Final Exam

BUAD 5032 – Fall 2021

---

1. **Objectives**
   The purpose of this assessment is to give you an opportunity to show a clear understanding of the skills you have learned in the second part of this course. Mainly, it is intended to assess your understanding of Time Series forecasting.

2. **What You Will Need**
   Access to R and RStudio.

3. **What You Will Hand In**
   Submit an R script that you create (Final*FirstnameLastname*.R) via Blackboard - Final.

4. **Due Date**
   Wednesday 15, 2021. The final must be turned in by 8:00am EST. Late assignments will be penalized.

5. **Note on Collaboration**
   This is a Category A assignment. Specifically, you may not receive help from anyone on this assignment. It must be 100% your own work. *It is an honor code offense to give or receive any assistance on these assignments*. Your personal class notes are allowed. You are not allowed to use the internet.

6. **Honor Code**
   The Pledge: "As a member of the William and Mary community, I pledge on my honor not to lie, cheat, or steal, either in my academic or personal life. I understand that such acts violate the Honor Code and undermine the community of trust, of which we are all stewards."

Requirement: You must type and print the honor pledge at the beginning of your R script. Failure to do so will result in a 20-point deduction.

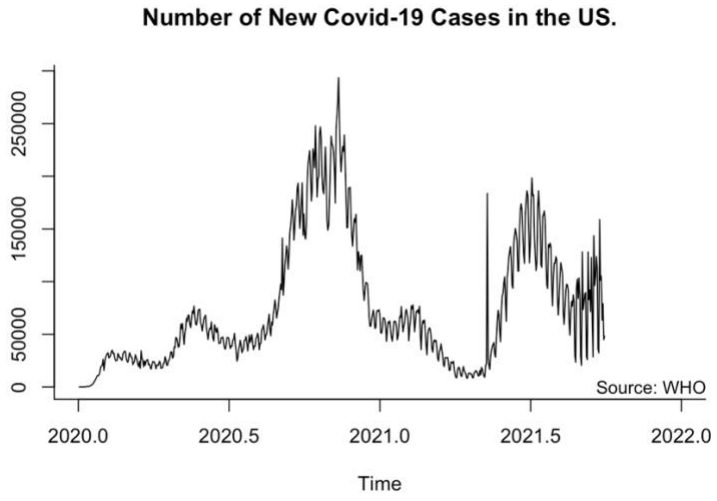
Time Series Forecasting

In this exercise you will forecast Covid-19 new cases in the US with real data from WHO. Plot checks are provided, but feel free to ignore. Keep completing the exam even if these plots don't match for you.

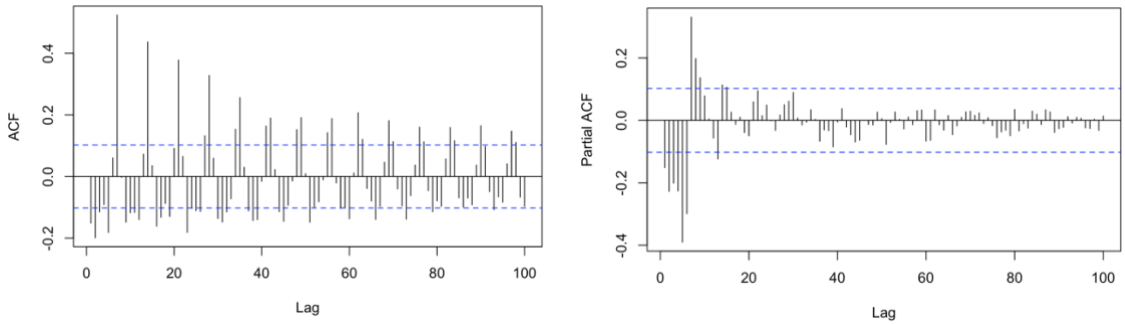
1) Start by cleaning up the environment with rm(list=ls()), loading the "forecast" and "glue" packages and uploading the "Covid.csv" file and saving it to an object called "covid"..

2) Create a time series object (ts) from the "new_cases" variable with "frequency=365" starting at (2020,3). Create a train set that starts 3/2/20 and ends 11/30/21. Call this time series object *train.* Create a test set that start 12/1/21 and ends 12/7/21. Call this time series object *test.* Hint: the end of the train set should be set to 2021.7475, and the start of the test should be 2021.748.

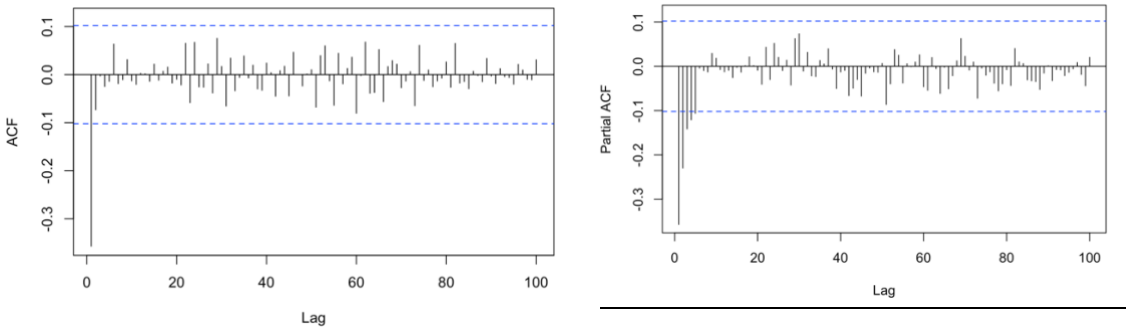
3) Plot the training set. In particular, set the main title to "Number of New Covid Cases in the US", remove the y-label, set *bty="l"*, and the x-limit from 2020-2022. Add text that reads "Source: WHO". Hint: use the text() function with "cex=1". Your plot should look like this:



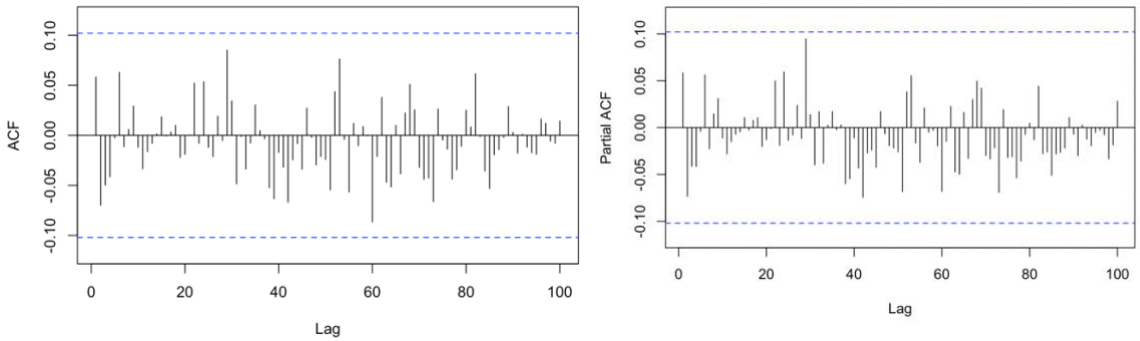
4) Use the *ndiffs()* function to determine whether your series is stationary. Plot the ACF and PACF functions for the differenced series and a 99% confidence interval (we will use this confidence interval for the rest of our analysis). Use the option "lag.max=100" to better observe the pattern. Note the slow decaying behavior in the ACF, spikes every 7 days, and some spikes around the 7th lag in the PACF. Your plots should look like this:



5) Estimate an ARIMA(3,1,0) for your seasonal model with the *Train* data set. Save the model in an object called *Smodel.* Check the residuals and confirm that the seasonal pattern is no longer present in the data by checking the residuals in the ACF and PACF plots. Only estimate the seasonal model. Make sure you set the non-seasonal component to an ARIMA(0,1,0) to make the series stationary and "period=7". Your new ACF and PACF:

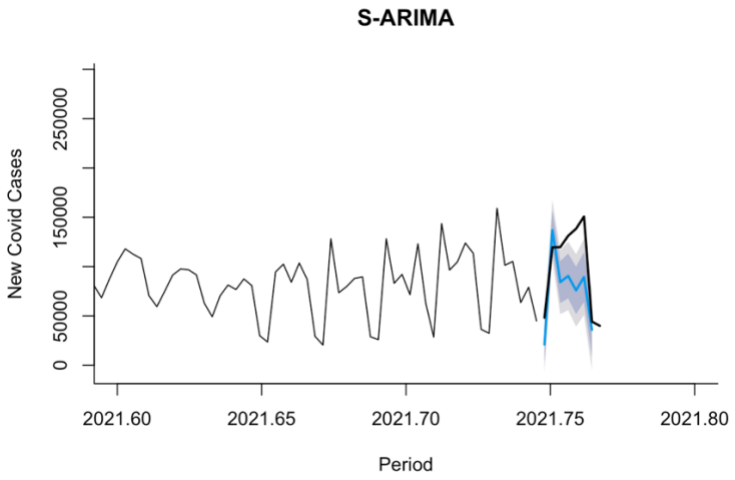


6) Propose a model that includes both the non-seasonal and seasonal components from what you observe in 5). Print a statement using glue explaining the patterns observed and the reasoning behind your choice. Call your model *model_sarima.* Note: This is a textbook pattern. There is only one correct answer. The ACF and PACF of your residuals should look like this after running the model with both seasonal and non-seasonal components:



7) Run the command “auto.arima()” to estimate a second model. Call this model *model_auto*. Does this model perform better than the model in 6)? Use glue() to answer the question. Use the BIC to justify your answer. Note: Ignore the warning sign when running auto.arima if you get one.

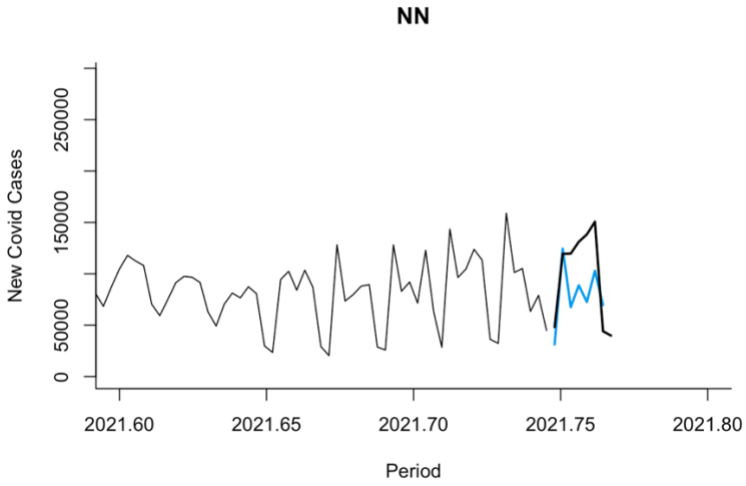
8) Use the “forecast()” function to forecast the best model for the test data period. Call the forecast object *forecast_sarima*. Plot the train set, the test set and your forecast. To do this use the plot() function and pass the forecast_sarima object you just created. Set the “xlim=c(2021.6,2021.8)”. Include the test set in your plot by using the lines() function. Here is the resulting graph:



9) Save the accuracy of the model (on the test set) in a data frame called *Accuracy_Models.* Only include the first five measures of accuracy. Name the column generated *S_Arima*.

10) Set the seed to 10. Estimate a Neural Network model using the train data. Run: *nnetar(train, scale.inputs = TRUE)*. Call this model *Model_NN*. Use the “forecast()” function once again to forecast the NN model for the test data period. Call the forecast object *forecast_NN*. Save the first five accuracy measures of the NN model as a new column in the Accuracy_*Models* data frame. Call the column created *NN*.

11) Plot the forecast of your model along with the training set and test set. Here is the plot generated:



12) Consider creating an ensemble of your models. Choose the weight of each forecast (S-Arima and NN) such that it minimizes the RMSE of the ensemble's predictions on the test set. Precision should be set at two decimal places.
Calculate accuracy measures of the ensemble (with the best alpha) and include it in the *Accuracy_Models* data frame. Print a statement using glue() with your answer and final recommendation. Which model should we use to predict covid when considering the RMSE?
Hint: create a vector of alpha's "*seq(0,1,0.01)*". Then use a for loop to create a vector of RMSE's for all the alphas in the vector. Retrieve the alpha that yields the least RMSE on the test set.

Peer Evaluation:

Complete the peer evaluation doc file. The average grade of your peers is worth 10% of your grade.