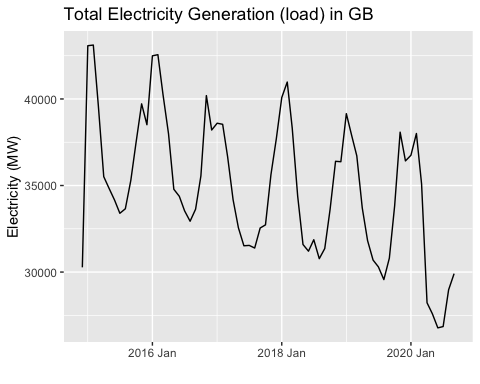
Final Project, Great Britain Generation

EZ

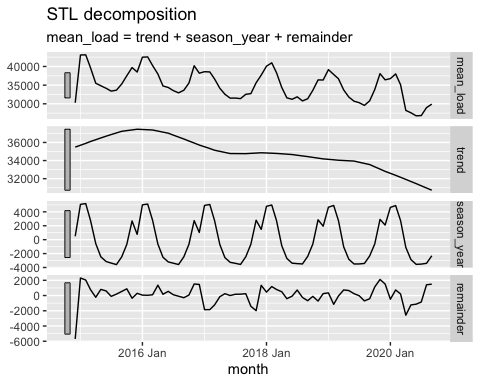
6/10/2021

The data provided is the total electricity generation (load) in Great Britain from 2015 to the present, in megawatts (MW). The measured variable from the model GB\_monthly selected was the mean\_load. This gave the monthly data of the Great Britain Generation. This data obtained will be used to forecast. The question is which is the best model to use to make this forecast.



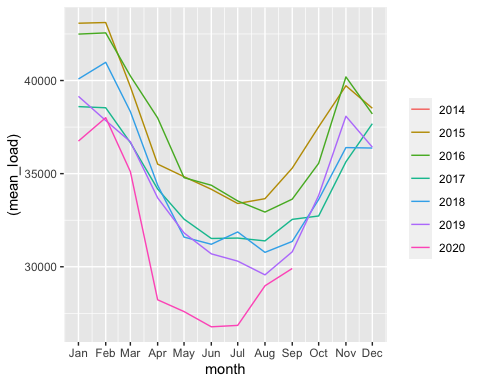
Outliers observed in the data are not shown. This data does show seasonality as there are peaks from December to February. This isn’t a surprise which this is Great Britain’s Winter. As more electricity is used during the Winter. These peaks do show consistency throughout the series. It’s hard to see if this data has fairly consistent variation, so a transformation can be useful here.

GB\_monthly %>% model(stl = STL(mean\_load ~ season(window = 11) + trend(window = 23))) %>% components() %>% autoplot()

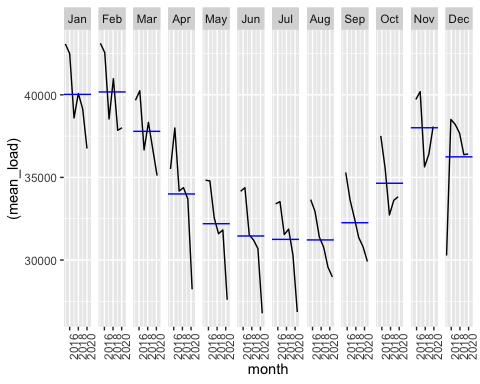


Plotting the decomposed data is often worthwhile to detect the type of trend and type of errors that one is dealing with.

GB\_monthly%>%gg\_season((mean\_load))



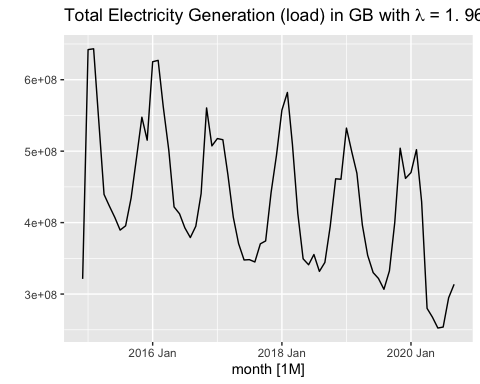
GB\_monthly%>%gg\_subseries((mean\_load))



Looking at the plot above shows seasonally. As of the beginning and the end of the year have high consumption of electricity. Between April and October, it’s at its lowest point whereas it’s a lot warmer and the consumption is at a lower rate.

Looking at the plots we discern the need for a transformation. But just to make sure, I used the Guerrero method to select the optimal lambda for a Box-Cox transformation. The specified lambda is far from zero in absolute value (i.e., ) so there’s probably no need for a transformation.

lambda\_GB <- GB\_monthly %>% features(mean\_load, features = guerrero) %>% pull(lambda\_guerrero)  
  
GB\_monthly %>%  
 autoplot(box\_cox(mean\_load, lambda\_GB)) +  
 labs(y = "",  
 title = latex2exp::TeX(paste0(  
 "Total Electricity Generation (load) in GB with $\\lambda$ = ",  
 round(lambda\_GB,2))))

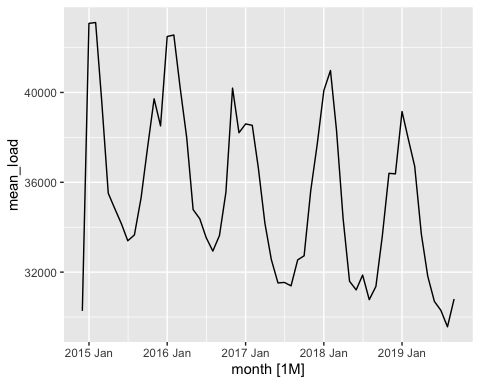


Creating Training and Test Sets with seasonal data. There is one full seasonal period as a test set, with the remainder being the training data. In this case use the final year of data (October 2019 through September 2020) as the holdout data.

GB\_test <- GB\_monthly%>%filter\_index("Oct 2019" ~ "Sep 2020")  
GB\_training <- anti\_join(GB\_monthly, GB\_test)

## Joining, by = c("month", "mean\_load")

autoplot(GB\_training, .vars = mean\_load)

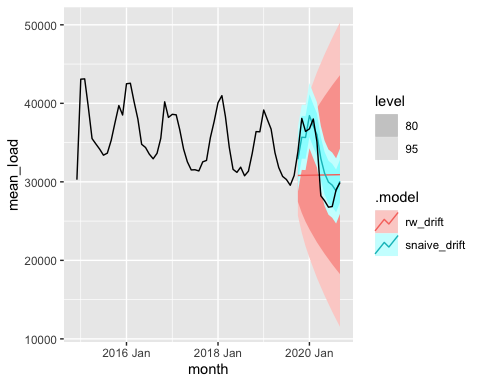


General Modeling Methodology

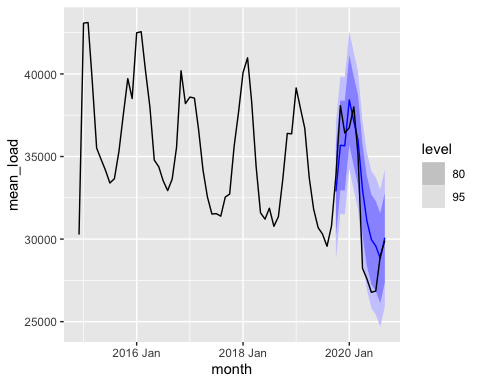
The training set has been experimented with some of the basic forecasting methods: random walk, mean, random walk with drift, seasonal naive, seasonal naive with drift. Now the forecast horizon, h, was set. Such that the last forecast based on your training set coincides with the last observation of the test set.

The model below includes the Box-Cox lambda in your models, as needed.

GB\_monthly\_sm <- GB\_training %>% model(  
 snaive\_drift = SNAIVE ( mean\_load ~ drift(), lambda = lambda\_GB ),  
 rw\_drift = RW ( mean\_load ~ drift(), lambda = lambda\_GB))  
  
GB\_monthly\_sm %>% forecast(h = "1 year" ) %>% autoplot(GB\_monthly)



GB\_monthly\_sm %>% select(snaive\_drift) %>% forecast(h = "1 year" ) %>% autoplot(GB\_monthly)



##Step 4: TSLM

Inside model() I integrated : tslm = TSLM(variable ~ trend() + season()) OR tslm = TSLM(variable ~ trend() + fourier(K = ZZZ)With the monthly data used, the former might work. For higher frequency data the harmonic regression is the way to go. I chose an optimal number (16) of Fourier terms by minimizing the AICc.

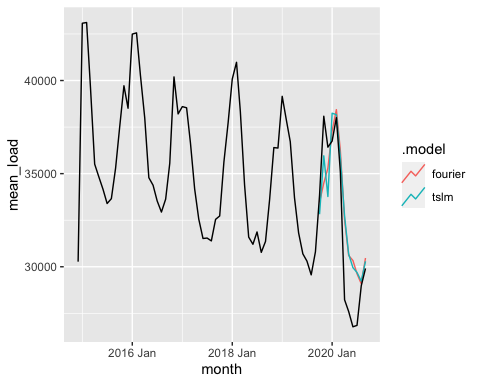
fit\_GB <-GB\_training

(.model, AICc, CV)

## # A tibble: 6 x 3  
## .model AICc CV  
## <chr> <dbl> <dbl>  
## 1 fourier3 877. 3489508.  
## 2 fourier4 879. 3528729.  
## 3 fourier1 879. 3763955.  
## 4 fourier2 879. 3711126.  
## 5 fourier6 880. 3417133.  
## 6 fourier5 881. 3510147.

The training model above shows 16 test for the fourier models.The best fit on the fourier training set is fourier3, which has the lowest AICc, and CV and this indicated a better accuracy in the model. This fourier3(K=3) was used in this forecast model below at 1 year horizon.

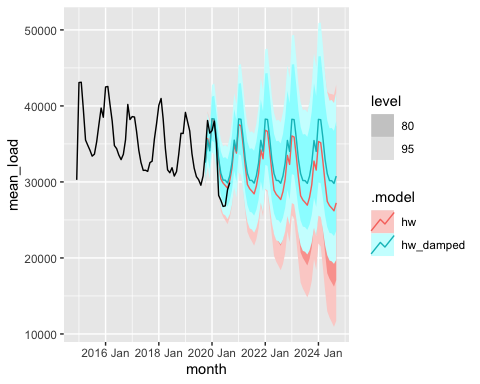
TSLM Model



Exponential smoothing and ETS

I skipped to H-W. There are four possibilities here for the trends: {additive, not damped}, {additive, damped}, {multiplicative (or exponential), not damped}, and {multiplicative (exponential), damped}. Since, there’s seasonality, I tried Holt-Winters’. In particular, this seasonality can be additive. All of these are bundled within the ETS allow ETS to optimize over the various options by minimizing AICc. ETS allows for the generation of confidence intervals.

ETS Models <- HW = ETS

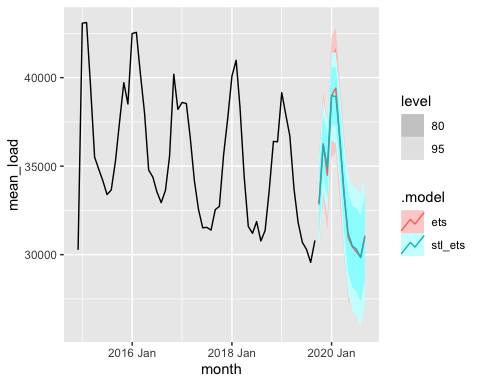


ETS of STL

ETS forecasts of STL seasonally adjusted data decomposes the data, seasonally adjusts it, then forecasts it with ETS.

The following code incorporates a Box-Cox transformation within the model. This includes a non-seasonally adjusted ETS with transformed data for comparison. Including the transformation this way ensures that the forecast data are back-transformed to original levels, and that appropriate bias adjustments are made to the forecasts.

stl\_ets <- decomposition\_model(  
 STL(box\_cox(mean\_load, lambda = lambda\_GB)),  
 ETS(season\_adjust ~ season("N"))  
)  
stl\_ets\_model <- GB\_training %>% model(  
 stl\_ets = stl\_ets,  
 ets = ETS(box\_cox(mean\_load, lambda = lambda\_GB))  
)  
  
stl\_ets\_forecast <- stl\_ets\_model %>% forecast(h = "1 year")  
  
stl\_ets\_forecast %>% autoplot(GB\_training)



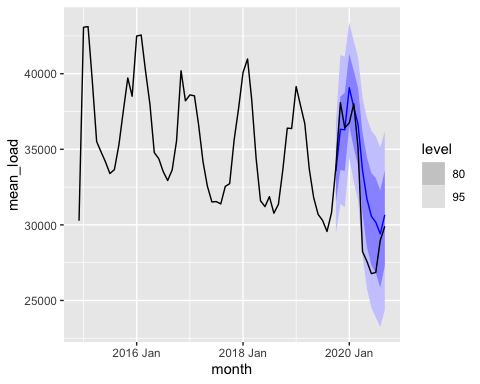
##Step 7: ARIMA

Thus far there hasn’t been any testing of residuals. If the data are non-stationary, then fitting an ARIMA model is a good option.

arima\_model <- GB\_training %>% model(  
 auto\_arima = ARIMA(box\_cox(mean\_load, lambda = lambda\_GB), stepwise = FALSE, approximation = FALSE)  
 )  
  
arima\_model %>% report()

## Series: mean\_load   
## Model: ARIMA(0,0,2)(0,1,0)[12]   
## Transformation: box\_cox(mean\_load, lambda = lambda\_GB)   
##   
## Coefficients:  
## ma1 ma2  
## 0.8883 0.4750  
## s.e. 0.1632 0.1879  
##   
## sigma^2 estimated as 1.997e+15: log likelihood=-875.03  
## AIC=1756.07 AICc=1756.64 BIC=1761.55

arima\_model\_forecast <- arima\_model %>% forecast(h = "1 year", bootstrap = TRUE)  
  
arima\_model\_forecast %>% autoplot(GB\_monthly)



tslm\_model %>% accuracy(GB\_monthly)

## # A tibble: 2 x 10  
## .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 fourier Test -944. 2314. 1792. -3.68 5.95 1.12 1.04 0.614  
## 2 tslm Test -907. 2300. 1871. -3.55 6.19 1.17 1.04 0.558

## # A tibble: 2 x 10  
## .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 hw Test -879. 2234. 1795. -3.43 5.94 1.12 1.01 0.545  
## 2 hw\_damped Test -1129. 2543. 2156. -4.34 7.16 1.35 1.15 0.613

stl\_ets\_forecast %>% accuracy(GB\_monthly)

## # A tibble: 2 x 10  
## .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 ets Test -1511. 2634. 2288. -5.43 7.56 1.43 1.19 0.548  
## 2 stl\_ets Test -1525. 2624. 2247. -5.49 7.47 1.41 1.18 0.581

arima\_model\_forecast%>% accuracy(GB\_monthly)

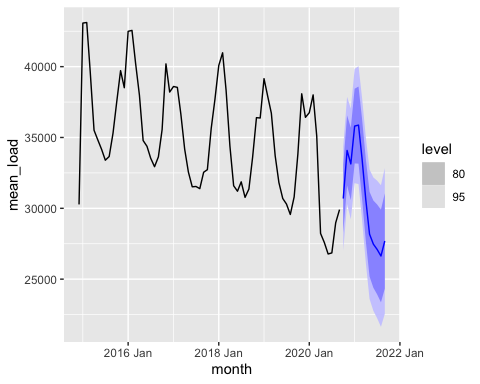
## # A tibble: 1 x 10  
## .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 auto\_arima Test -1609. 2634. 2007. -5.75 6.81 1.26 1.19 0.523

To narrow down to a few choices, I checked the accuracy measures, and looked at the forecasts to see which ones seem most plausible.

Use diagnostics (RMSE, etc.) to help identify which model or models appear worthwhile.

The forecasts of the top candidates to see if they pass visual inspection.

ets\_best\_models <- GB\_monthly %>% model(  
 hw = ETS(mean\_load ~ error("A") + trend("A") + season("A"))) %>%  
 forecast(h = "1 year" )   
ets\_best\_models%>%autoplot(GB\_monthly)



Finally, you take your top model or models and construct your actual forecasts. “Forecasting” your test set isn’t what you’re after… you actually want to forecast your variable out through September 2021. Write up your final forecasts.