HW2_Q3

2023-01-29

Download and examine the data set linked below that contains the maximum daily temperatures (averaged monthly; abbreviated aMDT) in the city of Dallas, TX from January 2000 to December 2020.

For questions requiring 95% confidence bands, use 1.96 standard deviations on each side of the mean to determine the 95% span.

Question 3a.

a. Consider the task of modeling this data using a SARIMA model. Based on your knowledge of monthly variation in temperature, what value would be most appropriate for the seasonal lag term, S?

ANSWER: “Data taken quarterly will exhibit the yearly repetitive period at s = 4 quarters. Natural phenomena such as temperature also have strong components corresponding to seasons. Hence, the natural variability of many physical, biological, and economic processes tends to match with seasonal fluctuations”[1]. Accordingly, it makes most sense to use S=4 to correspond with the seasons (spring temperatures will be different from Summer, Fall, and Winter. Though Spring is likely closer to fall then Summer or Winter, it’ll likely still be distinct enough to make a difference!)

Question 3b.

b. Using the seasonal lag selection in the previous subquestion, fit the SARIMA(,,)(,,) model to the full aMDT time series for all combinations of p, d, q, P, D, and Q in {0,1} except the four cases where P=1, D=0, Q=1, and d=0. (Hint: this means you should be checking 60 different combinations). In answering this question, you should fit the various models to the full data set (do not split it into a training/test split) and assume that delta = 0 (where delta is the constant term). Identify which of these models best fits the data and report the AICc value for this model and the estimated values of the unknown parameters.

```
### Set vars
AICc_matrix = matrix(data=NA, nrow=0, ncol=7) ### Init a matrix to store AICc values
fn = "MeanDallasTemps.csv"
delta = 0 ### Per HW prompt
S = 4 ### Chosen seasonal lag = 4 for the seasons

### Read in CSV
df1 = read.csv(fn)

### Store fit data in the matrix initialized above. Length of final matrix = 60
for (p in 0:1) {
  for (d in 0:1) {
    for (q in 0:1) {
      for (P in 0:1) {
        for (D in 0:1) {
          for (Q in 0:1) {
            if (P!=1 || D!=0 || Q!=1 || d!=0) {
            ### Fit WITHOUT plots. DO NOT SPLIT DATASET, use full df. No constant = True as delta = 0
            arma_res = sarima(df1$AvgTemp, p = p, d = d, q = q, P=P, D=D, Q=Q, S=S, no.constant=TRUE,
details=FALSE)

            AICc_matrix = rbind(AICc_matrix, c(p,d,q,P,D,Q,arma_res$AICc)) ### Store AICc in a matrix
for easy ref
            }
          }
        }
      }
    }
  }
}

### Convert to a dataframe for easier use
AICc_df = as.data.frame(AICc_matrix)
col_nm = c("p","d","q","P","D","Q","AICc")
colnames(AICc_df) = col_nm

### Get the lowest AICc from the generated matrix for the best params
min_AICc = min(AICc_df$AICc)
AICc_min_df = filter(AICc_df, AICc == min_AICc)

### Display AICc matrix for reference
print(AICc_df)
```

From Shumway and Stoffer: “The value of k yielding the minimum AIC specifies the best model[1]”:

ANSWER: The model orders that best fits the time series are (using SARIMA(p,d,q)x(P,D,Q)S format): SARIMA(1, 1, 0)x(0, 0, 1) with S = 4 and an AICc value of 6.661616

ANSWER: The estimated values of the unknown parameters are

```
sarima(df1$AvgTemp, p = AICc_min_df$p, d = AICc_min_df$d, q = AICc_min_df$q, P=AICc_min_df$P, D=AICc_
min_df$D, Q=AICc_min_df$Q, S=S, no.constant=TRUE, details = FALSE)$ttable
```

```
##      Estimate     SE t.value p.value
## ar1    0.4575 0.0661  6.9260       0
## sma1  -0.3029 0.0715 -4.2332       0
```

Question 3c.

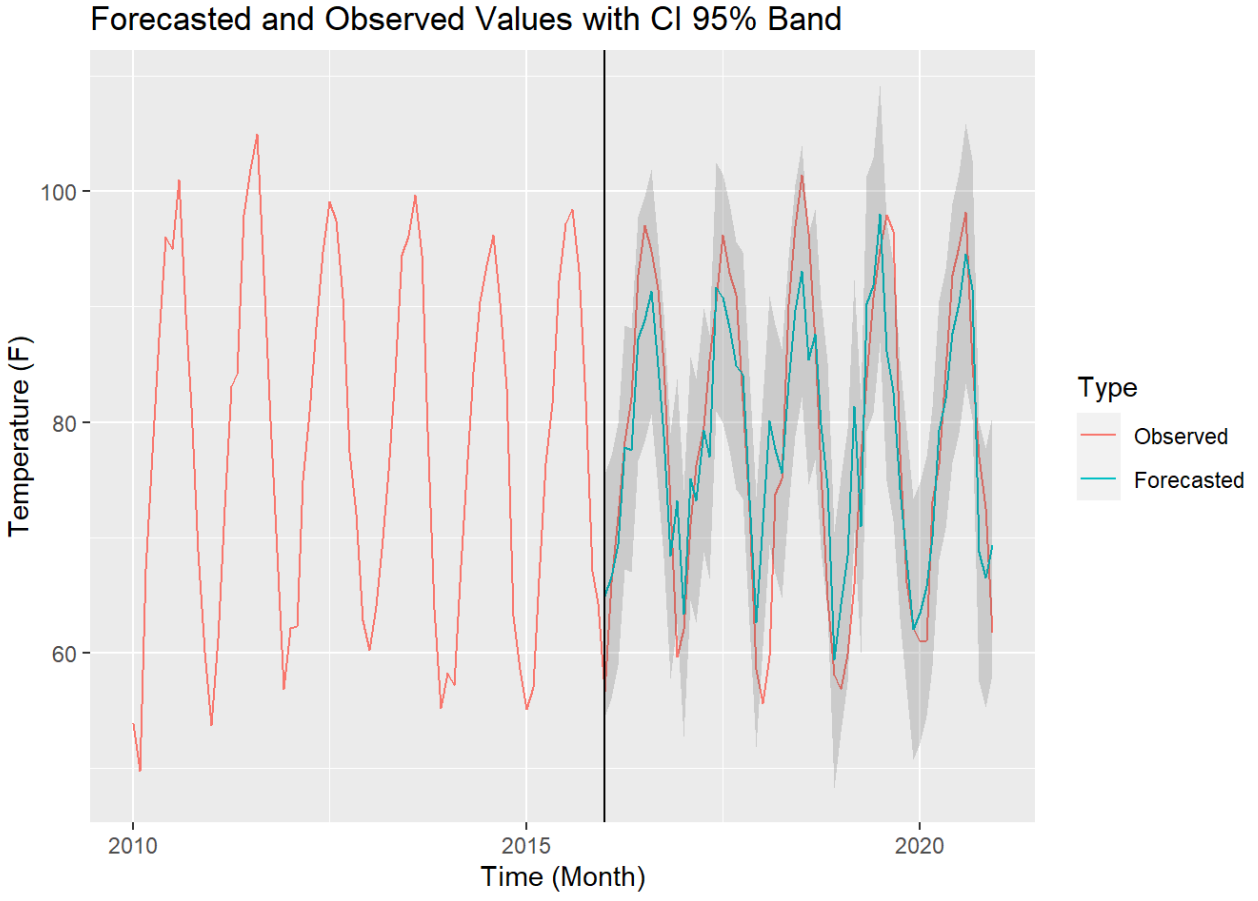
c. Consider the task of forecasting the aMDT twelve months in advance. For the last five years of data (2016-2020), predict the value of aMDT using all of the data up until one year prior to the prediction (i.e. predict the aMDT for January 2016 using all of the data up to and including January 2015, then add in the observed aMDT for February 2015 and predict aMDT for February 2016, etc.). Use the values of p, q, d, P, Q, D, and S as determined to be best in part b, but update your coefficients at every time step using the new data.

Create a plot of the one-year-in-advance predictions and 95% confidence bands superimposed on a time series plot of the observed aMDT values from January 2010 to December 2020.

ANSWER: Plot below

```
### Set vars
train_month = as.Date("2015-01-01")
fn = "MeanDallasTemps.csv"
df2 = read.csv(fn)
last_month_value = tail(df2,n=1)$Month

### Cast all dates in column to DATE type
df2["Month"] = as.Date(df2$Month)

### Add in an empty column for inserting/appending forecasting
df2 = cbind(df2, ForecastedAvgTemps=NA)
df2 = cbind(df2, ForecastedStDev=NA)

### Forecast
month_range = which(df2$Month == as.Date("2020-12-01")) - which(df2$Month == train_month)

for (i in 0:month_range) {
  ### Create a moving window of subsetted data for forecasting. INCLUDES the month of interest (i.e.
jan 2015)
  current_n = which(df2$Month == train_month) + i
  forecast_subset = window(df2$AvgTemp, start = 1, end = current_n)

  fit_for = sarima.for(forecast_subset,
                       n.ahead = 12,
                       p = AICc_min_df$p,
                       d = AICc_min_df$d,
                       q = AICc_min_df$q,
                       P=AICc_min_df$P,
                       D=AICc_min_df$D,
                       Q=AICc_min_df$Q,
                       S=4,
                       plot = FALSE,
                       no.constant = TRUE)

  ### Grab the last value of the forecast for plotting and insert into our dataframe
  df2[current_n+12,]$ForecastedAvgTemps = tail(fit_for$pred, n=1)
  df2[current_n+12,]$ForecastedStDev = sqrt(tail(fit_for$se, n=1))

  if (current_n + 12 >= 252) {
    break
  }

}

### Combine the data into a single dataframe with bind rows. Combine for plotting
combined_fitted_df <- bind_rows(
  data.frame(Time = df2$Month, Type = factor(rep("Observed", length(df2$Month)), levels = c("Observe
d", "Forecasted")), x = as.numeric(df2$AvgTemp)), ### Observed data
  data.frame(Time = df2$Month, Type = factor(rep("Forecasted", length(df2$Month)), levels = c("Observ
ed","Forecasted")), x = as.numeric(df2$ForecastedAvgTemps)), ### Forecasted data
  )

### Plot predicted, forecasted, and CI bands
fitted_ggplot <- ggplot(combined_fitted_df, aes(x = Time)) +
```

```
                  geom_line(aes(y = x, col = Type)) +
                  ## Per HW prompt, use 1.96 stdevs ON EACH SIDE of the mean for 95% CI bands
                  geom_ribbon(data = df2,
                              aes(x = Month,
                                  ymin = ForecastedAvgTemps - 1.96*ForecastedStDev, ### Per HW promp
t, use 1.96 stdevs ON EACH SIDE of the mean for 95% CI bands
                                  ymax = ForecastedAvgTemps + 1.96*ForecastedStDev),
                              alpha = .175) +
                  geom_vline(xintercept = as.Date("2016-01-01")) +
                  ggtitle("Forecasted and Observed Values with CI 95% Band") +
                  xlab("Time (Month)") +
                  ylab("Temperature (F)") +
                  xlim(as.Date("2010-01-01"), as.Date("2020-12-01"))


### Display plot
fitted_ggplot
```




```
FcstTemp2018 = df2[which(df2$Month == as.Date("2018-01-01")), ]$ForecastedAvgTemps
stdev2018 = df2[which(df2$Month == as.Date("2018-01-01")), ]$ForecastedStDev
upper_bound = FcstTemp2018 + stdev2018*1.96
lower_bound = FcstTemp2018 - stdev2018*1.96
```

Report the one-year-in-advance prediction of aMDT for January 2018, along with the upper and lower bounds of the prediction interval. (Hint: Making one-year-in-advance predictions with newly added data at each time step may require a for loop).

ANSWER: The predicted value for January 2018 is: 71.29 with 95% CI upper bound: 82 and 95% CI lower bound: 60.58

Question 3d.

d. Now consider an alternative model for the the aMDT data that does not have a seasonal component. Report the AICc value for an ARIMA(3,1,1) model fit to the full aMDT data set.

```
### Set vars
df3 = read.csv(fn)

arma311 = sarima(df3$AvgTemp, p = 3, d = 1, q = 1, no.constant=TRUE, details=FALSE)
```

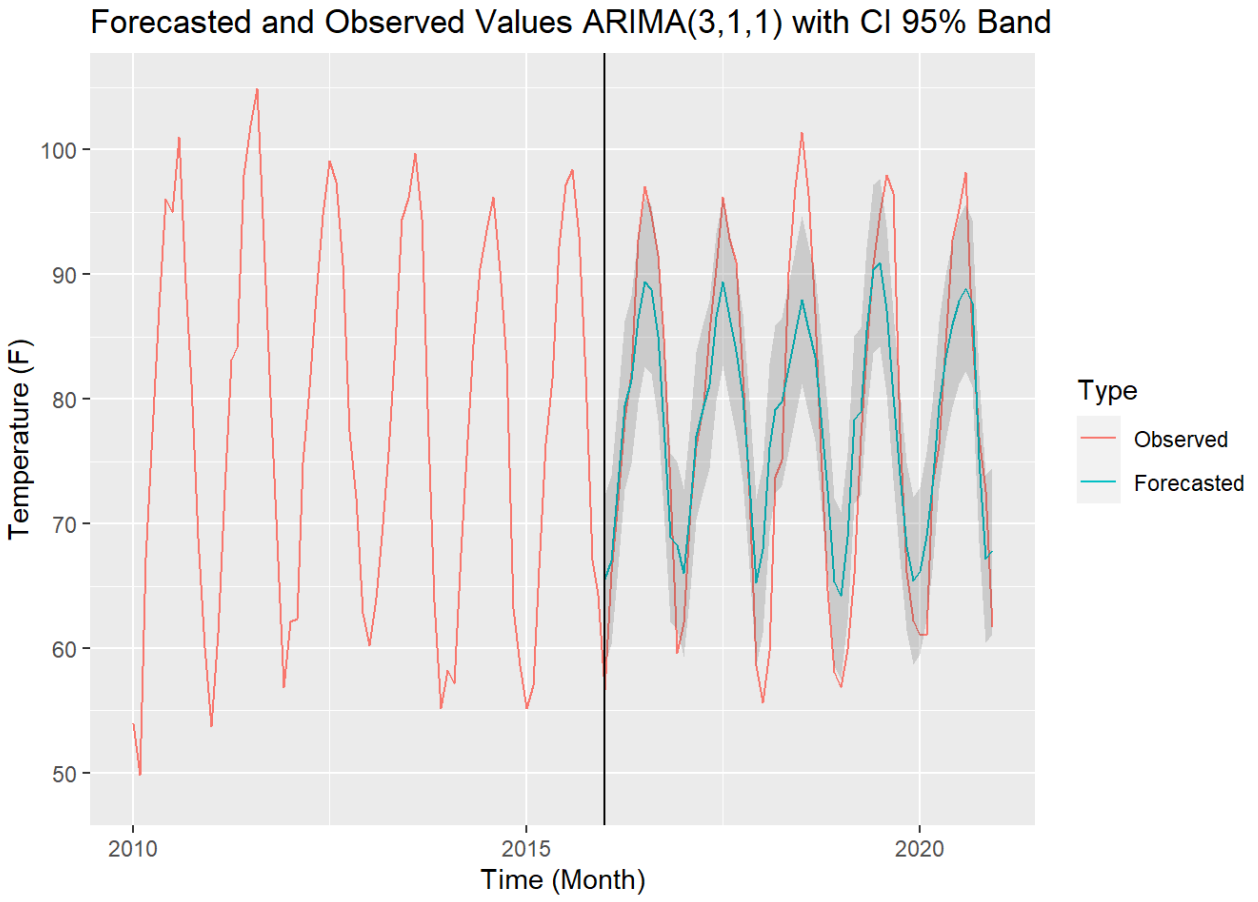
ANSWER: ARIMA(3,1,1) AICc = 5.9960597

Refit the model to make one-year-in-advance predictions of aMDT for the last five years of the observation window (2016-2020) as you did in the previous subquestion. Plot your predictions and 95% confidence bounds, along with the true observed values shown. Set your x-axis to span January 2010 to December 2020.

ANSWER: ARIMA(3,1,1) Plot Below

```
### Set vars
train_month = as.Date("2015-01-01")
last_month_value = tail(df3,n=1)$Month

### Cast all dates in column to DATE type
df3["Month"] = as.Date(df3$Month)

### Add in an empty column for inserting/appending forecasting
df3 = cbind(df3, ForecastedAvgTemps=NA)
df3 = cbind(df3, ForecastedStDev=NA)

### Forecast
month_range = which(df3$Month == as.Date("2020-12-01")) - which(df3$Month == train_month)

for (i in 0:month_range) {
  ### Create a moving window of subsetted data for forecasting. INCLUDES the month of interest (i.e.
jan 2015)
  current_n = which(df3$Month == train_month) + i
  forecast_subset = window(df3$AvgTemp, start = 1, end = current_n)

  fit_for = sarima.for(forecast_subset,
                       n.ahead = 12,
                       p = 3,
                       d = 1,
                       q = 1,
                       plot = FALSE,
                       no.constant = TRUE)

  ### Grab the last value of the forecast for plotting and insert into our dataframe
  df3[current_n+12,]$ForecastedAvgTemps = tail(fit_for$pred, n=1)
  df3[current_n+12,]$ForecastedStDev = sqrt(tail(fit_for$se, n=1))

  if (current_n + 12 >= 252) {
    break
  }

}

### Combine the data into a single dataframe with bind rows. Combine for plotting
combined_fitted_df <- bind_rows(
  data.frame(Time = df3$Month, Type = factor(rep("Observed", length(df3$Month)), levels = c("Observe
d", "Forecasted")), x = as.numeric(df3$AvgTemp)), ### Observed data
  data.frame(Time = df3$Month, Type = factor(rep("Forecasted", length(df3$Month)), levels = c("Observ
ed","Forecasted")), x = as.numeric(df3$ForecastedAvgTemps)), ### Forecasted data
  )

### Plot predicted, forecasted, and CI bands
fitted_ggplot <- ggplot(combined_fitted_df, aes(x = Time)) +
                  geom_line(aes(y = x, col = Type)) +
                  ## Per HW prompt, use 1.96 stdevs ON EACH SIDE of the mean for 95% CI bands
                  geom_ribbon(data = df3,
                              aes(x = Month,
                                  ymin = ForecastedAvgTemps - 1.96*ForecastedStDev, ### Per HW promp
t, use 1.96 stdevs ON EACH SIDE of the mean for 95% CI bands
```

```
                                ymax = ForecastedAvgTemps + 1.96*ForecastedStDev),
                            alpha = .175) +
                geom_vline(xintercept = as.Date("2016-01-01")) +
                ggtitle("Forecasted and Observed Values ARIMA(3,1,1) with CI 95% Band") +
                xlab("Time (Month)") +
                ylab("Temperature (F)") +
                xlim(as.Date("2010-01-01"), as.Date("2020-12-01"))


### Display plot
fitted_ggplot
```




```
FcstTemp2018 = df3[which(df3$Month == as.Date("2018-01-01")), ]$ForecastedAvgTemps
stdev2018 = df3[which(df3$Month == as.Date("2018-01-01")), ]$ForecastedStDev
upper_bound = FcstTemp2018 + stdev2018*1.96
lower_bound = FcstTemp2018 - stdev2018*1.96
```

Additionally, report your one-year-in-advance prediction for the aMDT for January 2018, along with your upper and lower bounds of your prediction interval.

ANSWER: The predicted value for January 2018 is: 67.93 with 95% CI upper bound: 74.64 and 95% CI lower bound: 61.22

Does the fitted model produce predictions that capture seasonal behavior? How do the predictions from the ARIMA(3,1,1) model that does not include a specific seasonal component compare to the predictions from the model fitted in part c?

ANSWER: While the fitted model does produce results, they aren't not as accurate (upon visual inspection and looking at the AICc value, which is slightly higher for the non-seasonal model) as the model that used the seasonal components. Though both models follow the period/frequency of the seasonal trend relatively well, it's easy to see that the predictions generated by the seasonal model more accurately captures the peaks of the cyclical periods. The amplitudes more closely match and the CI's are a bit wider, expanding to include most of the observed temperature peaks. Overall, the seasonal model (as expected) performs better! Though it's of note that the CI bands appear to be relatively the same.

Question 3e.

e. Report the AICc value for an ARIMA(12,1,0) model fit to the full aMDT data set.

```
df4 = read.csv(fn)

arma1210 = sarima(df4$AvgTemp, p = 12, d = 1, q = 0, no.constant=TRUE, details=FALSE)
```

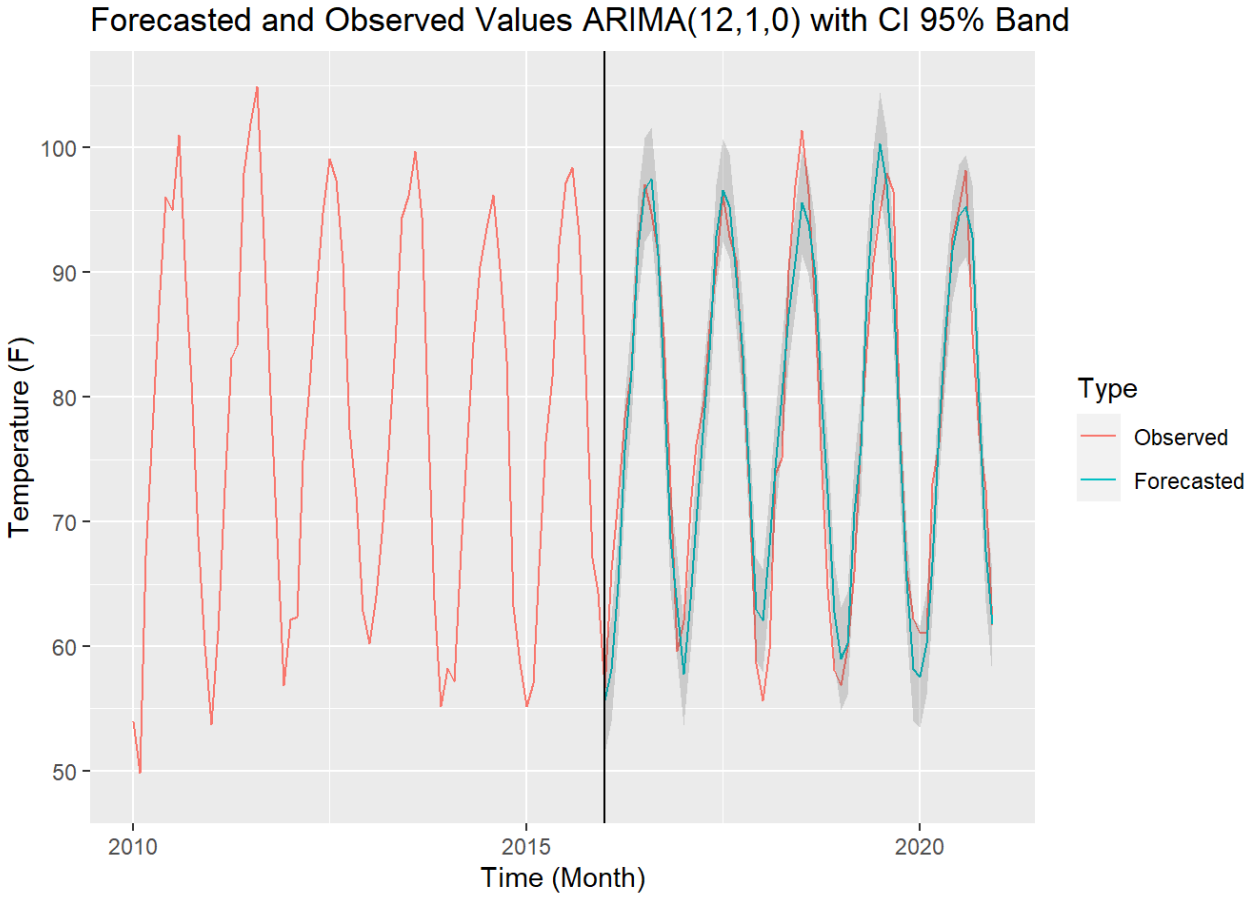
ANSWER: ARIMA(12,1,0) AICc = 5.5322948

Refit the model to make one-year-in-advance predictions of aMDT for the last five years of the observation window (2016-2020) as you did in the previous subquestion. Plot your predictions and 95% confidence bounds, along with the true observed values shown in. Set your x-axis to span January 2010 to December 2020.

ANSWER: ARIMA(12,1,0) Plot Below

```
### Set vars
train_month = as.Date("2015-01-01")
last_month_value = tail(df4,n=1)$Month

### Cast all dates in column to DATE type
df4["Month"] = as.Date(df4$Month)

### Add in an empty column for inserting/appending forecasting
df4 = cbind(df4, ForecastedAvgTemps=NA)
df4 = cbind(df4, ForecastedStDev=NA)

### Forecast
month_range = which(df4$Month == as.Date("2020-12-01")) - which(df4$Month == train_month)

for (i in 0:month_range) {
  ### Create a moving window of subsetted data for forecasting. INCLUDES the month of interest (i.e.
jan 2015)
  current_n = which(df4$Month == train_month) + i
  forecast_subset = window(df4$AvgTemp, start = 1, end = current_n)

  fit_for = sarima.for(forecast_subset,
                       n.ahead = 12,
                       p = 12,
                       d = 1,
                       q = 0,
                       plot = FALSE,
                       no.constant = TRUE)

  ### Grab the last value of the forecast for plotting and insert into our dataframe
  df4[current_n+12,]$ForecastedAvgTemps = tail(fit_for$pred, n=1)
  df4[current_n+12,]$ForecastedStDev = sqrt(tail(fit_for$se, n=1))

  if (current_n + 12 >= 252) {
    break
  }

}

### Combine the data into a single dataframe with bind rows. Combine for plotting
combined_fitted_df <- bind_rows(
  data.frame(Time = df4$Month, Type = factor(rep("Observed", length(df4$Month)), levels = c("Observe
d", "Forecasted")), x = as.numeric(df4$AvgTemp)), ### Observed data
  data.frame(Time = df4$Month, Type = factor(rep("Forecasted", length(df4$Month)), levels = c("Observ
ed","Forecasted")), x = as.numeric(df4$ForecastedAvgTemps)), ### Forecasted data
  )

### Plot predicted, forecasted, and CI bands
fitted_ggplot <- ggplot(combined_fitted_df, aes(x = Time)) +
                  geom_line(aes(y = x, col = Type)) +
                  ## Per HW prompt, use 1.96 stdevs ON EACH SIDE of the mean for 95% CI bands
                  geom_ribbon(data = df4,
                              aes(x = Month,
                                  ymin = ForecastedAvgTemps - 1.96*ForecastedStDev, ### Per HW promp
t, use 1.96 stdevs ON EACH SIDE of the mean for 95% CI bands
```

```
                            ymax = ForecastedAvgTemps + 1.96*ForecastedStDev),
                        alpha = .175) +
    geom_vline(xintercept = as.Date("2016-01-01")) +
    ggtitle("Forecasted and Observed Values ARIMA(12,1,0) with CI 95% Band") +
    xlab("Time (Month)") +
    ylab("Temperature (F)") +
    xlim(as.Date("2010-01-01"), as.Date("2020-12-01"))


### Display plot
fitted_ggplot
```




```
FcstTemp2018 = df4[which(df4$Month == as.Date("2018-01-01")), ]$ForecastedAvgTemps
stdev2018 = df4[which(df4$Month == as.Date("2018-01-01")), ]$ForecastedStDev
upper_bound = FcstTemp2018 + stdev2018*1.96
lower_bound = FcstTemp2018 - stdev2018*1.96
```

Additionally, report your one-year-in-advance prediction for the aMDT for January 2018, along with your upper and lower bounds of your prediction interval.

ANSWER: The predicted value for January 2018 is: 62.06 with 95% CI upper bound: 66.18 and 95% CI lower bound: 57.95

Does the fitted model produce predictions that capture seasonal behavior? How do the predictions from the ARIMA(12,1,0) model compare to the predictions from the models fitted in parts c and d?

ANSWER: The fitted model *does* produce predictions that seem to capture seasonal behavior quite well! In fact, this model seems to fit the observed data much better than parts c) and d). This could be due to the fact that the order of the AR component is very high (12) which could potentially be overfitting the model to the data. It's possible that as the order of the terms in the SARIMA model increase it tends to lead to overfitting (or perhaps this dataset is just better fit with higher order functions). Either way, this model most closely matches the amplitudes and peaks of the observed data and has a matching frequency/period. It also has the lowest AICc model, making this the best (assuming no overfitting) of the three models (including tighter CI bounds, indicating increased confidence and lower variation with this model).

Citations

[1]Shumway, R.H. and Stoffer, D.S. (2017). Time Series Analysis and Its Applications with R Examples (4th Ed.), Springer.