Statistical Methods for Discrete Response, Time Series, and Panel Data: Live Session 8

Devesh Tiwari

7/5/2017

Main Topics Covered in Lecture 8:

```
- Autoregressive (AR) models
    - Lag (or backshift) operators
    - Properties of the general AR(p) model
    - Simulation of AR Models
    - Estimation, model diagnostics, model identification, model selection, assumption testing, and sta

- Moving Average (MA) Models
    - Lag (or backshift) operators
    - Mathematical formulation and derivation of key properties
    - Simulation of MA(q) models
    - Estimation, model diagnostics, model identification, model selection, assumption testing, and sta
```

Readings:

CM2009: Paul S.P. Cowpertwait and Andrew V. Metcalfe. *Introductory Time Series with R.* Springer. 2009.

```
- Ch. 3.1, 3.2, 4.5, 6.1 - 6.4
```

SS2016: Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Applications.* EZ Edition with R Examples

```
- Ch. 3.1 - 3.6
```

HA: Rob J Hyndman and George Athanasopoulos. Forecasting: Principles and Practice.

```
- 8.2, 8.3, 8.4
```

Agenda for the Live Session

1. Recap (5 mins)
2. Breakout 1: Review questions (15 mins in group, 5 mins discussion)
3. Breakout 2: EDA, AR, and MA models (15 mins in group; 15 mins discussion)
4. Breakout 3: Comparing models (15 mins in group; 15 mins discussion)

ARIMA Modeling Procedure Recap

Last week, we talked about linear time-regression. This week, we begin univariate time-series analysis, a modelling process in which the current observation is a function of prior observations. Time-series data that are stationary in the mean can be generated by an autoregressive model or a moving avaerage model. ARMA models have both components.

When fitting the class of ARIMA model to a set of time series data, the following procedure provides a useful general approach.

1. Examine the data structure.
2. Plot the data. EDA (in general). Identify any unusual observations.

(SKIP IN THIS LECTURE) 3. If necessary, transform the data (using a Box-Cox transformation) to stabilize the variance.

(SKIP IN THIS LECTURE) 4. If the data are non-stationary: take first differences of the data until the data are stationary.

5. Examine the ACF/PACF: Is an AR(p) or MA(q) model appropriate?
6. Try your chosen model(s), and use appropriate metrics to choose a model.
7. Model evaluation Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
8. Once the residuals look like white noise, calculate forecasts.

Review Questions

Answer the questions below. I will post the answers after the breakout session, if you do not understand how to get the correct answer, email me and schedule an office hours appointment.

1. Consider the pure Moving Average process below, where ω_t is a white noise process:

$$x_t = \omega_t + \beta_1 \omega_{t-1}$$

Under what circumstances is this process stationary in the mean?

Answer: Always.

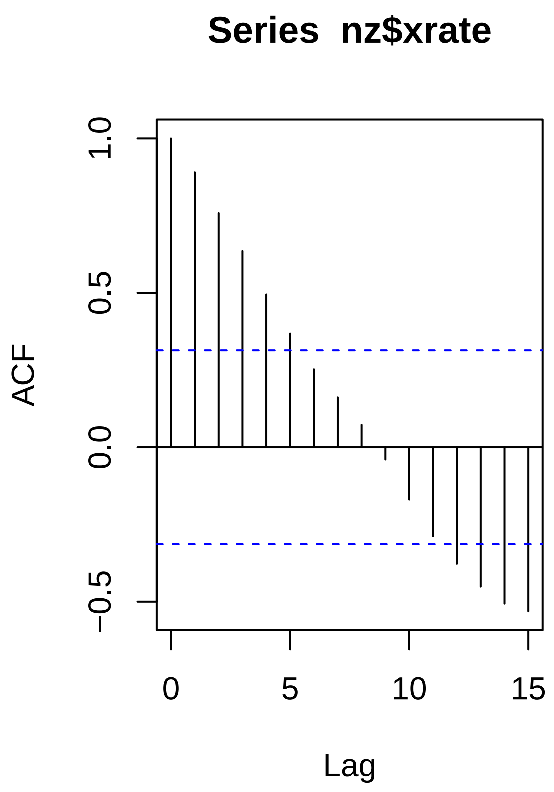
(a) Always
(b) Never, this is not an AR(p) process.
(c) When the absolute value of β_1 is less than 1.
(d) When the absolute value of the roots of the characteristic polynomial is greater thane 1.

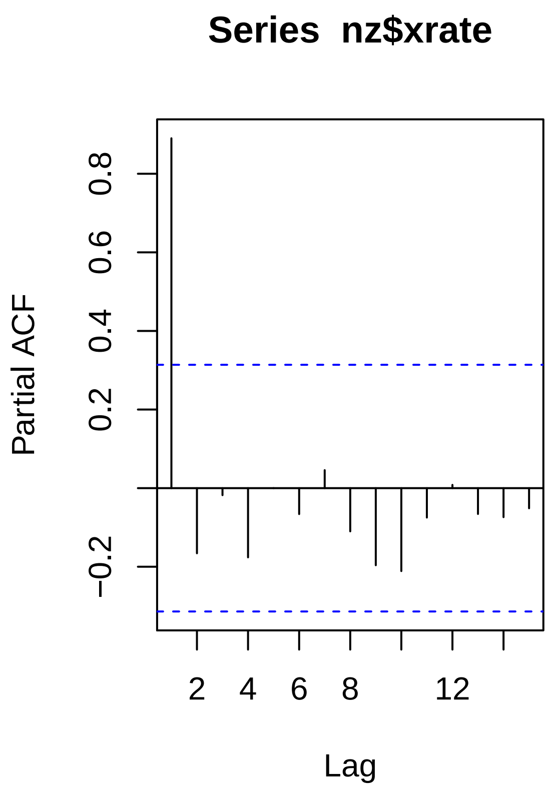
2. Consider a time series dataset on which you fit an MA(q') model and an AR(p') model. You are about to fit an AMRA(p,q) model. As a rule of thumb, how many parameters (p+q) will be present in the optimal model?

Answer: D

(a) Impossible to tell. Adding more terms always increases explanatory power.
(b) Less than or equal to p' + q'
(c) Less than the max of p' or q'
(d) Less than the min of p' or q'

3. Consider the ACF and PACF charts of a time series below.





An analyst says that fitting an MA model might be inappropriate here. Do you agree? Why or why not?

The ACF/PACF charts have the clear signature of an AR(1) model.

4. Consider the following output from the Arima function. Rewrite this equation with and without the backshift operator. Do you think that this model is stationary in the mean? Why or why not? Hint: The intecept is not the intercept in the output.

According to the documentation, the "intercept" from the Arima output is really the mean of the series. Recall that:

$$\mu = \alpha_0/(1-\alpha_1)$$

$$x_t = -0.163 + 0.056 * \alpha_1 + \omega_t$$

$$(1-0.056\mathbf{B}) = -0.163 + \omega_t$$

```
## Series: sim.ar1
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1  intercept
##       0.2187    -0.0426
## s.e.  0.1018     0.1502
##
## sigma^2 estimated as 1.413:  log likelihood=-158.19
## AIC=322.39   AICc=322.64   BIC=330.2
##
## Training set error measures:
```

```
##                        ME    RMSE       MAE      MPE    MAPE      MASE
## Training set 0.008200745 1.17674 0.9367887 87.08405 110.285 0.8177288
##                      ACF1
## Training set -0.009118537
```

Breakout Session 2: Fitting AR and MA models to a time-series

In the EDA stage, answer the following questions.

(1) How would you describe the time-series in words? Try comparing this particular time-series with a white noise series. What do you see?

(2) Is this time-series stationary in the mean?

(3) Is this time-series stationary in the variance?

(4) Is this time-series stationary in the covariance?

After the EDA stage, we are going to model this time-series as an AR(p) and an MA(q) process. PLEASE DO NOT SKIP THIS PART BY MODELLING THE DATA AS AN ARMA MODEL OR BY USING THE AUTO.ARIMA FUNCTION!!

(5) Note that the AIC is minimized at AR(9) but it declines pretty slowly. Examine the residuals of AR(9) models and simpler models. Based on your residuals analysis, how many lags should we include if we were modelling this only as an AR model?

(6) Repeat step 5, but for an MA model. How many lags should we include if we were modelling this as an MA model?

The AIC is declining slowly and it will likely take more than 15 lags to find the right model. Given that we cuold find a min. for a pure AR model, it seems the case that we cannot fit this a pure MA model on this time-series.

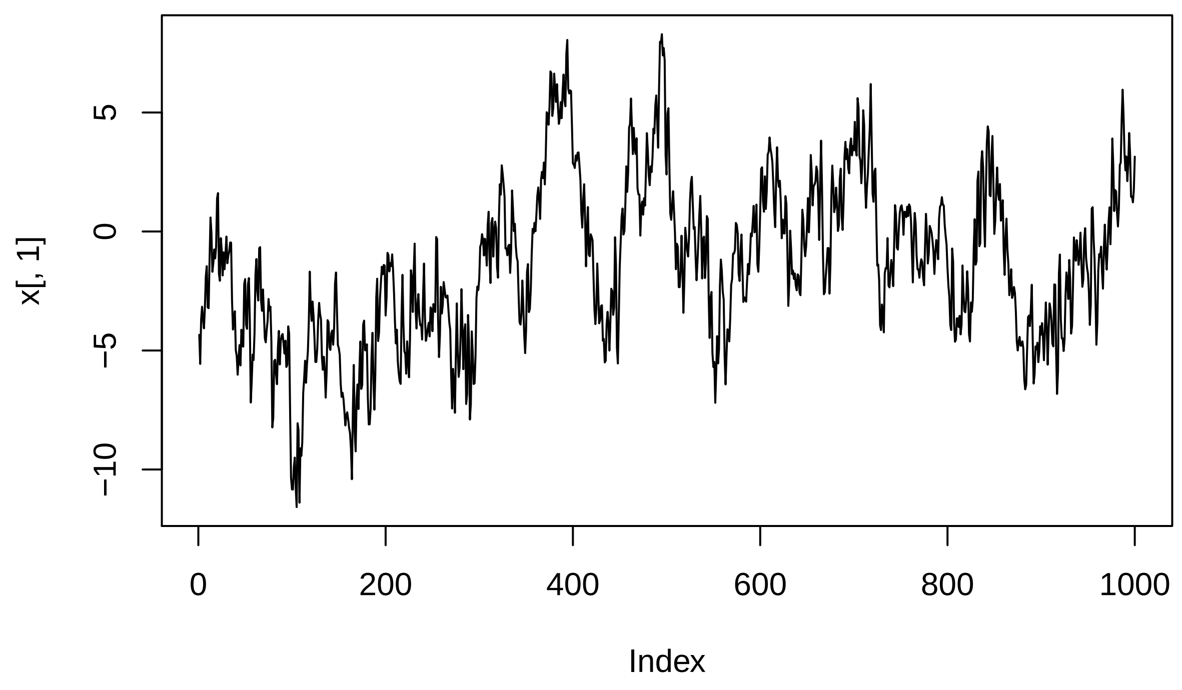
```r
x <- read.csv("week8SeriesTiwari.csv")
cbind(head(x), tail(x))
```

```
##           x        x
## 1 -4.341226 3.234460
## 2 -5.563720 1.455413
## 3 -3.694864 1.472621
## 4 -3.160604 1.225259
## 5 -3.692553 1.757768
## 6 -4.069171 3.144503
```

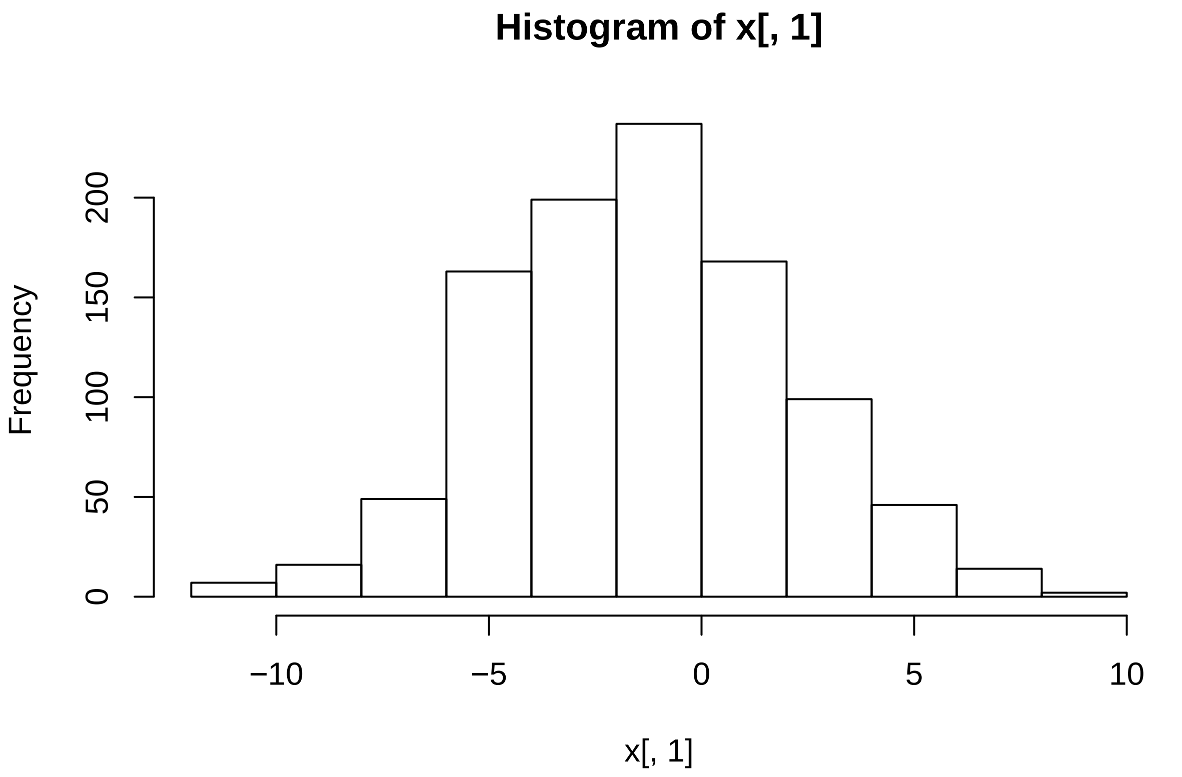
```r
dim(x)
```

```
## [1] 1000    1
```

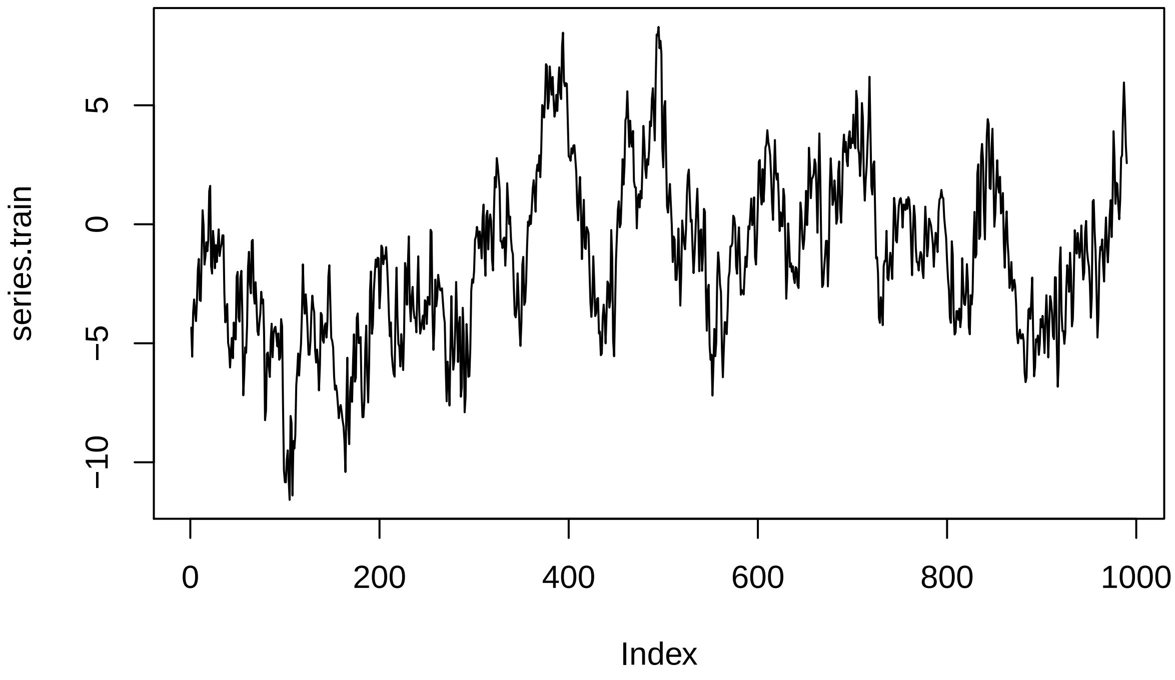
```r
plot(x[, 1], type = "l")
```



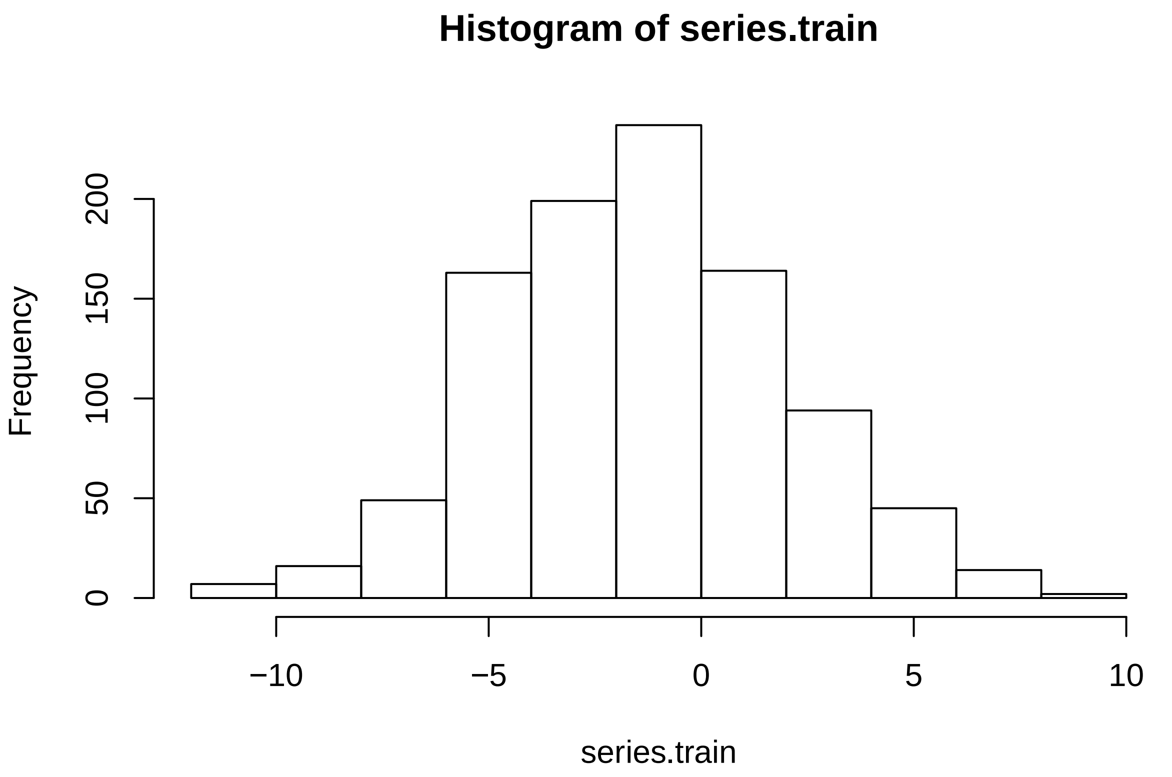
```r
hist(x[, 1])
```



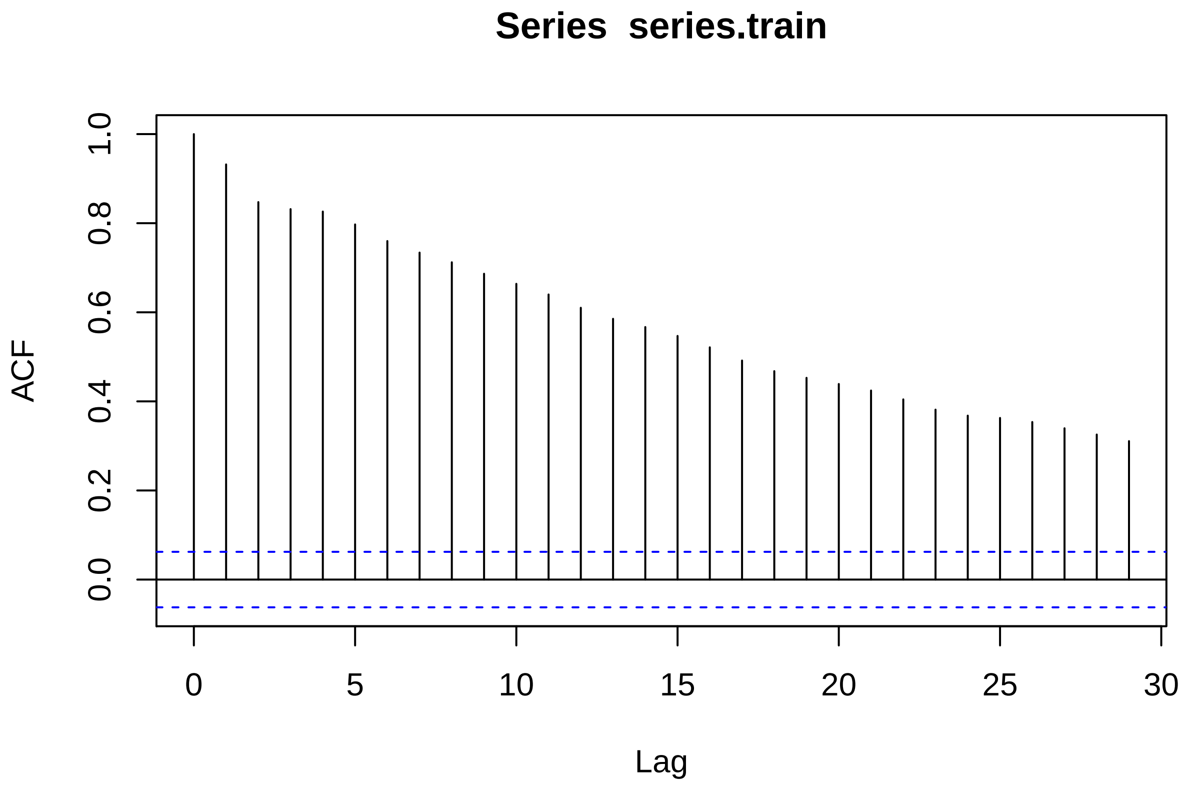
```r
# Be sure to check for any missing values or unusual
# observations!

### Split dataset into training and test sets

series.train <- x[1:990, 1]
series.test <- x[991:1000, 1]

plot(series.train, type = "l")
```



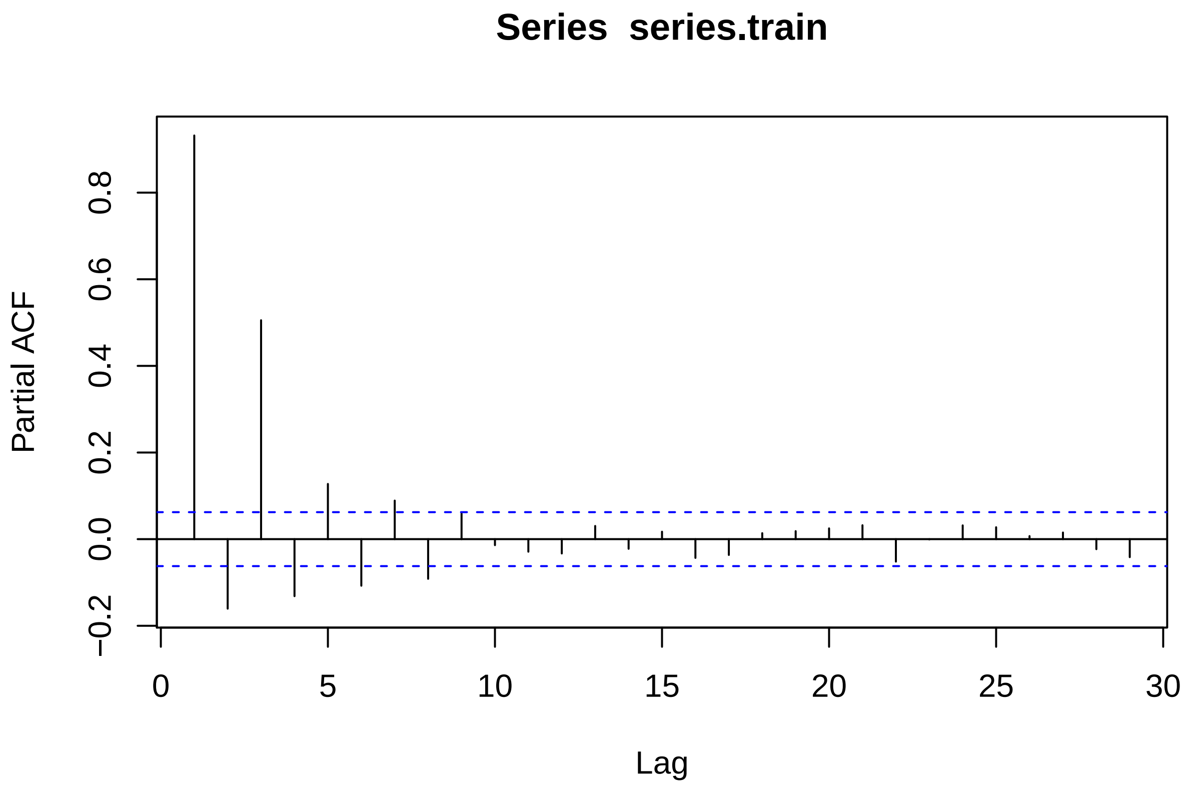
```r
hist(series.train)
```



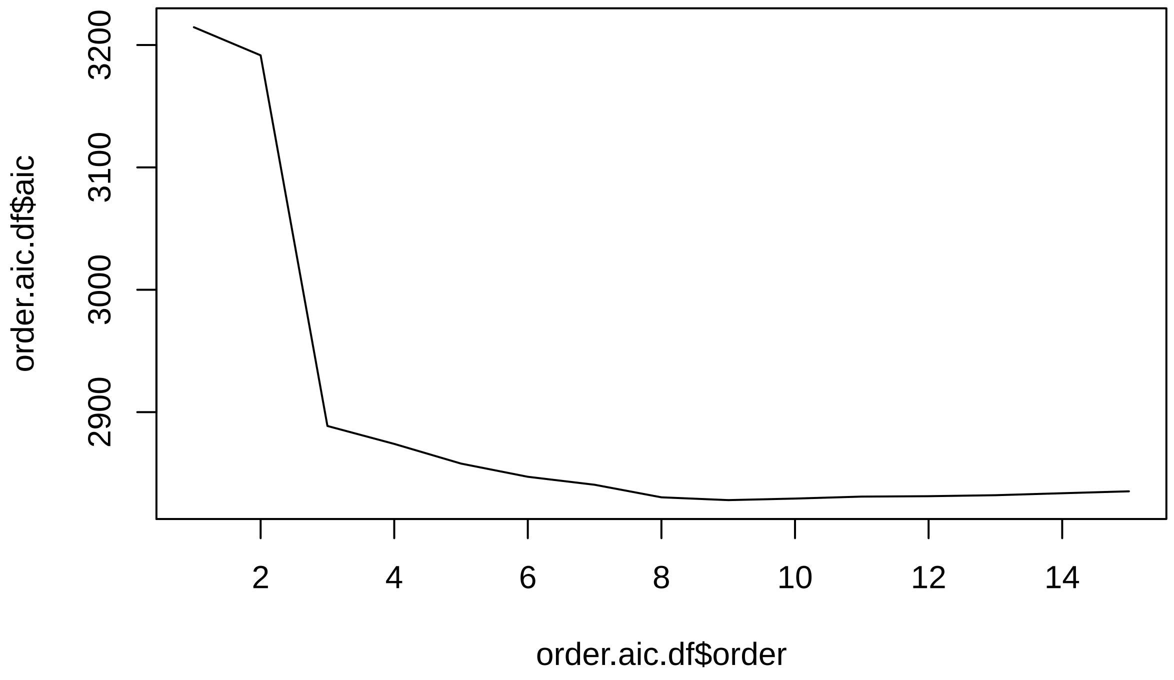
```r
acf(series.train)
```



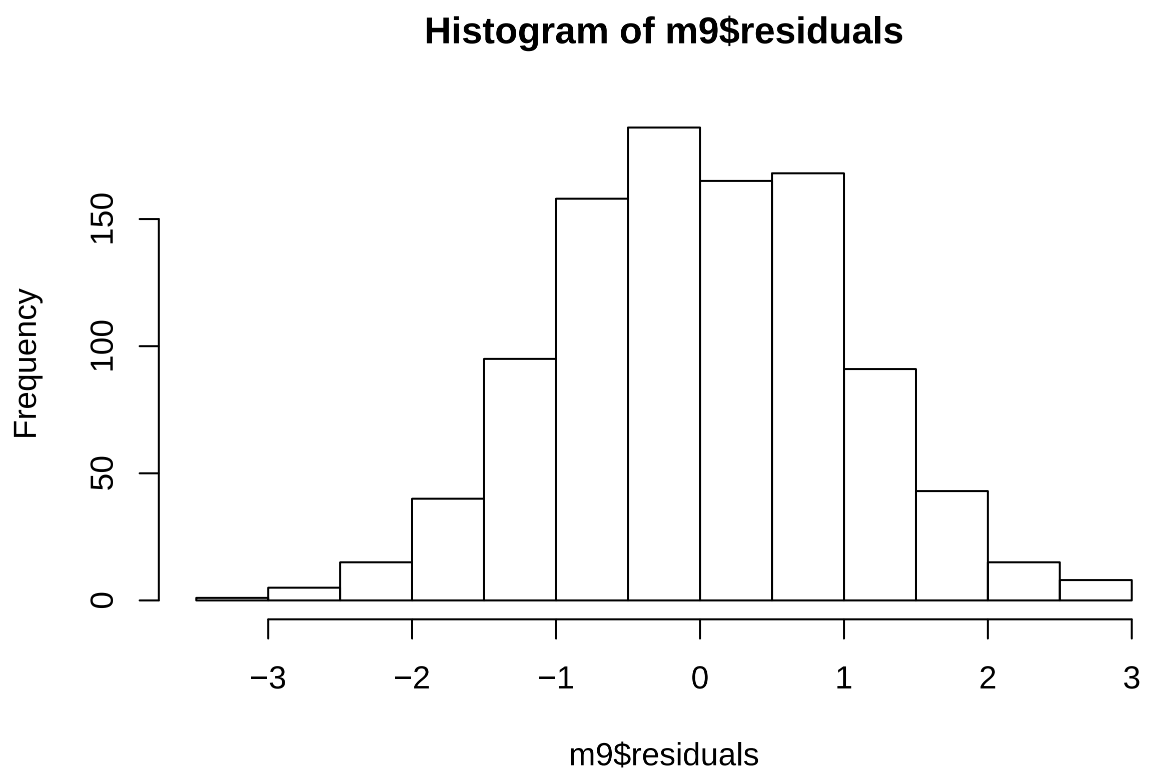
```r
pacf(series.train)
```



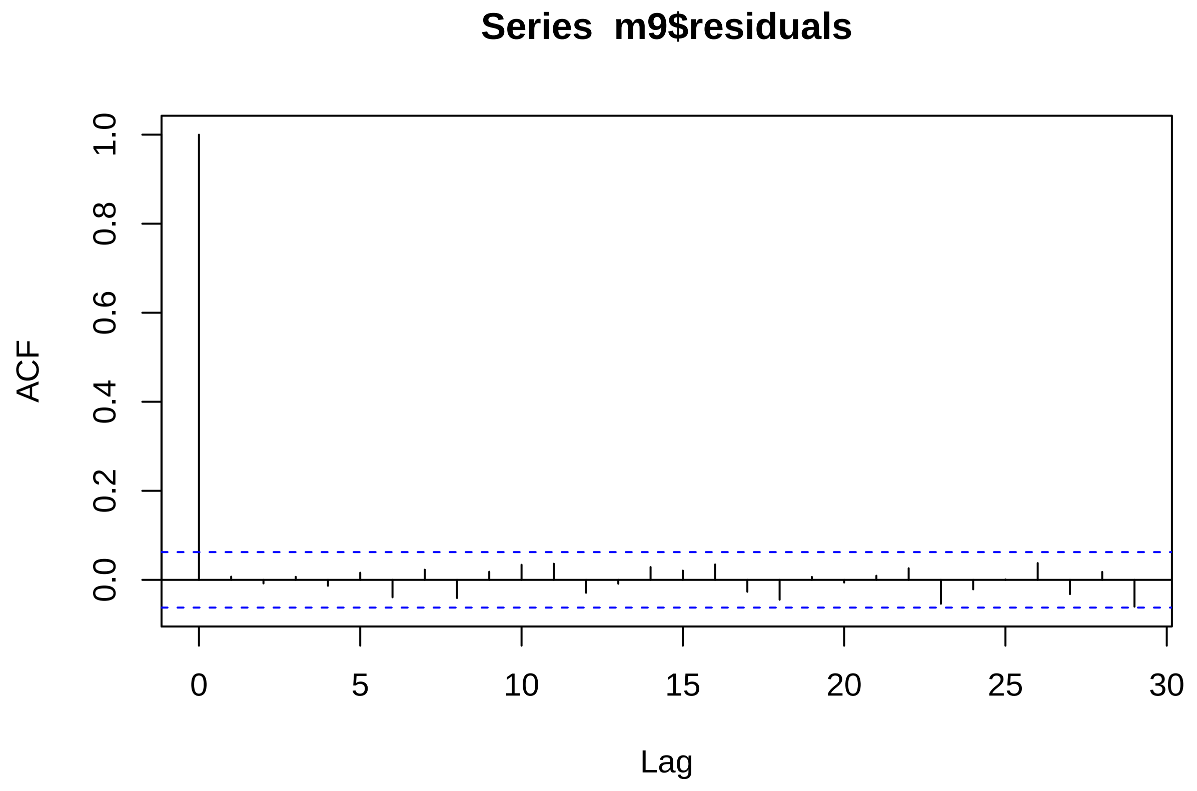
```
## Model this as an AR model only!
order.aic.list <- list()
for (p in 1:15) {
    m0 <- Arima(series.train, order = c(p, 0, 0), method = "ML")
    order.aic.list[[p]] <- data.frame(order = p, aic = m0$aic)
}

order.aic.df <- bind_rows(order.aic.list)
plot(order.aic.df$order, order.aic.df$aic, type = "l")
```



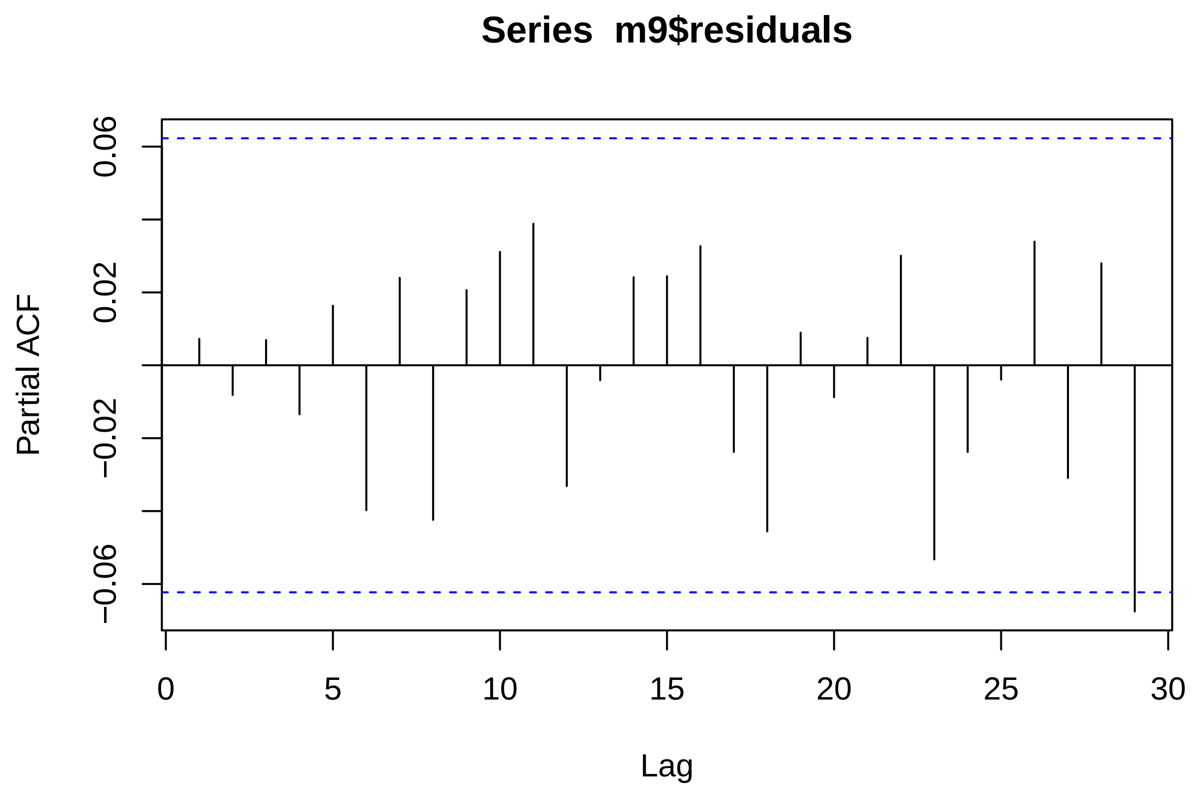
```
## Note, AIC is minimized with AR(9) and that the AIC declines
## pretty slowly after AR(3)


m9 <- Arima(series.train, order = c(8, 0, 0), method = "ML")

# plot(m9$residuals)
hist(m9$residuals)
```



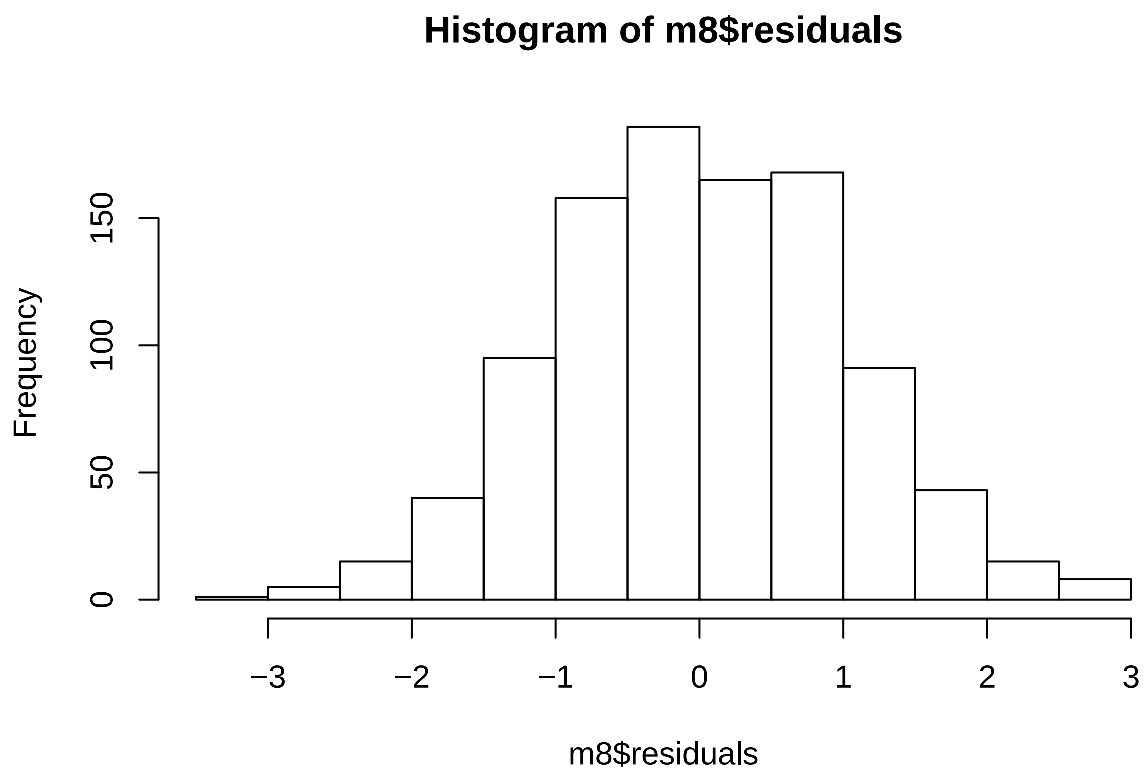
```
acf(m9$residuals)
```



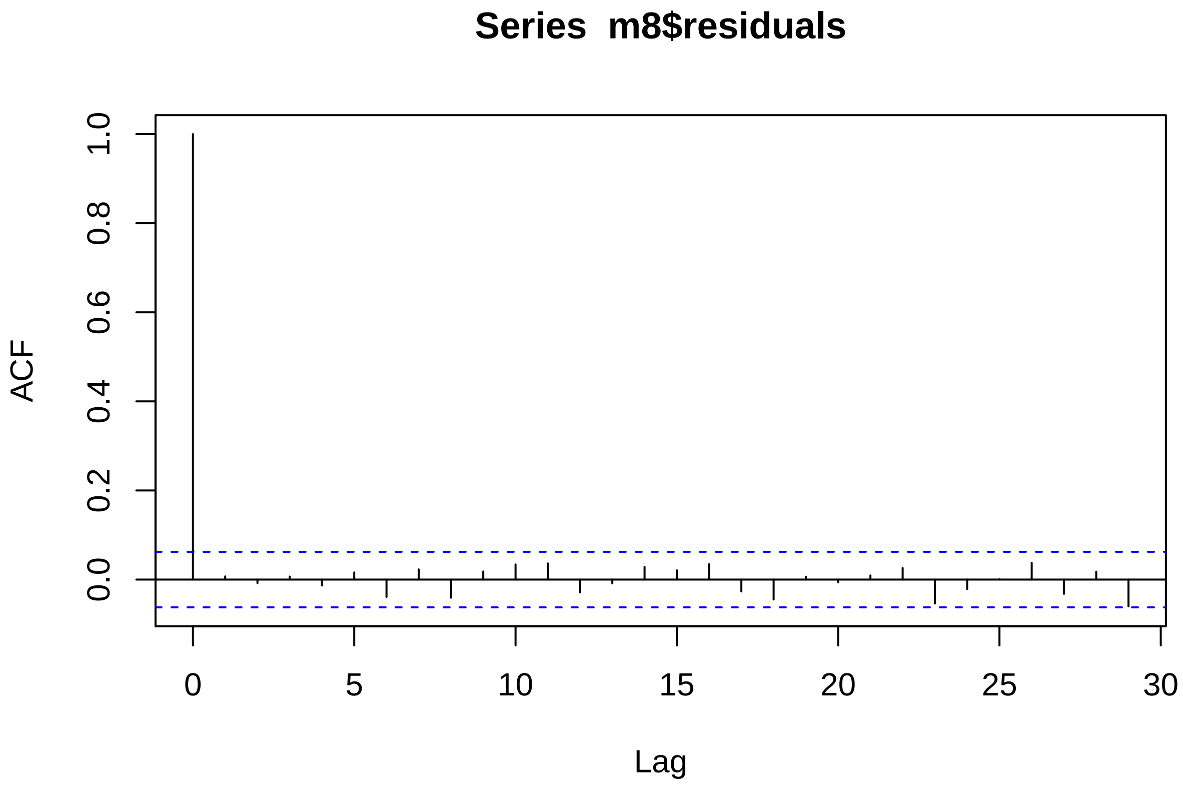
```
pacf(m9$residuals)
```



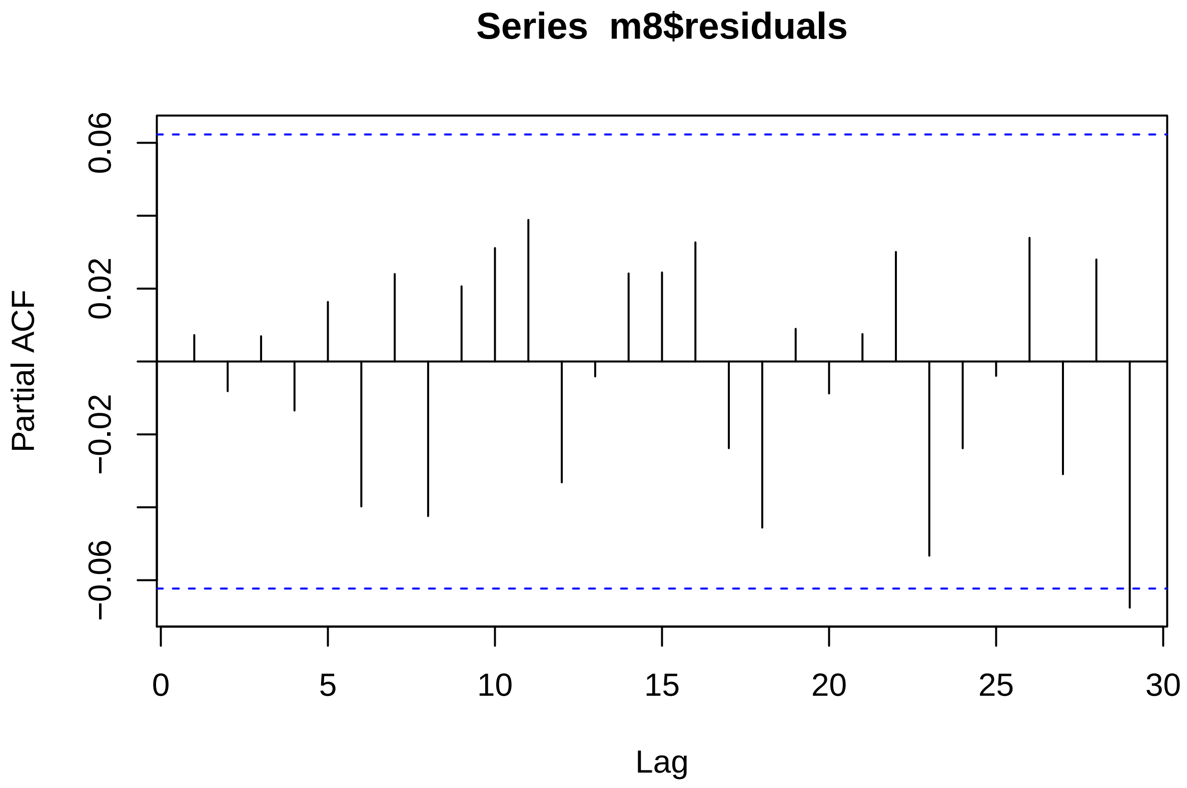
```r
m8 <- Arima(series.train, order = c(8, 0, 0), method = "ML")

# plot(m9$residuals)
hist(m8$residuals)
```



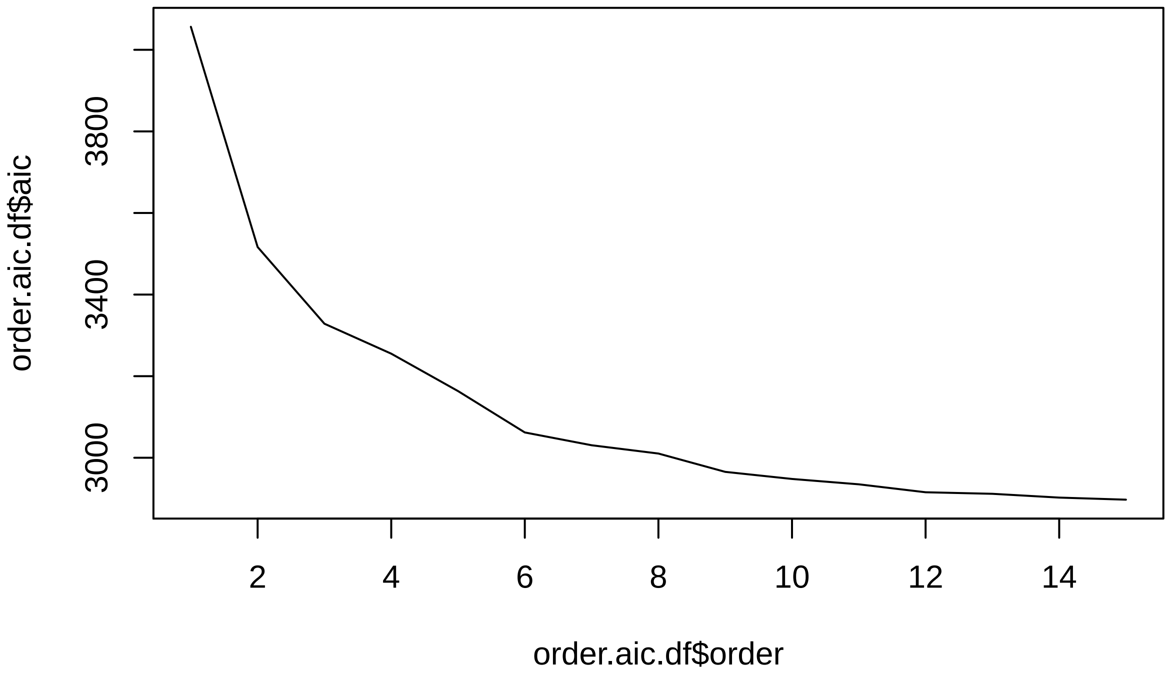
```r
acf(m8$residuals)
```



```r
pacf(m8$residuals)
```



```r
## Check the residuals for AR(9) and smaller until you get an
## AR model you think is correct.

# Repeat this exercise with an MA model

order.aic.list <- list()
for (q in 1:15) {
    m0 <- Arima(series.train, order = c(0, 0, q), method = "ML")
    order.aic.list[[q]] <- data.frame(order = q, aic = m0$aic)
}

order.aic.df <- bind_rows(order.aic.list)
plot(order.aic.df$order, order.aic.df$aic, type = "l")
```



```r
order.aic.df$order[order.aic.df$aic == min(order.aic.df$aic)]
```

```
## [1] 15
```

Breakout Session 3

This time-series was simulated from an ARMA(4,2) process. We now can compare three different models based on forecasting accuracy: AR(p), MA(q), and ARMA(4,2). Let's compare each model based on their ability to predict the test-data, which has 10 observations. Fill in the neccessary code below and answer the following questions:

(1) Based on your analysis, which of the three models is most apporipriate for this data?

The ARMA(4,2) does just as good of job as the AR(9) but is much simpler.

(2) Model the data using the *auto.arima()* function. Does this function yield an ARMA(4,2) as well? Why or why not?

No, it does not. It yields an Arima(3,1,2). The auto.arima function selects the best model according to some IC criteria. Strictly speaking, it is not kosher to compare AIC or BIC values if the data are differnced or not.

(3) Suppose that you had to generte a forecast from the training dataset without using these time-series tools. How would you do that? What would the test-error of that statistic be?

I could only use the mean value. I would get an RMSE of 4, which is much worse than what I got using an ARMA model.

(4) For the AR(p) and MA(q) models, create a 30 step-ahead forecast and plot the values from those forecasts. What do you notice about the forecasts as the time-horizon increases?

Straighforward.

```r
ar.order <- 8
ma.order <- 15

ar.model <- Arima(series.train, order = c(ar.order, 0, 0), method = "ML")

ma.model <- Arima(series.train, order = c(0, 0, ma.order), method = "ML")

arma.model <- Arima(series.train, order = c(4, 0, 2), method = "ML")

fcst.ar <- forecast.Arima(ar.model, h = 10)
fcst.ma <- forecast.Arima(ma.model, h = 10)
fcst.arma <- forecast.Arima(arma.model, h = 10)
# fcst.ma INSERT CODE HERE fcst.arma INSERT CODE HERE

# USE THIS FUNCTION
calculate_rmse <- function(fcast, test) {
    rmse <- sqrt(mean((fcast - test)^2))
}

print(calculate_rmse(fcst.ar$mean, series.test))
```

```
## [1] 0.9153583
```

```r
print(calculate_rmse(fcst.ma$mean, series.test))
```

```
## [1] 1.886571
```

```r
print(calculate_rmse(fcst.arma$mean, series.test))
```

```
## [1] 0.9185351
```

```r
print(calculate_rmse(rep(mean(series.train), 10), series.test))
```

```
## [1] 4.00177
```

```r
auto.arima(series.train)
```

```
## Series: series.train
## ARIMA(3,1,2)
##
## Coefficients:
##           ar1      ar2      ar3     ma1      ma2
##       -0.2473  -0.3363  -0.1650  0.5541  -0.2008
## s.e.   0.1551   0.0572   0.0771  0.1553   0.1037
##
## sigma^2 estimated as 1.014:  log likelihood=-1408.16
## AIC=2828.32   AICc=2828.4   BIC=2857.7
```

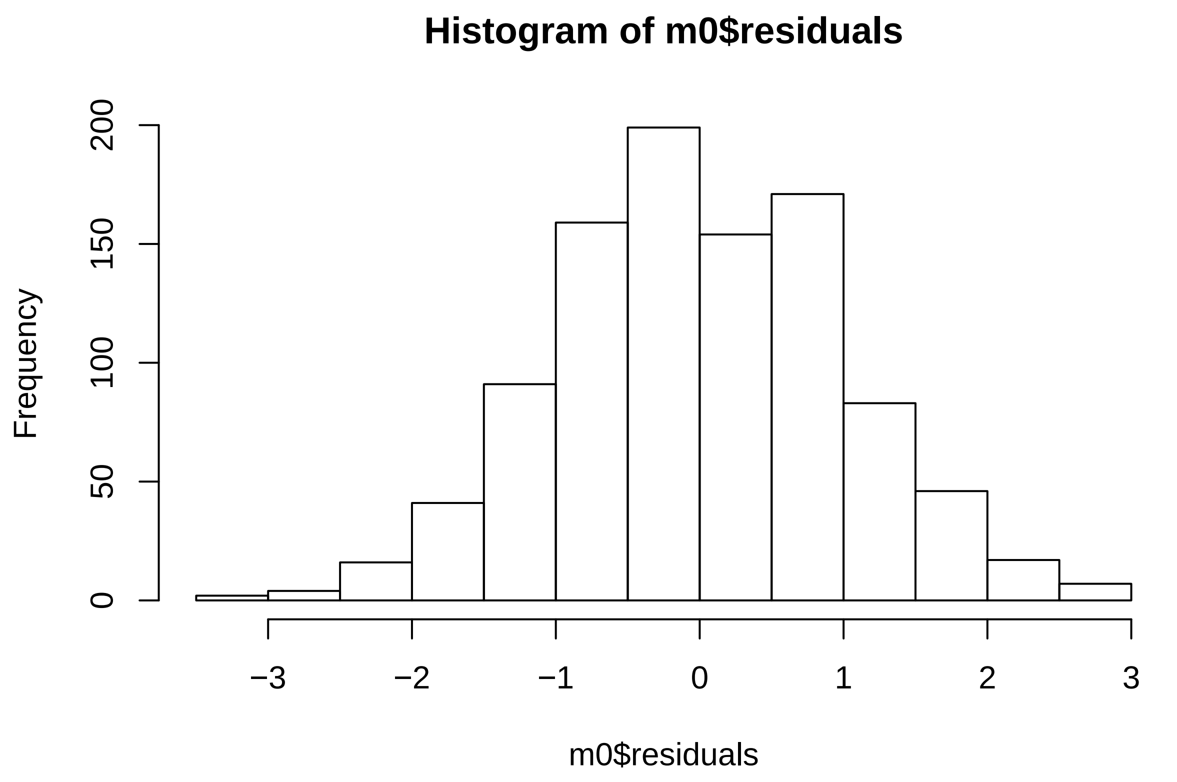
Modeling the ARMA series

1. AIC is minimized at ARMA(4,1)
2. Check reiduals of ARMA(4,1), ARMA(4,0), ARMA(3,1), ARMA(3,2)
3. Residuals of ARMA(4,1) and ARMA(3,2) are very similar. Others do not have residuals resembling white noise.
4. Both of these models have similar forecasting accuracy. I would probably select ARMA(4,1) because it uses more of the prior observations than ARMA(3,2), but both are more or less the same.

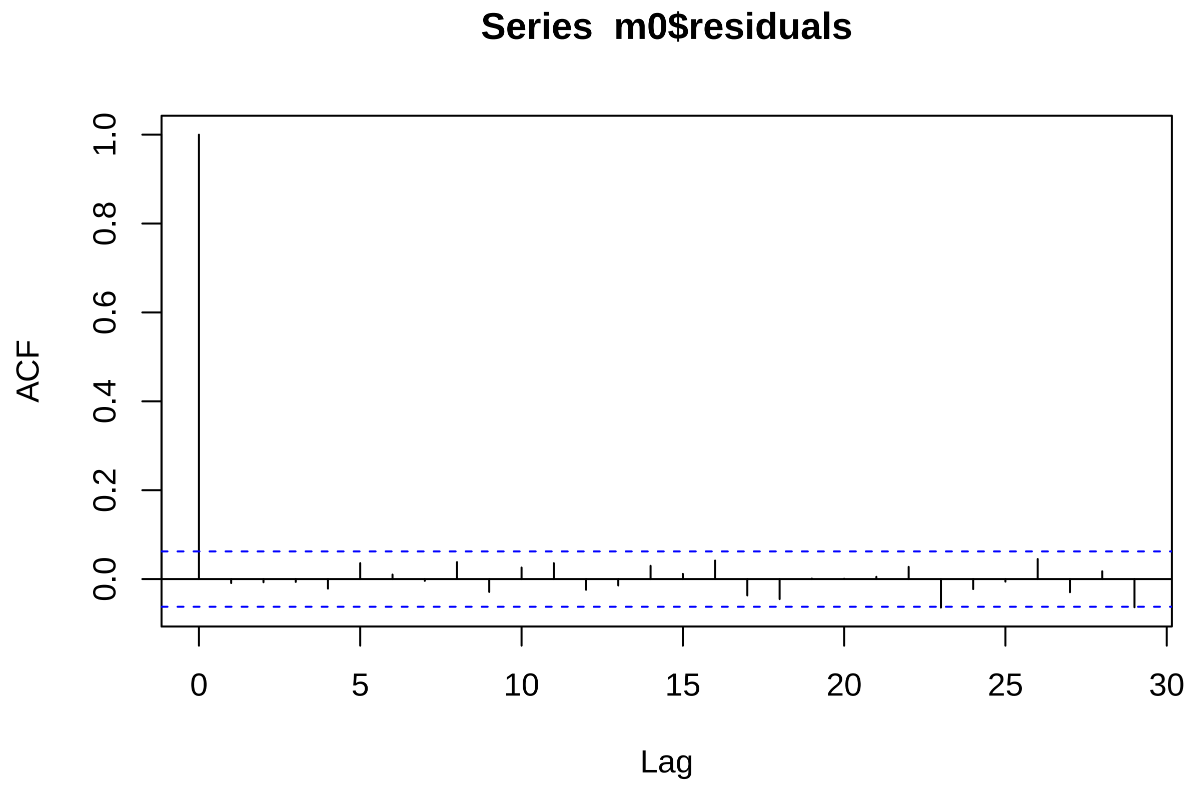
```r
for (p in 0:4) {
    for (q in 0:4) {

        m <- Arima(series.train, order = c(p, 0, q), method = "ML")
        print(c(p, q, m$aic))
    }
}
```

```
## [1]    0.000    0.000 5233.273
## [1]    0.000    1.000 4056.471
## [1]    0.000    2.000 3516.522
## [1]    0.00    3.00 3328.56
## [1]    0.000    4.000 3255.161
## [1]    1.000    0.000 3214.546
## [1]    1.000    1.000 3007.911
## [1]    1.000    2.000 2858.853
## [1]    1.000    3.000 2848.209
## [1]    1.000    4.000 2832.758
## [1]    2.000    0.000 3191.571
## [1]    2.000    1.000 2954.829
## [1]    2.000    2.000 2853.677
## [1]    2.000    3.000 2861.965
## [1]    2.000    4.000 2852.394
## [1]    3.000    0.000 2888.685
## [1]    3.000    1.000 2848.771
## [1]    3.000    2.000 2825.932
## [1]    3.000    3.000 2823.664
## [1]    3.000    4.000 2825.176
## [1]    4.000    0.000 2874.034
## [1]    4.000    1.000 2822.708
## [1]    4.000    2.000 2823.193
## [1]    4.000    3.000 2824.879
## [1]    4.00    4.00 2826.86
```

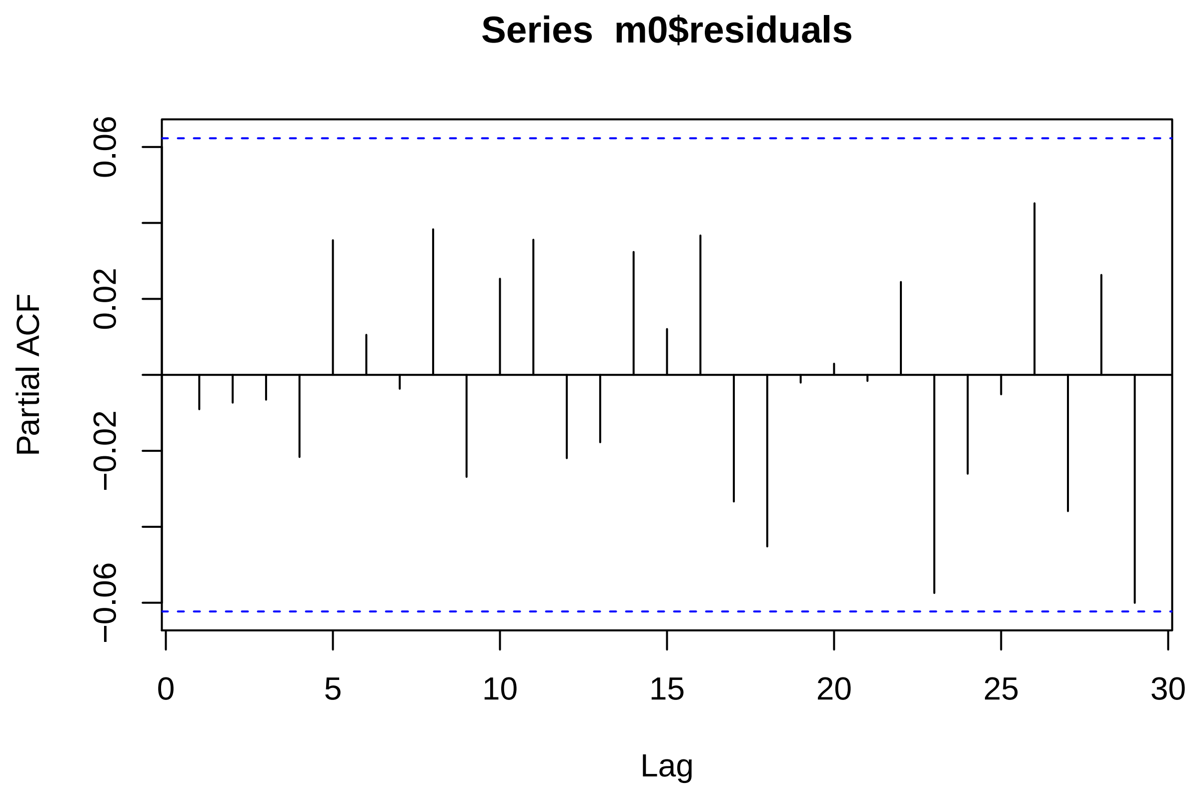
```r
m0 <- Arima(series.train, order = c(4, 0, 1), method = "ML")
hist(m0$residuals)
```



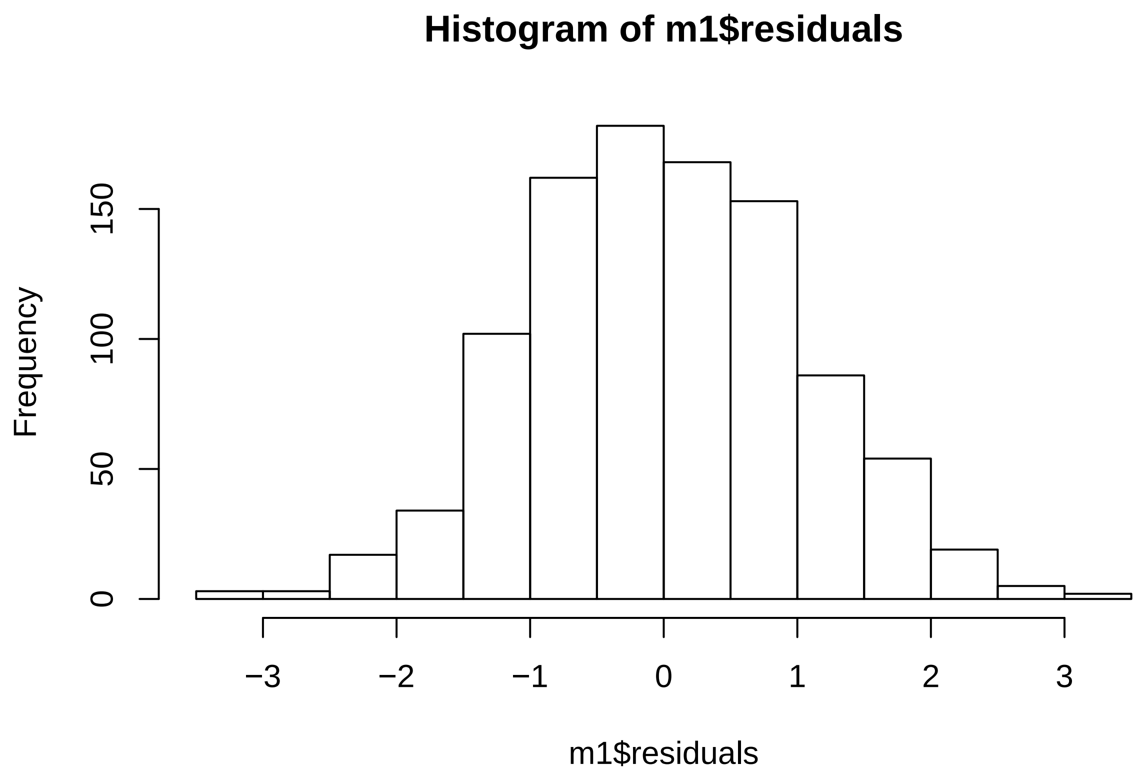
```r
acf(m0$residuals)
```



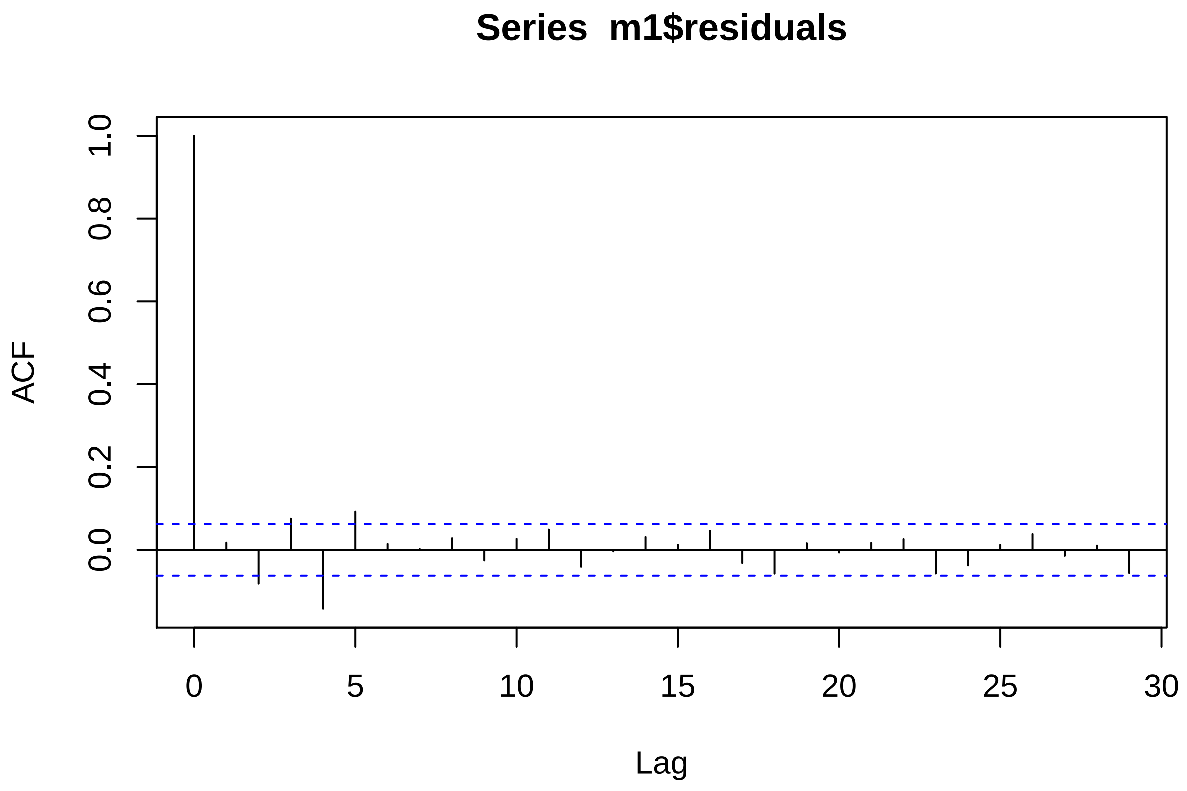
```r
pacf(m0$residuals)
```



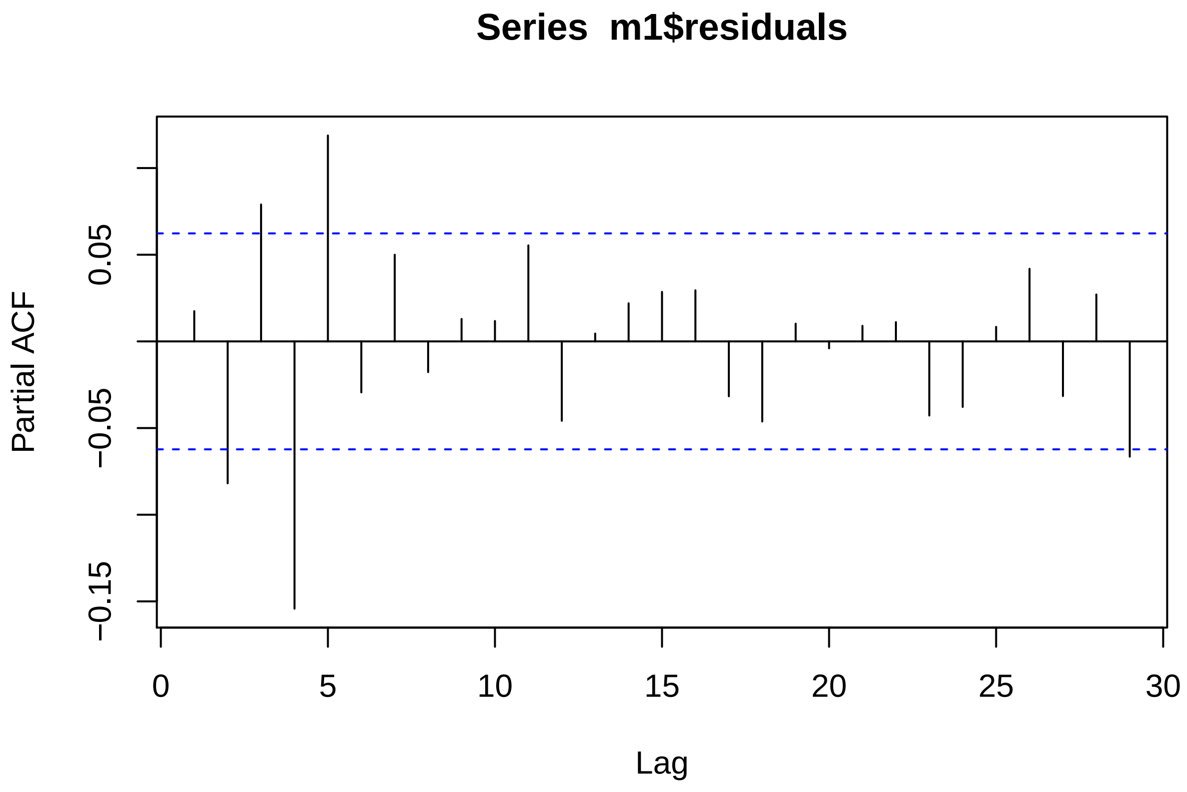
```r
m1 <- Arima(series.train, order = c(4, 0, 0), method = "ML")
hist(m1$residuals)
```



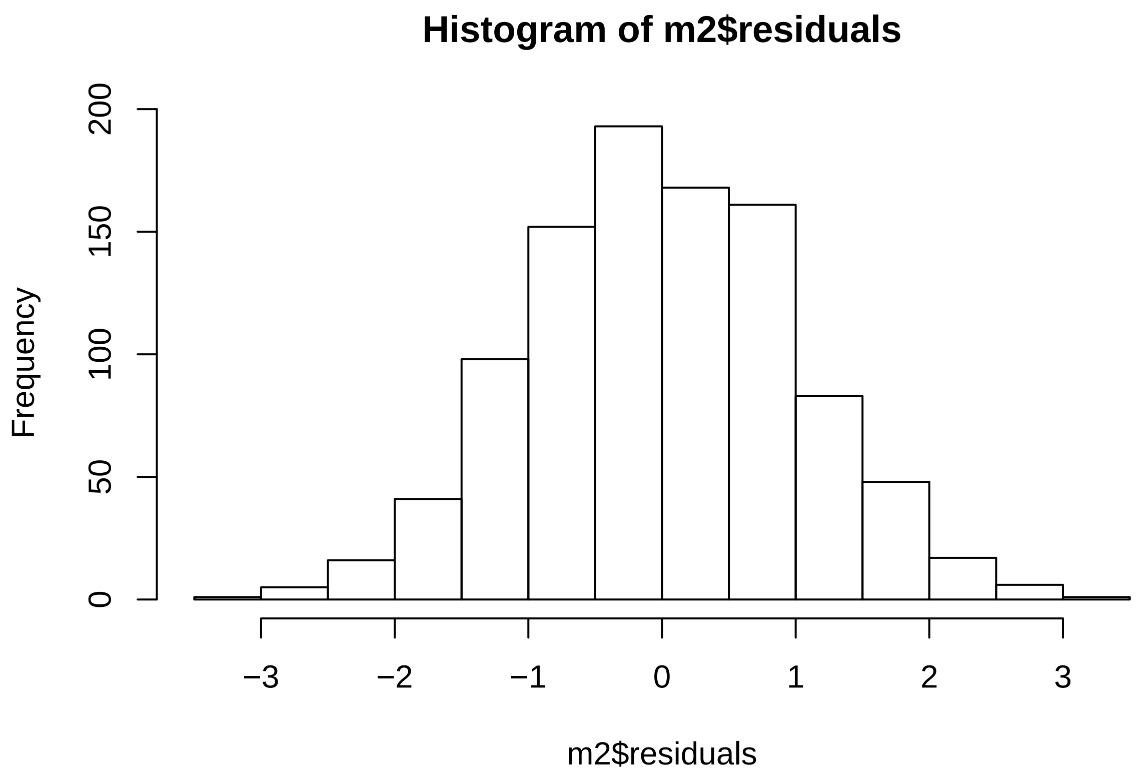
```r
acf(m1$residuals)
```



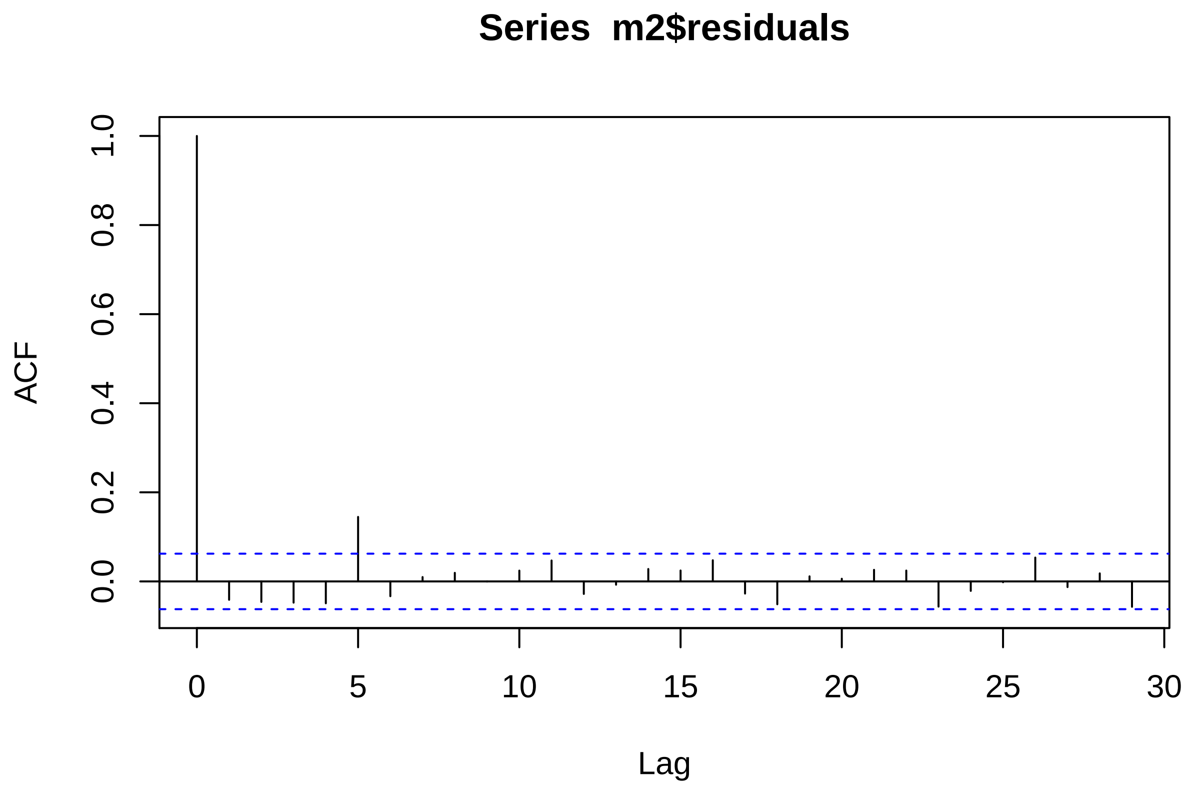
```r
pacf(m1$residuals)
```



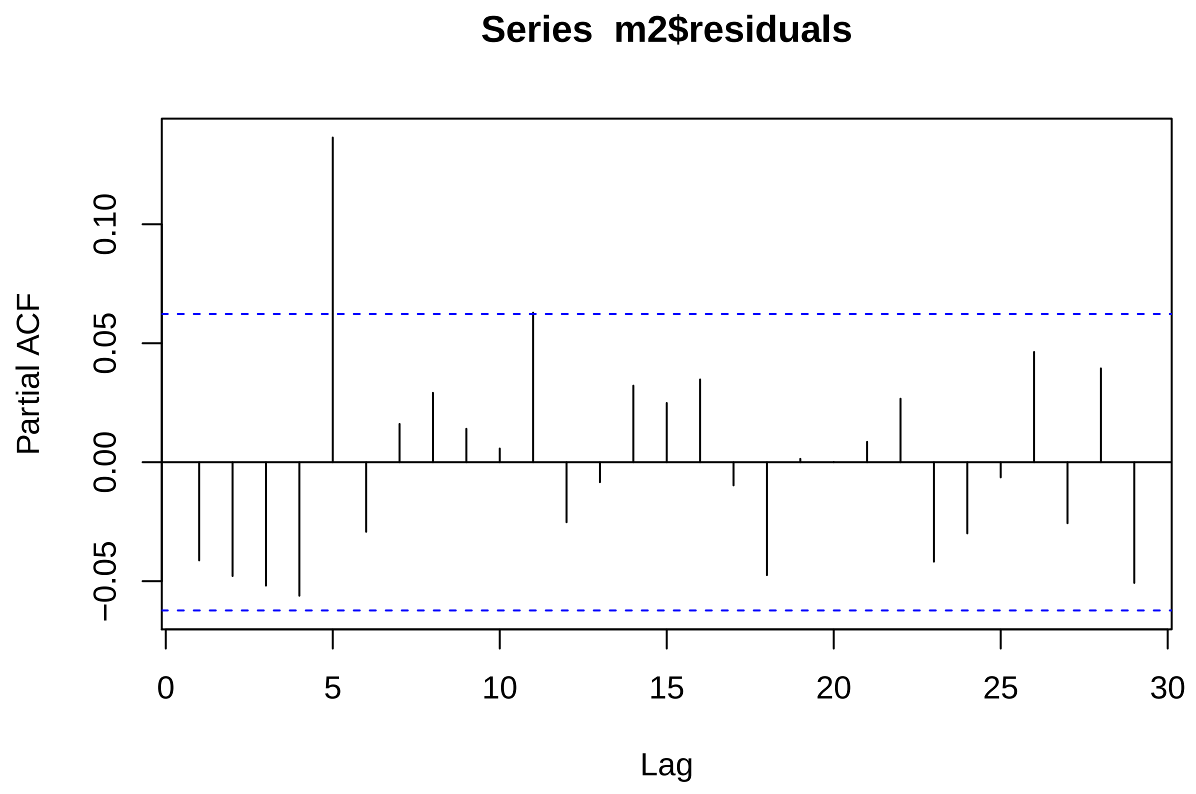
```r
m2 <- Arima(series.train, order = c(3, 0, 1), method = "ML")
hist(m2$residuals)
```



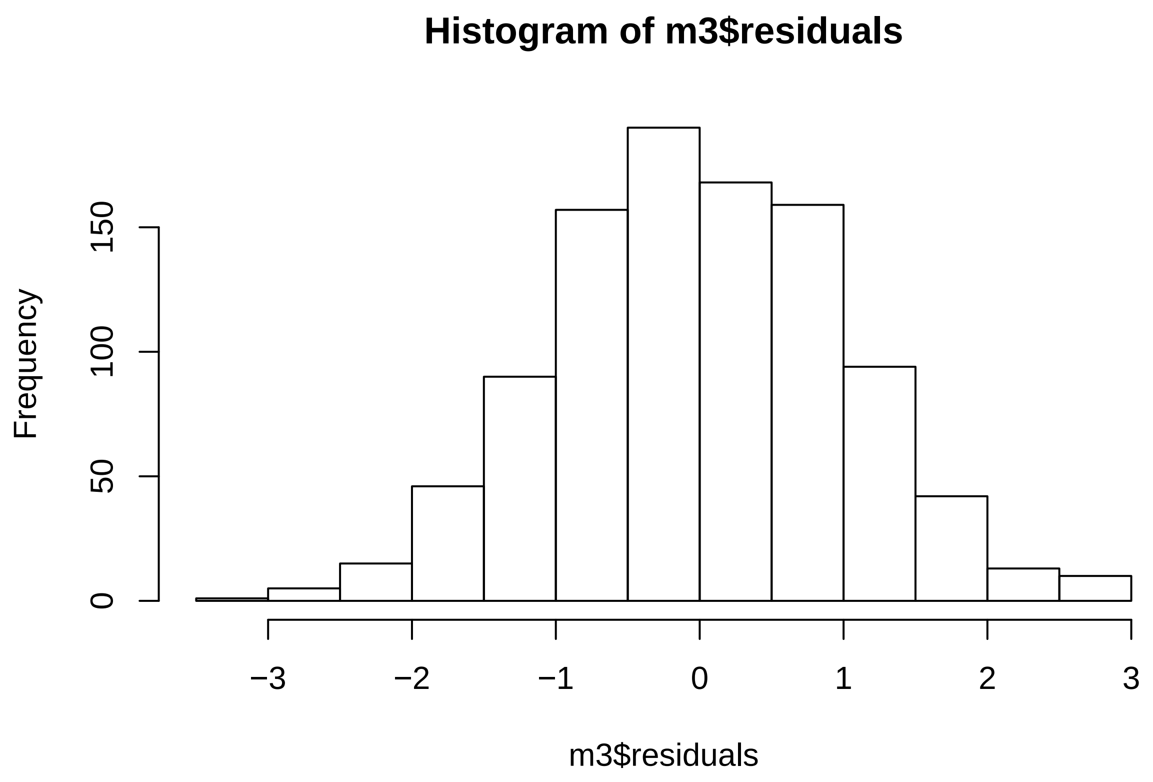
```r
acf(m2$residuals)
```



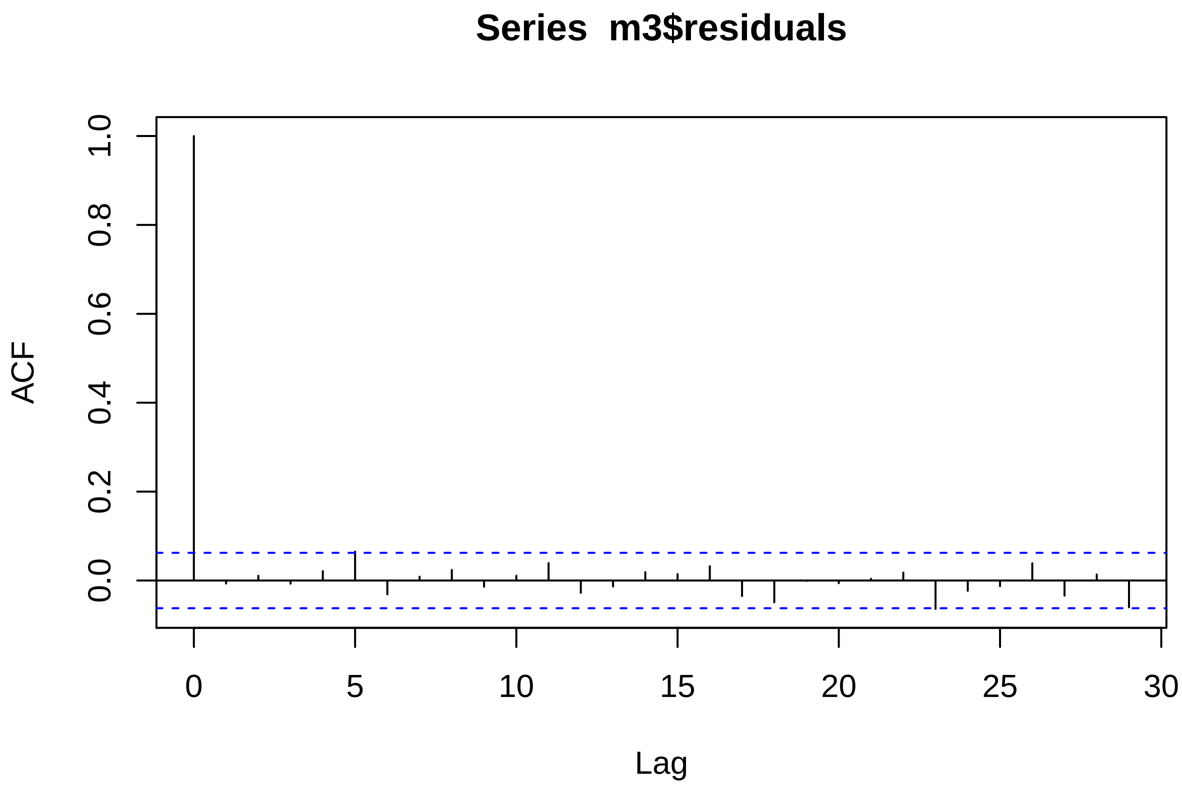
```r
pacf(m2$residuals)
```



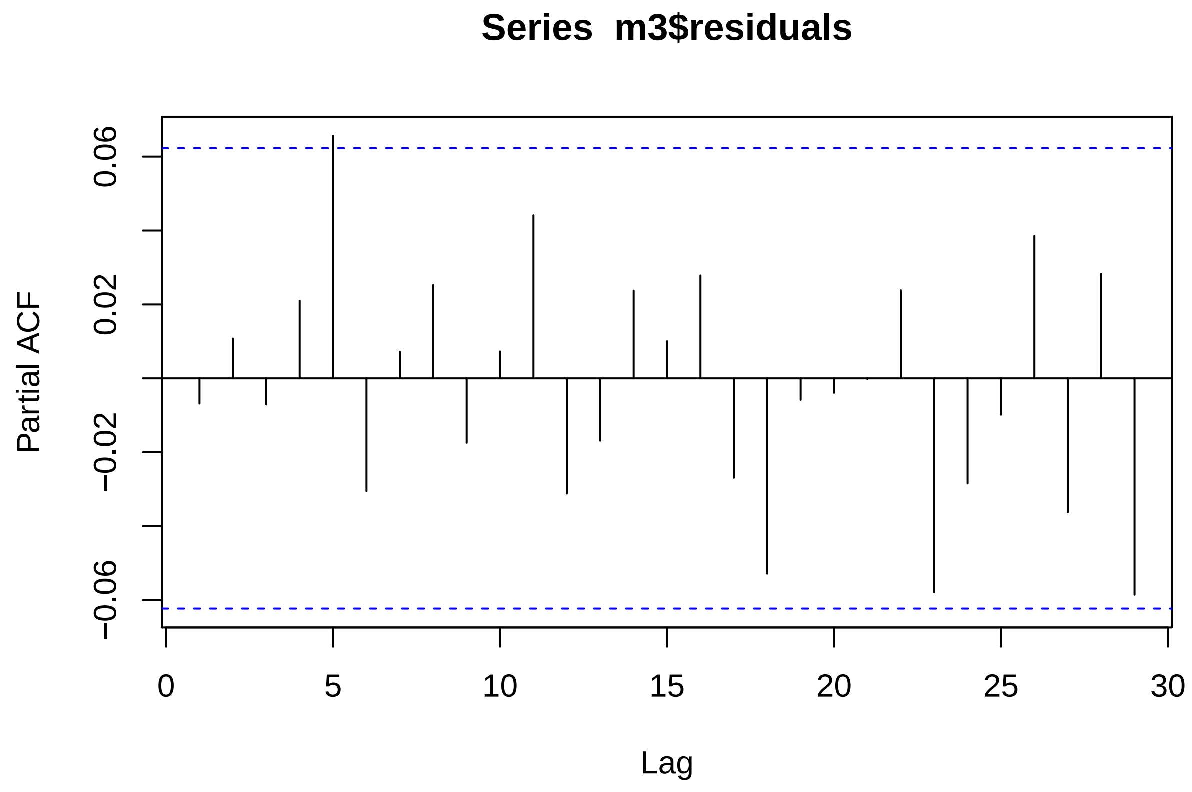
```r
m3 <- Arima(series.train, order = c(3, 0, 2), method = "ML")
hist(m3$residuals)
```



```r
acf(m3$residuals)
```



```r
pacf(m3$residuals)
```



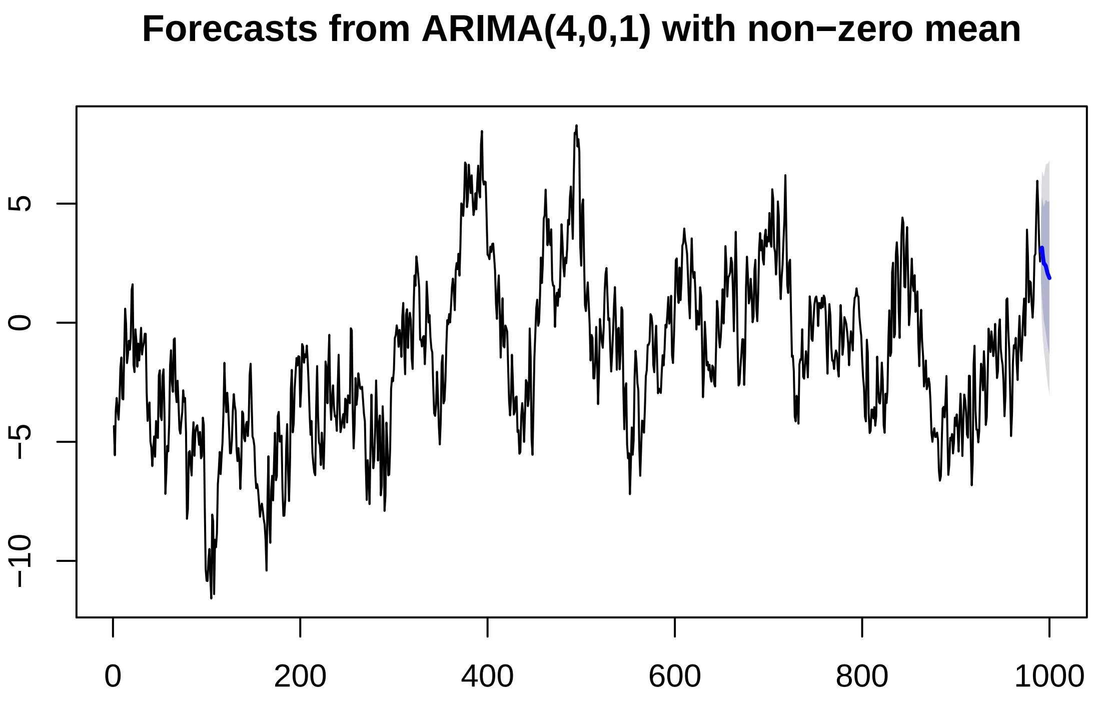
```r
f.arma41 <- forecast(m0, h = 10)
f.arma32 <- forecast(m3, h = 10)

print(calculate_rmse(f.arma41$mean, series.test))
```

```
## [1] 0.9099332
```

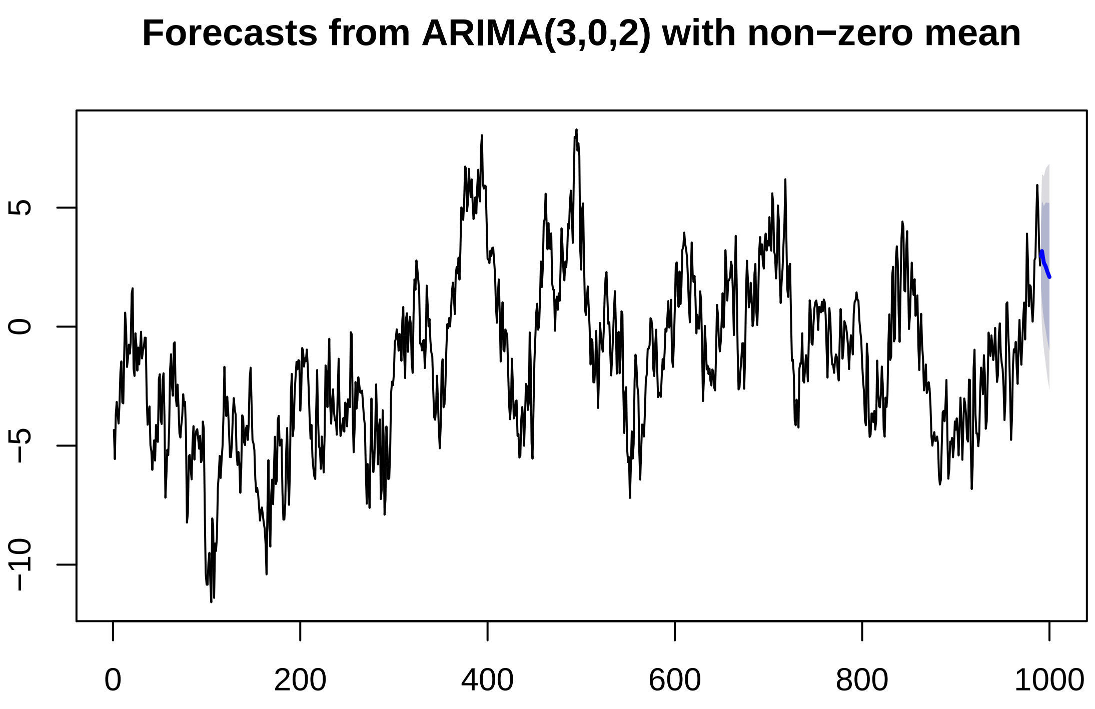
```r
print(calculate_rmse(f.arma32$mean, series.test))
```

```
## [1] 0.9002497
```

```r
plot(f.arma41)
```



```r
plot(f.arma32)
```


Forecasts from ARIMA(3,0,2) with non−zero mean
5
0
−5
−10
0
200
400
600
800
1000