Time Series HW # 5

Liam Flaherty

Professor Martin

NCSU: ST534-001

October 7, 2024

1) Analyze the fourth data set on Moodle.

a. Determine possible models for the data set using diagnostics such as the ACF, PACF, and white noise test. Include a unit root test and discuss those results as well.

Our first step is to plot the data, which we do in Figure 0.1 below.

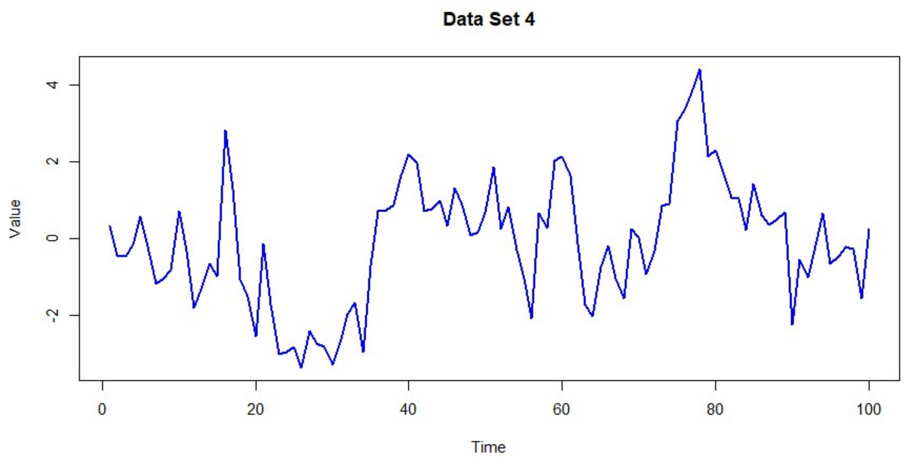

Figure 0.1: Data Set 4 Time Series

The Ljung-Box White Noise test has a p-value on the order of machine-epsilon for lags of 6 and 12 (see Figure 0.2 below)– we need to fit a model.

```
> whitenoise12=Box.test(ts_data4,
+                       lag=12,
+                       type="Ljung-Box")
> whitenoise12

        Box-Ljung test

data:  ts_data4
X-squared = 168.44, df = 12, p-value < 2.2e-16
```

Figure 0.2: White Noise Test For Data Set 4

The Augmented-Dickey Fuller test in Figure 0.3 suggests that we might need to take a difference.

```
> ###2b. Augmented Dickey-Fuller (Unit Root Test)###
> adf_result_data4=adf.test(ts_data4)          #Null is that there is a root#
> adf_result_data4                             #since p is 0.2, don't reject null; assume non-stationary#

        Augmented Dickey-Fuller Test

data:  ts_data4
Dickey-Fuller = -2.8152, Lag order = 4, p-value = 0.2389
alternative hypothesis: stationary
```

Figure 0.3: Augmented-Dickey Fuller Test (Unit Root Test)

After doing so, our ACF and PACF for the differenced data are shown in Figure 0.4. Notice that neither the ACF nor PACF seems to completely die off.

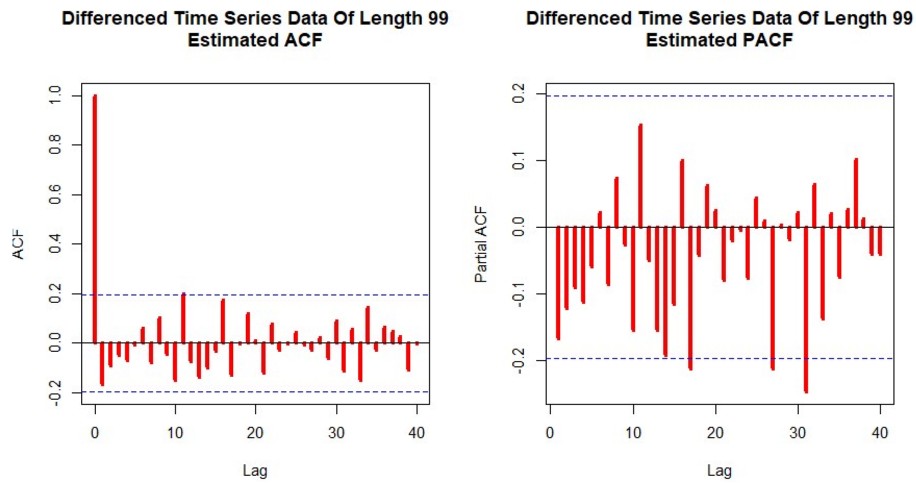

Figure 0.4: ACF And PACF Of Differenced Data

Again performing the white-noise test, this time on the differenced series, we see that the model is not distinguishable from white noise in Figure 0.5.

```
> diff_data4=diff(ts_data4, differences = 1)
> whitenoise6=Box.test(diff_data4,                #Do we need to fit model?#
+                      lag=6,
+                      type="Ljung-Box")
> whitenoise6                                    #large p \implies no#

        Box-Ljung test

data:  diff_data4
X-squared = 4.8704, df = 6, p-value = 0.5605

> whitenoise12=Box.test(diff_data4,               #Do we need to fit model?#
+                      lag=12,
+                      type="Ljung-Box")
> whitenoise12                                   #large p \implies no#

        Box-Ljung test

data:  diff_data4
X-squared = 14.568, df = 12, p-value = 0.2659
```

Figure 0.5: White Noise Test For Differenced Data

Our assessment of the model agrees with the `auto.arima()` function from R's `forecast` package; it recommends an ARIMA(0,1,0) model.

```
> auto.arima(ts_data4)
Series: ts_data4
ARIMA(0,1,0)

sigma^2 = 1.28:  log likelihood = -152.69
AIC=307.38   AICc=307.42   BIC=309.97
```

Figure 0.6: auto.arima() For Data Set 4

b. Fit the models that you identified as good possibilities and compare their fits using output diagnostics such as the residual test for white noise, AIC, SBC, etc.

Just for thoroughness, we test a few different models up to order 3 in Figure 0.7.

```r
###2d. Try A Few Different Models###
ARIMA_model=vector()
aic=vector()
bic=vector()
LBtest=vector()

for (p in 1:4) {
  for (d in 1:2) {
    for (q in 1:4) {
            model=arima(ts_data4,
                        order=c(p-1,d-1,q-1))
            resid=residuals(model)

            ARIMA_model[4*2*(p-1)+4*(d-1)+q]=paste0("ARIMA(", p-1, ",", d-1, ",", q-1, ")")
            aic[4*2*(p-1)+4*(d-1)+q]=round(AIC(model),2)
            bic[4*2*(p-1)+4*(d-1)+q]=round(BIC(model),2)
            LBtest[4*2*(p-1)+4*(d-1)+q]=round(Box.test(resid, lag=21, type="Ljung-Box")$p.value,2)
    }
  }
}

df=data.frame(ARIMA_model, aic, bic, LBtest)
df=df[order(df$aic),]
df
```

Figure 0.7: R Code To Derive Model Diagnostics

Figure 0.8 below sorts our diagnostics from lowest to highest AIC values.

```
> df=df[order(df$aic),]
> df
    ARIMA_model    aic    bic LBtest
14 ARIMA(1,1,1) 301.39 309.17   0.47
9  ARIMA(1,0,0) 301.54 309.36   0.52
15 ARIMA(1,1,2) 302.93 313.31   0.51
22 ARIMA(2,1,1) 302.97 313.35   0.50
10 ARIMA(1,0,1) 303.21 313.63   0.55
17 ARIMA(2,0,0) 303.23 313.65   0.54
27 ARIMA(3,0,2) 304.75 322.99   0.64
23 ARIMA(2,1,2) 304.77 317.75   0.54
16 ARIMA(1,1,3) 304.79 317.77   0.56
30 ARIMA(3,1,1) 304.83 317.81   0.56
18 ARIMA(2,0,1) 305.13 318.16   0.58
11 ARIMA(1,0,2) 305.14 318.16   0.59
25 ARIMA(3,0,0) 305.15 318.18   0.59
6  ARIMA(0,1,1) 305.54 310.73   0.33
7  ARIMA(0,1,2) 305.70 313.49   0.52
32 ARIMA(3,1,3) 306.38 324.55   0.56
13 ARIMA(1,1,0) 306.50 311.69   0.24
31 ARIMA(3,1,2) 306.55 322.12   0.46
24 ARIMA(2,1,3) 306.77 322.34   0.55
21 ARIMA(2,1,0) 307.05 314.83   0.46
8  ARIMA(0,1,3) 307.09 317.47   0.42
12 ARIMA(1,0,3) 307.13 322.76   0.59
19 ARIMA(2,0,2) 307.13 322.76   0.59
26 ARIMA(3,0,1) 307.13 322.76   0.59
5  ARIMA(0,1,0) 307.38 309.97   0.15
29 ARIMA(3,1,0) 308.25 318.63   0.41
20 ARIMA(2,0,3) 309.13 327.37   0.58
28 ARIMA(3,0,3) 310.65 331.49   0.57
4  ARIMA(0,0,3) 313.15 326.17   0.06
3  ARIMA(0,0,2) 320.13 330.55   0.00
2  ARIMA(0,0,1) 334.72 342.53   0.00
1  ARIMA(0,0,0) 384.68 389.89   0.00
```

Figure 0.8: Model Diagnostics For Data Set 4

In terms of AIC, the ARIMA(0,1,0) is actually one of the worst performing models. If one model had to be chosen, we would prefer the ARIMA(1,0,0) for the sake of parsimony. It is only slightly worse than the ARIMA(1,1,1) model (the combined difference in AIC and BIC in the two models is less than 0.5) while having a higher Ljung-Box p-value and two less parameters.

2) Analyze the quarterly beer data set on Moodle.

a. Determine possible models for the data using diagnostics such as the ACF and white noise test. Include a unit root test and discuss those results as well.

Our first step is to plot the data, which we do in Figure 0.9 below.

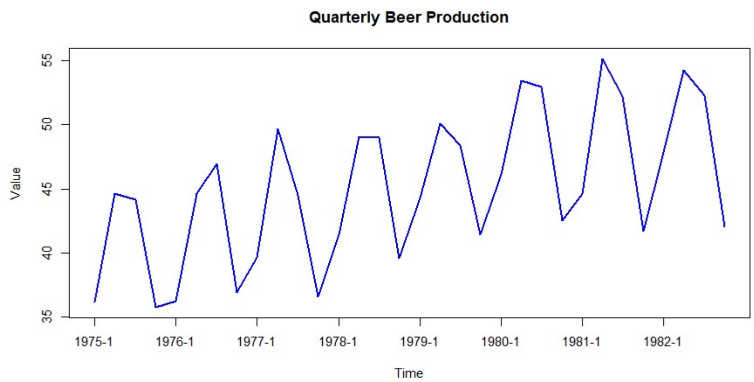

Figure 0.9: Quarterly Beer Time Series

It is clear from inspection that we have seasonal data with period 4. We are dealing with a limited amount of data (thirty-two total observations with a period of 4 means 8 seasonal observations), but at least visually, it seems that the series is trending; all but one of the seven points is larger than it's previous value. Applying the Augmented Dickey-Fuller Test at seasonal increments provides evidence for the alternative hypothesis that the seasonal lags are actually stationary. This is shown in Figure 0.10 below along with the differenced plot. We will test both when building our models.

```
> ts_beer_seasonal=ts_beer[seq(2, length(ts_beer), by=4)]   #Only look at every 4th#
> ts_beer_seasonal
[1] 44.60 44.63 49.72 49.07 50.09 53.44 55.18 54.27
> sadf_result=suppressWarnings(adf.test(ts_beer_seasonal))
> sadf_result

        Augmented Dickey-Fuller Test

data:  ts_beer_seasonal
Dickey-Fuller = -5.5155, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary
```

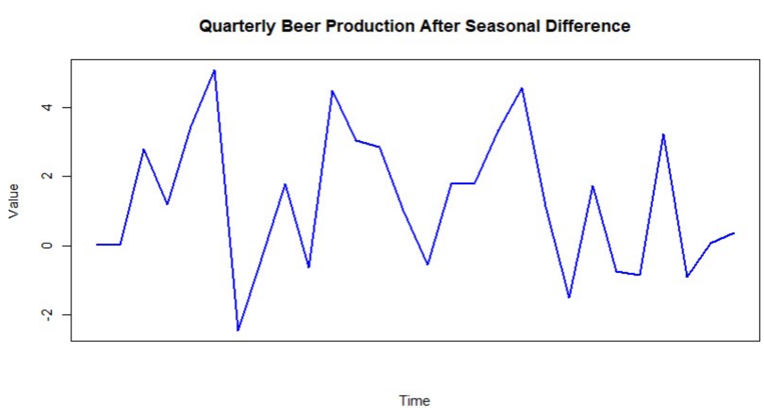

Figure 0.10: Quarterly Beer Time Series After Seasonal Difference

Our initial ACF and PACF plots are shown in Figure 0.11 below. The seasonal lags in the ACF plot seem to quickly die out while the first seasonal lag in the PACF seems pronounced and subsequently cuts off. A natural choice for the seasonal portion of the model is an MA(1).

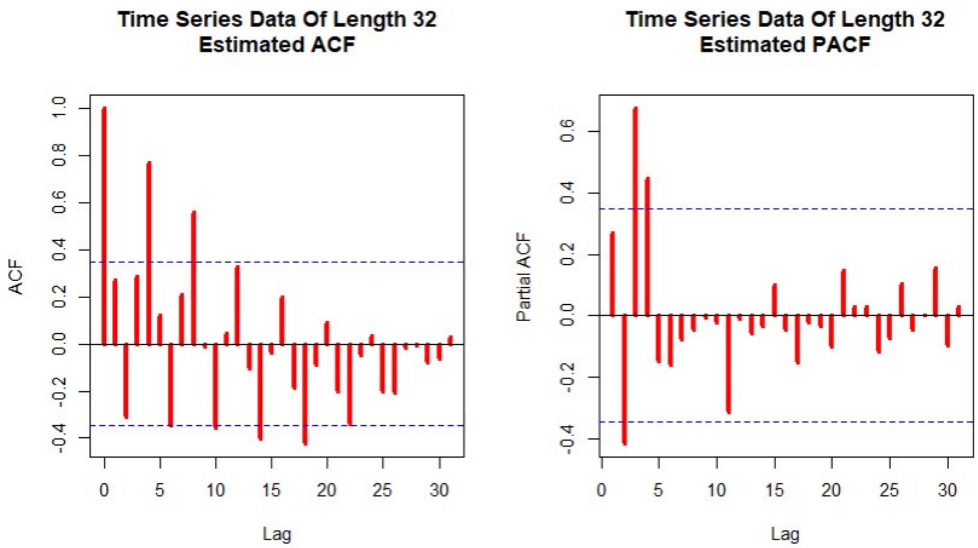

Figure 0.11: ACF And PACF Of Quarterly Beer

We can also look at the ACF and PACF after taking a seasonal difference. This is shown in Figure 0.12. Notice that the ACF has a semi-gradual sinusoidal decay, and the PACF cuts off after the first seasonal lag. An argument could be made that neigher really cuts off but instead gradually decays. From that perspective, some choices for the seasonal portion could be an ARIMA(1,1,0), and ARIMA(0,1,1), or an ARIMA(1,1,1)

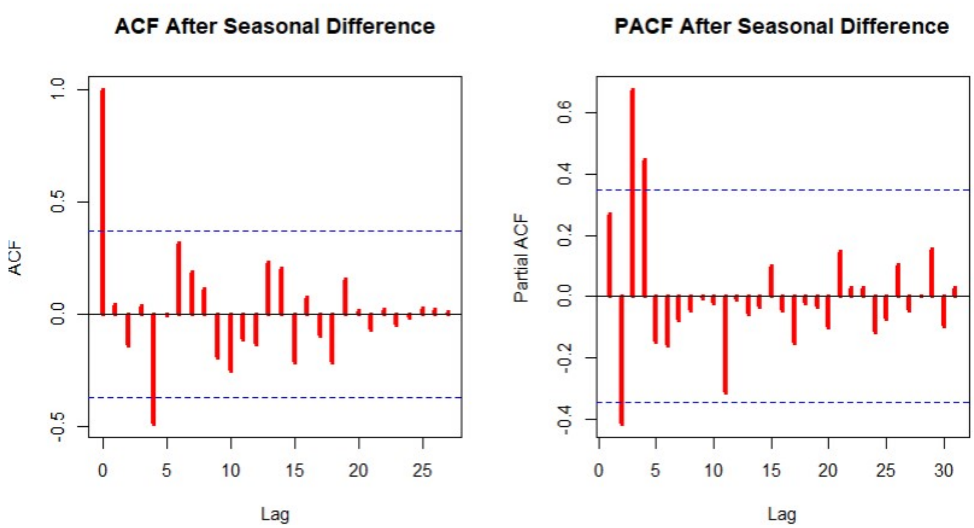

Figure 0.12: ACF And PACF After Seasonal Difference

We now try to determine the non-seasonal part of the model. We can test if a difference is needed by using the Augmented Dickey-Fuller test. Under any reasonable alpha level, we fail to reject the null hypothesis of "there is a unit root". The results are shown in Figure 0.13 below.

```
> adf_result_beer=adf.test(ts_beer)
> adf_result_beer

        Augmented Dickey-Fuller Test

data:  ts_beer
Dickey-Fuller = -1.4835, Lag order = 3, p-value = 0.7721
alternative hypothesis: stationary
```

Figure 0.13: ADF For Beer Data

The ACF and PACF, after taking a difference, are shown in Figure 0.14 below. Notice that the ACF dies out immediately, while the PACF cuts off after the third lag.

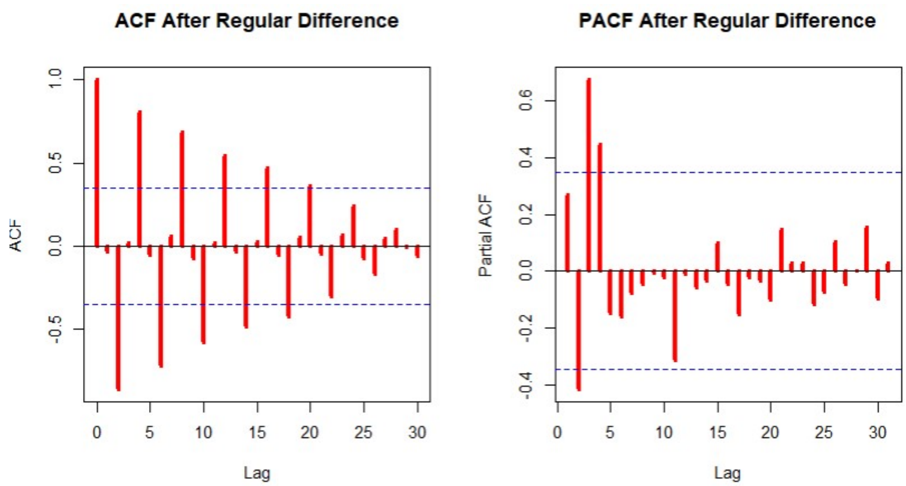

Figure 0.14: ACF And PACF After Regular Difference

Taken together, we have a couple of models that might be good fits: an ARIMA(0,1,3)(0,0,1), an ARIMA(0,1,3)(1,1,0), an ARIMA(0,1,3)(0,1,1), and an ARIMA(0,1,3)(1,1,1) all seem reasonable.

b. Fit the models that you identified as good possibilities and compare their fits using output diagnostics such as the residual test for white noise, AIC, SBC, etc.

For thoroughness, we test all models with $p, q, P,$ and Q terms less than 4 with d and D terms less than 2. The script to run this is shown in Figure 0.15 below.

```
###3f. Try A Few Different Models###
ARIMAs_model=vector()
aic=vector()
bic=vector()
LBtest=vector()
s=4                    #clear from data#

for (p in 1:4) {
  for (d in 1:2) {
    for (q in 1:4) {
      for (P in 1:4) {
        for (D in 1:2) {
          for (Q in 1:4) {
            mycount=(((((p-1)*2 + (d-1))*4 + (q-1))*4 + (P-1))*2 + (D-1))*4 + Q
            mymodel=paste0("ARIMA(", p-1, ",", d-1, ",", q-1, ")(", P-1, ",", D-1, ",", Q-1, ")",s)

            tryCatch({
              model=arima(ts_beer,
                          order=c(p-1,d-1,q-1),
                          seasonal=list(order=c(P-1,D-1,Q-1), period=s))
              resid=residuals(model)

              ARIMAs_model[mycount]=mymodel
              aic[mycount]=round(AIC(model),2)
              bic[mycount]=round(BIC(model),2)
              LBtest[mycount]=round(Box.test(resid, lag=21, type="Ljung-Box")$p.value,2)
            }, error=function(e) {
              ARIMAs_model[mycount]=mymodel
              aic[mycount]=999
              bic[mycount]=999
              LBtest[mycount]=999
            })
          }
        }
      }
    }
  }
}

df=data.frame(ARIMAs_model, aic, bic, LBtest)
df=df[order(df$aic),]
df
```

Figure 0.15: Script For Fitting Models

The top ten models in terms of AIC are shown in Figure 0.16 below. Of the models we planned to test in part 2a, the ARIMA(0,1,3)(1,1,1) actually was the second best model overall. Right behind was our ARIMA(0,1,3)(0,1,1) model. Since the data was so short, the marginally better AIC and BIC from the model with a second seasonal MA term is not as convincing as the ARIMA(0,1,3)(1,1,1) we proposed. For that reason, we would prefer that model best of all.

```
> df[1:10,]
            ARIMAs_model    aic    bic LBtest
231 ARIMA(0,1,3)(0,1,2)4 111.78 119.55   0.82
238 ARIMA(0,1,3)(1,1,1)4 111.97 119.74   0.80
230 ARIMA(0,1,3)(0,1,1)4 112.28 118.75   0.35
245 ARIMA(0,1,3)(2,1,0)4 112.51 120.28   0.82
487 ARIMA(1,1,3)(0,1,2)4 112.80 121.87   0.86
486 ARIMA(1,1,3)(0,1,1)4 113.14 120.92   0.51
494 ARIMA(1,1,3)(1,1,1)4 113.20 122.27   0.81
237 ARIMA(0,1,3)(1,1,0)4 113.24 119.71   0.52
647 ARIMA(2,1,0)(0,1,2)4 113.63 120.11   0.74
166 ARIMA(0,1,1)(0,1,1)4 113.70 117.59   0.31
```

Figure 0.16: Model Selection Criteria For Beer Data