HW3

Part 3

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?

We could consider that the temperature changes each season of the year, i.e., spring, summer, autumn and winter. So, each three months the temperature will change. Hence, given that we have the averaged monthly maximum daily temperature, we have that

$S = 3$ (months)

b. Using the seasonal lag selection in the previous subquestion, fit the SARIMA$(p,d,q)\times(P,D,Q)_s$ 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 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.

The best fitted model is

$$SARIMA(1,1,0)\times(1,0,1)_3$$

with

$$AIC_c = 6.669289197$$

$\phi_1 = 0.5152$

$\underline{\Phi}_1 = 0.4026$

$\Theta = -0.6361$

$\sigma_w^2 = 44.57$

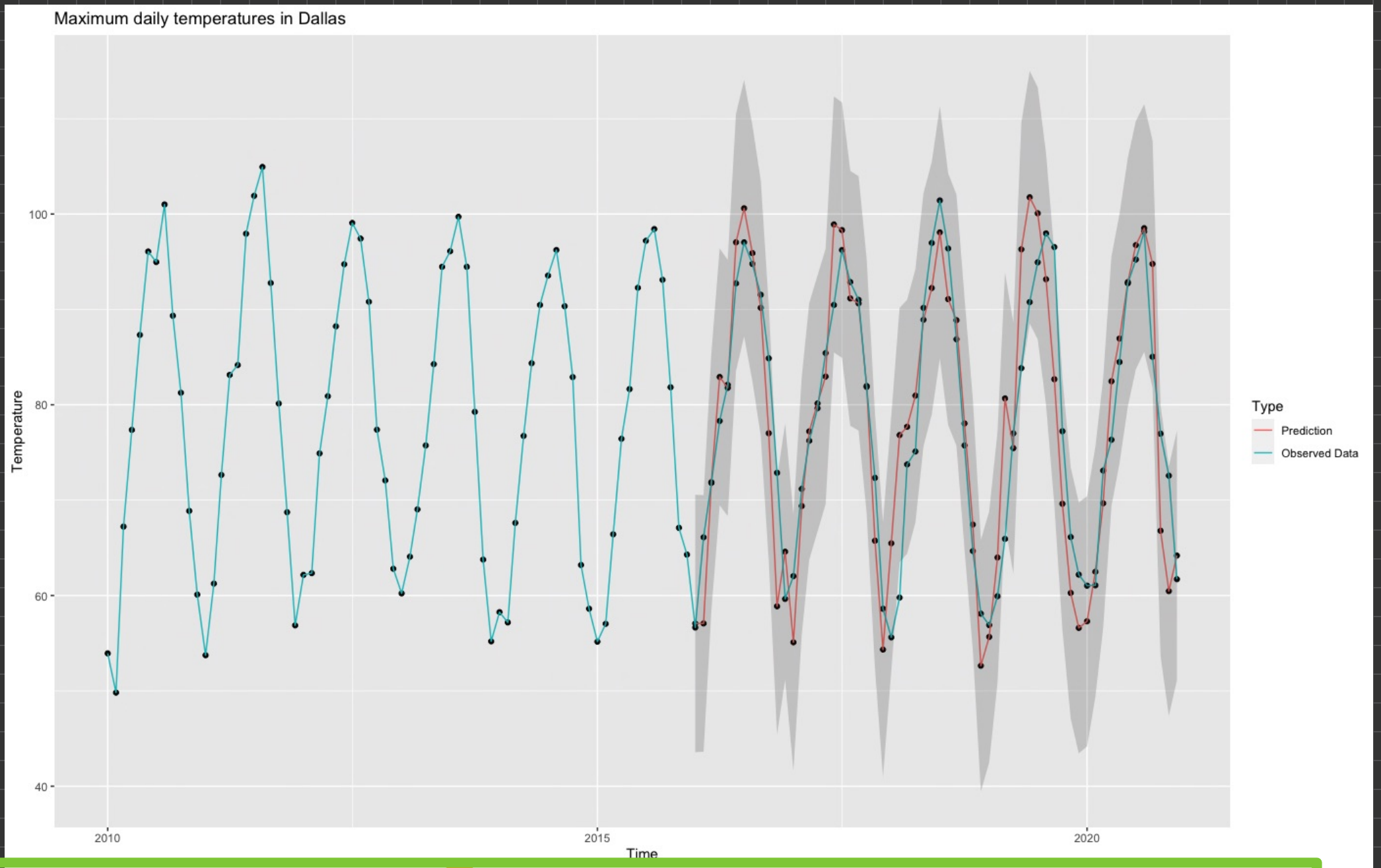
Values of the unKnown parameters

CODE FOR FITTING ALL MODELS AND OBTAIN THE BEST ONE

```r
data1 <- read.csv("/Users/rafa/Documents/Master Austin/MAESTRÍA_AUSTIN/Advanced Predictive Models/HW2/file3.xls", header=TRUE, stringsAsFactors=FALSE)
avg_temps<-data1$AvgTemp
#Fitting to data all combinations except  for P=1,D=0,Q=1 and d=0
count<-0
lowest<-100
for (p_ar in 0:1)
{
  for (q_ar in 0:1)
  {
    for (d_ar in 0:1)
    {
      for (P__ar in 0:1 )
      {
        for (Q__ar in 0:1 )
        {
          for (D__ar in 0:1 )
          {
            if (d_ar!=0 | P__ar !=1 | D__ar!=0 | Q__ar != 1)
              {
              # Fit model
              fit <- sarima(avg_temps,
                            p = p_ar,
                            d = d_ar,
                            q = q_ar,
                            P=P__ar,
                            D=D__ar,
                            Q=Q__ar,
                            S=3,
                            no.constant = TRUE,
                            details = FALSE)

            if (fit$AICc < lowest)
            {
              p_low<-p_ar
              d_low<-d_ar
              q_low<-q_ar
              P_low<-P__ar
              D_low<-D__ar
              Q_low<-Q__ar
              lowest<-fit$AICc
            }
            # Examine estimated model parameters
            print(paste("SARIMA(p=",p_ar,"d=",d_ar, "q=", q_ar,  ") x (P=",P__ar,",D=",D__ar,"Q=",Q__ar,") with AICc =,", fit$AICc))
          }
        }
      }

    }

    }
  }
}
```

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 $p, d, q, P, D,$ and 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)

Result ↓



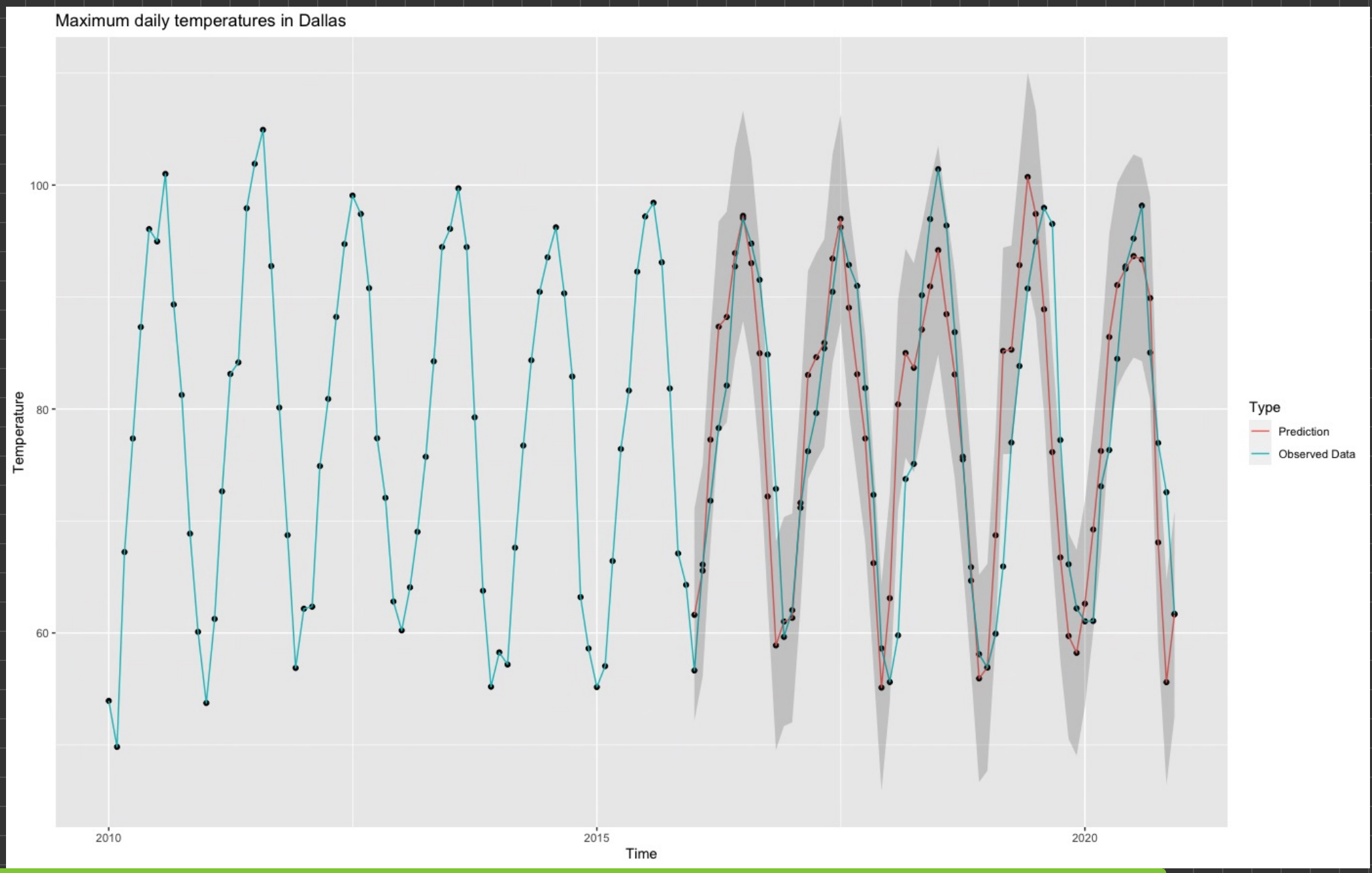
Prediction of AMDT For January 2018 → 65.47467

with Lower Bound = 52.12594

Upper Bound = 78.82339

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?

The AICc for ARIMA(3,1,1) is 5.99606 Result Plot ↓



Prediction of AMDT For January 2018 → 63.10129

with Lower Bound = 53.81456

Upper Bound = 72.38802

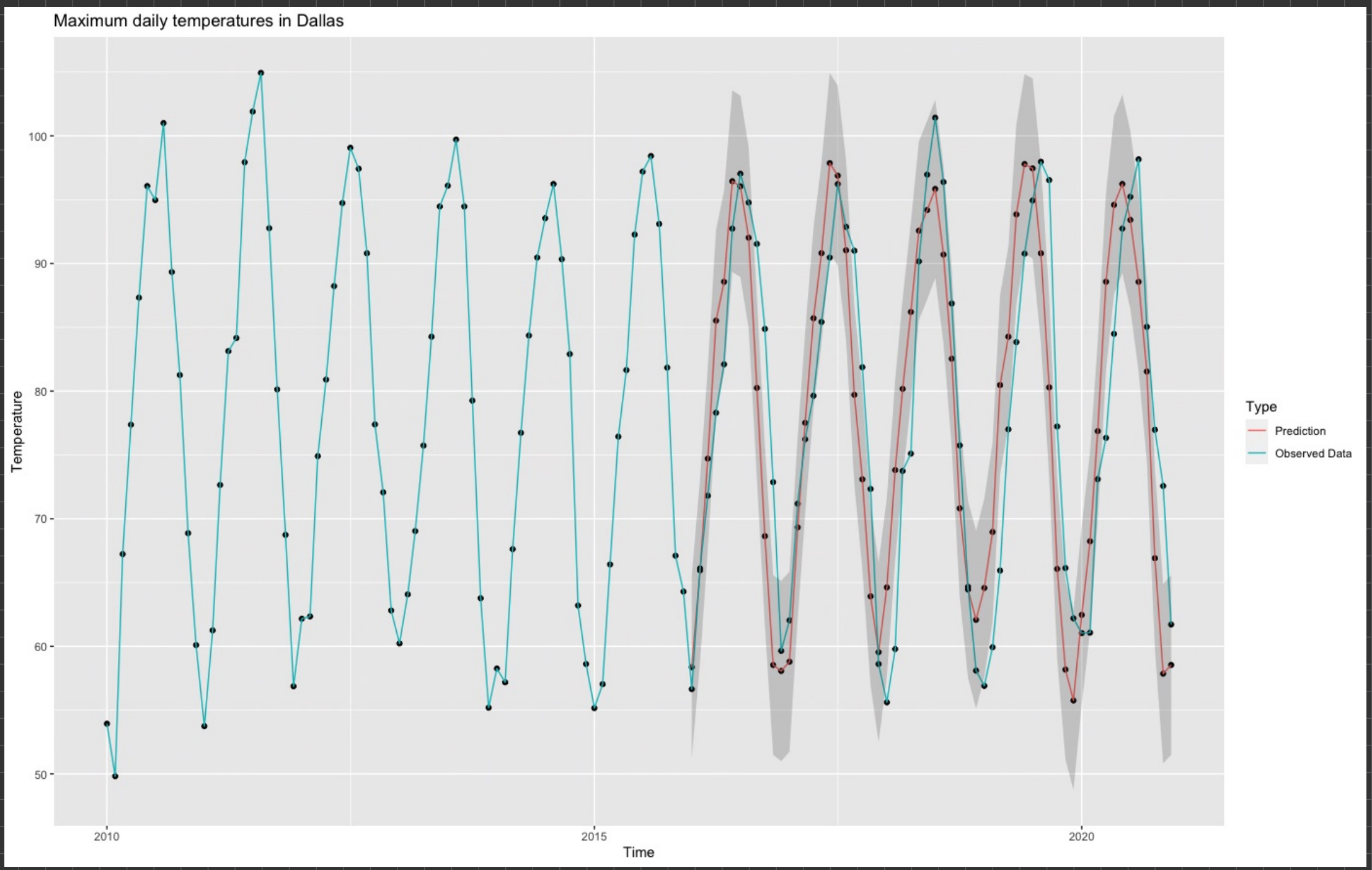
* The model indeed produce predictions that capture seasonal behaviour. The model captures cycles

* Comparison to model in C:

The two models are extremely similar in accuracy of prediction. However, the variance of this model is smaller, and the prediction band always cover the observed data. Hence, taking into account this point, this last model is slightly better than the last model.

e. 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?

The AICc for ARIMA(12,1,0) is 5.5322295



Prediction of AMDT For January 2018 with Lower Bound = 57.60232

Upper Bound = 71.63128

64.61688

* The model indeed produce predictions that capture seasonal behaviour. The model captures cycles

The model is better than models in b and c. The variance is smaller and the prediction bandwith covers the observed data. The prediction line is smoother, so much less noise than the other two models.

APPENDIX (CODE FOR C, D and E)

```
#-------------------------------question c-----------------------------------------

#Index of Month of January 2015
limit<-12*15+1
# Number of Predictions from 2015 to 2020 (5 years)
limit2<-limit+12*5-1

preds<-numeric(limit2-limit)
SE<-numeric(limit2-limit)

#predicting for each day in one year advance
for (i in limit:limit2)
{
  x_train<-data1[-(i+1:length(data1$Month)),]
  fit_for <- sarima.for(x_train$AvgTemp,
                        n.ahead = 12,
                        p = 1,
                        d = 1,
                        q = 0,
                        P=1,
                        D=0,
                        Q=1,
                        S=3,
                        plot = F)
  preds[i-limit+1]<-fit_for$pred[1]
  SE[i-limit+1]<-fit_for$se[1]

  print(x_train$Month[length(x_train$Month)])
  print(preds[i-limit+1])
  print(preds[i-limit+1]- 1.96*SE[i-limit+1])
  print(preds[i-limit+1]+1.96*SE[i-limit+1])
}
#Creating the data frames
pred_lim1<-limit+12
pred_lim2<-limit2+12

lim10<-12*10+1
lim20<-pred_lim2
pred_time<-data.frame(Type=factor(rep("Prediction", pred_lim2-pred_lim1+1)),Time=as.Date(data1$Month[pred_lim1:pred_lim2]),Temperature=preds)
f2010_2020<-data.frame(Type=factor(rep("Observed Data", lim20-lim10+1)),Time=as.Date(data1$Month[lim10:lim20]),Temperature=data1$AvgTemp[lim10:lim20])

fit_data <- bind_rows(pred_time,f2010_2020)
fit_data
# Plot data and forecasts
gg_fit <- ggplot(fit_data,aes(x = Time,y=Temperature,group=Type),colour=Type) +geom_point()+geom_line(aes(color=Type)) +
  geom_ribbon(data = pred_time,
              aes(x = Time,
                  ymin = Temperature - 1.96*SE,
                  ymax = Temperature + 1.96*SE),
              alpha = .2)+
  labs(title = "Maximum daily temperatures in Dallas")

gg_fit

```

```r
#------------------------------question d----------------------------------------
data1 <- read.csv("/Users/rafa/Documents/Master Austin/MAESTRÍA_AUSTIN/Advanced Predictive Models/HW2/file3.xls", header=TRUE, stringsAsFactors=FALSE)
#fitting to obtain AIC_c
fit <- sarima(data1$AvgTemp,
              p = 3,
              d = 1,
              q = 1,
              no.constant = TRUE,
              details = FALSE)
fit
#Index of the month January
limit<-12*15+1
#Number of predictions in 5 years
limit2<-limit+12*5-1
preds<-numeric(limit2-limit)
SE<-numeric(limit2-limit)
for (i in limit:limit2)
{
  x_train<-data1[-(i+1:length(data1$Month)),]
  fit_for <- sarima.for(x_train$AvgTemp,
                        n.ahead = 12,
                        p = 3,
                        d = 1,
                        q = 1,
                        plot = F)
  preds[i-limit+1]<-fit_for$pred[1]
  #Standard error
  SE[i-limit+1]<-fit_for$se[1]
  #Getting the prediction and upper and lowe bound
  print(x_train$Month[length(x_train$Month)])
  print(preds[i-limit+1])
  print(preds[i-limit+1]- 1.96*SE[i-limit+1])
  print(preds[i-limit+1]+1.96*SE[i-limit+1])
}
#Making dataframes
pred_lim1<-limit+12
pred_lim2<-limit2+12
lim10<-12*10+1
lim20<-pred_lim2

pred_time<-data.frame(Type=factor(rep("Prediction", pred_lim2-pred_lim1+1)),Time=as.Date(data1$Month[pred_lim1:pred_lim2]),Temperature=preds)
f2010_2020<-data.frame(Type=factor(rep("Observed Data", lim20-lim10+1)),Time=as.Date(data1$Month[lim10:lim20]),Temperature=data1$AvgTemp[lim10:lim20])
fit_data <- bind_rows(pred_time,f2010_2020)
fit_data
# Plot data and forecasts
gg_fit <- ggplot(fit_data,aes(x = Time,y=Temperature,group=Type),colour=Type) +geom_point()+geom_line(aes(color=Type)) +
  geom_ribbon(data = pred_time,
              aes(x = Time,
                  ymin = Temperature - 1.96*SE,
                  ymax = Temperature + 1.96*SE),
              alpha = .2)+
  labs(title = "Maximum daily temperatures in Dallas")

gg_fit
```

```r

#Same procedure as d) ...just changing the model to ARIMA(12,1,0)
data1 <- read.csv("/Users/rafa/Documents/Master Austin/MAESTRÍA_AUSTIN/Advanced Predictive Models/HW2/file3.xls", header=TRUE, stringsAsFactors=FALSE)
fit <- sarima(data1$AvgTemp,
              p = 12,
              d = 1,
              q = 0,
              no.constant = TRUE,
              details = FALSE)
fit
limit<-12*15+1
limit2<-limit+12*5-1
preds<-numeric(limit2-limit)
SE<-numeric(limit2-limit)
for (i in limit:limit2)
{
  x_train<-data1[-(i+1:length(data1$Month)),]
  fit_for <- sarima.for(x_train$AvgTemp,
                        n.ahead = 12,
                        p = 12,
                        d = 1,
                        q = 0,
                        plot = F)
  preds[i-limit+1]<-fit_for$pred[1]
  SE[i-limit+1]<-fit_for$se[1]
  print(x_train$Month[i])
  print(preds[i-limit+1])
  print(preds[i-limit+1]- 1.96*SE[i-limit+1])
  print(preds[i-limit+1]+1.96*SE[i-limit+1])
}
pred_lim1<-limit+12
pred_lim2<-limit2+12
lim10<-12*10+1
lim20<-pred_lim2
pred_time<-data.frame(Type=factor(rep("Prediction", pred_lim2-pred_lim1+1)),Time=as.Date(data1$Month[pred_lim1:pred_lim2]),Temperature=preds)
f2010_2020<-data.frame(Type=factor(rep("Observed Data", lim20-lim10+1)),Time=as.Date(data1$Month[lim10:lim20]),Temperature=data1$AvgTemp[lim10:lim20])
fit_data <- bind_rows(pred_time,f2010_2020)
fit_data
# Plot data and forecasts
gg_fit <- ggplot(fit_data,aes(x = Time,y=Temperature,group=Type),colour=Type) +geom_point()+geom_line(aes(color=Type)) +
  geom_ribbon(data = pred_time,
              aes(x = Time,
                  ymin = Temperature - 1.96*SE,
                  ymax = Temperature + 1.96*SE),
              alpha = .2)+
  labs(title = "Maximum daily temperatures in Dallas")

gg_fit

```