THE UNIVERSITY OF QUEENSLAND AUSTRALIA | CREATE CHANGE

ECON3350 - Applied Econometrics for Macroeconomics and Finance

Tutorial 3: Forecasting Univariate Processes – II

Tutor: Francisco Tavares Garcia

Report 1 – due 11 April - Instructions

Instructions

The project consists of three research questions. Please answer all questions as clearly, concisely and completely as possible. Each question is worth 50 marks, for a total of 150 marks. This report will constitute 20% of your overall grade in this course.

We suggest that you use R for all empirical work involved. However, you should be able to use another statistical software (e.g. Eviews, Stata, Python, etc.) without a problem. If you do choose to work with an alternative software, please note that support for software-specific issues from the course coordinator and tutors may be very limited.

Please upload your report via the "Turnitin" submission link (in the "Assessment / Research Report" folder). Please note that hard copies *will not* be accepted. At the moment, the due date is 1:00 PM on 28 March 2025, but please check BlackBoard regularly for announcements regarding any changes to this. Your report should be a write-up of your answers (in PDF format, single-spaced, and in 12 font size).[1]

You are allowed to work on this assignment with others, i.e., you can discuss how to answer the questions with your classmate(s) and even use AI as a research tool. However, this is not a group assignment, which means that the report must be written individually and by you: you must answer all the questions in your own words and submit your report separately. The marking system will check for similarities and AI content, and UQ's student integrity and misconduct policies on plagiarism *strictly apply*.

Report 1 – due 11 April - Question 1

Questions

The dataset for Questions 1 and 2 is contained in `report1.csv`. The variables are quarterly time-series of macroeconomic indicators in Australia for the period 1995Q1—2023Q4 (116 observations). In particular, the dataset contains the following variables:

1. Use the data provided to choose three (3) ARIMA(p, d, q) models for inflation, π_t. Use each of these three models to forecast π_t for 2023 and 2024 (two years or equivalently eight quarters past the end of the sample). In doing so, please consider how such forecasts may be useful for policy, and consequently, ensure your inference is aligned with the underlying motivation. Make sure to address all potential sources of uncertainty on a conceptual level, and to the extent possible, quantitatively.

Report 1 – due 11 April - Question 2

2. Use the data provided to obtain inference on the stability of the term structure of interest rates. In particular, investigate the following questions:

   (a) Is there evidence of nonstationarity in inflation, Δp_t, or in any of the following four interest rates $\{r_{M1,t}, r_{M3,t}, r_{Y2,t}, r_{Y3,t}\}$?

   (b) Are there any identifiable equilibrium relationships among the four interest rates?

   (c) Are each of the following spreads stationary?

   - $s_{t,m3-m1} = r_{M3,t} - r_{M1,t}$,
   - $s_{t,y2-3m} = r_{Y2,t} - r_{M3,t}$,
   - $s_{t,y3-y2} = r_{Y3,t} - r_{Y2,t}$, and
   - $s_{t,y5-r} = r_{Y5,t} - r_t$

   (d) Use a regression of Δp_t on $s_{t,y5-r}$ estimated by ols to investigate support for a relationship between these two.

In answering these questions, please consider how the answers may be useful for policy, and consequently, ensure your inference is aligned with the underlying motivation. Also, please discuss the key assumptions underlying any conclusions you obtain.

Report 1 – due 11 April - Question 3

3. The dataset for Question 3 is contained in `report2.csv`. The variables are daily time-series of two equity returns and a measure of market volatility for the period 29/06/2011—28/06/2021 (2541 observations, note the absence of weekends and holidays). The dataset contains the following variables:

   (a) Use the data provided to obtain inference on the volatility of $r_{WES,t}$ and $r_{WPL,t}$. This should include discussion of any testing for the presence of volatility and model selection. Report only the important results that guide your conclusions, the estimated final model and estimated volatility for each process.

   (b) Compare and contrast the estimates of volatility from your models in part (a) to the $p_{VIX,t}$.

   (c) Investigate the probability of a return less than 0.01% for $r_{WES,t}$ and $r_{WBC,t}$ on each of the days 29/06/2021, 30/06/2021 and 1/07/2021.

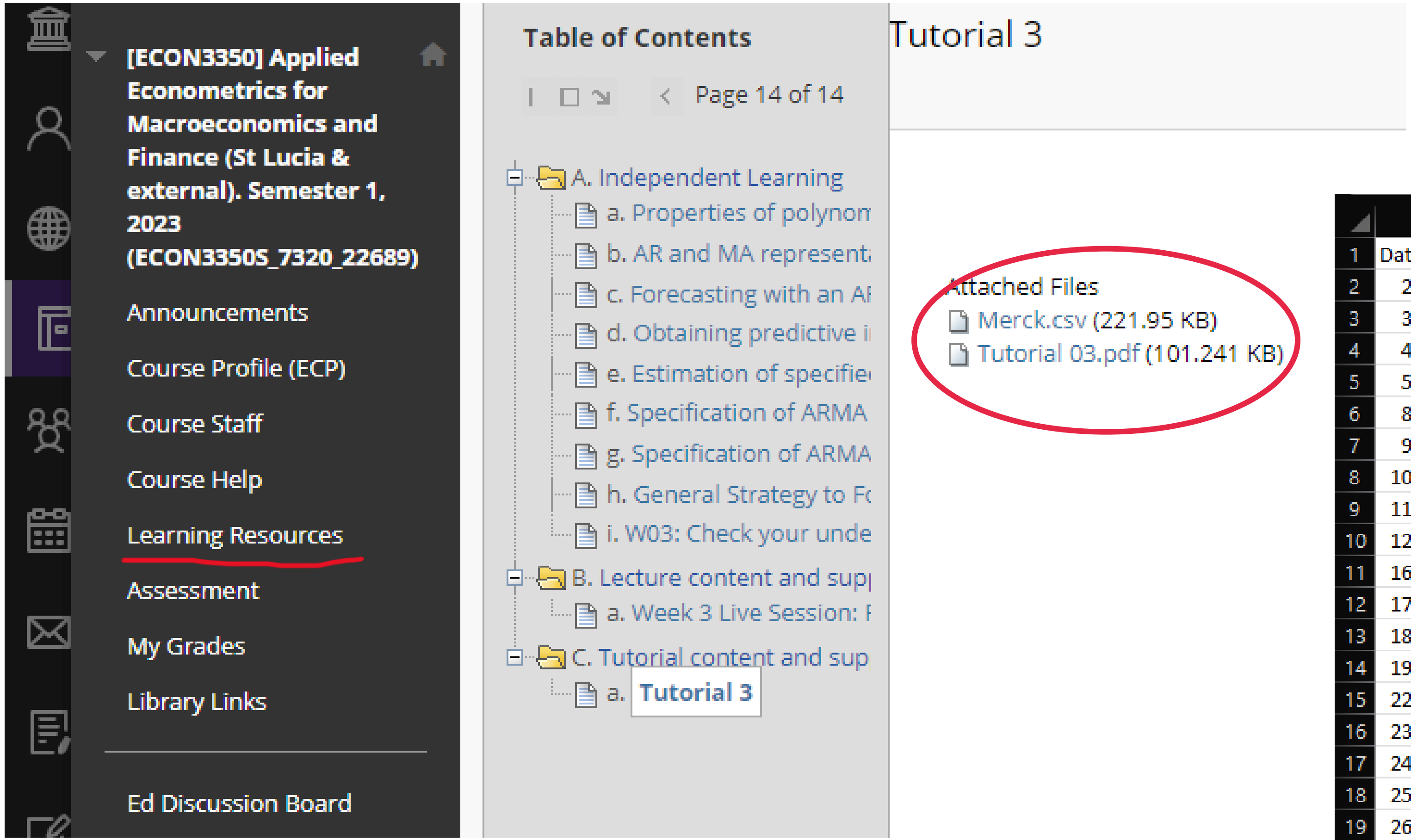
In answering these questions, please consider how the answers may be useful for risk management, and consequently, ensure your inference is aligned with the underlying motivation. Also, please discuss the key assumptions underlying any conclusions you obtain.

Tutorial 3: Forecasting Univariate Processes - II

At the end of this tutorial you should be able to:

- estimate a set of specified ARMA models using `for` loops;
- reduce the set of models using information criteria and residuals analysis;
- generate forecasts and predictive intervals for a specified ARMA(p, q);
- compare and evaluate forecasts obtained from different ARMA models.

Let’s download the tutorial and the dataset.



	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj_Close	Volume
2	2/01/2001	93.375	95.25	92.5	93	46.240479	6042900
3	3/01/2001	92.8125	93	88.125	89.125	44.313808	9836000
4	4/01/2001	88.4375	88.5625	83.75	85	42.262817	18325900
5	5/01/2001	85	85.8125	83.125	83.3125	41.423759	9350500
6	8/01/2001	84	84.5	82.6875	83.5	41.516994	4907600
7	9/01/2001	83.9375	85	83.0625	84	41.765598	6097100
8	10/01/2001	84.0625	84.8125	82.0625	83.1875	41.361622	6172900
9	11/01/2001	83.5	83.625	80.875	81.625	40.584728	6659900
10	12/01/2001	81.875	83.3125	81	81.4375	40.491486	5038600
11	16/01/2001	81.375	84.4375	80.5	83.3125	41.423759	5500900
12	17/01/2001	83	83.3125	81	81.25	40.398262	6459000
13	18/01/2001	81.5	84	81.375	82.875	41.206242	5191200
14	19/01/2001	82.125	83.6875	82.125	82.4375	40.988705	5380400
15	22/01/2001	83.625	83.625	82.125	82.3125	40.926544	4398300
16	23/01/2001	80.875	80.875	78.5625	79.5625	39.559231	11432900
17	24/01/2001	79.75	80	78.625	78.9375	39.248486	8412000
18	25/01/2001	79.5625	82	79.4375	81.875	40.709045	7299200
19	26/01/2001	82.75	84	81.9375	82.25	40.895493	5292300
20	29/01/2001	82.300003	83.449997	79.800003	80.309998	39.930901	4093500

Now, let's download the script for the tutorial.

- Copy the code from Github,
  - https://github.com/tavaresgarcia/teaching
- Save the scripts in the same folder as the data.

About Merck

NYSE - Delayed Quote • USD

Merck & Co., Inc. (MRK) Follow Compare

94.57 -0.14 (-0.15%) **94.64** +0.07 +(0.07%)

At close: March 14 at 4:00:02 PM EDT After hours: March 14 at 7:59:38 PM EDT

Start Trading >>
Plus500 CFD Service. Your capital is at risk.

1D 5D 1M 6M YTD 1Y 5Y All Key Events

Previous Close	94.71	Market Cap (intraday)	238.888B
Open	94.28	Beta (5Y Monthly)	0.36
Bid	94.61 x 800	PE Ratio (TTM)	14.03
Ask	94.80 x 1200	EPS (TTM)	6.74
Day's Range	93.45 - 94.88	Earnings Date	Apr 24, 2025
52 Week Range	81.04 - 134.63	Forward Dividend & Yield	3.24 (3.43%)
Volume	12,675,420	Ex-Dividend Date	Mar 17, 2025
Avg. Volume	14,600,911	1y Target Est	112.29

About Merck

Strong Q4 and 2024 worldwide sales growth

Merck

FULL YEAR WORLDWIDE SALES[1]

$64.2B

+7% growth
+10% ex-Exchange[2]

Q4 2024 WORLDWIDE SALES[1]

$15.6B

+7% growth
+9% ex-Exchange[2]

Human Health

$14.0B

+7% growth
+8% ex-Exchange[2]

Animal Health

$1.4B

+9% growth
+13% ex-Exchange[2]

Advancing and broadening our diverse pipeline

Infectious Disease

- FDA acceptance of BLA for clesrovimab to help protect infants from RSV
- Topline results for islatravir + doravirine for HIV treatment

Oncology

- Positive topline results for **subcutaneous pembrolizumab**
- Licensed **MK-2010**, a PD-1/VEGF bispecific antibody, from LaNova

Cardiometabolic

- Positive topline results for **WINREVAIR** from ZENITH trial
- Licensed **MK-4082**, an oral GLP-1 receptor agonist, from Hansoh

1. The file `Merck.csv` contains daily data of stock prices of Merck & Co., Inc. (MRK) during 2001–2013. In what follows, we use y_t to denote the adjusted closing prices (`Adj_Close` in the data) at time t.

   (a) Load the data into R and construct a data set with observations in the range 1 January 2011—31 January 2012.

Solution We will need to install and load the `forecast` library in R.

```
library(forecast)
```

Now, load the data using the `read.delim` command with the `sep = ","` option as it is comma delimited.

```
mydata <- read.delim("Merck.csv", header = TRUE,  sep = ",")
```

Use `Date` (first column) in the data set to select the required range, then save `Adj_Close` as the y_t variable.

```
sel_sample <- mydata$Date >= as.Date("2011-01-01") &
              mydata$Date <= as.Date("2012-01-31")
y <- as.matrix(mydata$Adj_Close[sel_sample])
```

(b) Construct the following variables:

- changes in prices: $\Delta y_t = y_t - y_{t-1}$;
- log returns: $r_t = \log(y_t/y_{t-1})$.

Solution Differencing is easily done with `diff` command and lagging a series can be done with indexing operations (although there are also other ways).

```
Dy <- diff(y)
r <- as.matrix(log(y[2:nrow(y)]) - log(lag(y[1:nrow(y) - 1])))

colnames(y) <- "Stock Prices of Merck (MRK)"
colnames(Dy) <- "Changes in Stock Prices"
colnames(r) <- "Log Returns"
```

(c) Draw time series plots of y_t, Δy_t and r_t; comment on the stationarity of the processes that may have generated these observations.

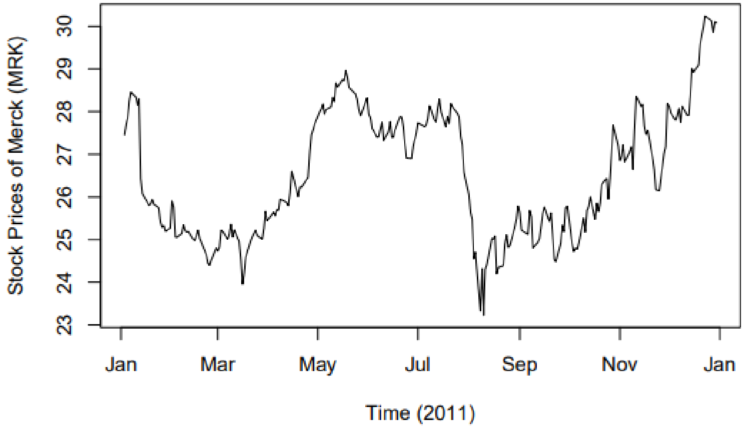
Solution Use the `Date` variable in the data set to select a subsample for 2011, then plot each series using the `plot` command.

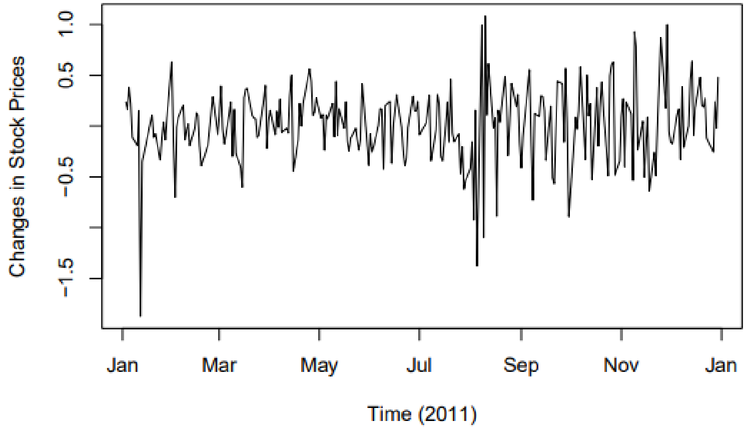
```
sel2011 <- mydata$Date[sel_sample] <= as.Date("2011-12-31")
dates = as.Date(mydata$Date[sel_sample])
plot(dates[sel2011], y[sel2011], type = "l", xlab = "Time (2011)",
     ylab = colnames(y))
```

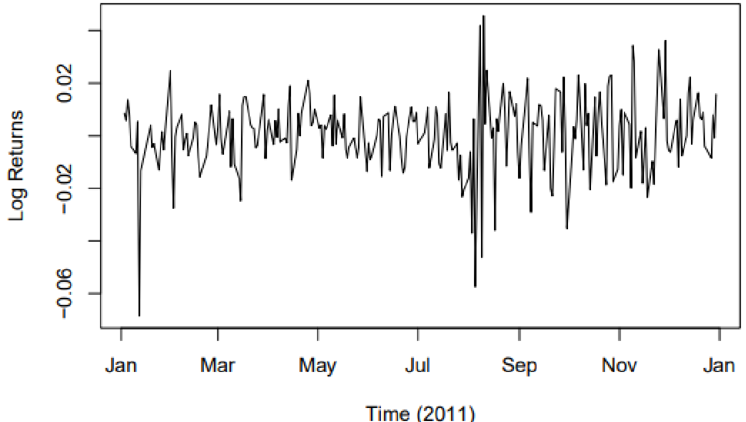
```
plot(dates[sel2011], Dy[sel2011], type = "l", xlab = "Time (2011)",
     ylab = colnames(Dy))
```

```
plot(dates[sel2011], r[sel2011], type = "l", xlab = "Time (2011)",
     ylab = colnames(r))
```

The process $\{y_t\}$ is likely not stationary as its mean appears to vary over time. The process $\{\Delta y_t\}$ seems to have a zero mean, but its variance may depend on time. We will cover time-varying variance later in the course.







(d) Compute and plot the sample ACFs and PACFs of y_t and Δy_t. Comment on your findings.

Solution Use the `acf` and `pacf` commands as in Tutorial 1.

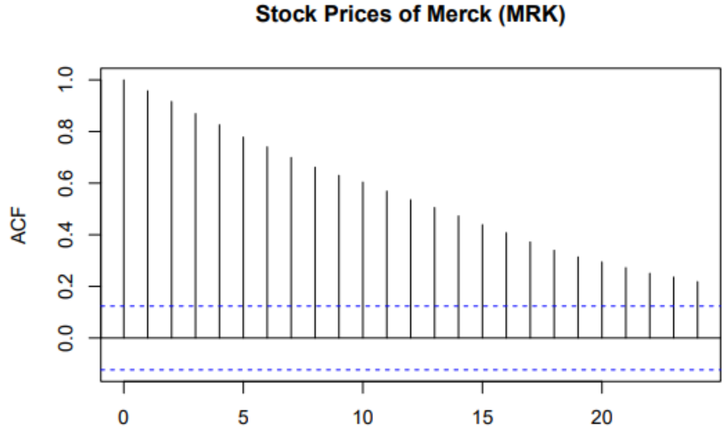
```
acf(y[sel2011], main = colnames(y))
```

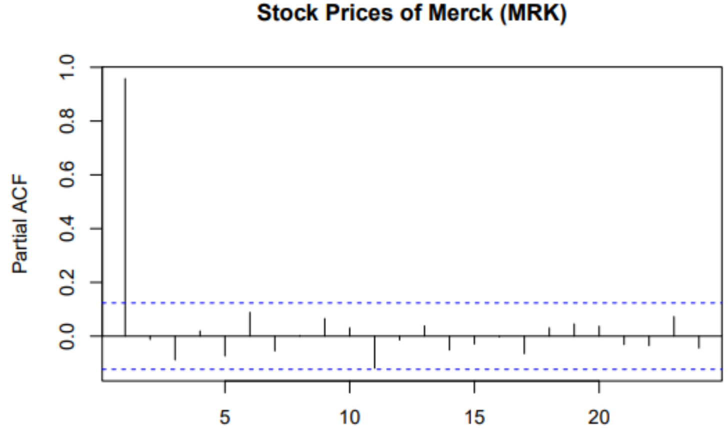
```
pacf(y[sel2011], main = colnames(y))
```

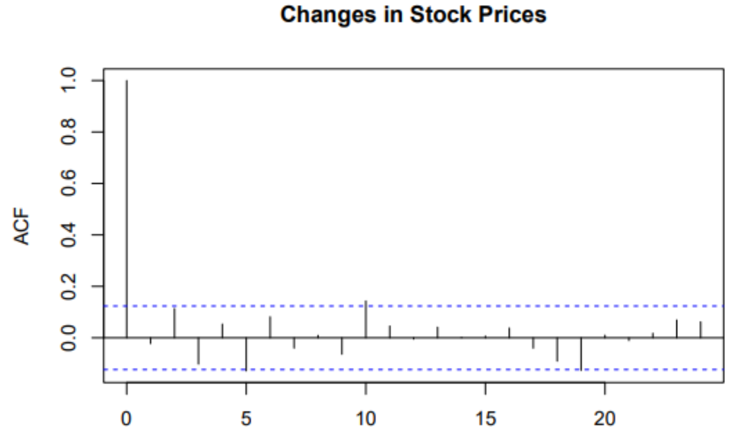
```
acf(Dy[sel2011], main = colnames(Dy))
```

For prices (y_t), the SACF decays but the SPACF drops to zero after one lag. Also, the first PAC is near 1, suggesting an AR(1) model of the form $y_t = y_t + \epsilon_t$. You should be able to see a connection to the implied model for Δy_t.

For price changes (Δy_t), the SPACF drops to zero after one lag but the SACF oscillates. Also, the first AC is near 1, suggesting an MA(1) model of the form $\Delta y_t = \epsilon_{t-1} + \epsilon_t$. You should notice the somewhat contradictory conclusions we have reached with this line of reasoning. Can you think of any possible explanations?







(e) Propose and estimate 25 ARMA(p, q) models for Δy_t.

Solution The `Arima` command from the `forecast` package makes it easy to estimate a specified ARMA(p, q). We can put this command inside a nested for loop to process a lot of ARMA models quickly. For this exercise, we estimate models with combinations of $p = 0, \ldots, 4$ and $q = 0, \ldots, 4$. For each estimated model, we extract the estimated AIC and BIC, then save these in a neat matrix, which we call `ic`, to be examined *ex-post*.

```
ic <- matrix( nrow = 25, ncol = 4 )
colnames(ic) <- c("p", "q", "aic", "bic")
for (p in 0:4)
{
  for (q in 0:4)
  {
    fit_p_q <- Arima(Dy, order = c(p, 0, q))
    ic[p * 5 + q + 1,] = c(p, q, fit_p_q[["aic"]], fit_p_q[["bic"]])
  }
}
```

	p	q	aic	bic
1	0	0	221.1652	228.3695
2	0	1	222.9201	233.7265
3	0	2	221.3797	235.7882
4	0	3	221.7444	239.7550
5	0	4	223.3386	244.9513
6	1	0	222.8619	233.6683
7	1	1	218.0345	232.4430
8	1	2	217.6980	235.7086
9	1	3	219.6798	241.2926
10	1	4	221.6430	246.8578
11	2	0	220.8859	235.2944
12	2	1	217.7698	235.7804
13	2	2	219.6627	241.2755
14	2	3	221.5682	246.7830
15	2	4	223.5379	252.3549

(f) Use the AIC and BIC to reduce the set of ARMA(p, q) models.

Solution We start by examining our constructed `ic` matrix.

```
print(ic)
```

The AIC and BIC appear to wildly disagree in their rankings of ARMA models. Unfortunately, there is no systematic approach to resolving this conflicting information!

We need to proceed in a sensible way, so we look at the top 10 specifications preferred by the AIC as well as the top 10 preferred by the BIC. This is easy to do by sorting the matrix `ic`.

```
ic_aic <- ic[order(ic[,3], decreasing = FALSE),][1:10,]
ic_bic <- ic[order(ic[,4], decreasing = FALSE),][1:10,]
print(ic_aic)
```

```
print(ic_bic)
```

For our purpose, next, it is sensible to select the models that make the top 10 in both lists. However, this is not a rule by any means—in other contexts, it is likely that a different approach will be more sensible.

```
adq_set = list(c(1, 0, 1), c(1, 0, 2), c(2, 0, 1), c(3, 0, 0))
```

```
> print(ic)
      p q      aic      bic
 [1,] 0 0 221.1652 228.3695
 [2,] 0 1 222.9201 233.7265
 [3,] 0 2 221.3797 235.7882
 [4,] 0 3 221.7444 239.7550
 [5,] 0 4 223.3386 244.9513
 [6,] 1 0 222.8619 233.6683
 [7,] 1 1 218.0345 232.4430
 [8,] 1 2 217.6980 235.7086
 [9,] 1 3 219.6798 241.2926
[10,] 1 4 221.6430 246.8578
[11,] 2 0 220.8859 235.2944
[12,] 2 1 217.7698 235.7804
[13,] 2 2 219.6627 241.2755
[14,] 2 3 221.5682 246.7830
[15,] 2 4 223.5379 252.3549
[16,] 3 0 219.9955 238.0061
[17,] 3 1 219.6965 241.3092
[18,] 3 2 221.4811 246.6960
[19,] 3 3 221.2772 250.0942
[20,] 3 4 219.9602 252.3793
[21,] 4 0 221.6630 243.2757
[22,] 4 1 221.6951 246.9099
[23,] 4 2 223.4237 252.2406
[24,] 4 3 219.5087 251.9277
[25,] 4 4 219.5611 255.5823
```

```
> print(ic_aic)
      p q      aic      bic
 [1,] 1 2 217.6980 235.7086
 [2,] 2 1 217.7698 235.7804
 [3,] 1 1 218.0345 232.4430
 [4,] 4 3 219.5087 251.9277
 [5,] 4 4 219.5611 255.5823
 [6,] 2 2 219.6627 241.2755
 [7,] 1 3 219.6798 241.2926
 [8,] 3 1 219.6965 241.3092
 [9,] 3 4 219.9602 252.3793
[10,] 3 0 219.9955 238.0061
> print(ic_bic)
      p q      aic      bic
 [1,] 0 0 221.1652 228.3695
 [2,] 1 1 218.0345 232.4430
 [3,] 1 0 222.8619 233.6683
 [4,] 0 1 222.9201 233.7265
 [5,] 2 0 220.8859 235.2944
 [6,] 1 2 217.6980 235.7086
 [7,] 2 1 217.7698 235.7804
 [8,] 0 2 221.3797 235.7882
 [9,] 3 0 219.9955 238.0061
[10,] 0 3 221.7444 239.7550
```

(g) Draw time series plots of the estimated residuals you obtained for the ARMA models selected in part (f). Comment on your findings. Run the Ljung-Box test (at the $\alpha = 5\%$ significance level) to test the white noise hypothesis on estimated residuals obtained from each ARMA in the set obtain in part (f) and report the test results. Use this information to identify the adequate set of specified ARMAs.

The First-Order Autoregressive Model

One simple way to model a stochastic process is with the "regression":

$$y_t = a_0 + a_1 y_{t-1} + u_t.$$

This is called the first order auto-regressive model, or AR(1).

To make it useful in practice, we need assumptions about u_t.

The "classical regression" assumptions are:

- Mean-independence: $\mathrm{E}(u_t \mid y_{t-1}, y_{t-2}, \dots) = 0.$
- Homoscedasticity: $\mathrm{Var}(u_t \mid y_{t-1}, y_{t-2}, \dots) = \sigma_u^2.$

Mean-independence is crucial, but homoscedasticity can be relaxed.

Mean-independence implies zero-autocorrelation: $\mathrm{corr}(u_t, u_{t-k}) = 0$ for $k = 1, 2, \dots.$

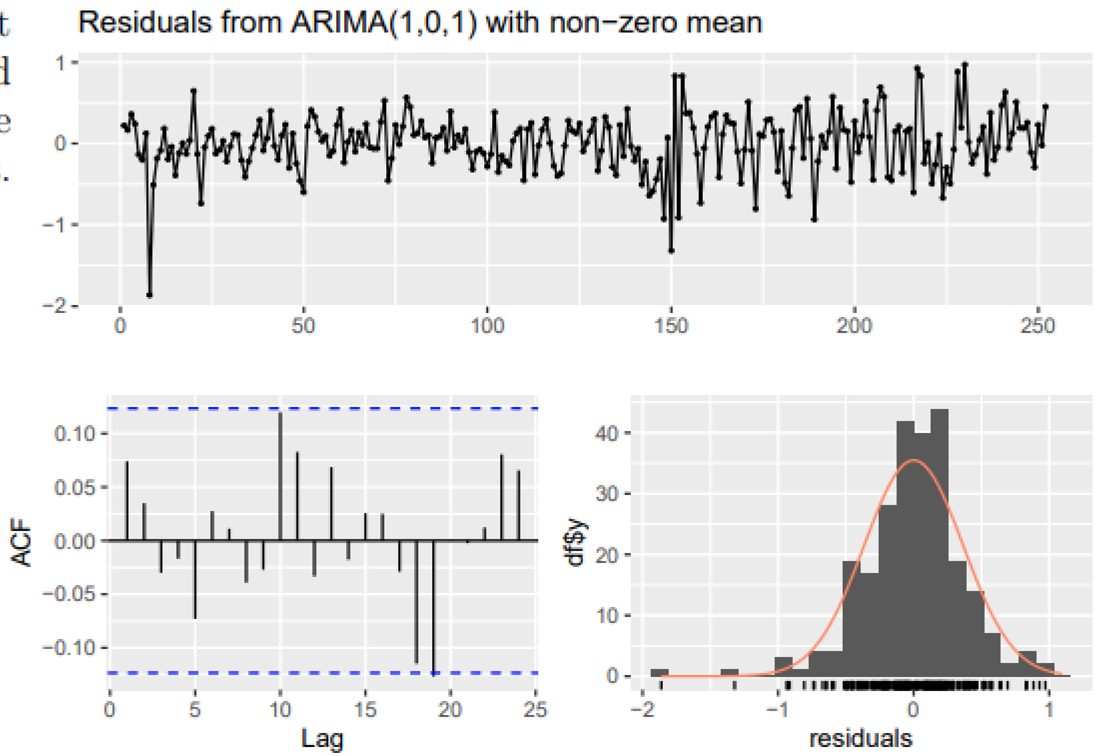
Adm · Tut 03 · Q1 · a · b · c · d · e · f · g · h · i · j

(g) Draw time series plots of the estimated residuals you obtained for the ARMA models selected in part (f). Comment on your findings. Run the Ljung-Box test (at the $\alpha = 5\%$ significance level) to test the white noise hypothesis on estimated residuals obtained from each ARMA in the set obtain in part (f) and report the test results. Use this information to identify the adequate set of specified ARMAs.

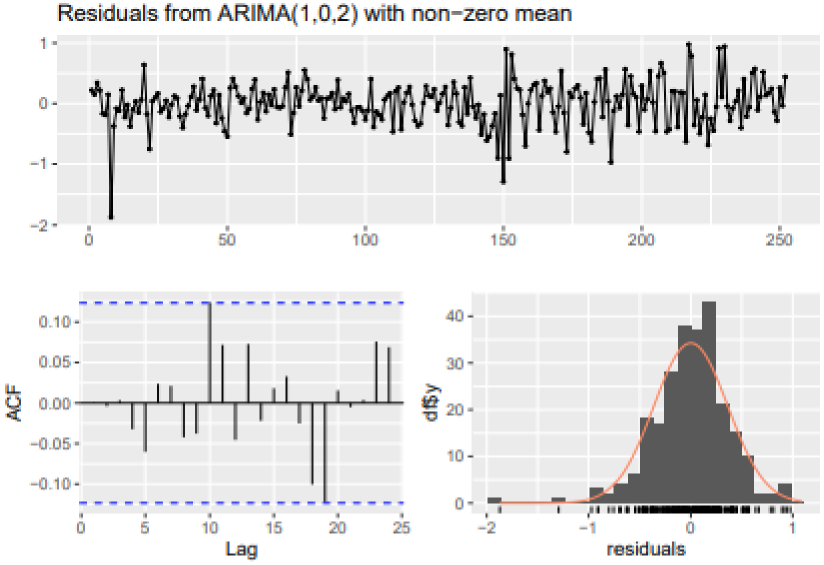
Solution The `forecast` package simplifies residuals analysis with the command `checkresiduals`. It will plot the estimated residuals series along with the SACF and a histogram. It also automatically performs the Ljung-Box test using $K = 10$ by default and the $\alpha = 5\%$ significance level.

```
for (i in 1:length(adq_set))
{
  checkresiduals(Arima(Dy[sel2011], order = adq_set[[i]]))
}
```

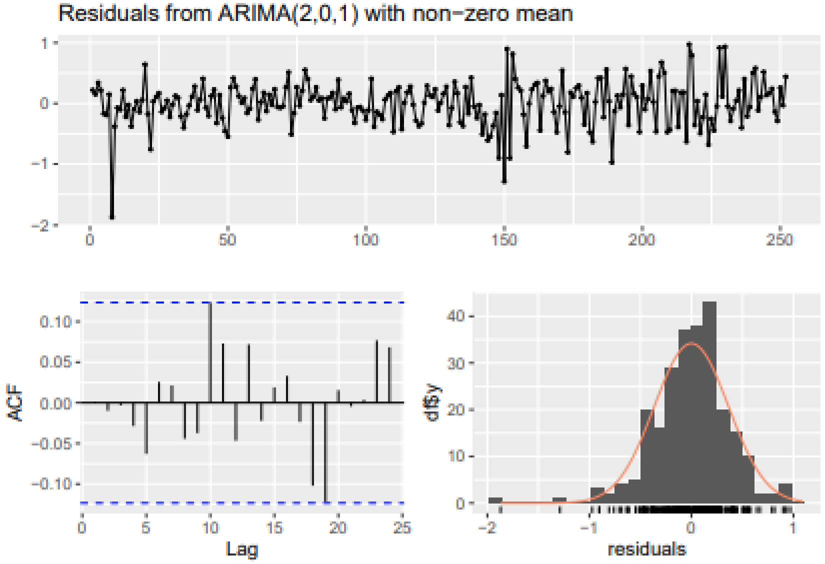
Nothing clearly stands out to suggest a *blatent* problem with correlated residuals, so we proceed with the four specifications in the set.



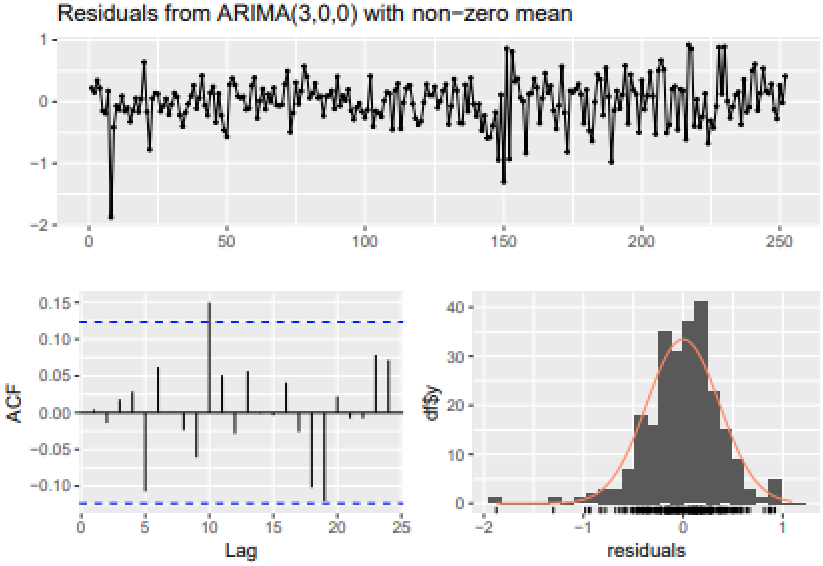

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1) with non-zero mean
## Q* = 8.0188, df = 8, p-value = 0.4316
##
## Model df: 2.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,2) with non-zero mean
## Q* = 6.42, df = 7, p-value = 0.4917
##
## Model df: 3.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,1) with non-zero mean
## Q* = 6.4833, df = 7, p-value = 0.4846
##
## Model df: 3.   Total lags used: 10
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,0) with non-zero mean
## Q* = 11.424, df = 7, p-value = 0.1212
##
## Model df: 3.   Total lags used: 10
```

(h) Forecast changes in MRK stock prices in January, 2012. For each ARMA model in the adequate set, compare your predicted price changes with real price changes in the data. Compare the forecasts you obtained as well as their "quality" across ARMA models and comment on the robustness of the generated forecasts.

Solution We forecast each model in the adequate set using the `forecast` command within a nested for loop. The option `level = c(68, 95)` instructs the command to construct two sets of predictive intervals: one with 95% coverage and another with 68% coverage.

```
hrz = sum(sel_sample) - sum(sel2011)
xticks <- c(sum(sel_sample) - 3 * hrz + c(1, 2 * hrz, 3 * hrz))
actual_Dy <- as.matrix(Dy[!sel2011])
fcst_Dy <- vector(mode = "list", length(adq_set))
for (i in 1:length(adq_set))
{
  model_p_q <- adq_set[[i]]
  fcst_Dy[[i]] <- forecast(Arima(Dy[sel2011], order = model_p_q),
                           h = hrz, level = c(68, 95))

  title_p_q <- paste("ARMA(", as.character(model_p_q[1]), ", ",
                     as.character(model_p_q[3]), ")", sep = "")

  plot(fcst_Dy[[i]], include = hrz * 2, ylab = colnames(Dy),
       main = title_p_q, xaxt = "n")
  lines(sum(sel2011) + 1:hrz, actual_Dy)
  axis(1, at = xticks, labels = dates[xticks])
}
```

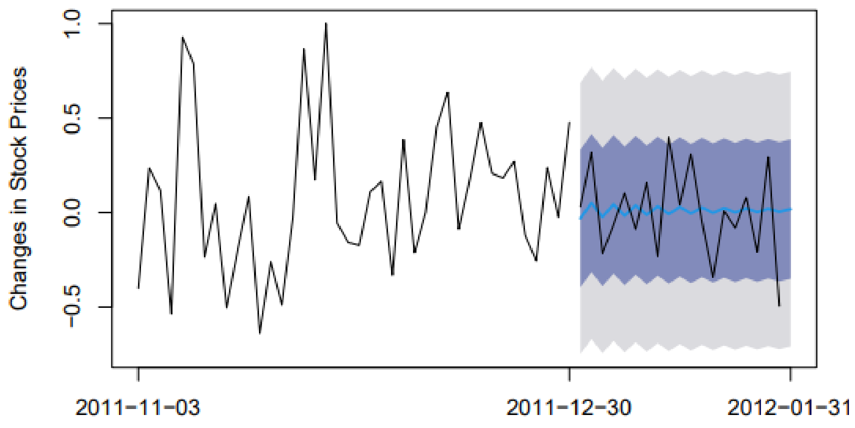
When we use the `plot` command with output from the `forecast` command, we get a nice depiction of how the data is extrapolated into the future, complete with predictive intervals to capture uncertainty. We can also add the actual outcomes in the forecast period to help us compare the forecast performance of each ARMA in the adequate set.

We first note that predictive intervals for price changes (Δy_t) appear to have a fixed width even as the forecast horizon increases (from 1 day to 20 days).

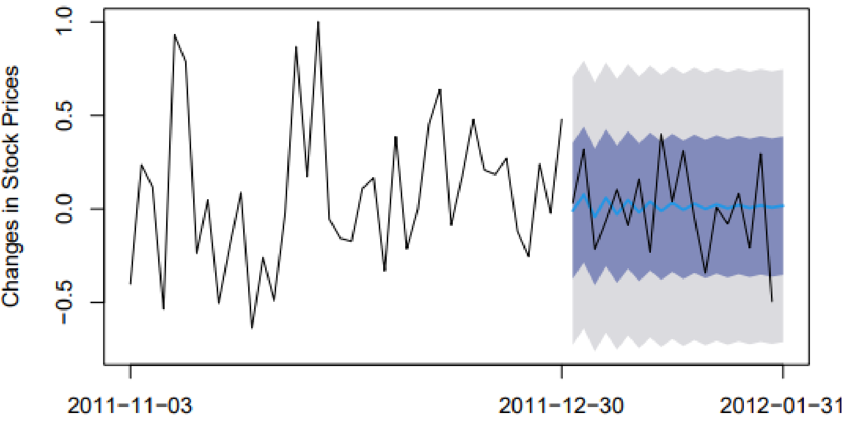
Comparing further across specifications, it is clear that all four generate very similar forecasts for January 2012. Therefore, the specification differences between them are not important for our purpose.

Alternatively, we may conclude that our forecast of price changes is *robust* to minor differences in the specification of ARMA models in that we cannot clearly distinguish between them with our diagnostic tools.

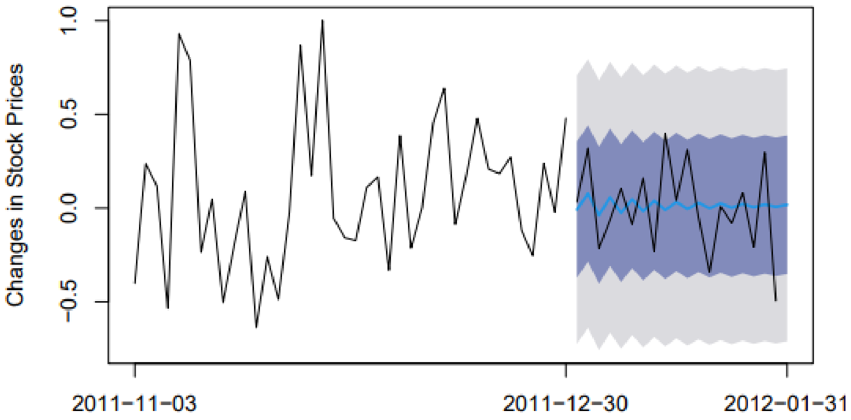
ARMA(1, 1)



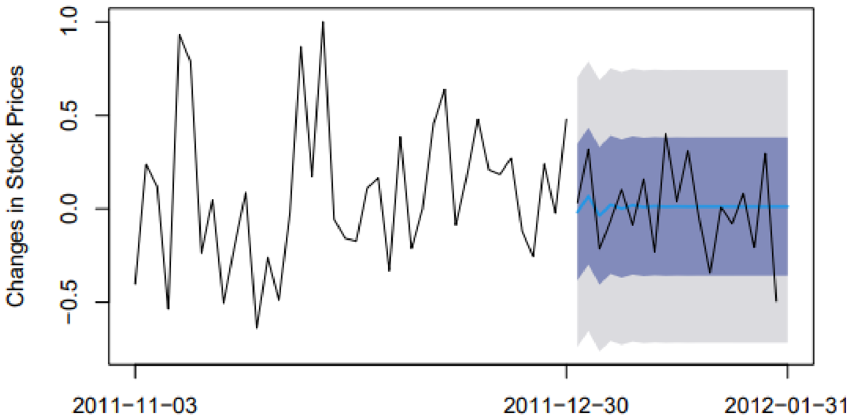
ARMA(1, 2)



ARMA(2, 1)



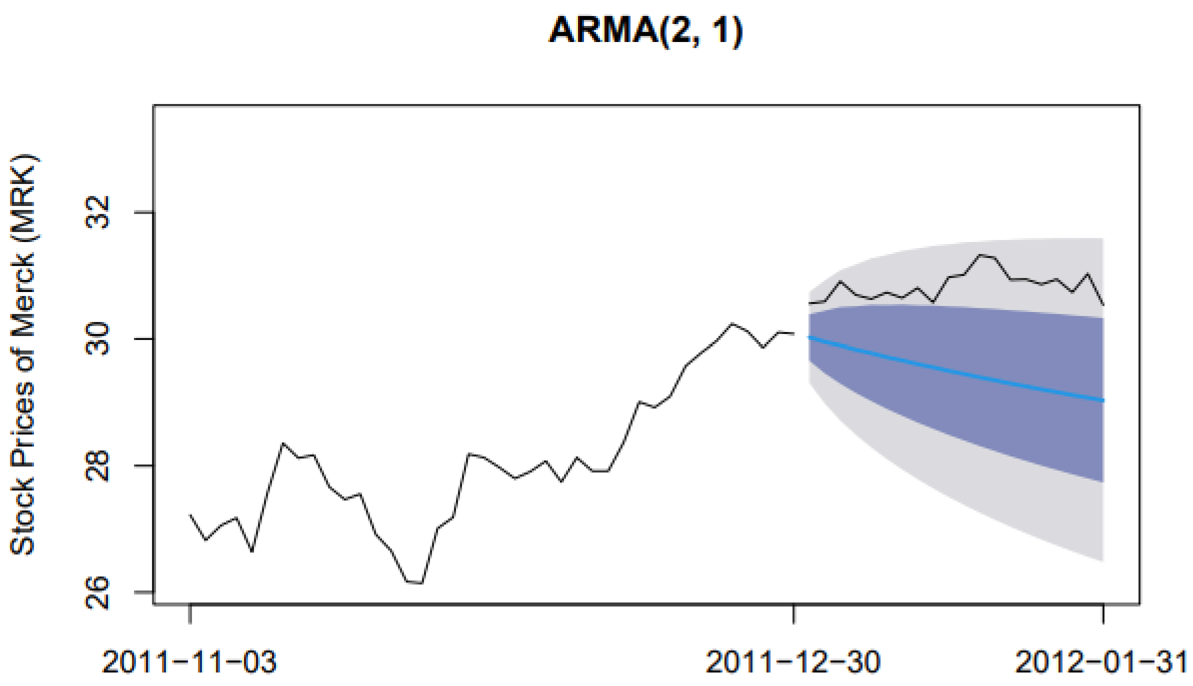
ARMA(3, 0)



(i) Forecast MRK prices y_t (levels this time, instead of changes) using an ARMA(2, 1) model only. Compare your predicted prices with real prices in the data. Compare the price forecasts obtained in this part with price forecasts obtained by transforming the forecasts in part (h). HINT: you will need convert predicted prices changes to predicted prices.

Solution To forecast prices y_t, we use the same approach but replace Dy with y throughout.

```
actual_y <- as.matrix(y[!sel2011])
fcst_y_lev = forecast(Arima(y[sel2011], order = c(2, 0, 1)),
                      h = hrz, level = c(68, 95) )

plot(fcst_y_lev, include = hrz * 2, ylab = colnames(y),
     main = "ARMA(2, 1)", xaxt = "n", ylim = c(26.1, 33.4))
lines(sum(sel2011) + 1:hrz, actual_y)
axis(1, at = xticks, labels = dates[xticks])
```



The forecasts we obtain here are for prices, where as the forecasts that were obtained for part (h) were for price changes. In order to compare them, we need to convert predicted price changes to prices, which is achieved by cumulatively suming:

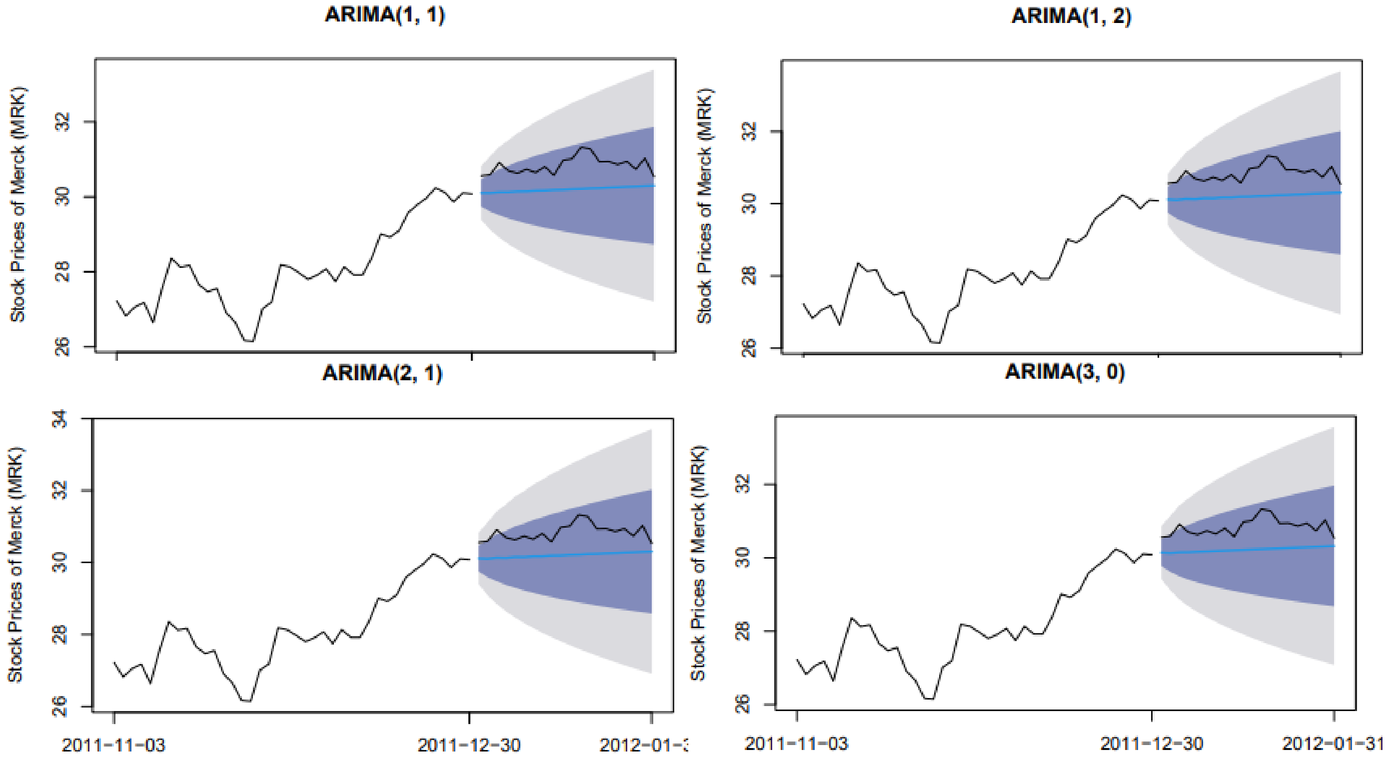
$$y_t = y_0 + \sum_{j=1}^{t} \Delta y_j,$$

where y_0 in our case is the last observation in the “pre-sample”.

However, if we did that manually, we would need to re-calculate the predictive intervals manually as well. We do not want to do this (and it does not work by simply cumulatively summing the interval limits either)!

Instead, we use another option in the `Arima` command. In particular, we pass the option `order = c(p, 1, q)` instead of `order = c(p, 0, q)`. The "1'' in the middle instructs R to *difference* y_t when estimating the ARMA parameters. However, it will automatically reverse the differencing when computing forecasts and predictive intervals! This specification is called the ARIMA(p, q). You can verify (by trying it yourself) that an ARIMA(p, q) for y_t yields the same AR and MA coefficient estimates as an ARMA(p, q) for Δy_t.

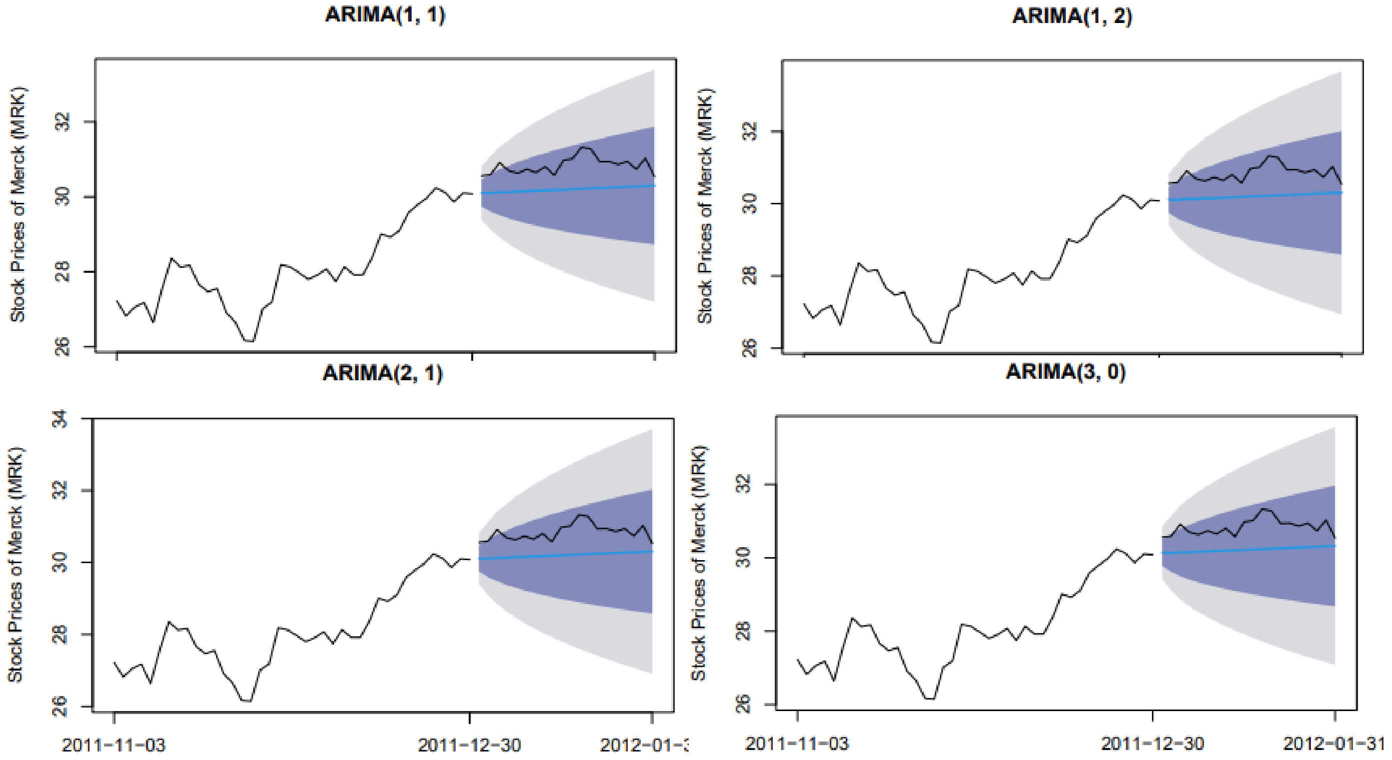
```
y0 <- mydata$Adj_Close[sum(mydata$Date < as.Date("2011-01-01") - 1)]
y_ext = as.matrix(c(y0, y[sel2011]))
fcst_y <- vector(mode = "list", length(adq_set))
for (i in 1:length(adq_set))
{
  model_p_q <- adq_set[[i]]
  model_p_q[2] = 1
  fcst_y[[i]] <- forecast(Arima(y_ext, order = model_p_q,
                          include.constant = T),
                          h = hrz, level = c(68, 95) )

  title_p_q <- paste("ARIMA(", as.character(model_p_q[1]), ", ",
                     as.character(model_p_q[3]), ")", sep = "")

  plot(fcst_y[[i]], include = hrz * 2, ylab = colnames(y),
       main = title_p_q, xaxt = "n")
  lines(1 + sum(sel2011) + 1:hrz, actual_y)
  axis(1, at = 1 + xticks, labels = dates[xticks])
}
```



A number of interesting observations emerge in comparing these forecast results. The first is that all ARIMAs in the adequate set generate very similar forecasts (including predictive intervals). The second is that the predictive intervals for y_t increase as the forecast horizon increases (they are narrower for forecasts in the beginning of the forecasting period and wider towards the end of the forecast period). How might this relate to the stationarity properties of $\{y_t\}$?

Comparing the forecasts obtained from the ARIMA(1,1) to those obtained from the ARMA(2,1), we see that both produce predictive intevals that increase as the horizon increases. However, the ARMA(2,1) predictive intervals indicate that prices should fall in January 2012. This is slightly different from what the ARIMA(1,1) produces—although the predictive intervals from the two specifications largely overlap, the ARIMA(1,1) forecast clearly puts more weight on higher prices in January.

When comparing to the actual observations in January 2012, it is easy to see that the ARIMA forecasts are better (which can be confirmed by formal metrics).



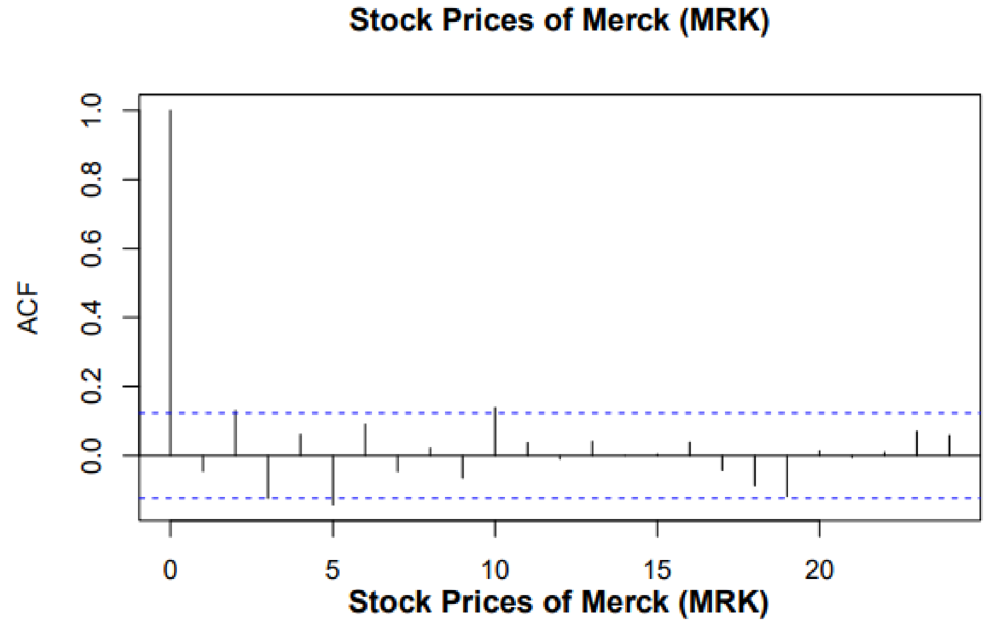
(j) OPTIONAL: Repeat parts (d)–(h) for log returns r_t. Note that here you will forecast daily returns $(y_t - y_{t-1})/y_{t-1}$ in January, 2012. Hint: Recall that $(y_t - y_{t-1})/y_{t-1} \approx r_t$.

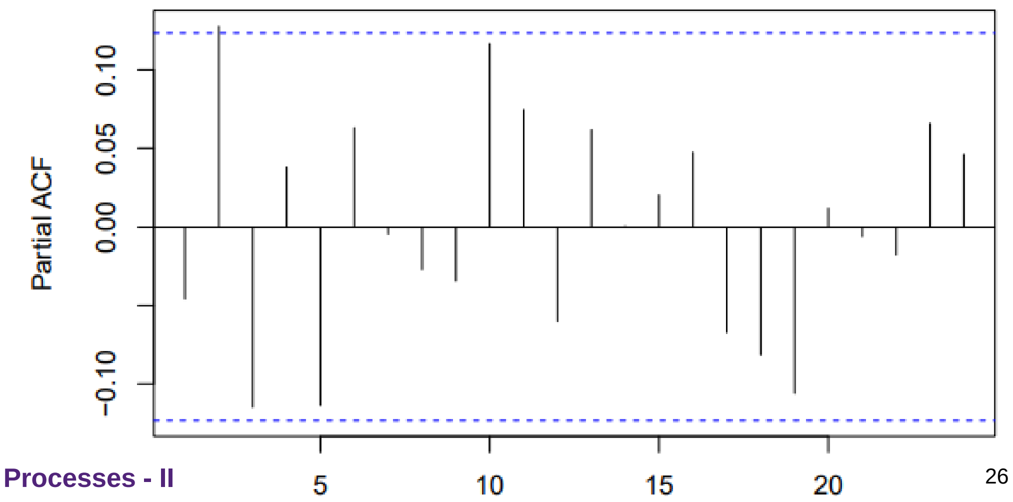
Solution The steps are nearly the same as those for working with Δy_t.

```
acf(r[sel2011], main = colnames(y))
```

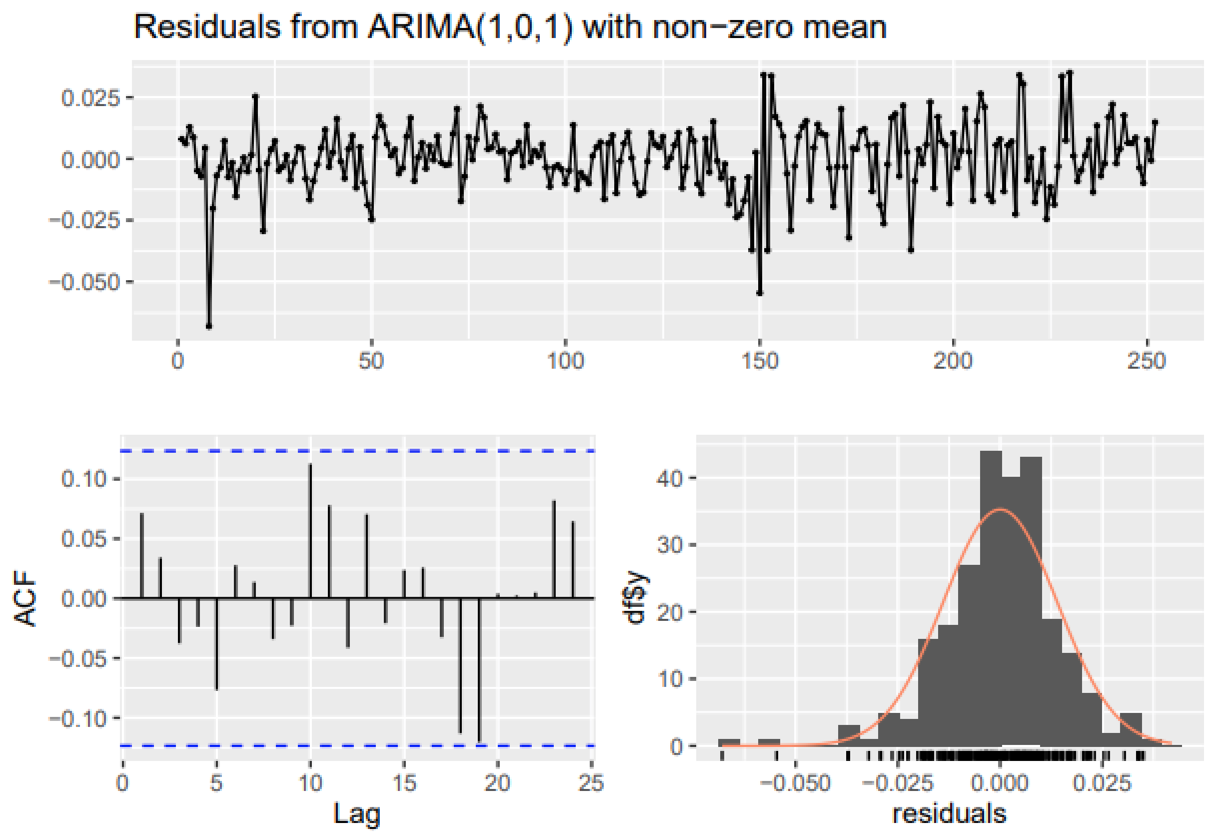
```
pacf(r[sel2011], main = colnames(y))
```

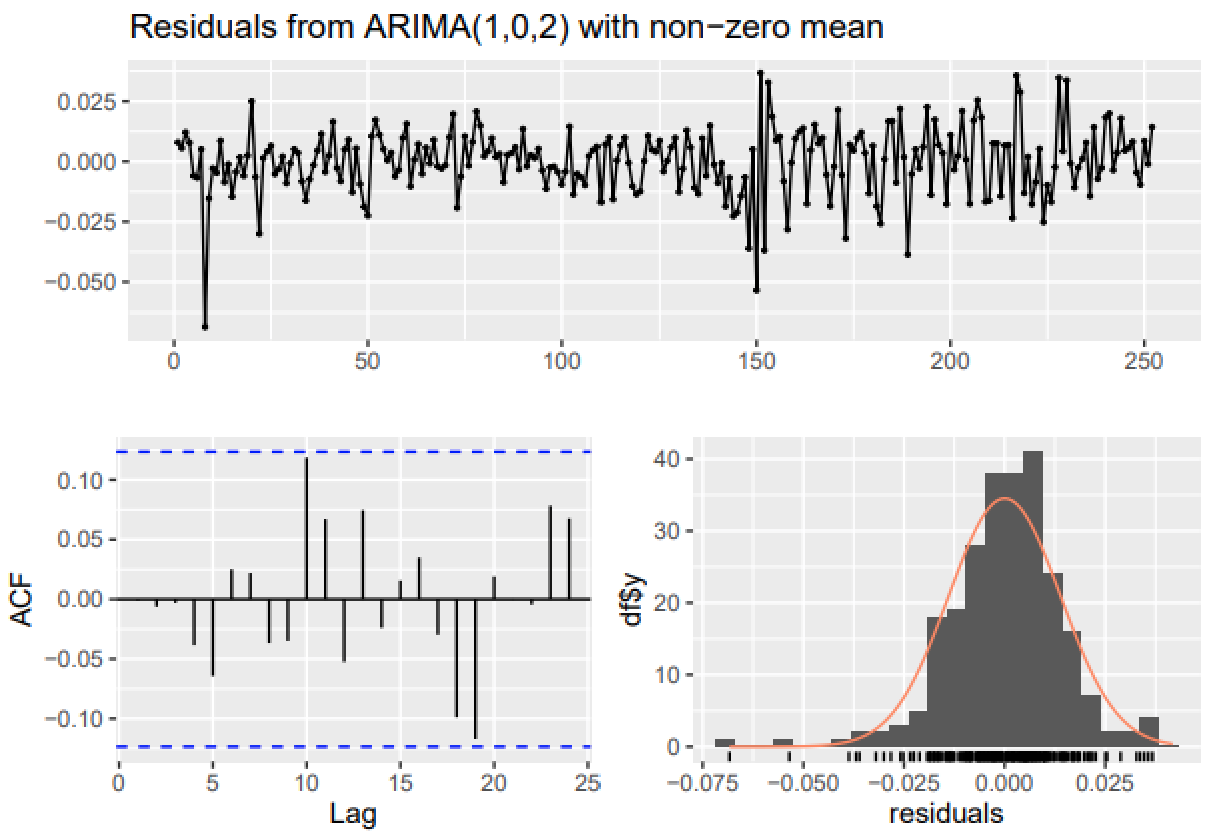
```
ic <- matrix( nrow = 25, ncol = 4 )
colnames(ic) <- c("p", "q", "aic", "bic")
for (p in 0:4)
{
  for (q in 0:4)
  {
    fit_p_q <- Arima(r, order = c(p, 0, q))
    c(p * 5 + q + 1, p, q)
    ic[p * 5 + q + 1,] = c(p, q, fit_p_q[["aic"]], fit_p_q[["bic"]])
  }
}

ic_aic <- ic[order(ic[,3], decreasing = FALSE),][1:10,]
ic_bic <- ic[order(ic[,4], decreasing = FALSE),][1:10,]

adq_set = list(c(1, 0, 1), c(1, 0, 2), c(2, 0, 1), c(3, 0, 0))

for (i in 1:length(adq_set))
{
  checkresiduals(Arima(r[sel2011], order = adq_set[[i]]))
}
```





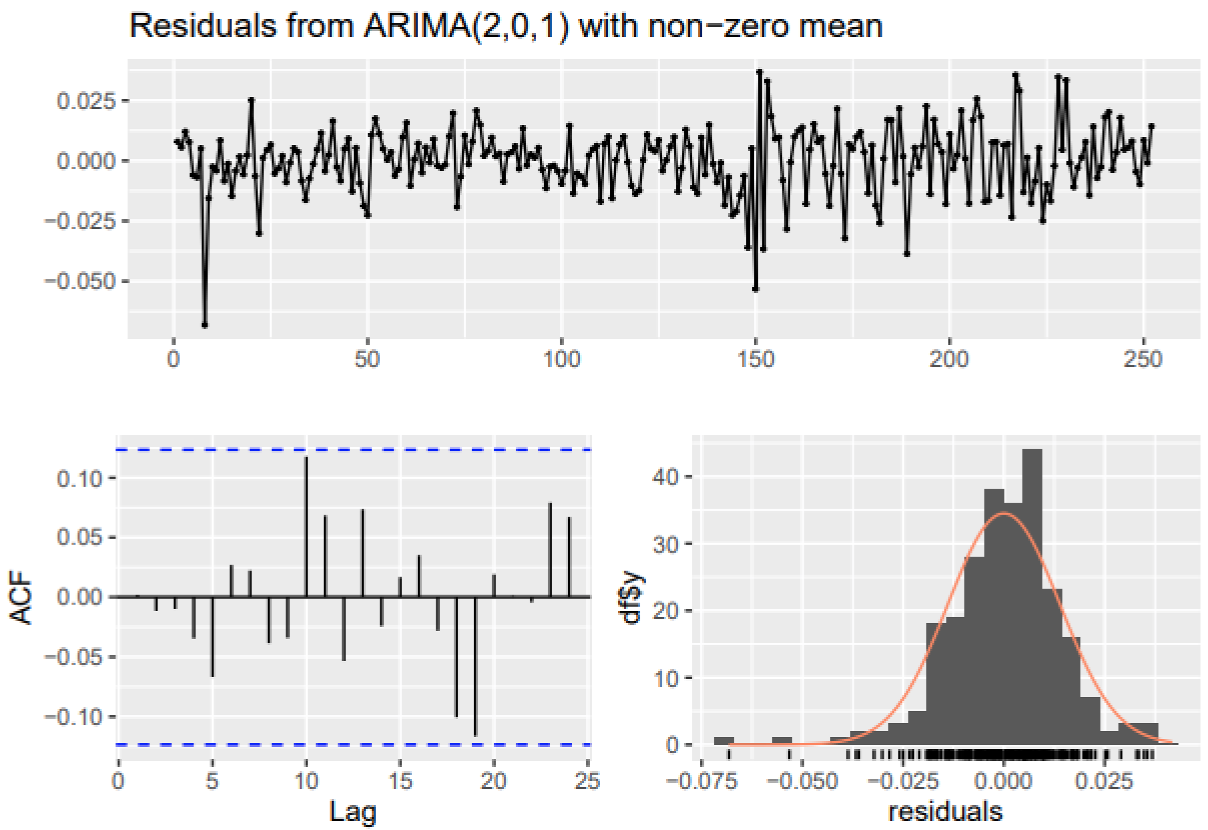
THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

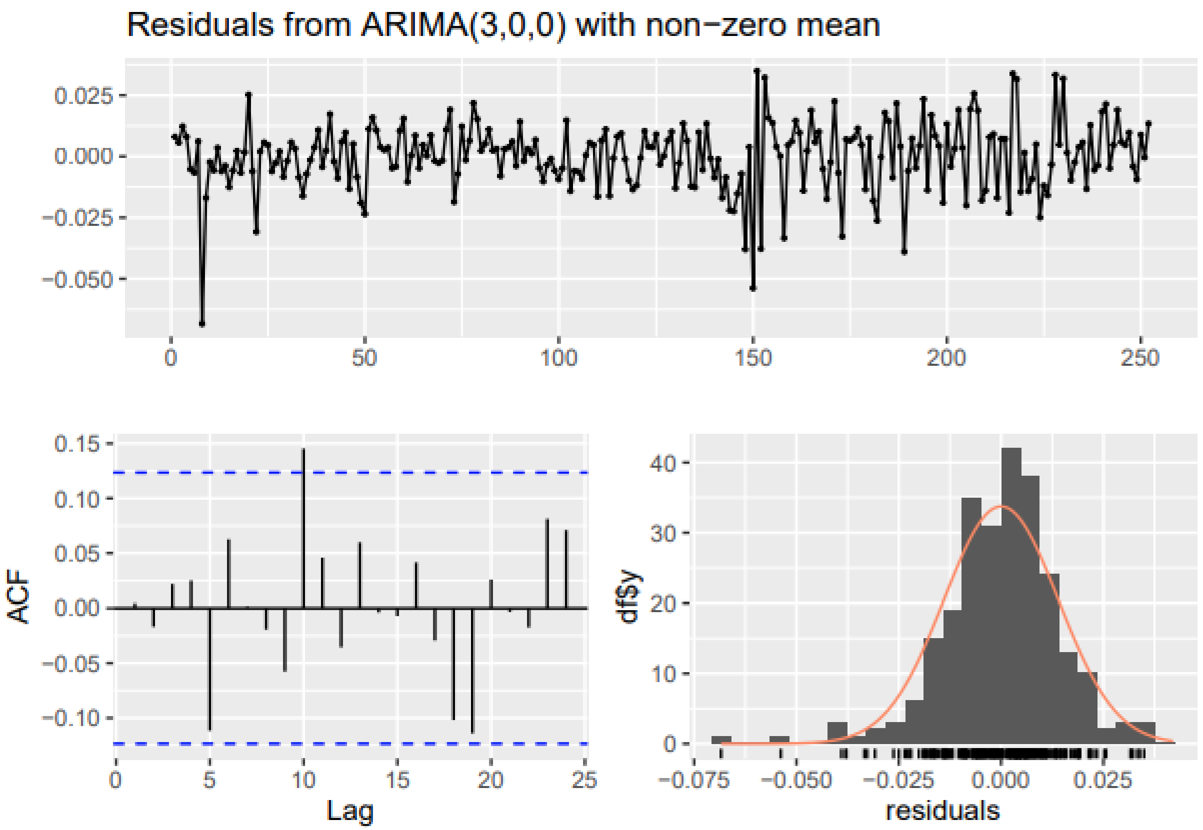





```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1) with non-zero mean
## Q* = 7.6925, df = 8, p-value = 0.4641
##
## Model df: 2.   Total lags used: 10
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,2) with non-zero mean
## Q* = 6.1518, df = 7, p-value = 0.5221
##
## Model df: 3.   Total lags used: 10
```



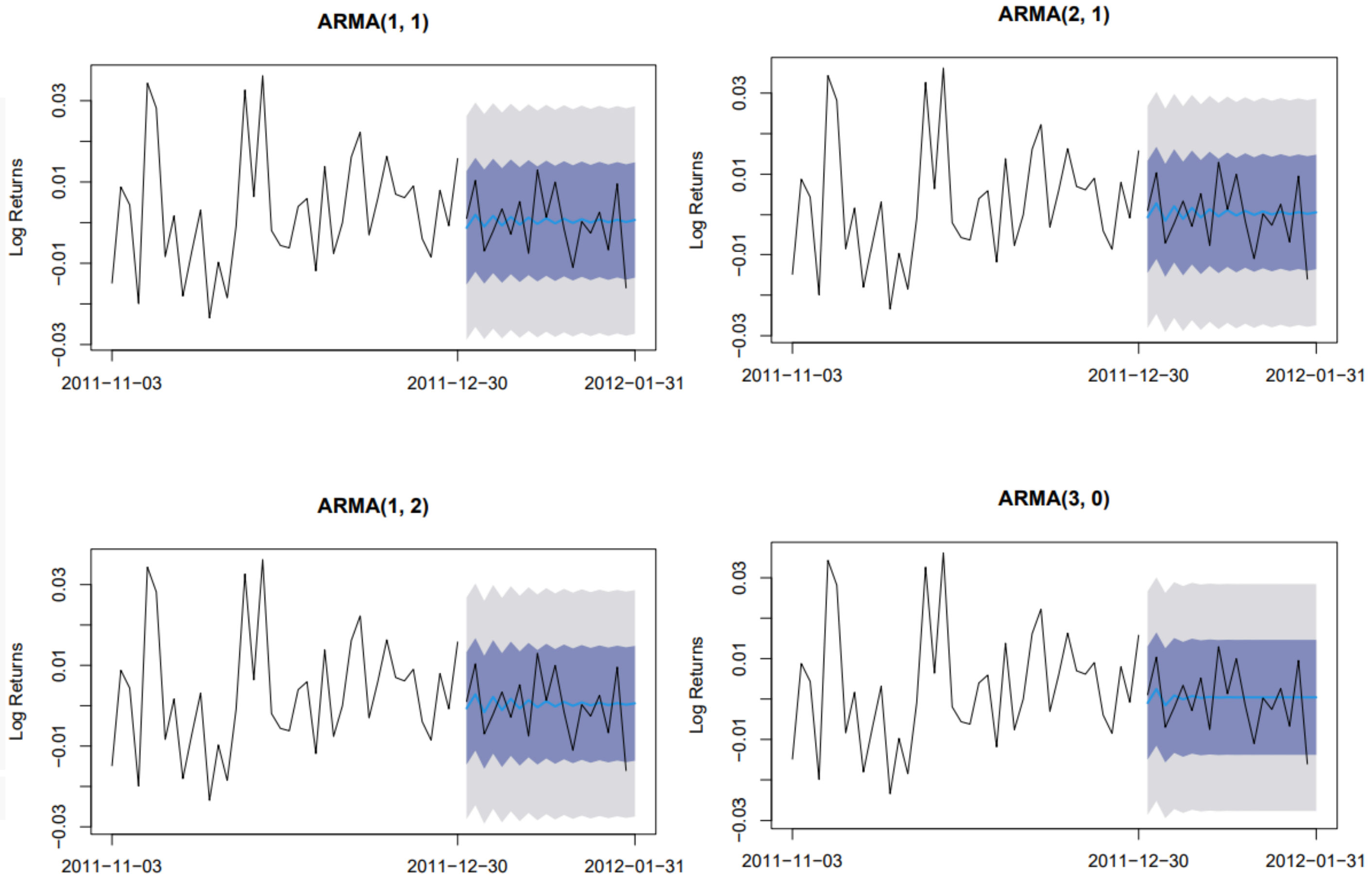



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,1) with non-zero mean
## Q* = 6.2143, df = 7, p-value = 0.515
##
## Model df: 3.   Total lags used: 10
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,0) with non-zero mean
## Q* = 11.181, df = 7, p-value = 0.1309
##
## Model df: 3.   Total lags used: 10
```

(j) OPTIONAL: Repeat parts (d)–(h) for log returns r_t. Note that here you will forecast daily returns $(y_t - y_{t-1})/y_{t-1}$ in January, 2012. Hint: Recall that $(y_t - y_{t-1})/y_{t-1} \approx r_t$.

```r
hrz <- sum(sel_sample) - sum(sel2011)
xticks <- c(sum(sel_sample) - 3 * hrz + c(1, 2 * hrz, 3 * hrz))
actual_r <- as.matrix(r[!sel2011])
fcst_r <- vector(mode = "list", length(adq_set))
for (i in 1:length(adq_set))
{
  model_p_q <- adq_set[[i]]
  fcst_r[[i]] <- forecast(Arima(r[sel2011], order = model_p_q),
                          h = hrz, level = c(68, 95))

  title_p_q <- paste("ARMA(", as.character(model_p_q[1]), ", ",
                     as.character(model_p_q[3]), ")", sep = "")

  plot(fcst_r[[i]], include = hrz * 2, ylab = colnames(r),
       main = title_p_q, xaxt = "n")
  lines(sum(sel2011) + 1:hrz, actual_r)
  axis(1, at = xticks, labels = dates[xticks])
}
```



THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Tutorial 3: Forecasting Univariate Processes - II

At the end of this tutorial you should be able to:

- estimate a set of specified ARMA models using `for` loops;
- reduce the set of models using information criteria and residuals analysis;
- generate forecasts and predictive intervals for a specified ARMA(p, q);
- compare and evaluate forecasts obtained from different ARMA models.

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA
CREATE CHANGE

Thank you

Francisco Tavares Garcia

Academic Tutor | School of Economics

tavaresgarcia.github.io

Reference

Tsay, R. (2010). Analysis of Financial Time Series, 3rd Edition, John Wiley & Sons.

CRICOS code 00025B