ARIMA Models

Subhalaxmi Rout

06/14/2021

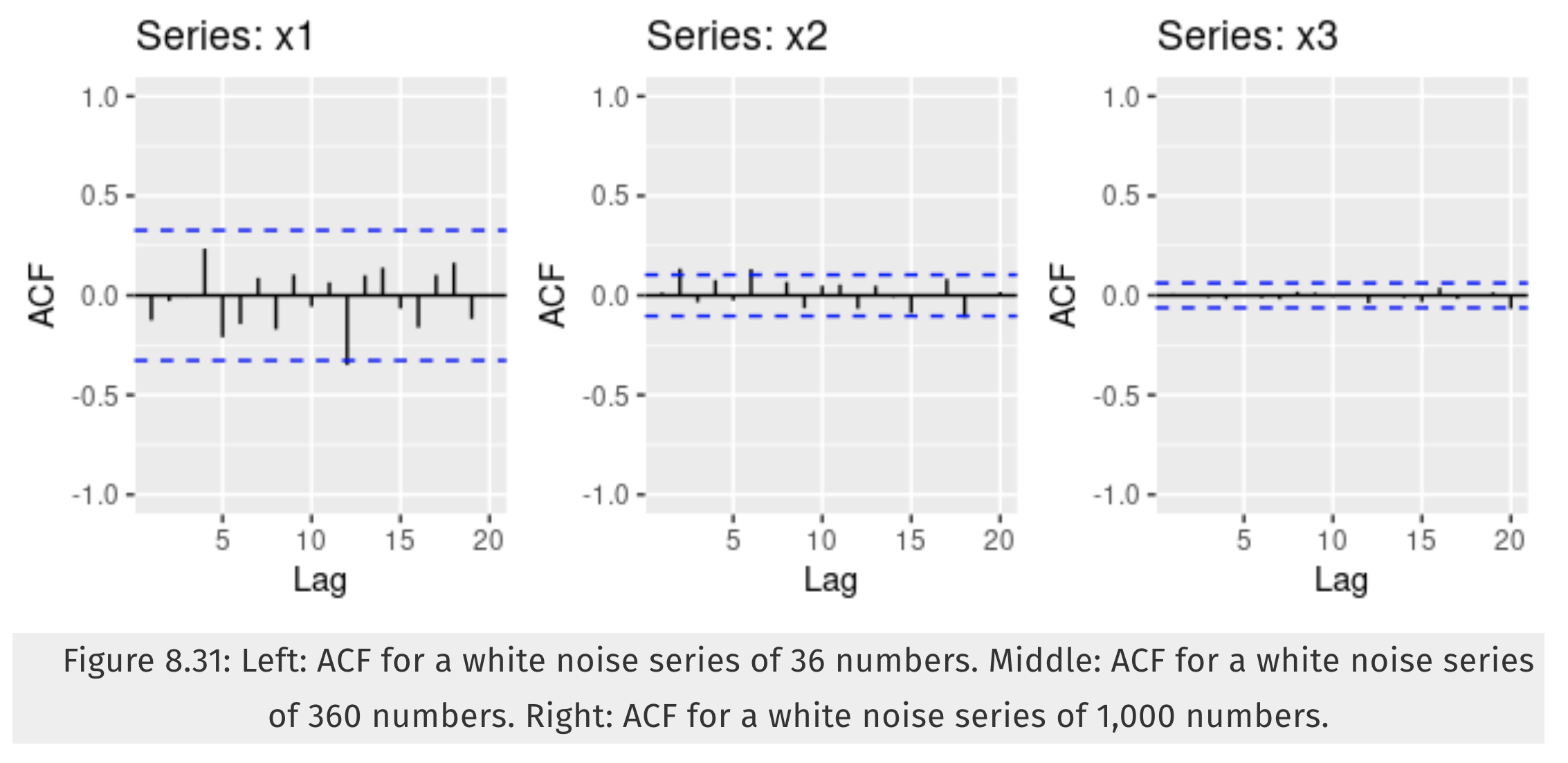
library(knitr)  
library(fpp2)  
library(tseries)  
library(gridExtra)

### Forecasting: Principles and Practices (Chapter 8 - ARIMA Models)

#### 8.1

**Figure 8.3 shows the ACFs for 36 random numbers, 360 random numbers and 1,000 random numbers.**

knitr::include\_graphics("https://raw.githubusercontent.com/SubhalaxmiRout002/DATA624/main/Week3/Screen%20Shot%202021-06-17%20at%207.34.43%20PM.png")



**Explain the differences among these figures. Do they all indicate that the data are white noise?**

The sample size increases from 36 to 360 to 1,000, the autocorrelations appear to decrease and the critical values appear to shrink. Series X1 and X3 looks white noise series. Series X2 autocorrelations at lag 2 and lag 6 are statistically significant, so series X2 is not white noise serise.

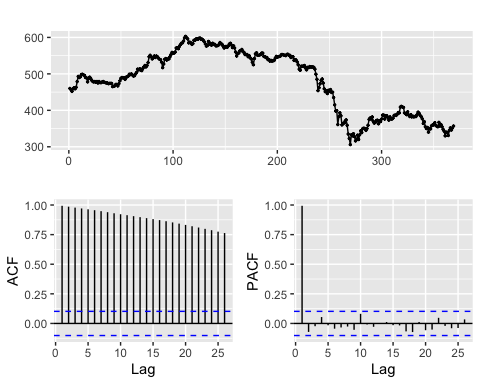
**Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?**

The critical values are at different distances from the mean of zero since critical values for white noise are supposed to lie within where T is length of time series. In this case, as T gets bigger (from 36 to 360 to 1,000), range gets smaller. So the autocorrelations different in each figure.

#### 8.2

**A classic example of a non-stationary series is the daily closing IBM stock price series (data set ibmclose). Use R to plot the daily closing prices for IBM stock and the ACF and PACF. Explain how each plot shows that the series is non-stationary and should be differenced.**

ggtsdisplay(ibmclose)



For data to be stationary, the statistical properties of a system do not change over time, this means, variances and covariances should be the same over the period of time. The plot does not show the mean to remain same over the time. Looking at the ACF plot, the critical values decrease slowly. PACF shows that no autocorrelations after lag 1 are statistically significant. All of this suggest that data is non-stationary.

We can check the stationarity by performing Augmented Dickey-Fuller Test. If the p-value less than 0.05, then series is stationary or else non-stationary.

tseries::adf.test(ibmclose)

##   
## Augmented Dickey-Fuller Test  
##   
## data: ibmclose  
## Dickey-Fuller = -1.7063, Lag order = 7, p-value = 0.701  
## alternative hypothesis: stationary

p-value is > 0.05, so the series is non-stationary.

To get stationary data, IBM stock data need differencing.

#### 8.6

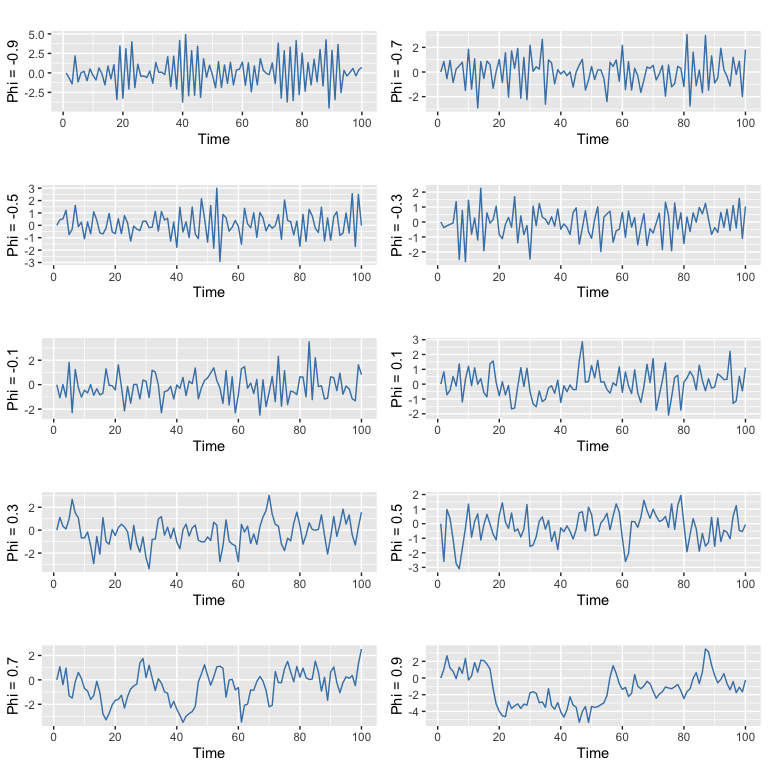
Use R to simulate and plot some data from simple ARIMA models.  **Use the following R code to generate data from an AR(1) model with and .The process starts with .**

The AR(1) model with the parameter () =

y <- ts(numeric(100))  
e <- rnorm(100)  
for(i in 2:100)  
 y[i] <- 0.6\*y[i-1] + e[i]

**Produce a time plot for the series. How does the plot change as you change ?**

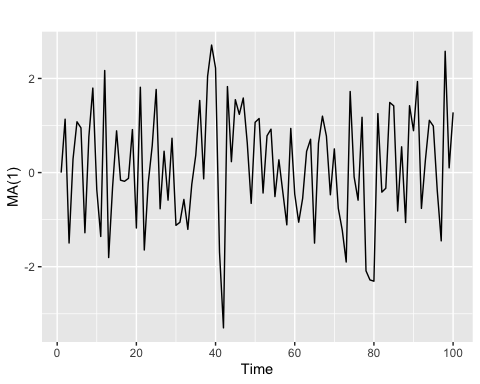
phi\_change <- function(phi){  
 y <- ts(numeric(100))  
 e <- rnorm(100)  
 for(i in 2:100){  
 y[i] <- phi\*y[i-1] + e[i]  
 }  
 return(y)  
}  
  
p1<- autoplot(phi\_change(-0.9), col = "steelblue") + ylab("Phi = -0.9")  
p2 <- autoplot(phi\_change(-0.7), col = "steelblue") + ylab("Phi = -0.7")  
p3 <- autoplot(phi\_change(-0.5), col = "steelblue") + ylab("Phi = -0.5")  
p4 <- autoplot(phi\_change(-0.3), col = "steelblue") + ylab("Phi = -0.3")  
p5<- autoplot(phi\_change(-0.1), col = "steelblue") + ylab("Phi = -0.1")  
p6 <- autoplot(phi\_change(0.1), col = "steelblue") + ylab("Phi = 0.1")  
p7 <- autoplot(phi\_change(0.3), col = "steelblue") + ylab("Phi = 0.3")  
p8 <- autoplot(phi\_change(0.5), col = "steelblue") + ylab("Phi = 0.5")  
p9 <- autoplot(phi\_change(0.7), col = "steelblue") + ylab("Phi = 0.7")  
p10 <- autoplot(phi\_change(0.9), col = "steelblue") + ylab("Phi = 0.9")  
  
gridExtra::grid.arrange(p1, p2, p3,   
 p4, p5, p6,  
 p7, p8, p9, p10,  
 nrow = 5)



As phi increase variation of y also increases.

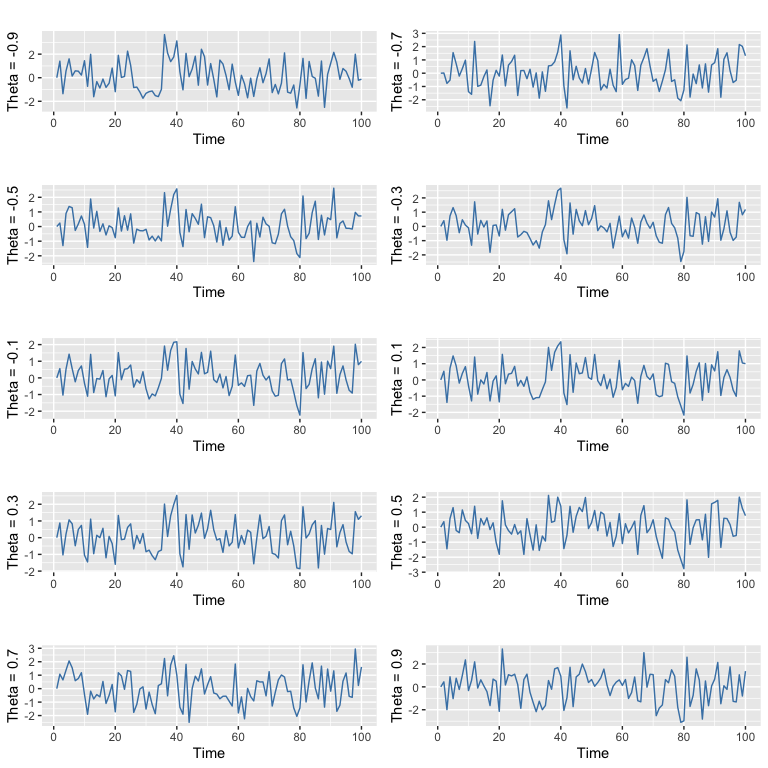
**Write your own code to generate data from an MA(1) model with and .**

ma1 <- function(theta){  
 y\_ma <- ts(numeric(100))  
 e\_ma <- rnorm(100)  
 for(i in 2:100){  
 y\_ma[i] <- theta\*e\_ma[i-1] + e[i]  
 }  
 return(y\_ma)  
}  
  
autoplot(ma1(0.6)) + ylab("MA(1)")



Produce a time plot for the series. How does the plot change as you change .

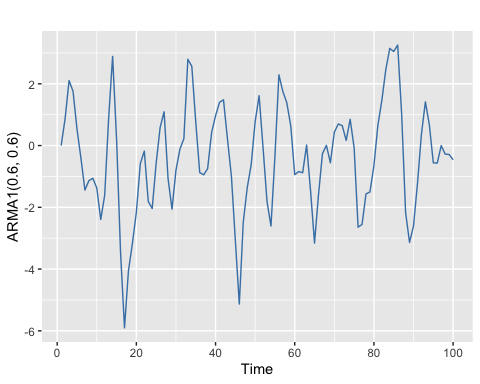
p1<- autoplot(ma1(-0.9), col = "steelblue") + ylab("Theta = -0.9")  
p2 <- autoplot(ma1(-0.7), col = "steelblue") + ylab("Theta = -0.7")  
p3 <- autoplot(ma1(-0.5), col = "steelblue") + ylab("Theta = -0.5")  
p4 <- autoplot(ma1(-0.3), col = "steelblue") + ylab("Theta = -0.3")  
p5<- autoplot(ma1(-0.1), col = "steelblue") + ylab("Theta = -0.1")  
p6 <- autoplot(ma1(0.1), col = "steelblue") + ylab("Theta = 0.1")  
p7 <- autoplot(ma1(0.3), col = "steelblue") + ylab("Theta = 0.3")  
p8 <- autoplot(ma1(0.5), col = "steelblue") + ylab("Theta = 0.5")  
p9 <- autoplot(ma1(0.7), col = "steelblue") + ylab("Theta = 0.7")  
p10 <- autoplot(ma1(0.9), col = "steelblue") + ylab("Theta = 0.9")  
  
gridExtra::grid.arrange(p1, p2, p3,   
 p4, p5, p6,  
 p7, p8, p9, p10,  
 nrow = 5)



The variation occures as the changes of Theta but the variations are not subtle.

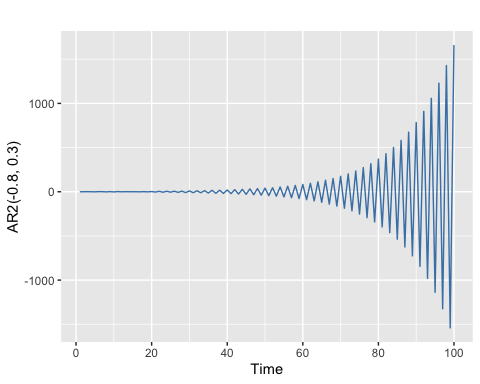
**Generate data from an ARMA(1,1) model with , = 0.6 and .**

ARMA1 <- function( phi1, theta1) {  
 y <- ts(numeric(100))  
 e <- rnorm(100)  
   
 for (i in 2:100) {  
   
 y[i] <- phi1 \* y[i -1] + theta1 \* e[i -1] + e[i]  
   
 }  
 return(y)  
}   
  
autoplot(ARMA1(0.6, 0.6), col = "steelblue")



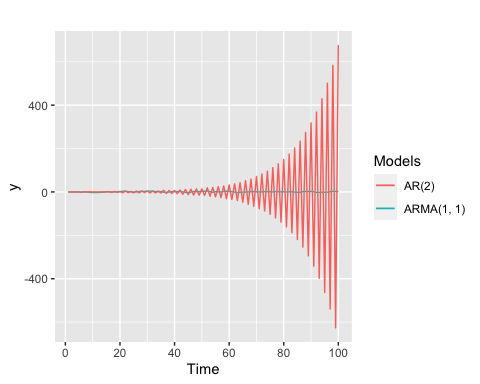
Generate data from an AR(2) model with , and .(Note that these parameters will give a non-stationary series.)

AR2 <- function(phi1, phi2){  
 y <- ts(numeric(100))  
 e <- rnorm(100)  
 for (i in 3:100){  
 y[i] <- phi1 \* y[i - 1] + phi2 \* y[i - 2] + e[i]  
 }  
 return(y)  
}  
autoplot(AR2(-0.8, 0.3), col = "steelblue")



Graph the latter two series and compare them.

autoplot(ARMA1(0.6,0.6), series = "ARMA(1, 1)") +  
 autolayer(AR2(-0.8, 0.3), series = "AR(2)") +  
 ylab("y") +  
 guides(colour = guide\_legend(title = "Models"))



AR(2) model has increased oscillation.

#### 8.8

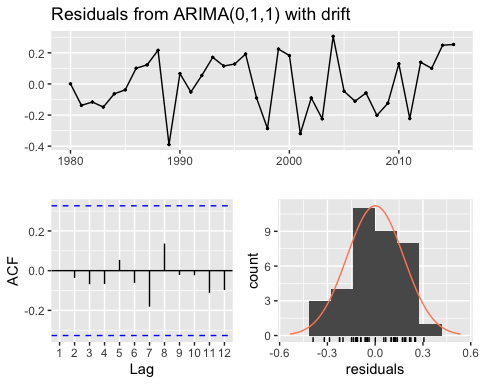
Consider austa, the total international visitors to Australia (in millions) for the period 1980-2015.

**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.**

auto\_fit <- auto.arima(austa)  
auto\_fit

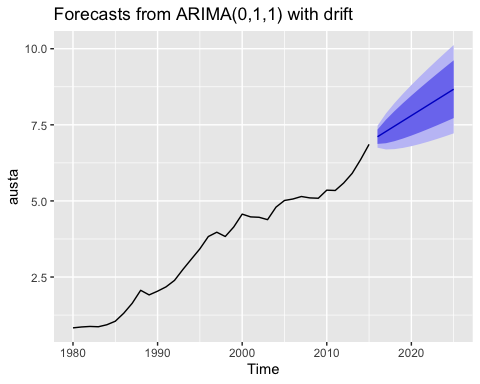
## Series: austa   
## ARIMA(0,1,1) with drift   
##   
## Coefficients:  
## ma1 drift  
## 0.3006 0.1735  
## s.e. 0.1647 0.0390  
##   
## sigma^2 estimated as 0.03376: log likelihood=10.62  
## AIC=-15.24 AICc=-14.46 BIC=-10.57

checkresiduals(auto\_fit)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1) with drift  
## Q\* = 2.297, df = 5, p-value = 0.8067  
##   
## Model df: 2. Total lags used: 7

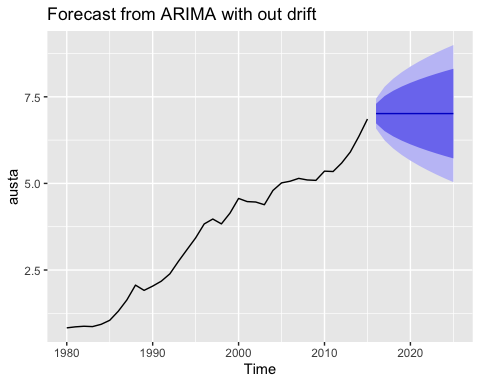
autoplot(forecast(auto\_fit, h = 10))



* Selected model : ARIMA(0,1,1)
* Yes, residuals look like white noise.
* Above plot shows the forecast for next 10 years.

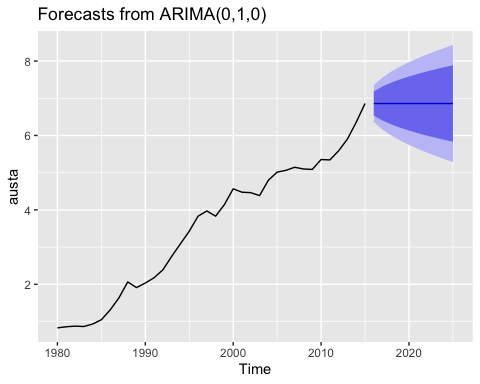
**Plot forecasts from an ARIMA(0,1,1) model with no drift and compare these to part a. Remove the MA term and plot again.**

fit <- Arima(austa, order = c(0,1,1), include.drift = FALSE)  
arima\_wo\_dft <- autoplot(forecast(fit, h = 10)) + ggtitle("Forecast from ARIMA with out drift")  
arima\_wo\_dft



Forecast without drift has broader confidence interval and trend is not upward.

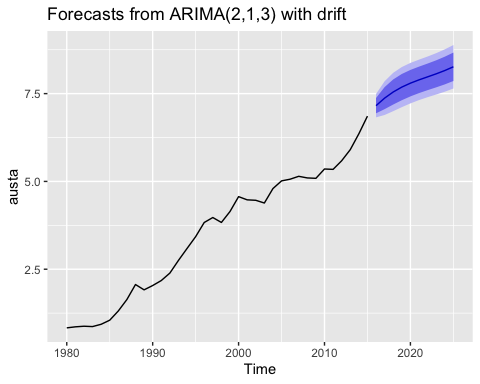
fit <- Arima(austa, order = c(0,1,0), include.drift = FALSE)  
p\_wo\_ma <- autoplot(forecast(fit, h = 10))  
p\_wo\_ma



The forecast without MA, increasing Trend does not show in the forecast.

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

fit <- Arima(austa, order = c(2,1,3), include.drift = TRUE)  
autoplot(forecast(fit, h = 10))

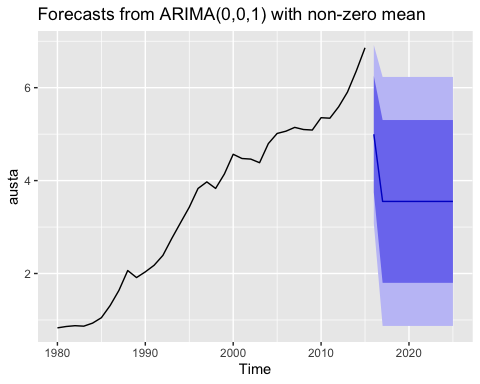


The forecast shows the upward trend.

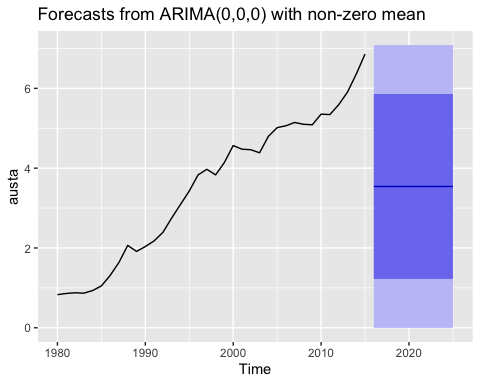
# Need to work on   
  
#fit\_2 <- Arima(austa, order = c(2,1,3), include.drift = TRUE, include.constant = FALSE)  
#fit\_2  
#autoplot(forecast(fit\_2, h = 10))

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

fit\_4 <- Arima(austa, order = c(0,0,1), include.constant = TRUE)  
autoplot(forecast(fit\_4, h = 10))



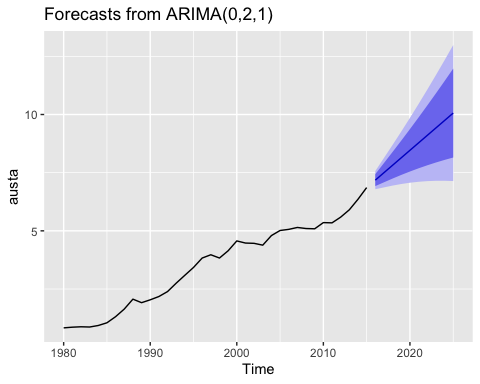
fit\_5 <- Arima(austa, order = c(0,0,0), include.constant = TRUE)  
autoplot(forecast(fit\_5, h = 10))



All the forecast are the mean of the data history,

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

fit\_6 <- Arima(austa, order = c(0,2,1), include.constant = FALSE)  
autoplot(forecast(fit\_6, h = 10))



The forecast showing upward trend.