the approaches developed in this work can be extended to account for the richness of hybrid energy grid management. This research serves to enhance data-driven models that can assist in sustainable solutions for cities like Tetouan, with ever-increasing energy requirements by providing accurate and interpretable predictability.

Reference

1. Bianco, V., Manca, O., & Nardini, S. (2013). Linear regression models to forecast electricity consumption in Italy. Energy Sources, Part B: Economics, Planning, and Policy, 8(1), 86-93.
2. Kaur, H., & Ahuja, S. (2017). Time series analysis and prediction of electricity consumption of health care institutions using the ARIMA model. In Proceedings of Sixth International Conference on Soft Computing for Problem Solving: SocProS 2016, Volume 2 (pp. 347-358). Springer Singapore.
3. Mahia, F., Dey, A. R., Masud, M. A., & Mahmud, M. S. (2019, December). Forecasting electricity consumption using the ARIMA model. In 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-6). IEEE.
4. Forcone MV. Power consumption forecasting with Sarima & TBATS [Internet]. Kaggle; 2023 [cited 2024 Dec 2]. Available from: https://www.kaggle.com/code/mariavirginiaforcone/power-consumption-forecasting-with-sarima-tbats/notebook
5. Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.

Github Link of the Code and Data
https://github.com/onmywaysh/timeseries

6510 Project

Tianle Zhong, Kai Sun

2024-11-28

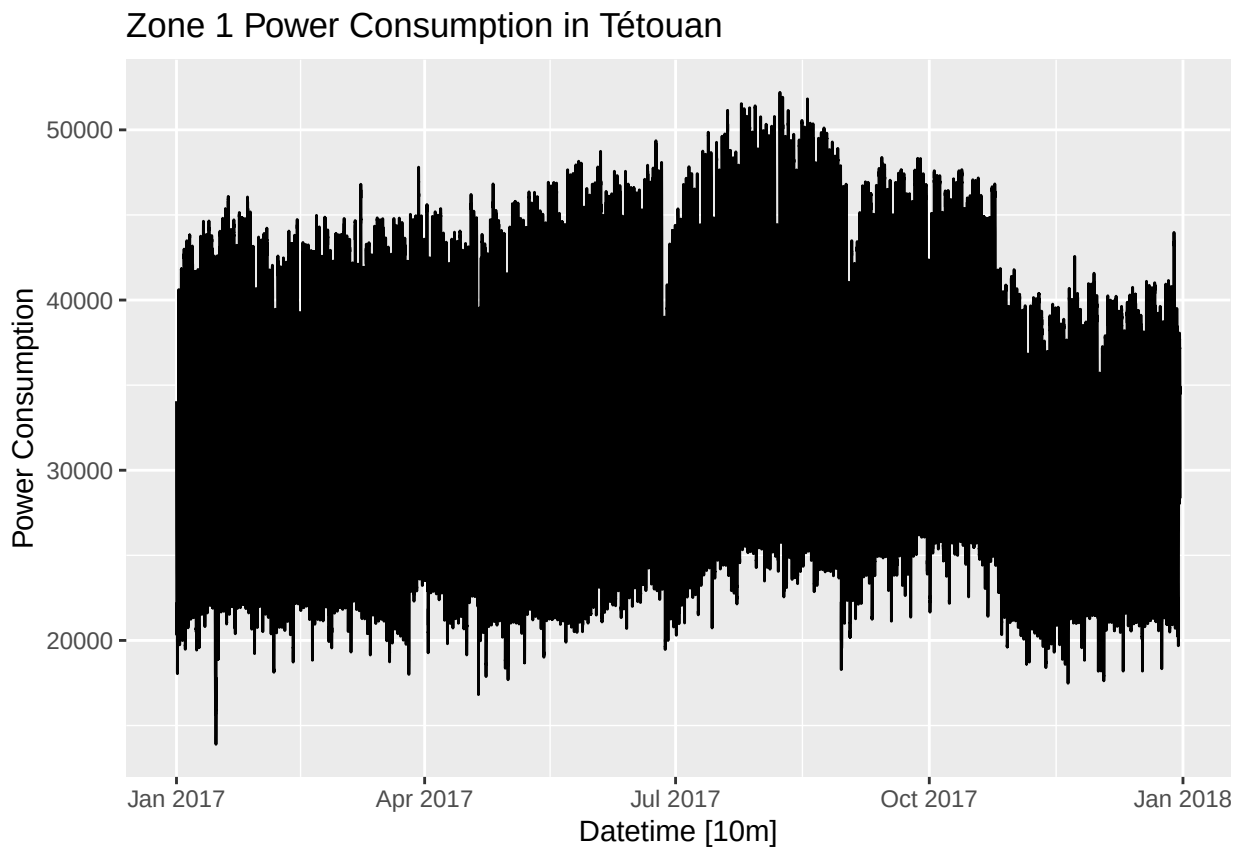
```
library(readxl)
library(lubridate)
library(fpp3)
power <- read_excel("powerconsumption.xlsx")
head(power)
```

```
## # A tibble: 6 x 9
##   Datetime      Temperature Humidity WindSpeed GeneralDiffuseFlows DiffuseFlows
##   <chr>               <dbl>    <dbl>     <dbl>               <dbl>        <dbl>
## 1 1/1/2017 0:00        6.56     73.8     0.083               0.051        0.119
## 2 1/1/2017 0:10        6.41     74.5     0.083               0.07         0.085
## 3 1/1/2017 0:20        6.31     74.5     0.08                0.062        0.1
## 4 1/1/2017 0:30        6.12     75       0.083               0.091        0.096
## 5 1/1/2017 0:40        5.92     75.7     0.081               0.048        0.085
## 6 1/1/2017 0:50        5.85     76.9     0.081               0.059        0.108
## # i 3 more variables: PowerConsumption_Zone1 <dbl>,
## #   PowerConsumption_Zone2 <dbl>, PowerConsumption_Zone3 <dbl>
```

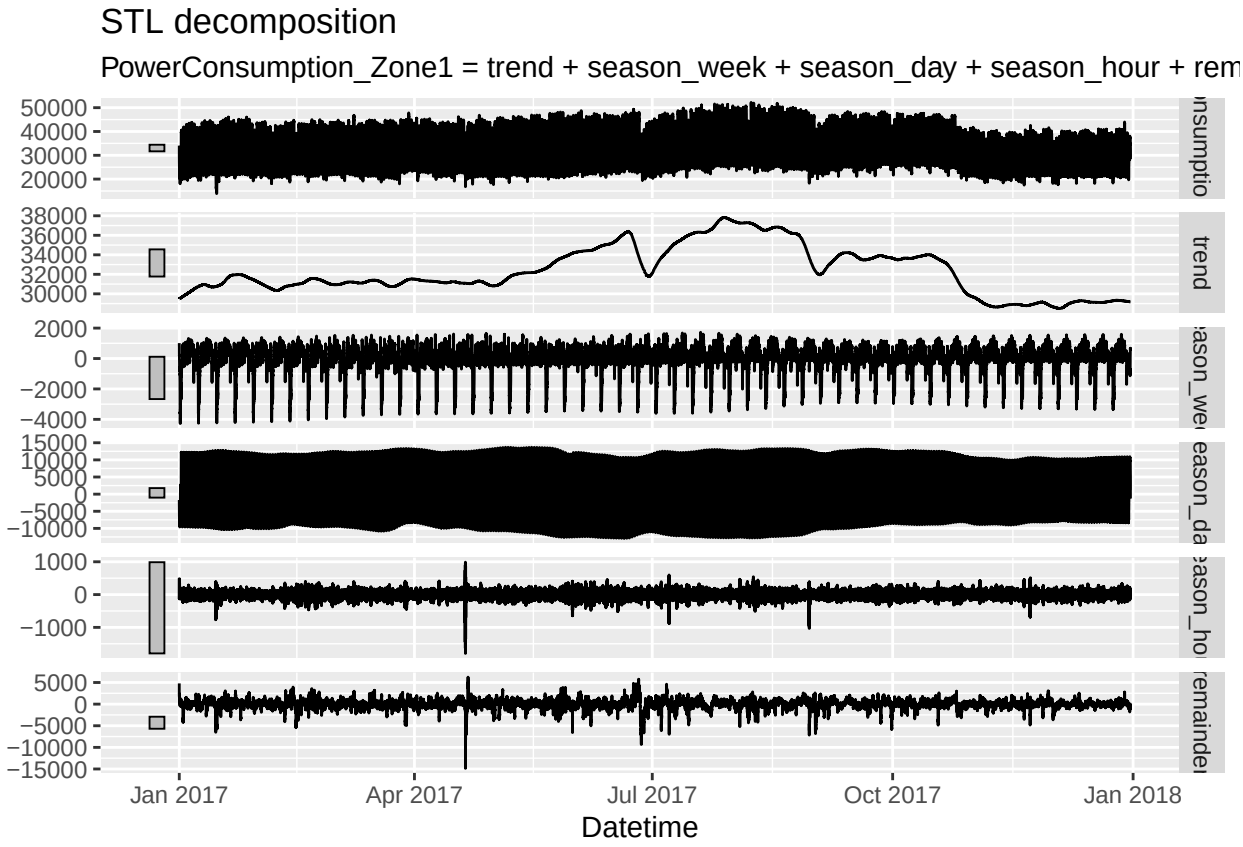
```
#convert the Datetime column to a proper date-time format
power <- power %>%
  mutate(Datetime = mdy_hm(Datetime)) %>%
  as_tsibble(index = Datetime)

#check the missing values
colSums(is.na(power))
```

```
##                Datetime             Temperature                Humidity
##                       0                       0                       0
##               WindSpeed    GeneralDiffuseFlows           DiffuseFlows
##                       0                       0                       0
## PowerConsumption_Zone1 PowerConsumption_Zone2 PowerConsumption_Zone3
##                       0                       0                       0
```

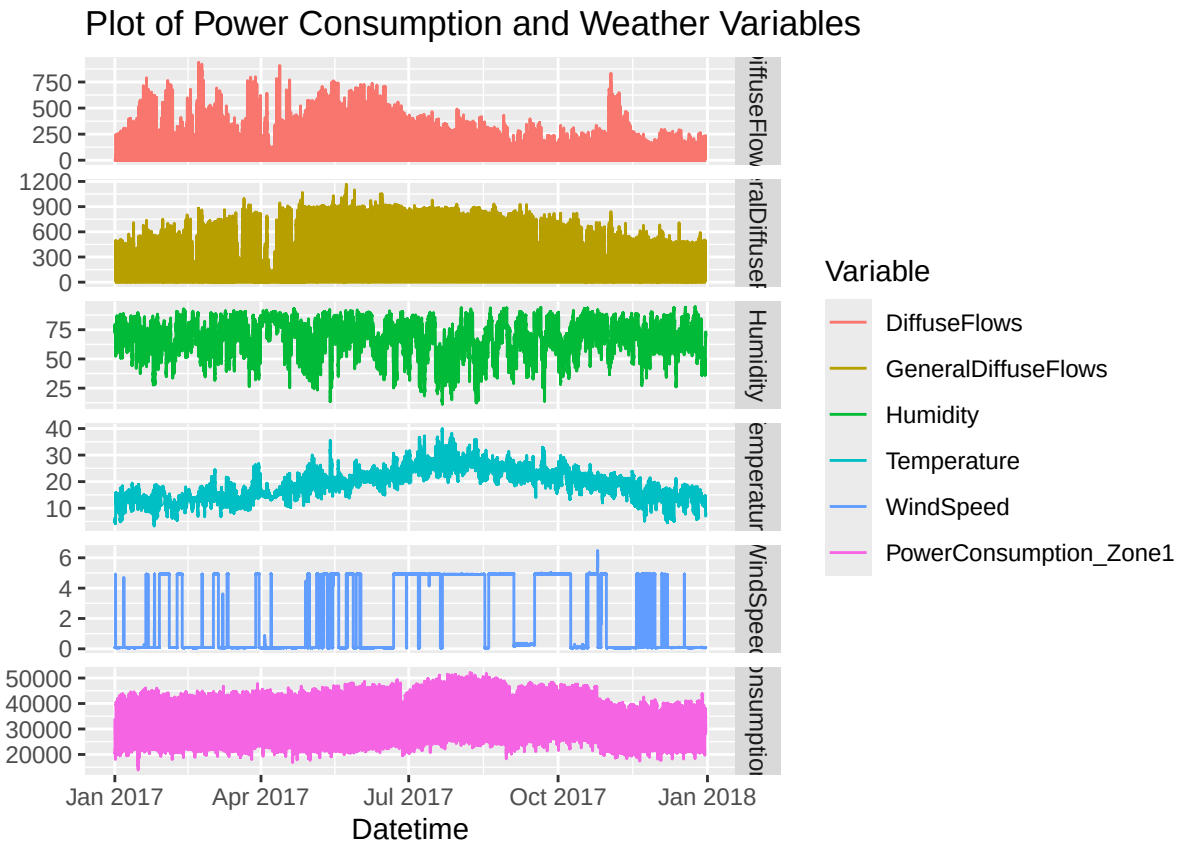
```
#explore the data
power %>% autoplot(PowerConsumption_Zone1)+
  labs(title = "Zone 1 Power Consumption in Tétouan", y = "Power Consumption")
```




```
#decomposition
dcmp <- power %>%
  model(stl = STL(PowerConsumption_Zone1))
components(dcmp) %>% autoplot()
```

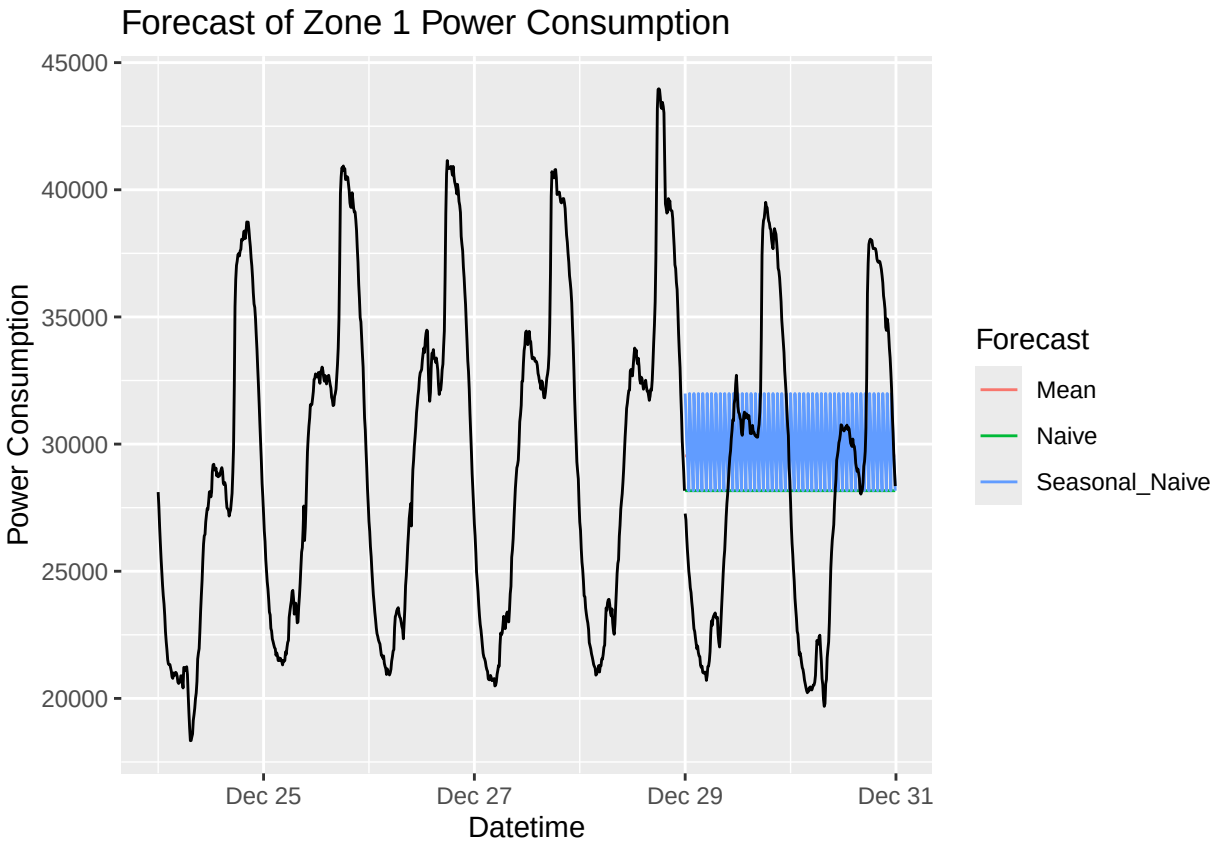



```
## visualize the relationship between power consumption and weather variables
power %>%
  pivot_longer(c(PowerConsumption_Zone1, Temperature, Humidity, WindSpeed, GeneralDiffuseFlows, Diffusel
  mutate(name = factor(name, levels = c("DiffuseFlows", "GeneralDiffuseFlows", "Humidity", "Temperature
  ggplot(aes(x = Datetime, y = value, color=name)) + geom_line() +
  facet_grid(name ~ ., scales = "free_y") + ylab("")+
  labs(y = "", color = "Variable") +
  ggtitle("Plot of Power Consumption and Weather Variables")
```




```
#Due to the computational runtime, we have decided to focus only on the data from the last week
power <- power %>%
  filter(Datetime>=as.POSIXct("2017-12-24 00:00:00", tz = "Africa/Casablanca"))
train <- power %>%
  filter(Datetime>=as.POSIXct("2017-12-24 00:00:00", tz = "Africa/Casablanca") &
           Datetime<=as.POSIXct("2017-12-28 23:59:59", tz = "Africa/Casablanca"))
test <- power %>%
  filter(Datetime>=as.POSIXct("2017-12-29 00:00:00", tz = "Africa/Casablanca"))
```

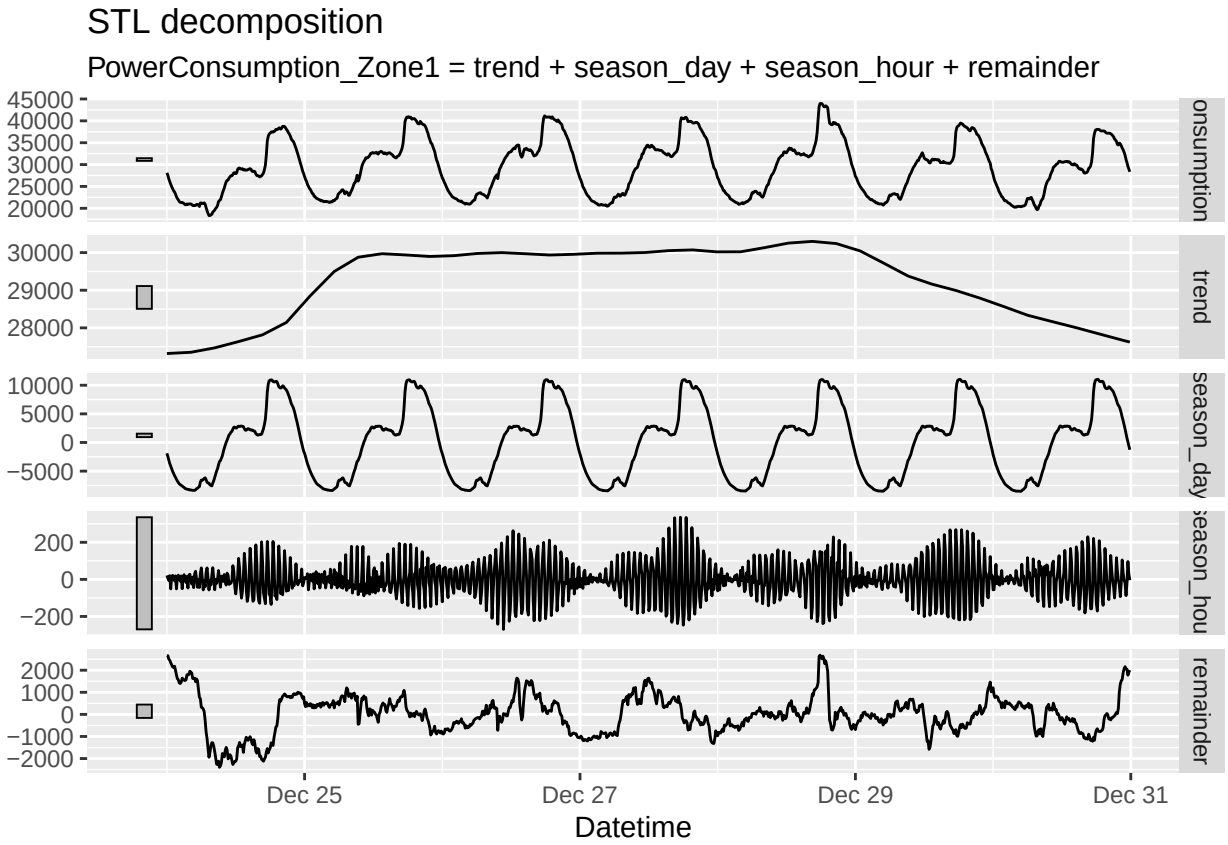
```
#fit the benchmark models (mean,naive,snaive)
benchmark_fit <- train %>%
  model(
    Mean = MEAN(PowerConsumption_Zone1),
    Naive = NAIVE(PowerConsumption_Zone1),
    Seasonal_Naive = SNAIVE(PowerConsumption_Zone1)
)
#forecast
benchmark_fc <- benchmark_fit %>%
  forecast(new_data = test)
#plot the forecasts
benchmark_fc %>%
  autoplot(train, level = NULL) +
  autolayer(test, PowerConsumption_Zone1, colour = "black")+
  labs(y = "Power Consumption",
       title = "Forecast of Zone 1 Power Consumption")+
  guides(colour = guide_legend(title = "Forecast"))
```




```
#residual check
aug <- benchmark_fit %>%
  augment()
aug %>%
  features(.innov, ljung_box, lag = 10)
```

```
## # A tibble: 3 x 3
##   .model         lb_stat lb_pvalue
##   <chr>            <dbl>     <dbl>
## 1 Mean             6216.         0
## 2 Naive            1123.         0
## 3 Seasonal_Naive   2848.         0
```

```
#the data shows daily seasonality
dcmp2 <- power %>%
  model(stl = STL(PowerConsumption_Zone1))
components(dcmp2) %>% autoplot()
```




```
#fit a ARIMA model
fit <- train %>%
  model(ARIMA(PowerConsumption_Zone1))
report(fit)
```

```
## Series: PowerConsumption_Zone1
## Model: ARIMA(2,1,1)
##
## Coefficients:
##          ar1      ar2      ma1
##       1.4313  -0.4775  -0.7680
## s.e.  0.1163   0.0931   0.1035
##
## sigma^2 estimated as 139167:  log likelihood=-5276.82
## AIC=10561.64   AICc=10561.69   BIC=10579.95
```

```
# try some other models by adding the seasonality
fit <- train %>%
  model(
    arima1 = ARIMA(PowerConsumption_Zone1 ~ pdq(2, 1, 1)),
    arima2 = ARIMA(PowerConsumption_Zone1 ~ pdq(2, 1, 1)+ PDQ(0, 1, 0, period = 144)),
    arima3 = ARIMA(PowerConsumption_Zone1 ~ pdq(2, 1, 1)+ PDQ(1, 1, 0, period = 144)),
    arima4 =ARIMA(PowerConsumption_Zone1 ~ pdq(2, 1, 1)+ PDQ(0, 1, 1, period = 144))
    )
```

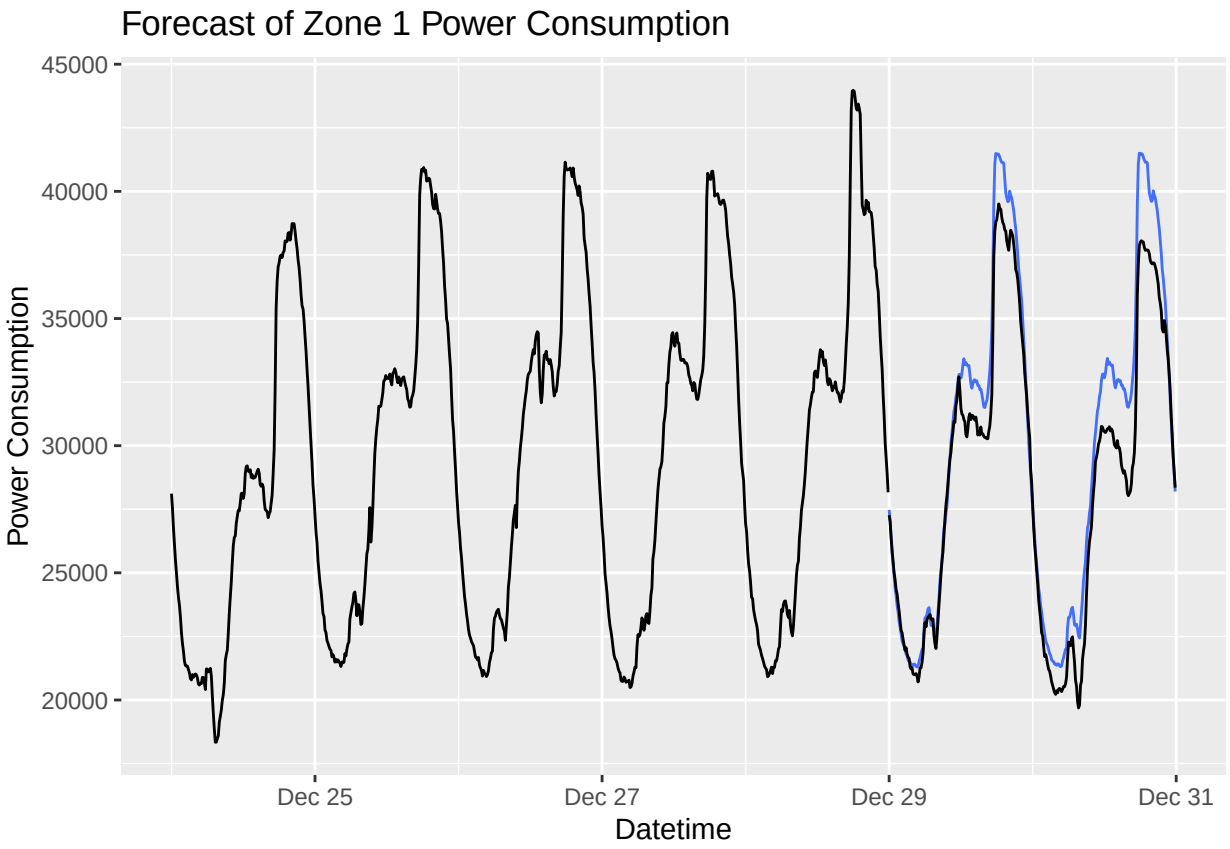
```
## Warning: It looks like you're trying to fully specify your ARIMA model but have not said if a constar
```

```
## You can include a constant using ‘ARIMA(y~1)‘ to the formula or exclude it by adding ‘ARIMA(y~0)‘.

## Warning: 1 error encountered for arima3
## [1] Could not find an appropriate ARIMA model.
## This is likely because automatic selection does not select models with characteristic roots that may
## For more details, refer to https://otexts.com/fpp3/arima-r.html#plotting-the-characteristic-roots
```

```
fit %>%
  glance() %>%
  arrange(AICc) %>%
  select(.model, AICc) #we see that arima3 fails to generate
```

```
## # A tibble: 3 x 2
##   .model   AICc
##   <chr>   <dbl>
## 1 arima4  8223.
## 2 arima2  8373.
## 3 arima1 10562.
```

```
#use the best arima model
arima_fit <- train %>%
  model(ARIMA(PowerConsumption_Zone1 ~ pdq(2, 1, 1)+ PDQ(0, 1, 1, period = 144)))

#forecast
arima_fc <- arima_fit %>%
  forecast(new_data = test)
#plot the forecasts
arima_fc %>%
  autoplot(train, level = NULL) +
  autolayer(test, PowerConsumption_Zone1, colour = "black")+
  labs(y = "Power Consumption",
       title = "Forecast of Zone 1 Power Consumption")+
  guides(colour = guide_legend(title = "Forecast"))
```




```
#residual check
aug3 <- arima_fit %>%
  augment()
aug3 %>%
  features(.innov, ljung_box, lag = 10)
```

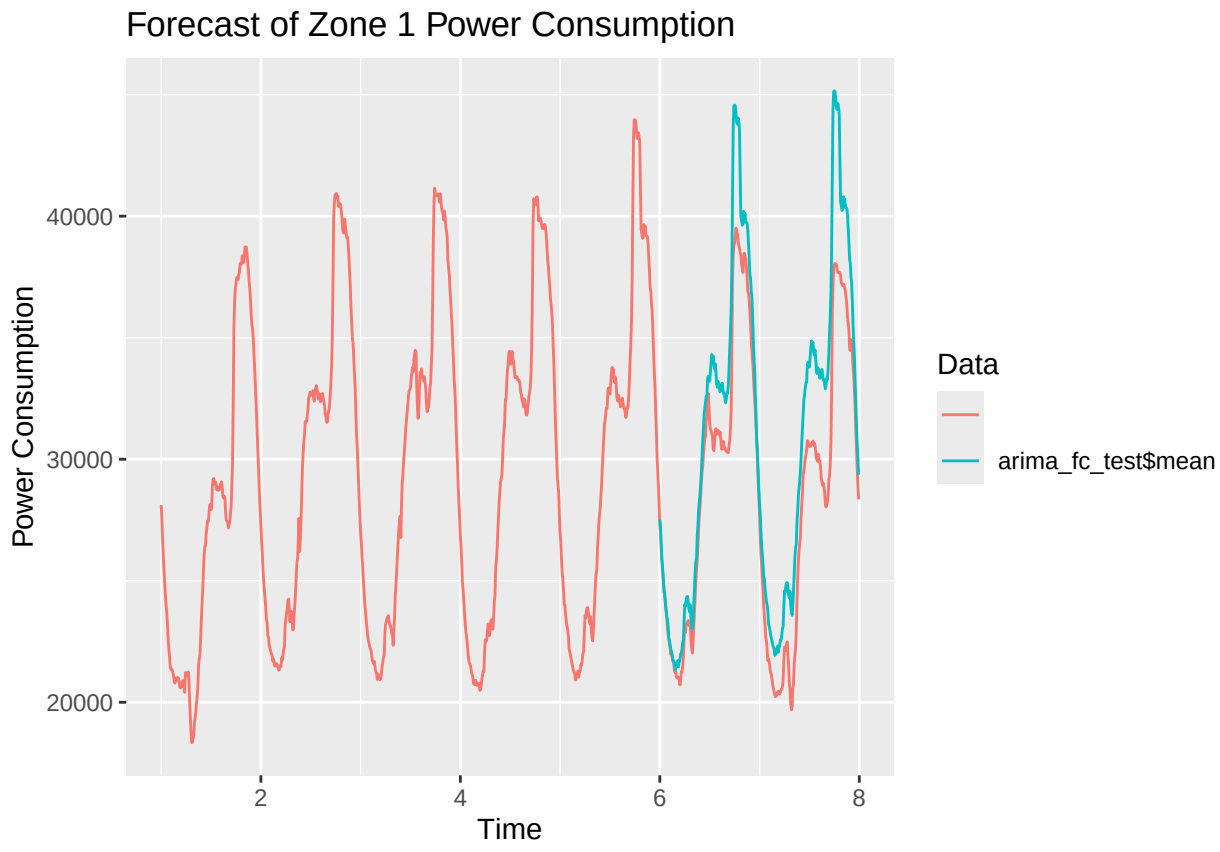
```
## # A tibble: 1 x 3
##   .model                                                  lb_stat lb_pvalue
##   <chr>                                                     <dbl>     <dbl>
## 1 ARIMA(PowerConsumption_Zone1 ~ pdq(2, 1, 1) + PDQ(0, 1, 1, ~    15.8     0.106
```

```
#fit a dynamic regression
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
data_train <- train[, c("Datetime", "Temperature", "Humidity", "PowerConsumption_Zone1")]
test_data <- test[, c("Datetime", "Temperature", "Humidity", "PowerConsumption_Zone1")]

# set time series data
y <- ts(train$PowerConsumption_Zone1, frequency = 144)
xreg <- as.matrix(data_train[, c("Temperature", "Humidity")])
```

```r
test_xreg_future<-as.matrix(test_data[, c("Temperature", "Humidity")])

arima_fit_test <- auto.arima(y, xreg = xreg)
# use test data to forecast
arima_fc_test <- forecast(arima_fit_test,
                      xreg =test_xreg_future)

# plot results
autoplot(ts(power$PowerConsumption_Zone1, frequency = 144), series = " ") +
  autolayer(arima_fc_test$mean) +
  labs(
    y = "Power Consumption",
    title = "Forecast of Zone 1 Power Consumption"
  ) +
  guides(colour = guide_legend(title = "Data"))
```




```r
#A simple test for Ljung-Box
residuals<-residuals(arima_fit_test)
Box.test(residuals, lag = 20, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 18.875, df = 20, p-value = 0.53
```

```r
#cross validation for three benchmark methods and ARIMA
cv_stretch <- train %>%
  stretch_tsibble(.init = 288, .step = 72) %>%
  filter(.id!=max(.id))
cv_fit <- cv_stretch %>%
  model(
    Mean = MEAN(PowerConsumption_Zone1),
    Naive = NAIVE(PowerConsumption_Zone1),
    Season_naive = SNAIVE(PowerConsumption_Zone1),
    ARIMA = ARIMA(PowerConsumption_Zone1 ~ pdq(2, 1, 1)+ PDQ(0, 1, 1, period = 144))
  )
```

```
## Warning: 2 errors (1 unique) encountered for ARIMA
## [2] Not enough data to estimate a model with those options of P and Q. Consider allowing smaller valu
```

```r
cv_forecast <- cv_fit %>% forecast(h = "1 day")
cv_forecast %>% accuracy(train)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 72 observations are missing between 2017-12-29 00:00:00 and 2017-12-29 11:50:00
```

```
## # A tibble: 4 x 10
##   .model       .type     ME  RMSE   MAE     MPE  MAPE  MASE RMSSE  ACF1
##   <chr>        <chr>  <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ARIMA        Test    72.9 1741. 1414. -0.0397  4.66 0.774 0.709 0.983
## 2 Mean         Test  1833.  6829. 5974.  1.32   19.8  3.27  2.78  0.993
## 3 Naive        Test   294.  7047. 5869. -3.83   20.4  3.21  2.87  0.996
## 4 Season_naive Test  -364.  6792. 5676. -6.15   20.2  3.11  2.76  0.979
```

```r
#cross validation for dynamic regression
# define dynamic arima
arimax_forecast <- function(y, h, xreg_train, xreg_future) {
  fit <- auto.arima(y, xreg = xreg_train)
  summary(fit)
  fc<-forecast(fit, h = h, xreg = xreg_future)
  return(tail(fc$mean, n = 72))  # return the last 72 results
}

#start CV
start_length=288
i=start_length
h_value=72

pred_store <- array(NA, dim = c(length(y)))   #array to store the result data

while (i<length(y)) {
  fc_value=arimax_forecast(y[1:i],h=h_value,xreg[1:i],xreg[(i+1):(i+h_value)])
  pred_store[(i+1):(i+h_value)]=fc_value
  i=i+h_value
}

pred=pred_store[(start_length+1):length(y)]
```

```r
actual=y[(start_length+1):length(y)]
rmse_cv=sqrt(mean((pred - actual)^2))  #calculate RMSE

print(rmse_cv)
```

```
## [1] 3889.308
```