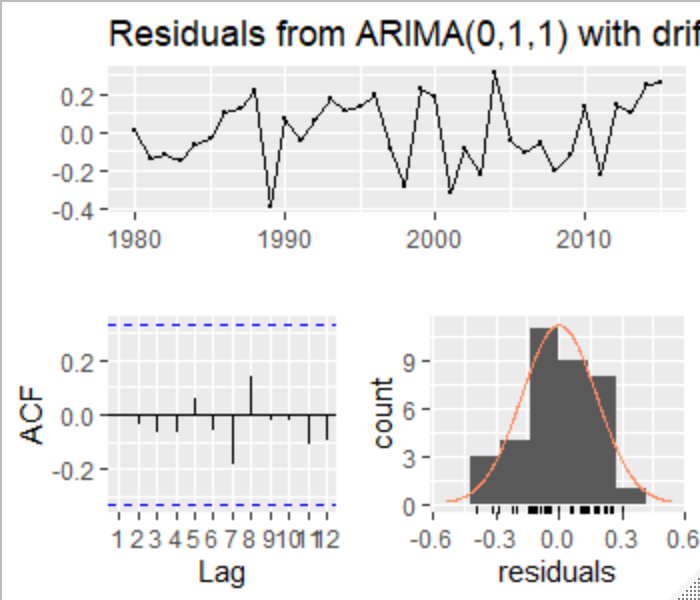
**Exercise 8.8 and 8.9**

**Exercise 8.8**

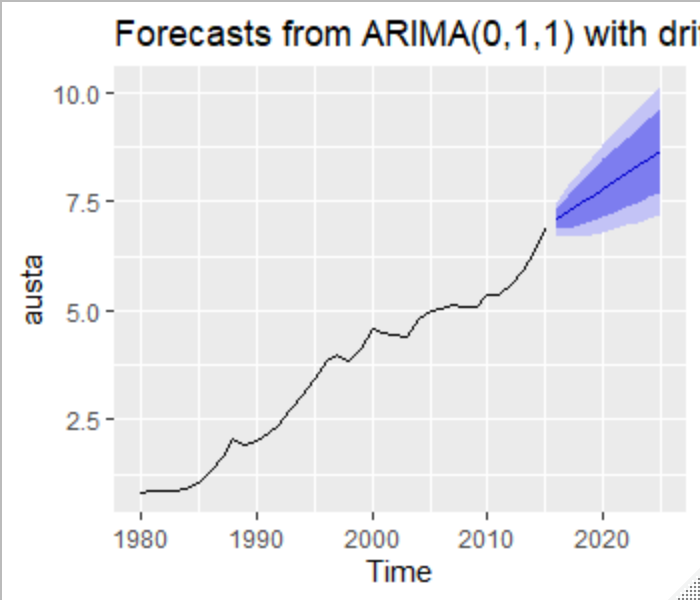
Consider austa, the total international visitors to Australia (in millions) for the period 1980-2015.

1. Use auto.arima() to find an appropriate ARIMA model. What model was selected. Check that the residuals look like white noise. Plot forecasts for the next 10 periods.

The model that was selected was ARIMA(0,1,1) with drift. The residuals look pretty good: it looks like white noise, the ACF is well below threshold, and thee residuals are roughly normally distributed

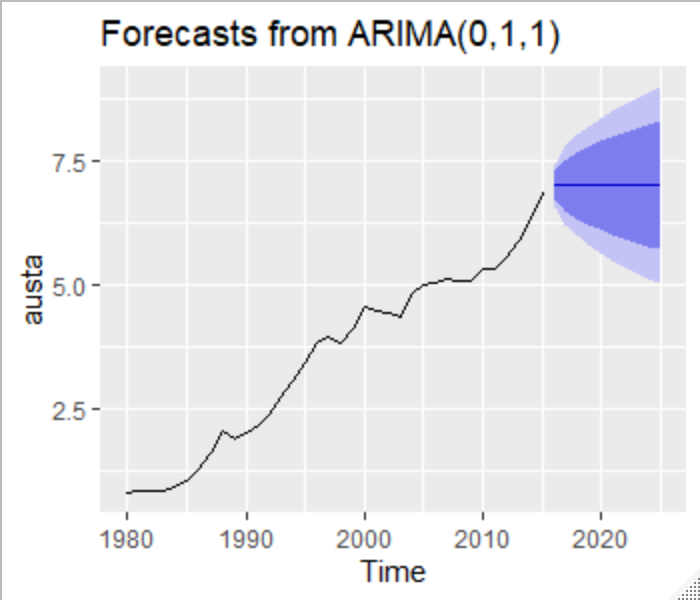


The forecast for the next 10 periods

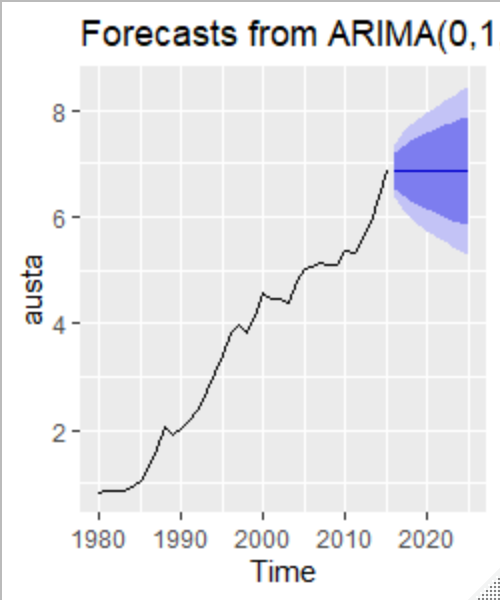


1. Plot forecasts from an ARIMA(0,1,1) model with no drift and compare these to part a. Remove the MA term and plot again.

The ARIMA(0,1,1) forecast model with no drift. Without the drift the forecast just level off and continues on compared to the previous model



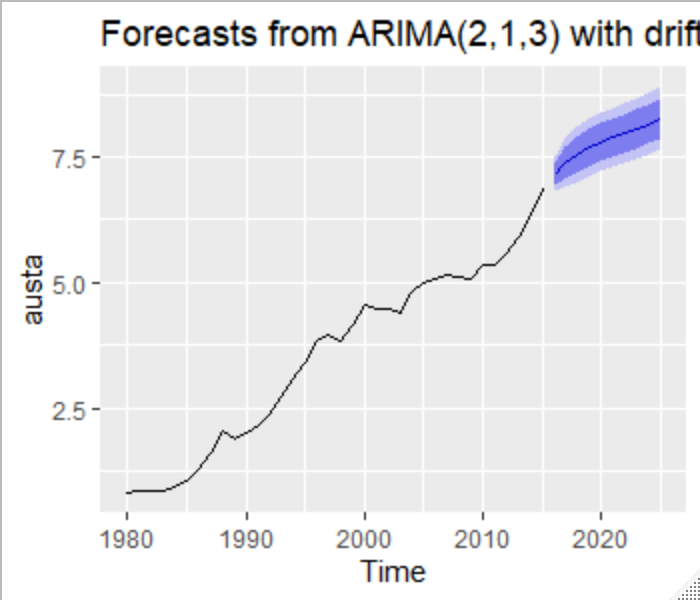
ARIMA(0,1,0) model



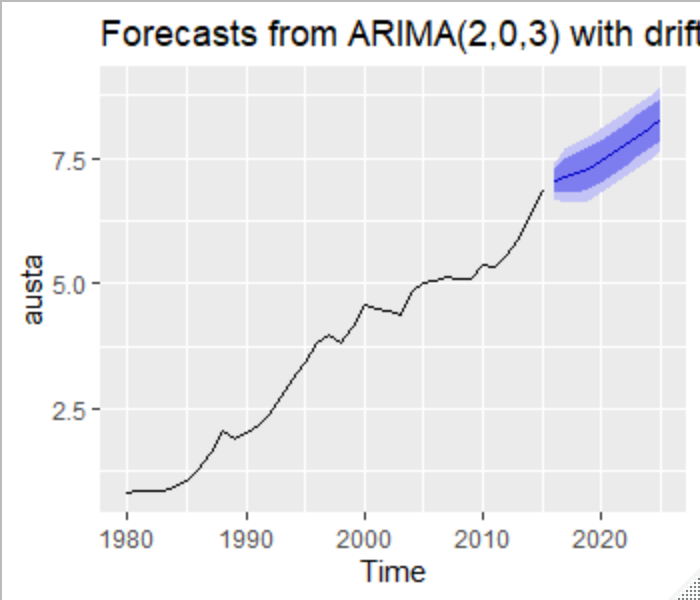
This forecast look to be a smidge wider than the model with an MA component.

1. Plot forecasts from an ARIMA(2,1,3) model with drift. Remove the constant and see what happens.

ARIMA(2,1,3) model with drift

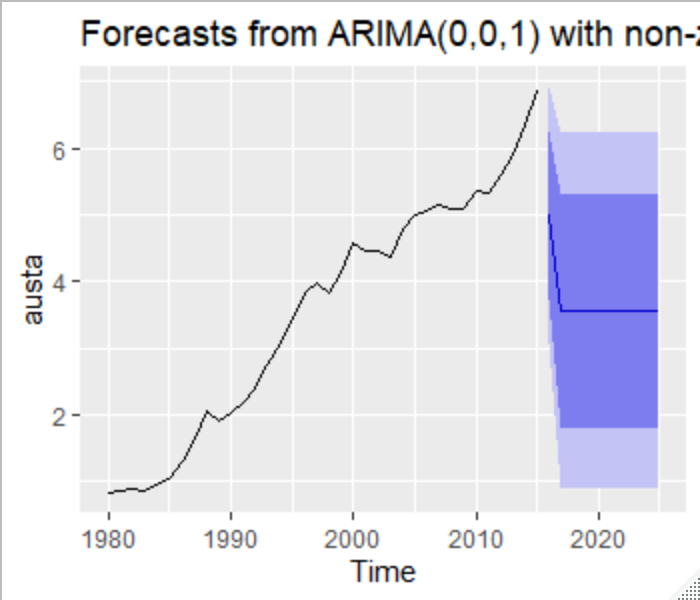


ARIMA(2,0,3) model with drift

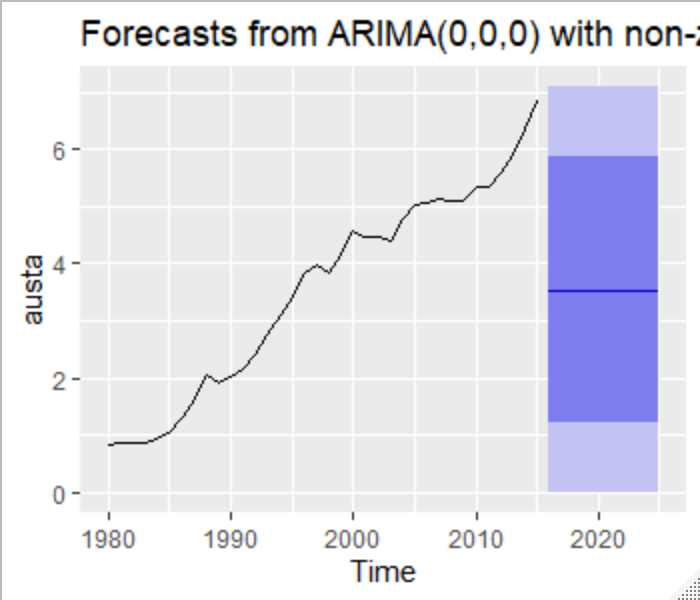


1. Plot forecasts from an ARIMA(0,0,1) model with a constant. Remove the MA term and plot again.

ARIMA(0,0,1) model with constant

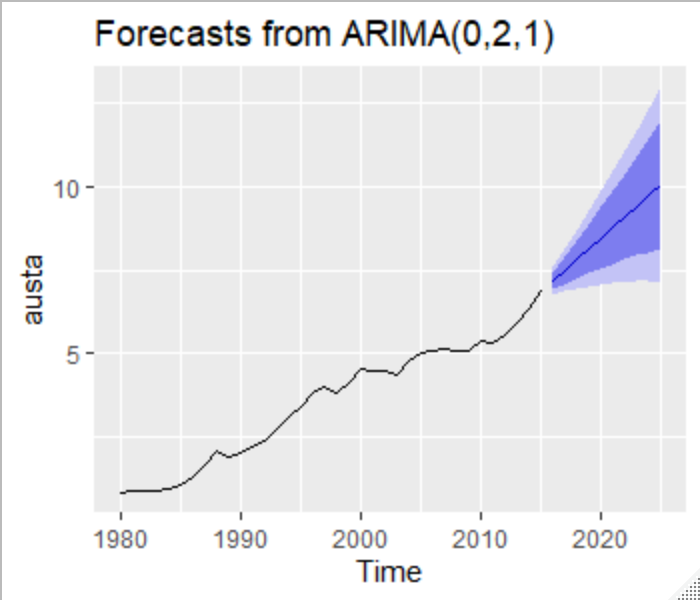


ARIMA(0,0,0) mode with constant



1. Plot forecasts from an ARIMA(0,2,1) model with no constant.

ARIMA(0,2,1) model with no constant



**Exercise 8.9**

For the usgdp series:

1. if necessary, find a suitable Box-Cox transformation for the data;

I determined that it was necessary to transform the data. I utilized the ur.kpss() function of the urca package to determine the best route. I landed on the following

diff(diff(log(usgdp),12),1) %>% ur.kpss() %>% summary()

1. fit a suitable ARIMA model to the transformed data using auto.arima();

Using the auto.arima() model it determined that an ARIMA(1,0,2)(2,0,2)[4] with a non-zero mean was the best model

1. try some other plausible models by experimenting with the orders chosen;

I tried 4 other methods that were plausible in my mind and here they are:

fit <- Arima(usgdp\_transform, order=c(1,0,2), seasonal = c(2,0,4))

checkresiduals(fit)

fit2 <- Arima(usgdp\_transform, order=c(2,0,2), seasonal = c(3,0,4))

checkresiduals(fit2)

fit3 <- Arima(usgdp\_transform, order=c(1,1,2), seasonal = c(2,1,4))

checkresiduals(fit3)

fit4 <- Arima(usgdp\_transform, order=c(1,1,1), seasonal = c(2,1,3))

checkresiduals(fit4)

fit5 <- Arima(usgdp\_transform, order=c(2,1,3), seasonal = c(1,0,3))

checkresiduals(fit5)

fit %>% forecast(h=24) %>% autoplot()

fit2 %>% forecast(h=24) %>% autoplot()

fit3 %>% forecast(h=24) %>% autoplot()

fit4 %>% forecast(h=24) %>% autoplot()

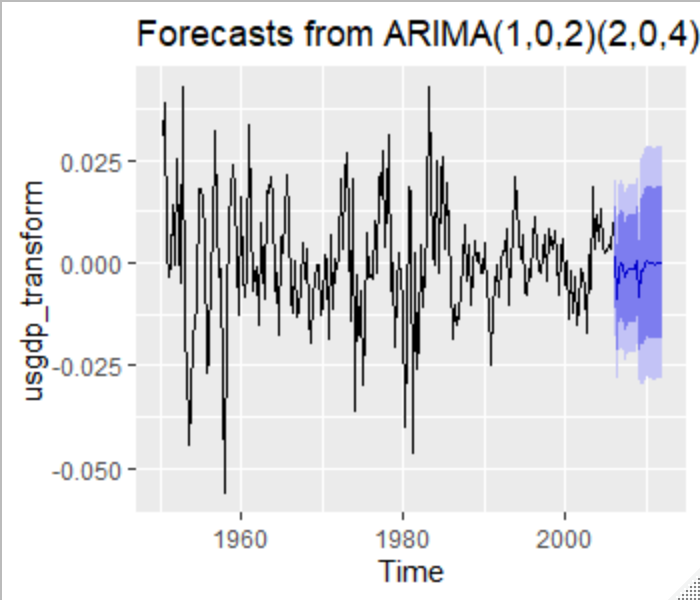
fit5 %>% forecast(h=24) %>% autoplot()

1. choose what you think is the best model and check the residual diagnostics;

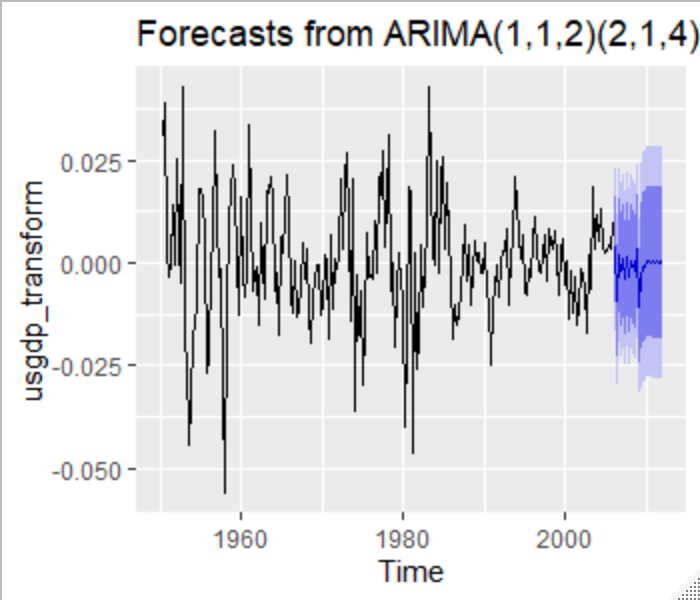
I chose fit3 to compare to the auto.arima() model which was ARIMA(1,1,2)(2,1,4)[4]

1. produce forecasts of your fitted model. Do the forecasts look reasonable?

These are the forecasts from the auto.arima() model



And this is the forecast from the fit that I chose



1. compare the results with what you would obtain using ets() (with no transformation).

For this I didn’t really understand what I was supposed to look at since I am comparing it to transformed graphs

