Assignment 3

Shaun Harrington

Introduction

I'll be using data from the EIA to predict wind generation in Texas using historical market prices. Theory suggests that higher prices will incentivize an increase in supply, i.e. more electricity production. Because wind turbines are relatively cheap to build and Texas' unique electricity market enables a wide number of suppliers to join the market, it would make sense for prices to be a good predictor of wind turbine generation in this region.

One source of volatility that could hide this relation would be the variation of monthly winds. I was unable to find historical wind data for Texas, but an alternative data series that could work as a replacement is capacity factor. This is defined as $\frac{Average MWh}{Total MW Capacity}$. When prices are positive (or above marginal cost $\approx \$0$), wind turbines will generate as much electricity as possible. Thus while a drop in capacity factor could be due to negative pricing, it is much more likely to be from a lack of wind.

```r
knitr::opts_chunk$set(
    echo = TRUE,
    message = FALSE,
    warning = FALSE,
    fig.width = 8
)

library(tidyverse)
library(fpp3)
library(fredr)
library(scales)
library(jsonlite)
library(zoo)

theme_set(theme_bw())

eia.key <- Sys.getenv("EIA_API_KEY")
```

```r
fn_query_eia <- function( api_url = NULL,
  the_series_id, the_source = "steo", the_frequency = "monthly", the_facet = "seriesId",
  the_offset = 0, the_length = 5000, the_eia_key = eia.key){

    if(is.null(api_url)){
      the_url = "https://api.eia.gov/v2/"

      # Query must be no more than 5,000
      if(the_length > 5000) break

      get_call <- paste0(the_url, the_source, "/data/?", paste(
        paste0("frequency=", the_frequency),
        "data[0]=value",
        paste0("facets[", the_facet, "][]=", the_series_id),
        "sort[0][column]=period",
        "sort[0][direction]=desc",
        paste0("offset=", the_offset),
        paste0("length=", the_length),
        sep = "&"
      ))

      eia_list <- fromJSON(str_c(get_call, "&api_key=", the_eia_key))

      eia_data <- eia_list$response$data

      eia_data %>%
        as_tibble() %>%
        return()
      }

    else{

      eia_list <- fromJSON(str_c(api_url, "&api_key=", the_eia_key))

      eia_data <- eia_list$response$data

      eia_data %>%
        as_tibble() %>%
        return()

    }
  }
```

Get Data

```
url.wind <- "https://api.eia.gov/v2/electricity/electric-power-operational-data/data/?freq

data.wind.gen <- fn_query_eia(api_url = url.wind)

data.price <- fn_query_eia(
  the_series_id = "ELWHU_TX", the_source = "steo", the_facet = "seriesId"
)

data.wind.cf <- fn_query_eia(api_url = "https://api.eia.gov/v2/total-energy/data/?frequenc
```

The data is split into two datasets, a training and testing dataset. The testing set are the most recent 12 months, while the training set are the 48 months preceding that. Lags of monthly market prices will be used, however, market entrants will likely be more interested in pricing trends rather than price shocks, so lagged rolling averages will also be calculated.

The wind turbine capacity factor will be unknown for the forecast period and the monthly averages of the training set will be used for this period. The inclusion of this variable will be to help explain some of the variation in the test set so more representative weights or coefficients for prices may be estimated.

```
data <- left_join(
  x = data.price %>%
    select(period, value) %>%
    rename(date = period, price = value) %>%
    mutate(date = ym(date) %>% yearmonth()),
  y = data.wind.gen %>%
    select(period, generation) %>%
    rename(date = period) %>%
    mutate(date = ym(date) %>% yearmonth()),
  by = "date"
) %>%
  left_join(
    y = data.wind.cf %>%
    select(period, value) %>%
    rename(date = period, capacity.factor = value) %>%
    mutate(date = ym(date) %>% yearmonth()),
    by = "date"
  ) %>%
  arrange(date) %>%
  mutate(
    price_lag12 = lag(price, n = 12),
    price_lag18 = lag(price, n = 18),
    price_lag24 = lag(price, n = 24),
    price_lag30 = lag(price, n = 30),
    price_lag36 = lag(price, n = 36),
    price_lag48 = lag(price, n = 48),
    price_lag12_ma = rollmean(price, k = 12, fill = NA, align = "right") %>%
      lag(n = 12),
    price_lag24_ma = rollmean(price, k = 24, fill = NA, align = "right") %>%
      lag(n = 24),
    price_lag36_ma = rollmean(price, k = 36, fill = NA, align = "right") %>%
      lag(n = 36),
    price_lag48_ma = rollmean(price, k = 48, fill = NA, align = "right") %>%
      lag(n = 48)
  ) %>%
  drop_na()

avg.capacity <- data %>%
  mutate(month = month(date)) %>%
  group_by(month) %>%
  summarize(capacity.factor = mean(capacity.factor)) %>%
  ungroup()

test <- data %>%
  select(-capacity.factor) %>%
  slice_max(order_by = date, n = 12) %>%
  mutate(date = yearmonth(date), month = month(date)) %>%
  left_join(avg.capacity, by = "month") %>%
  select(-month) %>%
  tsibble()

train <- data %>%
  slice_max(order_by = date, n = 12*10) %>%
  mutate(date = yearmonth(date)) %>%
  anti_join(y = test, by = "date") %>%
  tsibble()
```

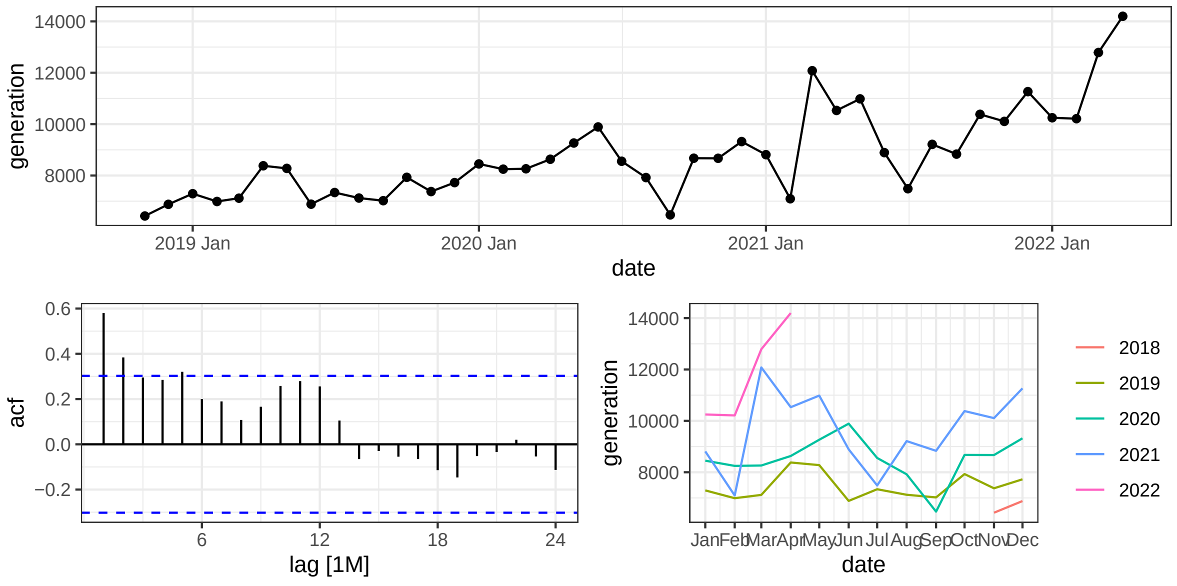
```
data <- data %>%
  tsibble(index = date)
```

Preliminary Analysis

Data Exploration

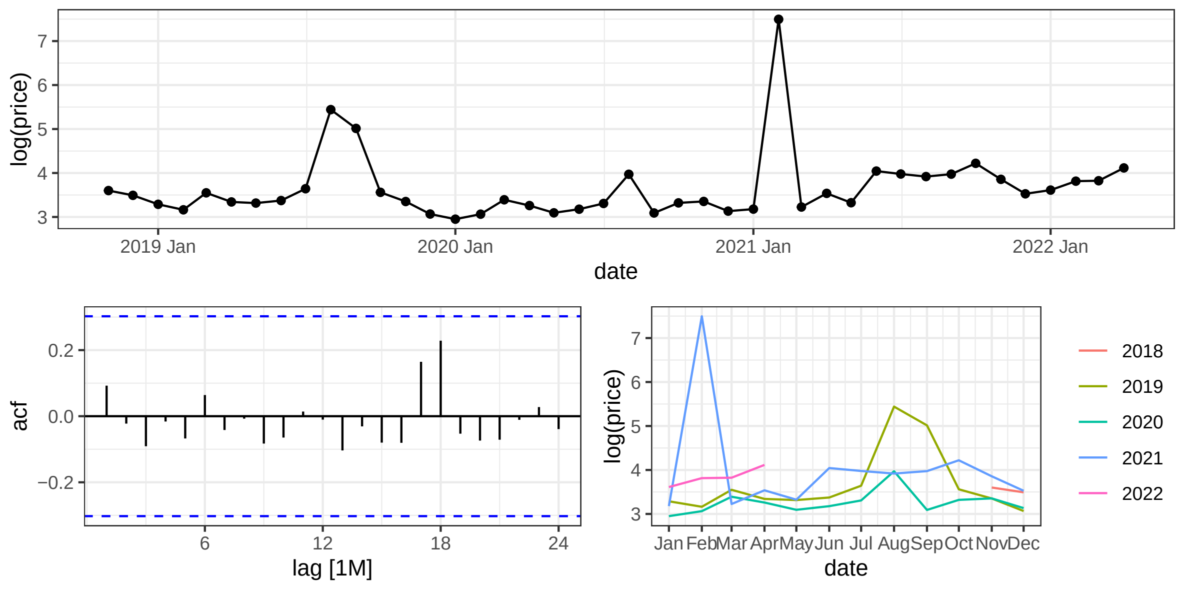
Monthly Wind Generation in Texas

```
train %>%
  gg_tsdisplay(generation, lag_max = 24)
```



Monthly Average Wholesale Electricity Price in Texas (log scale)

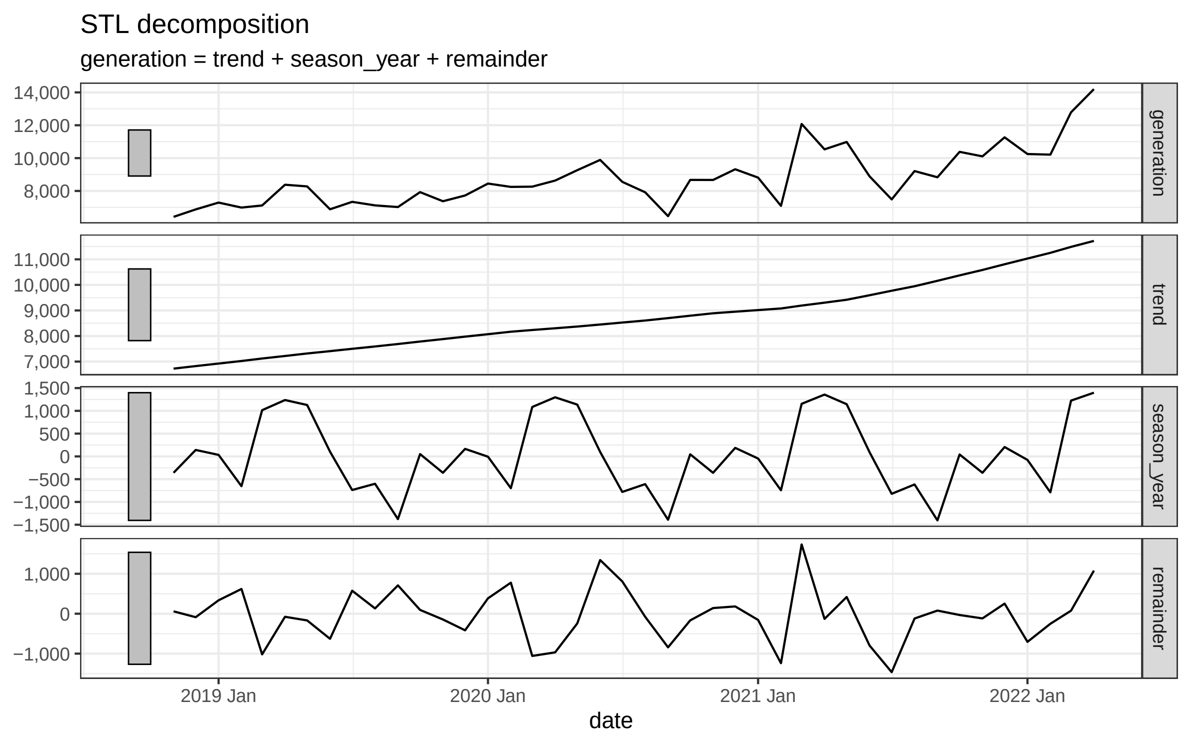
```
train %>%
  gg_tsdisplay(log(price), lag_max = 24)
```



Decomposition

The STL Decomposition breaks out the data into the trend, season, and remainder components. The seasonality will largely be driven by wind though I would suspect repairs in the shoulder months playing a role. The trend will show the general increase in supply over time. The remainder will contain wind deviations from normal, unexpected turbine outages, and other unforeseeable factors.

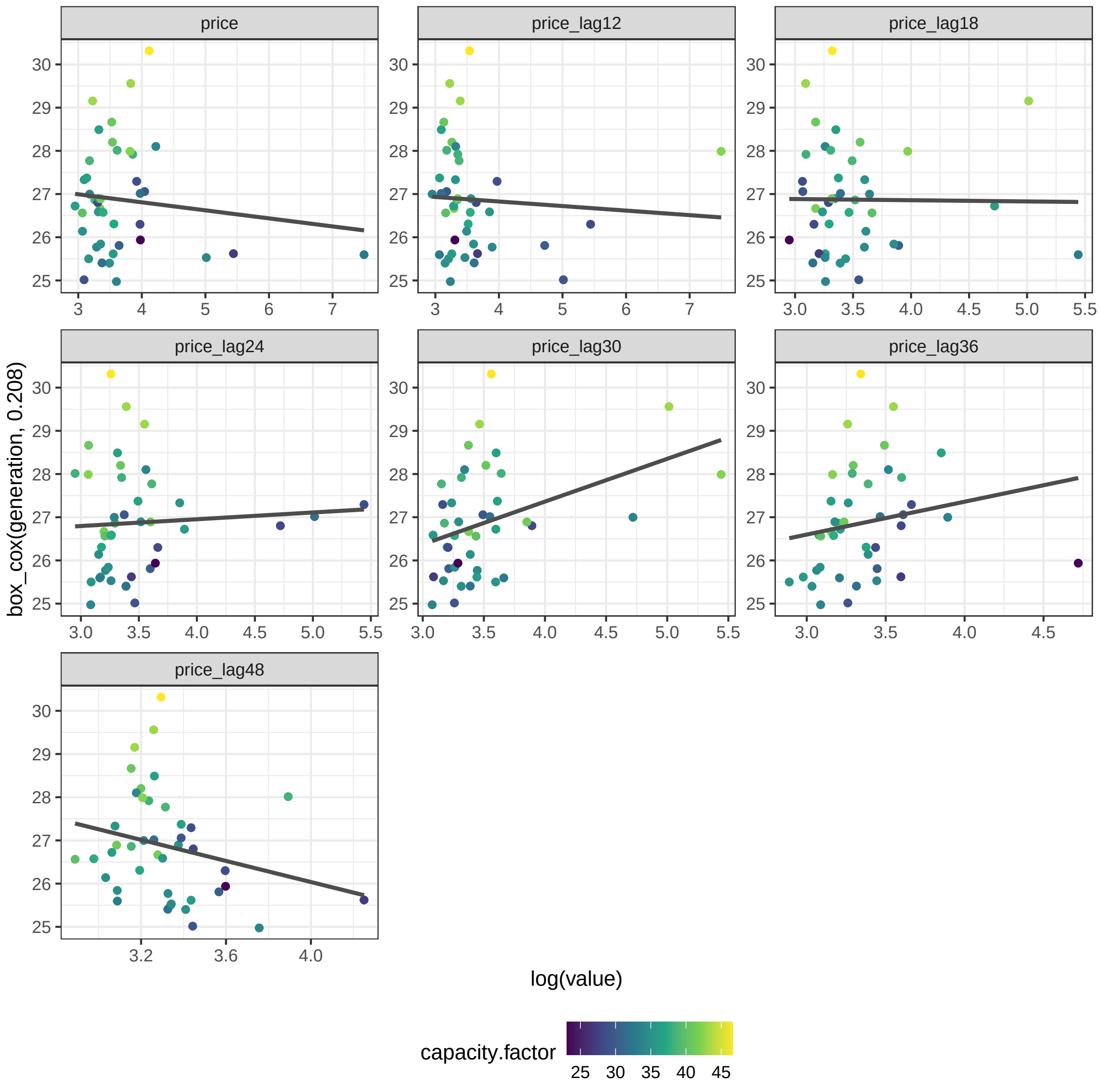
```
train %>%
  model(STL(generation)) %>%
  components() %>%
  autoplot() +
  scale_y_continuous(labels = label_comma())
```



Correlations with Lags of Price

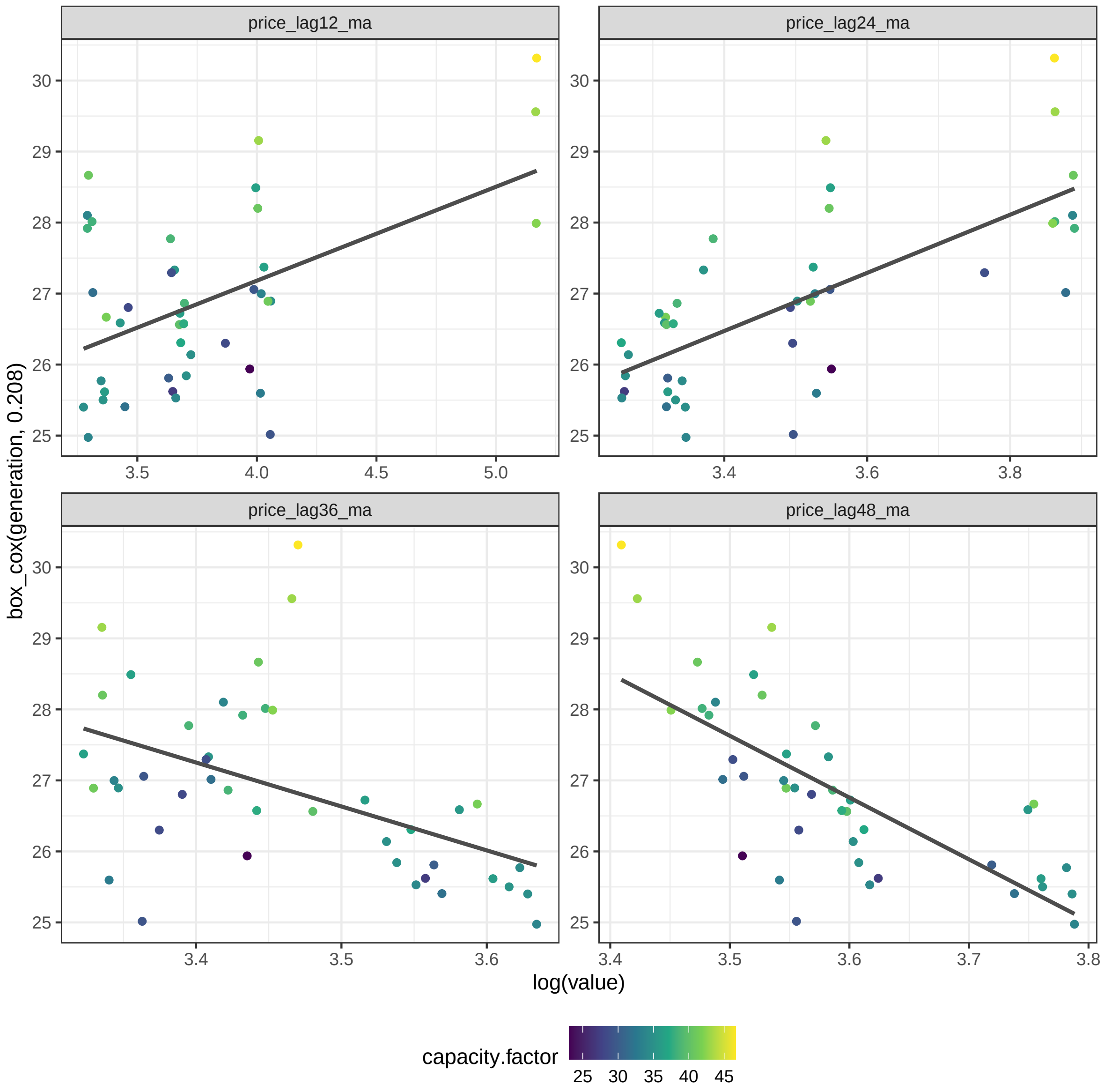
Various outliers skew a linear fit between lags of price and wind generation. Many of these outliers are months with low capacity factors/wind. Looking past the outliers, the lags 30 and 36 do appear more correlated, but still rather loosely.

```r
train %>%
  pivot_longer(
    -c(date, generation, capacity.factor, contains("ma")),
    names_to = "lag", values_to = "value"
  ) %>%
  ggplot(aes(
    x = log(value),
    # x = value,
    y = box_cox(generation, .208),
    # y = generation,
    color = capacity.factor
  )) +
  geom_point() +
  geom_smooth(method = "lm", se = F, scales = "free", color = "gray30") +
  facet_wrap(lag ~ ., scales = "free") +
  scale_color_viridis_c() +
  theme(
    legend.position = "bottom"
  )
```



Including lagged 12-month moving averages cleans up much of the price volatility. The 24-month lag shows the most significant correlation with negative correlation appearing after. The negative correlation is likely the opposite affect occurring: increased supply places a downward pressure on prices.

```r
train %>%
  pivot_longer(c( contains("ma")), names_to = "lag", values_to = "value") %>%
  ggplot(aes(
    x = log(value),
    # x = value,
    y = box_cox(generation, .208),
    # y = generation,
    color = capacity.factor
  )) +
  geom_point() +
  geom_smooth(method = "lm", se = F, scales = "free", color = "gray30") +
  facet_wrap(lag ~ ., scales = "free") +
  scale_color_viridis_c() +
  theme(
    legend.position = "bottom"
  )
```



Modeling

Estimation

Three models will be estimated: an ETS, an auto-ARIMA, and a neural network with 4 nodes. A fourth ensemble model will be a simple average of the others.

```
(fit <- train %>%
  model(
    "ets" = ETS(box_cox(generation, .208)),
    "arima" = ARIMA(
      box_cox(generation, .208) ~ log(price_lag12_ma) +
        log(price_lag24_ma) + log(price_lag36_ma) +
        log(capacity.factor)
    ),
    "nn" = NNETAR(
      box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 4, scale_inputs = TRUE
    )
  ) %>%
   mutate(ensemble = (ets + arima + nn) / 3))
```

```
# A mable: 1 x 4
          ets                                      arima                nn
      <model>                                    <model>           <model>
1 <ETS(M,A,N)> <LM w/ ARIMA(0,0,0)(0,1,0)[12] errors> <NNAR(1,1,4)[12]>
# ... with 1 more variable: ensemble <model>
```

The coefficients of the ARIMA model show how generation has responded to historical prices for the training set. The 24 and 36 month lagged moving averages show a positive correlation while the closer 12 month lag has a negative correlation. However, it's worth pointing out that the standard errors on these are quite large so any statistical significance is doubtful.

```
fit %>%
  select(arima) %>%
  report()
```

```
Series: generation
Model: LM w/ ARIMA(0,0,0)(0,1,0)[12] errors
Transformation: box_cox(generation, 0.208)

Coefficients:
      log(price_lag12_ma)  log(price_lag24_ma)  log(price_lag36_ma)
                  -0.1654               0.4638               0.1770
s.e.               0.1501               0.6389               0.7221
      log(capacity.factor)  intercept
                    6.4261     0.7710
```

```
s.e.                0.6247     0.1754

sigma^2 estimated as 0.1915:  log likelihood=-15.04
AIC=42.07   AICc=45.73   BIC=50.48
```

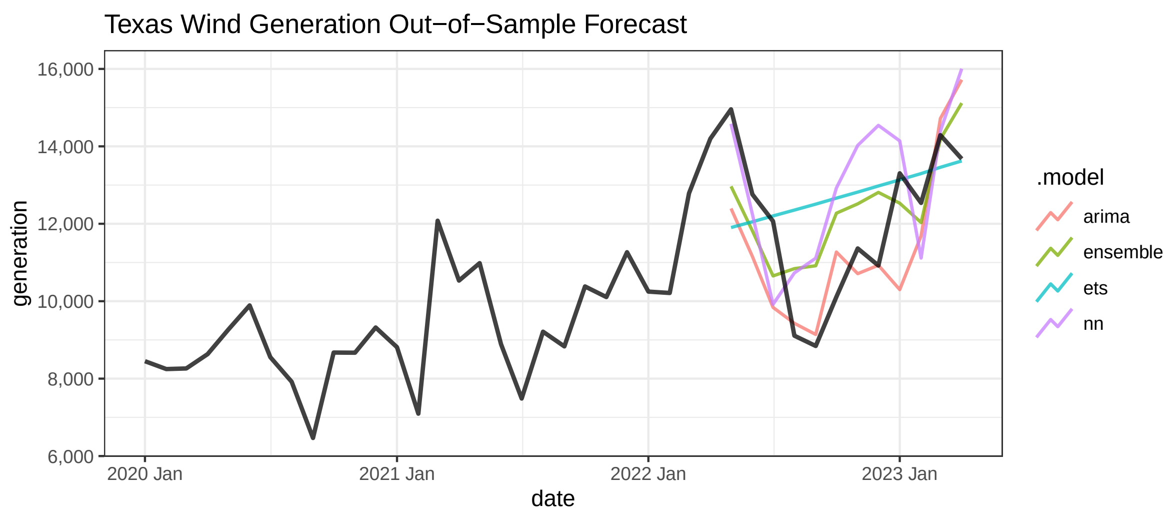
Forecast

The models were trained on data prior to 2022-05-01. The forecast period is the interval 2022-05-01 UTC–2023-04-01 UTC.

For the forecast period, the average capacity factor by month will be used along with lagged transformations of price.

```r
fx <- fit %>%
    forecast(test, times = 500)


fx %>%
  autoplot(
      level = NULL, size = .75, alpha = .75
    ) +
  autolayer(
    data %>% tsibble() %>% filter(year(date)>=2020), generation,
    size = 1, alpha = .75#, linetype = "dashed"
  ) +
  ggtitle("Texas Wind Generation Out-of-Sample Forecast") +
  scale_y_continuous(labels = label_comma()); fx %>%
  accuracy(test, measures = point_accuracy_measures) %>%
  arrange(RMSE)
```



```
# 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 ensemble Test  -394. 1490. 1347. -4.90 12.1    NaN   NaN 0.407
2 arima    Test   553. 1585. 1263.  3.91  9.97   NaN   NaN 0.227
3 ets      Test  -754. 1999. 1556. -8.97 14.5    NaN   NaN 0.478
4 nn       Test  -983. 2022. 1726. -9.68 15.6    NaN   NaN 0.435
```

The ensemble model outperforms the rest, likely a result of the forecasts not being biased one way or the other in comparison to the actual generation. The ARIMA and NN models are very comparable and the ETS is at the bottom.

Cross Validation

One issue with neural networks is selecting the appropriate number of nodes. One way this can be handled is through cross validation. By splitting the dataset into various components and training models on each of these, we can compare how each does over multiple out-of-sample forecasts. We'll split the training set into 9 segments and train neural nets with 2, 3, 4, 5, 6, 7, and 8 nodes to determine which would perform best on average.

```
train.cv <- train %>%
  stretch_tsibble(.init = 24, .step = 1) %>%
  relocate(.id, everything())

train.cv$.id %>% max()
```

```
[1] 19
```

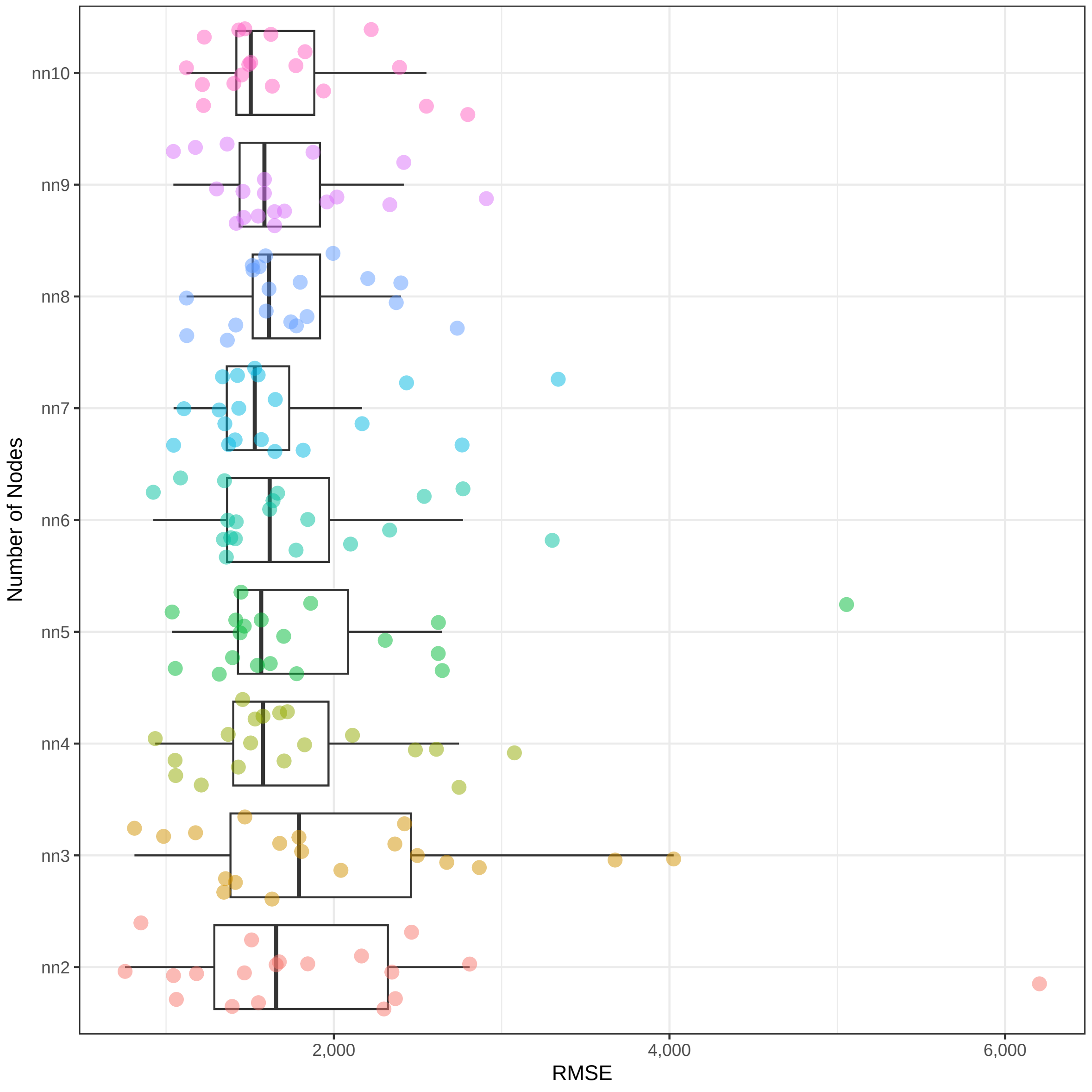
```
fit.cv <- train.cv %>%
  model(
    "nn2" = NNETAR(
      box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 2, scale_inputs = TRUE
    ),
     "nn3" = NNETAR(
      box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 3, scale_inputs = TRUE
    ),
     "nn4" = NNETAR(
      box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 4, scale_inputs = TRUE
    ),
     "nn5" = NNETAR(
      box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 5, scale_inputs = TRUE
    ),
     "nn6" = NNETAR(
      box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 6, scale_inputs = TRUE
    ),
     "nn7" = NNETAR(
      box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 7, scale_inputs = TRUE
    ),
     "nn8" = NNETAR(
     box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 8, scale_inputs = TRUE
    ),
     "nn9" = NNETAR(
     box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 9, scale_inputs = TRUE
    ),
     "nn10" = NNETAR(
     box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 10, scale_inputs = TRUE
    ))
```

To evaluate the cross validated error, predictions will be made on the 12 months following each data partition.

```
fx.cv <- fit.cv %>%
  forecast(
    new_data(train.cv, n = 12) %>% left_join(data, by = "date") %>% drop_na(),
    times = 50
  )



fx.cv %>%
  accuracy(data, by = c(".model", ".id")) %>%
  mutate(.model = factor(.model, levels = paste("nn", c(2:10), sep = ""))) %>%
  ggplot(aes(x = RMSE, y = .model)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = .model), width = .2, size = 3, alpha = .5) +
  scale_x_continuous(labels = label_comma()) +
  ylab("Number of Nodes") +
  theme(legend.position = "none")
```



Cross validation has found the model with 8 nodes to minimize the median RMSE and it also has a much smaller variance than most other models.

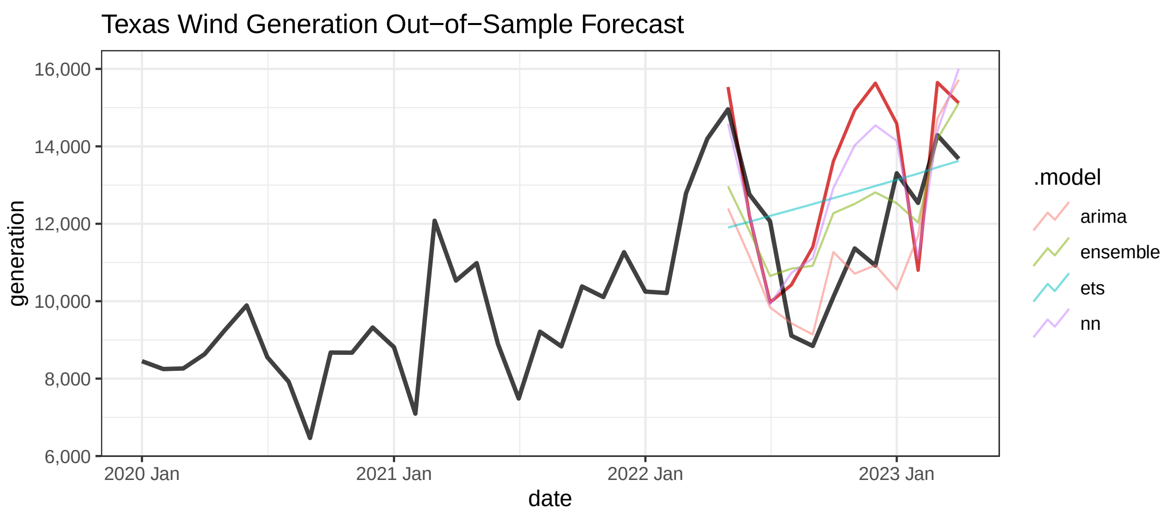
A final model will be fit using all the training data to predict the test set.

```
(fit.final <- train %>%
  model(
    "nn_final" = NNETAR(
      box_cox(generation, .208) ~ log(price_lag12_ma) +
        (capacity.factor),
      n_nodes = 8, scale_inputs = TRUE
    )))
```

```
# A mable: 1 x 1
           nn_final
            <model>
1 <NNAR(1,1,8)[12]>
```

```
  fx.final <- fit.final %>%
    forecast(test, times = 500)


  fx.final %>%
    autoplot(
        level = NULL, size = .75, alpha = .75, color = "red3"
      ) +
    autolayer(
      data %>% tsibble() %>% filter(year(date)>=2020), generation,
      size = 1, alpha = .75#, linetype = "dashed"
    ) +
    autolayer(fx, level = NULL, linewidth = .5, alpha = .5) +
    ggtitle("Texas Wind Generation Out-of-Sample Forecast") +
    scale_y_continuous(labels = label_comma()); fx.final %>%
    accuracy(test, measures = point_accuracy_measures) %>%
    arrange(RMSE)
```



```
# A tibble: 1 x 10
  .model   .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>    <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 nn_final Test  -1328. 2401. 2059. -12.6  18.5   NaN   NaN 0.477
```

However, despite all that, the model performs worse on the test dataset than any of the other models tested.