harinris_Homework5

Harin Rishabh

2024-04-21

Chapter 9 - R Commands

Exhibit 9.1

```
library(TSA)
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

```
data(color)
m1.color = arima(color, order=c(1,0,0))
m1.color
```

```
##
## Call:
## arima(x = color, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.5705    74.3293
## s.e.  0.1435     1.9151
##
## sigma^2 estimated as 24.83:  log likelihood = -106.07,  aic = 216.15
```

Exhibit 9.2

```
data(tempdub)
tempdub1=ts(c(tempdub,rep(NA,24)),start=start(tempdub),
freq=frequency(tempdub))
har.=harmonic(tempdub,1)
m5.tempdub=arima(tempdub,order=c(0,0,0),xreg=har.)
newhar.=harmonic(ts(rep(1,24), start=c(1976,1),freq=12),1)
plot(m5.tempdub,n.ahead=24,n1=c(1972,1),newxreg=newhar., type='b',ylab='Temperature',
xlab='Year')
```

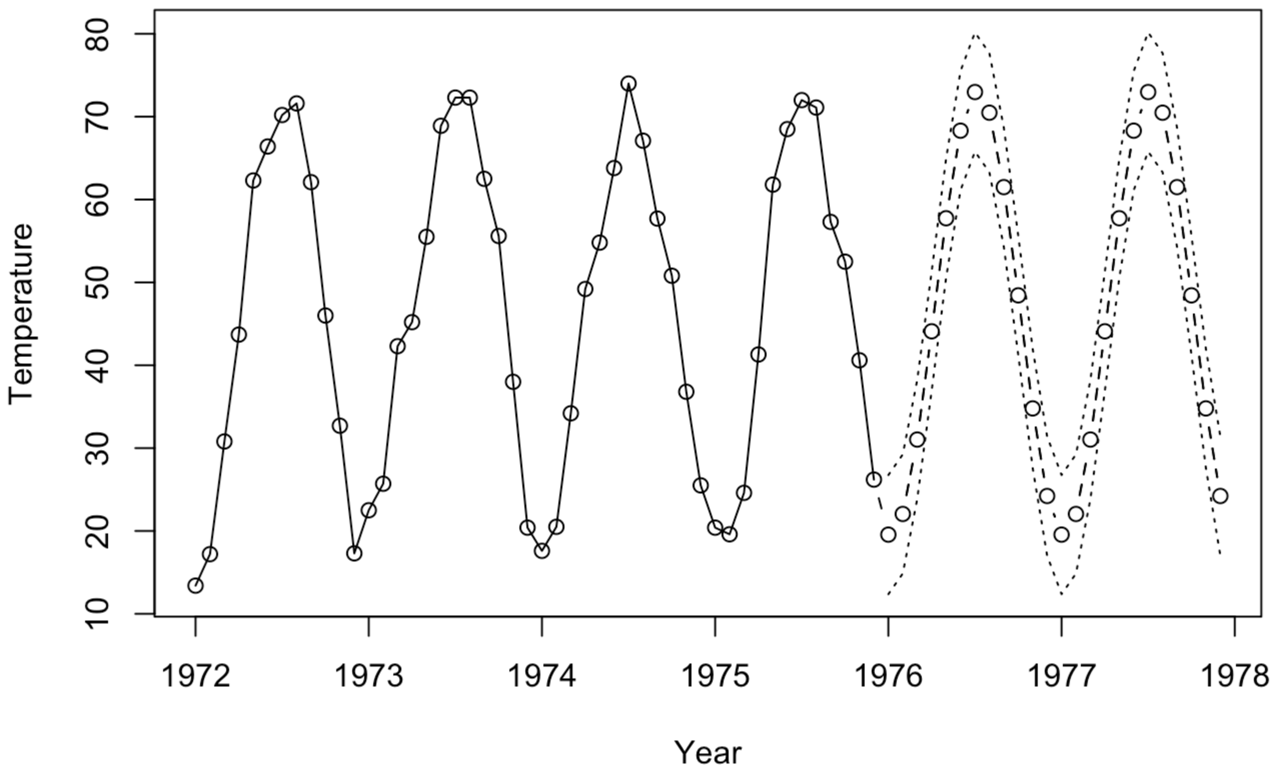


Exhibit 9.3

```
data(color)
m1.color=arima(color,order=c(1,0,0))
plot(m1.color,n.ahead=12,type='b', xlab='Time', ylab='Color Property')
abline(h=coef(m1.color)[names(coef(m1.color))=='intercept'])
```

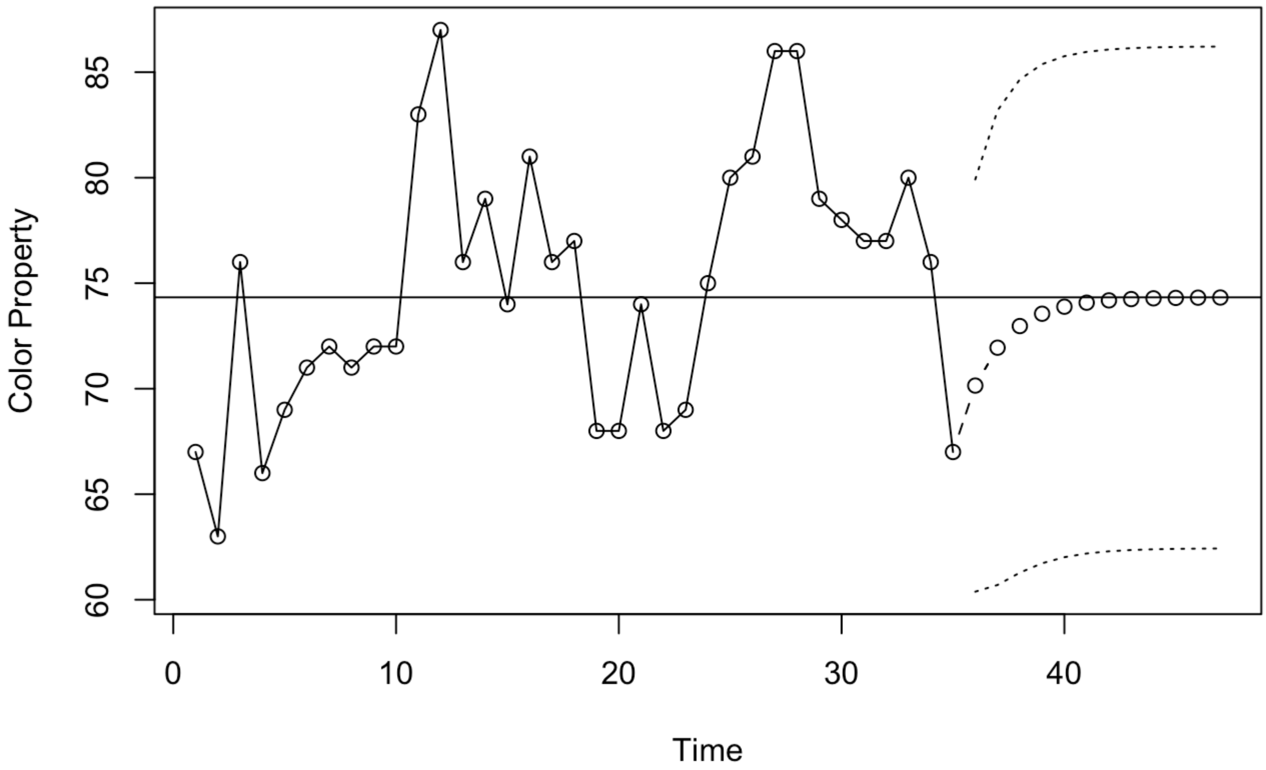
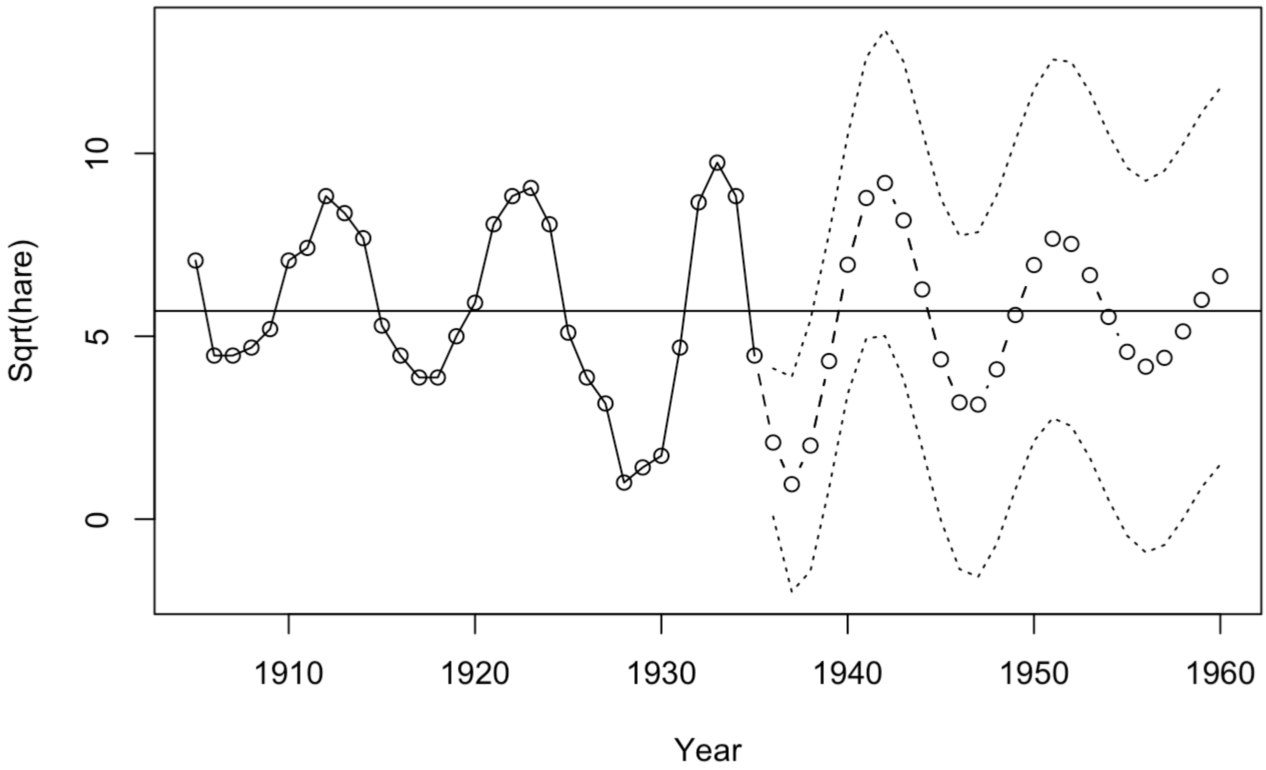


Exhibit 9.4

```
data(hare)
m1.hare=arima(sqrt(hare),order=c(3,0,0))
plot(m1.hare, n.ahead=25,type='b',xlab='Year',ylab='Sqrt(hare)')
abline(h=coef(m1.hare)[names(coef(m1.hare))=='intercept'])
```



Chapter 10 - R Commands

Exhibit 10.1

```
data(co2)
plot(co2,ylab='CO2')
```

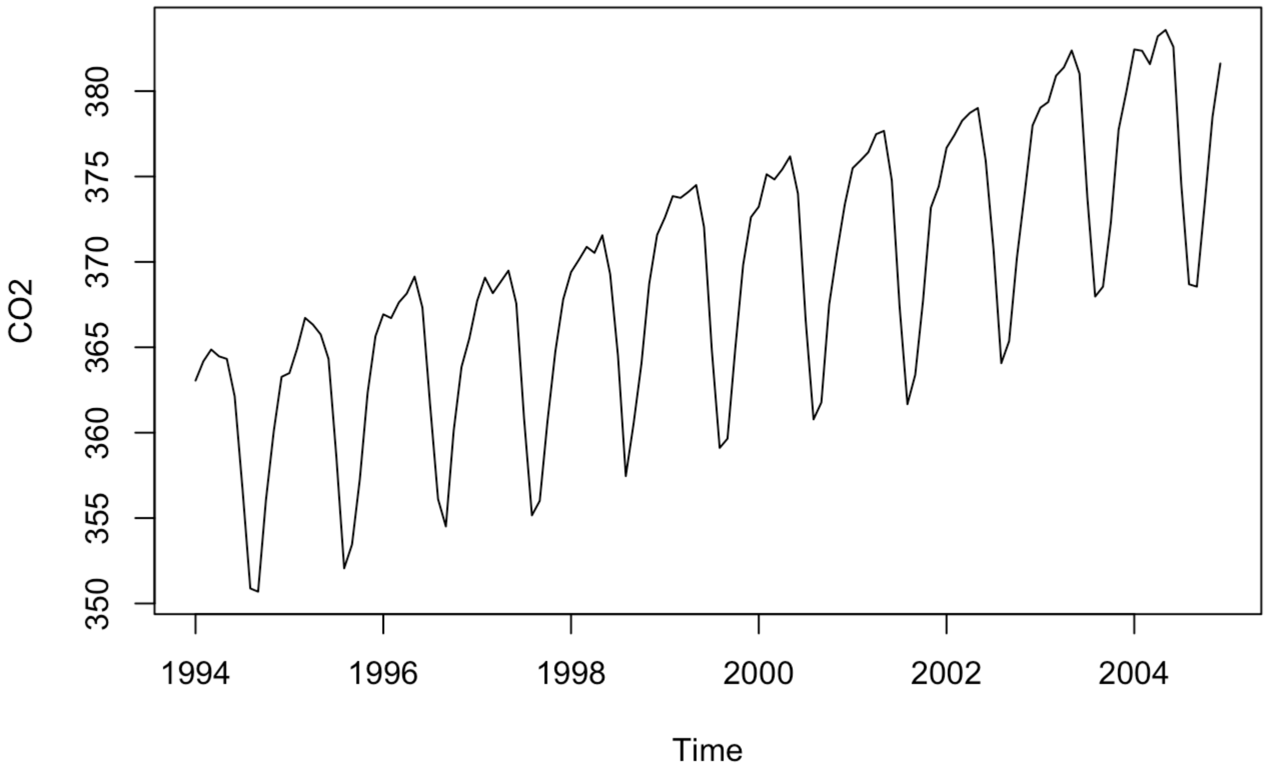


Exhibit 10.2

```
plot(window(co2,start=c(2000,1)),ylab='CO2')
Month=c('J','F','M','A','M','J','J','A','S','O','N','D')
points(window(co2,start=c(2000,1)),pch=Month)
```

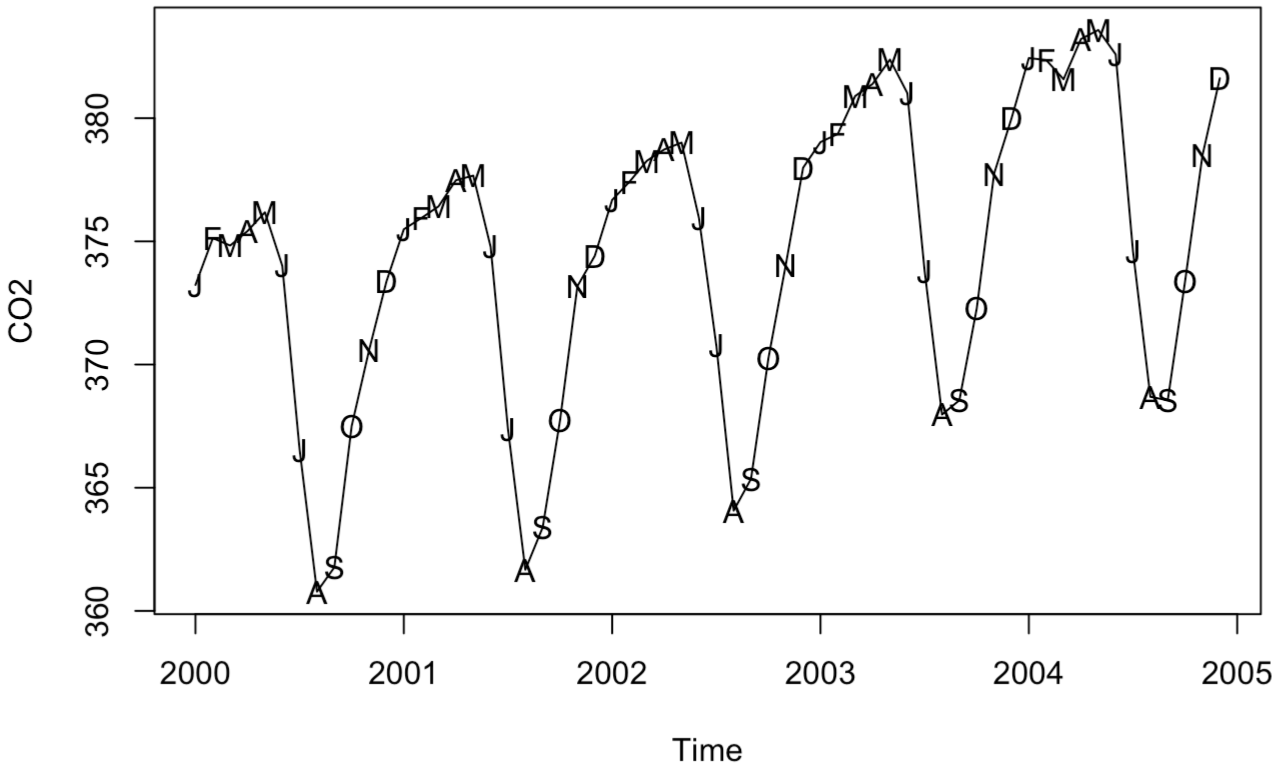


Exhibit 10.3

```
par(mfrow=c(1,2))
plot(y=ARMAacf(ma=c(0.5,rep(0,10),0.8,0.4),lag.max=13)[-1],x=1:13,type='h',
xlab='Lag k',ylab=expression(rho[k]),axes=F,ylim=c(0,0.6))
points(y=ARMAacf(ma=c(0.5,rep(0,10),0.8,0.4),lag.max=13)[-1],x=1:13,pch=20)
abline(h=0)
axis(1,at=1:13, labels=c(1,NA,3,NA,5,NA,7,NA,9,NA,11,NA,13))
axis(2)
text(x=7,y=.5,labels=expression(list(theta==-0.5,Theta==-0.8)))
plot(y=ARMAacf(ma=c(-0.5,rep(0,10),0.8,-0.4),lag.max=13)[-1],x=1:13,type='h',
xlab='Lag k',ylab=expression(rho[k]),axes=F)
points(y=ARMAacf(ma=c(-0.5,rep(0,10),0.8,-0.4),lag.max=13)[-1],x=1:13,pch=20)
abline(h=0)
axis(1,at=1:13, labels=c(1,NA,3,NA,5,NA,7,NA,9,NA,11,NA,13))
axis(2)
text(x=7,y=.35,labels=expression(list(theta==0.5,Theta==-0.8)))
```

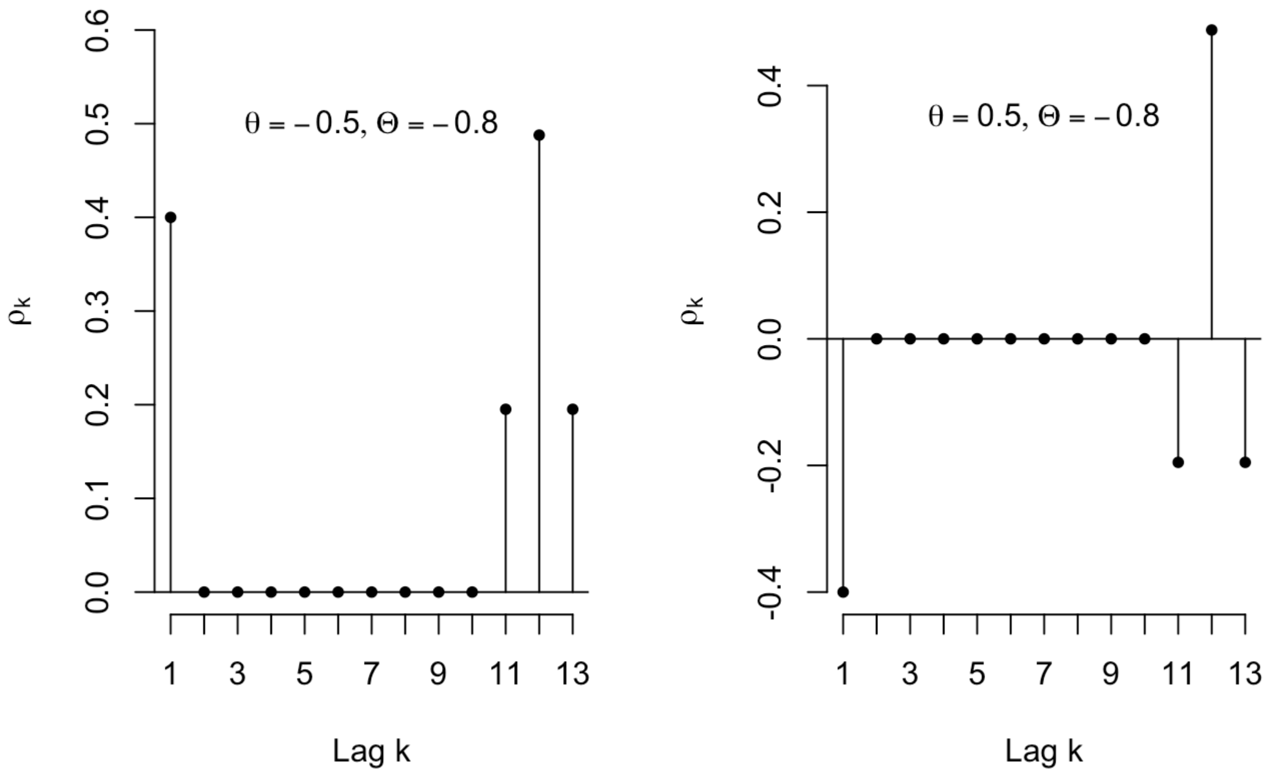


Exhibit 10.5

```
acf(as.vector(co2),lag.max=36)
```

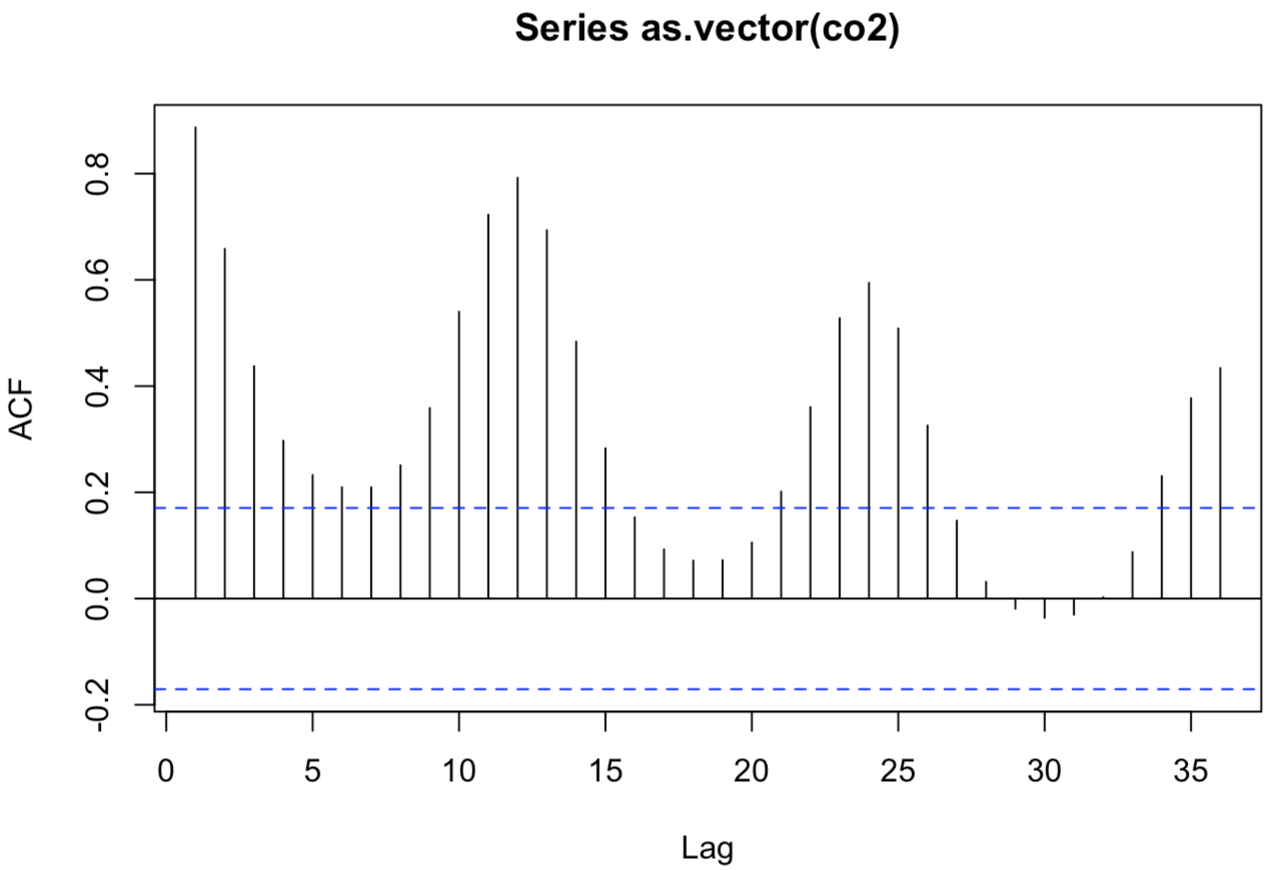


Exhibit 10.6

```
plot(diff(co2),ylab='First Difference of CO2',xlab='Time')
```

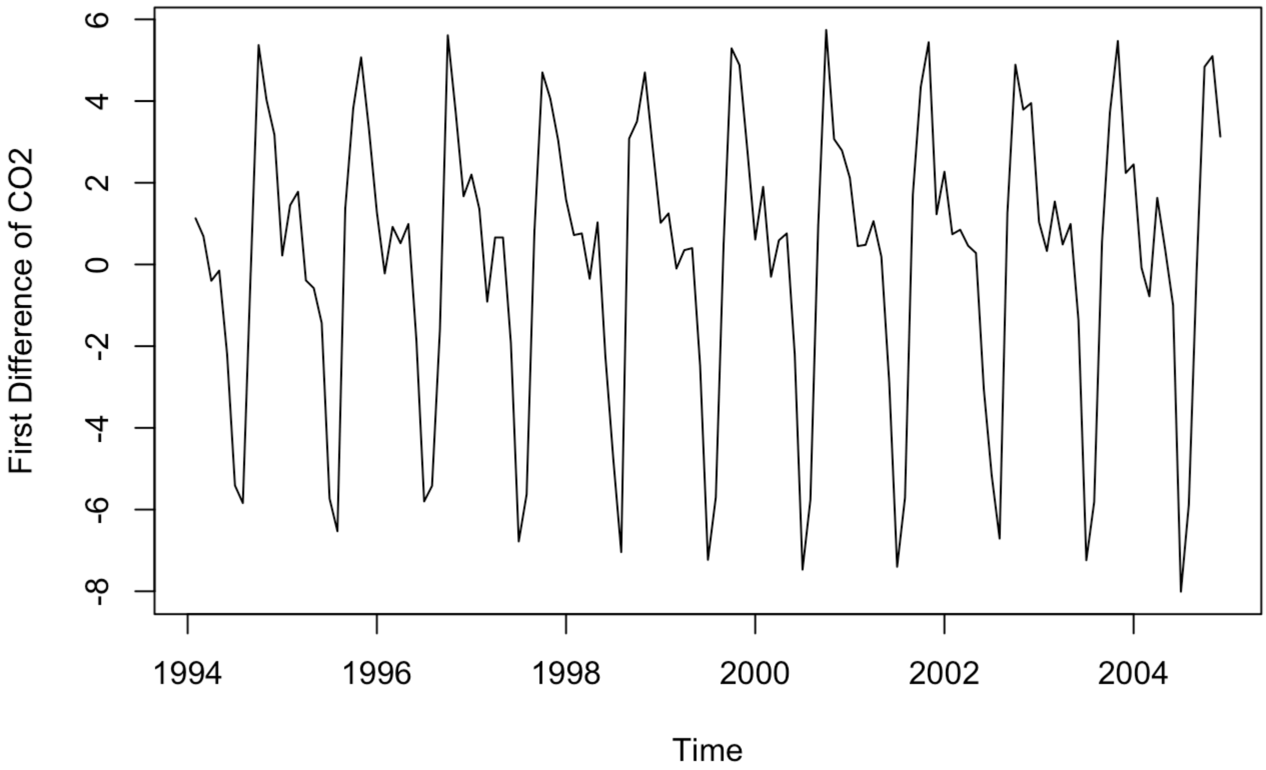


Exhibit 10.7

```
acf(as.vector(diff(co2)),lag.max=36)
```

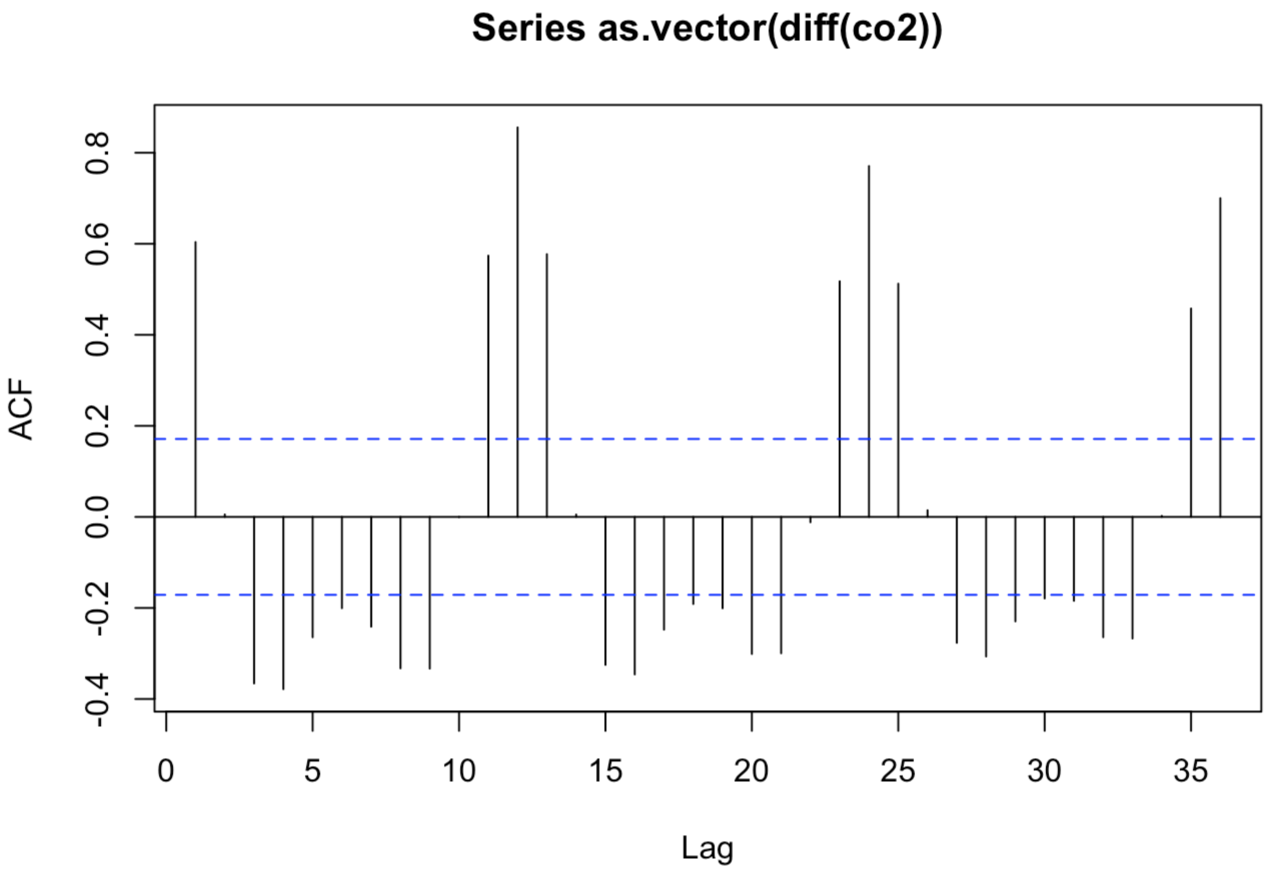


Exhibit 10.8

```
plot(diff(diff(co2),lag=12),xlab='Time',
ylab='First and Seasonal Difference of CO2')
```

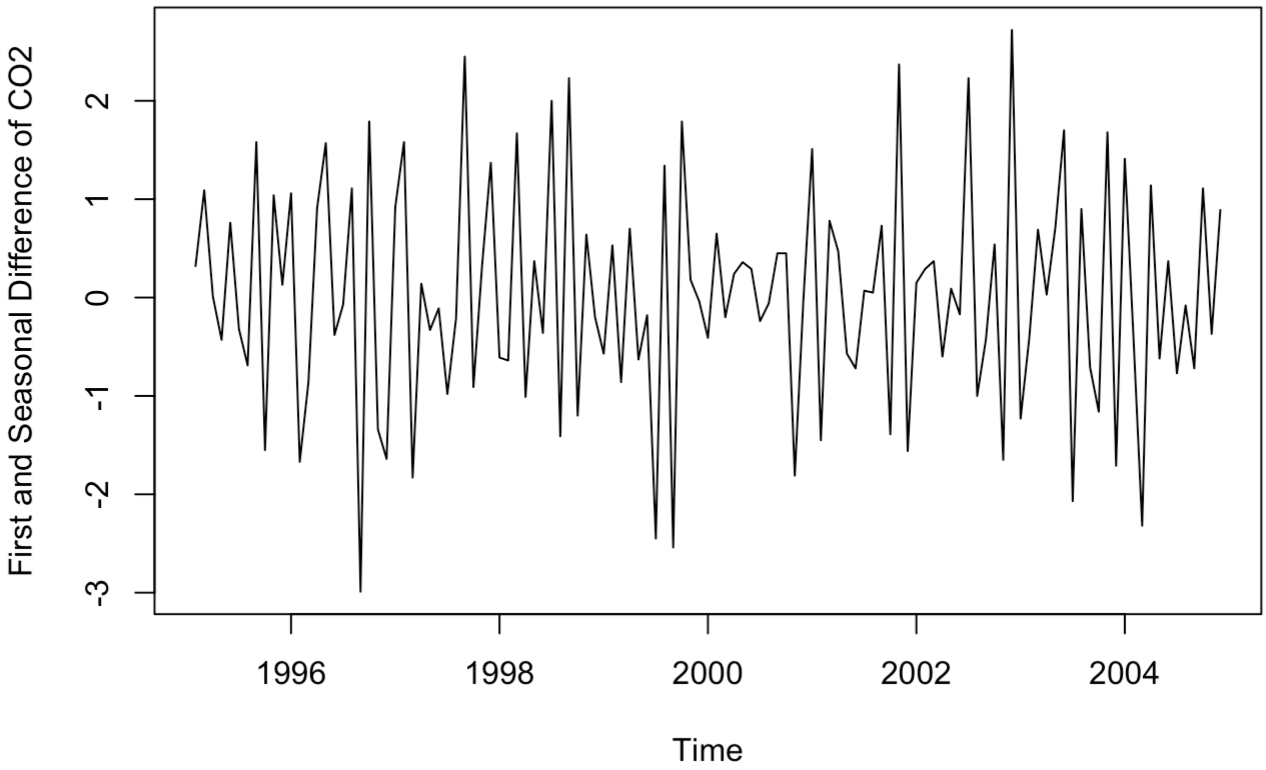


Exhibit 10.9

```
acf(as.vector(diff(diff(co2),lag=12)),lag.max=36,ci.type='ma')
```

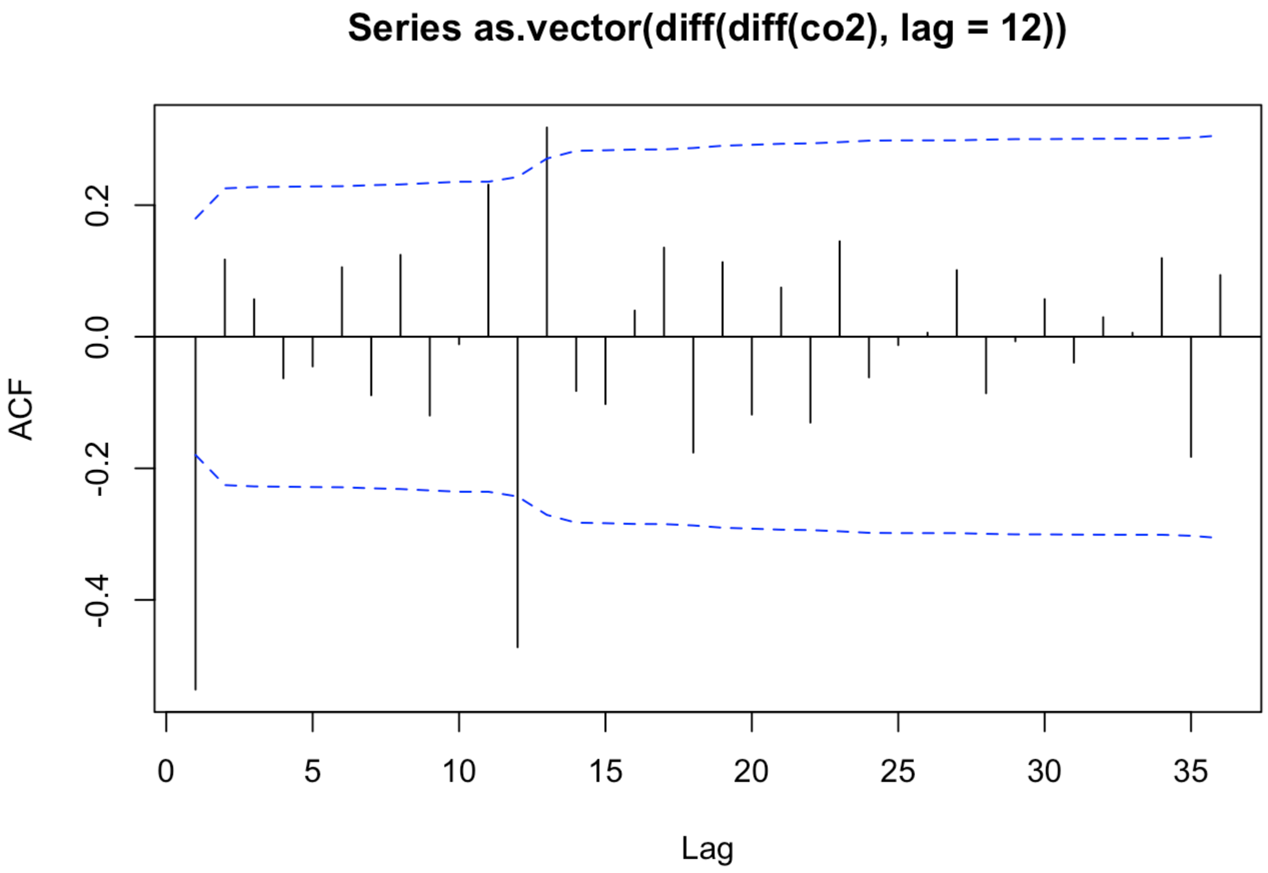


Exhibit 10.10

```
m1.co2=arima(co2,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12))
m1.co2
```

```
##
## Call:
## arima(x = co2, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 1
2))
##
## Coefficients:
##           ma1     sma1
##       -0.5792  -0.8206
## s.e.   0.0791   0.1137
##
## sigma^2 estimated as 0.5446:  log likelihood = -139.54,  aic = 283.08
```

Exhibit 10.10

```
plot(window(rstandard(m1.co2),start=c(1995,2)), ylab='Standardized Residuals',type
='o')
abline(h=0)
```

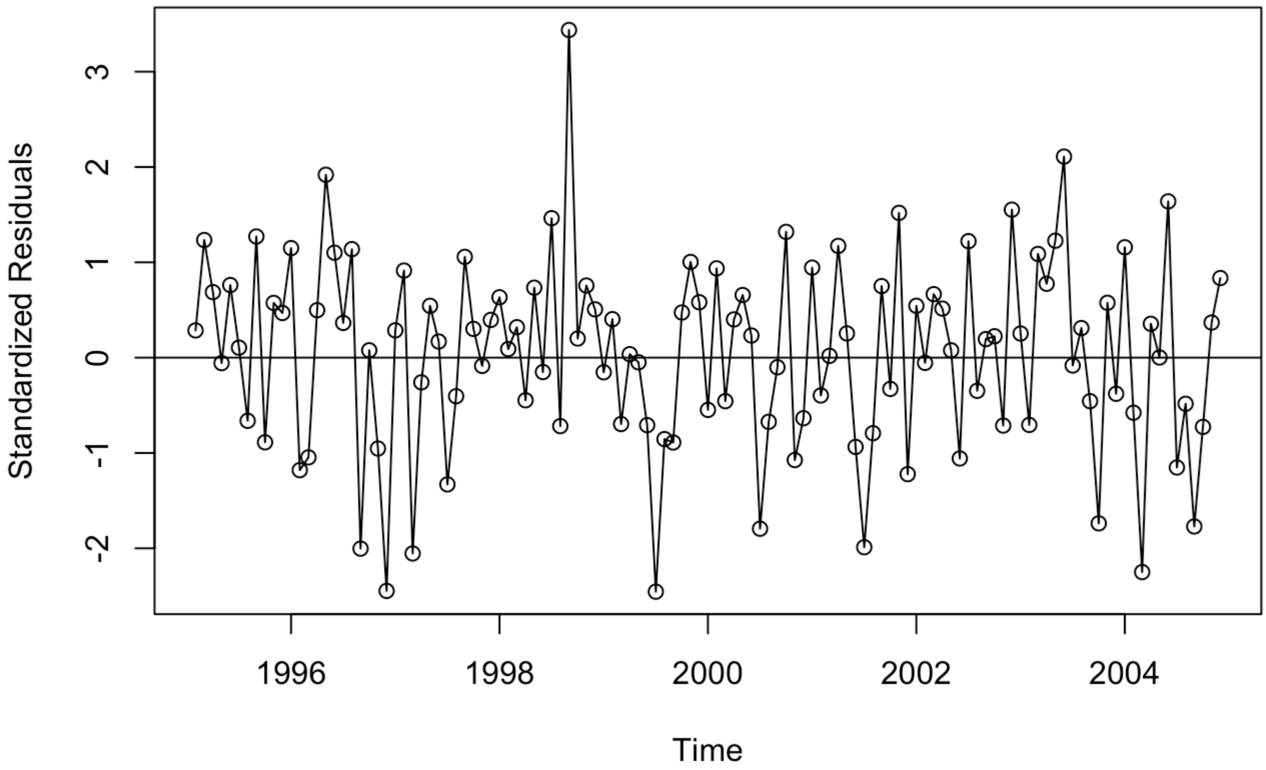


Exhibit 10.12

```
acf(as.vector(window(rstandard(m1.co2),start=c(1995,2))), lag.max=36)
```

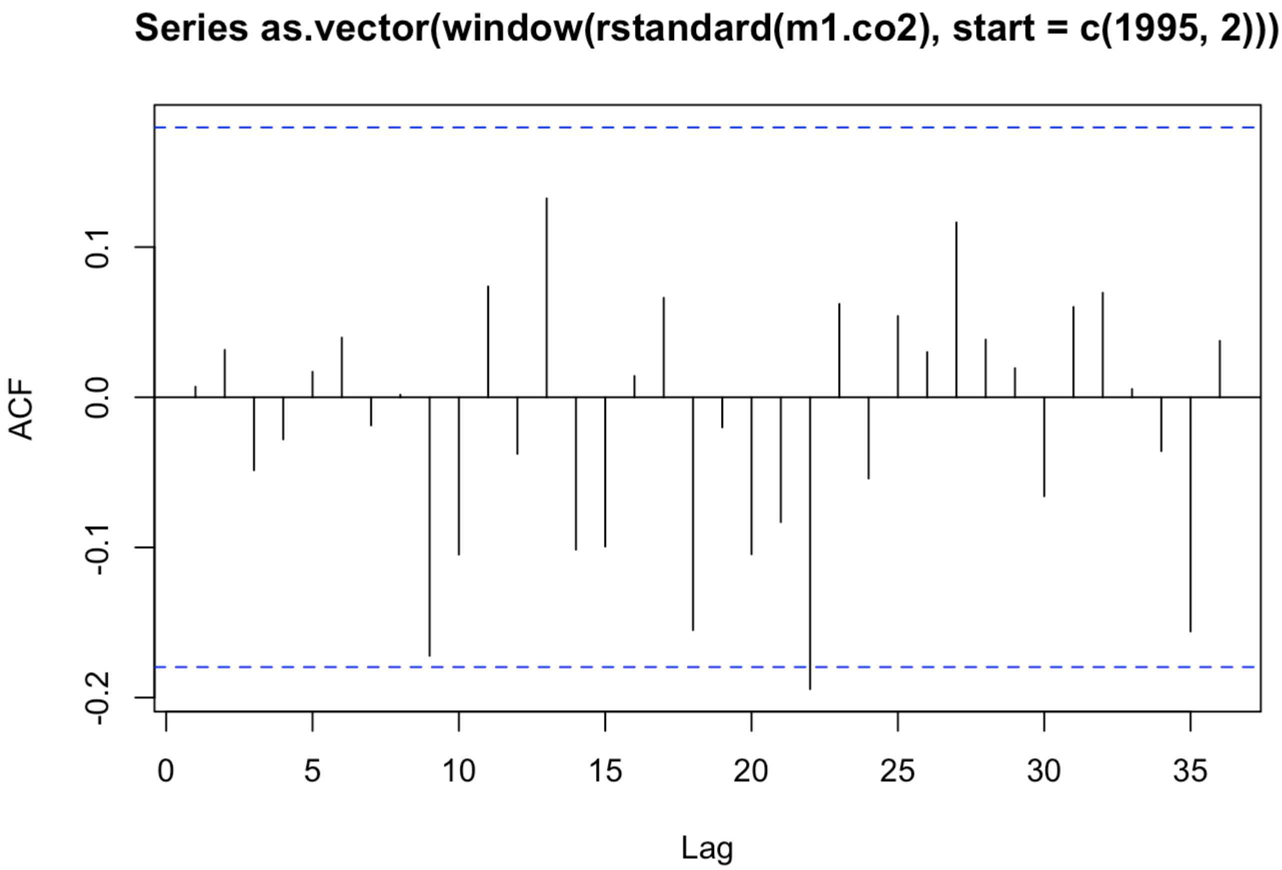


Exhibit 10.16

```
plot(m1.co2,n1=c(2003,1),n.ahead=24,xlab='Year',type='o',
ylab='CO2 Levels')
```

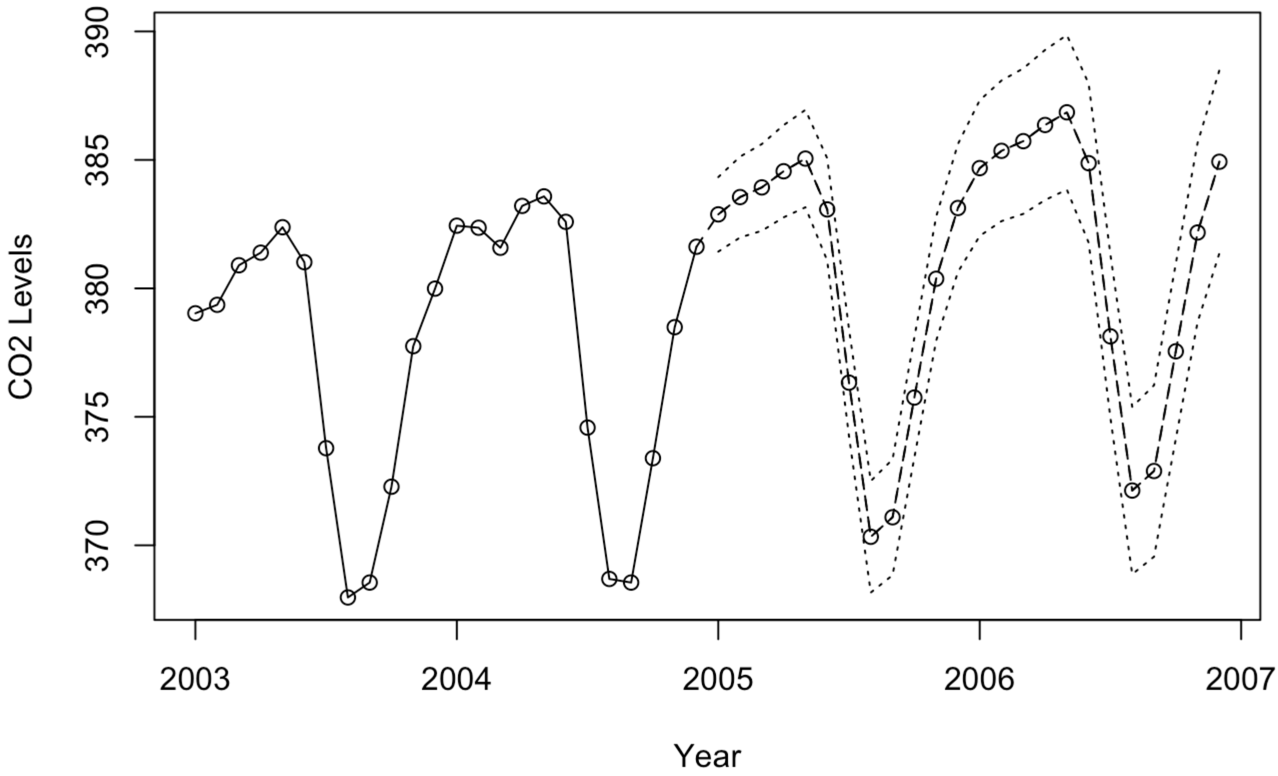
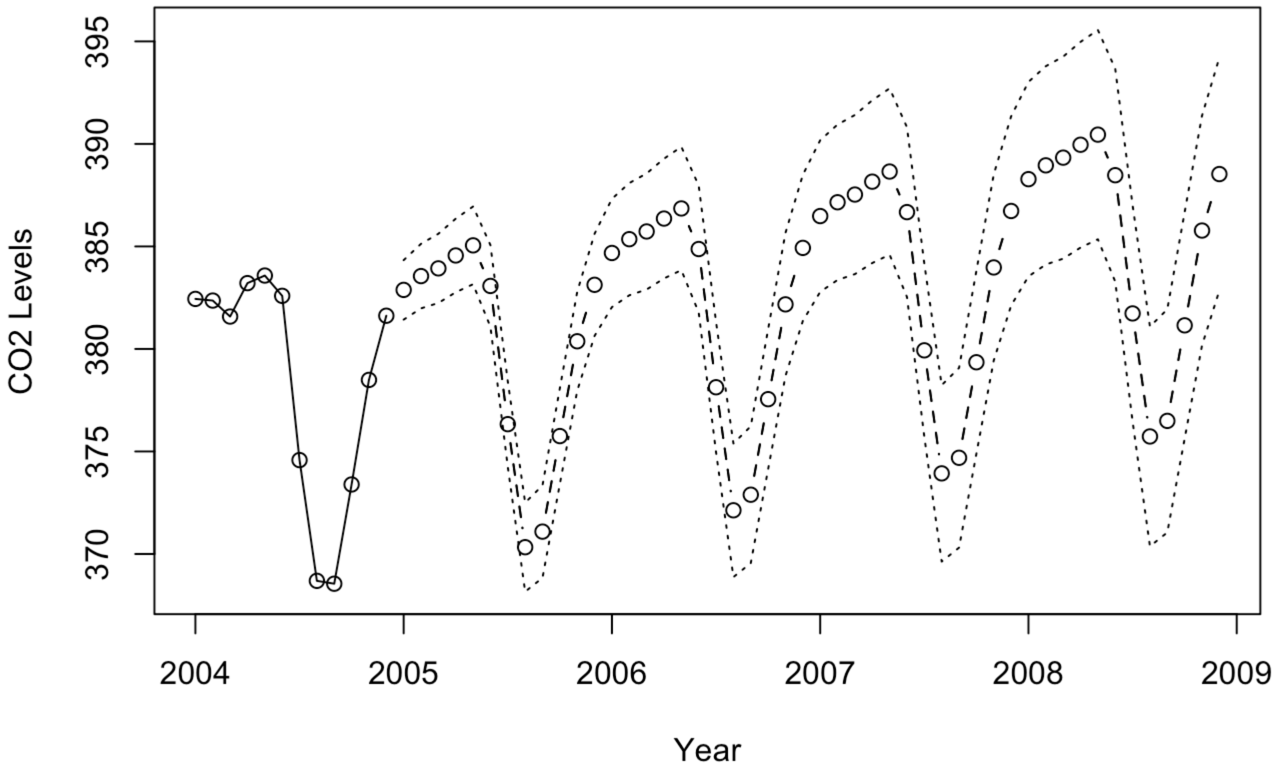


Exhibit 10.17

```
plot(m1.co2,n1=c(2004,1),n.ahead=48,xlab='Year',type='b',
ylab='CO2 Levels')
```



Exercises

Ex 9.9

```
set.seed(123)
series=arima.sim(n=48,list(ar=0.8))+100
actual=window(series,start=41); series=window(series,end=40)
```

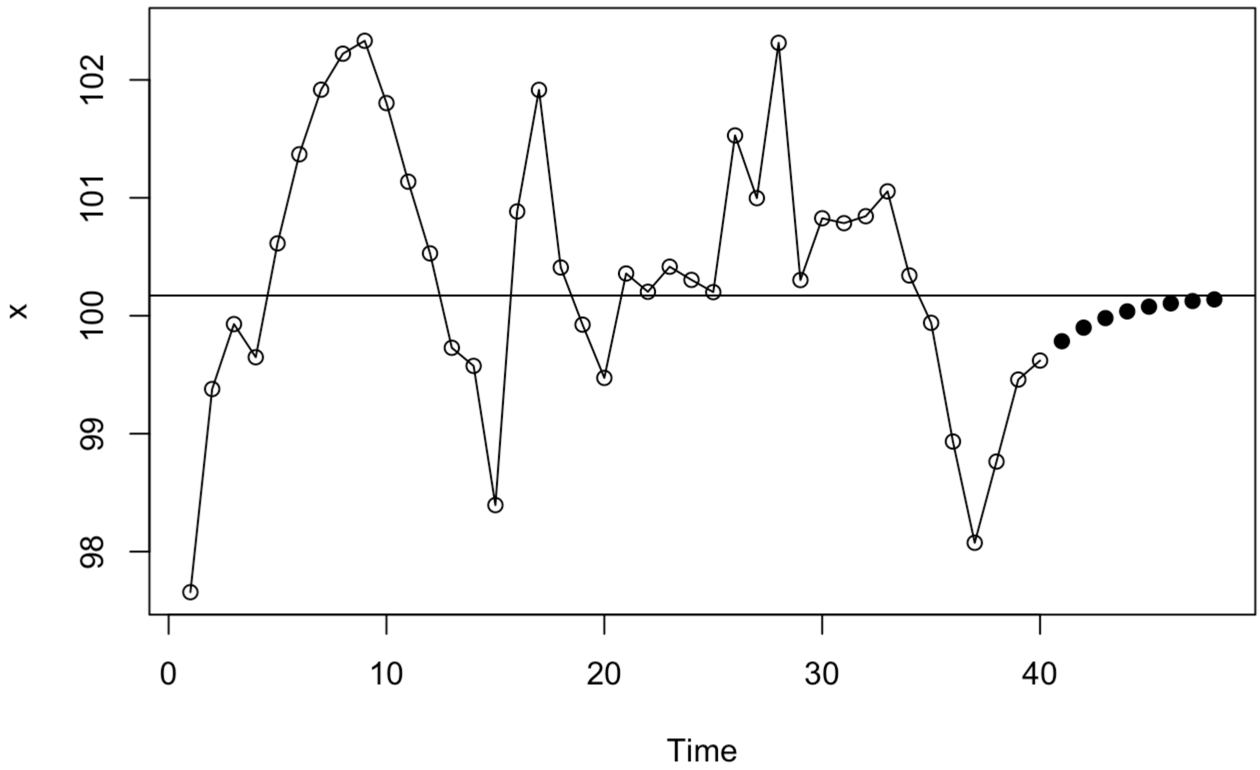
a

```
model=arima(series,order=c(1,0,0))
model
```

```
##
## Call:
## arima(x = series, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.7027   100.1716
## s.e.  0.1229     0.4279
##
## sigma^2 estimated as 0.6876:  log likelihood = -49.61,  aic = 103.21
```

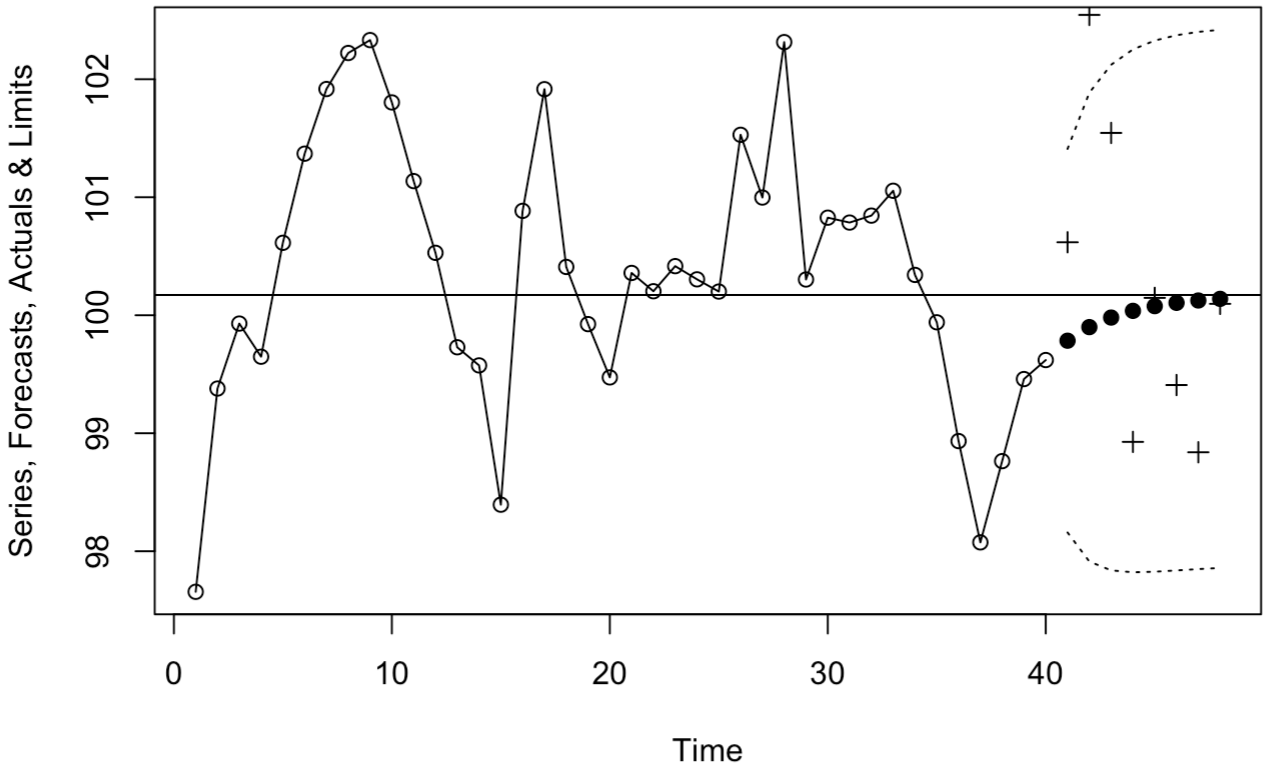
b

```
plot(model,n.ahead=8,col=NULL,pch=19)
abline(h=coef(model)[names(coef(model))=='intercept'])
```



c, d The forecasts are plotted as solid circles. The actual values are solid circles. They lie within the 95% confidence intervals which is depicted by the dotted line.

```
 plot(model,n.ahead=8,ylab='Series, Forecasts, Actuals & Limits',pch=19)
points(x=(41:48),y=actual,pch=3) # Add the actual future values to the plot
abline(h=coef(model)[names(coef(model))=='intercept'])
```



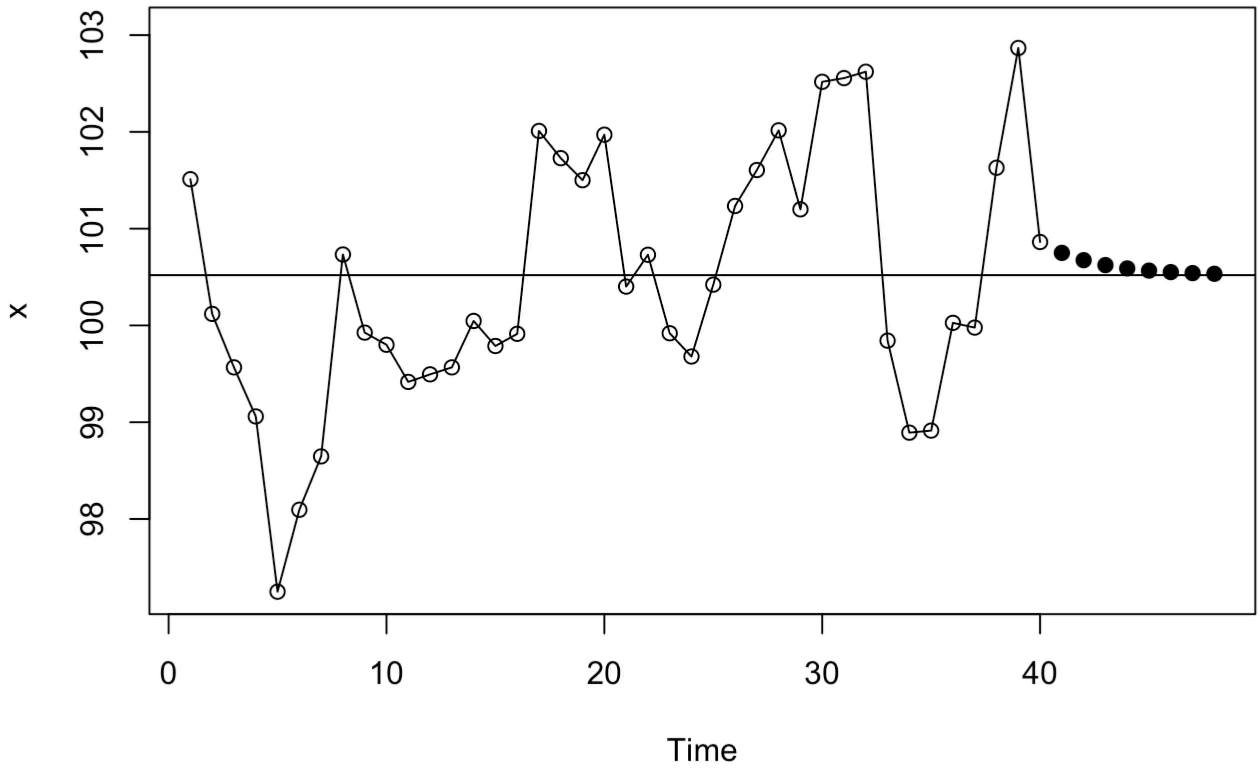
e

Predictions within 95% confidence intervals.

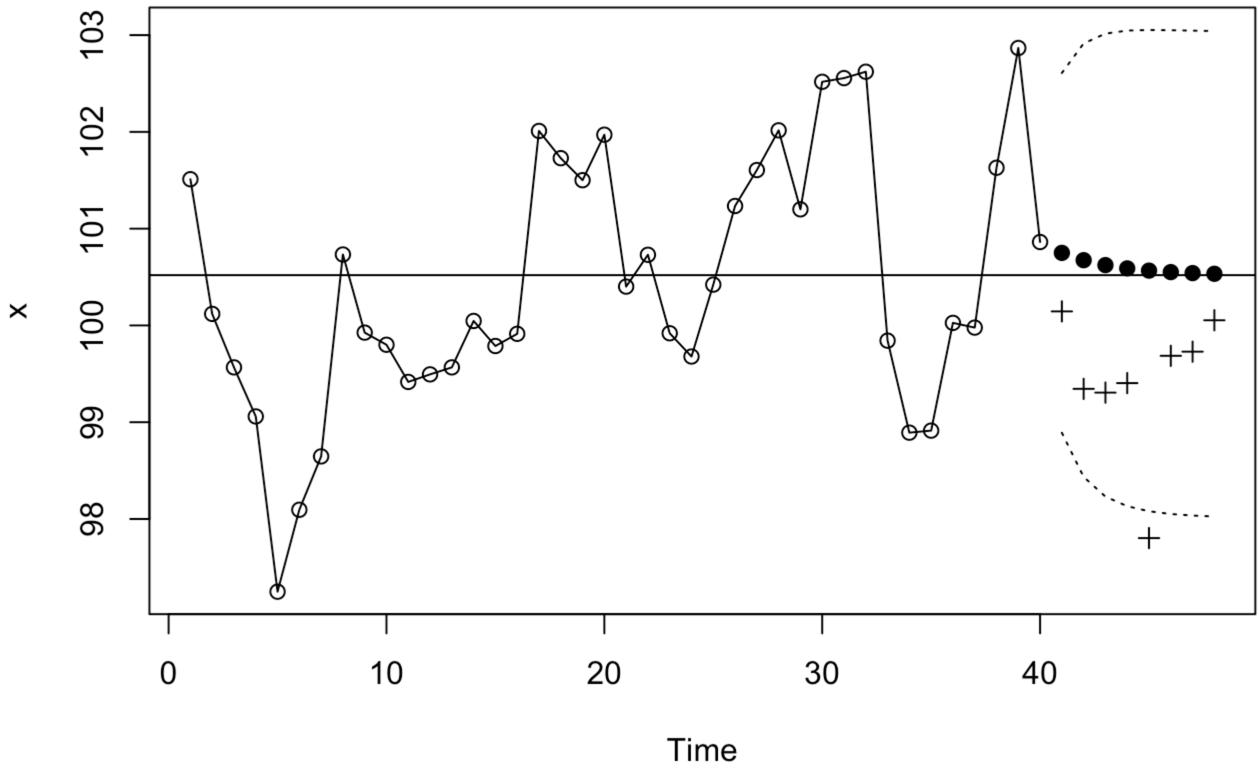
```
set.seed(456)
series=arima.sim(n=48,list(ar=0.8))+100
actual=window(series,start=41); series=window(series,end=40)
model=arima(series,order=c(1,0,0)); model
```

```
##
## Call:
## arima(x = series, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.6720   100.5196
## s.e.  0.1131     0.4366
##
## sigma^2 estimated as 0.8997:  log likelihood = -54.94,  aic = 113.89
```

```
plot(model,n.ahead=8,col=NULL,pch=19)
abline(h=coef(model)[names(coef(model))=='intercept'])
```




```
plot(model,n.ahead=8,pch=19)
points(x=(41:48),y=actual,pch=3)
abline(h=coef(model)[names(coef(model))=='intercept'])
```



Ex 9.12

```
set.seed(123)
series = arima.sim(model=list(ma=c(-1, 0.6)), n=36) + 100
actual = window(series,start=33)
series = window(series,end=32)
actual
```

```
## Time Series:
## Start = 33
## End = 36
## Frequency = 1
## [1] 100.48052 100.39394 100.35823  99.79735
```

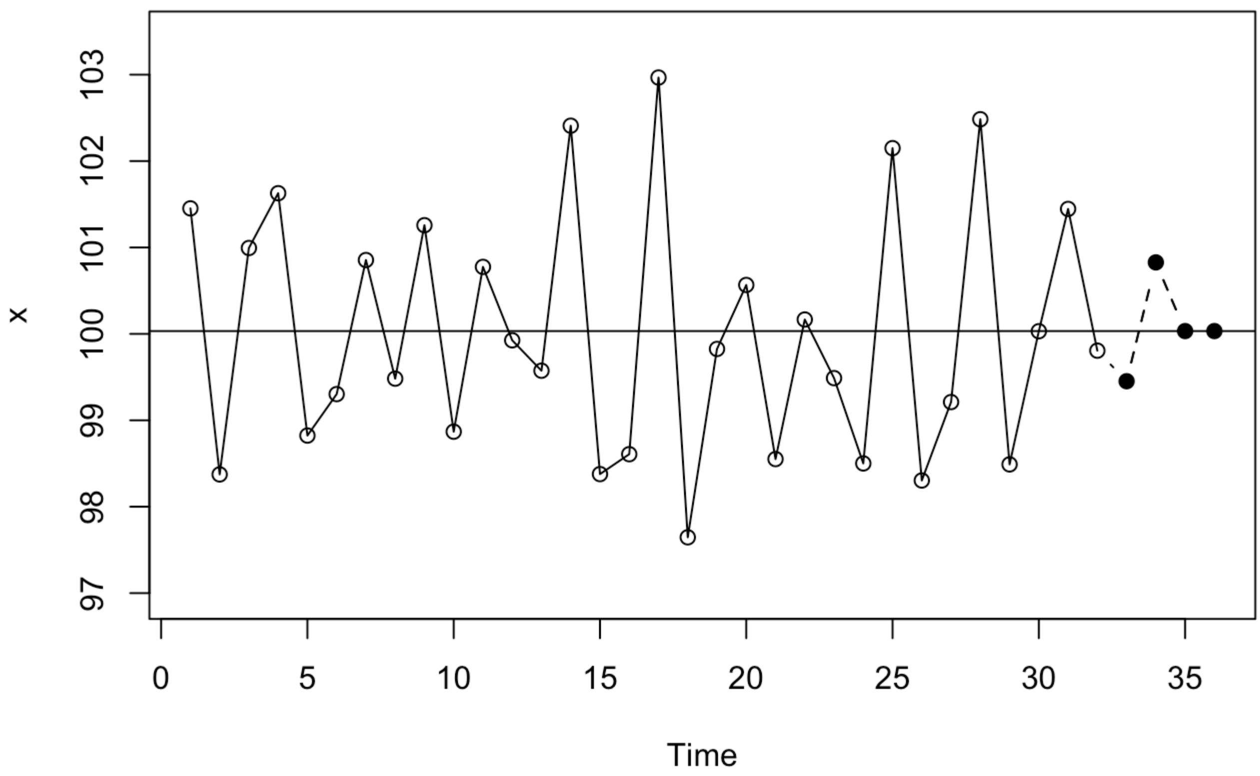
a

```
model=arima(series,order=c(0,0,2))
model
```

```
##
## Call:
## arima(x = series, order = c(0, 0, 2))
##
## Coefficients:
##           ma1     ma2  intercept
##       -1.2776  1.0000   100.0326
## s.e.   0.1403  0.1641     0.1023
##
## sigma^2 estimated as 0.6759:  log likelihood = -42.23,  aic = 90.47
```

b

```
result=plot(model,n.ahead=4,col=NULL,pch=19)
abline(h=coef(model)[names(coef(model))=='intercept'])
```



c

The values past lag 2 are the estimated process mean.

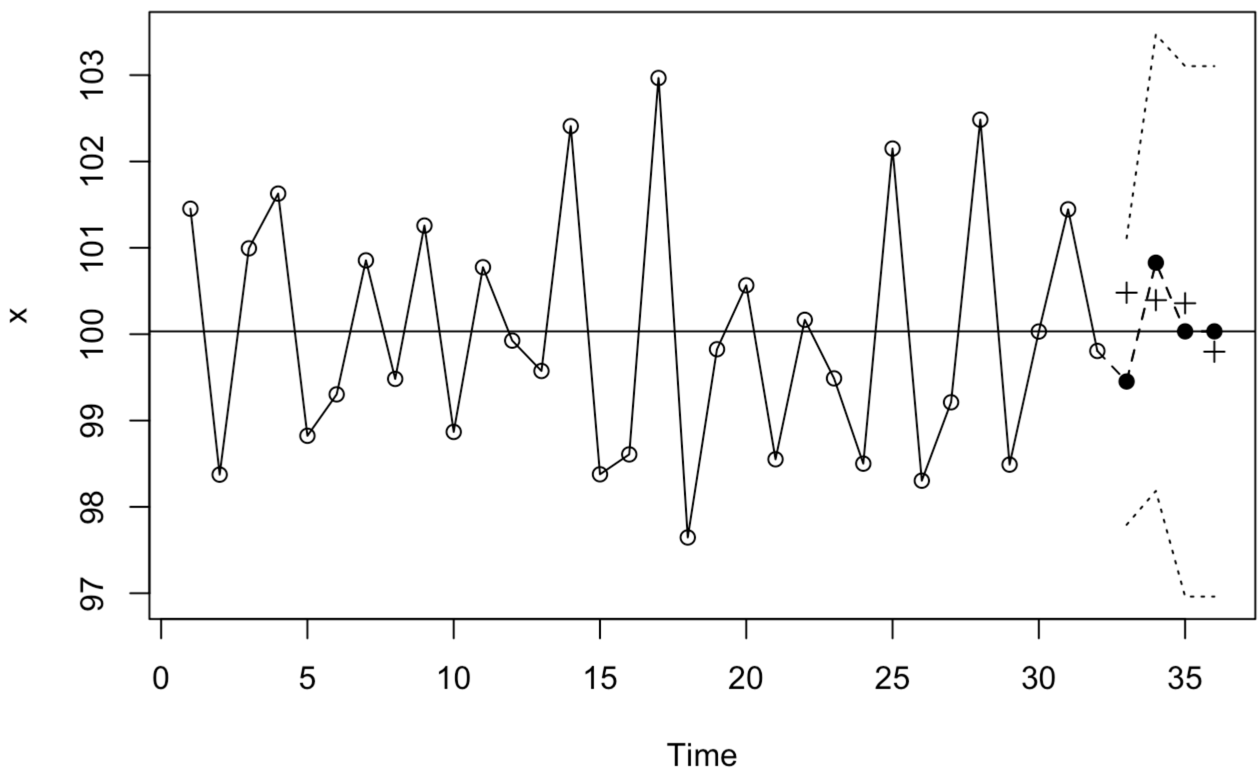
d, e

The predicted values are pretty close to the actuals and definitely fall between the 95% CI.

```
forecast=result$pred
cbind(actual,forecast)
```

```
## Time Series:
## Start = 33
## End = 36
## Frequency = 1
##       actual forecast
## 33 100.48052  99.4522
## 34 100.39394 100.8275
## 35 100.35823 100.0326
## 36  99.79735 100.0326
```

```
plot(model,n.ahead=4,type='o',pch=19)
points(x=(33:36), y=actual, pch=3)
abline(h=coef(model)[names(coef(model))=='intercept'])
```

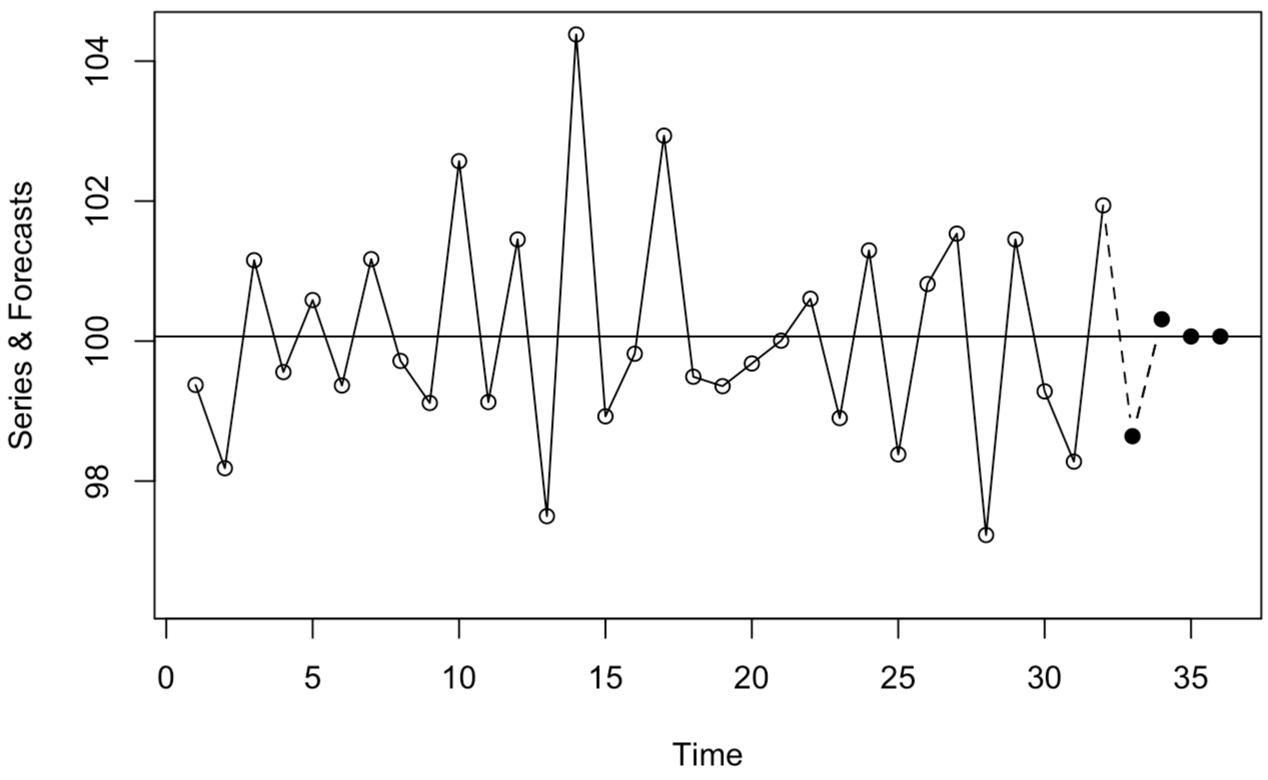


f The predicted values are pretty close to the actuals and definitely fall between the 95% CI.

```
set.seed(456)
series=arima.sim(n=36,list(ma=c(-1,0.6)))+100
actual=window(series,start=33); series=window(series,end=32)
model=arima(series,order=c(0,0,2)); model
```

```
##
## Call:
## arima(x = series, order = c(0, 0, 2))
##
## Coefficients:
##           ma1     ma2  intercept
##       -0.8423  0.5129   100.0658
## s.e.   0.1799  0.1795     0.1381
##
## sigma^2 estimated as 1.36:  log likelihood = -50.82,  aic = 107.64
```

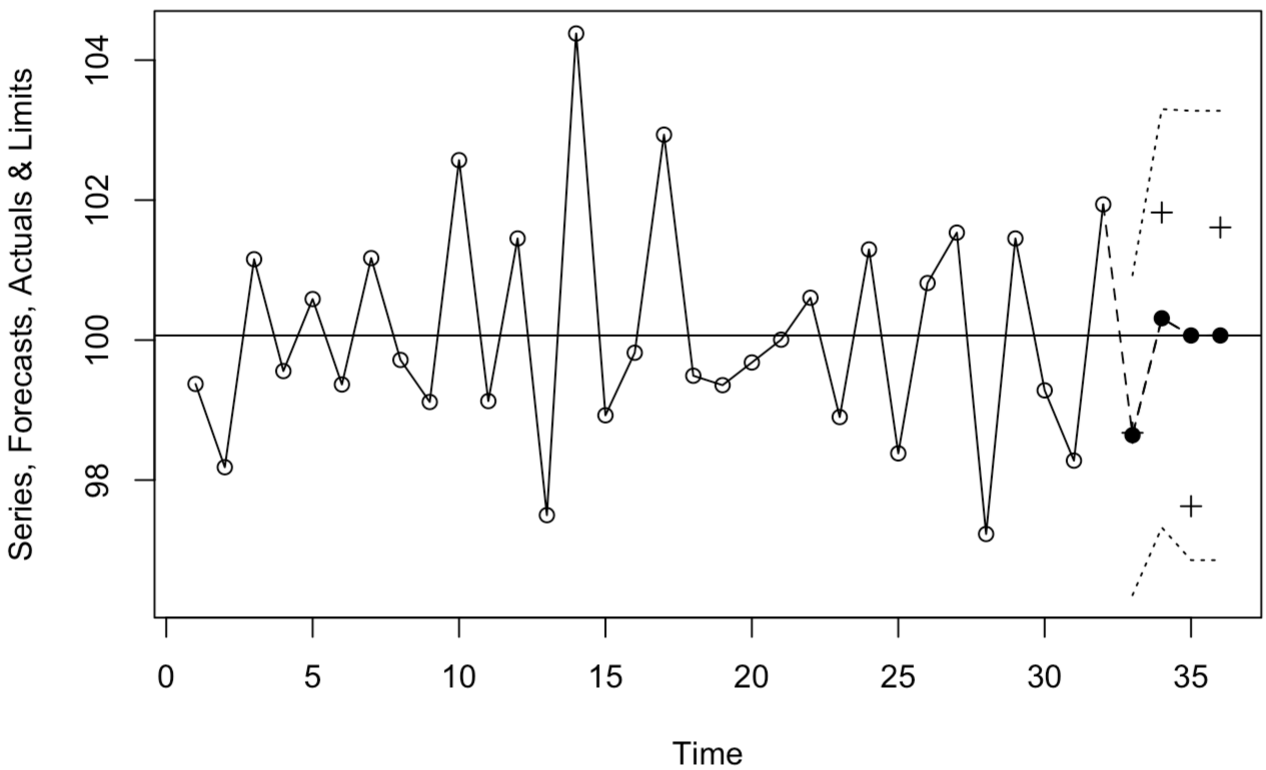
```
result=plot(model,n.ahead=4,ylab='Series & Forecasts',col=NULL,pch=19)
abline(h=coef(model)[names(coef(model))=='intercept'])
```




```
forecast=result$pred; cbind(actual,forecast)
```

```
## Time Series:
## Start = 33
## End = 36
## Frequency = 1
##       actual  forecast
## 33  98.67512  98.64093
## 34 101.82272 100.31139
## 35  97.62612 100.06575
## 36 101.60974 100.06575
```

```
plot(model,n.ahead=4,ylab='Series, Forecasts, Actuals & Limits',type='o',pch=19)
points(x=(33:36),y=actual,pch=3)
abline(h=coef(model)[names(coef(model))=='intercept'])
```



Ex 9.13

```
set.seed(123)
series=arima.sim(n=50,list(ar=0.7,ma=0.5))+100
actual=window(series,start=41)
series=window(series,end=40)
```

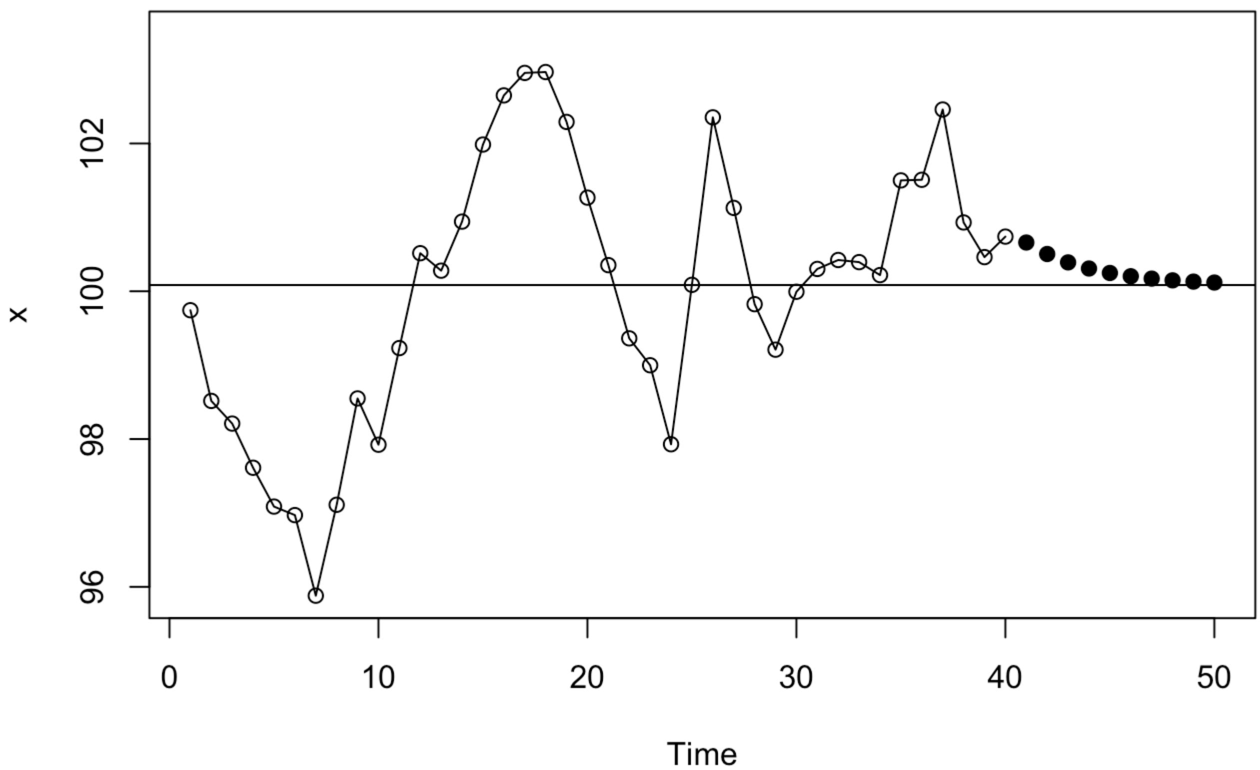
a

```
model=arima(series,order=c(1,0,1))
model
```

```
##
## Call:
## arima(x = series, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.7298  0.4031   100.0842
## s.e.  0.1210  0.1960     0.6730
##
## sigma^2 estimated as 0.7712:  log likelihood = -52.29,  aic = 110.58
```

b

```
result=plot(model,n.ahead=10,col=NULL,pch=19)
abline(h=coef(model)[names(coef(model))=='intercept'])
```



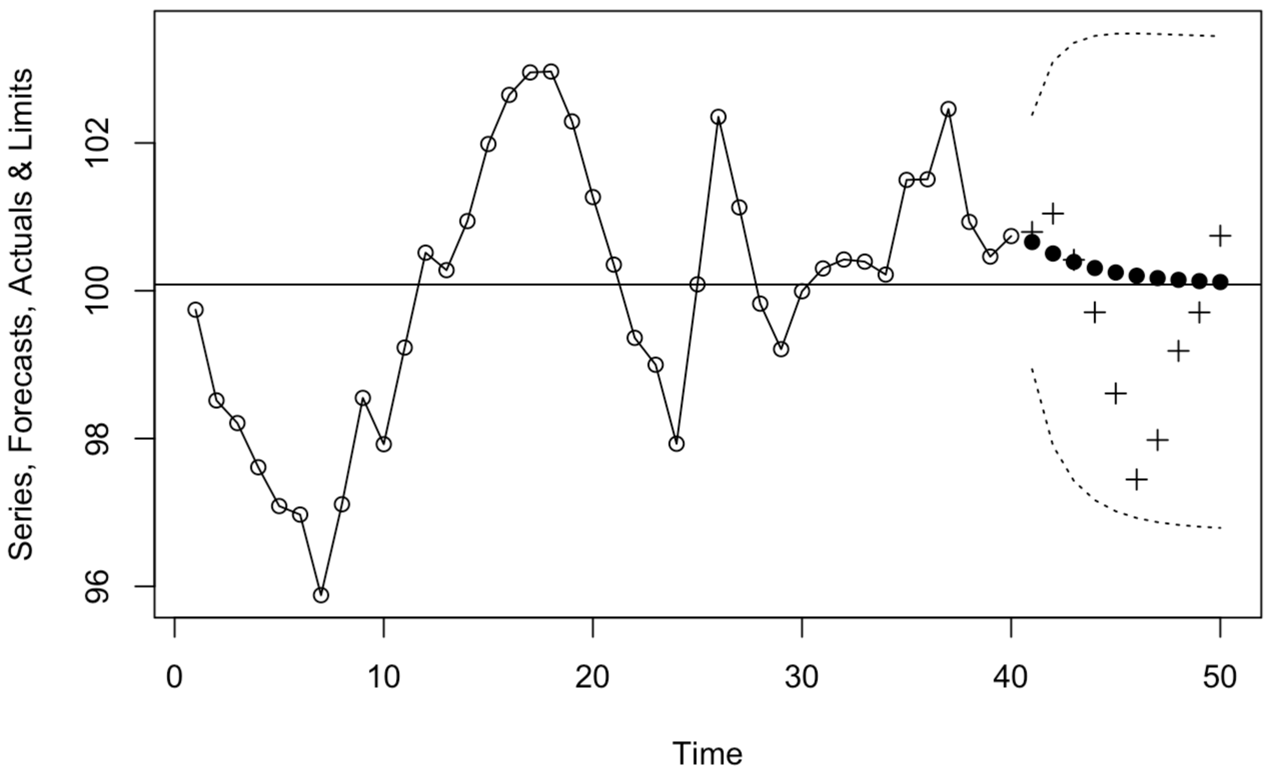
The forecasts approach the series mean.

```
forecast=result$pred
cbind(actual,forecast)
```

```
## Time Series:
## Start = 41
## End = 50
## Frequency = 1
##       actual forecast
## 41 100.79563 100.6596
## 42 101.04455 100.5041
## 43 100.41868 100.3907
## 44  99.70871 100.3079
## 45  98.61092 100.2474
## 46  97.44656 100.2033
## 47  97.98023 100.1712
## 48  99.18613 100.1477
## 49  99.70740 100.1305
## 50 100.74395 100.1180
```

d

```
plot(model,n.ahead=10,ylab='Series, Forecasts, Actuals & Limits',pch=19)
points(x=(41:50),y=actual,pch=3)
abline(h=coef(model)[names(coef(model))=='intercept'])
```



The predicted series decays towards mean. However, the actual values are pretty far apart. However, the predictions still fall between 95% CI.

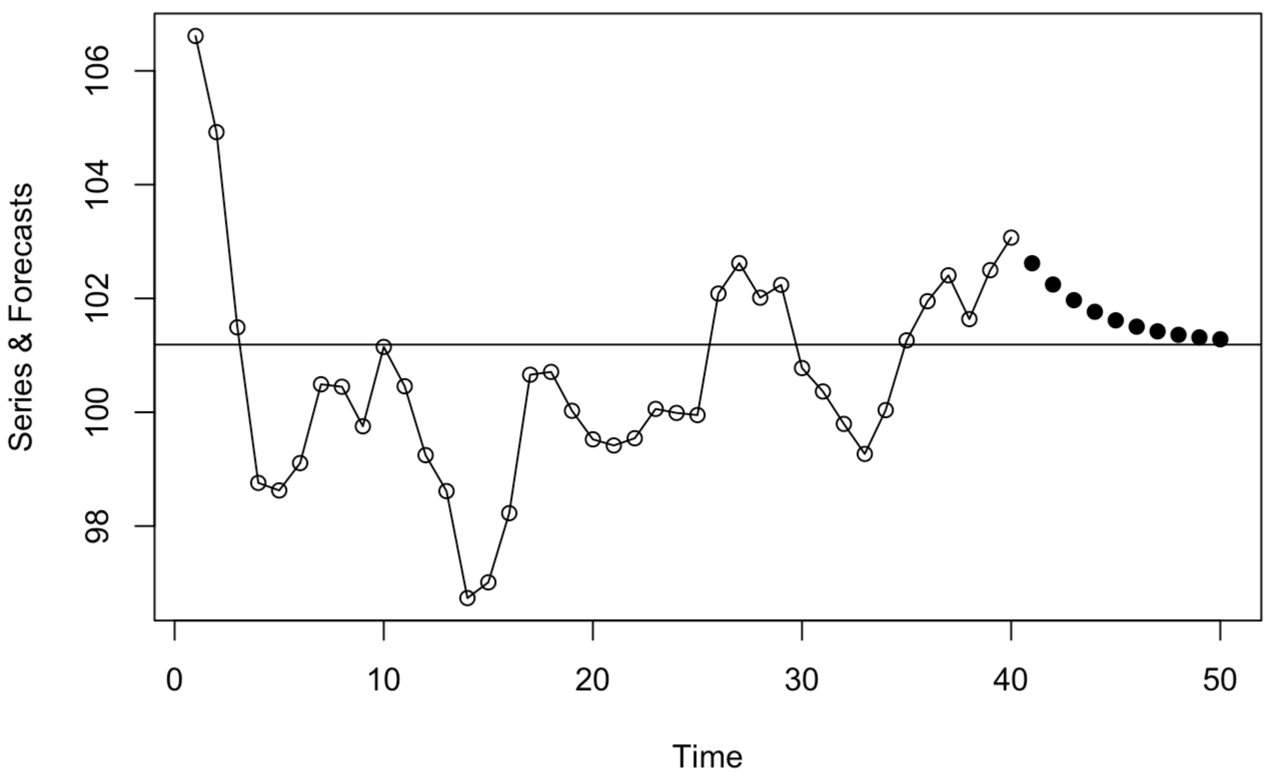
e

The predicted series decays towards mean. However, the actual values are pretty far apart. However, the predictions still fall between 95% CI.

```
set.seed(456)
series=arima.sim(n=50,list(ar=0.7,ma=0.5))+100
actual=window(series,start=41); series=window(series,end=40)
model=arima(series,order=c(1,0,1)); model
```

```
##
## Call:
## arima(x = series, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.7390  0.5978   101.1884
## s.e.  0.1425  0.1530     0.9530
##
## sigma^2 estimated as 0.9824:  log likelihood = -57.38,  aic = 120.77
```

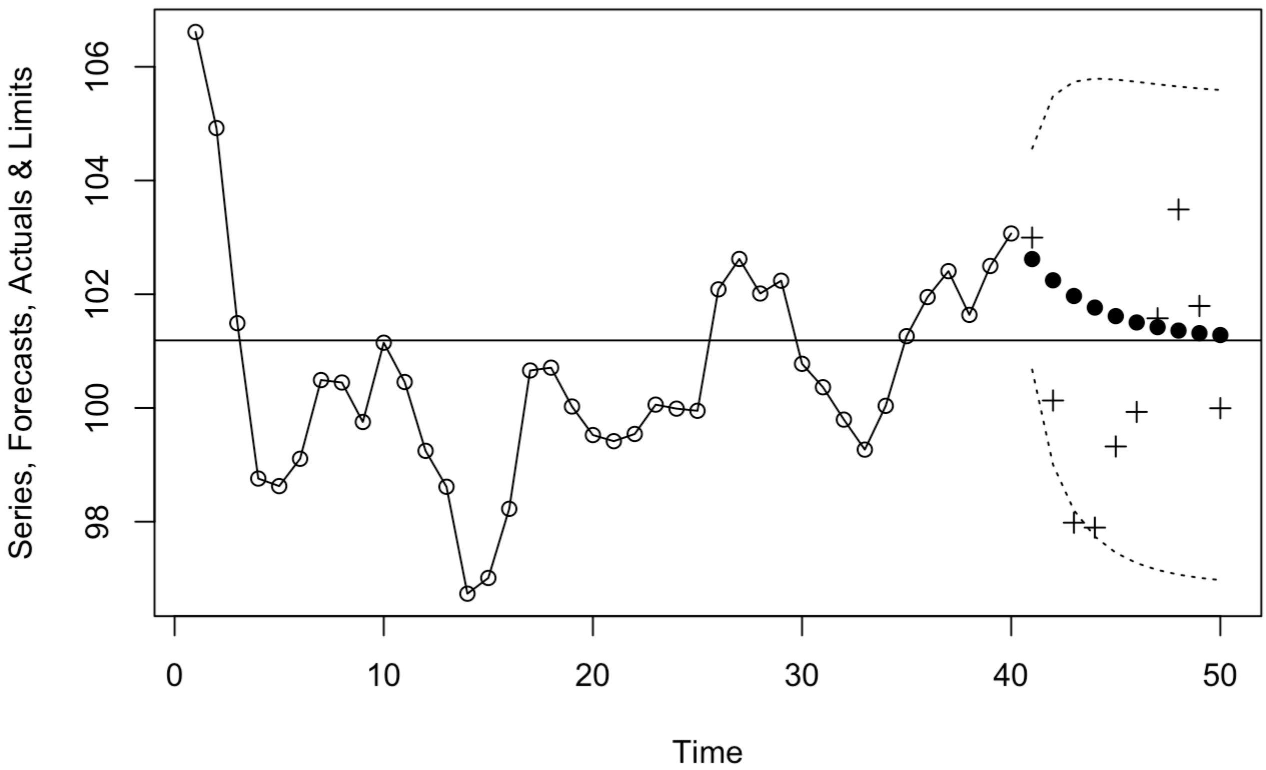
```
result=plot(model,n.ahead=10,ylab='Series & Forecasts',col=NULL,pch=19)
abline(h=coef(model)[names(coef(model))=='intercept'])
```




```
forecast=result$pred; cbind(actual,forecast)
```

```
## Time Series:
## Start = 41
## End = 50
## Frequency = 1
##       actual forecast
## 41 102.99595 102.6184
## 42 100.13103 102.2451
## 43  97.98230 101.9694
## 44  97.89602 101.7655
## 45  99.32286 101.6149
## 46  99.93042 101.5036
## 47 101.57683 101.4213
## 48 103.49073 101.3606
## 49 101.79379 101.3156
## 50  99.99537 101.2824
```

```
plot(model,n.ahead=10,ylab='Series, Forecasts, Actuals & Limits',pch=19)
points(x=(41:50),y=actual,pch=3)
abline(h=coef(model)[names(coef(model))=='intercept'])
```



Ex 9.16

```
set.seed(123)
series=(arima.sim(n=45,list(order=c(0,2,2),ma=c(-1,0.75)))[-1])[-1]
actual=window(series,start=41); series=window(series,end=40)
```

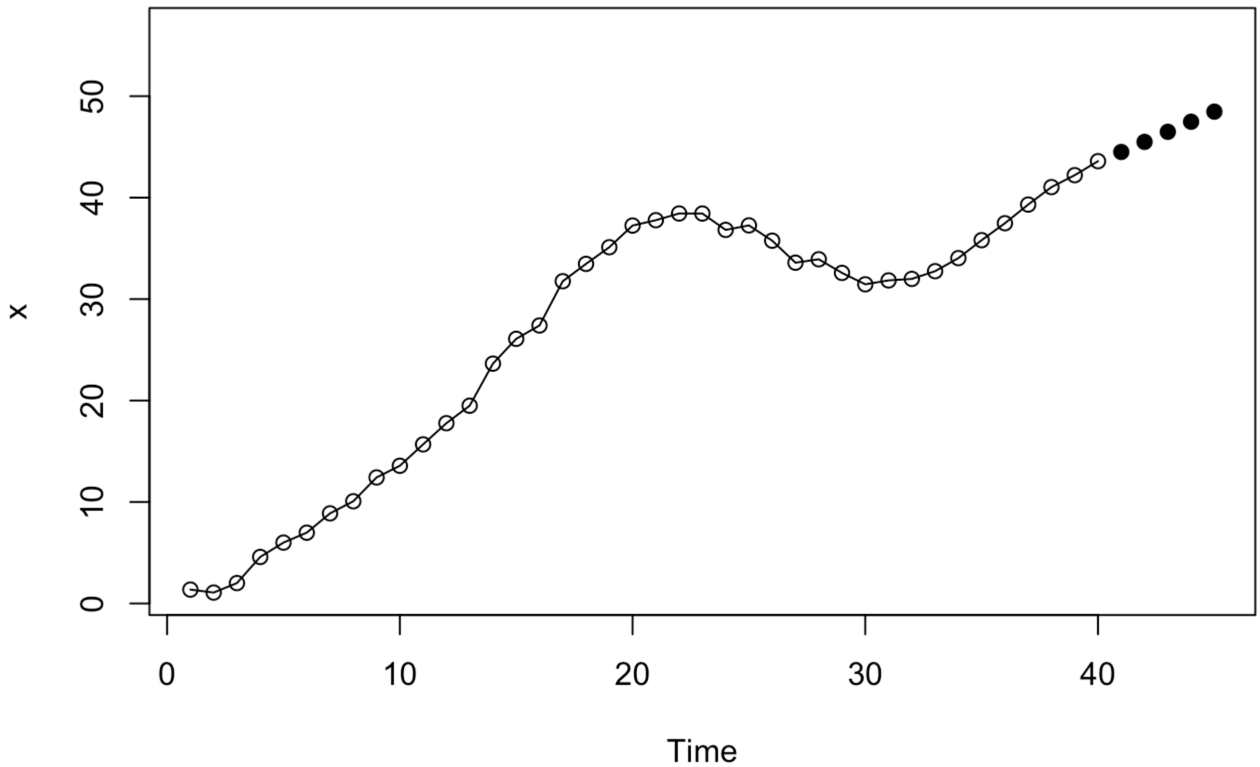
a

```
model=arima(series,order=c(0,2,2))
model
```

```
##
## Call:
## arima(x = series, order = c(0, 2, 2))
##
## Coefficients:
##           ma1     ma2
##       -1.1284  1.0000
## s.e.   0.1394  0.2126
##
## sigma^2 estimated as 0.6732:  log likelihood = -49.59,  aic = 103.18
```

b

```
result=plot(model,n.ahead=5,col=NULL,pch=19)
```

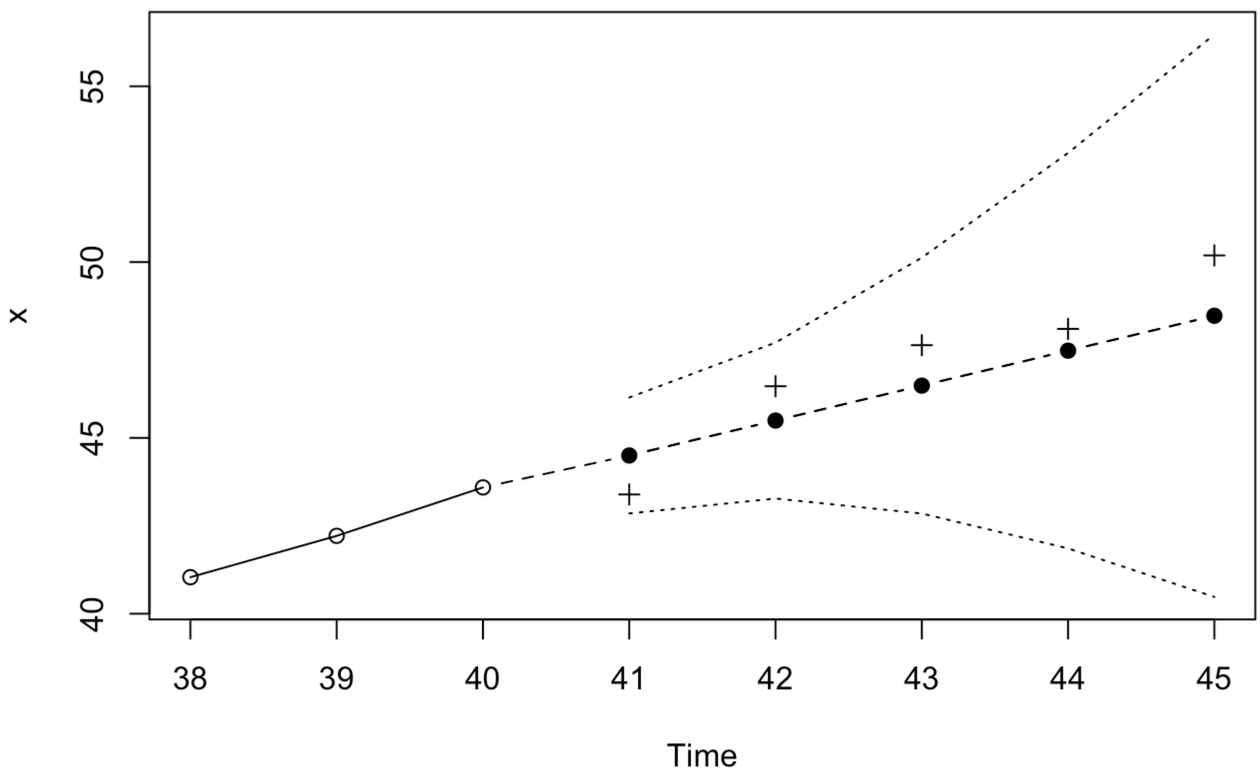


c

```
forecast=result$pred
cbind(actual,forecast)
```

```
## Time Series:
## Start = 41
## End = 45
## Frequency = 1
##      actual forecast
## 41 43.39144 44.50005
## 42 46.46910 45.49358
## 43 47.63673 46.48711
## 44 48.10000 47.48064
## 45 50.18946 48.47417
```

```
plot(model,n1=38,n.ahead=5, pch=19)
points(x=seq(41,45),y=actual,pch=3)
```



The forecast limits spread out as the lead time increases.

```
lower=result$lpi; upper=result$upi; cbind(lower,actual,upper)
```

```
## Time Series:
## Start = 41
## End = 45
## Frequency = 1
##       lower   actual    upper
## 41 42.85156 43.39144 46.14854
## 42 43.27639 46.46910 47.71077
## 43 42.84883 47.63673 50.12539
## 44 41.85830 48.10000 53.10298
## 45 40.47626 50.18946 56.47209
```

The lead 1 forecast is a little above the lower forecast limit so that all of the forecasts are within the 95% limits in this simulation.

Ex 10.8

a

```
data(co2)
month.=season(co2)
trend=time(co2)
model=lm(co2~month.+trend)
summary(model)
```

```
##
## Call:
## lm(formula = co2 ~ month. + trend)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73874 -0.59689 -0.06947  0.54086  2.15539
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -3290.5412    44.1790 -74.482  < 2e-16 ***
## month.February      0.6682     0.3424   1.952 0.053320 .
## month.March         0.9637     0.3424   2.815 0.005715 **
## month.April         1.2311     0.3424   3.595 0.000473 ***
## month.May           1.5275     0.3424   4.460 1.87e-05 ***
## month.June         -0.6761     0.3425  -1.974 0.050696 .
## month.July         -7.2851     0.3426 -21.267  < 2e-16 ***
## month.August      -13.4414     0.3426 -39.232  < 2e-16 ***
## month.September   -12.8205     0.3427 -37.411  < 2e-16 ***
## month.October      -8.2604     0.3428 -24.099  < 2e-16 ***
## month.November     -3.9277     0.3429 -11.455  < 2e-16 ***
## month.December     -1.3367     0.3430  -3.897 0.000161 ***
## trend               1.8321     0.0221  82.899  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8029 on 119 degrees of freedom
## Multiple R-squared:  0.9902, Adjusted R-squared:  0.9892
## F-statistic: 997.7 on 12 and 119 DF,  p-value: < 2.2e-16
```

All of the regression coefficients are statistically significant except for the seasonal effects for February and June. These have p-values just above 0.05.

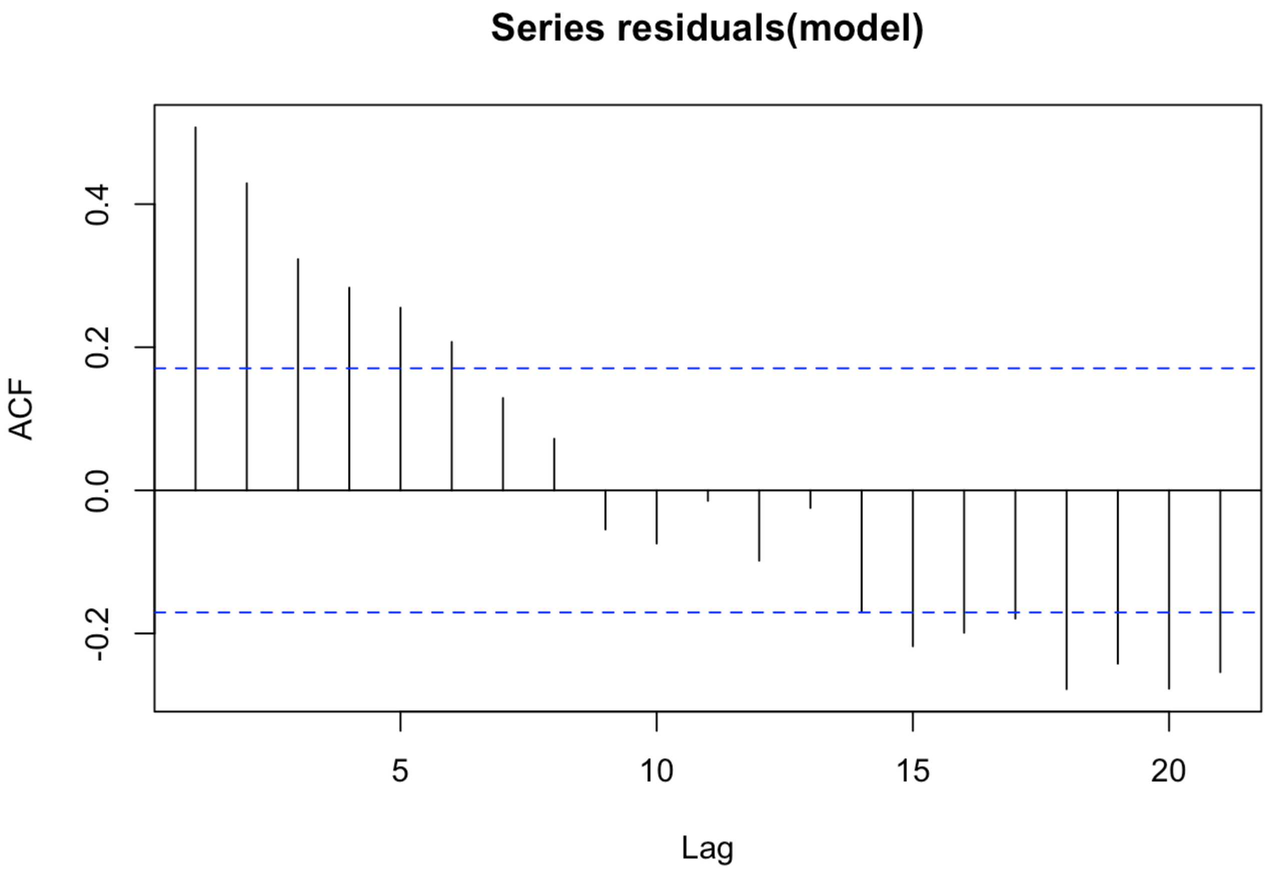
b

99.02%

c

There is a seasonal trend.

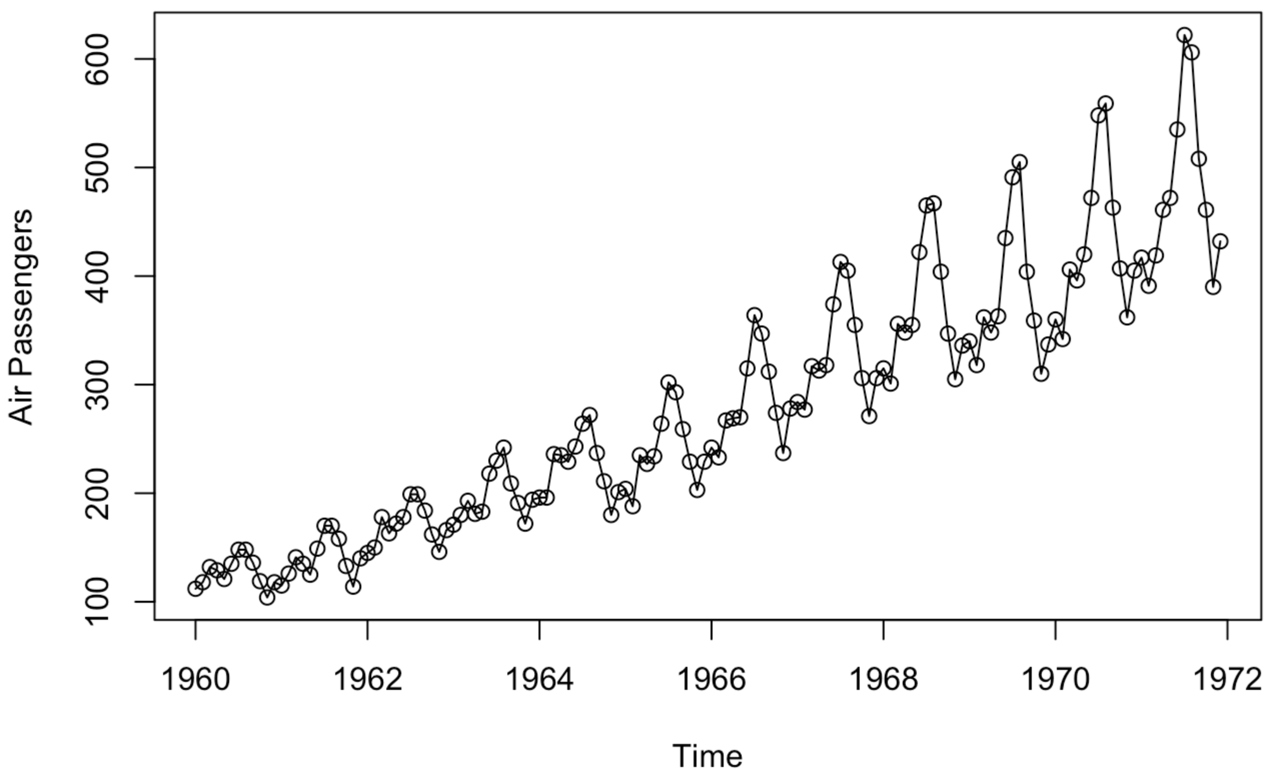
```
acf(residuals(model))
```



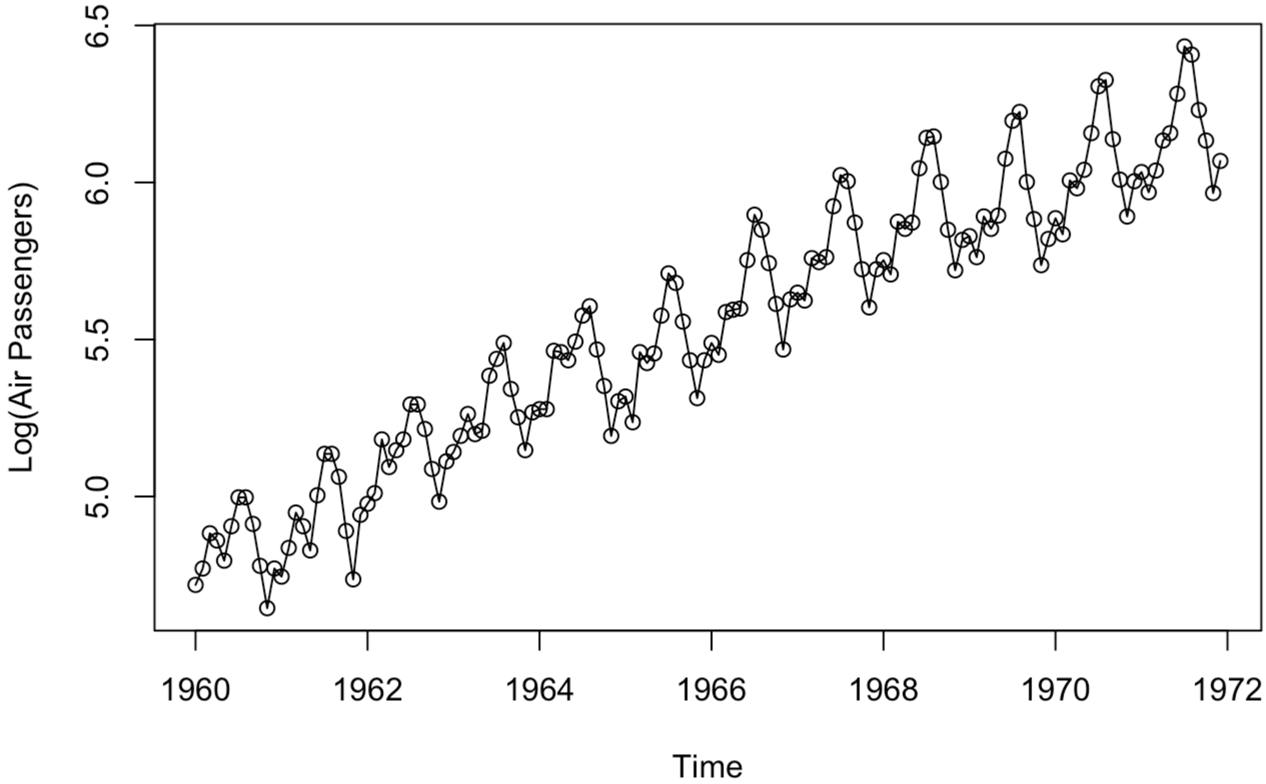
Ex 10.9

a

```
data(airpass); plot(airpass, type='o',ylab='Air Passengers')
```



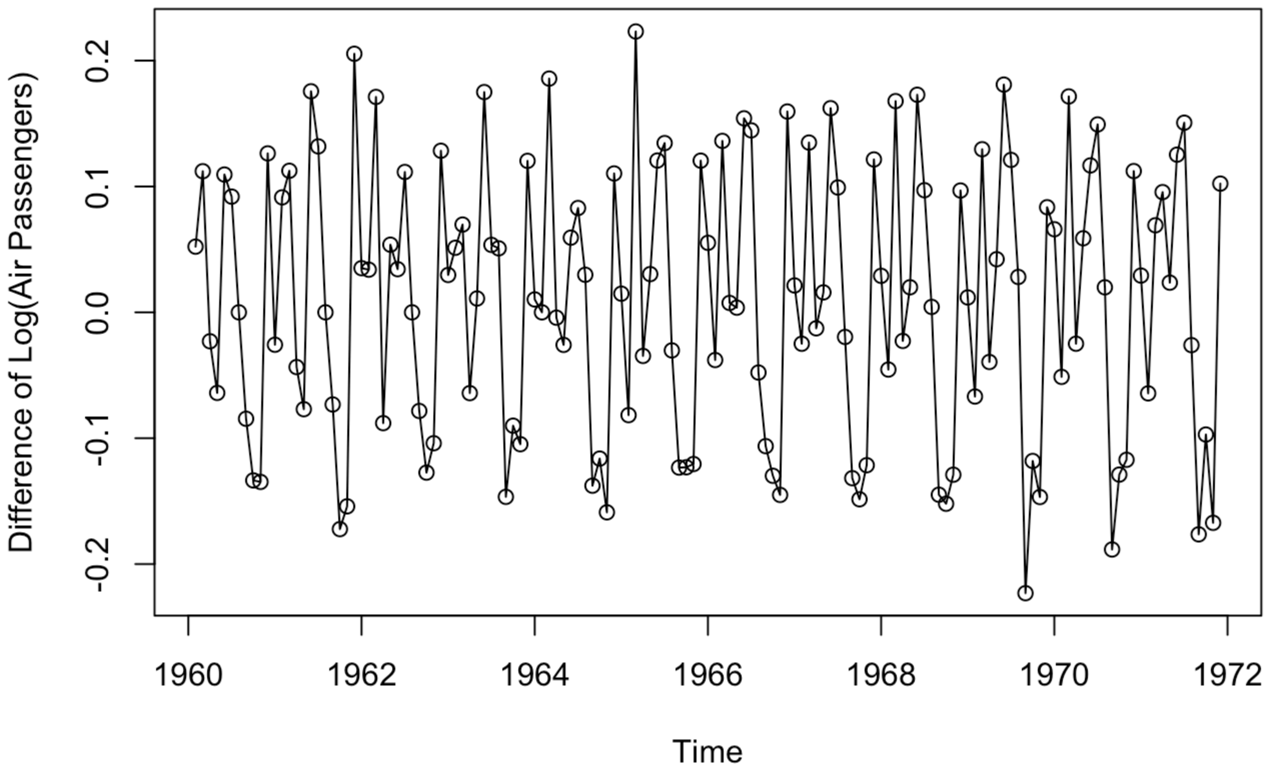

```
plot(log(airpass), type='o',ylab='Log(Air Passengers)')
```



The graph of the logarithms displays a much more constant variation around the upward "trend."

b

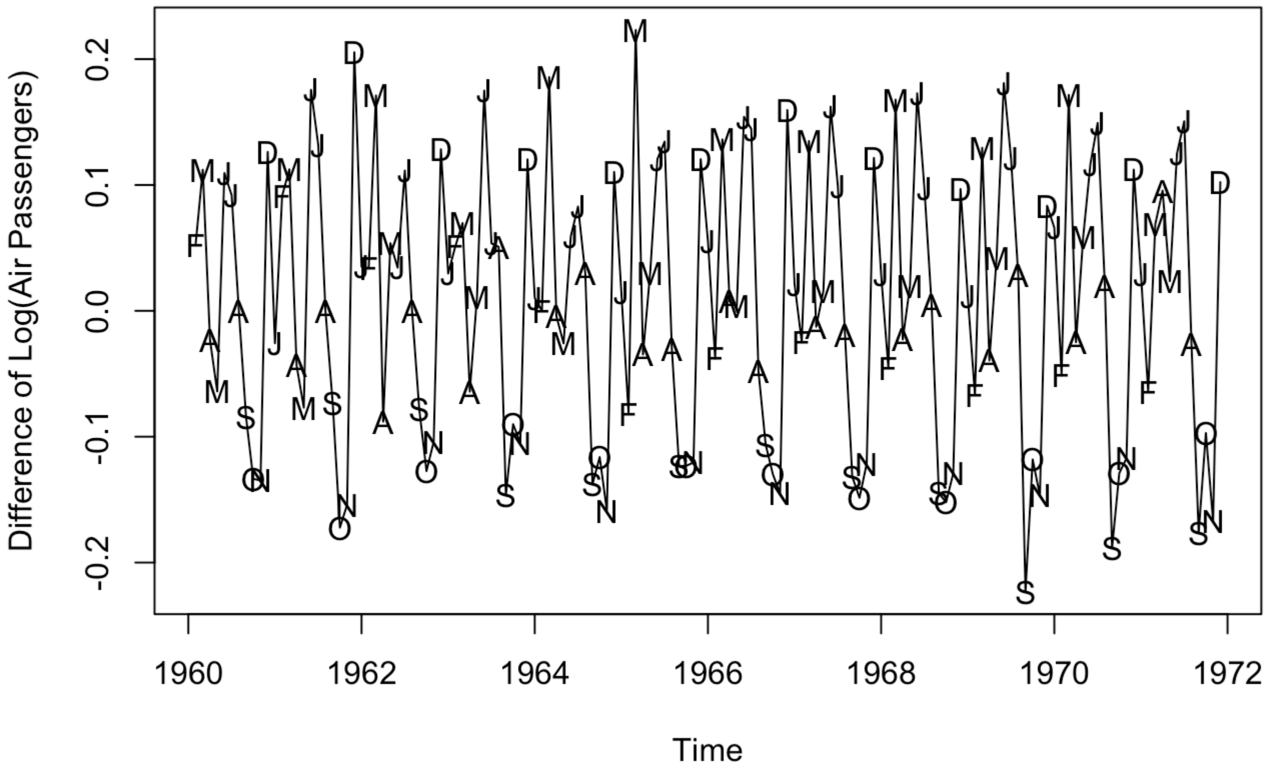
```
plot(diff(log(airpass)),type='o',ylab='Difference of Log(Air Passengers)')
```



c

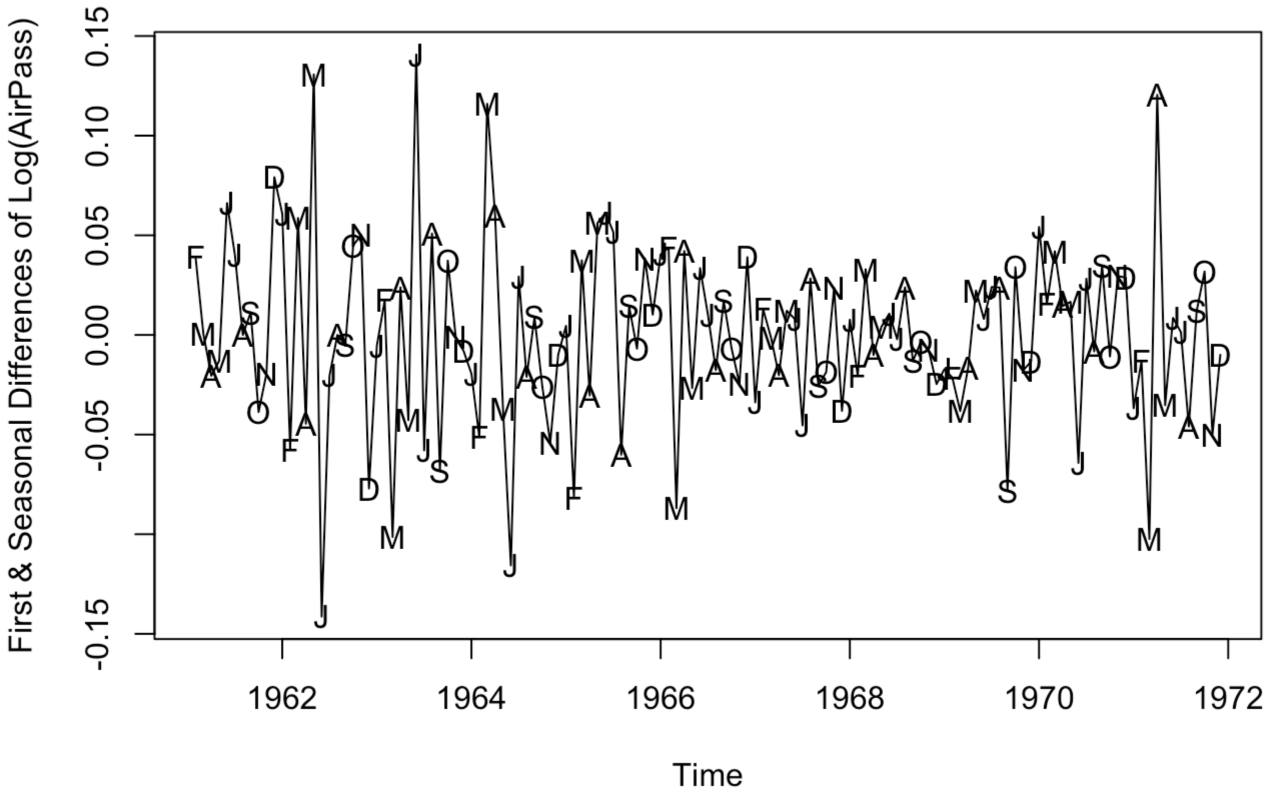
```
plot(diff(log(airpass)),type='l',ylab='Difference of Log(Air Passengers)')

points(diff(log(airpass)),x=time(diff(log(airpass))),
pch=as.vector(season(diff(log(airpass)))))
```



The seasonality can be observed by looking at the plotting symbols carefully. Septembers, Octobers, and Novembers are mostly at the low points and Decembers mostly at the high points.

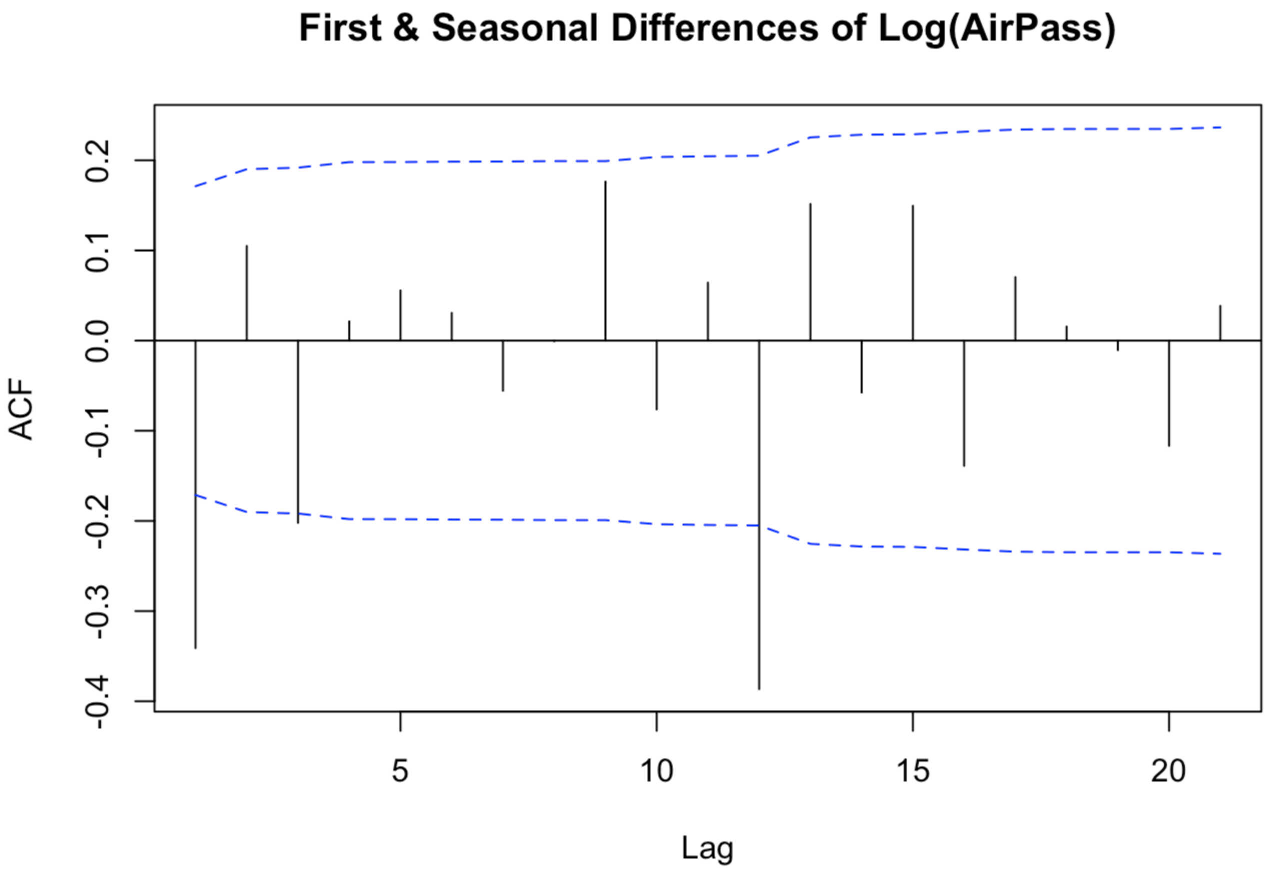
```
plot(diff(diff(log(airpass)),lag=12),type='l',
ylab='First & Seasonal Differences of Log(AirPass)')
points(diff(diff(log(airpass)),lag=12),x=time(diff(diff(log(airpass)),lag=12)),pch=a
s.vector(season(diff(diff(log(airpass)),lag=12))))
```



Some Decembers are high and some low. Similarly, some Octobers are high and some low.

d

```
acf(as.vector(diff(diff(log(airpass)),lag=12)),ci.type='ma',
main='First & Seasonal Differences of Log(AirPass)')
```



Although there is a "significant" autocorrelation at lag 3, the most prominent autocorrelations are at lags 1 and 12. We need to investigate airline model further.

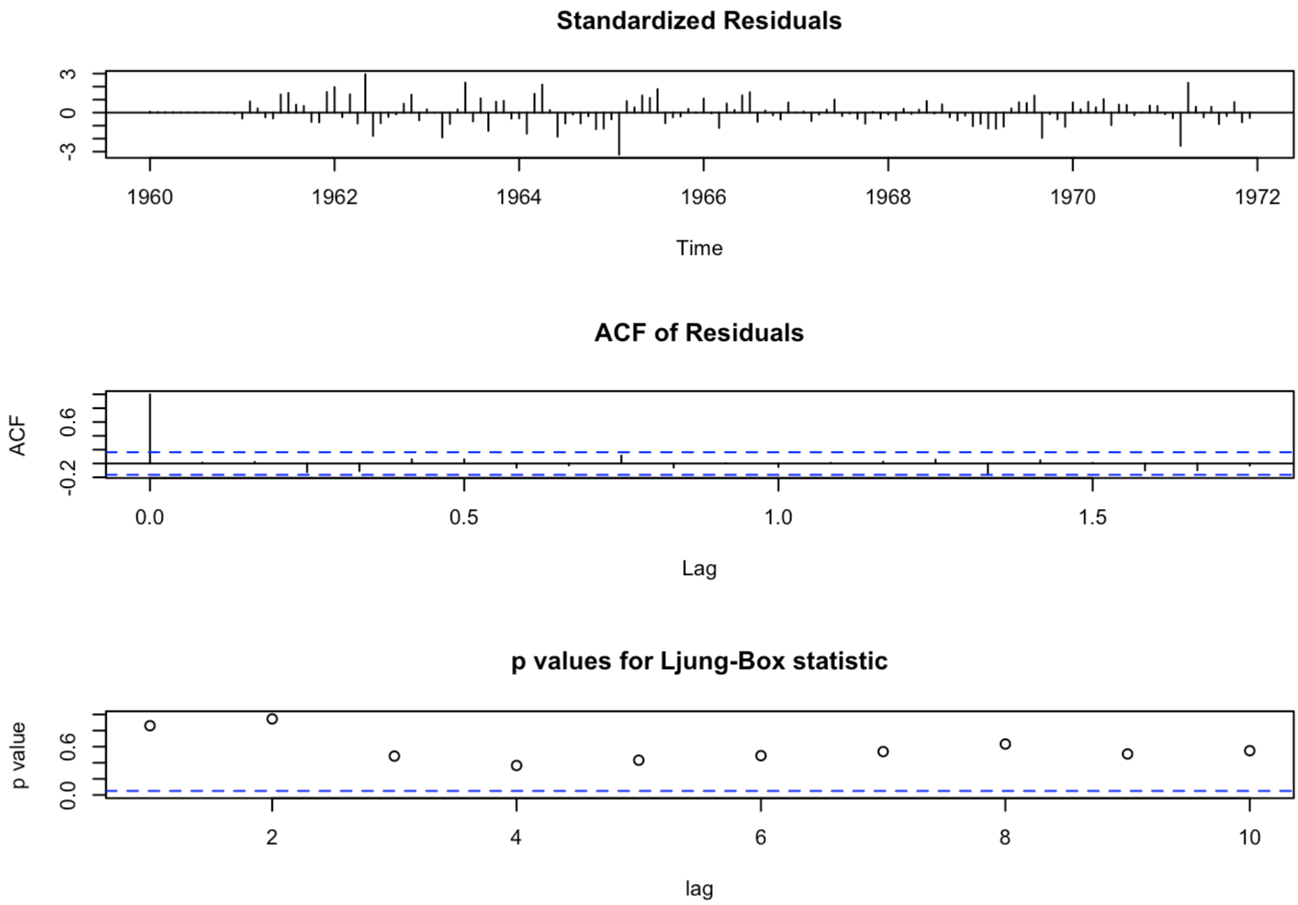
```
model=arima(log(airpass),order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12))
model
```

```
##
## Call:
## arima(x = log(airpass), order = c(0, 1, 1), seasonal = list(order = c(0, 1,
##     1), period = 12))
##
## Coefficients:
##           ma1     sma1
##       -0.4018  -0.5569
## s.e.   0.0896   0.0731
##
## sigma^2 estimated as 0.001348:  log likelihood = 244.7,  aic = -485.4
```

Notice that both the seasonal and nonsrasonal ma parameters are significant.

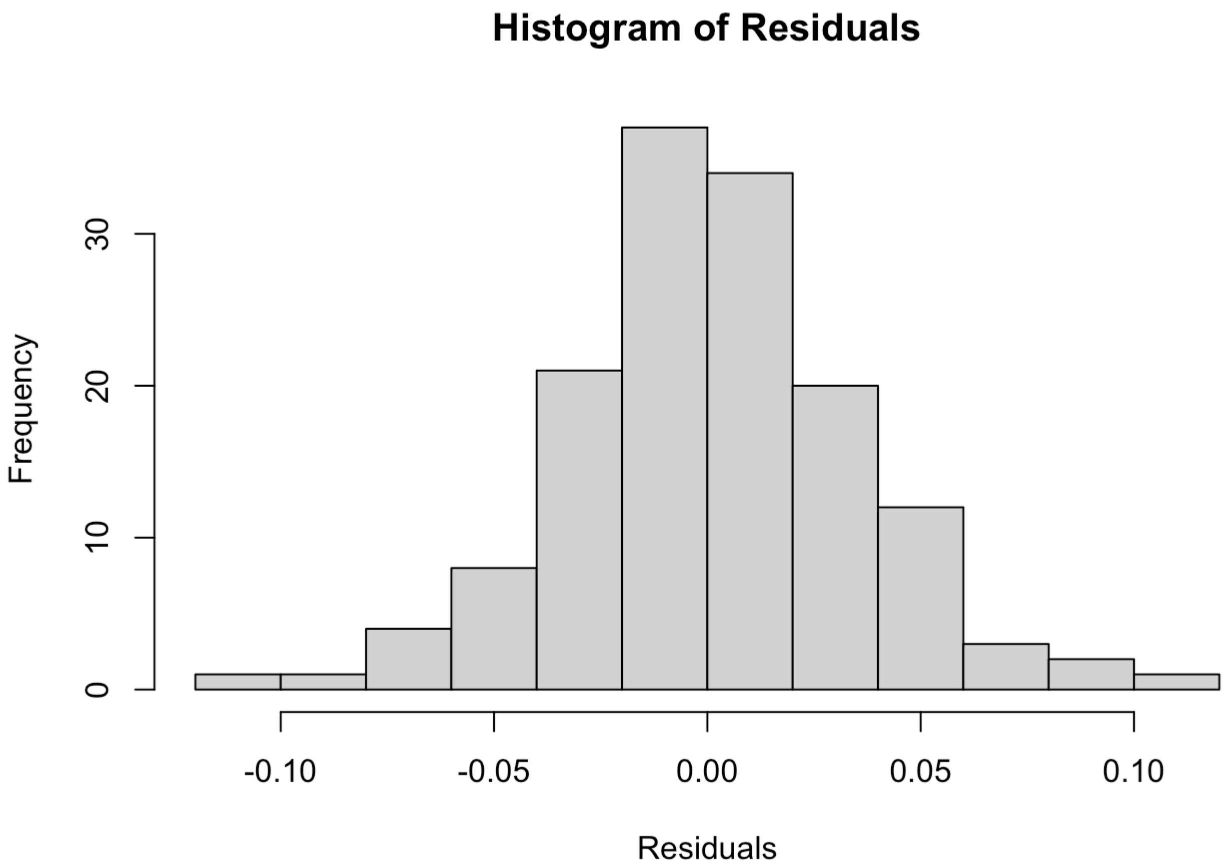
f

```
tsdiag(model)
```

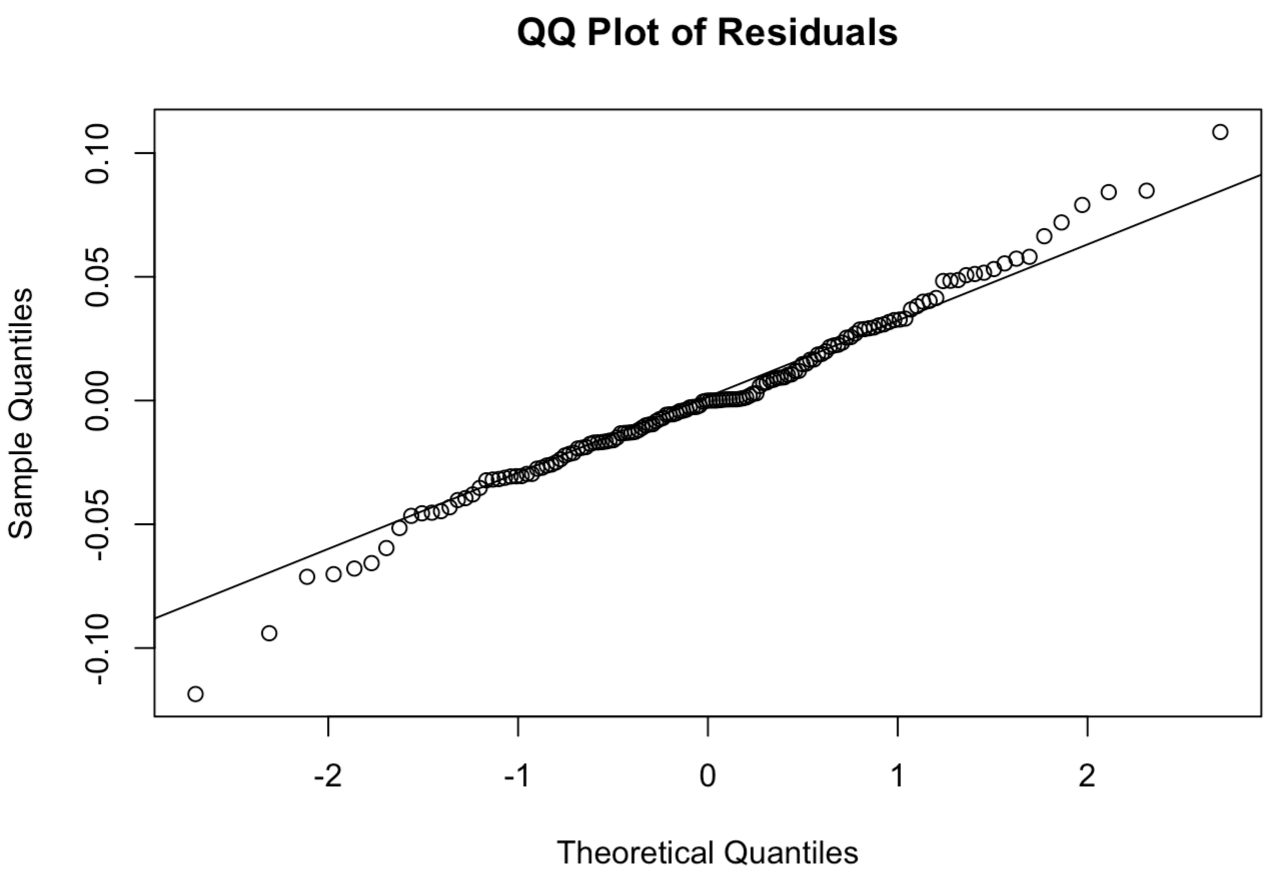


None of these three plots indicate difficulties with the model. There are no outliers and little autocorrelation in the residuals, both individually and jointly. looking at normality.

```
hist(residuals(model), xlab='Residuals', main='Histogram of Residuals')
```




```
qqnorm(residuals(model), main="QQ Plot of Residuals")
qqline(residuals(model))
```



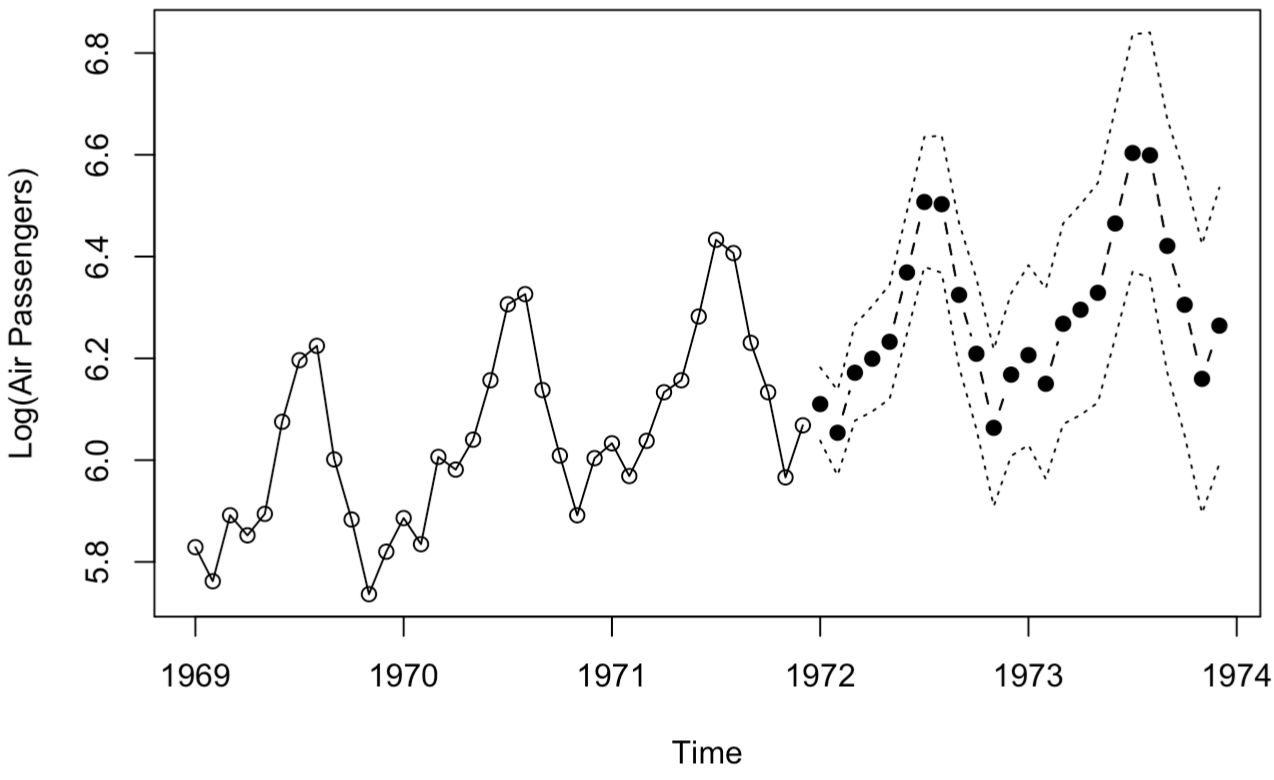
The distribution of the residuals is symmetric but the Q-Q plot indicates that the tails are lighter than a normal distribution. Let's investigate using Shapiro-Wilk test.

```
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.98637, p-value = 0.1674
```

The Shapiro-Wilk test does not reject normality of the error terms at any of the usual significance levels and we proceed to use the model for forecasting.

```
plot(model, n1=c(1969, 1), n.ahead=24, pch=19, ylab='Log(Air Passengers)')
```



The forecasts follow the seasonal and upward "trend" of the time series. The forecast limits provide us with a clear measure of the uncertainty in the forecasts. For completeness, we also plot the forecasts and limits in original terms.