FN project

```
df = read.table('FN Project data.csv', header = TRUE, sep = ',')

library(timeSeries)
```

```
## Loading required package: timeDate
```

```
#Create Time Series object for Visualization
dft = ts(df, frequency = 12, start=c(2005, 1))
cht = ts(df$Chicken, frequency = 12, start = c(2005, 1))
corn = ts(df$Corn, frequency = 12, start = c(2005, 1))
```

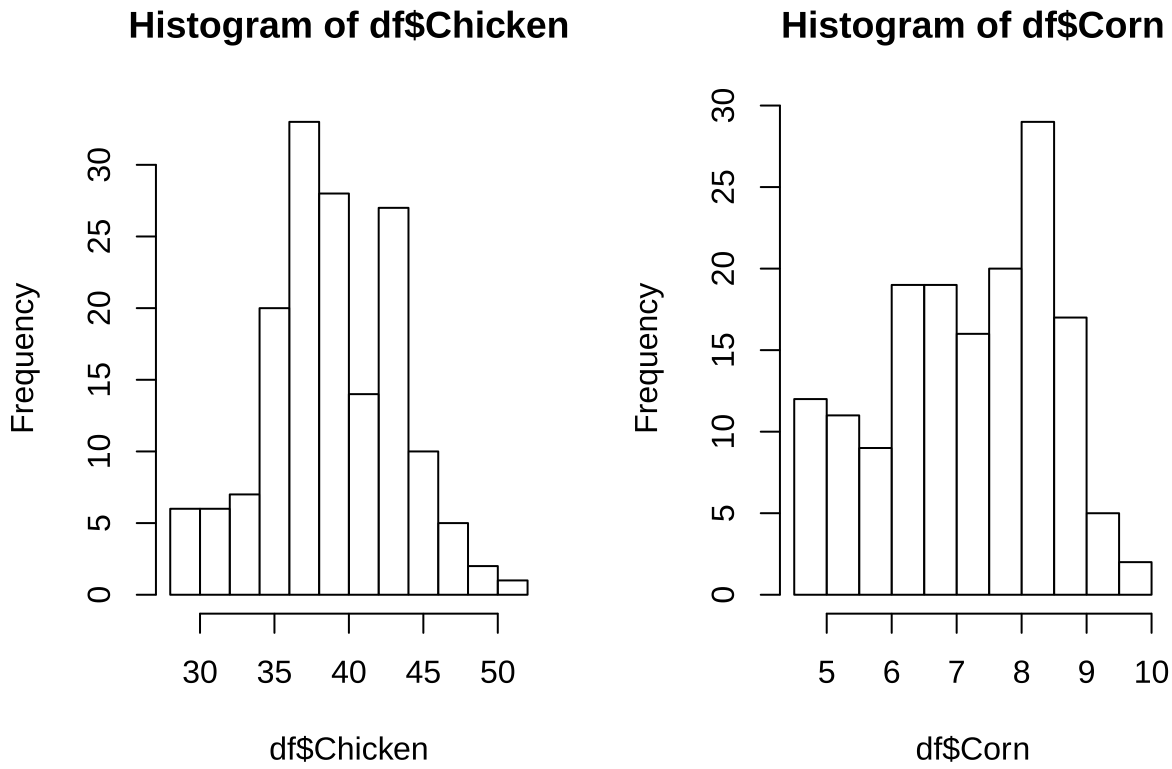
Check missing value

```
sum(is.na(df))
```

```
## [1] 0
```

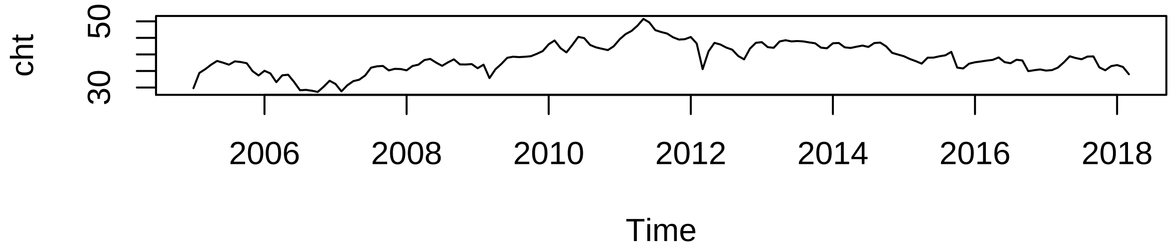
First, let see the distribution of Chicken and Corn

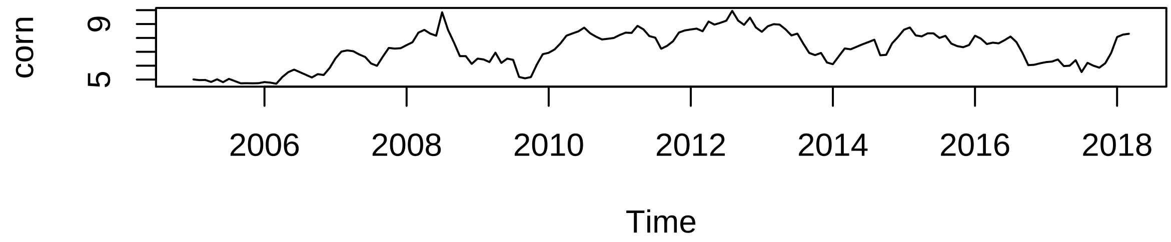
```
par(mfrow = c(1,2))
hist(df$Chicken)
hist(df$Corn)
```



#Time-Series chart of Chicken and Corn

```r
par(mfrow = c(2,1))
plot(cht)
plot(corn)
```





#Time-Series chart of Chicken and Corn

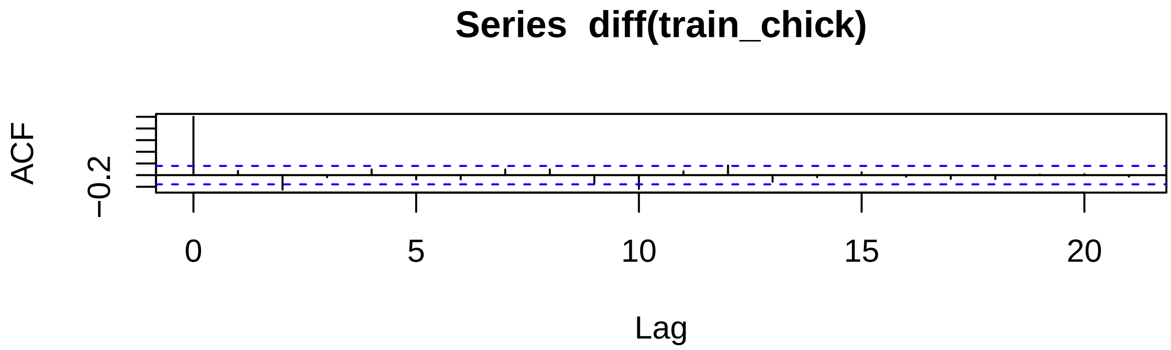
```r
test = lm(Chicken ~ Corn, data = df)
summary(test)
```

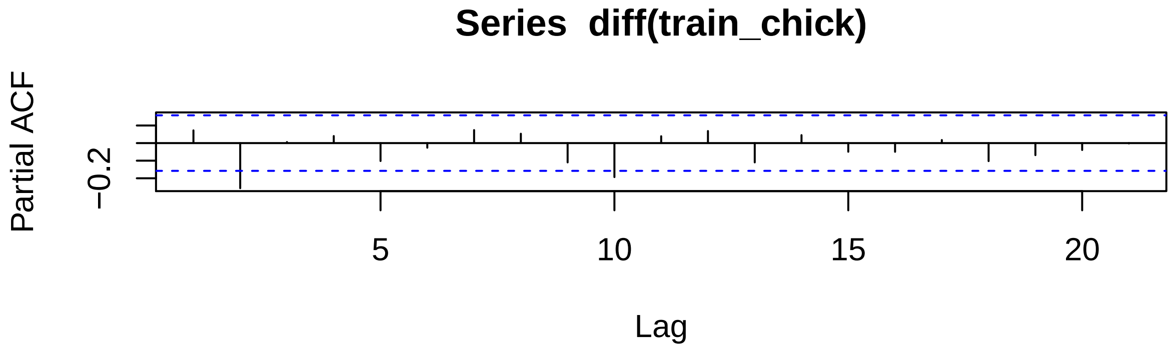
```
##
## Call:
## lm(formula = Chicken ~ Corn, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7656 -2.6522 -0.2751  3.0366  9.1355
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.4480     1.6486   15.44  < 2e-16 ***
## Corn          1.8743     0.2266    8.27 5.35e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.727 on 157 degrees of freedom
## Multiple R-squared:  0.3034, Adjusted R-squared:  0.299
## F-statistic:  68.4 on 1 and 157 DF,  p-value: 5.346e-14
```

Create train-test data for Chicken

Plot ACF/PACF

```
train_chick = df$Chicken[1:155]
test_chick = df$Chicken[156:159]

par(mfrow = c(2, 1))
acf(diff(train_chick))
pacf(diff(train_chick))
```





#Build model

```
model1 = arima(train_chick, order = c(10, 1, 0), fixed = c(0, NA, 0, 0, 0, 0, 0, 0, 0, NA))
```

```
## Warning in arima(train_chick, order = c(10, 1, 0), fixed = c(0, NA, 0, 0, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
model1
```

```
##
## Call:
## arima(x = train_chick, order = c(10, 1, 0), fixed = c(0, NA, 0, 0, 0, 0, 0,
##     0, 0, NA))
##
## Coefficients:
##       ar1      ar2  ar3  ar4  ar5  ar6  ar7  ar8  ar9     ar10
##         0  -0.2413    0    0    0    0    0    0    0  -0.2249
## s.e.    0   0.0791    0    0    0    0    0    0    0   0.0788
##
## sigma^2 estimated as 2.235:  log likelihood = -280.79,  aic = 567.58
```

```r
model2 = arima(train_chick, order = c(2, 1, 0), fixed = c(0, NA))
```

```
## Warning in arima(train_chick, order = c(2, 1, 0), fixed = c(0, NA)): some
## AR parameters were fixed: setting transform.pars = FALSE
```

```r
model2
```

```
##
## Call:
## arima(x = train_chick, order = c(2, 1, 0), fixed = c(0, NA))
##
## Coefficients:
##       ar1      ar2
##         0  -0.2698
## s.e.    0   0.0808
##
## sigma^2 estimated as 2.36:  log likelihood = -284.73,  aic = 573.45
```

```r
model3 = arima(train_chick, order = c(0, 2, 2))
model3
```

```
##
## Call:
## arima(x = train_chick, order = c(0, 2, 2))
##
## Coefficients:
##           ma1      ma2
##       -0.8363  -0.1637
## s.e.   0.1132   0.1091
##
## sigma^2 estimated as 2.519:  log likelihood = -290.17,  aic = 586.33
```

```r
model4 = arima(train_chick, order = c(2, 2, 2))
model4
```

```
##
## Call:
## arima(x = train_chick, order = c(2, 2, 2))
##
## Coefficients:
##          ar1      ar2      ma1      ma2
##       0.0725  -0.2685  -0.9689  -0.0254
## s.e.  0.2585   0.0844   0.2682   0.2659
##
## sigma^2 estimated as 2.365:  log likelihood = -285.34,  aic = 580.68
```

Models summary

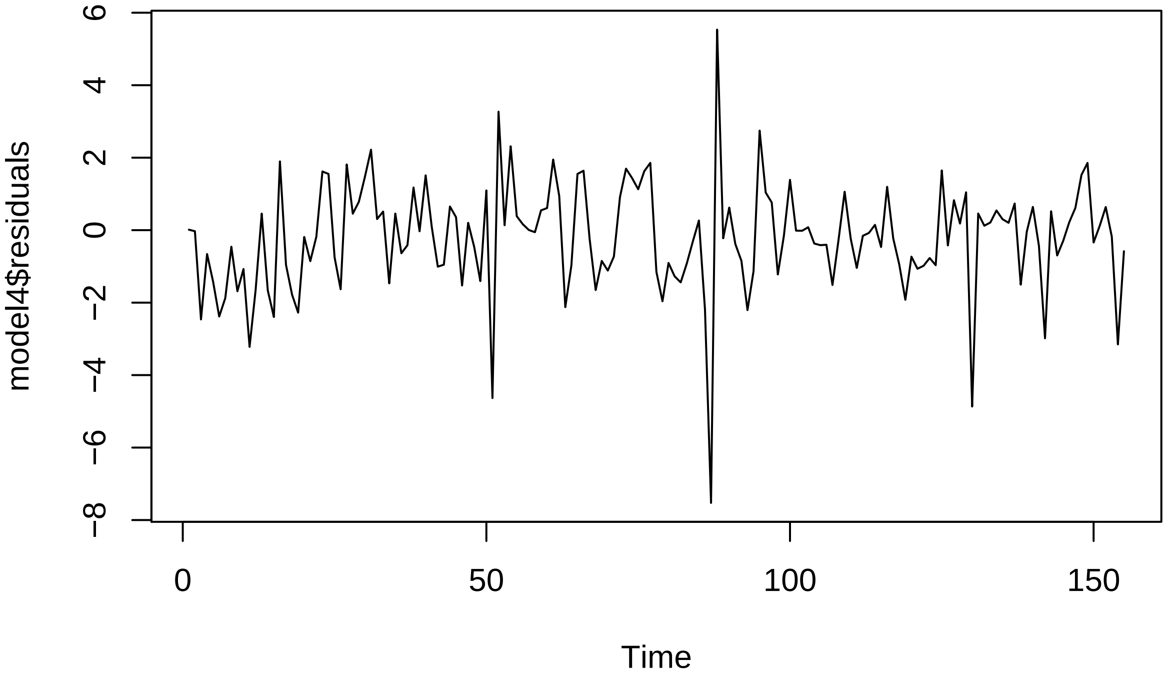
```r
Models = c('Model1','Model2','Model3','Model4')
AIC = c(model1$aic,model2$aic,model3$aic,model4$aic)
```

```
data.frame(Models, AIC)
```

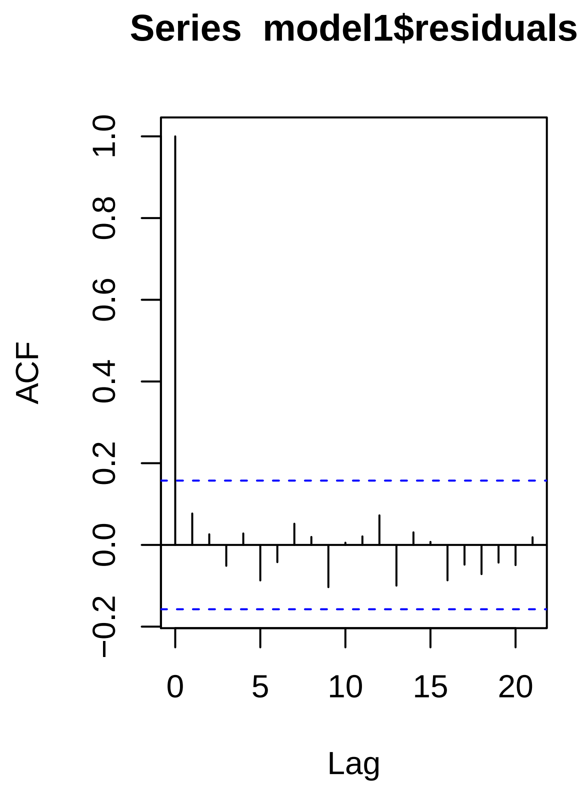
```
##   Models      AIC
## 1 Model1 567.5827
## 2 Model2 573.4506
## 3 Model3 586.3325
## 4 Model4 580.6810
```

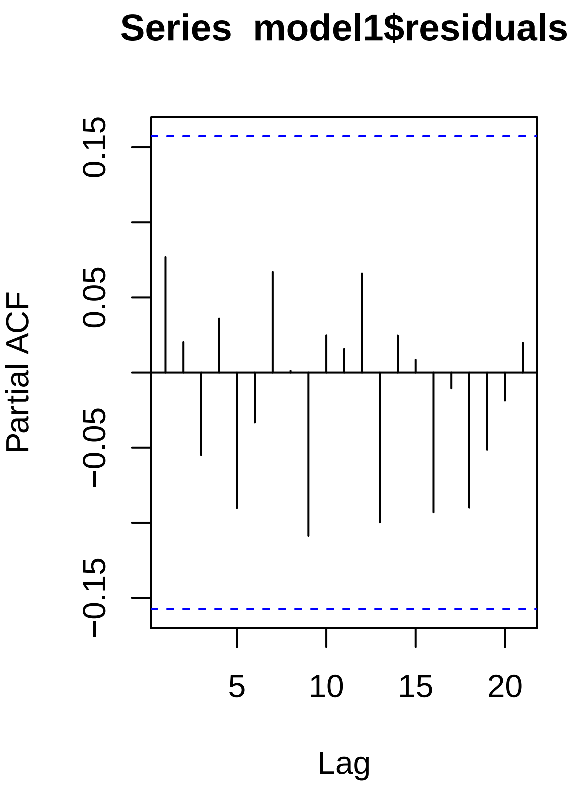
Check adequecy of model

```
plot(model4$residuals)
```

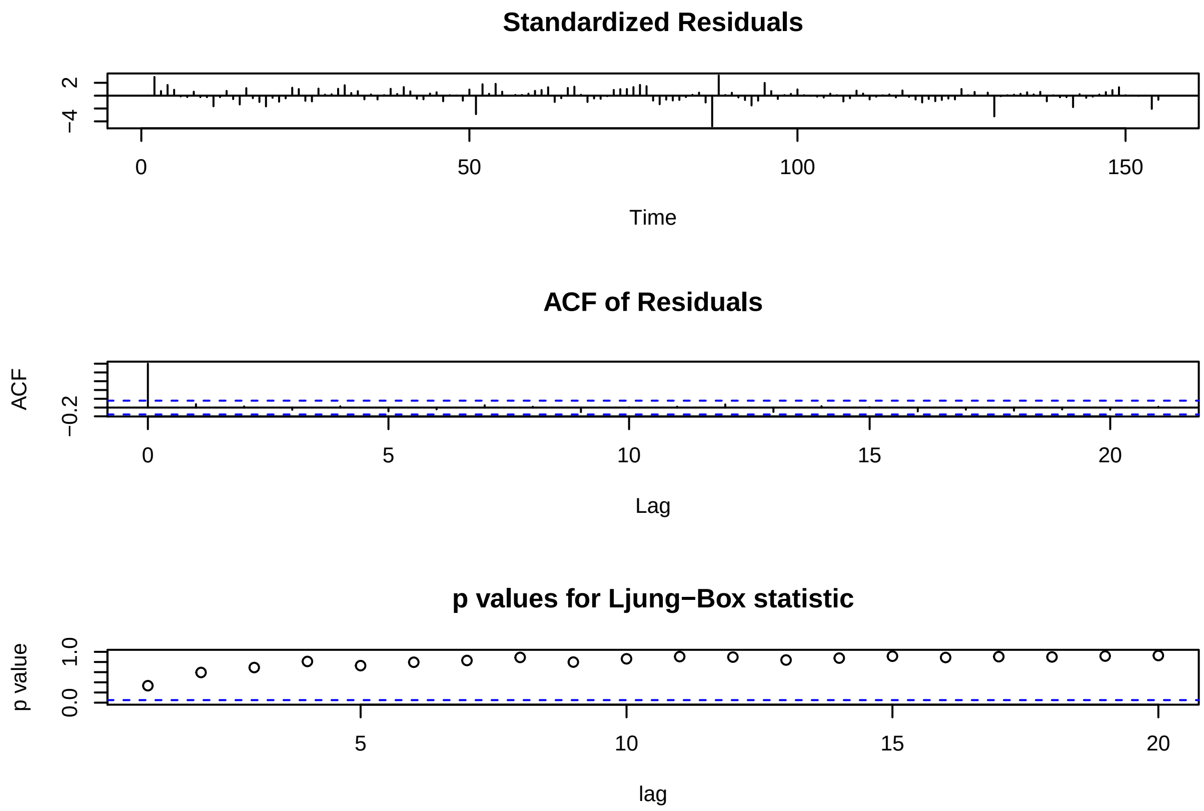


```
par(mfrow = c(1, 2))
acf(model1$residuals)
pacf(model1$residuals)
```


Series model1$residuals


Series model1$residuals

```r
tsdiag(model1,gof=20) #Visualize residuals of model
```

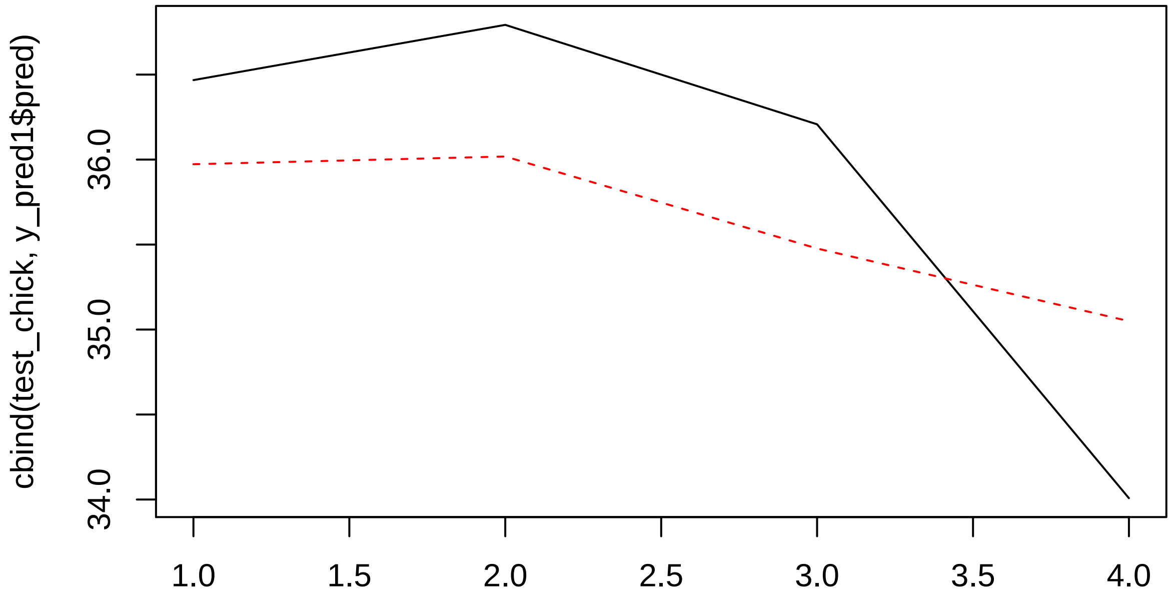


#Forcast

```r
library(forecast)
y_pred1 = predict(model1, 4)
accuracy(y_pred1$pred, test_chick)
```

```
##                 ME      RMSE       MAE      MPE     MAPE
## Test set 0.2396157 0.7847509 0.7603827 0.604335 2.135643
```

```r
matplot(cbind(test_chick, y_pred1$pred), type = 'l')
```



#Add Corn to model

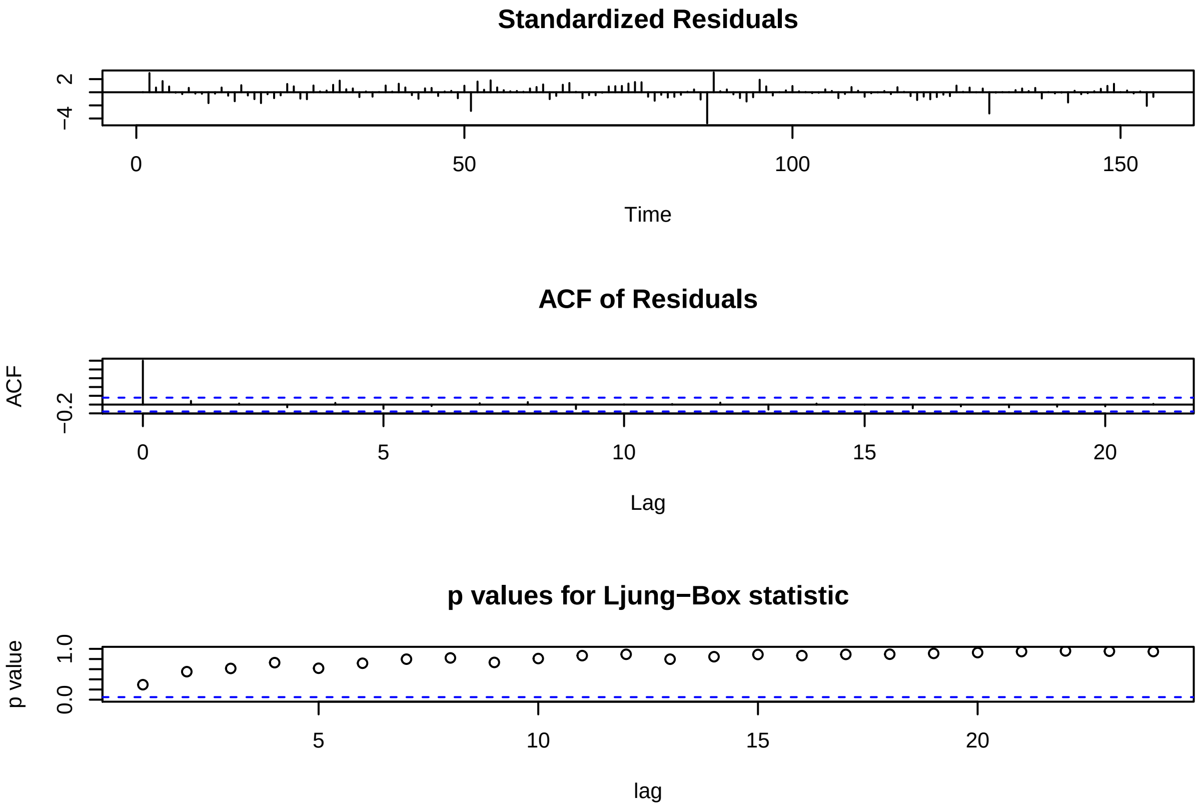
```
#Create Corn train/test
train_corn = df$Corn[1:155]
test_corn = df$Corn[156:159]

#Add Corn to previous best result model
model5 = arima(train_chick, order = c(10, 1, 0), fixed = c(0, NA, 0, 0, 0, 0, 0, 0, 0, NA, NA), xreg =
```

```
## Warning in arima(train_chick, order = c(10, 1, 0), fixed = c(0, NA, 0, 0, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
model5
```

```
##
## Call:
## arima(x = train_chick, order = c(10, 1, 0), xreg = train_corn, fixed = c(0,
##     NA, 0, 0, 0, 0, 0, 0, 0, NA, NA))
##
## Coefficients:
##       ar1      ar2  ar3  ar4  ar5  ar6  ar7  ar8  ar9     ar10  train_corn
##         0  -0.2412    0    0    0    0    0    0    0  -0.2310      0.3556
## s.e.    0   0.0793    0    0    0    0    0    0    0   0.0791      0.2637
##
## sigma^2 estimated as 2.209:  log likelihood = -279.89,  aic = 567.78
```

```r
tsdiag(model5,gof=24)
```


Standardized Residuals

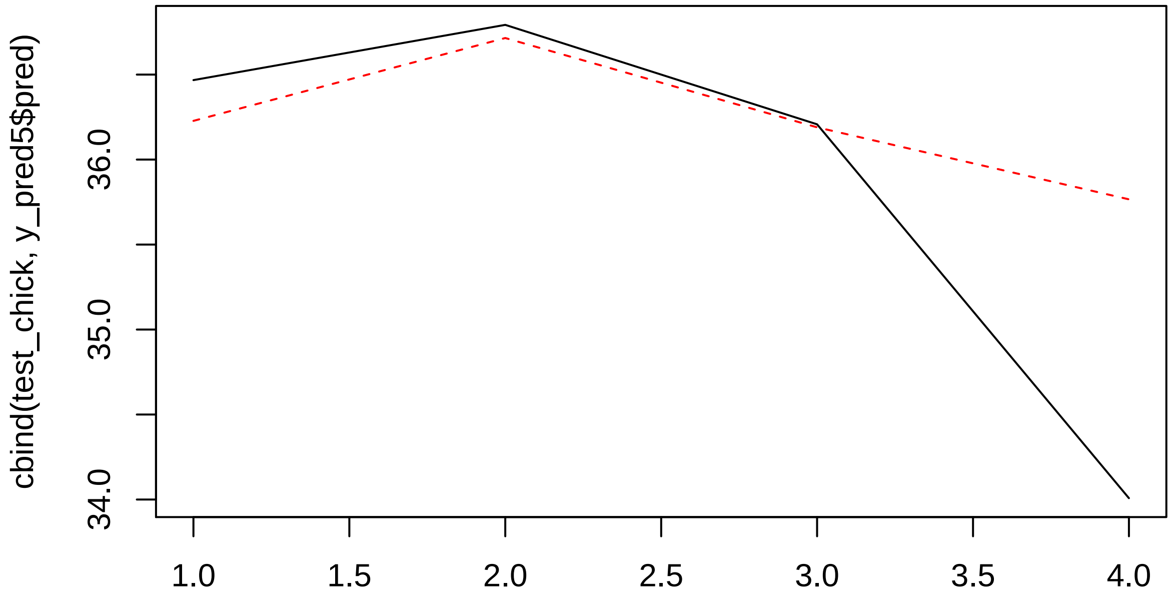
ACF of Residuals

p values for Ljung−Box statistic

```r
y_pred5 = predict(model5,4, newxreg = test_corn )
accuracy(y_pred5$pred, test_chick)
```

```
##                   ME      RMSE       MAE       MPE     MAPE
## Test set -0.3558275 0.8880972 0.5232461 -1.063288 1.521615
```

```r
matplot(cbind(test_chick, y_pred5$pred),type= 'l')
```

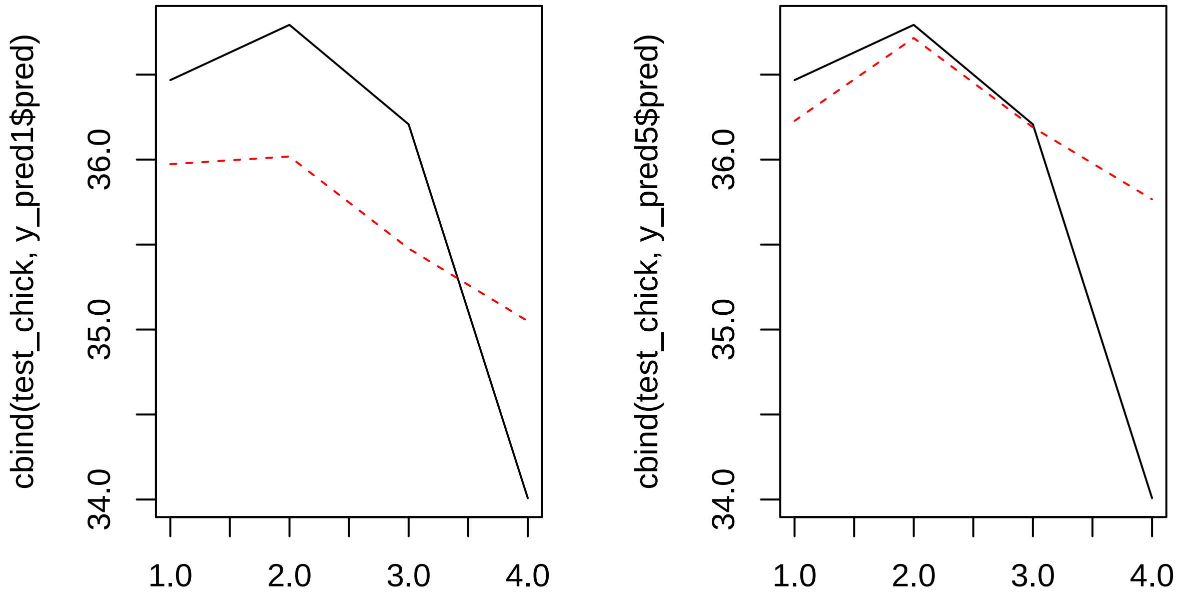


#Compare result with previous model

```
Models2 = c('Arima(10,1,0) Fixed','Model1 with Corn')
AIC2 = c(model1$aic,model5$aic)
MAE = c(accuracy(y_pred1$pred, test_chick)[3], accuracy(y_pred5$pred, test_chick)[3])
data.frame(Models2, AIC2, MAE)
```

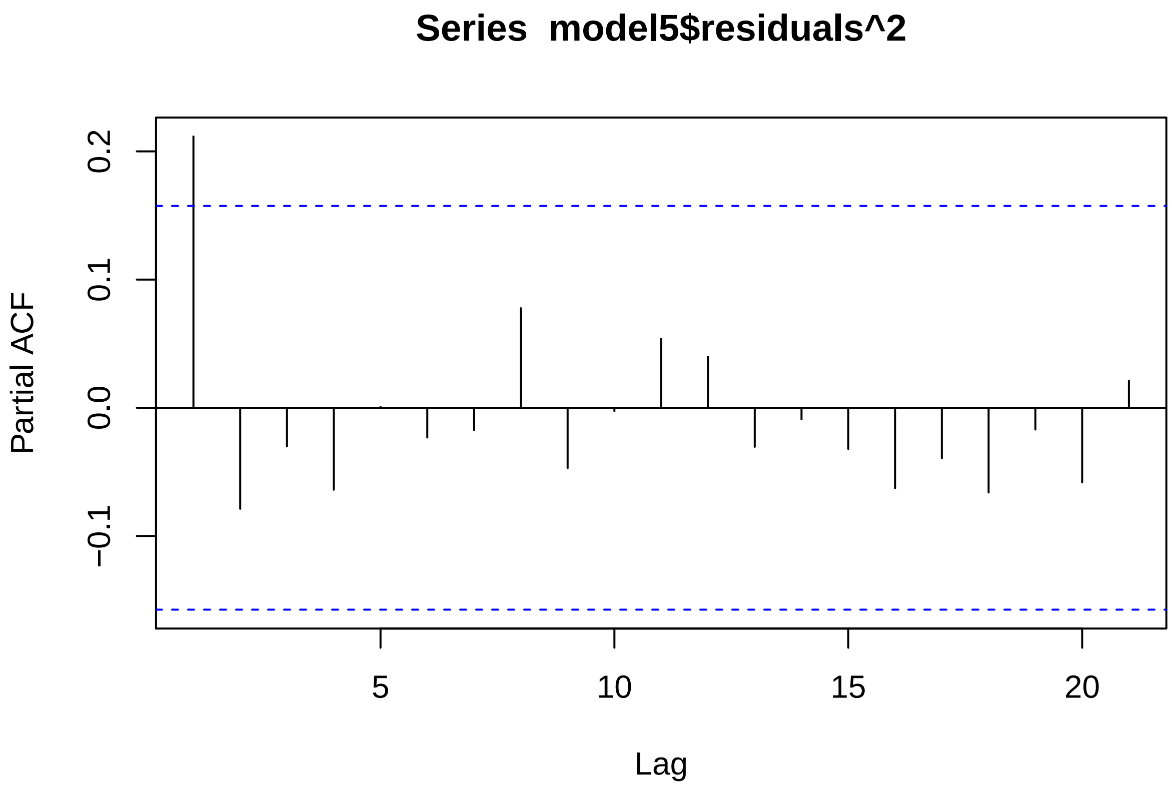
```
##               Models2     AIC2       MAE
## 1 Arima(10,1,0) Fixed 567.5827 0.7603827
## 2    Model1 with Corn 567.7774 0.5232461
```

```
par(mfrow = c(1, 2))
matplot(cbind(test_chick, y_pred1$pred),type= 'l')
matplot(cbind(test_chick, y_pred5$pred),type= 'l')
```



Test Arch effect

```
pacf(model5$residuals^2)
```


Series model5$residuals^2
Partial ACF
0.2
0.1
0.0
−0.1
5
10
15
20
Lag