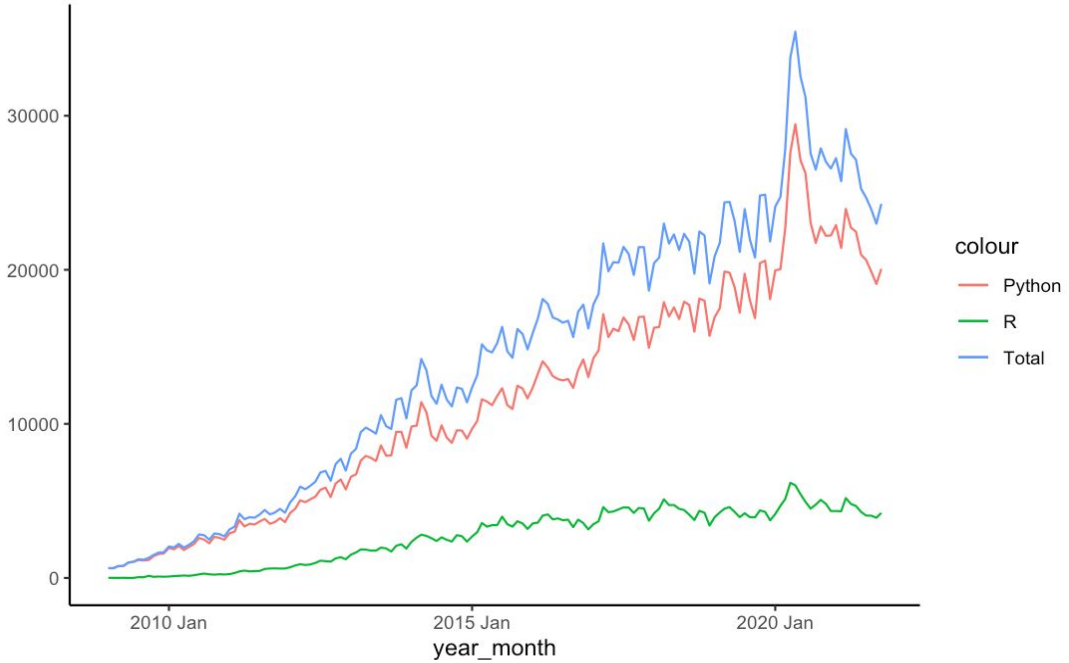
Forecasting StackOverflow R & Python Question Count

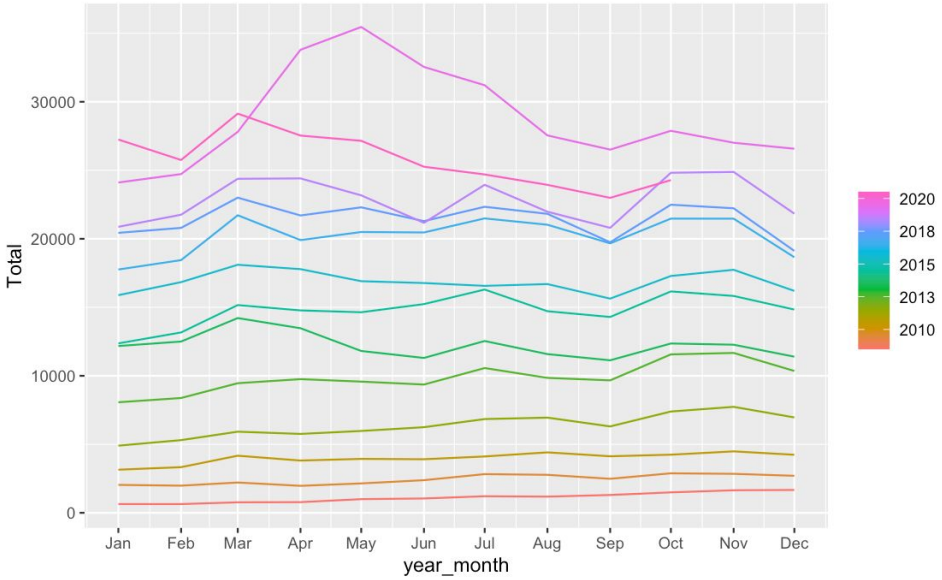
Ella Scholz
Fisher Latham
Sarah Carver

The Data Set



This series depicts the R, Python, and Total (R + Python) Question counts from StackOverflow over time.

Seasonality



gg_season() output

- Here we can see subtle consistencies with season as we look at year after year, pointing towards seasonality.
- We can also see a step change year over year suggesting an additive trend to the series.

Overall Approach

- Our first step was to clean our data and get a grasp on the combined series as a whole.
  - This included investigating and analyzing overall patterns in the series, such as seasonality and trend.
- Next, we ran a variety of forecasting models in order to determine which model would be the most accurate.
  - We chose the Root Mean Square Error (RSME) value as an indicator of accuracy.
  - The smaller the value, the higher its accuracy.
- Finally, we analyzed all of the models and their RSME value in order to select the most accurate model.

Data Cleaning 1

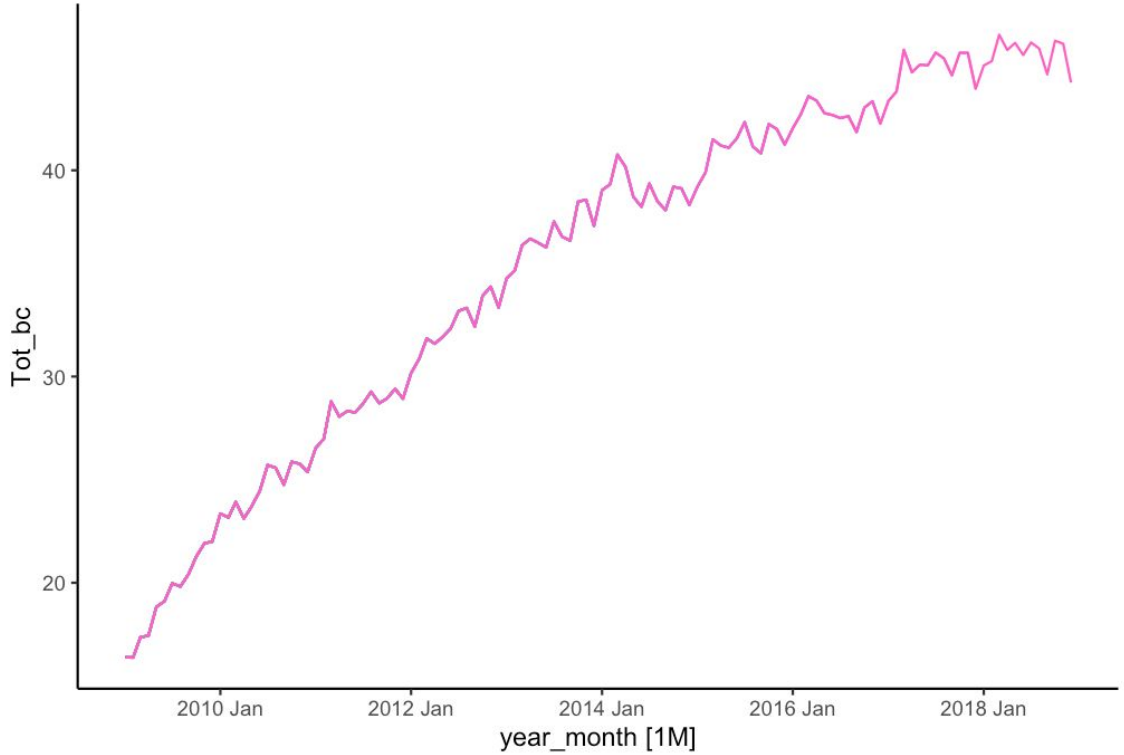
- Created a Date column by concatenating the separate time columns.
  - Formated day, month, year as dd-mm-YYYY (i.e. 01-01-2020)
  - Then, we converted to year month (January 2020) to make the date more readable.
- Constructed a Total column.
  - Total questions = Python questions + R questions
- Remove all incomplete years (ie 2008).
  - Contained months only for half of the year

Data Cleaning 2

- Replaced all missing values (NA) with 0.
  - We selected this fill method because having days with 0 questions is reasonable.
- Aggregated up to monthly.
  - The 0's (that replaced missing values) would not affect the overall count.
- Double our checked data.
  - Manually checked a handful of months to ensure that the clean data was correct.
  - Ensured there were no missing months, or values (regular time series).
- Transformed Total questions column
  - Made data more constant and linear
  - Box-coxed the variable to help with nonconstant variance

Cross Validation TRAIN Data Set After Data Cleaning:

→ From this graph, we now have a full workable data set with less heteroskedasticity.

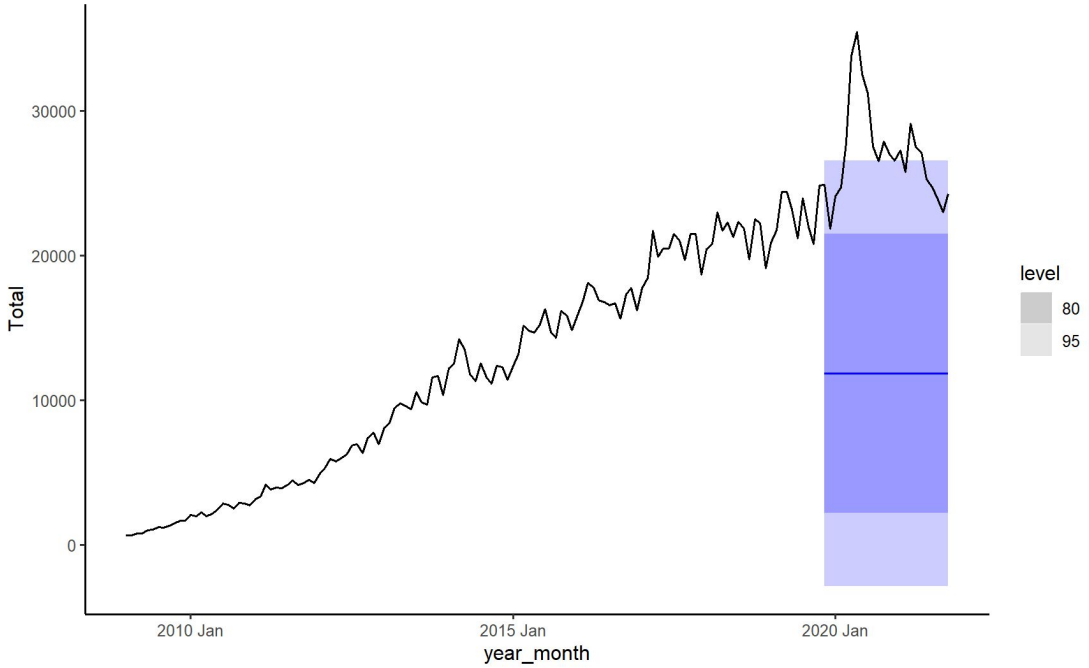


Simple Models

- Initially, we chose to forecast with the four simplest models so as to get a read on their effectiveness.
- The following Models, along with their RSME were analyzed:
  - Mean Model
  - Naïve Model
  - Seasonal Naïve Model
  - Drift Model

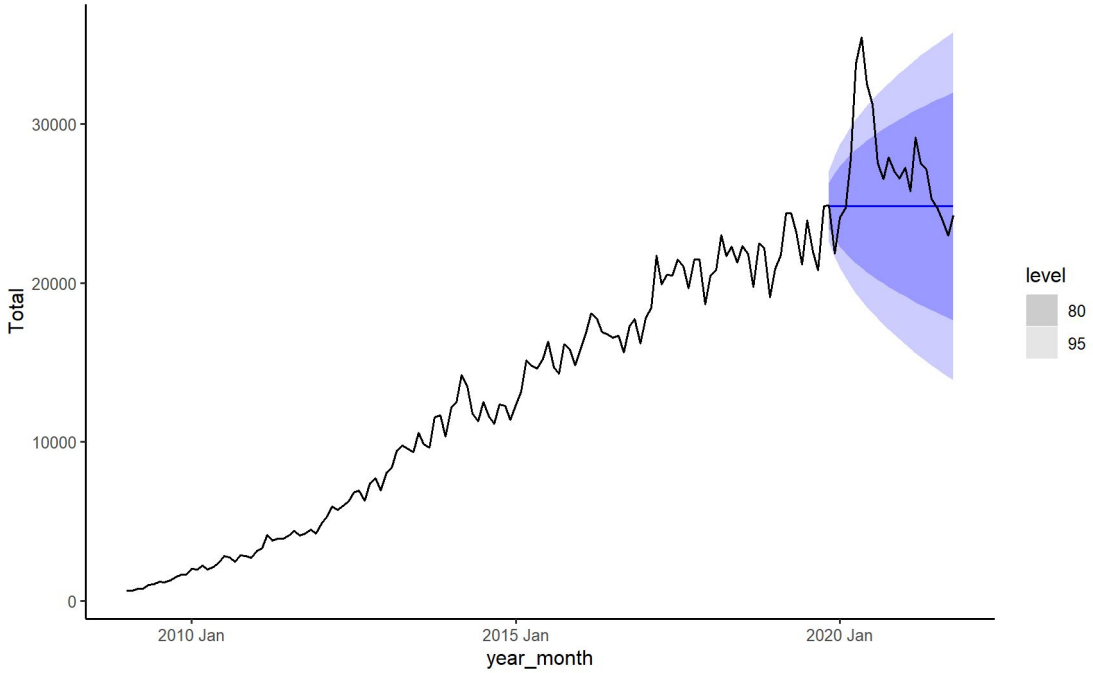
The Mean Model

- This model predicts the future values as the average of the entire series.
- The RMSE value for this model is 15,591.
- Thus, on average, our mean model predictions are within 15,591 questions of the right answer.



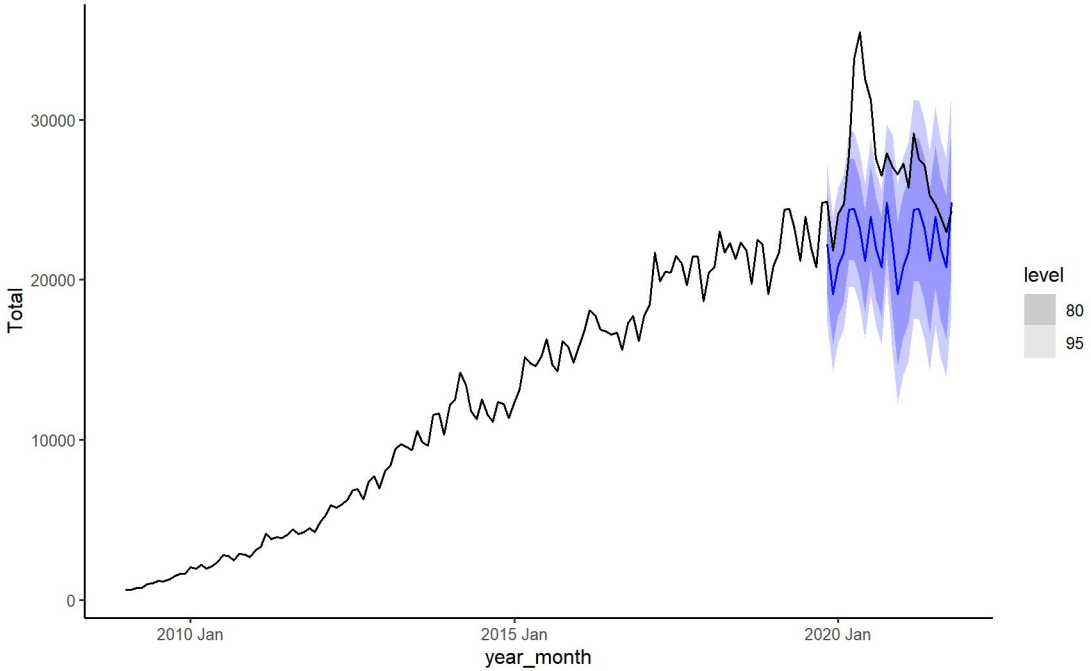
The Naïve Model

- This model predicts the future values as the last value of the entire series.
- The RMSE value for this model is 3,994.
- Thus, on average, our mean model predictions are within 3,994 questions of the right answer.



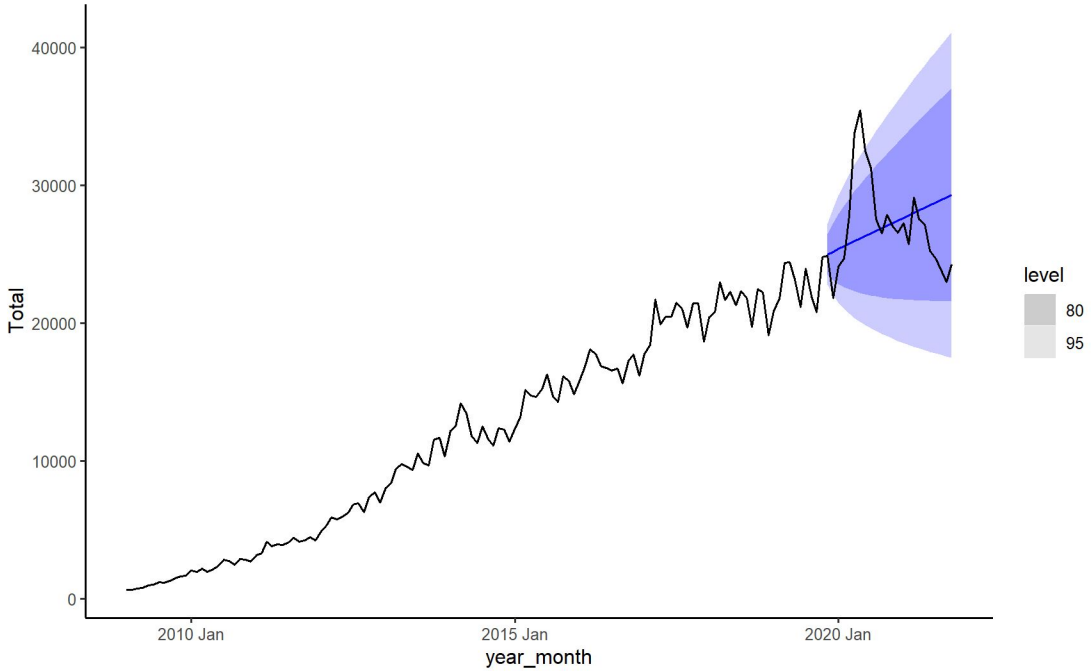
The Seasonal Naïve Model

- This model predicts the future values as the value of the last season.
  - Thus, the model is able to account for seasonality.
- The RMSE value for this model is 5,588.
- Thus, on average, our mean model predictions are within 5,588 questions of the right answer.



The Drift Model

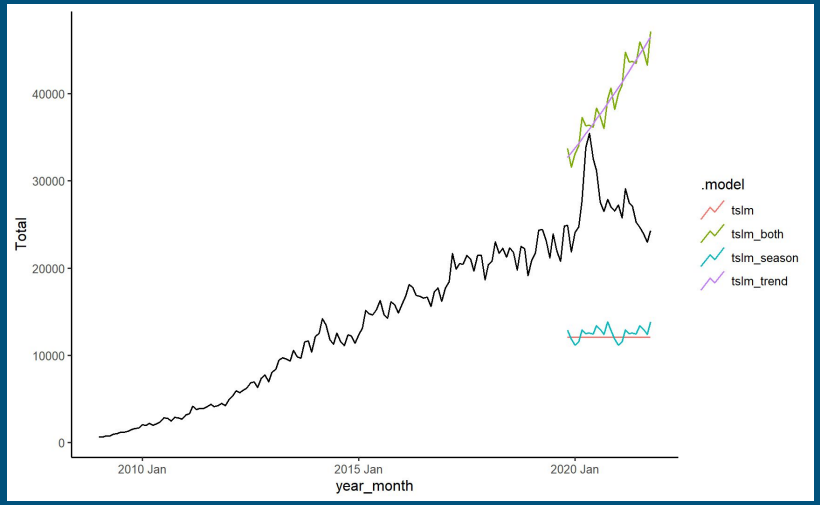
- This model predicts the future values by drawing a line through the first and last point of the series.
- The RMSE value for this model is 3,834.
- Thus, on average, our mean model predictions are within 3,834 questions of the right answer.



Time Series Linear Model (TSLM)

- Next, we chose to forecast with different modifications of the time series linear model.
- The following Models, along with their Root Mean Square Error (RSME), were analyzed:
  - TSLM with no features
  - TSLM with just trend
  - TSLM with just season
  - TSLM with both trend and seasonality

Different Modifications of the Time Series Linear Model



TSLM: Essentially the mean model

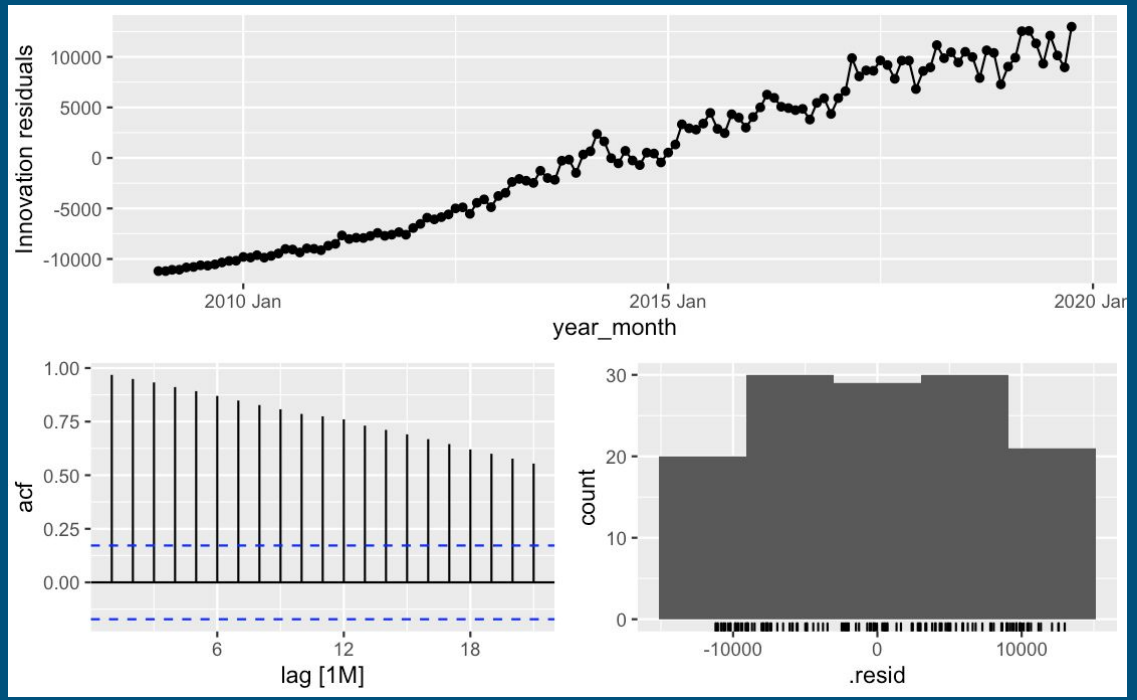
TSLM_Both: Accounts for both seasonality and trend

TSLM_Trend: Accounts for just series trend

TSLM_Season: Accounts for just series seasonality

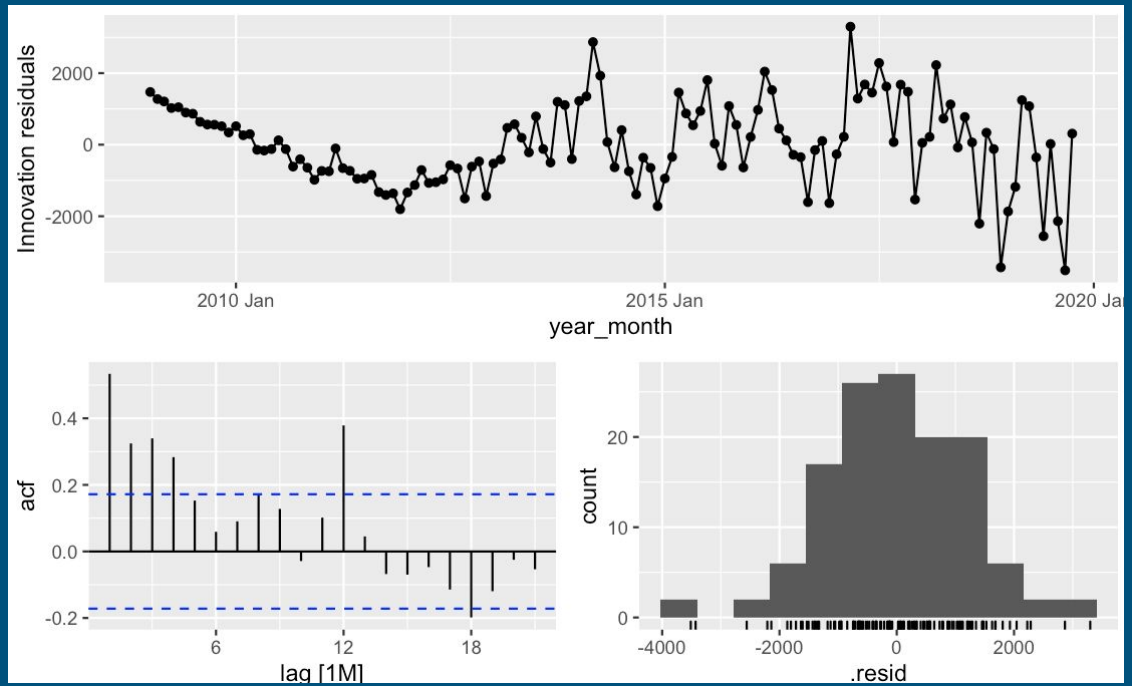
TSLM Model

- In both the top graphic and bottom left graphic, we can see that there is something that is not being accounted for within the series, which looks to be trend as the graph has a continuous upward.
- The RMSE value for this model is 15,364.
- Thus, on average, our TSLM model predictions are within 15,364 questions of the right answer.



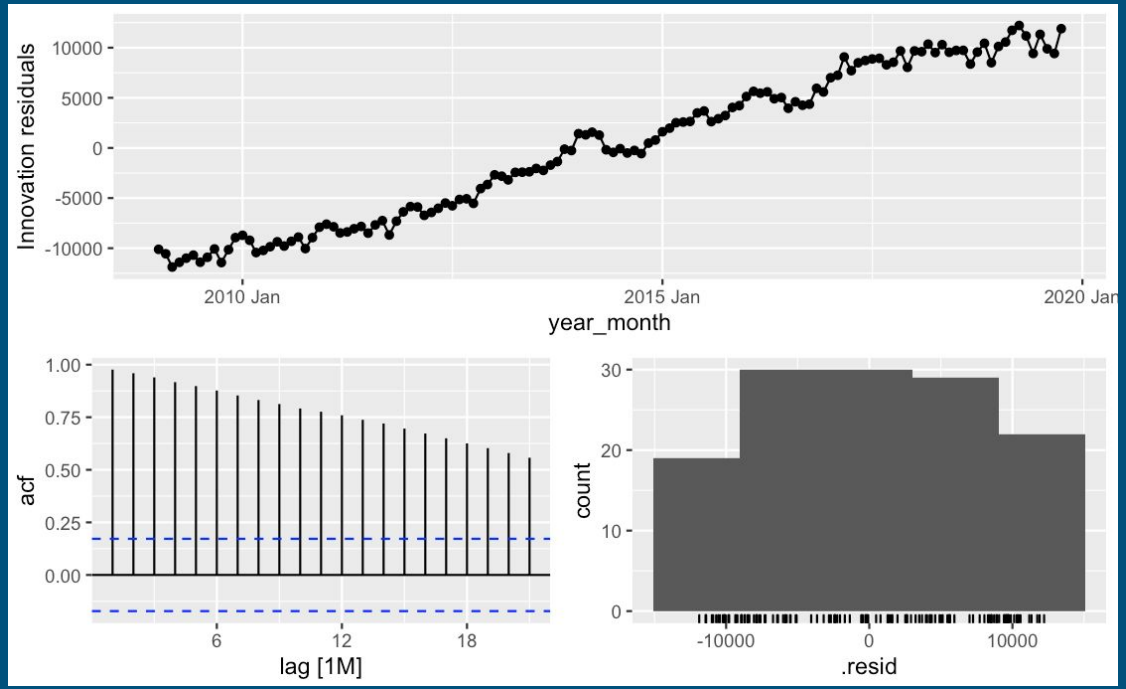
TSLM With Trend Only

- In both the top graphic, we can see that we have now accounted for trend as our series is more level. However, we can now see that there is some hints at seasonality as we see continuous troughs and peaks
- The RMSE value for this model is 13,625.
- Thus, on average, our TSLM model predictions are within 13,625 questions of the right answer.



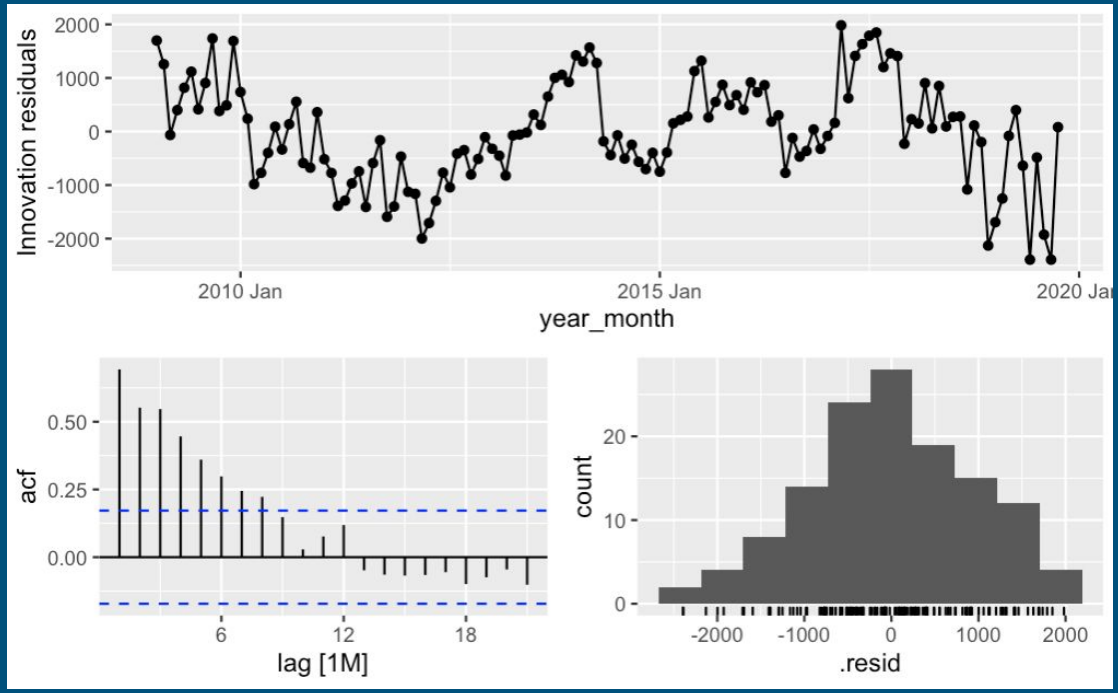
TSLM With Season Only

- As we account for season and take out the trend aspect, we can see that trend is still a major factor left in our residual plot.
- The RMSE value for this model is 16,097.
- Thus, on average, our TSLM model predictions are within 16,097 questions of the right answer.



TSLM With Both Trend and Season

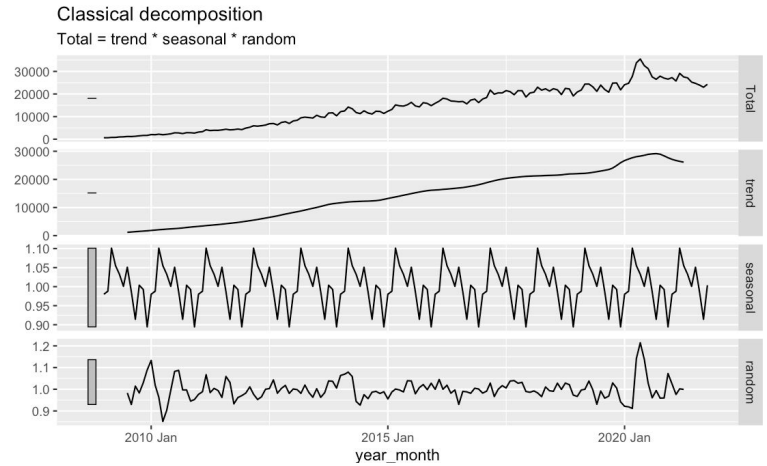
- As we account for both season and trend in our TSLM model we can see that the topic graphic is more sporadic and un-patterned. Known as white-noise, this is what we are looking for in our residual plots. Meaning this is most likely our best candidate for the TSLM modification.
- The RMSE value for this model is 13,641.
- Thus, on average, our TSLM model predictions are within 13,641 questions of the right answer.



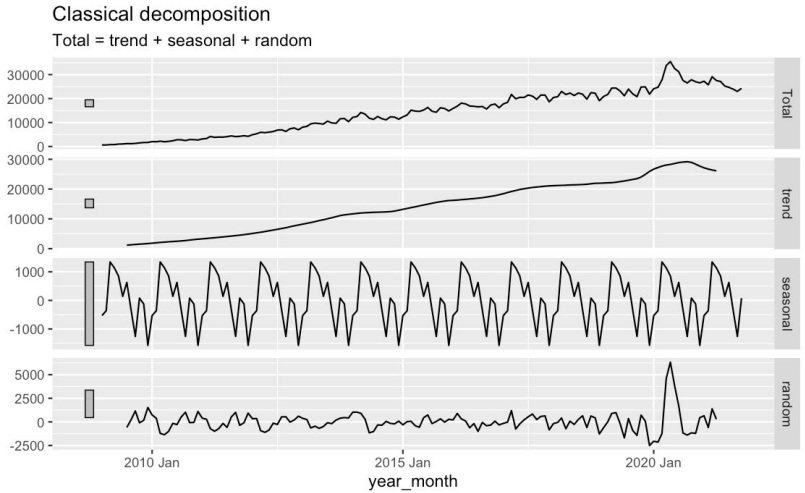
Exponential Smoothing Model (ETS)

- Next, we chose to forecast with different modifications of the exponential smoothing models.
- Specifically, we used Holt-Winters, that accounts for both seasonality and trend. As we saw earlier these both had significant effects on accounting for the series
- The following Models, along with their Root Mean Square Error (RSME), were analyzed:
  - Autoselected ETS
  - ETS with all Additive features
  - ETS with mixed features

Looking at Seasonality and Trend



Multiplicative Decomposition



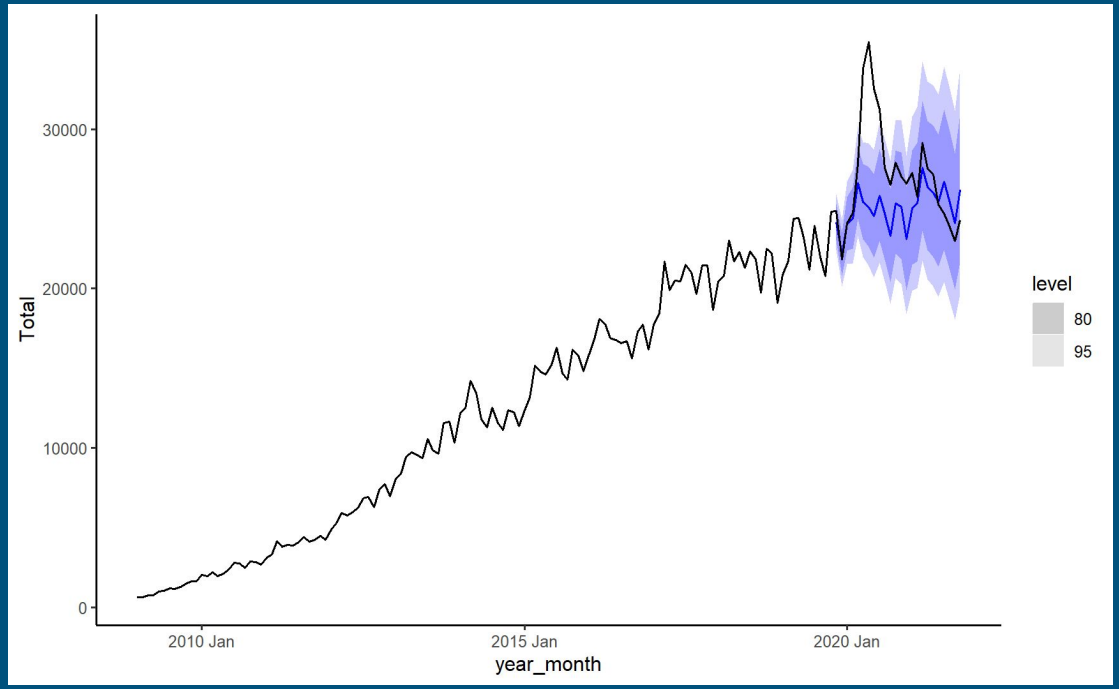
Additive Decomposition

Comparing these two decompositions, we see that additive trend and seasonality is able to capture more of the patterns in our data than the multiplicative alternative. The additive error looks more like white noise. Thus, we believe that our data has additive trend, seasonality, and error, which we will use to determine the features of the following ETS models.

Autoselected ETS

Additive Error
Additive Dampened Trend
Additive Season

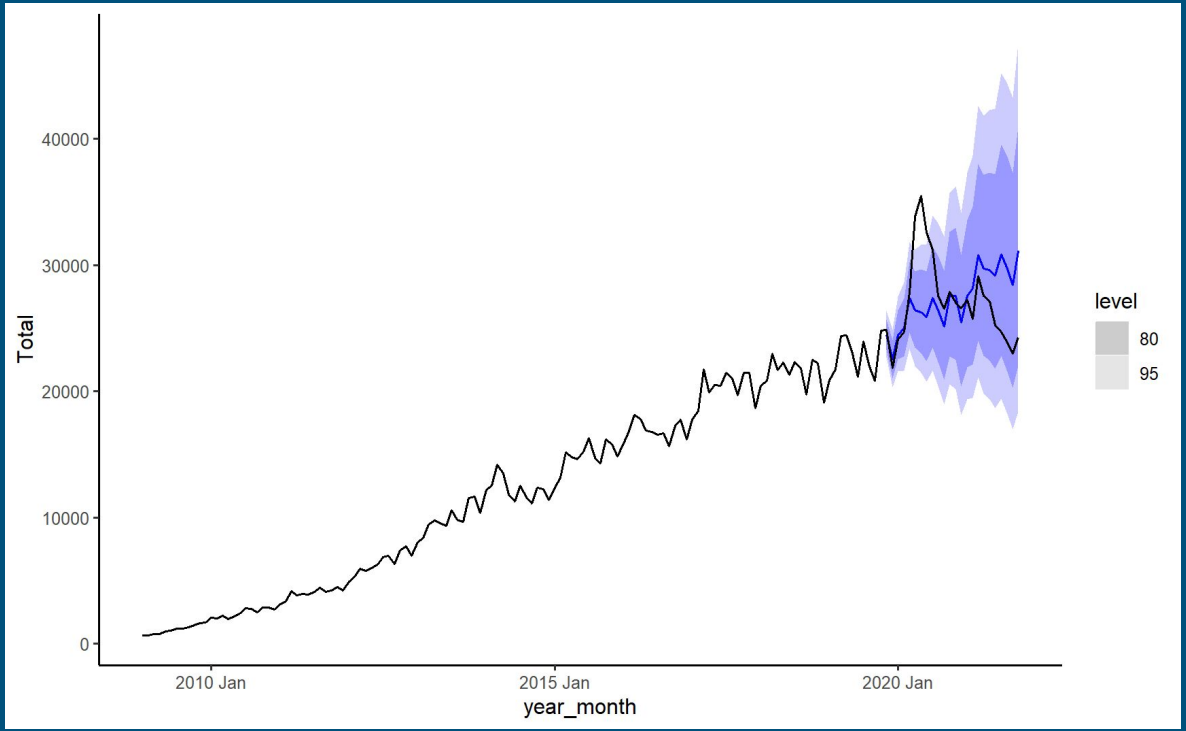
- Here we allowed the computer to select the model combination it best saw fit within the ETS model. This is a very pessimistic outlook on the series as we dampen trend significantly as we forecast.
- The RMSE value for this model is 3,737.
- Thus, on average, our TSLM model predictions are within 3,737 questions of the right answer.



ETS

Additive Error
Additive Trend
Additive Season

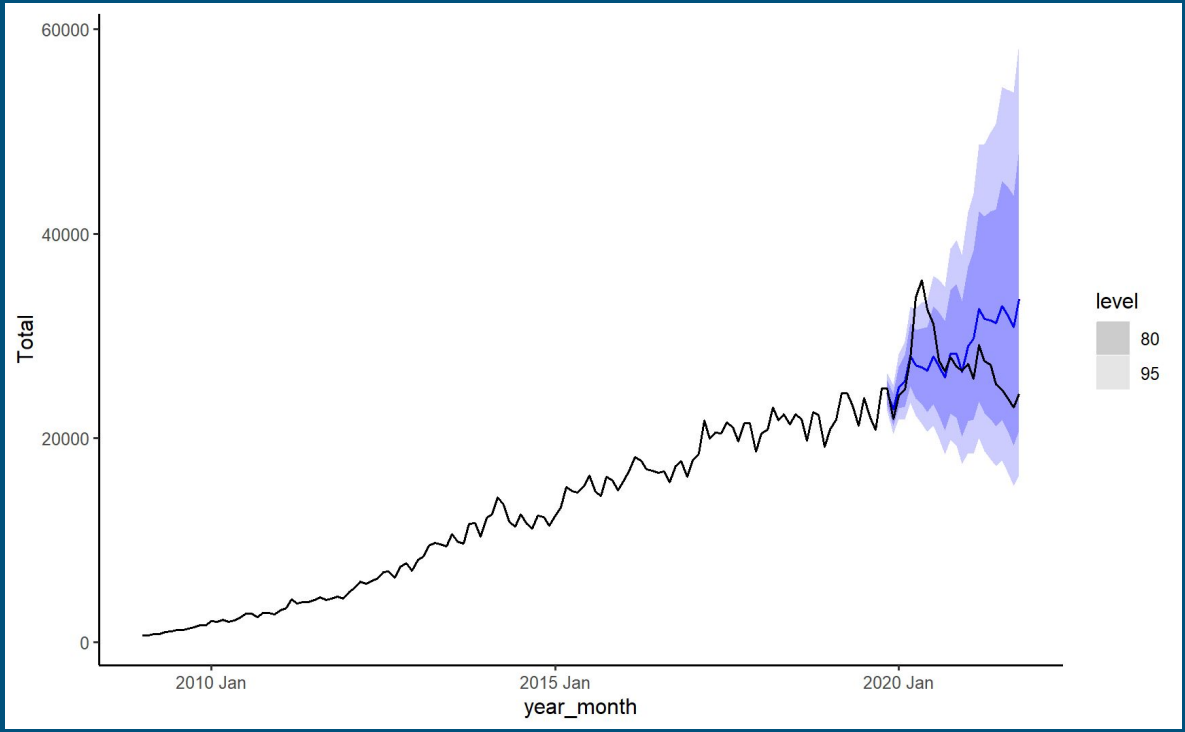
- For this model we selected to account for all additive elements as we saw that the overall series looked to be additive in all of these aspects
- The RMSE value for this model is 4,017.
- Thus, on average, our TSLM model predictions are within 4,017 questions of the right answer.



ETS

Additive Error
Multiplicative Trend
Additive Season

- Because the overall series looked like it could have a slight multiplicative trend, so we decided to test its validity within the ETS model.
- The RMSE value for this model is 4,654.
- Thus, on average, our TSLM model predictions are within 4,654 questions of the right answer.

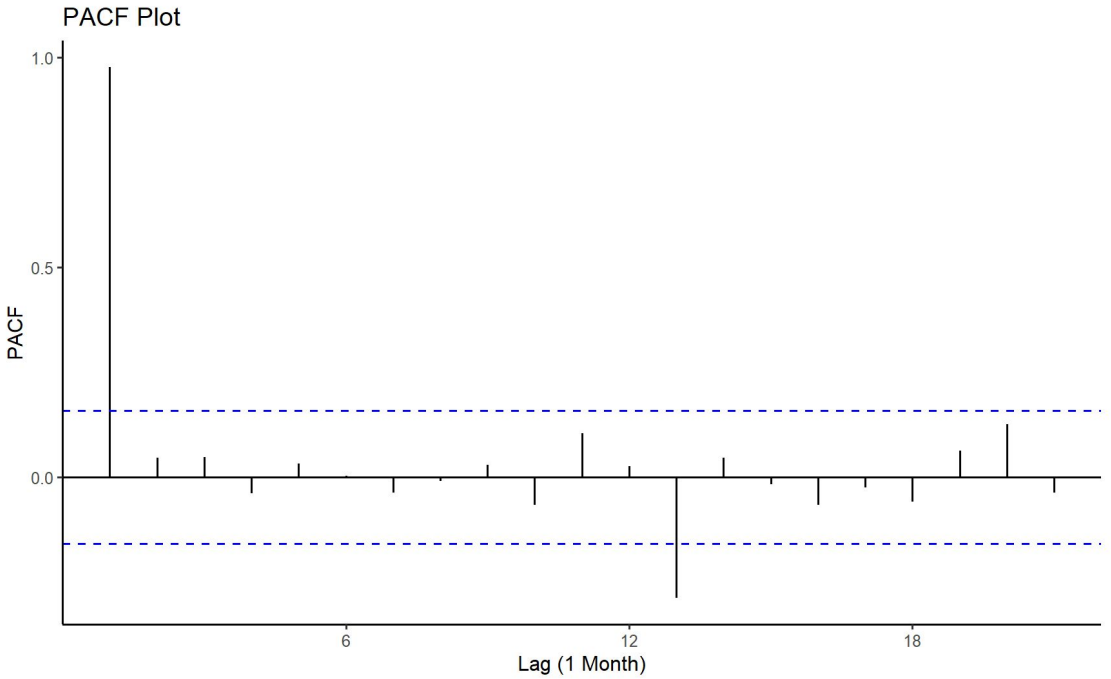


Autoregressive Integrated Moving Average Model (ARIMA)

- Next, we chose to forecast with different modifications to the ARIMA.
- The following Models, along with their Root Mean Square Error (RSME), were analyzed:
  - Auto Arima
  - Arima with manually selected features

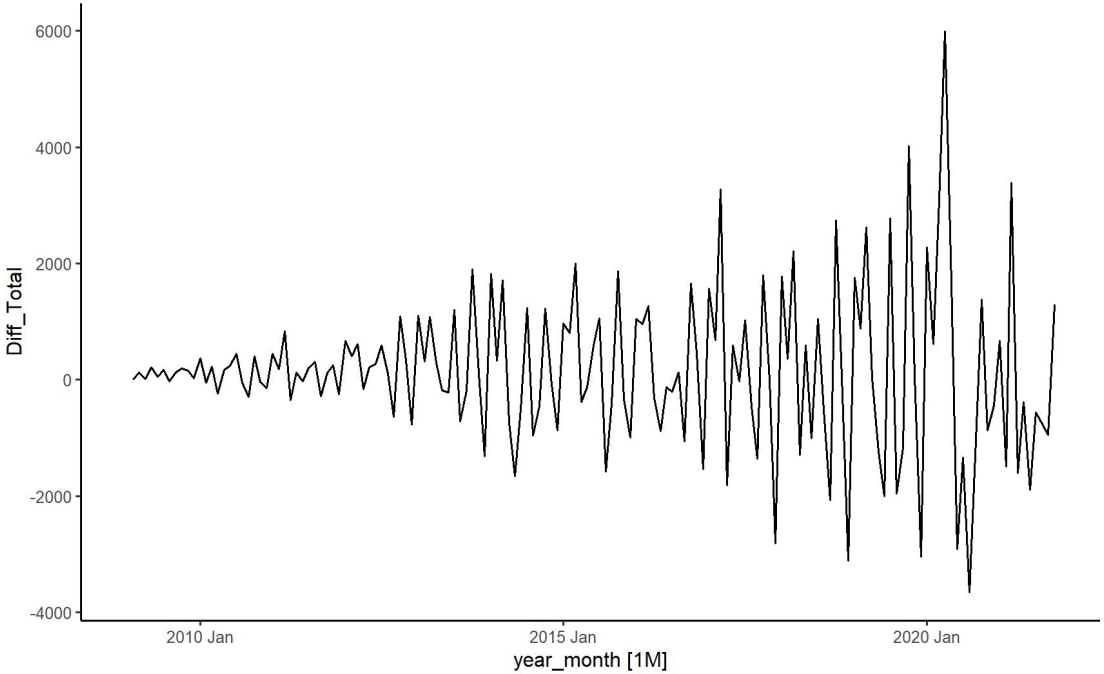
Partial Autocorrelation Function (PACF)

- We can from our PACF plot that we should have 1 autoregressive term for both p and P.
- “p” is the number of non-seasonal lags.
  - 1 big spike at start
- “P” is the number of seasonal lags.
  - 1 big spike at 13 months



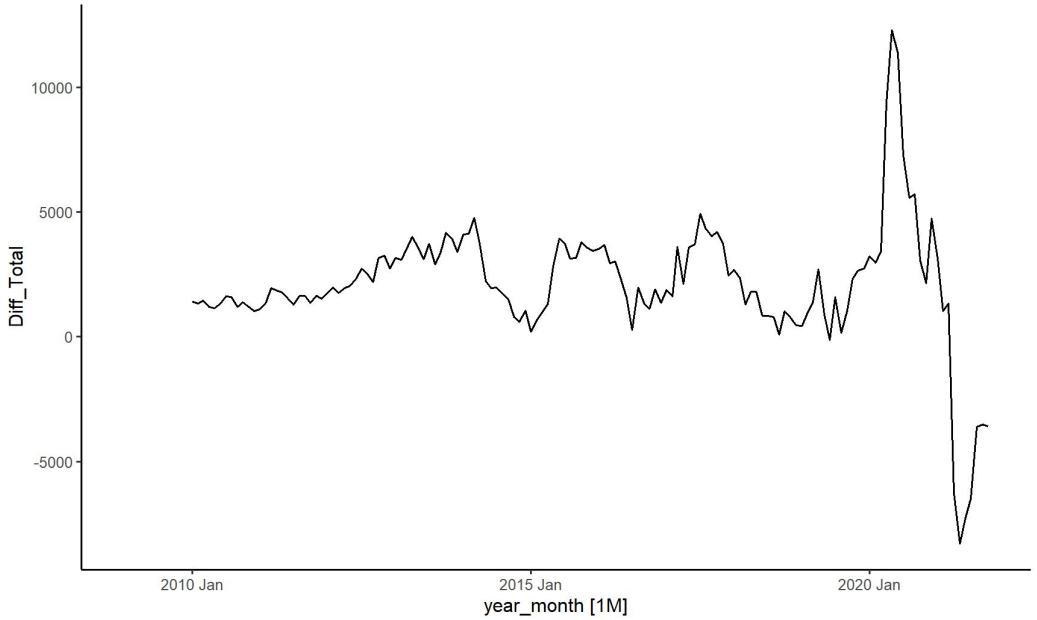
Stability/ Difference

- We can see that after 1 difference, our data is stable.
  - Thus, d is 1.
- For context, “d” is the number of times the data is differenced.



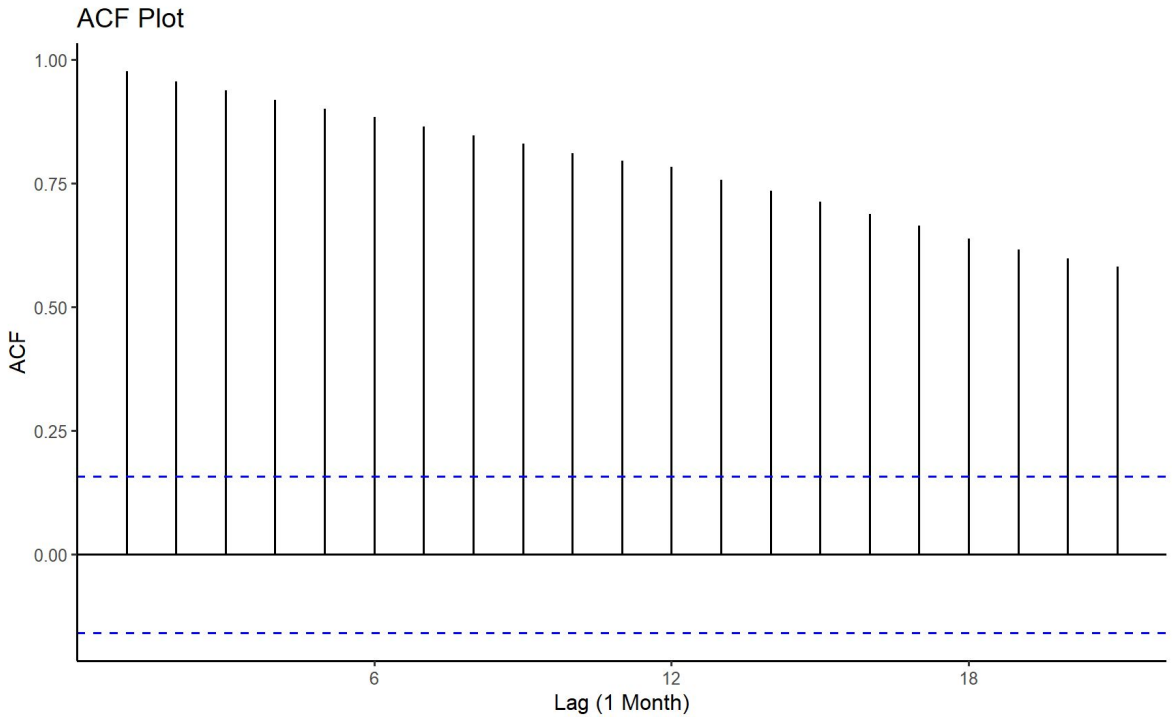
Stability/ Seasonal Difference

- After 1 seasonal difference, our data also becomes relatively stable.
  - Thus, D is 1.
- For context, “D” is the number of times the data is seasonally differenced.



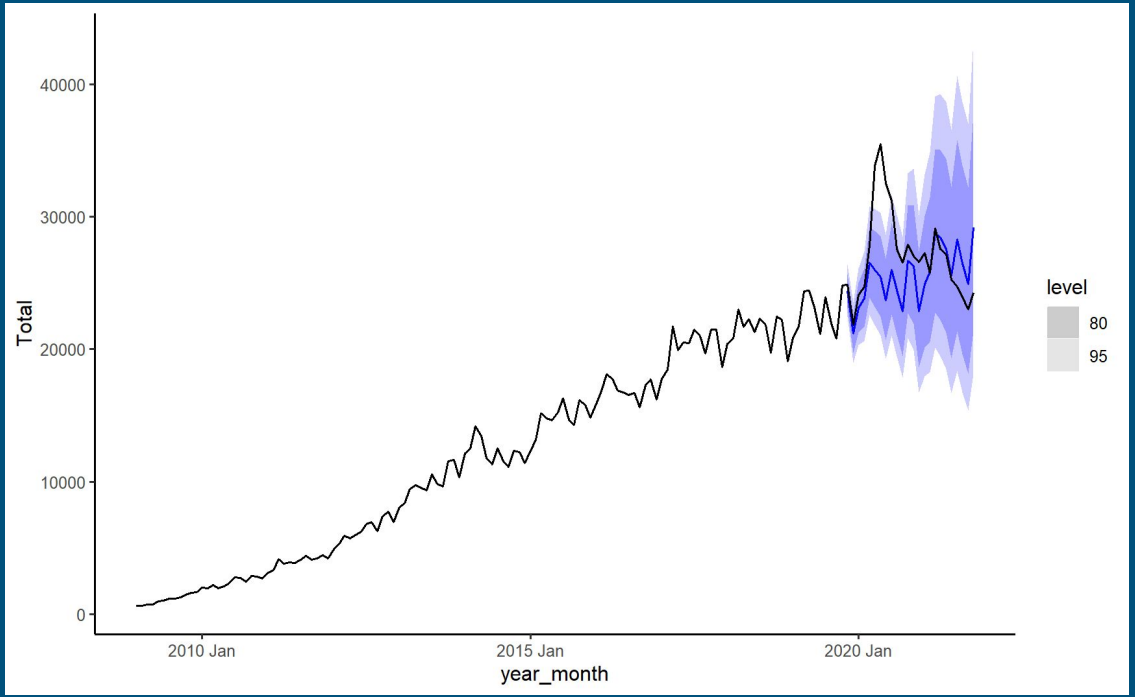
ACF

- We can see from our ACF plot that we should have 0 lagged errors for both q and Q.
- “q” is the number of non-seasonal lagged errors.
- “Q” is the number of seasonal lagged errors.



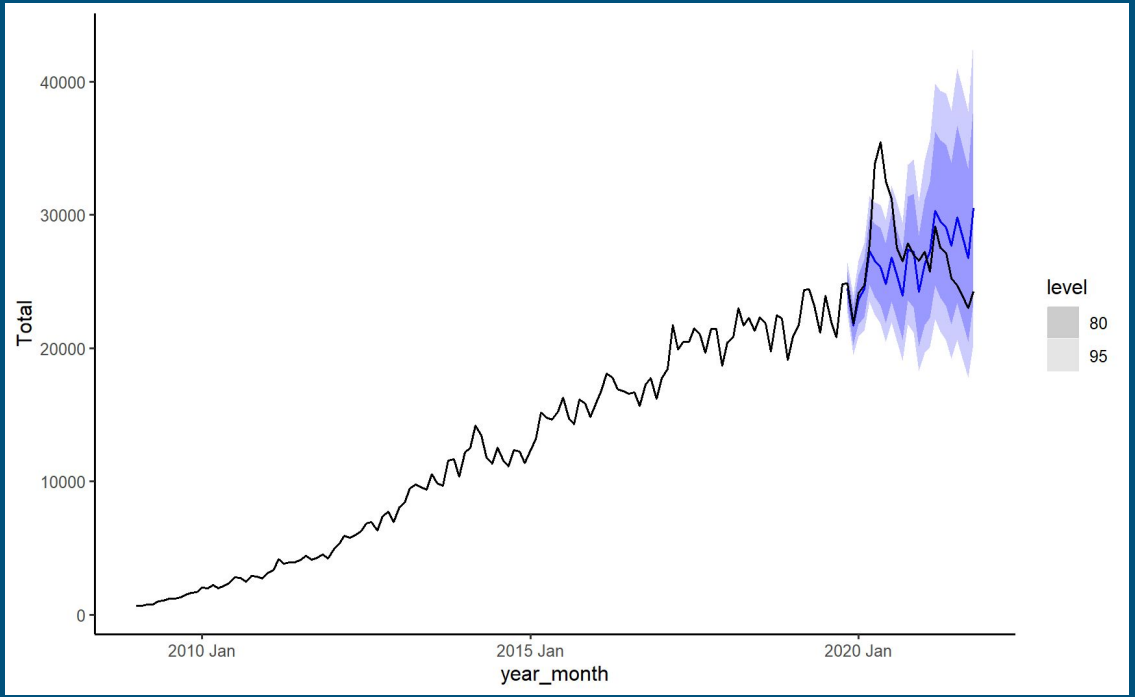
Manual ARIMA (ARIMA(0,1,1) x (0,1,1)[12])

- The arima model, was the hand chosen ARIMA based on the previous p,d,q and P,D,Q.
- RMSE is 4,950.34.
- Thus, on average, our manual ARIMA model predictions are within 4,951 questions of the right answer.



Auto ARIMA

- This was the computer model based on the training data.
- RMSE of 3, 839.
  - This means that on average, our predictions are within 3,839 questions of the right answer with the auto ETS model.



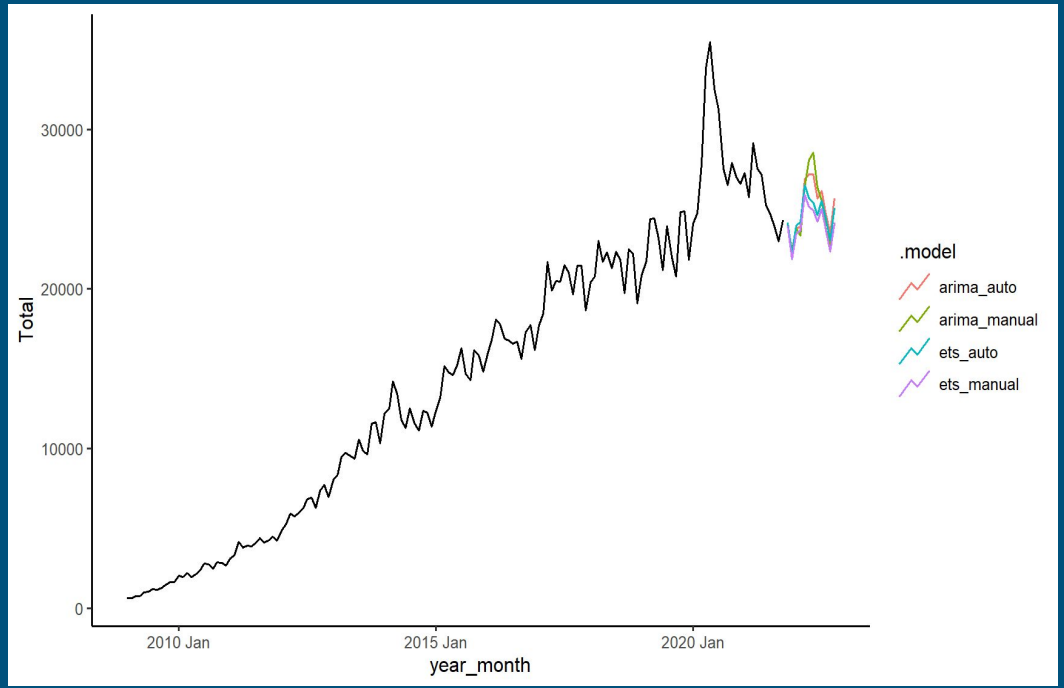
Selecting a Model

Exponential Smoothing Model

with additive errors, dampened additive trend, and additive seasonality

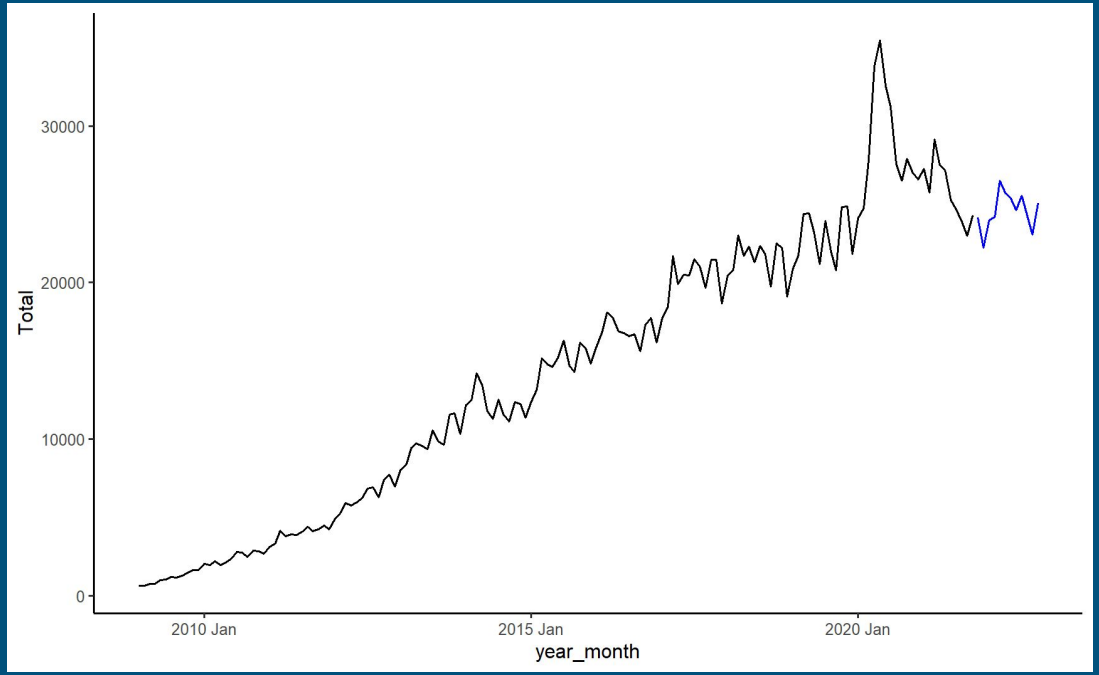
Comparing and Selecting a Model Utilizing Cross Validation

- Utilizing on cross validation, we are going to choose the auto ETS model as it had the lowest RMSE of 1,182 compared to the other models that had a RMSE of ~1,400.
- This means that on average, our predictions are within 1,182 questions of the right answer with the auto ETS model.



Final Forecast

- We chose our auto ETS model to forecast total questions 12 months into the future.
- Our auto ETS model is a Holt-Winter model with additive errors, dampened additive trend, and additive seasonality.



Final Selection & Thoughts

- Although some simpler models seem to work better on the data, we could see through cross-validation that the more complex models like ETS and ARIMA performed better on multiple tests leading us to believe the more complex models would perform better for new data.
- Through the cross validation, in which we cycled through the data multiple times to get an average answer, we saw that the auto ETS model with damped trend performed the best on multiple tests.
- Perhaps forecasting somewhat cautiously, it makes the most sense to use the ETS auto model, especially after an unusual spike during the COVID-19 lockdown era.