Applied Economic Forecasting

Your Name Here

Homework #4b - Spring 2021

Instructions: In all cases, please ensure that your graphs and visuals have properly titles and axes labels, where necessary. For your convenience, I have posted my R markdown file on our course website so that you can open and alter as you see fit. Refer to the output, whenever appropriate, when discussing the results. **Lastly, remember that creativity (coupled with relevance) will be rewarded.**

Forecasting the US Unemployment Rate

Between the period January 2009 – December 2019, the U.S. unemployment rates were on the decline. We know that this has changed since the January 2020 as a result of the pandemic.

In our analysis below, we will pull data from `FRED` using the `quantmod` package and see how well we could have forecasted the unemployment statistics for 2019 using our basic time series decomposition techniques.

(a) Using the `getSymbols` command from the `quantmod` package, import the non-seasonally adjusted monthly US unemployment (`UNRATENSA`) data from FRED.

(b) Edit the code below to store the data from January 2010 to December 2019 and convert to a `ts` object called `unrate.ns`.

```
# Remember to set eval to TRUE to run!!
yourvarhere <- UNRATENSA["2010::2019"]
unrate.ns <- ts(coredata(unrate), start  = youenterstart, frequency = youenterfreq)
```

(c) Report and discuss the (i) time series plot and (ii) any seasonality and/or trend in the data.

Based on your plot, would you agree that an additive model is appropriate? Explain.

(d) Using the `window` command, assign the values in `unrate.ns` before 2018 as a training set called `unrate.train`. Assign the values after 2018 to a variable called `unrate.test`.

(e) Using the **training set** above and a `STL` decomposition with a periodic seasonal windows, present the plot of the decomposition of the `unrate.train` series. You will find it best to store the model result first.

Note: Use `unrate.train[,1]` in the `stl` command. It seems to throw an error otherwise.

(f) Produce a plot of the **seasonally adjusted** data.

(g) Superimpose the seasonally adjusted and trend-ycle series onto the training data.

You will need to change your line colors as we did in class to ensure that each series is visible.

(h) Using a `naive` model and the `forecast()` function, produce predictions for the unemployment data for 2019. *I am asking you to produce forecasts of the reseasonalized series for the next 12 months.*

- Please store your predictions in a variable called `pred.unrate`.
- Create an autoplot of `pred.unrate`.

(i) Now use the `naive` method within the `stlf()` function to reseasonalize and plot the results as you did above. **Ensure that you keep using the training data here.**

(j) Produce an autoplot of the test data and autolayer the prediction values (`pred.unrate[["mean"]]`). Briefly discuss the visuals here.

The emphasis of your discussion should be on how well this simple STL model captures the dynamics of the actual data that we tried to forecast.