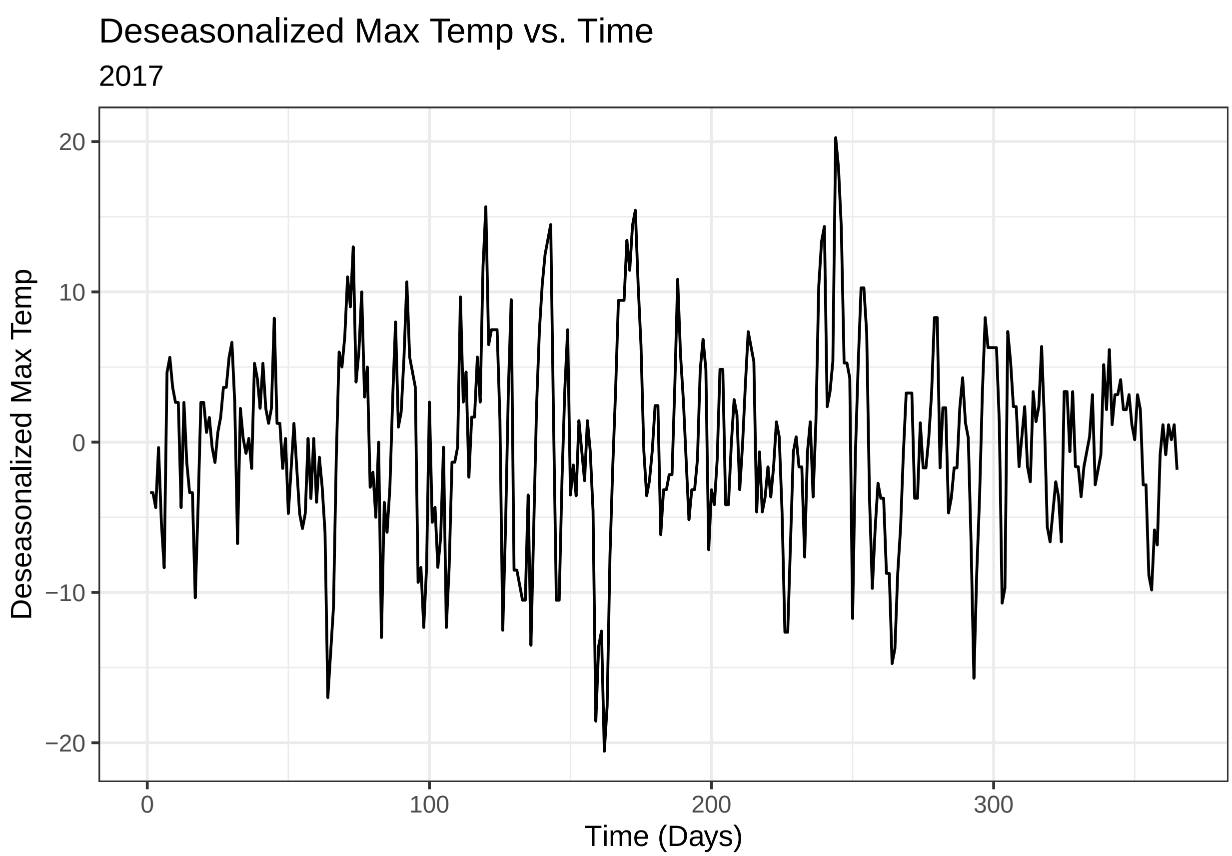
ECN190 Homework 4

Kevin Chen (914861432) John Mayhew (914807483)

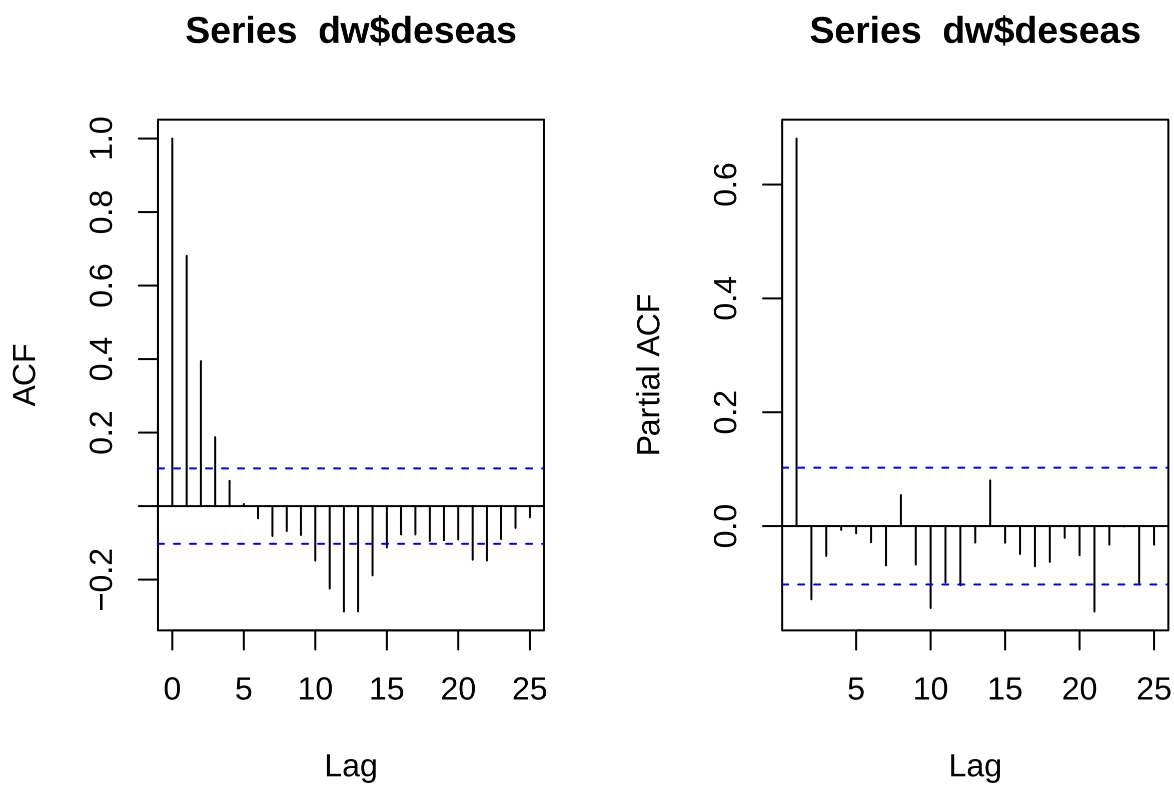
5/6/2020

1. Focus on year 2017. Construct a deseasonalized maximum temperature time series of year 2017.

```
##         date precip maxtemp mintemp year month day
## 1 2017-01-01   0.00      51      40 2017     1   1
## 2 2017-01-02   0.00      51      40 2017     1   2
## 3 2017-01-03   0.67      50      44 2017     1   3
## 4 2017-01-04   1.05      54      43 2017     1   4
## 5 2017-01-05   0.16      49      35 2017     1   5
## 6 2017-01-06   0.00      46      33 2017     1   6
```



2. Plot the autocorrelation graph of the deseasonalized maximum temperature time series of year 2017. Then plot the partial autocorrelation graph.



We see a sudden drop in the PACF plot but a gradual decline in the ACF plot, which means an AR model should be used for this time series data.

3. Run an AR(1) regression using the deseasonalized maximum temperature time series of year 2017. Then test for the assumption of no serial correlation using the BreuschGodfrey test.

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  ar1.model
## LM test = 6.0266, df = 1, p-value = 0.01409
```

The p-value from this BG test is less than 0.05, this means we reject H_0 and conclude that there is serial correlation in this model. We should consider adding more lags into this model to account for this significant serial correlation.

4. Run an AR(2) regression using the deseasonalized maximum temperature time series of year 2017. Then test for the assumption of no serial correlation using the BreuschGodfrey test.

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  ar2.model
## LM test = 1.0522, df = 1, p-value = 0.305
```

The p-value from this BG test is greater than 0.05, this means we fail to reject H_0 and conclude that there is no serial correlation in this model. This makes sense as previously, we concluded there was evidence of serial correlation. We fix this by adding more lags, which is what we did by using an AR(2) model, and as a result, there is no longer serial correlation.

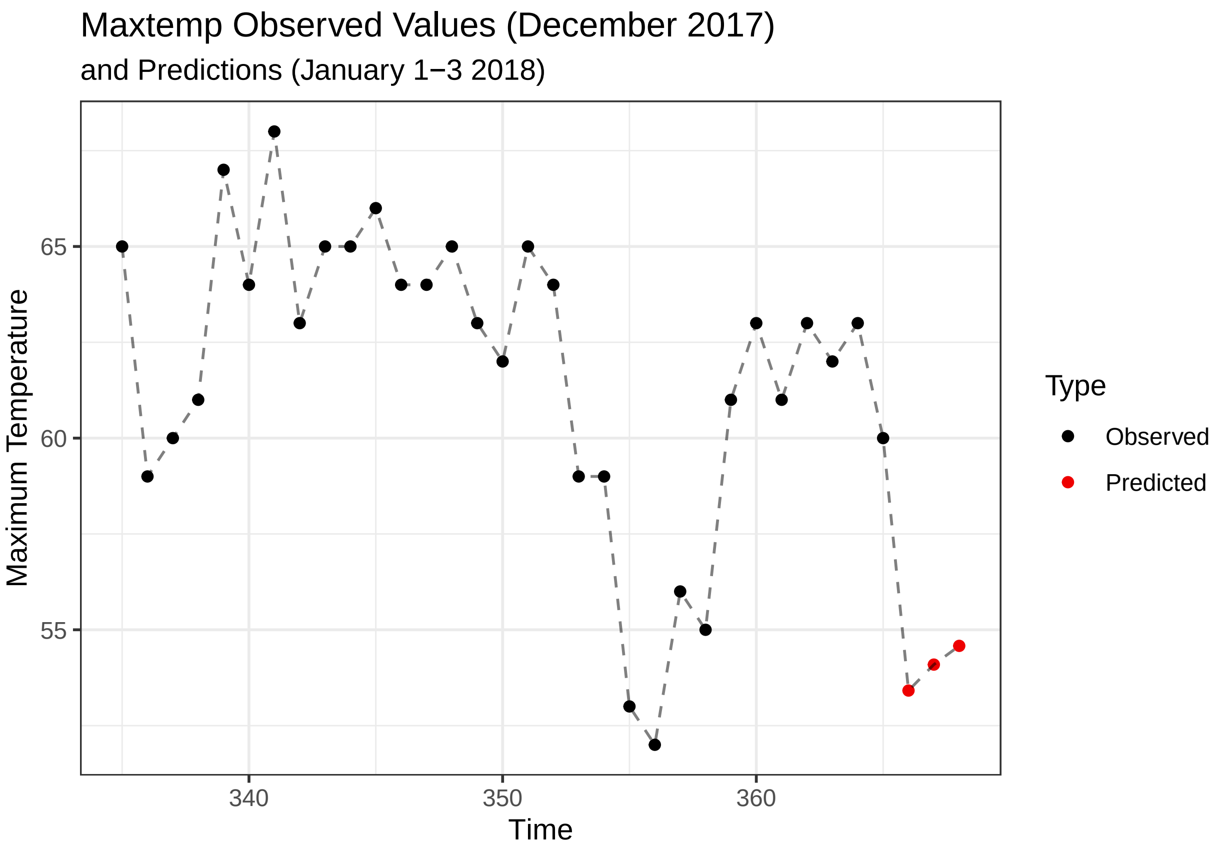
5. Now, fit the deseasonalized maximum temperature time series of year 2017 with an ARIMA(p,d,q) model. Use the "auto.arima" command in the "forecast" package to automatically pick p, d, and q. What regression model does the R command end up picking for this time series?

```
## Series: dw$deseas
## ARIMA(2,0,1) with zero mean
##
## Coefficients:
##          ar1      ar2      ma1
##       1.6522  -0.7019  -0.9892
## s.e.  0.0370   0.0369   0.0104
##
## sigma^2 estimated as 20.8:  log likelihood=-1071.36
## AIC=2150.72   AICc=2150.83   BIC=2166.32
```

The R command automatically selects an ARMA(2,1) model.

6. Use the "forecast" command in the "forecast" package to forecast the deseasonalized maximum temperature of Jan. 1-3, 2018 using the model you obtained in the last question. Note that these are the deseasonalized time series. How would you forecast the raw maximum temperature of Jan. 1-3, 2018?

```
## [1] -0.9401883 -0.2628302  0.2256515
```



To forecast the raw maximum temperature for Jan 1-3 for 2018, we need to manually add back the seasonality into the forecast of deseasonalized maximum temperature. Since the month is Januaray, this means all the dummy variables in our seasonality model = 0, meaning we just add the intercept to our deseasonalized values to get the raw maximum temperature prediction of Jan 1-3 of 2018.

Appendix

```
library(tidyverse)
library(ggthemes)
library(Hmisc)
library(lmtest)
library(forecast)

DavisWeather = read.csv("DavisWeather.csv")[,-1]
head(DavisWeather)

dw = DavisWeather %>% filter(year == 2017)
seasonal.model = lm(maxtemp ~ as.factor(month), data = dw)
dw$deseas = seasonal.model$residuals
dw$t = 1:nrow(dw)
dw %>% ggplot(aes(x = t, y = deseas)) + geom_line() + theme_bw() +
  scale_x_continuous("Time (Days)") + scale_y_continuous("Deseasonalized Max Temp") +
  ggtitle("Deseasonalized Max Temp vs. Time", subtitle = "2017")
par(mfrow = c(1,2))
acf(dw$deseas)
pacf(dw$deseas)
par(mfrow = c(1,1))
dw$L1.deseas = Lag(dw$deseas, 1)
ar1.model = lm(deseas ~ L1.deseas, data = dw)
bgtest(ar1.model)
cat("\\newpage")
dw$L2.deseas = Lag(dw$deseas, 2)
ar2.model = lm(deseas ~ L1.deseas + L2.deseas, data = dw)
bgtest(ar2.model)
auto.model = auto.arima(dw$deseas);auto.model
cat("\\newpage")
f = forecast(auto.model, h = 3)
#c(f$x[350:365])

c(f$mean)
rawpred = seasonal.model$coefficients[1] + f$mean

Observed = dw$maxtemp[335:365]
Predicted = rawpred
mtemp = c(Observed, Predicted)
Type = c(rep("Observed", 31), rep("Predicted", 3))
t = 335:368
cbind.data.frame(t, mtemp, Type) %>%
  ggplot(aes(x = t, y = mtemp)) + geom_point(aes(color = Type)) +
  geom_line(linetype = "dashed", alpha = I(1/2)) + theme_bw() +
  ggtitle("Maxtemp Observed Values (December 2017)",
          subtitle = "and Predictions (January 1-3 2018)") +
  scale_x_continuous("Time") + scale_y_continuous("Maximum Temperature") +
  scale_color_manual(values = c("black", "red2"))
cat("\\newpage")
```