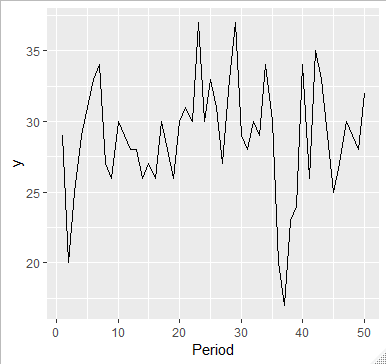
Time Series Homework 5

104304033 統計四 劉書宏

Chapter 5

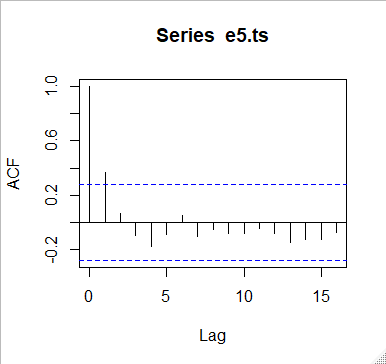
2.

(a)time series plot

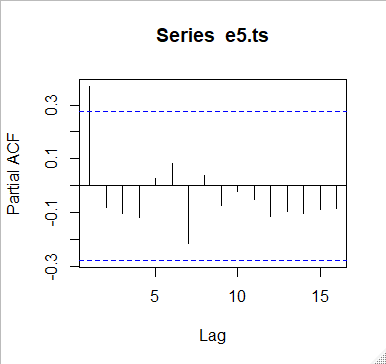


(b)

|  |  |
| --- | --- |
|  | x |
| 1 | 1.000 |
| 2 | 0.369 |
| 3 | 0.067 |
| 4 | -0.092 |
| 5 | -0.178 |
| 6 | -0.088 |
| 7 | 0.051 |
| 8 | -0.105 |
| 9 | -0.050 |
| 10 | -0.083 |
| 11 | -0.078 |
| 12 | -0.042 |
| 13 | -0.083 |
| 14 | -0.149 |
| 15 | -0.125 |
| 16 | -0.127 |
| 17 | -0.075 |



yes, there is significant autocorrelation in this time series, so we need to check pacf.



(c)

Series: e5.ts

ARIMA(1,0,0) with non-zero mean

Coefficients:

ar1 mean

0.3662 28.8970

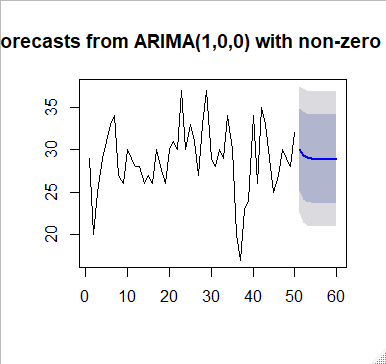
s.e. 0.1304 0.8175

sigma^2 estimated as 14.3: log likelihood=-136.5

AIC=279.01 AICc=279.53 BIC=284.75

I used auto.arima to get the most suitable model with order = c(1,0,0).

(d)



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | x |  |  |  |  |
| 1 | 0.096 | 19 | -2.569 | 37 | -8.639 |
| 2 | -8.935 | 20 | 2.164 | 38 | -1.541 |
| 3 | -0.639 | 21 | 1.699 | 39 | -2.738 |
| 4 | 1.530 | 22 | 0.333 | 40 | 6.896 |
| 5 | 2.065 | 23 | 7.699 | 41 | -4.766 |
| 6 | 3.333 | 24 | -1.864 | 42 | 7.164 |
| 7 | 3.601 | 25 | 3.699 | 43 | 1.868 |
| 8 | -3.766 | 26 | 0.601 | 44 | -1.399 |
| 9 | -2.202 | 27 | -2.667 | 45 | -3.935 |
| 10 | 2.164 | 28 | 4.798 | 46 | -0.470 |
| 11 | -0.301 | 29 | 6.601 | 47 | 1.798 |
| 12 | -0.935 | 30 | -2.864 | 48 | -0.301 |
| 13 | -0.569 | 31 | -0.935 | 49 | -0.935 |
| 14 | -2.569 | 32 | 1.431 | 50 | 3.431 |
| 15 | -0.836 | 33 | -0.301 |  |  |
| 16 | -2.202 | 34 | 5.065 |  |  |
| 17 | 2.164 | 35 | -0.766 |  |  |
| 18 | -1.301 | 36 | -9.301 |  |  |

5.

y(t) = 150 – 0.5 \* y(t-1) + e(t)

(a)

MA(1)

This is not a stationary time series process.

(b)

The mean of this time series is 150.

(c)

y(101) = 107.5 + e(101), so it may be larger than mean.

8.

MA(1) and AR(2)

y(t) = 50 + 0.8 \* y(t-1) + e(t) – 0.2 \* e(t-1)

(a)

This is not a stationary time series process.

(b)

The mean of this time series is 50.

(c)

y(101) = 266 + e(101) – 0.2 \* e(100), so it may be larger than mean.

11.

B.2

(a)

Series: b.2.ts

ARIMA(0,0,0) with non-zero mean

Coefficients:

mean

10374.9618

s.e. 20.8338

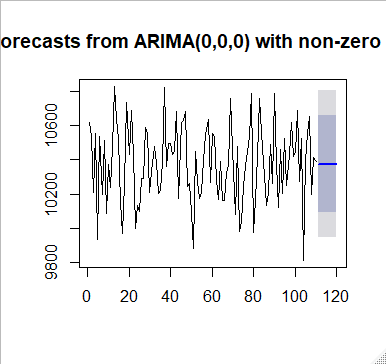
sigma^2 estimated as 48183: log likelihood=-748.63

AIC=1501.27 AICc=1501.38 BIC=1506.67

we can see that by auto.arima, the order = c(0,0,0) with zero mean in white noise, the errors are uncorrelated.

So, this model will predict 10 same points with mean = 10374.9618.

(b)



(c)

Comparing simple exponential smoothing with lambda = 0.1, I prefer simple exponential smoothing method, because the arima predict the last 10 points are all the same. That is not gonna happen.

(d)

Because of the same predict value, the confidence intervals are the same by this table.

|  |  |  |
| --- | --- | --- |
| Point.Forecast | Lo.95 | Hi.95 |
| 10374.96 | 9944.74 | 10805.19 |

26.

(a)

Series: b11.ts

ARIMA(0,1,0)

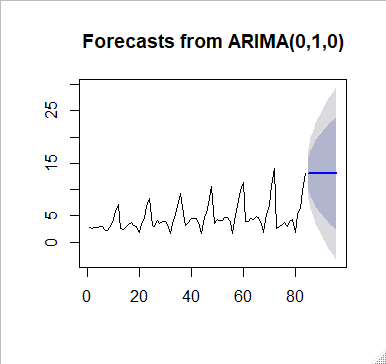
sigma^2 estimated as 5.912: log likelihood=-191.52

AIC=385.04 AICc=385.09 BIC=387.46

(b)

we can see that the arima model with (0,1,0), that means the model is just a random walk. And, if we take the first difference, then the model will become arima model with (0,0,0). Just a white noise. so the predict values are still all the same.

(c)



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | x |  |  |  |  |  |  |
| 1 | 0.002851 | 22 | 0.879 | 43 | -0.876 | 64 | -0.234 |
| 2 | -0.179 | 23 | 2.364 | 44 | -2.02 | 65 | 0.692 |
| 3 | 0.083 | 24 | 1.519 | 45 | 3.096 | 66 | -0.291 |
| 4 | -0.034 | 25 | -5.244 | 46 | 0.689 | 67 | -1.154 |
| 5 | 0.225 | 26 | -0.107 | 47 | 2.886 | 68 | -1.702 |
| 6 | 0.09 | 27 | 1.041 | 48 | 2.337 | 69 | 3.401 |
| 7 | -0.754 | 28 | -0.524 | 49 | -7.018 | 70 | 1.651 |
| 8 | -0.07 | 29 | 0.414 | 50 | 0.659 | 71 | 3.93 |
| 9 | 0.71 | 30 | 0.049 | 51 | -0.138 | 72 | 3.113 |
| 10 | 1.379 | 31 | -0.726 | 52 | -0.033 | 73 | -11.277 |
| 11 | 1.463 | 32 | -1.687 | 53 | 0.526 | 74 | 0.26 |
| 12 | 1.368 | 33 | 1.955 | 54 | 0.106 | 75 | 0.471 |
| 13 | -4.591 | 34 | 1.683 | 55 | -0.788 | 76 | 0.37 |
| 14 | -0.066 | 35 | 2.403 | 56 | -2.242 | 77 | -0.813 |
| 15 | 0.556 | 36 | 1.64 | 57 | 3.325 | 78 | 1.059 |
| 16 | 0.235 | 37 | -3.879 | 58 | 1.874 | 79 | 0.231 |
| 17 | 0.51 | 38 | -2.287 | 59 | 2.936 | 80 | -2.479 |
| 18 | -0.546 | 39 | 0.63 | 60 | 1.473 | 81 | 3.483 |
| 19 | -0.202 | 40 | 0.796 | 61 | -7.315 | 82 | 1.203 |
| 20 | -1.269 | 41 | 0.006 | 62 | -0.059 | 83 | 3.418 |
| 21 | 1.836 | 42 | 0.019 | 63 | 0.553 | 84 | 3.234 |

27.

(a)

(b)

I prefer the Winter’s method, because the predict values with arima model are all the same, that is unreal.

R code

### chapter 5

library(ggplot2)

library(forecast)

library(tseries)

## 2

#a

e5.1 = read.csv("E5.1.csv")

ggplot(e5.1, aes(period, yt)) + geom\_line() +xlab("Period") + ylab("y")

#b

e5.ts = ts(e5.1$yt)

e5.acf = acf(e5.ts)

write.csv(e5.acf$acf, file = "5.2b.csv")

e5.pacf = pacf(e5.ts)

#c

model = auto.arima(e5.ts, seasonal = FALSE)

model

forecast10 = forecast(model, h = 10)

write.csv(forecast10$residuals, file = "5.2c.csv")

plot(forecast(model, h = 10))

## 11

#a

b.2 = read.csv("b.2.csv")

b.2.ts = b.2$Sales..in.Thousands[1:110]

model = auto.arima(b.2.ts)

model

#b

forecast(model, h = 10)

plot(forecast(model, h = 10))

write.csv(forecast(model, h = 10), file = "5.11b.csv")

##26

B11 = read.csv("B.11.csv")

#a

b11.ts = ts(B11$Sales..in.Thousands.of.Bottles[1:84])

model = auto.arima(b11.ts)

model

#b

forecast12 = forecast(model, h = 12)

plot(forecast12)

#c

write.csv(forecast12$residuals, file = "5.26b.csv")