

# AAEC 5484: Applied Economic Forecasting

Your Name Here

## Homework #4 - Spring 2023

**Instructions:** In all cases, please ensure that your graphs and visuals have proper titles and axes labels, where necessary. Refer to the output, whenever appropriate, when discussing the results. **Lastly, remember that creativity (coupled with relevance) will be rewarded.**

### Question 1: Revisiting US Natural Gas Consumption

In the last homework, we explored the US Natural Gas Consumption series the U.S. Energy Information Administration (EIA) and determined that the seasonal naïve model was preferred.

Since none of the models (**drift**, **naïve**, **seasonal naïve**, and **mean**) were able to simultaneously capture the trend & seasonality in the consumption data, we will use a regression model to see if we can improve our forecasting ability.

1. Using the codes from your last homework, import the NG consumption data from the EIA into R. In particular, repeat steps 1 – 3 of Homework #3.

**Place this in a single code chunk. At the end of this process you should have `tsng` expressed in Billion Cubic Feet.**

2. Now, let us re-familiarize ourselves with the data.

Use the `autoplot()`, `ggsubseriesplot()`, and `ggACF()` functions to comment on your observations. **Be sure to explain how each graph allows you to conclude whether there is seasonality and/or trends in the data.**

**I would like to see you use the `grid.arrange()` function to combine your plots as a 3 x 1 grid. Remember to use the `fig.height` argument in your code chunk to increase the size of your plots, if needed.**

3. Repeat step 6 of HW3 and subset the `tsng` data into a training and test set, `train.ng` (ending December 2019) and `test.ng` (December 2020 - December 2022), respectively. Next, perform step 7 of HW 3 to ensure that your subsetting is correct. **A title is not necessary here.**
4. Using the `TSLM()` function, regress `Demand`, from `train.ng` on a trend.
  - Store the model fit as `trend.fit`.
  - Report the model summary.
  - Interpret the coefficient on trend.
5. Now produce a plot of the actual values (from the training data) with the fitted values from your regression superimposed.
  - **You might find it easiest to extract these variables from the `augment()` function and then plot this using the `ggplot()` command. See class notes.**
  - **Ensure that both series are appropriately labeled in the legend**
6. Using the `gg_tsresiduals()` function, perform a diagnostic test of the residuals from `trend.fit`.
  - **What pattern emerges according to your ACF and time plot? Be sure to explain, in detail, how you came to this conclusion?**

- Do these residuals appear to be White Noise? **Use the Ljung-Box Test (with max lag,  $\ell$ , set to  $2 \times m$ , where  $m$  is the frequency of the data) to support your conclusion.** Remember to explain the null and your conclusion.
- Using the **train data**, regress Demand on the `season()` dummies.
    - Store your model fit as `season.fit`.
    - Again, report the model results.
  - From your regression results, what is the average NG consumption during January within our training set?
  - How would you **interpret** the value on `season()year6`? To get full credit here, you must discuss both the sign and magnitude of the coefficient.
  - What is the average demand for **October** over the training sample?
  - Briefly interpret the coefficient of determination,  $R^2$  of this model.
  - Produce a time plot that contains both the fitted values (from `season.fit`) and the actual data in our training set.
    - **Be sure to include labels for each series.**
    - **Briefly comment on anything you notice here.**
  - Perform a diagnostic test and Ljung-Box test (with  $\ell = 2 \times m$ ) then comment on any discernible pattern in the residuals of `season.fit`. Specifically, are any of the assumptions possibly violated?

Useful hints for answering this question:

- **Does the autoplot of the residuals exhibit any noticeable pattern?**
  - **What does the ACF tell you about the residuals of the model stored in `season.fit`?**
  - **What does the Ljung-Box test tell you regarding autocorrelation?**
- Now use the `TSLM()` function to estimate a regression of Demand on a trend and the `season` dummies. Again, your focus is on the training data set.
    - Store the estimation results in a variable called `t.season.fit`.
    - Report the results of `t.season.fit`.
  - Produce a time plot of the fitted values of `t.season.fit` and the actual data, `train.ng`. Be sure to include labels for the each series.

Now briefly (1 sentence will suffice) comment on anything you notice here relative to the plot from `season.fit`.

- Again, perform a diagnostic test of this model and comment on any discernible pattern in the residuals of `t.season.fit`. Does the autocorrelation problem seem to be corrected?

Useful hints for answering this question: - What does the Ljung-Box test tell you regarding autocorrelation? - What does the ACF tell you about the residuals of this model? Do they appear to be white noise?

- If there is still autocorrelation in the residuals, can you think of at least one variable that might have been omitted from our regression but could improve the explanatory power of our model?
- Using the `glance()` function, determine the “best” regression model based on the in-sample data (training data set). Your criteria of interest are the `adj_r_squared`, AIC, AICc, BIC, and CV statistic. **Be sure to explain how you come to your conclusion.**

Present your results using the `knitr::kable()` function with `digits = 2` and **thousands separators**. I would like to see your `adj_r_squared` column expressed as  $\bar{R}^2$  using Latex notation.

- Use the `forecast()` function to produce the predictions for the **next 3 years** (the length of the test data) using the models stored in `trend.fit`, `season.fit`, and `t.season.fit`. Store these as `fore.trend`, `fore.seas`, and `fore.t.seas`, respectively.

**You are only required to store the model forecast results in this step.**

20. **I am interested in comparing the results above to the models from the last module (and HW3). In particular, I want to know if all this work was worth it and if the regression model outperformed our previously preferred model. I have solicited your help in this regard.**

Admittedly, there are several ways to merge the forecasts with the data and produce our plots. We will however take the path of least resistance. **Similar to slide 75 of our Lecture 4 notes and step 8 of HW3, reproduce all 7 model forecasts (made from `train.ng`) as a single object.** Call this object `ng.fore`.

**Be sure to give each model an appropriate name in the `model()` function.**

21. Now, visualizing the model fits for our 7 models– three (3) regression model and our 4 benchmark models– against the **test data set**, `test.ng`. You might find it easiest to simply `autolayer()` the forecasts (`ng.fore`) unto the `autoplot()` of `Demand` from `test.ng`. **Please turn off the PIs here.**
22. Using the `accuracy()` command, compare the forecasts of **all 7** models against the **full data**.
- Is the seasonal Naive model still preferred under the RMSE, MAPE, and MASE selection criteria? If not, which model now has that honor?
  - Display your results as a table using the `knitr::kable()` function:
  - Format your data using 2 decimal places.
  - Ensure you have appropriate column and row labels.
  - Include an appropriate table caption.