Q1: Let $I_n = \{X_1, X_2, \ldots, X_n\}$. We have

$$X_{n+1} = \phi_1 X_n + \phi_2 X_{n-1} + Z_{n+1},$$

$$X_{n+1|I_n} = \mathrm{E}\left[X_{n+1}|I_n\right] = \phi_1 X_n + \phi_2 X_{n-1},$$

$$X_{n+2} = \phi_1 X_{n+1} + \phi_2 X_n + Z_{n+2} = (\phi_1^2 + \phi_2)X_n + \phi_1\phi_2 X_{n-1} + \phi_1 Z_{n+1} + Z_{n+2},$$

and

$$X_{n+2|I_n} = \mathrm{E}\left[X_{n+2}|I_n\right] = (\phi_1^2 + \phi_2)X_n + \phi_1\phi_2 X_{n-1}.$$

Therefore,

$$\hat{X}_{n+2|I_n} = (\hat{\phi}_1^2 + \hat{\phi}_2)X_n + \hat{\phi}_1\hat{\phi}_2 X_{n-1}.$$

Since $\hat{\phi}_1 \approx \phi_1$ and $\hat{\phi}_2 \approx \phi_2$, we have

$$\hat{e}_{n+2|I_n} = X_{n+2} - \hat{X}_{n+2|I_n} \approx \phi_1 Z_{n+1} + Z_{n+2},$$

and the variance of the forecast error is:

$$\mathrm{Var}\left(\hat{e}_{n+2|I_n}\right) \approx \mathrm{Var}\left(\phi_1 Z_{n+1} + Z_{n+2}\right) = (1 + \phi_1^2)\sigma^2.$$

Q2: Let $I_n = \{X_1, X_2, \ldots, X_n\}$. Since $\{X_t\}$ follows the ARIMA(2,1,0), its first difference $D_n = X_n - X_{n-1}$ follows the AR(2) as follows: $D_n = \phi_1 D_{n-1} + \phi_2 D_{n-2} + Z_t$. We have

$$X_n - X_{n-1} = \phi_1(X_{n-1} - X_{n-2}) + \phi_2(X_{n-2} - X_{n-3}) + Z_t,$$

which implies:

$$X_n = (\phi_1 + 1)X_{n-1} + (\phi_2 - \phi_1)X_{n-2} - \phi_2 X_{n-3} + Z_n.$$

Follow the same steps in Question 1, we have

$$X_{n+1} = (\phi_1 + 1)X_n + (\phi_2 - \phi_1)X_{n-1} - \phi_2 X_{n-2} + Z_{n+1},$$

$$X_{n+1|I_n} = \mathrm{E}\left[X_{n+1}|I_n\right] = (\phi_1 + 1)X_n + (\phi_2 - \phi_1)X_{n-1} - \phi_2 X_{n-2},$$

and

$$\begin{aligned} X_{n+2} =& (\phi_1 + 1)X_{n+1} + (\phi_2 - \phi_1)X_n - \phi_2 X_{n-1} + Z_{n+2} \\ =& \big((\phi_1 + 1)^2 X_n + (\phi_1 + 1)(\phi_2 - \phi_1)X_{n-1} - (\phi_1 + 1)\phi_2 X_{n-2} + (\phi_1 + 1)Z_{n+1}\big) \\ &+ (\phi_2 - \phi_1)X_n - \phi_2 X_{n-1} + Z_{n+2} \\ =& C_1 X_n + C_2 X_{n-1} + C_3 X_{n-2} + (\phi_1 + 1)Z_{n+1} + Z_{n+2}, \end{aligned}$$

where $C_1 = (\phi_1 + 1)^2 + \phi_2 - \phi_1$, $C_2 = (\phi_1 + 1)(\phi_2 - \phi_1) - \phi_2$ and $C_3 = -(\phi_1 + 1)\phi_2$. Therefore,

$$X_{n+2|I_n} = \mathrm{E}\left[X_{n+2}|I_n\right] = C_1 X_n + C_2 X_{n-1} + C_3 X_{n-2}.$$

Since $\hat{\phi}_1 \approx \phi_1$ and $\hat{\phi}_2 \approx \phi_2$, we have

$$\hat{X}_{n+2|I_n} \approx C_1 X_n + C_2 X_{n-1} + C_3 X_{n-2},$$

and

$$\hat{e}_{n+2} = X_{n+2} - \hat{X}_{n+2|I_n} \approx (\phi_1 + 1)Z_{n+1} + Z_{n+2}.$$

Then the variance of the forecast error is:

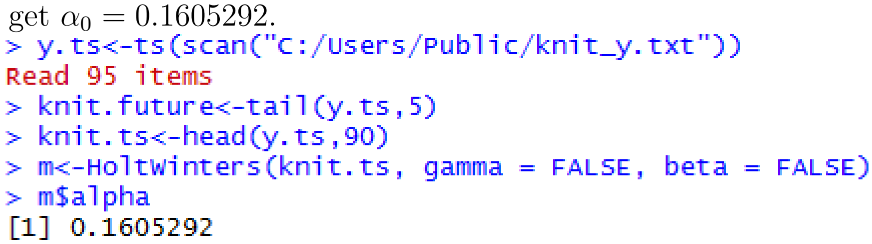
$$\operatorname{Var} \hat{e}_{n+2} \approx \operatorname{Var}\big((\phi_1 + 1)Z_{n+1} + Z_{n+2}\big) = (\phi_1^2 + 2\phi_1 + 2)\sigma^2.$$

Q3:

(a)

```
y.ts<-ts(scan("C:/Users/Public/knit_y.txt"))
knit.future<-tail(y.ts,5)
knit.ts<-head(y.ts,90)
m<-HoltWinters(knit.ts, gamma = FALSE, beta = FALSE)
m$alpha
```

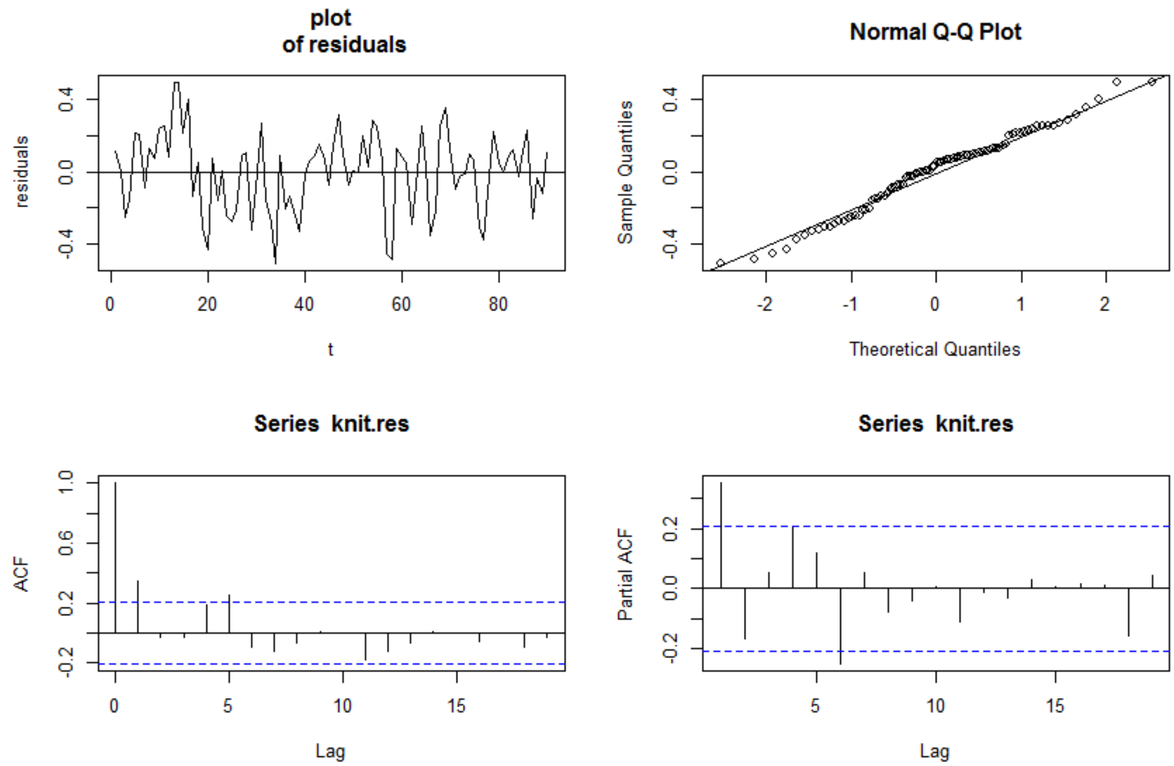
We get $\alpha_0 = 0.1605292$.



```
> y.ts<-ts(scan("C:/Users/Public/knit_y.txt"))
Read 95 items
> knit.future<-tail(y.ts,5)
> knit.ts<-head(y.ts,90)
> m<-HoltWinters(knit.ts, gamma = FALSE, beta = FALSE)
> m$alpha
[1] 0.1605292
```

(b) To fit a constant mean model $Y_t = \beta_0 + \varepsilon_t$ with stationary correlated residuals, we first estimate β_0 by the mean of the historical data (knit).

```
knit.mean<-mean(knit)
knit.res<-knit.ts-knit.mean
par(mfrow=c(2,2))
plot(knit.res,xlab="t",ylab="residuals",main="plot of residuals")
abline(a=0,b=0)
qqnorm(knit.res)
qqline(knit.res)
acf(knit.res)
pacf(knit.res)
```

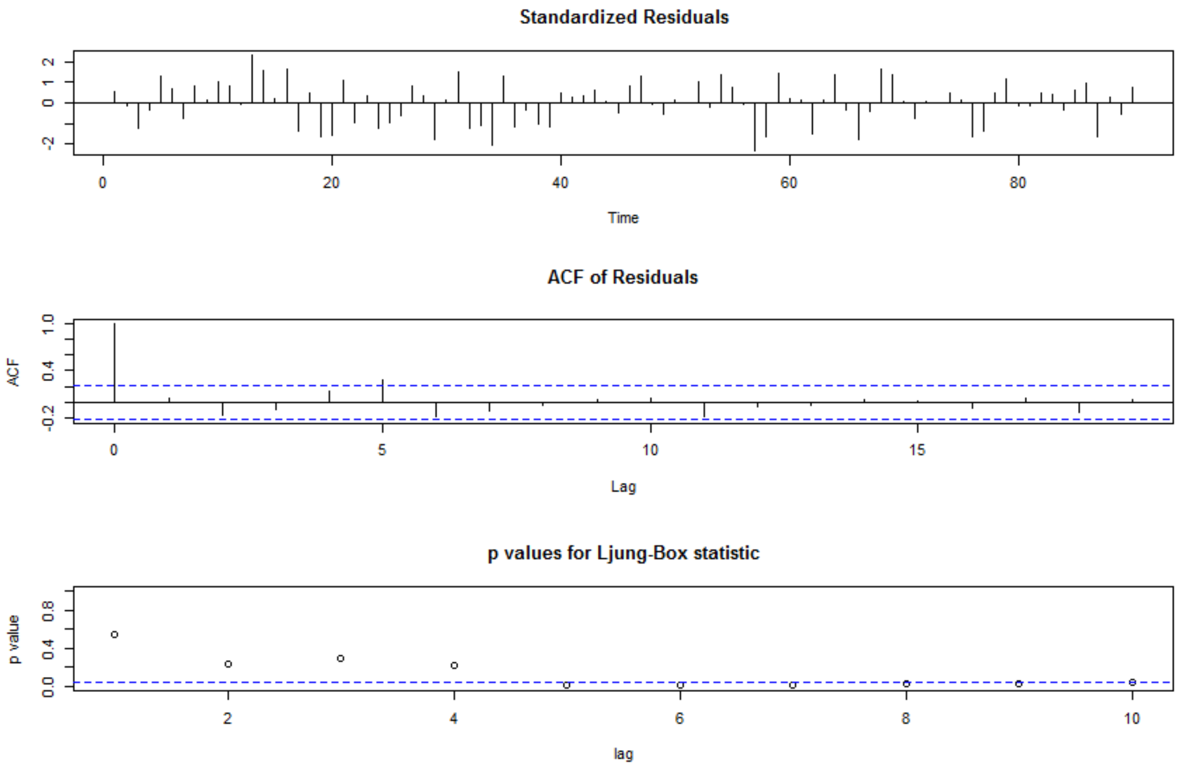


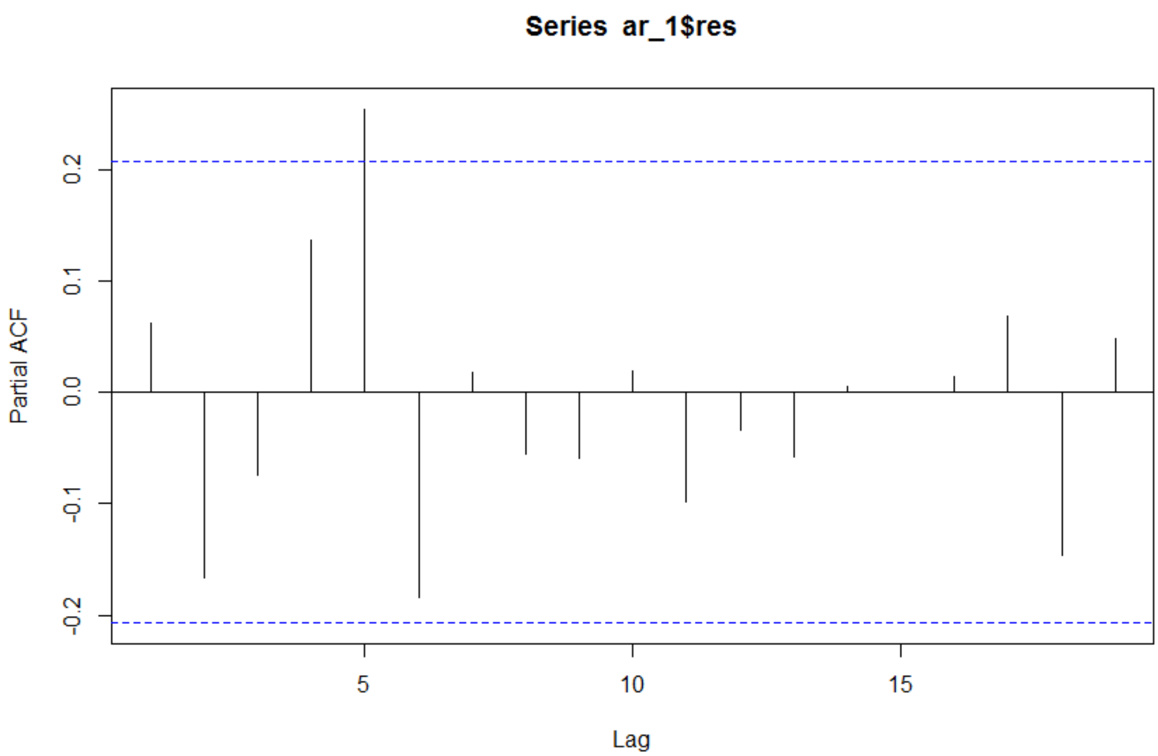
We find the residual time series shows no obvious deviations. The sample ACF shows a clear lag 1 correlation while the autocorrelations at the remaining lags are insignificant (except for lag 5 correlation, which is slightly significant but may be spurious). The sample PACF only shows a slightly correlation at lag 1, 4, and 6. Therefore we can try AR(1), MA(1) or even ARMA(1,1).

- **AR(1)**

```
ar_1=arima(knit.res,order=c(1,0,0), method="ML")
ar_1
tsdiag(ar_1)
```

```
Call:
arima(x = knit.res, order = c(1, 0, 0), method = "ML")

Coefficients:
         ar1  intercept
      0.3484     0.0013
s.e.  0.0983     0.0325

sigma^2 estimated as 0.04091:  log likelihood = 16.07,  aic = -26.14
```

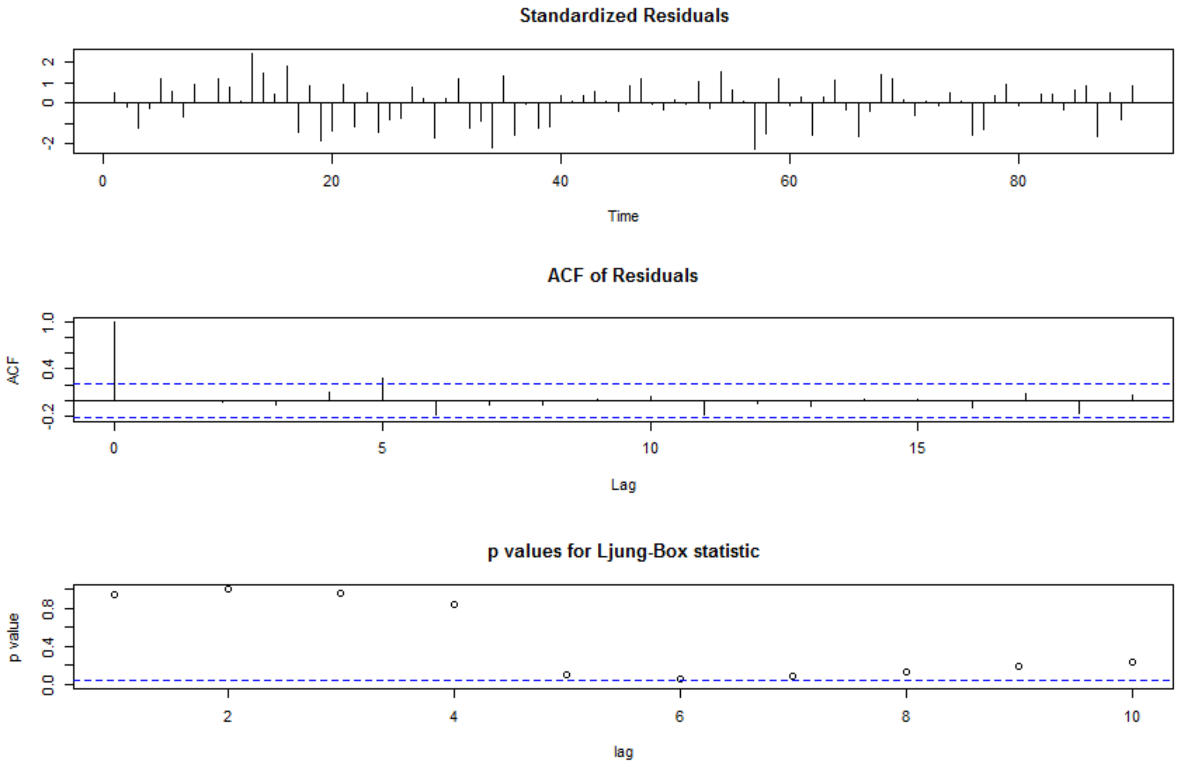


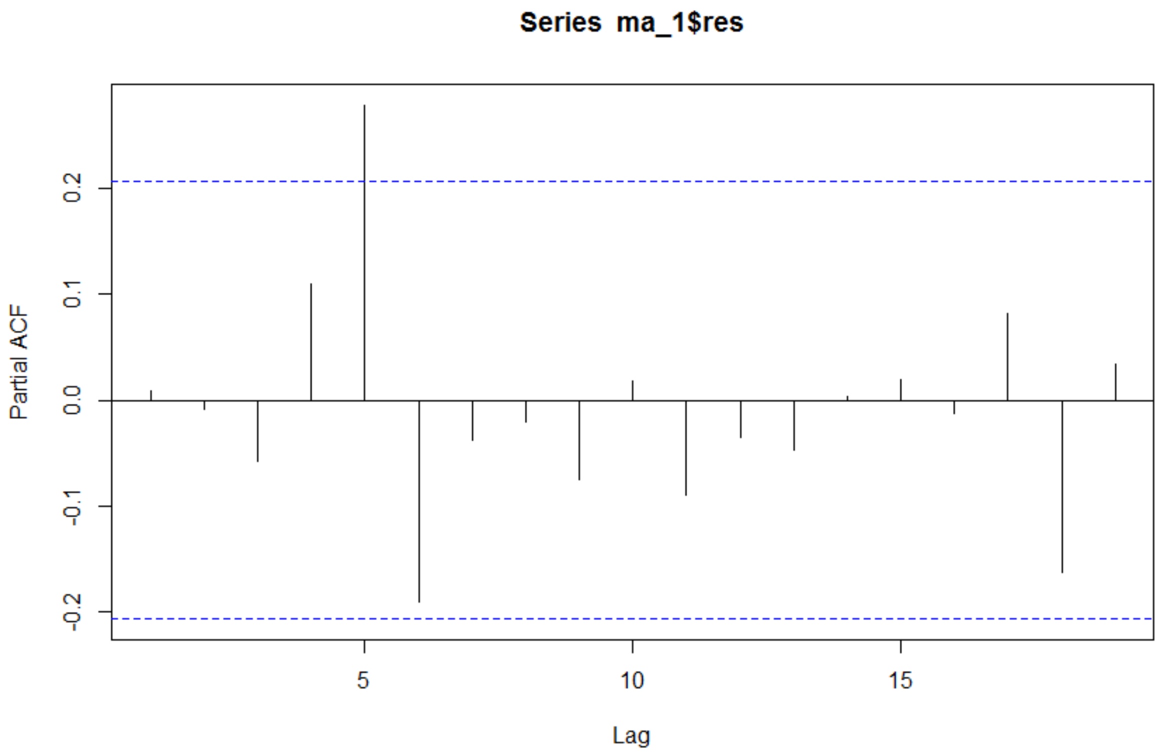


- **MA(1)**

```
ma_1=arima(knit.res,order=c(0,0,1), method="ML")
ma_1
tsdiag(ma_1)
```

```
Call:
arima(x = knit.res, order = c(0, 0, 1), method = "ML")

Coefficients:
         ma1  intercept
      0.4056     0.0011
s.e.  0.0970     0.0295

sigma^2 estimated as 0.03986:  log likelihood = 17.21,  aic = -28.42
```

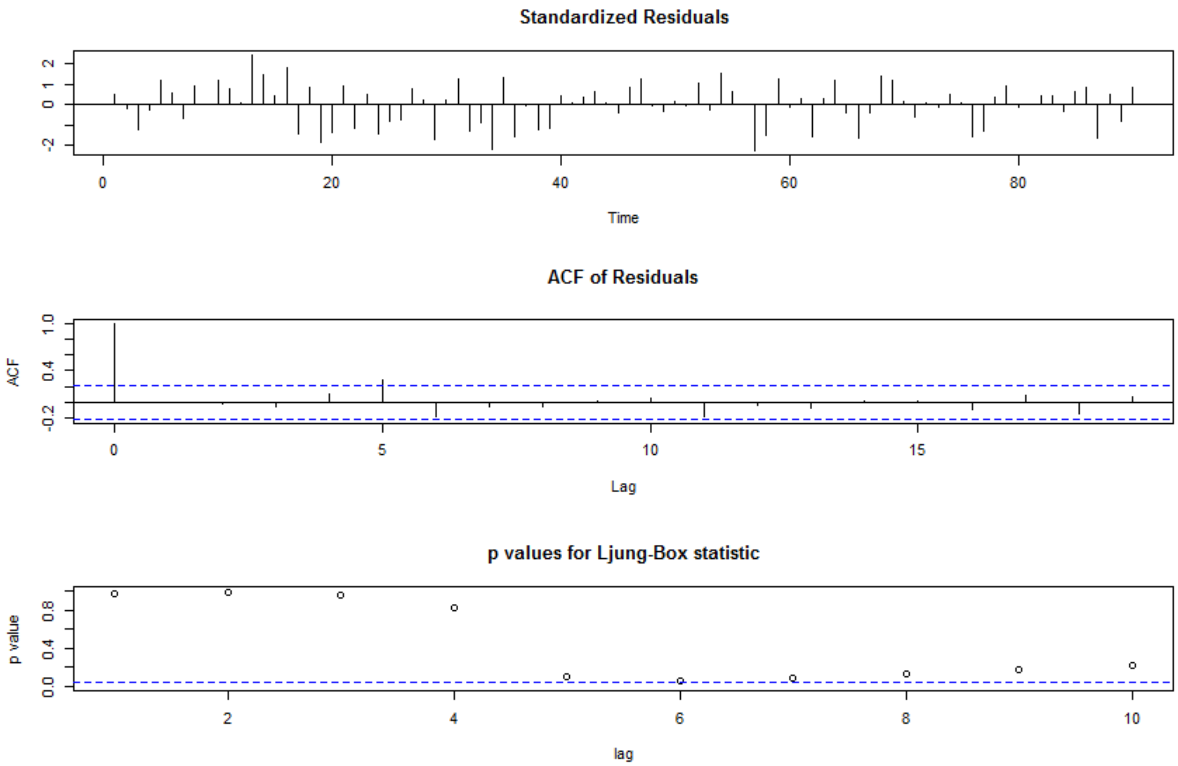


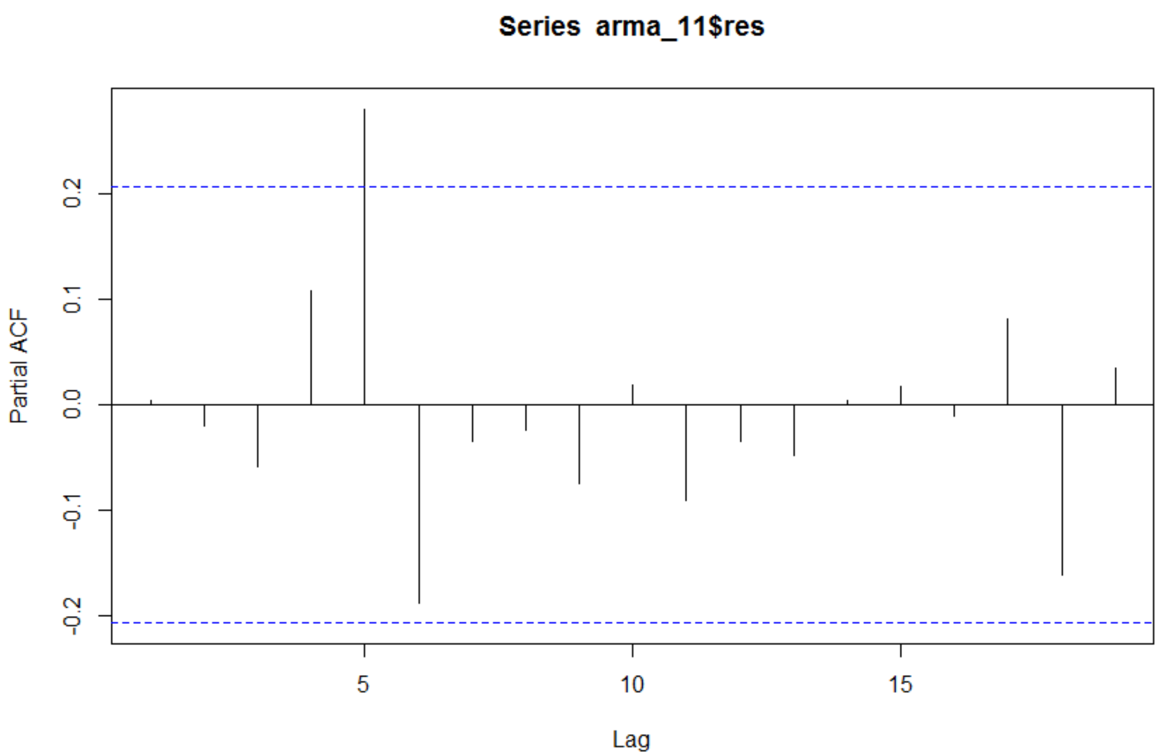


- **ARMA(1,1)**

```
arma_11=arima(knit.res,order=c(1,0,1), method="ML")
arma_11
tsdiag(arma_11)
```

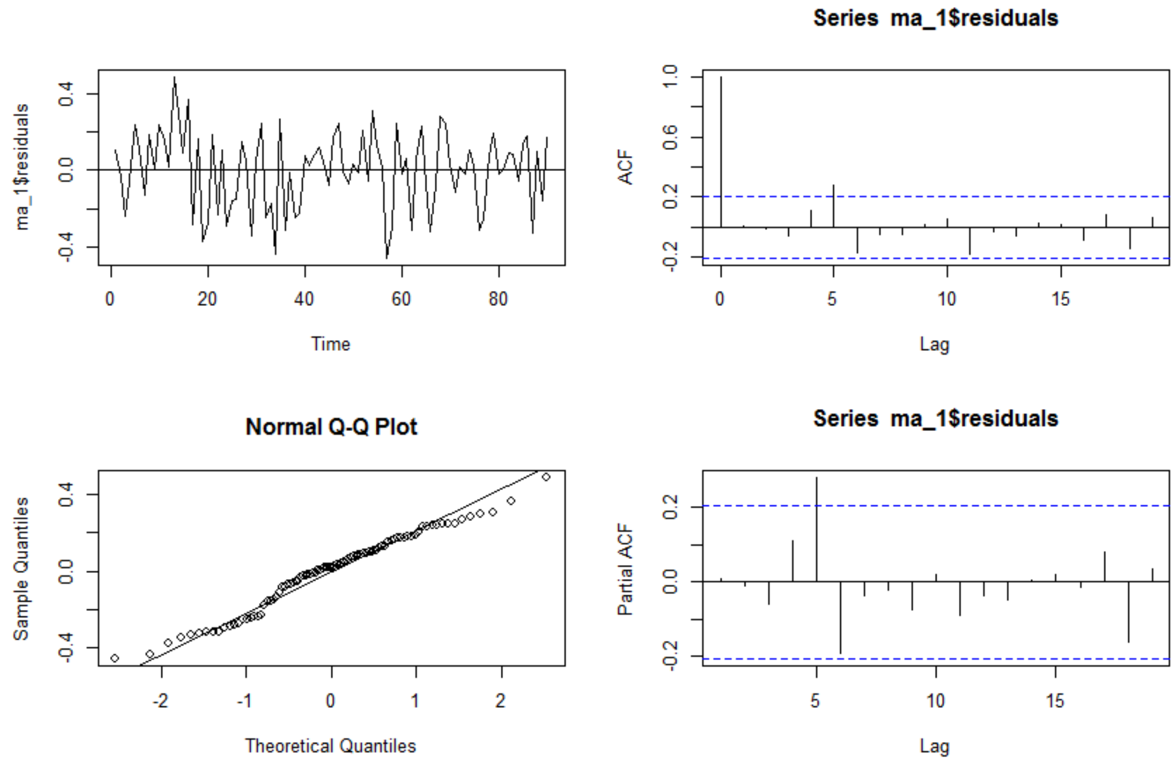
```
Call:
arima(x = knit.res, order = c(1, 0, 1), method = "ML")

Coefficients:
         ar1     ma1  intercept
      0.0335  0.3770     0.0012
s.e.  0.2630  0.2458     0.0299

sigma^2 estimated as 0.03986:  log likelihood = 17.22,  aic = -26.43
```





AR(1) does not pass the Ljung-Box test pass the white noise test. Both MA(1) and ARMA(1,1) pass the Ljung-Box test, and their ACF and PACF graphs are consistent with the white noise assumptions. MA(1) is a better model with smaller AIC. Therefore we choose MA(1) as our prediction model.

```
par(mfcol=c(2,2))
plot(ma_1$residuals)
abline(a=0,b=0)
qqnorm(ma_1$residuals)
qqline(ma_1$residuals)
acf(ma_1$residuals)
pacf(ma_1$residuals)
```



The residuals look like normal and identically distributed and no correlation seems to be present.

- **Prediction**

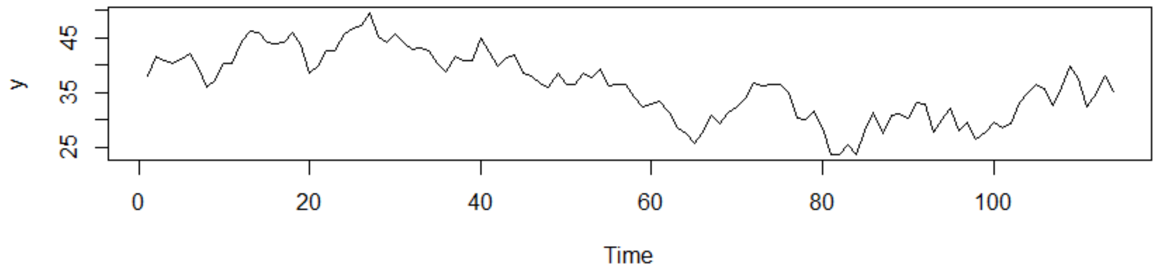
```
SOS_constant=0*c(1:5)
SOS_drift=0*c(1:5)
for (i in 1:5){
  knit.train=head(y.ts,90+i-1)

  ma_1=arima(knit.train,order=c(0,0,1), method="ML")
  p1<-predict(ma_1,1)
  SOS_constant[i]=(p1$pred-knit.future[i])^2

  m=HoltWinters(knit.train, gamma = FALSE, beta = FALSE)
  p2<-predict(m,1,prediction.interval= FALSE)
  SOS_drift[i]=(p2-knit.future[i])^2
}
sum(SOS_constant)
sum(SOS_drift)
```

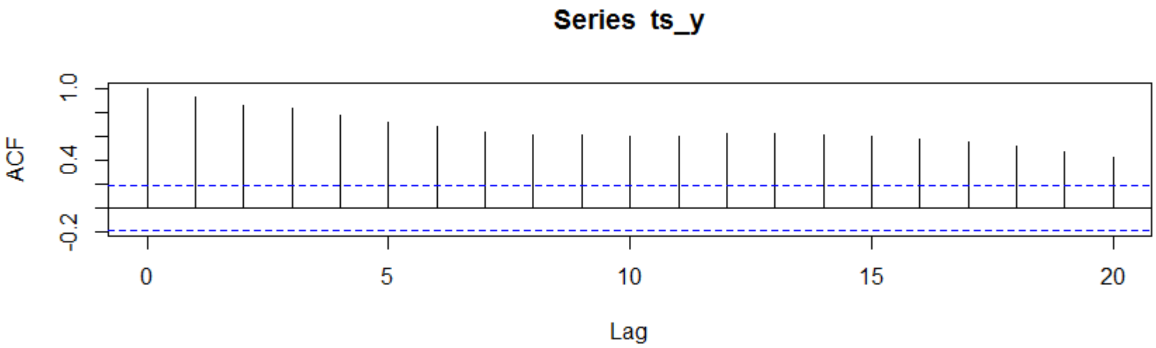
```
> sos_constant=0*c(1:5)
> sos_drift=0*c(1:5)
> for (i in 1:5){
+   knit.train=head(y.ts,90+i-1)
+
+   ma_1=arima(knit.train,order=c(0,0,1), method="ML")
+   p1<-predict(ma_1,1)
+   sos_constant[i]=(p1$pred-knit.future[i])^2
+
+   m=HoltWinters(knit.train, gamma = FALSE, beta = FALSE)
+   p2<-predict(m,1,prediction.interval= FALSE)
+   sos_drift[i]=(p2-knit.future[i])^2
+ }
> sum(sos_constant)
[1] 0.375529
> sum(sos_drift)
[1] 0.5017395
```

For the slowly-drifting mean model, the SOS is 0.5717395. For constant mean model, the SOS is 0.375529. Therefore, the constant mean model is better than slow drift mean model in this case.

Q4:

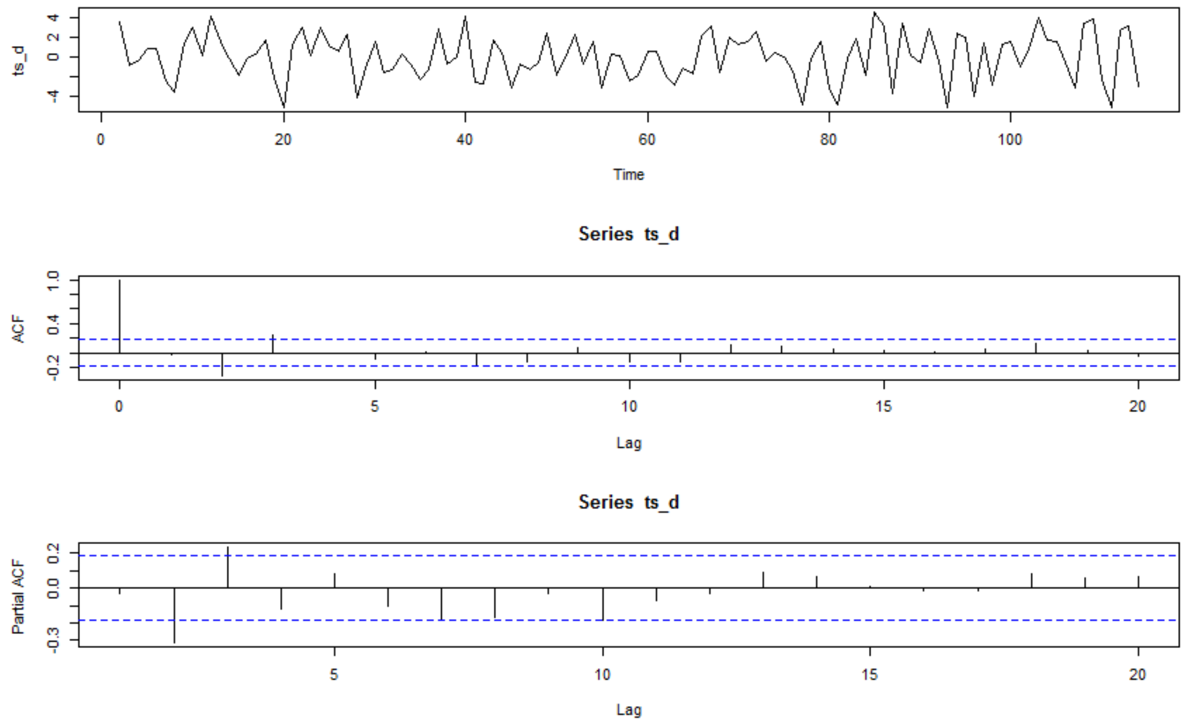
```
y<-scan("C:/Users/Public/sales2.txt")
par(mfrow=c(2,1))
plot(y,type='l',xlab="Time")
ts_y=ts(y)
acf(ts_y)
```





Plot of time series and its auto-correlation function strongly suggest that the series is not stationary. Therefore we create the differenced series and then plot it along with its SACF and SPACF.

```
par(mfrow=c(3,1))
ts_d <- diff(ts_y)
plot(ts_d, type="l")
acf(ts_d)
pacf(ts_d)
```

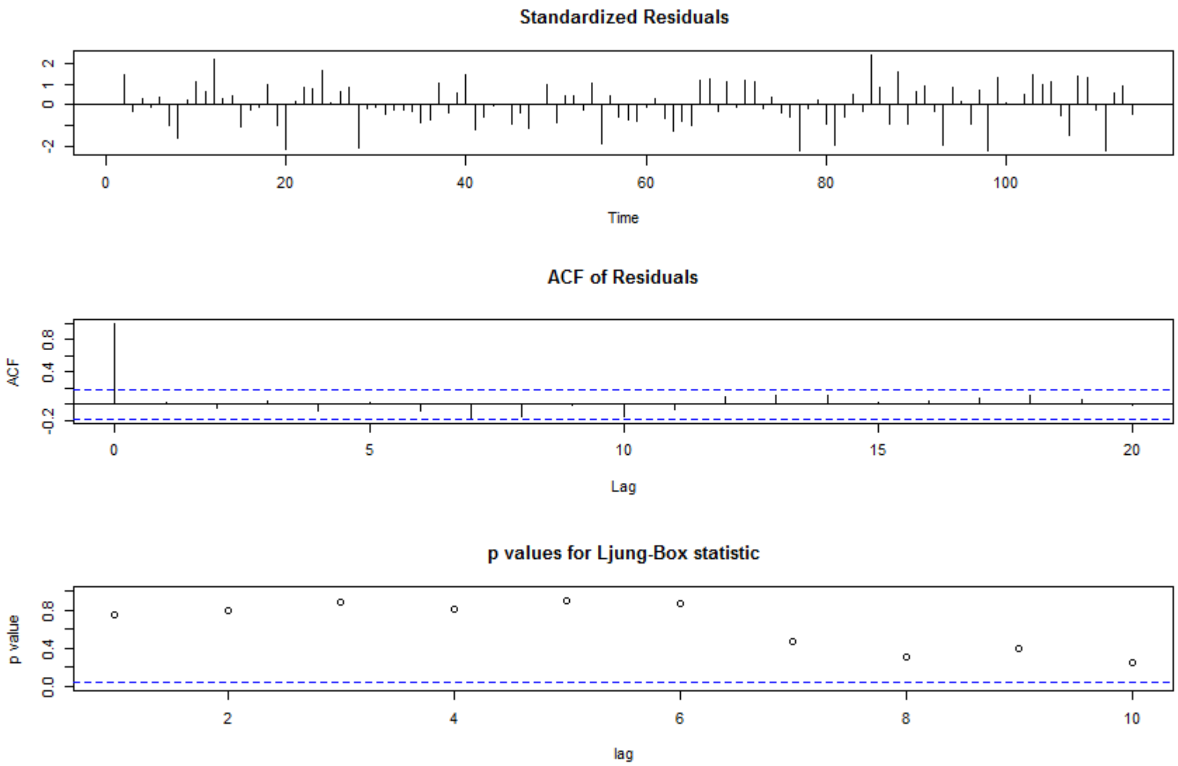


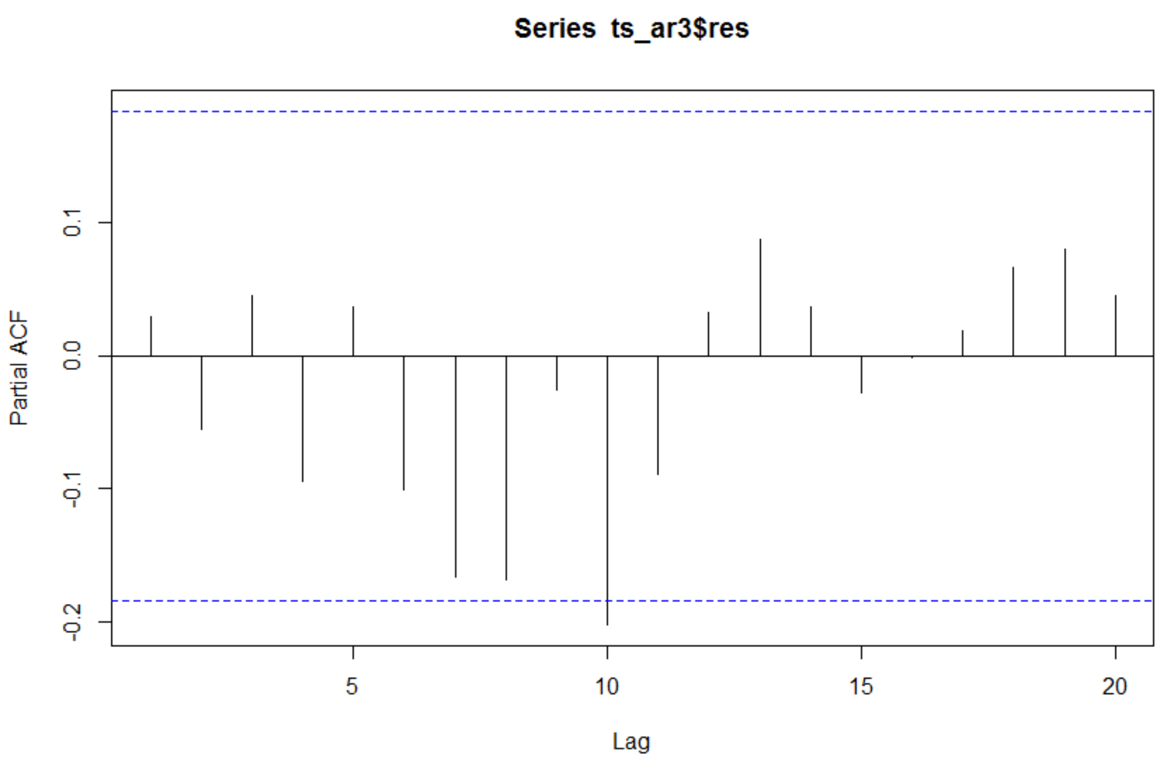
The differenced series appears to be stationary and fairly uncorrelated except at lags 2 and 3. The SPACF has significant spikes at lags 2 and 3. Thus we should try fitting AR(3) and MA(3).

- **ARIMA(3,1,0)**

```
ts_ar3 <- arima(ts_y, order=c(3,1,0),method="ML")
ts_ar3
tsdiag(ts_ar3)
pacf(ts_ar3$res)
```

```
Call:
arima(x = ts_y, order = c(3, 1, 0), method = "ML")

Coefficients:
         ar1      ar2     ar3
      0.0413  -0.3167  0.2500
s.e.  0.0922   0.0875  0.0923

sigma^2 estimated as 4.576:  log likelihood = -246.48,  aic = 500.96
```



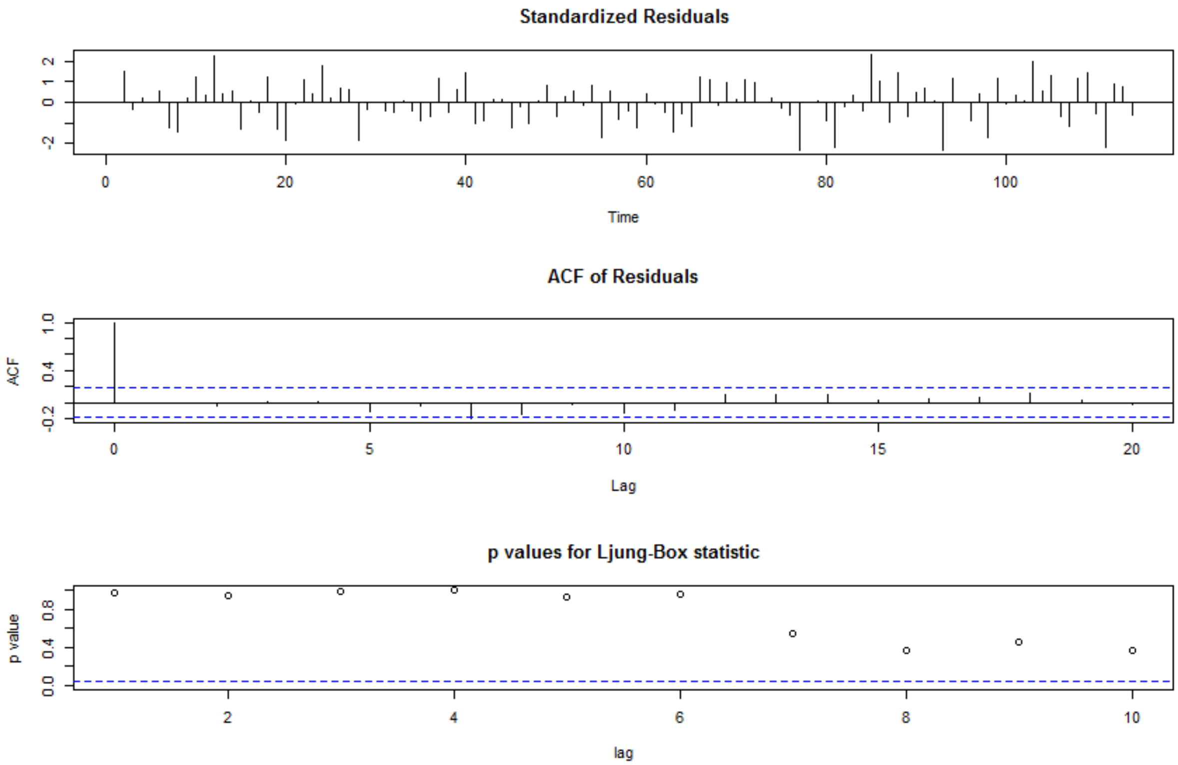


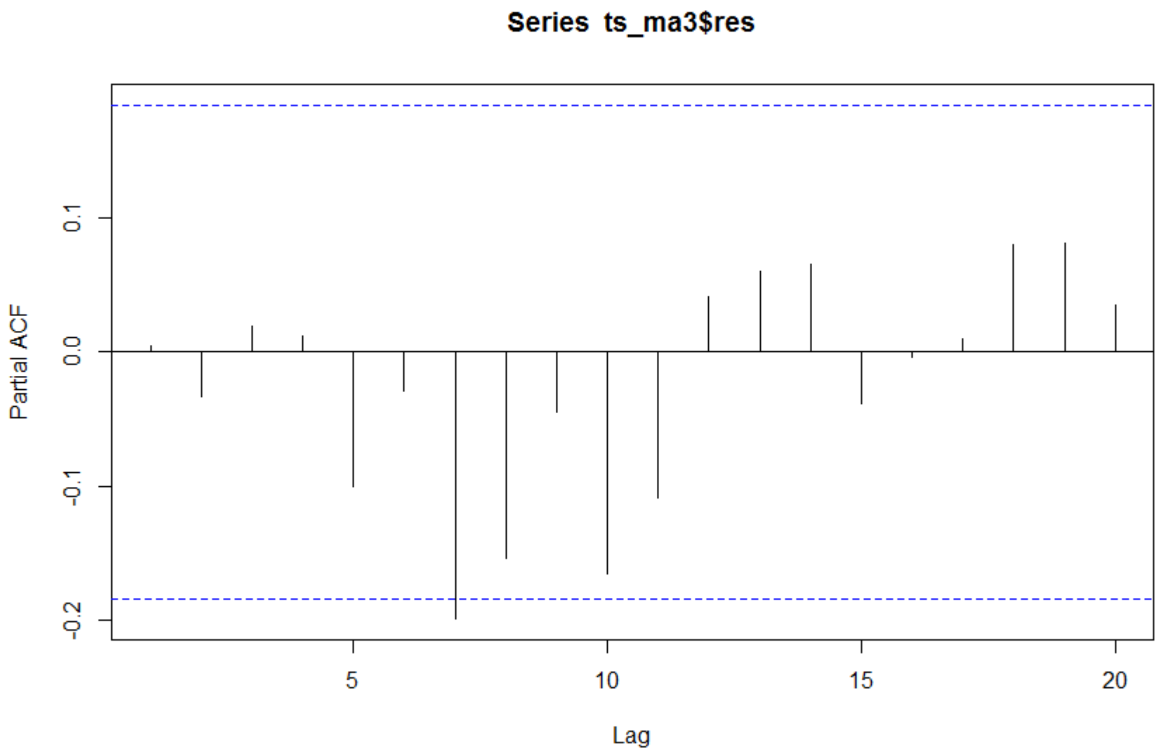
- **ARIMA(0,1,3)**

```
ts_ma3 <- arima(ts_y, order=c(0,1,3),method="ML")
ts_ma3
tsdiag(ts_ma3)
```

```
pacf(ts_ma3$res)
```

```
Call:
arima(x = ts_y, order = c(0, 1, 3), method = "ML")

Coefficients:
         ma1      ma2     ma3
      0.0847  -0.3427  0.2502
s.e.  0.0935   0.0888  0.1113

sigma^2 estimated as 4.468:  log likelihood = -245.22,  aic = 498.44
```

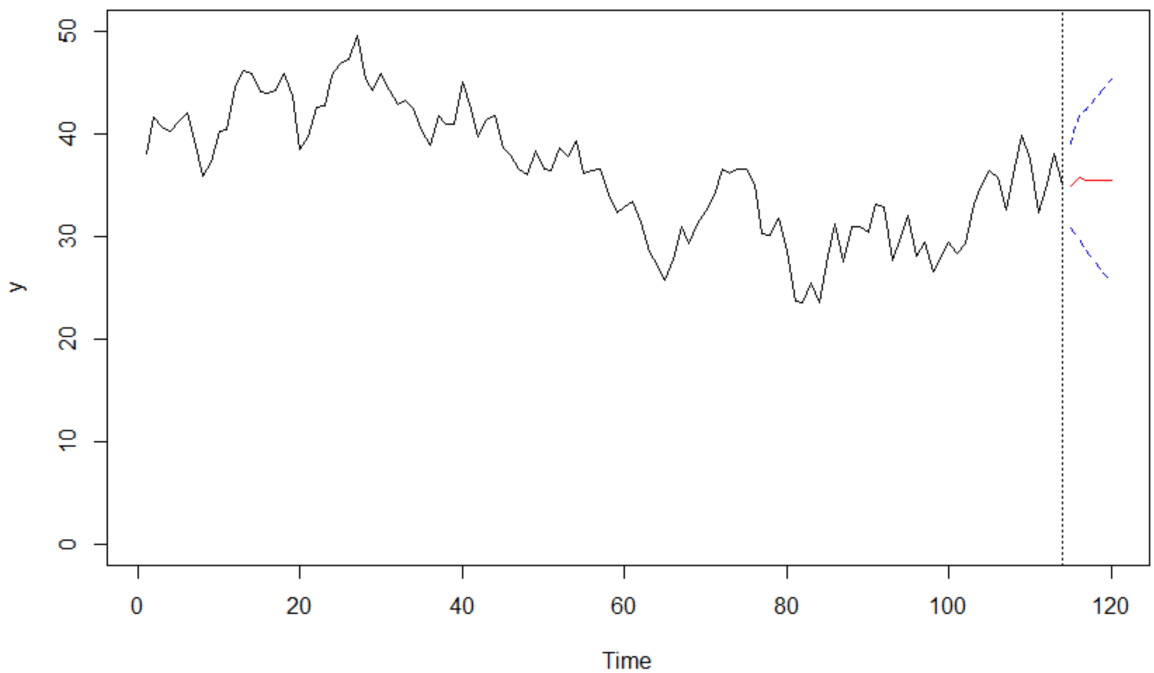




Both models pass the white noise test. However, the AIC of MA(3) is slightly smaller than that of AR(3). Thus we choose MA(3) to make the forecast.
To obtain the forecasts of $Y_{115}, Y_{116}, \ldots, Y_{120}$ using the MA(3) model on the differenced series, we can use:

```
pr<-predict(ts_ma3,n.ahead=6, se.fit = TRUE)
pr
u<-pr$pred+1.96*pr$se
l<-pr$pred-1.96*pr$se
par(mfcol=c(1,1))
plot(y,type='l',xlim=c(1,120),ylim=c(0,50),xlab="Time")
lines(pr$pred,col='red')
lines(u,col='blue',lty='dashed')
lines(l,col='blue',lty='dashed')
abline(v=114,lty='dotted')
```

```
$pred
Time Series:
Start = 115
End = 120
Frequency = 1
[1] 34.91191 35.77092 35.44132 35.44132 35.44132 35.44132

$se
Time Series:
Start = 115
End = 120
Frequency = 1
[1] 2.113694 3.118391 3.490583 4.072125 4.580418 5.037684
```


y
0
10
20
30
40
50
0
20
40
60
80
100
120
Time