DSC383 HW2

2024-06-16

Question 3

```
# Read in data as time series
mean_temps <- read.csv("MeanDallasTemps.csv")
mean_temps_ts <- as.ts(mean_temps$AvgTemp)
```

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
In a seasonal time series, S represents the number of time periods until a seasonal pattern repeats. Thus for a monthly variation in temperature, $S = 12$ would be appropriate, since we would expect the a temperature pattern to repeat itself after a year.

b.

Using the seasonal lag selection in the previous subquestion, fit the $SARIMA(p, d, q) \times (P, D, Q)$ model to the full aMDT time series for all combinations of p, d, q, P, D, Q in $0, 1$ except the four cases where $P = 1, D = 1, Q = 1$ 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 δ) 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.

```
aicc_table <- data.frame(
  "p" = integer(),
  "d" = integer(),
  "q" = integer(),
  "P" = integer(),
  "D" = integer(),
  "Q" = integer(),
  "AICc" = double()
)

# Loop through all combinations
orders <- c(0, 1)
for (p in orders) {
  for (d in orders) {
    for (q in orders) {
```

```
      for (P in orders) {
        for (D in orders) {
          for (Q in orders) {
            if (P!=1 | D!=0 | Q!=1 | d!=0) {
              sarima_fit <- sarima(mean_temps_ts,
                                   p=p, d=d, q=q, # Nonseasonal orders
                                   P=P, D=D, Q=Q, S=12, # Seasonal orders
                                   no.constant=T, details=F)  # No constant since delta=0
              fit_aicc <- sarima_fit$ICs[2]
              aicc_table[nrow(aicc_table)+1,] <- c(p, d, q, P, D, Q, fit_aicc)
            }
          }
        }
      }
    }
  }
}
```

Showing only the results of the 10 fits with the 10 smallest AICc values

```
# First check the length of the full table, should be 60
nrow(aicc_table)
## [1] 60
# Sort by ascending AICc
smallest_aicc_table <- arrange(aicc_table, AICc)
head(smallest_aicc_table, 10)
##    p d q P D Q     AICc
## 1  1 0 1 0 1 1 5.342478
## 2  1 0 1 1 1 1 5.351037
## 3  1 0 0 0 1 1 5.352432
## 4  1 0 0 1 1 1 5.360922
## 5  0 0 1 0 1 1 5.377597
## 6  0 0 1 1 1 1 5.386041
## 7  1 1 1 0 1 1 5.386422
## 8  1 1 1 1 1 1 5.395050
## 9  0 1 1 0 1 1 5.411794
## 10 0 1 1 1 1 1 5.419146
```

After sorting from least to greatest $AICc$, the model $SARIMA(1,0,1) \times (0,1,1)_{12}$ has the smallest $AICc$, thus best fits the data.

We can look at this model alone to identify the estimated values of the unknown parameters

```
sarima(mean_temps_ts,
       p=1, d=0, q=1,
       P=0, D=1, Q=1, S=12,
       no.constant=T, details=F)
## <><><><><><><><><><><><><><><>
##
## Coefficients:
##      Estimate     SE  t.value p.value
## ar1    0.7183 0.1351   5.3159  0.0000
## ma1   -0.4535 0.1764  -2.5709  0.0108
## sma1  -1.0000 0.0589 -16.9895  0.0000
```

```
##
## sigma^2 estimated as 10.15628 on 237 degrees of freedom
##
## AIC = 5.342055  AICc = 5.342478  BIC = 5.400065
##
```

For $SARIMA(1,0,1) \times (0,1,1)_{12}$:

$AICc = 5.342$

Estimated parameters
$AR = 0.718$
$MA = -0.454$
Seasonal MA $= -1.00$

Setting variables/functions for parts C, D, E

```
# Month column must be formatted as date
mean_temps$Month <- as.Date(mean_temps$Month)

# Stack observed data and predictions in a single data frame
type_lvls <- c("Observed", "Predicted")

# Observed data (all months from 2010-2020)
observed_plot_data <- subset(mean_temps, Month >= as.Date("2010-01-01"))

# Predictions (only last five years 2016-2020)
predicted_plot_data <- subset(mean_temps, Month >= as.Date("2016-01-01"))
```

Functions to be shared between parts c, d, and e

```
# Predictions will be appended to this df
initiate_combined_data <- function(){
  return(data.frame(Month = as.Date(observed_plot_data$Month),
                    AvgTemp = as.numeric(observed_plot_data$AvgTemp),
                    Type = factor("Observed", levels = type_lvls)))
}

# SE df to graph CI, only on predicted rows
initiate_se_data <- function() {
  return(data.frame(Month = as.Date(double()),
                    AvgTemp = double(),
                    SE = double()))
}

# Set the training window time series subtracting a year from the new month
get_train_set <- function(new_month) {
  # train_month is 1 year in advance
  train_month <- as.Date(new_month) - years(1)
  train_df <- subset(mean_temps, Month <= train_month)
  return(as.ts(train_df$AvgTemp))
}
```

```r
# Function to predict new month with model
fit_test_set <- function(train_set,
                         p=0, d=0, q=0,
                         P=0, D=0, Q=0, S=0) {
    return(sarima.for(train_set,
                      n.ahead = 12, # Predict 12 months in advance
                      p, d, q,
                      P, D, Q, S=S,
                      no.constant=T,
                      plot = F))
}

# Main function, obtains predicted values and SE for each month
run_model_predictions <- function(p=0, d=0, q=0,
                                  P=0, D=0, Q=0, S=0) {
  # Create data frames to fill
  combined_data <- initiate_combined_data()
  se_data <- initiate_se_data()

  # Iterate through months starting in 2016
  for (predict_month in predicted_plot_data$Month) {
    predict_month <- as.Date(predict_month)
    train_set <- get_train_set(predict_month)
    # Predicted values for the next 12 months
    predicted <- fit_test_set(train_set, p, d, q, P, D, Q, S)

    # Append to combined_data df
    new_month_temp <- tail(predicted$pred, n=1)  # Only grab last month of predicted data
    predicted_row <- data.frame(Month = predict_month,
                                AvgTemp = as.numeric(new_month_temp),
                                Type = factor("Predicted", levels = type_lvls))
    combined_data <- bind_rows(combined_data, predicted_row)

    # Append to se_data df
    new_month_se <- tail(predicted$se, n=1)  # Only grab last month of predicted data
    se_row <- data.frame(Month = predict_month,
                         AvgTemp = as.numeric(new_month_temp),
                         SE = as.numeric(new_month_se))
    se_data <- bind_rows(se_data, se_row)
  }
  return(list(combined_data=combined_data, se_data=se_data))
}


# Create plot with observed and predicted data
plot_model_predictions <- function(combined_data, se_data, title) {
  ggplot(combined_data,
         aes(x = Month)) +
  geom_line(aes(y = AvgTemp,
                col = Type),
            linewidth = 1) +
  geom_ribbon(data = se_data,
              aes(x = Month,
```

```
                  # 95% confidence
                  ymin = AvgTemp - 1.96*SE,
                  ymax = AvgTemp + 1.96*SE),
                  alpha = .2) +
  geom_vline(xintercept = as.Date("2016-01-01"),
             linewidth = 0.7) +
  scale_color_manual(values=c("deepskyblue3", "coral2")) +
  scale_x_date(breaks = seq(as.Date("2010-01-01"), as.Date("2020-12-01"),
               by = "2 years"),
               date_labels = "%Y") +
  labs(title = title,
       x = "Time",
       y = "Average Temperature")

}

# Get prediction value for specific month
pred_val_jan2018 <- function(combined_data) {
  return(combined_data$AvgTemp[combined_data$Month == "2018-01-01" &
                                 combined_data$Type == "Predicted"])
}

# Get prediction interval for specific month
pred_int_jan2018 <- function(combined_data, se_data) {
  pred_jan2018 <- pred_val_jan2018(combined_data)
  se_jan2018 <- se_data$SE[se_data$Month == "2018-01-01"]

  lower_bound <- pred_jan2018 - 1.96*se_jan2018
  upper_bound <- pred_jan2018 + 1.96*se_jan2018

  return(c(lower_bound, upper_bound))
}
```

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, d, q, P, D, Q 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. 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

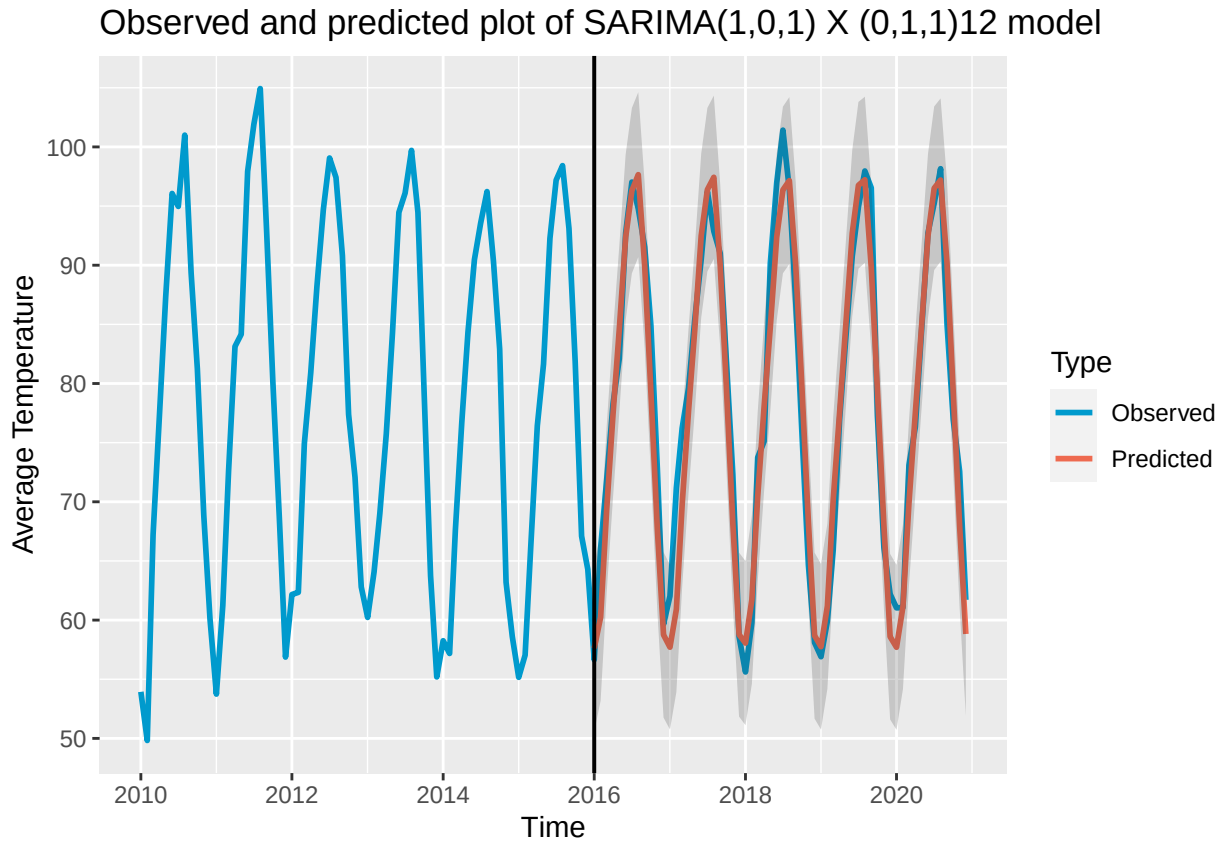
Using $SARIMA(1,0,1) \times (0,1,1)_{12}$ from part b,

```
# Create observed/predicted AND SE data frames
sarima_data <- run_model_predictions(p=1, d=0, q=1, P=0, D=1, Q=1, S=12)

# Plot data and predictions
```

```
sarima_plot <- plot_model_predictions(
  sarima_data$combined_data,
  sarima_data$se_data,
  "Observed and predicted plot of SARIMA(1,0,1) X (0,1,1)12 model"
)

sarima_plot
```



One-year-in-advance prediction and CI of aMDT for January 2018

```
# Prediction
pred_val_jan2018(sarima_data$combined_data)
## [1] 58.05647
# Interval
pred_int_jan2018(sarima_data$combined_data, sarima_data$se_data)
## [1] 51.11755 64.99539
```

For January 2018:
Predicted temperature = 58.056
Prediction interval = (51.118, 64.995)

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. 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. 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. 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

Using $ARIMA(3,1,1)$,

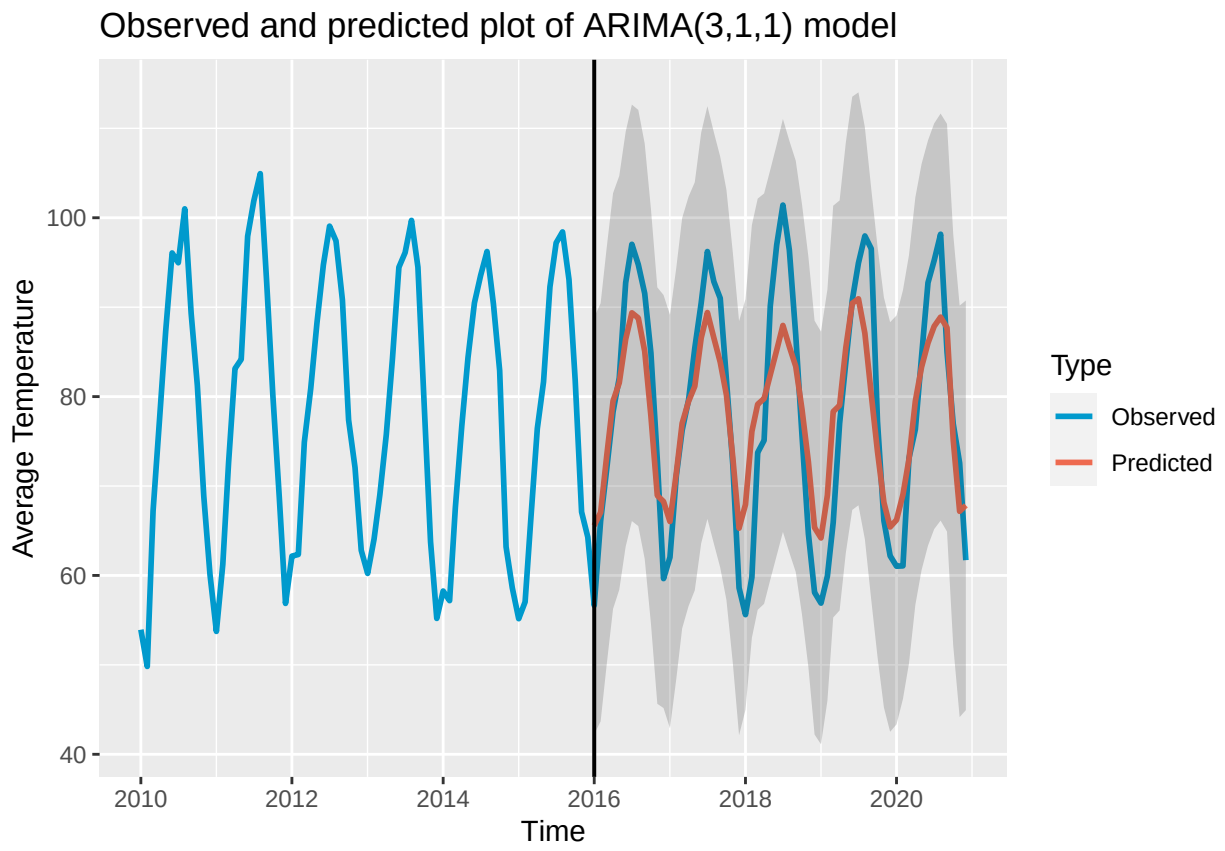
```
sarima(mean_temps_ts,
       p=3, d=1, q=1,
       no.constant=T, details=F)
## <><><><><><><><><><><><><><><>
##
## Coefficients:
##     Estimate     SE  t.value p.value
## ar1   1.0218 0.0525  19.4689   0.000
## ar2   0.0921 0.0835   1.1032   0.271
## ar3  -0.5602 0.0524 -10.6870   0.000
## ma1  -1.0000 0.0168 -59.6251   0.000
##
## sigma^2 estimated as 21.92383 on 247 degrees of freedom
##
## AIC = 5.995412  AICc = 5.99606  BIC = 6.06564
##
```

For $ARIMA(3,1,1)$:
$AICc = 5.995$

Now following the same steps as part c,

```
# Create observed/predicted AND SE data frames
arima1_data <- run_model_predictions(p=3, d=1, q=1)

# Plot data and predictions
arima1_plot <- plot_model_predictions(
  arima1_data$combined_data,
  arima1_data$se_data,
  "Observed and predicted plot of ARIMA(3,1,1) model"
)
arima1_plot
```



One-year-in-advance prediction and CI of aMDT for January 2018

```
# Prediction
pred_val_jan2018(arima1_data$combined_data)
## [1] 67.92878
# Interval
pred_int_jan2018(arima1_data$combined_data, arima1_data$se_data)
## [1] 44.96445 90.89312
```

For January 2018:
Predicted temperature = 67.929
Prediction interval = (44.964, 90.893)

The fitted model predictions $ARIMA(3,1,1)$ were able to detect the general behavior for seasonal increases and decreases in temperature, though not as well as $SARIMA(1,0,1) \times (0,1,1)_{12}$ in part c. The predicted temperature plot is not as accurate here, and the confidence interval is much wider with this model, indicating a less reliable model. This conclusion could also be told from the higher AICc value for $ARIMA(3,1,1)$.

e.

Now we work on the ARIMA model with a different set of parameters. Report the AICc value for an $ARIMA(12,1,0)$ model fit to the full aMDT data set. 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. 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. 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

Using $ARIMA(12,1,0)$,

```
sarima(mean_temps_ts,
       p=12, d=1, q=0,
       no.constant=T, details=F)
## <><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ar1    -0.5011 0.0635 -7.8901  0.0000
## ar2    -0.4233 0.0713 -5.9334  0.0000
## ar3    -0.5195 0.0692 -7.5098  0.0000
## ar4    -0.5261 0.0686 -7.6700  0.0000
## ar5    -0.5302 0.0675 -7.8597  0.0000
## ar6    -0.5895 0.0674 -8.7493  0.0000
## ar7    -0.5003 0.0664 -7.5318  0.0000
## ar8    -0.5727 0.0654 -8.7545  0.0000
## ar9    -0.5475 0.0683 -8.0201  0.0000
## ar10   -0.5335 0.0702 -7.5947  0.0000
## ar11   -0.1760 0.0741 -2.3743  0.0184
## ar12    0.0163 0.0663  0.2464  0.8056
##
## sigma^2 estimated as 12.79188 on 239 degrees of freedom
##
## AIC = 5.527072  AICc = 5.532295  BIC = 5.709665
##
```

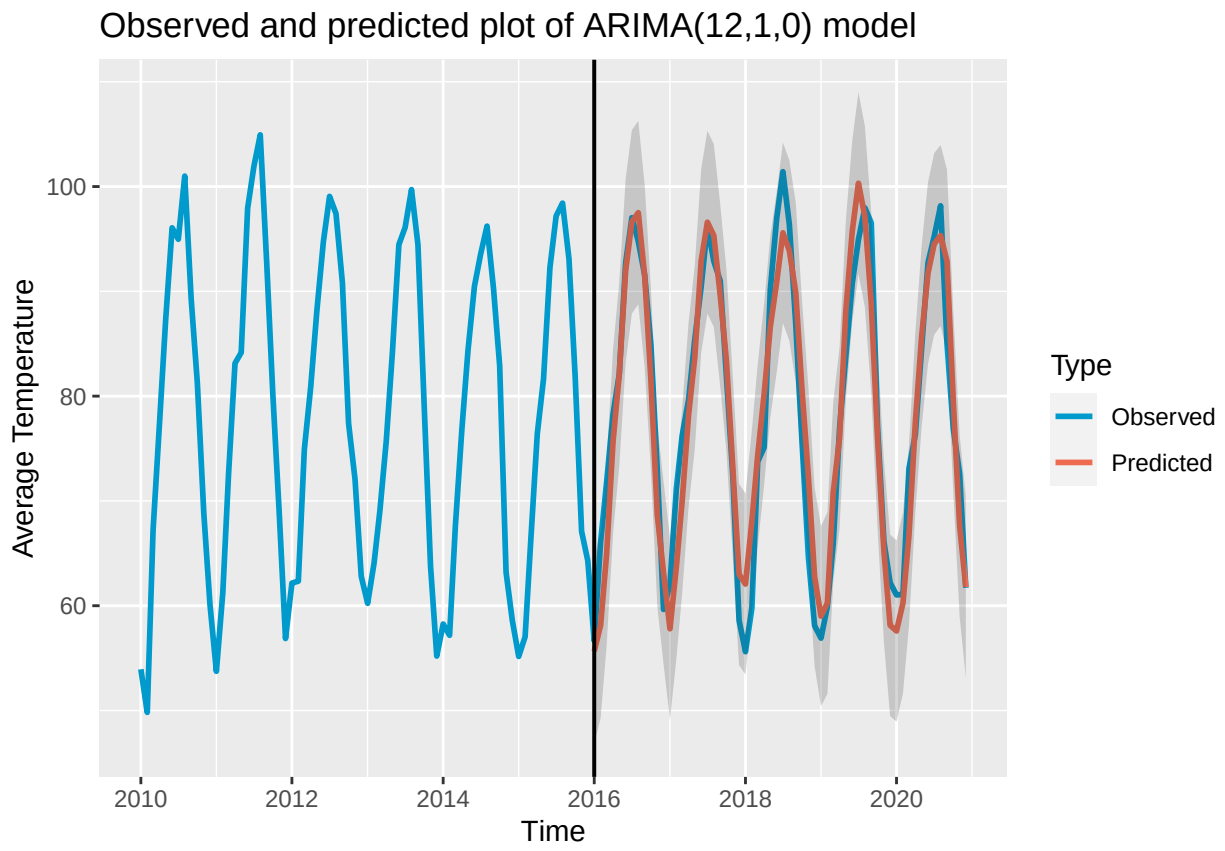
For $ARIMA(12,1,0)$:

$AICc = 5.527$

Now following the same steps as part c and d,

```
# Create observed/predicted AND SE data frames
arima2_data <- run_model_predictions(p=12, d=1, q=0)

# Plot data and predictions
arima2_plot <- plot_model_predictions(
  arima2_data$combined_data,
  arima2_data$se_data,
  "Observed and predicted plot of ARIMA(12,1,0) model"
)
arima2_plot
```



One-year-in-advance prediction and CI of aMDT for January 2018

```
# Prediction
pred_val_jan2018(arima2_data$combined_data)
## [1] 62.06421
# Interval
pred_int_jan2018(arima2_data$combined_data, arima2_data$se_data)
## [1] 53.42499 70.70343
```

For January 2018:
Predicted temperature = 62.064
Prediction interval = (53.425, 70.703)

The results for $ARIMA(12,1,0)$ are much more comparable to $SARIMA(1,0,1)\times(0,1,1)_{12}$ than the model in part d. This model is reliable in detecting seasonal behavior, which we can observe from the prediction plot following more closely to the observed data, as well as a tight confidence interval and relatively low AICc. While this model fits the better than $ARIMA(3,1,1)$, it is not quite as accurate as our initial model that included seasonality, $SARIMA(1,0,1)\times(0,1,1)_{12}$.