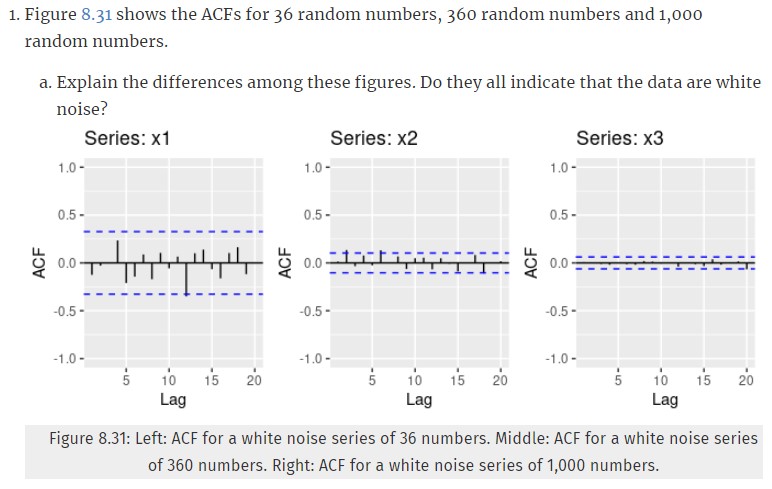
HW3 Arima Models

Group 4 - Subhalaxmi Rout, Kenan Sooklall, Devin Teran, Christian Thieme, Leo Yi

6/16/2021

## Forecasting: Principles and Practice

#### 8.1



The main difference between these figures are that the scale of both the ACF values and the 95% limit lines seems to be decreasing from left to right. Additionally, it appears that some of the patterns change from chart to chart with positive and negative ACF. It does appear that these charts indicate that the data are white noise as there is no identifiable pattern and the spikes don’t exceed the bounds more than 5% of the time.

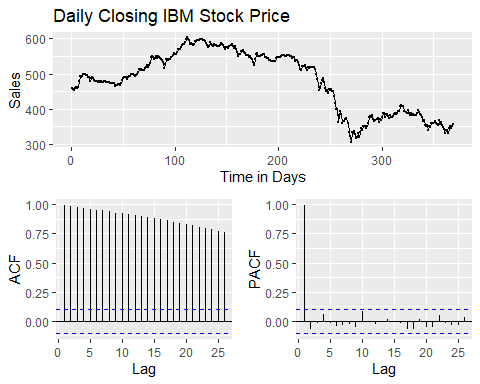
**Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?**

The critical values are at different distances because of the way those lines are calculated. They are calculated as where is the length of time in the series. Naturally, as T increases, that critical value will get smaller and smaller around the mean zero. As to why the autocorrelations are different in each figure, as part of the autocorrelation formula, we are using the mean of the series. The mean will change with the addition of samples and as such will change the linear relationship between the lagged values.

#### 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.**

ibmclose %>%  
ggtsdisplay(main = "Daily Closing IBM Stock Price", ylab = "Sales", xlab = "Time in Days")



Based on the plots above, the data is clearly non-stationary and should be differenced. We can tell in the first line plot that there are ups and downs and that each point on the graph relates to the point before it. When looking at the ACF, we can see that the plot decreases gradually, which is an indicator that differencing is needed because each lag is highly correlated and above the 95% bound indicating that the autocorrelation is significant. The first lag of the PACF plot here is almost one where the other lags are almost 0. This indicates that the data is non-stationary since the first lag is so large.

#### 8.6

**Use R to simulate and plot some data from simple ARIMA models.**

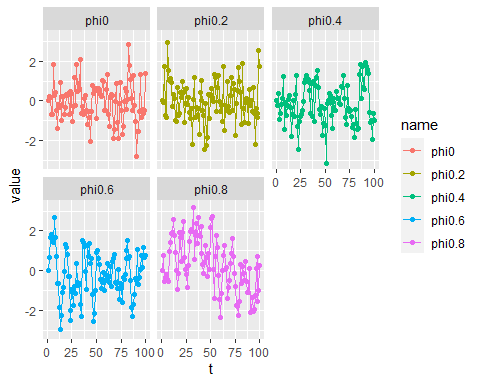
**Use the following R code to generate data from an AR(1) model with and .The process starts with .**

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 ?**

ar1 <- function(phi) {  
 y <- ts(numeric(100))  
 e <- rnorm(100)  
 for(i in 2:100) {  
 y[i] <- phi\*y[i-1] + e[i]  
 }  
 return (y)  
   
}

df <- data.frame(phi0=ar1(0.1), phi0.2=ar1(0.2), phi0.4=ar1(0.4), phi0.6=ar1(0.6), phi0.8=ar1(0.8)) %>% mutate(t=row\_number())  
df %>% pivot\_longer(cols = -c('t')) %>% ggplot(aes(x=t, y=value, color=name)) + geom\_line() + geom\_point() + facet\_wrap(~name, )



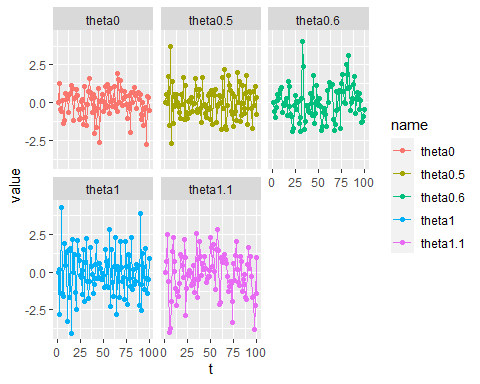
It appears that in most cases as gets larger, the variation of also increases.

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

ma1 <- function(phi) {  
 y <- ts(numeric(100))  
 e <- rnorm(100)  
 for(i in 2:100) {  
 y[i] <- phi\*e[i-1] + e[i]  
 }  
 return (y)  
}

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

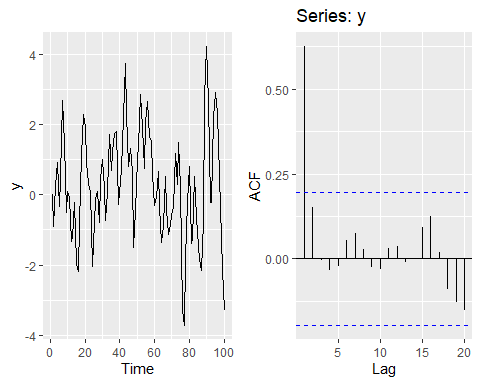
df <- data.frame(theta1=ma1(-1), theta0.5=ma1(-0.5), theta0=ma1(0), theta0.6=ma1(0.6), theta1=ma1(1)) %>% mutate(t=row\_number())  
df %>% pivot\_longer(cols = -c('t')) %>% ggplot(aes(x=t, y=value, color=name)) + geom\_line() + geom\_point() + facet\_wrap(~name, scales = 'fixed')



Similar to our charts above, it appears that as you increase theta, the variations on the increase as well.

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

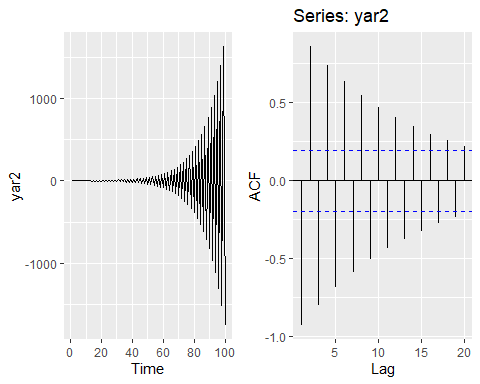
ma\_ar11 <- function(phi, theta) {  
 y <- ts(numeric(100))  
 e <- rnorm(100)  
 for(i in 2:100) {  
 y[i] = phi\*y[i-1] + theta\*e[i-1] + e[i]  
 }  
 return (y)  
}  
y = ma\_ar11(0.6, 0.6)  
p1 = autoplot(y)  
p2 = ggAcf(y)  
gridExtra::grid.arrange(p1, p2, ncol = 2)



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

ar2 <- function(phi\_1, phi\_2) {  
 y <- ts(numeric(100))  
 e <- rnorm(100, 1)  
 for(i in 3:100) {  
 y[i] <- phi\_1\*y[i-1] + phi\_2\*y[i-2] + e[i]  
 }  
 return (y)  
}  
yar2 = ar2(-0.8, 0.3)

p1 = autoplot(yar2)  
p2 = ggAcf(yar2)  
gridExtra::grid.arrange(p1, p2, ncol = 2)



**Graph the latter two series and compare them.**

The graphs are plotted in the charts above. The ARMA(1,1) model with , and , looks to be much more random in nature, however, the ACF plot does indicate some potential autocorrelation. The AR(2) model with , and has a definite pattern that intensifies as time increases. The ACF plot shows a clear trend. This data is not stationary.

#### 8.8

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

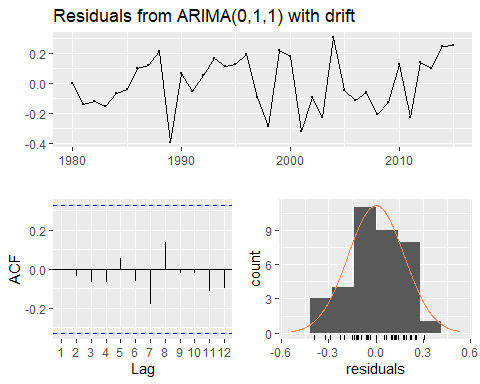
**Use auto.arima() to find an appropriate ARIMA model. Which model was selected? Check that the residuals look like white noise. Plot forecasts for the next 10 periods.**

fit <- auto.arima(austa)  
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

The model that was selected was ARIMA(0,1,1) which is simple exponential smoothing.

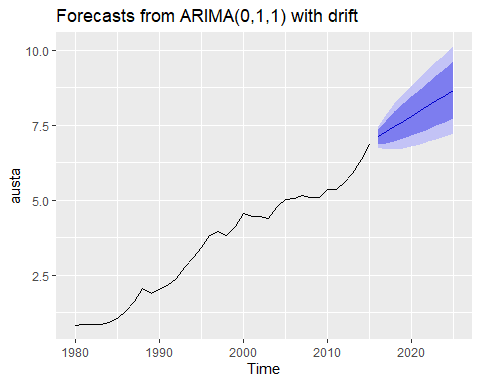
checkresiduals(fit)



##   
## 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

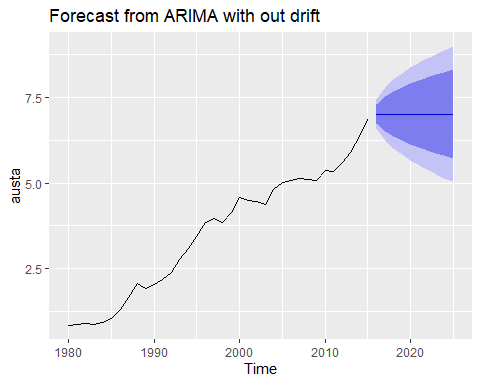
The ACF plot looks good and based on the Ljung-Box test results of p=0.8067, we’d fail to reject the null hypothesis and conclude that the residuals do resemble white noise.

autoplot(forecast(fit, h = 10))



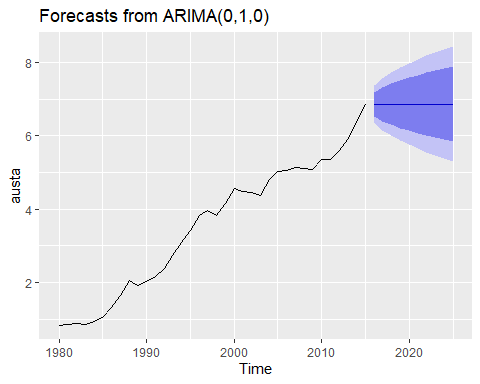
**Plot forecasts from an ARIMA(0,1,1) model with no drift and compare these to part a. Remove the MA term and plot again.**

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



The forecast without drift has a wider confidence interval and the trend is horizontal as opposed to increasing.

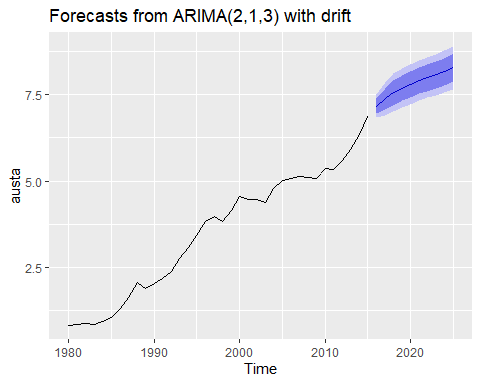
fit <- Arima(austa, order = c(0,1,0), include.drift = FALSE)  
p\_wo\_ma <- autoplot(forecast(fit, h = 10))  
p\_wo\_ma



The forecast without MA is almost indistinguishable from the previous plot, however, we see that the trend line is slightly higher.

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

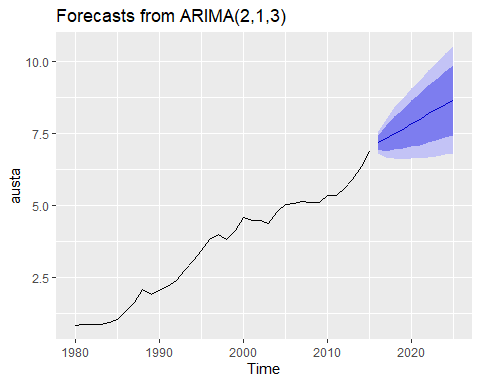
fit <- Arima(austa, order=c(2,1,3), method='ML', include.drift=T)  
autoplot(forecast(fit,h=10))



The forecast above shows the upward trend with a relatively narrow confidence interval.

Now, we’ll remove the constant.

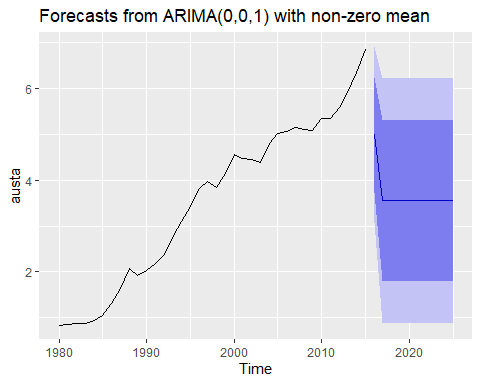
fit <- Arima(austa, order=c(2,1,3), method='ML', include.drift=T, include.constant =FALSE)  
autoplot(forecast(fit,h=10))



We can see that by removing the constant, the confidence interval has grown and has taken on a cone shape.

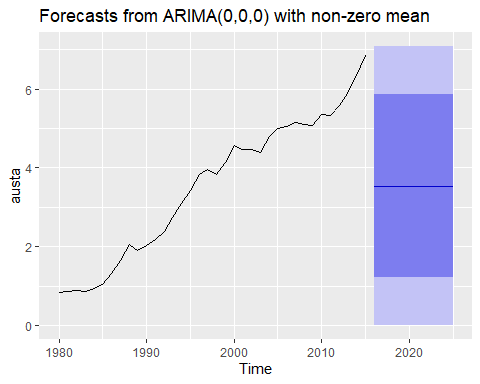
**Plot forecasts from an ARIMA(0,0,1) model with a constant. Remove the MA term and plot again.**

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



Now, let’s remove the MA term:

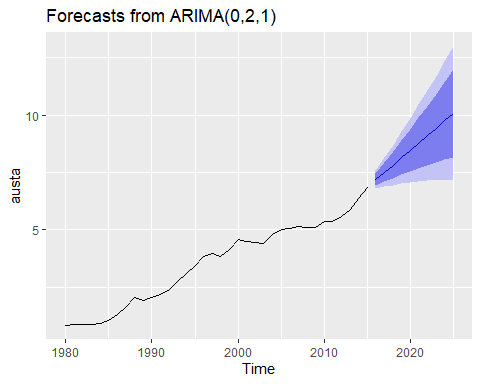
fit\_5 <- Arima(austa, order = c(0,0,0), include.constant = TRUE)  
autoplot(forecast(fit\_5, h = 10))



It appears that the forecast are the mean of the data history and the confidence interval spans the entire y-axis.

**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 shows an upward trend with a noticeable cone shaped confidence interval.