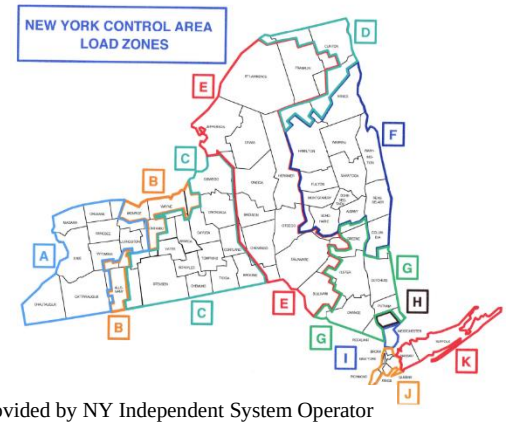
ARIMA Short-Term Load Forecast

Tyler Brown

November 22, 2023



Provided by NY Independent System Operator

Abstract

National Grid, a United Kingdom based utility organization, has territory within the New England area of the United States. Their vision is to "be at the heart of a clean, fair, and affordable energy future." With the nation pushing for zero emissions by 2050, they innovate energy generation through means other than fossil fuels. In the meantime, the company undergoes day-to-day management of their existing infrastructure for consumer utilities. In New York specifically, National Grid has jurisdiction over gas throughout parts of the state and electric only across upstate counties. National Grid utilizes 6 weather stations (one per zone) within New York state and 4 sensor values for system-level forecasts. The current weather stations used are from recognized airports, such as Buffalo (KBUF), Rochester (KROC), Syracuse (KSYR), Rome (KRME), Messena (KMSS), and Albany (KALB). The sensor values consist of temperature (TMP), feels like temperature (FLSK), humidity (HUM), and global horizontal irradiance (GHI). This information provided by the Data Transmission Network and Dataline (DTN) must be used to produce weather forecasts within each zone for electric load threshold management. While this produces decent results for zones A-F, there are alternative algorithms that can be used in modern day forecasting. It will be determined if these models are sufficient to implement National Grid's day-to-day short-term forecasting.

Keywords: *ARIMA, trend, ACF, residual, RMSE, MAPE, seasonality, forecast, white noise*

Introduction

Tyler Brown is a Robot Automation Analyst who travels across the nation assisting a variety of commercial customers with cleaning services through robotics and facilitates training to any willing party.

The Problem

National Grid is having trouble with their short-term load forecasting model. A threshold is set daily to prevent power outages or transformer overloads across New York state. When a threshold forecast is to be reached, National Grid must act prior to reaching it. The New York State regulators are cracking down on assigned thresholds being reached or surpassed and no action or notice is provided. Nation al Grid would like to know:

What modern-day model can be implemented in the day-to-day operations for short-term load forecasting?

Literature Review

For this project, I focused my research and literature review around short-term load forecasting strategies. The method will help my understanding of load as an existing predictor for future load forecasts. Research included the following subjects:

Research of Load Trends

The first researched subject involved load trends, since it is understood the usage

frequency of electricity can vary throughout the year. According to the Utilities Policy journal, Lindberg, Seljom, Madsen, Fischer, and Korpas (2019) believes the "electricity sector is only one part of the energy system; but current trends strengthen the interlinkages between the power system, and the heating and the transportation sector" (pages 102-119). The authors focused on current and future drivers that influence electricity load, regarding hourly profile and consumption level. It is their belief that temperature is within "most methodologies as an explanatory variable to account for seasonal variations". It describes recurrent daily and weekly load patterns. "Long-term forecasting considers long-term trends, unlike short-term forecasting" (Hernandez et al., 2014).

The Exponential Smoothing technique has been "modified to operate on seasonal time series directly with three separate smoothing constants for stationarity, trend and seasonality" (Takiyar and Singh et al., pp. 2). The fundamental technique is to find hidden trends and patterns from the databases created. "Conventional data decomposition algorithms extract the trend and periodic components from the original series" (Wei, Yin, C. Li, Wang, Qiao, C. Li, Zeng, and Fu et al., 2022). "Detrend singular spectrum fluctuation analysis removes the trend and periodic components to use linear and trigonometric functions" to fit the original series, then extract them from the series to get the trend and periodic components.

Research of Load Thresholds

The next researched subject involved load thresholds, as thresholds trigger a demand response. Both "increases and decreases in temperature increase the demand for electricity" (Momani et al, 2013). The "differential temperature above or below a threshold temperature" results in population response during holiday and working day. The threshold value produced can be used to "proactively trigger peak demand shaving and other demand response actions in order to reduce demand charges" (Aponte and McConky et al, 2022). It is up to consumers to set their own signal, or electricity demand threshold, at the beginning of each month so they can significantly reduce their peak load charges. If the threshold is set higher than the optimal level, the consumer will not be able to achieve all potential savings. Alternatively, if set below, the consumer could require demand response events, increasing the bill due to the inconvenience.

Peak demand alerts are "sent to each small and medium enterprise (SME) when the forecasted demand exceeds the predefined precaution threshold (Komatsu and Kimura et al, 2020). The demand alerts give the predicted time the contract demand may be exceeded. These are derived from 1-hour ahead to 24-hour ahead forecasts. "Income decile households can be considered the threshold for basic needs or a measure of electricity poverty" (He and Reiner et al, 2015). In Chinese households, there exists a threshold for electricity consumption with respect to income.

Research of Load Demand Response

"Electricity demand response refers to consumer actions that change the utility load profile in a way that reduces costs or improves grid security" (Gyamfi, Krumdieck, and Urmee et al, 2013). "Critical peak demand occurs when there is co-incident high usage among all the end use sectors; residential, industrial, and commercial." Critical peak demand risks power system failure. "The goal of Demand Response programs is to influence consumers to change their electricity consumption patterns, or demand, in response to the needs of the supplier" (Muratori, Schuelke-Leech, and Rizzoni et al, 2014).

With less electricity demand from customers, suppliers can lessen the bill by requesting less

electricity. “DR changes end-use customer electricity consumption patterns from those customers’ normal patterns in response to changes in the price of electricity over time” (Albadi and Saadany et al, 2008). Furthermore, to “incentive payments designed to induce lower electricity use at times of high wholesale market prices, or when system reliability is jeopardized”. Demand response programs have a primary goal to benefit both the consumer and supplier, resulting in a positive relationship between both parties.

Summary of Literature Review

The literature review explained the factors considered to predict load for demand response. The literature, however, did suggest utilizing additional features such as temperature and territory income. Changes in temperature can cause customers to utilize more cooling or heating appliances, requiring more electricity to their household. Higher income territories can afford an increase in payments without sacrificing comfortability, while lower income territories focus on managing a low bill from the supplier. These features could prove useful in determining the correlation between load, temperature, humidity, and more.

Methodology

As stated in the introduction, my research was focused on attempting to answer the question:

What modern-day model can be implemented in the day-to-day operations for short-term load forecasting?

My methodology for addressing this question consisted of 4 processes: Data Exploration, Data Preparation, Time Series Analysis, and ARIMA Modeling, as described below:

Data Exploration: The data has been provided by New York Independent System Operator Inc (NY ISO) from August 2001 to January 2005. I simplified the features to their zone label. Web-scraping occurred to obtain the remaining 27 attributes.

Data Preparation: The Weather Data consists of 7,680 observations and 28 total features. Only 6 features will be used for forecasting as the other features consist of forecast information. The date feature will need its data type switched to date.

Time Series Analysis: The goal in the time series process was to understand the predictors and their relationship with the response variable. As an initial method of exploration, the data was examined to look for trends and seasonality within each respective zone.

ARIMA Modeling: The description of the autocorrelations within the student enrollment data to predict future values based on past values by examining the differences between the values in the series. The multiple models will use each variable against load, a combination of variables, and BoxCox transformation versions of each model to ensure all models are presented and evaluated.

After selecting an appropriate time series model, the model will be applied to the test data set. The RMSE and MAPE will be used to evaluate the models and the selected regression model will be used to predict future load for National Grid territories. Results of models will then be compared against the actual load to determine the effectiveness of predicting utilizing the response variables.

Experimentation and Results

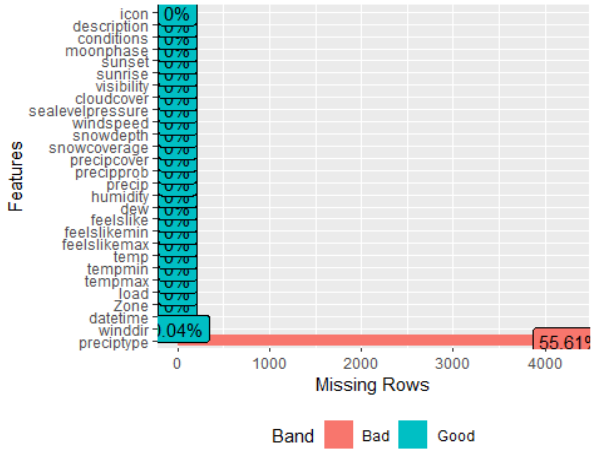
Data Exploration

The data set for my investigation contained quantitative data related to electric load in the Northern New York state counties between August 1, 2001, to January 31, 2005. There were 7,680 observations of 28 attributes, each representing affecting contributions to load.

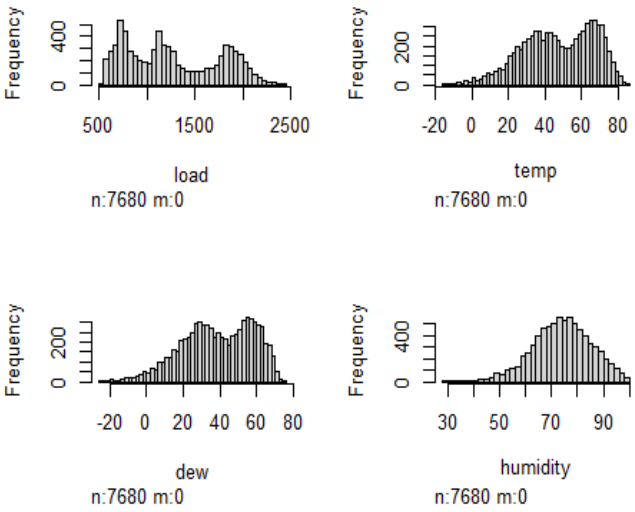
For each day, the average load (in megawatts) over a 24-hour period will be used with temperature (in Fahrenheit), dew point, and/or humidity as predictor variables. Below is a brief description of the variables within the data set:

var_name	var_desc
datetime	Date when weather was reported
Zone	Territories within jurisdiction
load	Avg electric (MW) over 24-hrs
temp	Temperature (Fahrenheit)
dew	Reported dewpoint of day
humidity	Reported humidity of day

First, I had to identify if there are any signs of missing data as shown below:



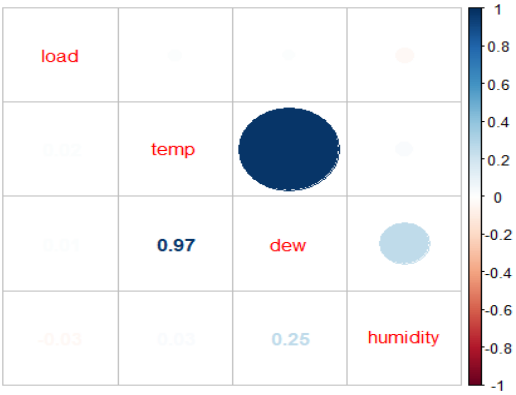
Temperature, dew point, and humidity were selected as predictor variables for the response variable (load) due to their significance in affecting electric utilization. For numerical data, I looked at the histograms to view the distribution of the selected predictor variables, as shown below:



The initial data analysis uncovered key features of the data set:

- Most recorded loads have relatively low readings
- Most temperatures are greater than 40 degrees Fahrenheit
- Most humidity levels are greater than 60
- Most dew point levels are greater than 20

Below, I visualize the correlation between all predictor and response variables:



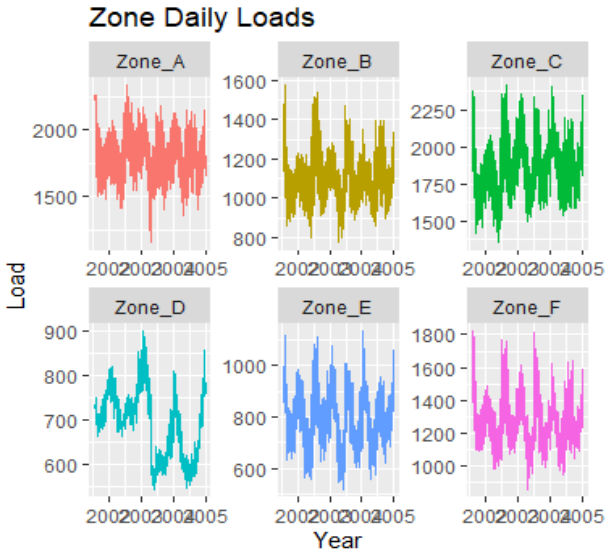
We can determine dew point has a high correlation with temperature and a moderate correlation with humidity. All other variables have little to no correlation with one another.

Data Preparation

Since the missing data falls within the variables not being used within this project, I will remove all attributes other than datetime, zone, load, temperature, dew point, and humidity. The data will need to be prepared prior to modeling. The datetime attribute will need to be changed from a character data type to a Date. This will allow the data set to be transformed as a time series. Additionally, the data will be split to compare forecast results with the actual observations. I will do this by having the training data before January 1st of 2005 and testing data for all remaining observations (Jan 2005).

Time Series Analysis

The goal in the time series process was to understand the predictors and their relationship with the response variable. As an initial method of exploration, the data was examined to look for trends and seasonality within each respective zone. To begin, I transformed the training data to a time series with the zones as the key and the datetime as the index. We can see each zone load data below:



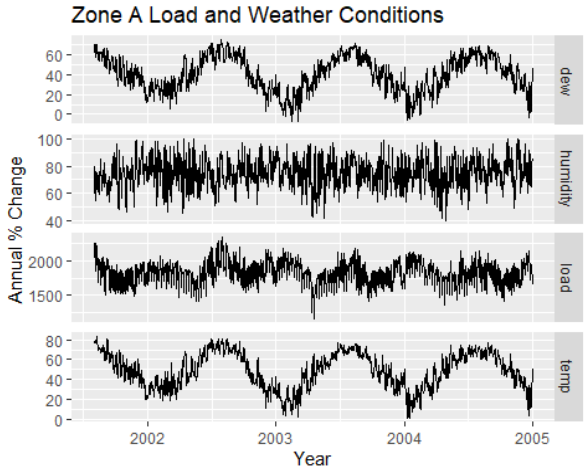
Each zone shows some signs of seasonality at the start of the year (within the winter season) and mid-year (within the summer season). This makes sense as during these seasons, citizens tend to use more electricity on heating and cooling appliances. There is no noticeable trend within the data as the data increases and decreases over the period. Looking at each zone individually will allow greater visibility.

ARIMA Modeling

In building the arima models, I created multiple iterations using the training data, and picked the one with the best fit. The models' performance metrics were subsequently compared against one another to allow us to select the ideal arima model for forecasting loads and comparing the forecasts with the test data set. Each of the models used the data set's load attribute as the dependent response variable, while the remaining were independent predictor variables. The most successful approach involved a low Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), as these are metrics the organization looks for when modeling. The multiple models will use each variable against load, a combination of variables, and BoxCox transformation versions of each model to ensure all models are presented and evaluated.

As shown below, the predictor and response variables are plotted with an annual percent change separately for Zone A:

Zone A



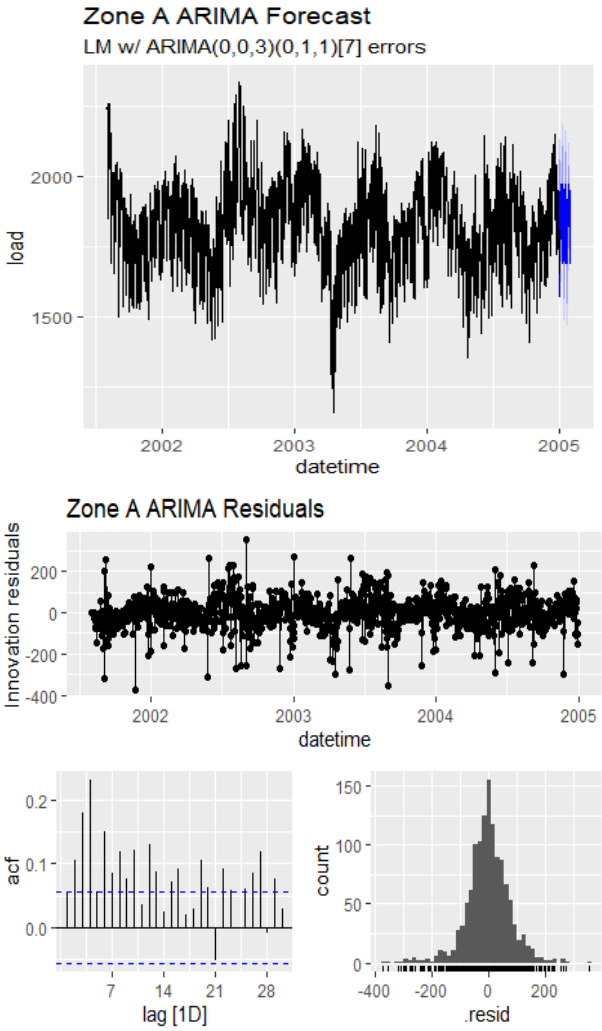
Viewing the plots, I have determined there is seasonality within dew point and temperature, as both are low at the start and high at mid-year. Humidity and load do not show immediate signs of seasonality. No variable shows trends from the plots.

.model	RMSE	MAPE
arima5	75.8	3.00
arima3	76.0	3.01
arima4	76.1	3.01

As shown, Zone A ran 16 models, but the ARIMA with load, temperature, dew point, and humidity without BoxCox transformation was deemed ideal. The RMSE is 75.82 and MAPE is 3.00. The ARIMA model is still to be determined.

Series: load Model: LM w/ ARIMA(0,0,3)(0,1,1)[7] errors

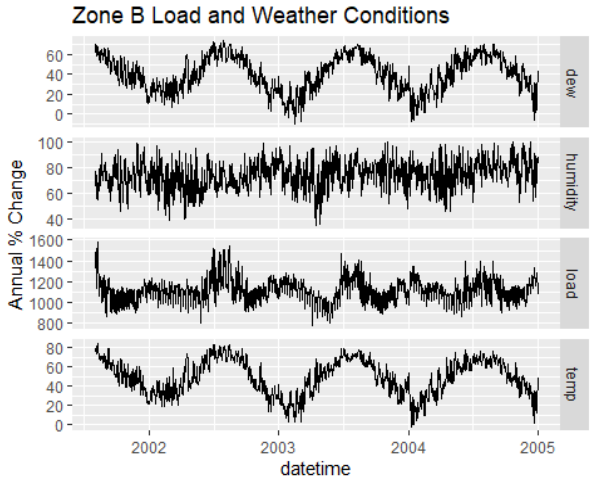
The ARIMA model selected was LM w/ ARIMA(0,0,3)(0,1,1)[7] errors. With the model, we can forecast January 2005.



.model	lb_stat	lb_pvalue
arima5	233.	0

As we can see, the model fits well with the training data. The residuals are normally distributed and do not resemble that of white noise, as less than 95% of the spikes in the ACF do not lie within the blue dashed boundary lines. Through the Ljung-Box test, I rejected the null hypothesis as the p-value is less than 0.05. With Zone A modeled, we will begin the same process for Zones B, C, D, E, and F.

Zone B



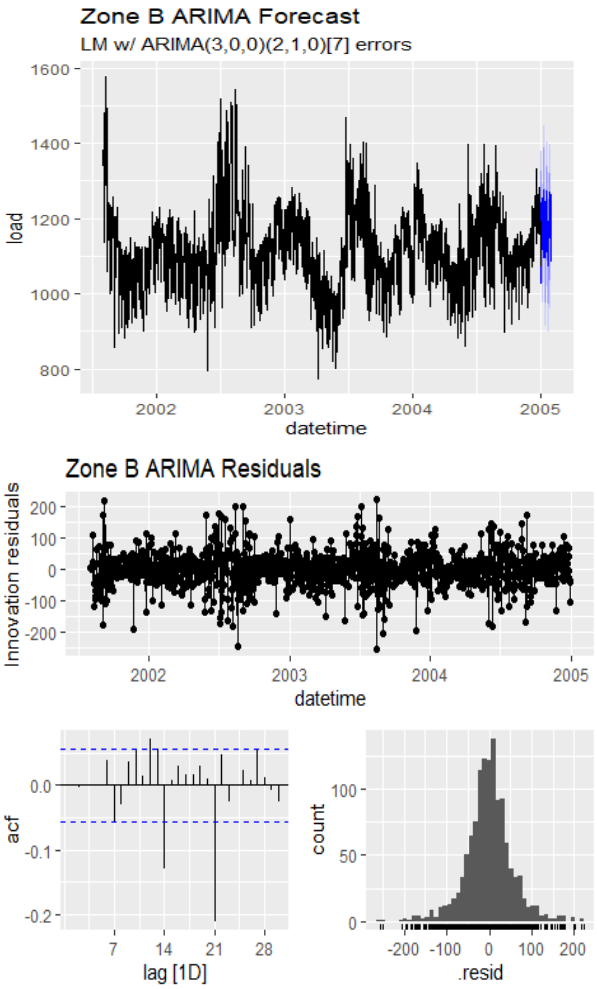
Viewing the plots, I have determined there is seasonality within dew point and temperature, as both are low at the start and high at mid-year. Humidity and load do not show immediate signs of seasonality. No variable shows trends from the plots.

.model	RMSE	MAPE
arima5	54.7	3.48
arima7	54.9	3.49
arima3	54.9	3.48

As shown, Zone B ran 16 models, but the ARIMA with load, temperature, dew point, and humidity without BoxCox transformation was deemed ideal. The RMSE is 54.73 and MAPE is 3.47. The ARIMA model is still to be determined.

Series: load
Model: LM w/ ARIMA(3,0,0)(2,1,0)[7] errors

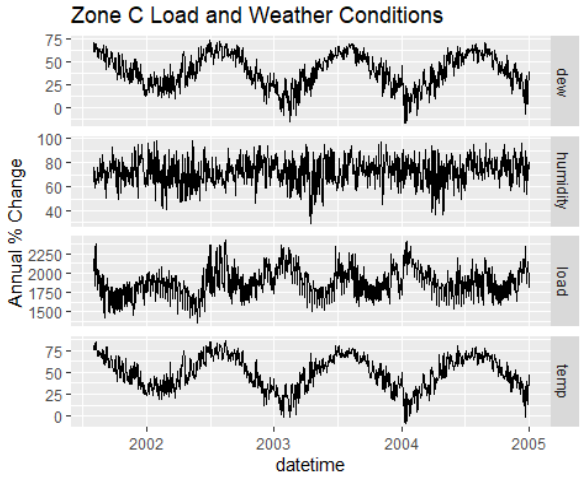
The ARIMA model selected was LM w/ ARIMA(3,0,0)(2,1,0)[7] errors. With the model, we can forecast January 2005.



.model	lb_stat	lb_pvalue
arima5	19.6	0.0758

As we can see, the model fits well with the training data. The residuals are normally distributed and do resemble that of white noise, as 95% of the spikes in the ACF do lie within the blue dashed boundary lines. Through the Ljung-Box test, I failed to reject the null hypothesis as the p-value is greater than 0.05.

Zone C



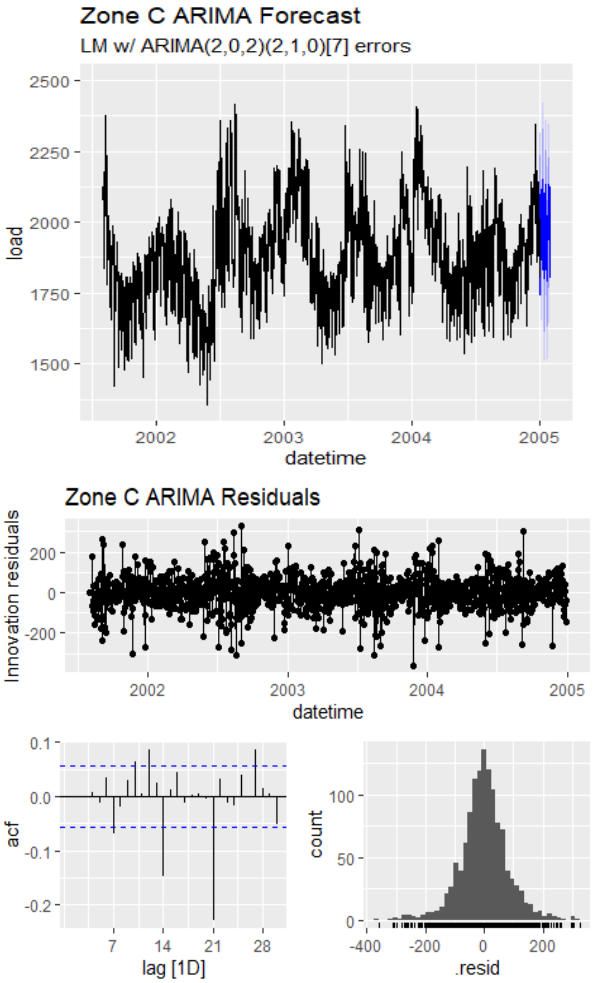
Viewing the plots, I have determined there is seasonality within dew point and temperature, as both are low at the start and high at mid-year. Humidity and load do not show immediate signs of seasonality. No variable shows trends from the plots.

.model	RMSE	MAPE
arima5	81.0	3.13
arima3	81.1	3.13
arima4	81.1	3.13

As shown, Zone C ran 16 models, but the ARIMA with load, temperature, dew point, and humidity without BoxCox transformation was deemed ideal. The RMSE is 81.03 and MAPE is 3.13. The ARIMA model is still to be determined.

Series: load
Model: LM w/ ARIMA(2,0,2)(2,1,0)[7] errors

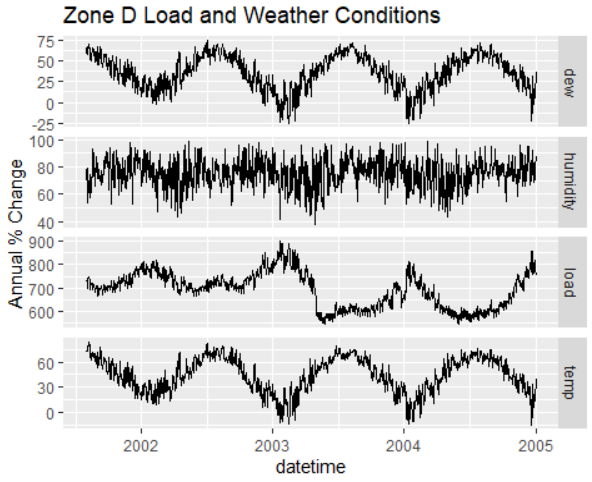
The ARIMA model selected was LM w/ ARIMA(2,0,2)(2,1,0)[7] errors. With the model, we can forecast January 2005.



.model	lb_stat	lb_pvalue
arima5	23.4	0.0247

As we can see, the model fits well with the training data. The residuals are normally distributed and do resemble that of white noise, as 95% of the spikes in the ACF do lie within the blue dashed boundary lines. Through the Ljung-Box test, I rejected the null hypothesis as the p-value is less than 0.05.

Zone D



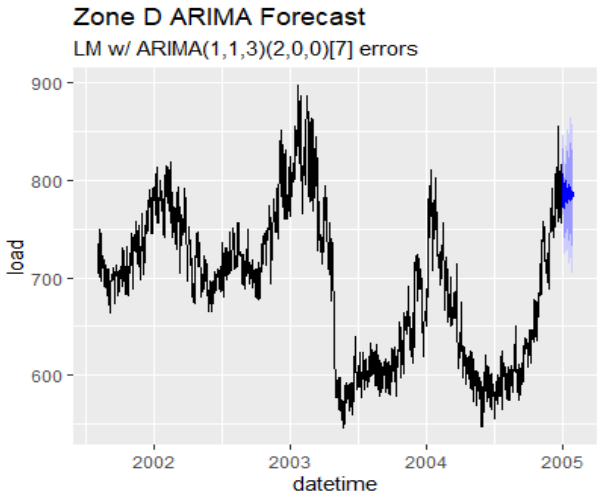
Viewing the plots, I have determined there is seasonality within dew point and temperature, as both are low at the start and high at mid-year. Humidity does not show immediate signs of seasonality. Load shows some signs of seasonality as it is high at the start and low at mid-year. No variable shows trends from the plots.

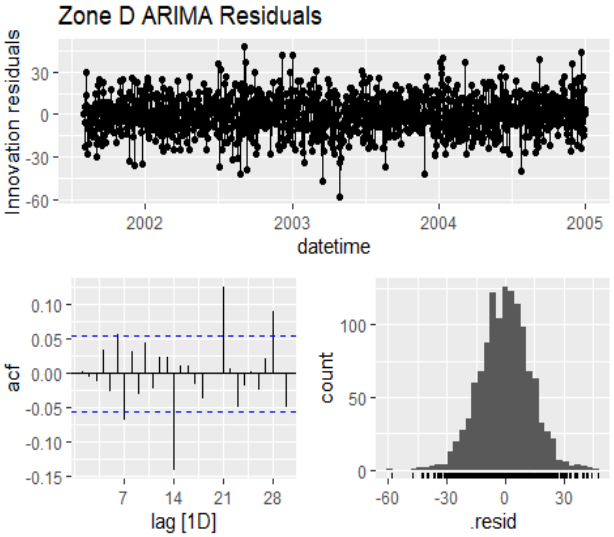
.model	RMSE	MAPE
arima5	13.2	1.49
arima_tran5	13.2	1.46
arima4	13.2	1.49

As shown, Zone D ran 16 models, but the ARIMA with load, temperature, dew point, and humidity without BoxCox transformation was deemed ideal. The RMSE is 13.15 and MAPE is 1.48. The ARIMA model is still to be determined.

Series: load Model: LM w/ ARIMA(1,1,3)(2,0,0)[7] errors

The ARIMA model selected was LM w/ ARIMA(1,1,3)(2,0,0)[7] errors. With the model, we can forecast January 2005.

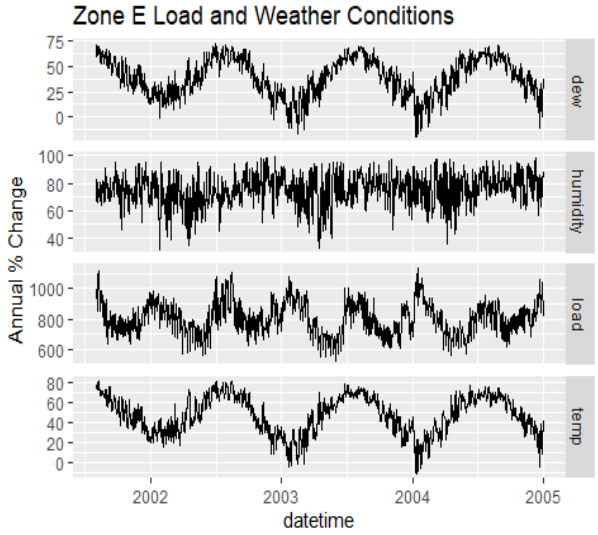




.model	lb_stat	lb_pvalue
arima5	19.0	0.0885

As we can see, the model fits well with the training data. The residuals are normally distributed and do resemble that of white noise, as 95% of the spikes in the ACF do lie within the blue dashed boundary lines. Through the Ljung-Box test, I failed to reject the null hypothesis as the p-value is greater than 0.05.

Zone E



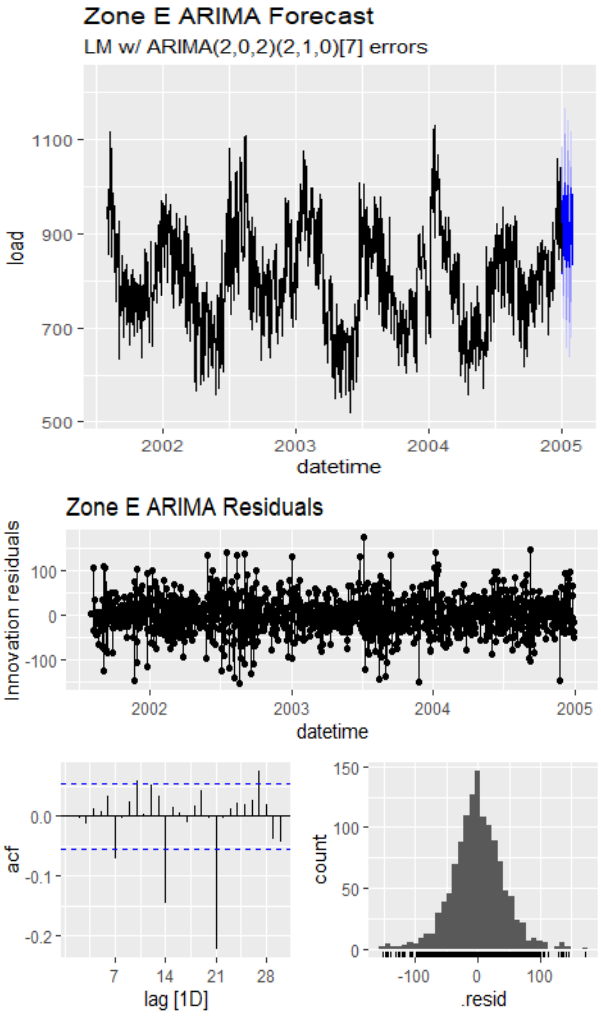
Viewing the plots, I have determined there is seasonality within dew point and temperature, as both are low at the start and high at mid-year. Humidity and load do not show immediate signs of seasonality. No variable shows trends from the plots.

.model	RMSE	MAPE
arima5	41.3	3.86
arima3	41.3	3.86
arima4	41.4	3.87

As shown, Zone E ran 16 models, but the ARIMA with load, temperature, dew point, and humidity without BoxCox transformation was deemed ideal. The RMSE is 41.31 and MAPE is 3.86. The ARIMA model is still to be determined.

Series: load
Model: LM w/ ARIMA(2,0,2)(2,1,0)[7] errors

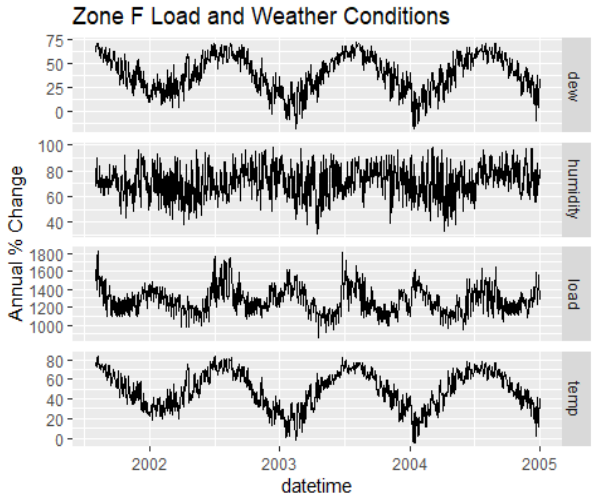
The ARIMA model selected was LM w/ ARIMA(2,0,2)(2,1,0)[7] errors. With the model, we can forecast January 2005.



.model	lb_stat	lb_pvalue
arima5	17.0	0.149

As we can see, the model fits well with the training data. The residuals are normally distributed and do resemble that of white noise, as 95% of the spikes in the ACF do lie within the blue dashed boundary lines. Through the Ljung-Box test, I failed to reject the null hypothesis as the p-value is greater than 0.05.

Zone F



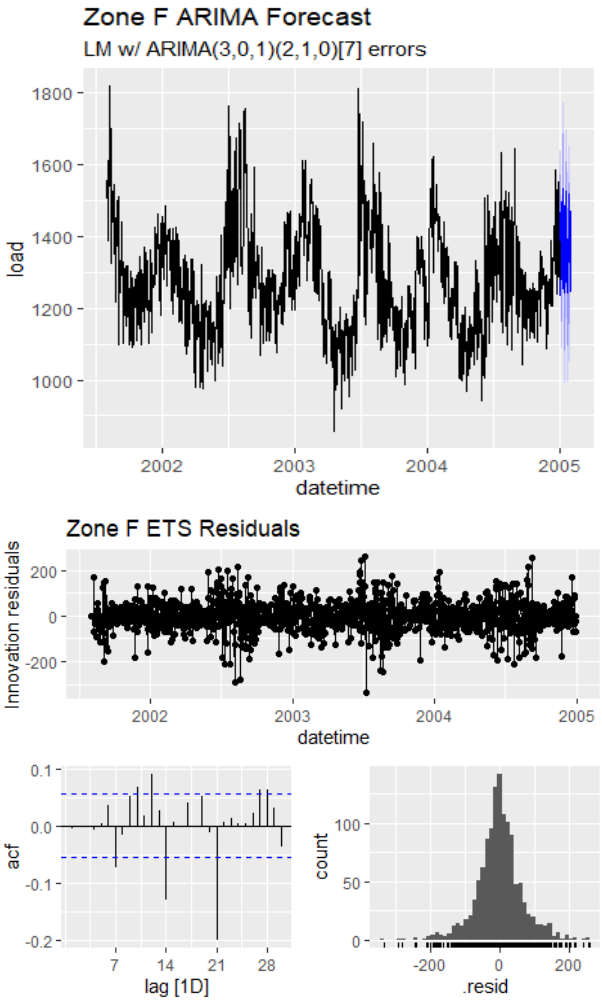
Viewing the plots, I have determined there is seasonality within dew point and temperature, as both are low at the start and high at mid-year. Humidity and load do not show immediate signs of seasonality. No variable shows trends from the plots.

.model	RMSE	MAPE
arima3	65.5	3.63
arima4	65.6	3.63
arima7	65.7	3.64

As shown, Zone F ran 16 models, but the ARIMA with load, temperature, and dew point without BoxCox transformation was deemed ideal. The RMSE is 65.53 and MAPE is 3.62. The ARIMA model is still to be determined.

Series: load
Model: LM w/ ARIMA(3,0,1)(2,1,0)[7] errors

The ARIMA model selected was LM w/ ARIMA(3,0,1)(2,1,0)[7] errors. With the model, we can forecast January 2005.



.model	lb_stat	lb_pvalue
arima3	28.4	0.00478

As we can see, the model fits well with the training data. The residuals are normally distributed and do resemble that of white noise, as 95% of the spikes in the ACF do lie within the blue dashed boundary lines. Through the Ljung-Box test, I rejected the null hypothesis as the p-value is less than 0.05.

Conclusions and Next Steps

The purpose of the models was to answer the question, what modern-day model can be implemented in the day-to-day operations for short-term load forecasting? I was able to look at 7,680 observations and 6 attributes. My methodology for addressing this question

consisted of 4 processes (Data Exploration, Data Preparation, Time Series Analysis, and ARIMA Modeling) across each of the 6 Zones.

In data exploration, the average load (in megawatts) over a 24-hour period was used with temperature (in Fahrenheit), dew point, and/or humidity as predictor variables. Missing data was identified, and Temperature, dew point, and humidity were selected as predictor variables for the response variable (load) due to their significance in affecting electric utilization. I determined dew point had a high correlation with temperature and a moderate correlation with humidity.

In data preparation, I removed all attributes other than datetime, zone, load, temperature, dew point, and humidity. The datetime attribute needed to be changed from a character data type to a Date. Additionally, the data was split to compare forecast results with the actual observations.

In time series analysis, the goal was to understand the predictors and their relationship with the response variable. As an initial method of exploration, the data was examined to look for trends and seasonality within each respective zone. I transformed the training data to a time series with the zones as the key and the datetime as the index. Each zone showed some signs of seasonality at the start of the year (within the winter season) and mid-year (within the summer season). There was no noticeable trend within the data as the data increases and decreases over the period.

Zone A produced an RMSE of 75.82 and MAPE of 3.00. The ARIMA model selected was LM w/ ARIMA(0,0,3)(0,1,1)[7] errors. The model fit well with the training data. The residuals were normally distributed and did not resemble that of white noise, as less than 95% of the spikes in the ACF did not lie within the blue dashed boundary lines. Through the Ljung-Box test, I rejected the null hypothesis as the p-value was less than 0.05.

Zone B produced an RMSE of 54.73 and MAPE of 3.47. The ARIMA model selected was LM w/ ARIMA(3,0,0)(2,1,0)[7] errors. The model fit well with the training data. The residuals were normally distributed and did resemble that of white noise, as 95% of the spikes in the ACF do lie within the blue dashed boundary lines. Through the Ljung-Box test, I failed to reject the null hypothesis as the p-value was greater than 0.05.

Zone C produced an RMSE of 81.03 and MAPE of 3.13. The ARIMA model selected was LM w/ ARIMA(2,0,2)(2,1,0)[7] errors. The model fit well with the training data. The residuals were normally distributed and did resemble that of white noise, as 95% of the spikes in the ACF did lie within the blue dashed boundary lines. Through the Ljung-Box test, I rejected the null hypothesis as the p-value was less than 0.05.

Zone D produced an RMSE of 13.15 and MAPE of 1.48. The ARIMA model selected was LM w/ ARIMA(1,1,3)(2,0,0)[7] errors. The model fits well with the training data. The residuals were normally distributed and did resemble that of white noise, as 95% of the spikes in the ACF did lie within the blue dashed boundary lines. Through the Ljung-Box test, I failed to reject the null hypothesis as the p-value was greater than 0.05.

Zone E produced an RMSE of 41.31 and MAPE of 3.86. The ARIMA model selected was LM w/ ARIMA(2,0,2)(2,1,0)[7] errors. The model fit well with the training data. The residuals were normally distributed and did resemble that of white noise, as 95% of the spikes in the ACF did lie within the blue dashed boundary lines. Through the Ljung-Box test, I failed to reject the null hypothesis as the p-value was greater than 0.05.

Zone F produced an RMSE of 65.53 and MAPE of 3.62. The ARIMA model selected was LM w/ ARIMA(3,0,1)(2,1,0)[7] errors. The model fit well with the training data. The

residuals were normally distributed and did resemble that of white noise, as 95% of the spikes in the ACF did lie within the blue dashed boundary lines. Through the Ljung-Box test, I rejected the null hypothesis as the p-value was less than 0.05. The values when comparing the actual loads with the load forecasts are fairly accurate.

Next steps may include:

- Implementing the models to operations
- Try other seasonal modeling
- Use hourly data instead of daily
- Adding more relevant attributes

Areas for Future Research

From my research, I realized the models can be improved more with more relevant attributes. Additionally, advanced models, such as SARIMA or SARIMAX, can be considered and applied in hopes for more accurate results.

Limitations

There were a few limitations with this project. Hardware, primarily memory storage, plays a big factor in the speed of modeling. At times, the model can freeze the program if there is not enough memory storage. Additionally, acquiring the data required web-scraping as there are no detailed data sets openly available without payment.

Bibliography

- Lindberg, K., Seljom, P., Madsen, H., Fischer, D., & Korpås, M. (2019). *Long-term electricity load forecasting: Current and future trends*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S0957178719300116
- Takiyar, S., & Singh, V. (2015). Trend Analysis and evolution of short-term load forecasting techniques. *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*. https://doi.org/10.1109/icrito.2015.7359233
- Wei, N., Fu, L., Yin, L., Li, C., Wang, W., Qiao, W., Li, C., & Zeng, F. (2022). *Short-term load forecasting using detrend singular spectrum fluctuation analysis*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S0360544222016255
- McConky, K. T., & Aponte, O. (2022). *Forecasting an electricity demand threshold to proactively trigger cost saving demand response actions*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S0378778822003929
- Reiner, D., & He, X. (2015). *Electricity demand and basic needs: Empirical evidence from China's households*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S0301421515302469
- M. Momani, "Factors Affecting Electricity Demand in Jordan," *Energy and Power Engineering*, Vol. 5 No. 1, 2013, pp. 50-58. https://www.scirp.org/html/7-6201408_26442.htm
- Kimura, O., & Komatsu, H. (2020). *Peak demand alert system based on electricity demand forecasting for smart meter data*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S0378778820307854
- Urmee, T., Gyamfi, S., & Krumdieck, S. (2013). *Residential peak electricity demand response—Highlights of some behavioural issues*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S1364032113002578

- Rizzoni, G., Muratori, M., & Schuelke-Leech, B.-A. (2014). *Role of residential demand response in modern electricity markets*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S1364032114001488
- El-Saadany, E. F., & Albadi, M. H. (2008). *A summary of demand response in electricity markets*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S0378779608001272
- H. Nguyen and C. K. Hansen, "Short-term electricity load forecasting with Time Series Analysis," *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, Dallas, TX, USA, 2017, pp. 214-221
- J. Y. Fan and J. D. McDonald, "A real-time implementation of short-term load forecasting for distribution power systems," in *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 988-994, May 1994
- M. T. Hagan and S. M. Behr, "The Time Series Approach to Short Term Load Forecasting," in *IEEE Transactions on Power Systems*, vol. 2, no. 3, pp. 785-791, Aug. 1987
- *Our vision and values*. National Grid's vision and values | National Grid Group. (n.d.). https://www.nationalgrid.com/about-us/our-vision-and-values
- *Forecasting: Principles and practice (3rd ed)*. OTexts. (n.d.). https://otexts.com/fpp3/

Appendix

```
library(readr)
library(tidyverse)
library(caret)
library(RANN)
library(psych)
library(DataExplorer)
library(corrplot)
library(lubridate)
library(stringi)
library(fpp3)
library(ggplot2)
library(ggfortify)
library(forecast)
library(openxlsx)
library(writexl)
library(dataMeta)
library(Hmisc)

options(warn=-1)
#import and view data
df =
read.xlsx("LoadWeatherData.xlsx")
link.df = df %>%
  select(datetime, Zone, load, temp,
dew, humidity)
var_desc = c("Date when weather was
reported",
                "Territories within
jurisdiction",
                "Average electricity
(Megawatts) over 24-hour period",
                "Temperature
(Fahrenheit)",
                "Reported dewpoint of
day",
                "Reported humidity of
day")

var_type = c(0, 1, 0, 0, 0, 0)

linker = build_linker(link.df,
variable_description = var_desc,
                      variable_type =
var_type)
linker
#plot missing data
plot_missing(df)
#view distribution of features
```

```
num.df = select(df, load, temp, dew,
humidity)
hist.data.frame(num.df)
#view correlation of features
corrplot.mixed(cor(num.df))
#selecting features for model
df = df %>%
  select(datetime, Zone, load, temp,
dew, humidity)

#transform data to Date format
df$datetime = as.Date(df$datetime,
"%m/%d/%Y", origin = "1997-01-01")

#split training and testing data
train.df = df %>%
  filter(datetime < "2005-01-01")
test.df = df %>%
  filter(datetime > "2004-12-31")
#transform training data to time
series
train.df = train.df %>%
  as_tsibble(index=datetime,
key=Zone)

#plot all loads by Zone
autoplot(train.df, load) +
  facet_wrap(~Zone, scales='free') +
  labs(x='Year', y='Load',
title='Zone Daily Loads')
#filter Zone A
A.df = train.df %>%
  filter(Zone == "Zone_A")

#plot all features
A.df %>%
  pivot_longer(load:humidity,
names_to = "Series", values_to =
"value") %>%
  ggplot(aes(x = datetime, y =
value)) +
  geom_line() +
  facet_grid(rows = vars(Series),
scales = "free_y") +
  labs(title = "Zone A Load and
Weather Conditions",
       y = "Annual % Change",
x="Year")
#assign lambda value
lambda = A.df %>%
  features(load, features=guerrero)
%>%
  pull(lambda_guerrero)

#multiple arima models
fit.A = A.df %>%
  model(
    arima1 = ARIMA(load),
    arima_tran1 =
ARIMA(box_cox(load,lambda)),
    arima2 = ARIMA(load ~ temp),
    arima_tran2 = ARIMA(box_cox(load,
lambda) ~ temp),
    arima3 = ARIMA(load ~ temp +
dew),
    arima_tran3 =
ARIMA(box_cox(load,lambda) ~ temp +
dew),
    arima4 = ARIMA(load ~ temp +
humidity),
    arima_tran4 =
ARIMA(box_cox(load,lambda) ~ temp +
humidity),
    arima5 = ARIMA(load ~ temp + dew
+ humidity),
    arima_tran5 =
ARIMA(box_cox(load,lambda) ~ temp +
dew + humidity),
    arima6 = ARIMA(load ~ dew),
    arima_tran6 = ARIMA(box_cox(load,
lambda) ~ dew),
    arima7 = ARIMA(load ~ dew +
humidity),
    arima_tran7 =
ARIMA(box_cox(load,lambda) ~ dew +
humidity),
    arima8 = ARIMA(load ~ humidity),
    arima_tran8 =
ARIMA(box_cox(load,lambda) ~
humidity)
  )

#arrange and choose best model from
RMSE
accuracy(fit.A) %>%
  arrange(RMSE) %>%
  select(.model, RMSE, MAPE) %>%
  head(3)
#report selected model
```

```
at.A = A.df %>%
  model(arima5 = ARIMA(load ~ temp +
dew + humidity))

report(at.A)
#calculate averages of features
df %>%
  filter(Zone == "Zone_A",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(temp))
df %>%
  filter(Zone == "Zone_A",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(dew))
df %>%
  filter(Zone == "Zone_A",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(humidity))
#forecast Jan 2005
pred.A = new_data(train.df, 31) %>%
  mutate(temp = 29.87097,
         dew = 23.33226,
         humidity = 77.26774)
fc.A = forecast(at.A, pred.A)
fc.A %>%
  autoplot(train.df) +
  labs(title='Zone A ARIMA Forecast',
subtitle='LM w/
ARIMA(0,0,3)(0,1,1)[7] errors')

#check residuals
fit.A %>%
  select(arima5) %>%
  gg_tsresiduals() +
  labs(title='Zone A ARIMA
Residuals')

#estimate null hypothesis
fit.A %>%
  select(arima5) %>%
  augment() %>%
  features(.innov, ljung_box, lag=
12)
B.df = train.df %>%
  filter(Zone == "Zone_B")
```

```
B.df %>%
  pivot_longer(load:humidity,
names_to = "Series", values_to =
"value") %>%
  ggplot(aes(x = datetime, y =
value)) +
  geom_line() +
  facet_grid(rows = vars(Series),
scales = "free_y") +
  labs(title = "Zone B Load and
Weather Conditions",
       y = "Annual % Change")
lambda = B.df %>%
  features(load, features=guerrero)
%>%
  pull(lambda_guerrero)

fit.B = B.df %>%
  model(
    arima1 = ARIMA(load),
    arima_tran1 =
ARIMA(box_cox(load,lambda)),
    arima2 = ARIMA(load ~ temp),
    arima_tran2 = ARIMA(box_cox(load,
lambda) ~ temp),
    arima3 = ARIMA(load ~ temp +
dew),
    arima_tran3 =
ARIMA(box_cox(load,lambda) ~ temp +
dew),
    arima4 = ARIMA(load ~ temp +
humidity),
    arima_tran4 =
ARIMA(box_cox(load,lambda) ~ temp +
humidity),
    arima5 = ARIMA(load ~ temp + dew
+ humidity),
    arima_tran5 =
ARIMA(box_cox(load,lambda) ~ temp +
dew + humidity),
    arima6 = ARIMA(load ~ dew),
    arima_tran6 = ARIMA(box_cox(load,
lambda) ~ dew),
    arima7 = ARIMA(load ~ dew +
humidity),
    arima_tran7 =
ARIMA(box_cox(load,lambda) ~ dew +
humidity),
    arima8 = ARIMA(load ~ humidity),
    arima_tran8 =
```

```
ARIMA(box_cox(load,lambda) ~
humidity)
  )
accuracy(fit.B) %>%
  arrange(RMSE) %>%
  select(.model, RMSE, MAPE) %>%
  head(3)
at.B = B.df %>%
  model(arima5 = ARIMA(load ~ temp +
dew + humidity))

report(at.B)
df %>%
  filter(Zone == "Zone_B",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(temp))
df %>%
  filter(Zone == "Zone_B",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(dew))
df %>%
  filter(Zone == "Zone_B",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(humidity))
pred.B = new_data(train.df, 31) %>%
  mutate(temp = 29.40968,
         dew = 23.15161,
         humidity = 78.46129)
fc.B = forecast(at.B, pred.B)
fc.B %>%
  autoplot(train.df) +
  labs(title='Zone B ARIMA Forecast',
subtitle='LM w/
ARIMA(3,0,0)(2,1,0)[7] errors')

fit.B %>%
  select(arima5) %>%
  gg_tsresiduals() +
  labs(title='Zone B ARIMA
Residuals')

fit.B %>%
  select(arima5) %>%
  augment() %>%
  features(.innov, ljung_box, lag=
12)
C.df = train.df %>%
  filter(Zone == "Zone_C")

C.df %>%
  pivot_longer(load:humidity,
names_to = "Series", values_to =
"value") %>%
  ggplot(aes(x = datetime, y =
value)) +
  geom_line() +
  facet_grid(rows = vars(Series),
scales = "free_y") +
  labs(title = "Zone C Load and
Weather Conditions",
       y = "Annual % Change")
lambda = C.df %>%
  features(load, features=guerrero)
%>%
  pull(lambda_guerrero)

fit.C = C.df %>%
  model(
    arima1 = ARIMA(load),
    arima_tran1 =
ARIMA(box_cox(load,lambda)),
    arima2 = ARIMA(load ~ temp),
    arima_tran2 = ARIMA(box_cox(load,
lambda) ~ temp),
    arima3 = ARIMA(load ~ temp +
dew),
    arima_tran3 =
ARIMA(box_cox(load,lambda) ~ temp +
dew),
    arima4 = ARIMA(load ~ temp +
humidity),
    arima_tran4 =
ARIMA(box_cox(load,lambda) ~ temp +
humidity),
    arima5 = ARIMA(load ~ temp + dew
+ humidity),
    arima_tran5 =
ARIMA(box_cox(load,lambda) ~ temp +
dew + humidity),
    arima6 = ARIMA(load ~ dew),
    arima_tran6 = ARIMA(box_cox(load,
lambda) ~ dew),
    arima7 = ARIMA(load ~ dew +
humidity),
    arima_tran7 =
ARIMA(box_cox(load,lambda) ~ dew +
humidity),
```

```
    arima8 = ARIMA(load ~ humidity),
    arima_tran8 =
ARIMA(box_cox(load,lambda) ~
humidity)
  )
accuracy(fit.C) %>%
  arrange(RMSE) %>%
  select(.model, RMSE, MAPE) %>%
  head(3)
at.C = C.df %>%
  model(arima5 = ARIMA(load ~ temp +
dew + humidity))

report(at.C)
df %>%
  filter(Zone == "Zone_C",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(temp))
df %>%
  filter(Zone == "Zone_C",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(dew))
df %>%
  filter(Zone == "Zone_C",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(humidity))
pred.C = new_data(train.df, 31) %>%
  mutate(temp = 29.56452,
         dew = 22.13548,
         humidity = 75.00323)
fc.C = forecast(at.C, pred.C)
fc.C %>%
  autoplot(train.df) +
  labs(title='Zone C ARIMA Forecast',
subtitle='LM w/
ARIMA(2,0,2)(2,1,0)[7] errors')

fit.C %>%
  select(arima5) %>%
  gg_tsresiduals() +
  labs(title='Zone C ARIMA
Residuals')

fit.C %>%
  select(arima5) %>%
  augment() %>%
  features(.innov, ljung_box, lag=
12)
D.df = train.df %>%
  filter(Zone == "Zone_D")

D.df %>%
  pivot_longer(load:humidity,
names_to = "Series", values_to =
"value") %>%
  ggplot(aes(x = datetime, y =
value)) +
  geom_line() +
  facet_grid(rows = vars(Series),
scales = "free_y") +
  labs(title = "Zone D Load and
Weather Conditions",
       y = "Annual % Change")
lambda = D.df %>%
  features(load, features=guerrero)
%>%
  pull(lambda_guerrero)

fit.D = D.df %>%
  model(
    arima1 = ARIMA(load),
    arima_tran1 =
ARIMA(box_cox(load,lambda)),
    arima2 = ARIMA(load ~ temp),
    arima_tran2 = ARIMA(box_cox(load,
lambda) ~ temp),
    arima3 = ARIMA(load ~ temp +
dew),
    arima_tran3 =
ARIMA(box_cox(load,lambda) ~ temp +
dew),
    arima4 = ARIMA(load ~ temp +
humidity),
    arima_tran4 =
ARIMA(box_cox(load,lambda) ~ temp +
humidity),
    arima5 = ARIMA(load ~ temp + dew
+ humidity),
    arima_tran5 =
ARIMA(box_cox(load,lambda) ~ temp +
dew + humidity),
    arima6 = ARIMA(load ~ dew),
    arima_tran6 = ARIMA(box_cox(load,
lambda) ~ dew),
    arima7 = ARIMA(load ~ dew +
humidity),
    arima_tran7 =
```

```
ARIMA(box_cox(load,lambda) ~ dew +
humidity),
    arima8 = ARIMA(load ~ humidity),
    arima_tran8 =
ARIMA(box_cox(load,lambda) ~
humidity)
  )
accuracy(fit.D) %>%
  arrange(RMSE) %>%
  select(.model, RMSE, MAPE) %>%
  head(3)
at.D = D.df %>%
  model(arima5 = ARIMA(load ~ temp +
dew + humidity))

report(at.D)
df %>%
  filter(Zone == "Zone_D",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(temp))
df %>%
  filter(Zone == "Zone_D",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(dew))
df %>%
  filter(Zone == "Zone_D",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(humidity))
pred.D = new_data(train.df, 31) %>%
  mutate(temp = 20.07097,
         dew = 14.75484,
         humidity = 80.25161)
fc.D = forecast(at.D, pred.D)
fc.D %>%
  autoplot(train.df) +
  labs(title='Zone D ARIMA Forecast',
subtitle='LM w/
ARIMA(1,1,3)(2,0,0)[7] errors')

fit.D %>%
  select(arima5) %>%
  gg_tsresiduals() +
  labs(title='Zone D ARIMA
Residuals')

fit.D %>%
  select(arima5) %>%
  augment() %>%
  features(.innov, ljung_box, lag=
12)
E.df = train.df %>%
  filter(Zone == "Zone_E")

E.df %>%
  pivot_longer(load:humidity,
names_to = "Series", values_to =
"value") %>%
  ggplot(aes(x = datetime, y =
value)) +
  geom_line() +
  facet_grid(rows = vars(Series),
scales = "free_y") +
  labs(title = "Zone E Load and
Weather Conditions",
       y = "Annual % Change")
lambda = E.df %>%
  features(load, features=guerrero)
%>%
  pull(lambda_guerrero)

fit.E = E.df %>%
  model(
    arima1 = ARIMA(load),
    arima_tran1 =
ARIMA(box_cox(load,lambda)),
    arima2 = ARIMA(load ~ temp),
    arima_tran2 = ARIMA(box_cox(load,
lambda) ~ temp),
    arima3 = ARIMA(load ~ temp +
dew),
    arima_tran3 =
ARIMA(box_cox(load,lambda) ~ temp +
dew),
    arima4 = ARIMA(load ~ temp +
humidity),
    arima_tran4 =
ARIMA(box_cox(load,lambda) ~ temp +
humidity),
    arima5 = ARIMA(load ~ temp + dew
+ humidity),
    arima_tran5 =
ARIMA(box_cox(load,lambda) ~ temp +
dew + humidity),
    arima6 = ARIMA(load ~ dew),
    arima_tran6 = ARIMA(box_cox(load,
lambda) ~ dew),
    arima7 = ARIMA(load ~ dew +
```

```
humidity),
    arima_tran7 =
ARIMA(box_cox(load,lambda) ~ dew +
humidity),
    arima8 = ARIMA(load ~ humidity),
    arima_tran8 =
ARIMA(box_cox(load,lambda) ~
humidity)
  )
accuracy(fit.E) %>%
  arrange(RMSE) %>%
  select(.model, RMSE, MAPE) %>%
  head(3)
at.E = E.df %>%
  model(arima5 = ARIMA(load ~ temp +
dew + humidity))

report(at.E)
df %>%
  filter(Zone == "Zone_E",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(temp))
df %>%
  filter(Zone == "Zone_E",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(dew))
df %>%
  filter(Zone == "Zone_E",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(humidity))
pred.E = new_data(train.df, 31) %>%
  mutate(temp = 26.59677,
         dew = 20.66774,
         humidity = 79.07097)
fc.E = forecast(at.E, pred.E)
fc.E %>%
  autoplot(train.df) +
  labs(title='Zone E ARIMA Forecast',
subtitle='LM w/
ARIMA(2,0,2)(2,1,0)[7] errors')

fit.E %>%
  select(arima5) %>%
  gg_tsresiduals() +
  labs(title='Zone E ARIMA
Residuals')
```

```
fit.E %>%
  select(arima5) %>%
  augment() %>%
  features(.innov, ljung_box, lag=
12)
F.df = train.df %>%
  filter(Zone == "Zone_F")

F.df %>%
  pivot_longer(load:humidity,
names_to = "Series", values_to =
"value") %>%
  ggplot(aes(x = datetime, y =
value)) +
  geom_line() +
  facet_grid(rows = vars(Series),
scales = "free_y") +
  labs(title = "Zone F Load and
Weather Conditions",
       y = "Annual % Change")
lambda = F.df %>%
  features(load, features=guerrero)
%>%
  pull(lambda_guerrero)

fit.F = F.df %>%
  model(
    arima1 = ARIMA(load),
    arima_tran1 =
ARIMA(box_cox(load,lambda)),
    arima2 = ARIMA(load ~ temp),
    arima_tran2 = ARIMA(box_cox(load,
lambda) ~ temp),
    arima3 = ARIMA(load ~ temp +
dew),
    arima_tran3 =
ARIMA(box_cox(load,lambda) ~ temp +
dew),
    arima4 = ARIMA(load ~ temp +
humidity),
    arima_tran4 =
ARIMA(box_cox(load,lambda) ~ temp +
humidity),
    arima5 = ARIMA(load ~ temp + dew
+ humidity),
    arima_tran5 =
ARIMA(box_cox(load,lambda) ~ temp +
dew + humidity),
    arima6 = ARIMA(load ~ dew),
    arima_tran6 = ARIMA(box_cox(load,
```

```
lambda) ~ dew),
    arima7 = ARIMA(load ~ dew +
humidity),
    arima_tran7 =
ARIMA(box_cox(load,lambda) ~ dew +
humidity),
    arima8 = ARIMA(load ~ humidity),
    arima_tran8 =
ARIMA(box_cox(load,lambda) ~
humidity)
  )
accuracy(fit.F) %>%
  arrange(RMSE) %>%
  select(.model, RMSE, MAPE) %>%
  head(3)
at.F = F.df %>%
  model(arima3 = ARIMA(load ~ temp +
dew))

report(at.F)
df %>%
  filter(Zone == "Zone_F",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(temp))
df %>%
  filter(Zone == "Zone_F",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(dew))
df %>%
  filter(Zone == "Zone_F",
between(datetime, as.Date("2004-12-
01"), as.Date("2004-12-31"))) %>%
  summarise(mean(humidity))
pred.F = new_data(train.df, 31) %>%
  mutate(temp = 28.39677,
         dew = 20.25484,
         humidity = 72.54516)
fc.F = forecast(at.F, pred.F)
fc.F %>%
  autoplot(train.df) +
  labs(title='Zone F ARIMA Forecast',
subtitle='LM w/
ARIMA(3,0,1)(2,1,0)[7] errors')

fit.F %>%
  select(arima3) %>%
  gg_tsresiduals() +
  labs(title='Zone F ETS Residuals')
```

```
fit.F %>%
  select(arima3) %>%
  augment() %>%
  features(.innov, ljung_box, lag=
12)
#filter test data by zone
act.A = test.df %>%
  filter(Zone == "Zone_A")
act.B = test.df %>%
  filter(Zone == "Zone_B")
act.C = test.df %>%
  filter(Zone == "Zone_C")
act.D = test.df %>%
  filter(Zone == "Zone_D")
act.E = test.df %>%
  filter(Zone == "Zone_E")
act.F = test.df %>%
  filter(Zone == "Zone_F")

#create list and add actual and
forecast values
results = list()

results[['Date']] = test.df$datetime
results[['A.Actual']] = act.A$load
results[['A.Forecast']] = fc.A$.mean
results[['B.Actual']] = act.B$load
results[['B.Forecast']] = fc.B$.mean
results[['C.Actual']] = act.C$load
results[['C.Forecast']] = fc.C$.mean
results[['D.Actual']] = act.D$load
results[['D.Forecast']] = fc.D$.mean
results[['E.Actual']] = act.E$load
results[['E.Forecast']] = fc.E$.mean
results[['F.Actual']] = act.F$load
results[['F.Forecast']] = fc.F$.mean

results = as.data.frame(results)

#export into excel file
write_xlsx(results,
           "JanuaryForecast.xlsx",
           col_names = TRUE,
           format_headers = TRUE)

head(results)
```