Project #4

STAT 878

Spring 2022

Nannan Wang

Complete the following problems below. Within each part, include your R program output with code inside of it and any additional information needed to explain your answer. Your R code and output should be formatted in the exact same manner as in the course notes.

1. (25 total points) The purpose of this problem is for you to find the best ARIMA model for the radon data set and then forecast future values for it. Perform all 6 steps of the model building process as given in the notes (exclude a forecast error examination). Make sure to include all code and output with your answers. For the forecasting part, state the forecasts and their confidence intervals at each hour for 12 hours into the future and include the corresponding plots.

* Step 1: If the data is stationary:

> radon <- read.csv(file="/Users/nannan/Library/CloudStorage/OneDrive-> UniversityofNebraska-Lincoln/2022 Spring/STAT-878/project data/radon.csv",header = T)

> head(radon)

> x <- radon$Radon

> plot(x=x,type = 'l',col="red")

> points(x=x,pch=20,col="blue")

Chart

Description automatically generated

From the plot above, there’s no big concern of non-stationary in the variance and mean, although at x=10 there’s a point that deviate from values of other point. Having one point deviating from other points is not a big concern in terms of stationary in the variance. Also, all points tend to oscillate in a range, which shows stationary in the mean. For checking the non-stationary in the mean, a ACF plot can be helpful as well:

> acf(x=x,lag.max = 20,type = "correlation") Chart, histogram

Description automatically generated

From the ACF plot, only the first lag has significant autocorrelation. Therefore, it shows stationary in the mean.

* Step 2: Construct plots of the estimated ACF and PACF of the stationary series

> par(mfrow=c(1,2))

> acf(x=x,type = "correlation",lag.max = 20)

> pacf(x=x,lag.max = 20)

Chart, histogram

Description automatically generated

From ACF plot, there’s only the first lag has significant autocorrelation, and it seems to tail off after lag 1. In the PACF plot, there’s only the first lag that has significant partial autocorrelation and it seems to tail off after lag 1. Considering these trend, I would try ARIMA(1,0,1) model.

In addition, I also want to try AR(1) model since in the PACF plot, the second lag has a smaller partial autocorrelation value compared to the first one, which may indicate a cut off to 0 trend after lag 1 in the PACF plot.

* Step 3: Find the estimated models using maximum likelihood estimation.

ARIMA(1,0,1) model:

> model.101 <- arima(x=x,order = c(1,0,1),include.mean = TRUE)

> model.101 Call:

arima(x = x, order = c(1, 0, 1), include.mean = TRUE)

Coefficients:

ar1 ma1 intercept

0.4969 -0.2154 6.7834

s.e. 0.2575 0.2862 0.2822

sigma^2 estimated as 2.926: log likelihood = -172.15, aic = 352.31

From the output, =0.4969, =-0.2154, =6.7834, and =2.926. Therefore, the model can be written as:

AR(1) model:

> model.100 <- arima(x=x, order =c(1,0,0),include.mean = TRUE)

> model.100

Call:

arima(x = x, order = c(1, 0, 0), include.mean = TRUE)

Coefficients:

ar1 intercept

0.3013 6.7843

s.e. 0.1054 0.2605

sigma^2 estimated as 2.943: log likelihood = -172.4, aic = 350.8

From the output, =0.3013, =6.7834, and =2.943. Therefore, the model can be written as:

* Step 4: For each model chosen, investigate the diagnostic measures:

ARIMA(1,0,1) model:

> model.101 <- arima(x=x,order = c(1,0,1),include.mean = TRUE)

> model.101

Call:

arima(x = x, order = c(1, 0, 1), include.mean = TRUE)

Coefficients:

ar1 ma1 intercept

0.4969 -0.2154 6.7834

s.e. 0.2575 0.2862 0.2822

sigma^2 estimated as 2.926: log likelihood = -172.15, aic = 352.31

> examine.mod(model.101,mod.name = "ARIMA(1,0,1)")

$z

ar1 ma1 intercept

1.9299203 -0.7527674 24.0411637

$p.value

ar1 ma1 intercept

0.05361672 0.45158970 0.00000000

Chart, line chart

Description automatically generated

A picture containing application

Description automatically generated

AR(1) model:

> model.100 <- arima(x=x, order =c(1,0,0),include.mean = TRUE)

> model.100

Call:

arima(x = x, order = c(1, 0, 0), include.mean = TRUE)

Coefficients:

ar1 intercept

0.3013 6.7843

s.e. 0.1054 0.2605

sigma^2 estimated as 2.943: log likelihood = -172.4, aic = 350.8

> examine.mod(model.100,mod.name = "AR(1)")

$z

ar1 intercept

2.859759 26.043351

$p.value

ar1 intercept

0.004239636 0.000000000

Chart, line chart

Description automatically generated

A picture containing graphical user interface

Description automatically generated

I think both AR(1) and ARIMA(1,0,1) are good for modeling the Radon data because there is no concern about outlier for both models. As shown in the QQ plots of two models, standardized residuals of both models follow closely the normal distribution. There is no significant values in the ACF, PACF, and Ljung-Box statistics plots from both models.

* Step 5: Using each model tht gets through 4, pick the best model based upon model parsimony and the AIC.

To pick the best model, I use AIC to help me make this decision:

> model.101

Call:

arima(x = x, order = c(1, 0, 1), include.mean = TRUE)

Coefficients:

ar1 ma1 intercept

0.4969 -0.2154 6.7834

s.e. 0.2575 0.2862 0.2822

sigma^2 estimated as 2.926: log likelihood = -172.15, aic = 352.31

> model.100

Call:

arima(x = x, order = c(1, 0, 0), include.mean = TRUE)

Coefficients:

ar1 intercept

0.3013 6.7843

s.e. 0.1054 0.2605

sigma^2 estimated as 2.943: log likelihood = -172.4, aic = 350.8

ARIMA(1,0,1) has AIC value 352.31 and AR(1) has AIC 350.8. Therefore I will propose AR(1) model for forecasting because it has smaller AIC and it only has fewer parameter to estimate compared to the ARIMA(1,0,1) model.

* Step 6: Forecasting

> fore.mod.100 <- predict(object=model.100, n.ahead=12, se.fit = TRUE)

> fore.mod.100

$pred

Time Series:

Start = 89

End = 100

Frequency = 1

[1] 8.054565 7.167053 6.899630 6.819051 6.794771 6.787455 6.785251 6.784586 6.784386 6.784326 6.784308 6.784302

$se

Time Series:

Start = 89

End = 100

Frequency = 1

[1] 1.715384 1.791564 1.798321 1.798933 1.798989 1.798994 1.798994 1.798995 1.798995 1.798995 1.798995 1.798995

Calculate 95% confidence interval:

> low <- fore.mod.100$pred-qnorm(p=0.975,mean=0,sd=1)\*fore.mod.100$se

> up <- fore.mod.100$pred+qnorm(p=0.975,mean=0,sd=1)\*fore.mod.100$se

> data.frame(low,up)

low up

1 4.692473 11.41666

2 3.655651 10.67845

3 3.374985 10.42427

4 3.293206 10.34490

5 3.268817 10.32072

6 3.261492 10.31342

7 3.259286 10.31122

8 3.258622 10.31055

9 3.258422 10.31035

10 3.258361 10.31029

11 3.258343 10.31027

12 3.258338 10.31027

Below is a table of the forecasts with their corresponding 95% confidence intervals:

|  |  |  |  |
| --- | --- | --- | --- |
| m |  | 95% C.I. Lower Limit | 95% C.I. Upper Limit |
| **1** | 8.054565 | 4.692473 | 11.41666 |
| **2** | 7.167053 | 3.655651 | 10.67845 |
| **3** | 6.899630 | 3.374985 | 10.42427 |
| **4** | 6.819051 | 3.293206 | 10.34490 |
| **5** | 6.794771 | 3.268817 | 10.32072 |
| **6** | 6.787455 | 3.261492 | 10.31342 |
| **7** | 6.785251 | 3.259286 | 10.31122 |
| **8** | 6.784586 | 3.258622 | 10.31055 |
| **9** | 6.784386 | 3.258422 | 10.31035 |
| **10** | 6.784326 | 3.258361 | 10.31029 |
| **11** | 6.784308 | 3.258343 | 10.31027 |
| **12** | 6.784302 | 3.258338 | 10.31027 |

The following plot shows observed values, forecasted values and 95% CI for forecasted values:

> # forecast plots

> plot(x = x, ylab = expression(x[t]), xlab = "t", type =

+ "o", col = "red", lwd = 1, pch = 20,

+ panel.first = grid(col = "gray", lty = "dotted"), xlim

+ = c(1, 100))

> lines(x = c(x - model.100$residuals, fore.mod.100$pred), lwd

+ = 1, col = "black", type = "o", pch = 17)

> lines(y = low, x = 89:100, lwd = 1, col = "darkgreen",

+ lty = "dashed")

> lines(y = up, x = 89:100, lwd = 1, col = "darkgreen",

+ lty = "dashed")

> legend(locator(1), legend = c("Observed", "Forecast",

+ "95% C.I."), lty = c("solid", "solid", "dashed"),

+ col = c("red", "black", "darkgreen"), pch = c(20,

+ 17, NA), bty = "n")

A picture containing text, wall

Description automatically generated

Zoomed in plot:

> # zoom in

> plot(x = x, ylab = expression(x[t]), xlab = "t", type =

+ "o", col = "red", lwd = 1, pch = 20,

+ panel.first = grid(col = "gray", lty = "dotted"), xlim

+ = c(80, 100))

> lines(x = c(x - model.100$residuals, fore.mod.100$pred), lwd

+ = 1, col = "black", type = "o", pch = 17)

> lines(y = low, x = 89:100, lwd = 1, col = "darkgreen",

+ lty = "dashed")

> lines(y = up, x = 89:100, lwd = 1, col = "darkgreen",

+ lty = "dashed")

> legend(locator(1), legend = c("Observed", "Forecast",

+ "95% C.I."), lty = c("solid", "solid", "dashed"),

+ col = c("red", "black", "darkgreen"), pch = c(20,

+ 17, NA), bty = "n")

Chart, line chart

Description automatically generated

1. (40 total points) The purpose of this problem is for you to find the best SARIMA model for the TB data set and then forecast future values for it.
   1. (7 points) Use the 2000-2007 years’ data (time points 1 to 96) only for the parts below.
      1. Investigate whether or not differencing with d = 1 and D = 1 (season length is 12) is necessary.

Yes, from the plot, it indicates non-stationary in the mean because there’s a trend in the time series that at lags that are multiples of 12 values tend to drop to a small value. The plot can be found by the following code:

> tb <- read.csv(file = "/Users/nannan/Library/CloudStorage/OneDrive-UniversityofNebraska-Lincoln/2022 Spring/STAT-878/project data/tb.csv",header = TRUE)

> tb.2007 <- tb[1:96,]

> x.2007 <- tb.2007$count

> plot(x=x.2007,type = "l",col="red")

> points(x=x.2007,pch=20,col="blue")

> abline(v=c(12,24,36,48,60,72,84,96),lty="dotted")

> axis(side=1,at=c(12,24,36,48,60,72,84,96),tck=-.01)

Chart, line chart

Description automatically generated

In addition, I looked at the acf plot of the 2000-2007 years data. From the ACF plot, the autocorrelations are very slowly going to 0 after lag 12, 24, 36, 48, 60, 72… . Therefore, first differences should be examined for s=12

> acf(x=x.2007,lag.max = 90)

> axis(side=1,at=c(12,24,36,48,60,72,84,96),tck=-.01)

> abline(v=c(12,24,36,48,60,72,84,96),lty="dotted")

Chart

Description automatically generated

Below is the code to examine (1-):

> x.s12 <- diff(x=x.2007,lag=12,differences = 1)

> plot(x=x.s12,type = "l",col="red")

> points(x=x.s12,pch=20,col="blue")

A picture containing different, several

Description automatically generated

There is no longer a nonstationary behavior in the resulting time series data.

* + 1. Michael Chen of the CDC used an ARIMA(0,1,1)×(0,1,1)12 model for the data. State the estimated model and evaluate it as we would in step 4 of the model building process.

I used the following R code to estimate the CDC model:

> model.CDC <- arima(x=x.2007,order = c(0,1,1),seasonal = list(order=c(0,1,1), period=12))

> model.CDC

Call:

arima(x = x.2007, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))

Coefficients:

ma1 sma1

-0.9720 -0.4518

s.e. 0.0609 0.1409

sigma^2 estimated as 2815: log likelihood = -450.58, aic = 907.16

From the output, =-0.972, =-0.4518, =2815.

The estimated model used by Michael Chen can be written as:

1. )(1-)=(1-0.4518)(1-0.972) Where =2815
   * 1. Comment on the appropriateness of the model used by the CDC.

I used the examine.mod() function to conduct those diagnostic tests. The R code and plots are:

> examine.mod(model.12,mod.name = "CDC model",max.lag = 50)

$z

ma1 sma1

-15.968296 -3.205972

$p.value

ma1 sma1

0.000000000 0.001346072

Chart, line chart

Description automatically generated

Graphical user interface, application

Description automatically generated

Although there’s not a big concern about outlier and normality assumption, there is evidence against the independent assumption because from the Ljung-Box-Pierce test because there are multiple Ljung-Box statistics having p-value less than 0.05 (highlighted in green box), which indicates autocorrelation among the standardized residuals. In addition, from the standardized residual ACF plot, there are significant autocorrelation at lag 2 and lag 16. In the standardized residual PACF plot, there are significant partial autocorrelation at lag 1 and lag 16. All these evidence violates independence assumption among standardized residuals. Thus, the CDC model is not valid. We should do more investigation on this model.

* 1. (25 points) Perform the first 5 steps of the model building process as given in the course notes. Use the 2000-2007 years’ data only (time points 1 to 96). Do not perform the “forecast error” part of Step #5 here.
* Step 1: Construct plots of x\_t vs. t and the estimated ACF to determine if the time series data is stationary:

> tb <- read.csv(file = "/Users/nannan/Library/CloudStorage/OneDrive-UniversityofNebraska-Lincoln/2022 Spring/STAT-878/project data/tb.csv",header = TRUE)

> tb.2007 <- tb[1:96,]

> x.2007 <- tb.2007$count

> x.s12 <- diff(x=x.2007,lag=12,differences = 1)

> plot(x=x.s12,type = "o",col="red")

> points(x=x.s12,col="blue",pch=20)

A picture containing text, different, various, several

Description automatically generated

After applying a seasonal differencing with S=12, the data doesn’t have the non-stationary in mean problem. The resulted data shows stationary in both mean and variance.

* Step 2: Construct plots of the estimated ACF and PACF of the stationary series

> model.000.010 <- arima(x=x.2007,order = c(0,0,0),seasonal = list(order=c(0,1,0), period=12))

> model.000.010

Call:

arima(x = x.2007, order = c(0, 0, 0), seasonal = list(order = c(0, 1, 0), period = 12))

sigma^2 estimated as 4085: log likelihood = -468.42, aic = 938.84

> par(mfrow=c(1,2))

> acf(x=model.000.010$residuals,lag.max = 100)

> abline(v=c(12,24,36,48,60,72,84,96),lty="dotted")

> axis(side=1,at=c(12,24,36,48,60,72,84,96),tck=-.01)

> pacf(x=model.000.010$residuals,lag.max = 100)

> abline(v=c(12,24,36,48,60,72,84,96),lty="dotted")

> axis(side=1,at=c(12,24,36,48,60,72,84,96),tck=-.01) Chart

Description automatically generatedFrom the ACF plot, lag 12 has a significant autocorrelation, lag 24 has a smaller autocorrelation. This shows a tailing off to 0 trend in terms of autocorrelation at lags that are multiples of 12. The PACF plot shows that lag 12 has a significant partial autocorrelation and lag 24 has a smaller partial autocorrelation. This shows a tailing off to 0 trend in terms of partial autocorrelation at lags that are multiples of 12. Therefore, I pick P=1, Q=1 for seasonal part. The model I will examine is ARIMA .

* Step 3: Find the estimated model using maximum likelihood estimation.

> model.000.111 <- arima(x=x.2007,order = c(0,0,0),seasonal = list(order=c(1,1,1), period=12))

> model.000.111

Call:

arima(x = x.2007, order = c(0, 0, 0), seasonal = list(order = c(1, 1, 1), period = 12))

Coefficients:

sar1 sma1

-0.6549 0.5698

s.e. 0.4149 0.4382

sigma^2 estimated as 4035: log likelihood = -468.02, aic = 942.03

* Step 4: Investigate the diagnostic measures:

> examine.mod(model.000.111,mod.name = "model.000.111",max.lag = 80)

$z

sar1 sma1

-1.578500 1.300352

$p.value

sar1 sma1

0.1144509 0.1934802

Chart, line chart

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

From the above diagnostic plots, the model ARIMA is not a valid model because there’s evidence in the Ljung-Box statistics plot against the independence assumption because after lag 20, almost all Ljung-Box statistics have a significant p-value.

From the standardized ACF and PACF plot, we could see that lag 2 has a marginally significant autocorrelation and it has a tailing off to 0 trend after lag 2. In the PACF plot, lag 2 has a marginally significant partial autocorrelation and it has a tailing off to 0 trend after lag 2. So, I will use p=2, q=2 for the non-seasonal part of the model:

> model.202.111 <- arima(x=x.2007,order = c(2,0,2),seasonal = list(order=c(1,1,1), period=12))

Warning message:

In arima(x = x.2007, order = c(2, 0, 2), seasonal = list(order = c(1, :

possible convergence problem: optim gave code = 1

> model.202.111

Call:

arima(x = x.2007, order = c(2, 0, 2), seasonal = list(order = c(1, 1, 1), period = 12))

Coefficients:

ar1 ar2 ma1 ma2 sar1 sma1

0.1018 0.8979 0.0207 -0.9792 0.2030 -0.6227

s.e. 0.0581 0.0579 0.0794 0.0788 0.2811 0.2725

sigma^2 estimated as 2620: log likelihood = -454.03, aic = 922.06

Diagnostic plots of model ARIMA :

Chart, line chart

Description automatically generated

Graphical user interface, application

Description automatically generated

From the diagnostic plots, there are less than 5% standardized residuals falling outside of the (-2,2) range, so there’s no concern about outlier problem. QQ plot shows that standardized residuals follow closely the normal distribution. Although lag 16 has significant autocorrelation and partial autocorrelation, considering the size is not far away from the critical values, having these two values should be fine. Also, there’s a big improvement in the Ljung-Box statistics plot compared to the ARIMA

Therefore, I will use ARIMA for this dataset.

* 1. (5 points) Perform the forecast error part of Step #5 using the model chosen in part b) and the true future values that are known. Split your analysis into two parts – one for the 2008 data and one for the 2009 data. Include in your answer:
     1. A table showing the forecasts with the future values.

> library(tidyverse)

> # 2008 and 2009 data

> tb2008 <- tb[97:108,]

> tb2009 <- tb[109:120,]

> mod.fit3 <- arima(x = x.2007, order = c(2, 0, 2), seasonal = list(order = c(1, 1, 1), period = 12))

Warning message:

In arima(x = x.2007, order = c(2, 0, 2), seasonal = list(order = c(1, :

possible convergence problem: optim gave code = 1

> mod.fit3

Call:

arima(x = x.2007, order = c(2, 0, 2), seasonal = list(order = c(1, 1, 1), period = 12))

Coefficients:

ar1 ar2 ma1 ma2 sar1 sma1

0.1018 0.8979 0.0207 -0.9792 0.2030 -0.6227

s.e. 0.0581 0.0579 0.0794 0.0788 0.2811 0.2725

sigma^2 estimated as 2620: log likelihood = -454.03, aic = 922.06

> fore.mod <- predict(object = mod.fit3, n.ahead = 24, se.fit = TRUE)

> fore.mod

$pred

Time Series:

Start = 97

End = 120

Frequency = 1

[1] 947.9835 949.6267 1149.6900 1109.8636 1125.5400 1127.7159 1035.4764 1077.8956 918.5288 853.0686 736.0353 459.3078 920.8055 924.5441

[15] 1124.1388 1081.5394 1103.4611 1095.8785 1002.4667 1048.9118 888.0019 831.0029 703.2061 428.9549

$se

Time Series:

Start = 97

End = 120

Frequency = 1

[1] 51.66137 51.97391 52.04263 52.26561 52.29895 52.46107 52.47447 52.59469 52.59747 52.68851 52.68623 52.75670 60.67420 60.94460 60.95330 61.15682

[17] 61.15347 61.31001 61.30217 61.42531 61.41719 61.51629 61.51007 61.59168

Table including true values, forecasted values, 95% CI for 2008:

> data.frame(x.2008,fore.mod$pred[1:12],low.2008,up.2008)

x.2008 fore.mod.pred.1.12. low.2008 up.2008

1 1000 947.9835 846.7291 1049.2379

2 971 949.6267 847.7597 1051.4936

3 1071 1149.6900 1047.6883 1251.6917

4 1159 1109.8636 1007.4249 1212.3023

5 1106 1125.5400 1023.0359 1228.0441

6 1041 1127.7159 1024.8941 1230.5377

7 1129 1035.4764 932.6283 1138.3245

8 994 1077.8956 974.8119 1180.9793

9 929 918.5288 815.4397 1021.6180

10 929 853.0686 749.8010 956.3361

11 700 736.0353 632.7722 839.2984

12 541 459.3078 355.9066 562.7091

Table including true values, forecasted values, 95% CI for 2009:

> data.frame(x.2009,fore.mod$pred[13:24],low.2009,up.2009)

x.2009 fore.mod.pred.13.24. low.2009 up.2009

1 776 920.8055 801.8863 1039.7247

2 888 924.5441 805.0949 1043.9933

3 1006 1124.1388 1004.6726 1243.6051

4 1035 1081.5394 961.6742 1201.4045

5 976 1103.4611 983.6025 1223.3196

6 1012 1095.8785 975.7131 1216.0440

7 965 1002.4667 882.3166 1122.6167

8 906 1048.9118 928.5204 1169.3032

9 818 888.0019 767.6264 1008.3774

10 771 831.0029 710.4332 951.5726

11 688 703.2061 582.6486 823.7637

12 535 428.9549 308.2374 549.6724

* + 1. MSE for 2008 and MSE for 2009

> tb <- read.csv(file = "/Users/nannan/Library/CloudStorage/OneDrive-UniversityofNebraska-Lincoln/2022 Spring/STAT-878/project data/tb.csv",header = TRUE)

> tb.2007 <- tb[1:96,]

> tb.2008 <- tb[97:108,]

> tb.2009 <- tb[109:120,]

> x.2007 <- tb.2007$count

> x.2008 <- tb.2008$count

> x.2009 <- tb.2009$count

> mse.2008 <- sum((fore.mod$pred[1:12]-x.2008)^2)/12

> mse.2008

[1] 4108.584

> mse.2009 <- sum((fore.mod$pred[13:24]-x.2008)^2)/12

> mse.2009

[1] 5262.213

MSE for 2008 is 4108.584. MSE for 2009 is 5262.213

* + 1. Results discussion

For 2008, 95% CI covers all true values. For 2009, 95% CI covers 9 true values.

Based on the MSE values for both years, it turns out that we have higher precision when forecasting values of 2008 than that of 2009. This result is intuitive because we usually have a higher precision for predicting closer values than values further away from the observed values.

* 1. (3 points) Create a plot displaying:
  2. Observed data from 2007-2009, forecasts for 2008-2009, corresponding forecast confidence intervals for 2008-2009

> plot(x = tb$count, ylab = expression(x[t]), xlab = "t", type = "o", col = "red", lwd = 1, pch = 20,

+ main = "Forecasted TB cases for 2008-2009", panel.first=grid(col = "gray", lty = "dotted"), xlim = c(1,120))

> lines(x = c(x.2007 - mod.fit3$residuals, fore.mod$pred), lwd = 1, col = "black", type = "o", pch = 17)

> lines(y = low, x =97:120, lwd = 1, col = "darkgreen", lty = "dashed")

> lines(y = up, x = 97:120, lwd = 1, col = "darkgreen", lty = "dashed")

> legend(locator(1), legend = c("Observed", "Forecast",

+ "95% C.I."), lty = c("solid", "solid", "dashed"),

+ col = c("red", "black", "darkgreen"), pch = c(20,

+ 17, NA), bty = "n")A picture containing text, bunch, different, line

Description automatically generated