Forecasting of USD/CAD exchange rate using methods for time series analysis

Arina Sitnikova

July 2022

Summary

The exchange rate is one of the key indicators of relative prices in open economies. That makes forecasting exchange rates extremely relevant. Hence, many researchers tried to determine the most suitable model – from the generalized financial models to the newest deep machine learning algorithms with the highest possible prediction ability. In the most recent studies, authors used such techniques as neural networks; however, the results often turned out contradictory from currency to currency.

This project aims to predict the US dollar to the Canadian dollar exchange rate using the Bank of Canada data from January 1971 to June 2022. I built two types of forecast-specific models: ARIMA and VAR. ARIMA belongs to the class of models that explains a given series based on its past values (or lags). VAR is of great use for multiple time series (i.e. with more than one explanatory variable), and each variable is a function of not only its past lags but also past lags of all other variables. For the VAR model, I decided to include CPI (consumer price index), the interest and unemployment rates based on the previous research.

To compare the performance of each model against all other models, I used several accuracy metrics as the criterion yet considered MSE the most illustrative one due to the assumptions and data structure (e.g. absence of outliers). In addition, there were several specifications of the VAR model: I tried various combinations of the variables based on the correlation between the exchange rate and the other three variables. I also examined Granger causality, which helped determine whether one series was relevant in forecasting another. It turned out that the interest and unemployment rates could not offer as much explanatory power as CPI.

As a result, thanks to the complexity of the VAR model, it was efficient in forecasting. However, the ARIMA model also performed notably well since the exchange rate's variance was mostly explained by its shocks instead of the other variables. It also should be noted that the parsimonious specification of ARIMA did not capture any trends and performed extremely poorly; for this reason, it was reasonable to estimate a model with a more substantial number of lags.

Univariate analysis

The original monthly data of the USD/CAD exchange rate was converted to the time series format. The series is from January 1971 to June 2022. In total, there are 618 observations.

```
ex_rate <- read_excel("~/Desktop/exch_data_cad2.xlsx")
rate_ts <- ts(ex_rate, start = c(1971, 1), frequency = 12)
```

Confirming the size of the dataset:

```
length(rate_ts)
```

```
## [1] 618
```

Taking a look at the first 24 months of the data…

```
head(rate_ts,24)
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 1971 1.0118 1.0075 1.0064 1.0077 1.0087 1.0213 1.0213 1.0130 1.0130 1.0047
## 1972 1.0059 1.0047 0.9985 0.9957 0.9889 0.9795 0.9840 0.9824 0.9830 0.9827
##         Nov    Dec
## 1971 1.0039 0.9993
## 1972 0.9874 0.9968
```

And the last 24 months:

```
tail(rate_ts,24)
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2020                                           1.3499 1.3222 1.3228 1.3215
## 2021 1.2724 1.2699 1.2574 1.2496 1.2126 1.2219 1.2529 1.2603 1.2671 1.2437
## 2022 1.2616 1.2716 1.2658 1.2628 1.2852 1.2814                            
##         Nov    Dec
## 2020 1.3068 1.2808
## 2021 1.2570 1.2794
## 2022              
```

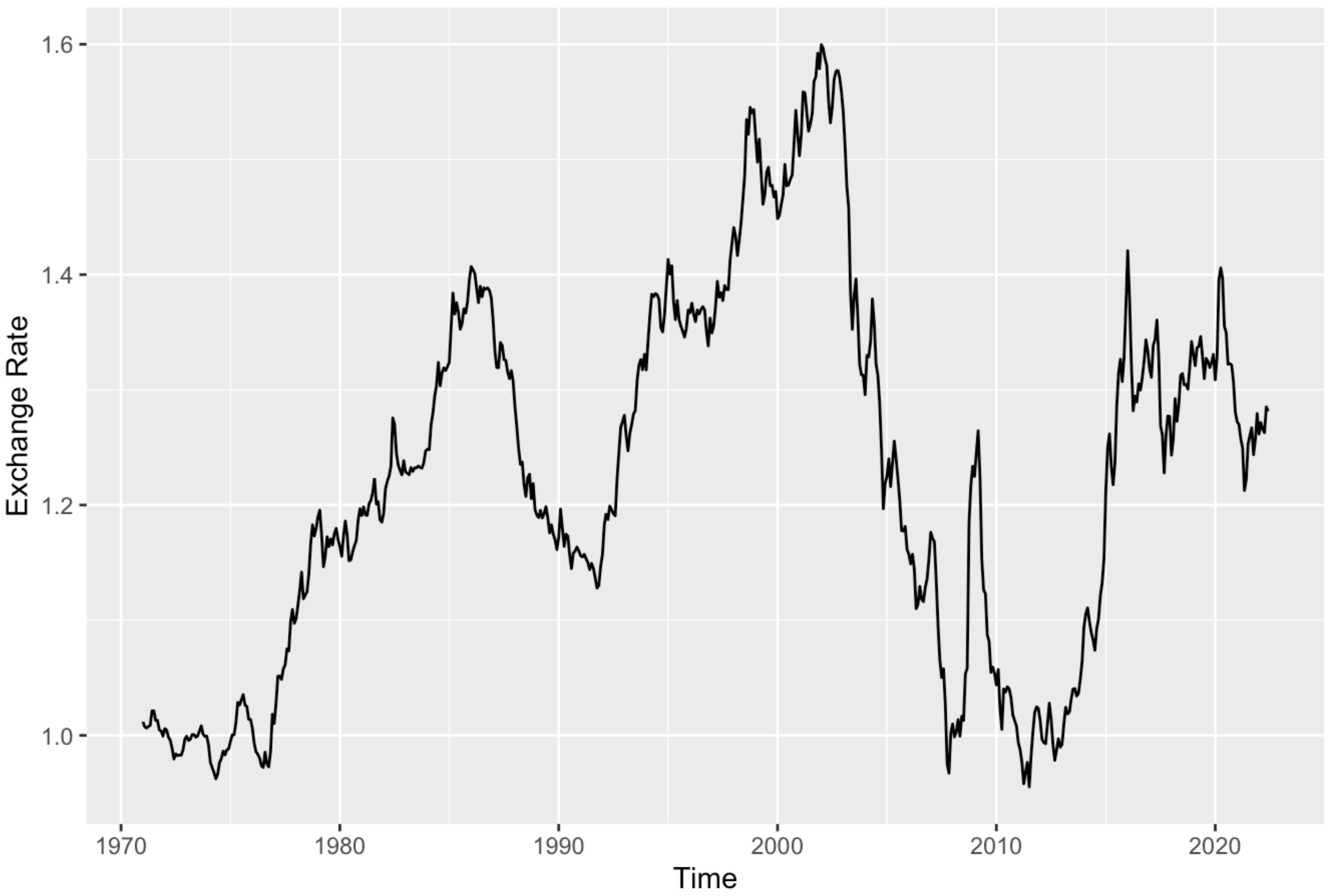
For training purposes, creating a separate dataset containing all data up to T-1 period (June 2021):

```
train = window(rate_ts, end = c(2021,06))
```

Plotting the series:

```
autoplot(rate_ts) + ggtitle("Figure 1. Exchange Rate, USD to CAD") + ylab('Exchange R
ate')
```

Figure 1. Exchange Rate, USD to CAD



We can assume that there are several structural breaks. A structural break occurs when we see a sudden change in the time series, which might be the case here. Structural breaks can lead to immense forecasting errors and unreliability of the model in general. If structural breaks are present, we should take this into account to avoid errors and make our forecast more precise. The potential structural breaks occured as per below:

```
library(strucchange)
bp_ts <- breakpoints(rate_ts ~ 1)
#summary(bp_ts)
break_point <- breakpoints(rate_ts ~ 1)
break_date <- breakdates(break_point)
f1 <- c(rep(0, 91), rep(1, 277 - 91), rep(0, length(train) - 277))
f2 <- c(rep(0, 277), rep(1, 405 - 277), rep(0, length(train) - 405))
f3 <- c(rep(0, 405), rep(1, 500 - 405), rep(0, length(train) - 500))
f4 <- c(rep(0, 500), rep(1, length(train) - 500))
matrr <- matrix(c(f1,f2,f3,f4), ncol = 4)
cat("Years with possible structural breaks:", round(breakdates(break_point), 0))
```

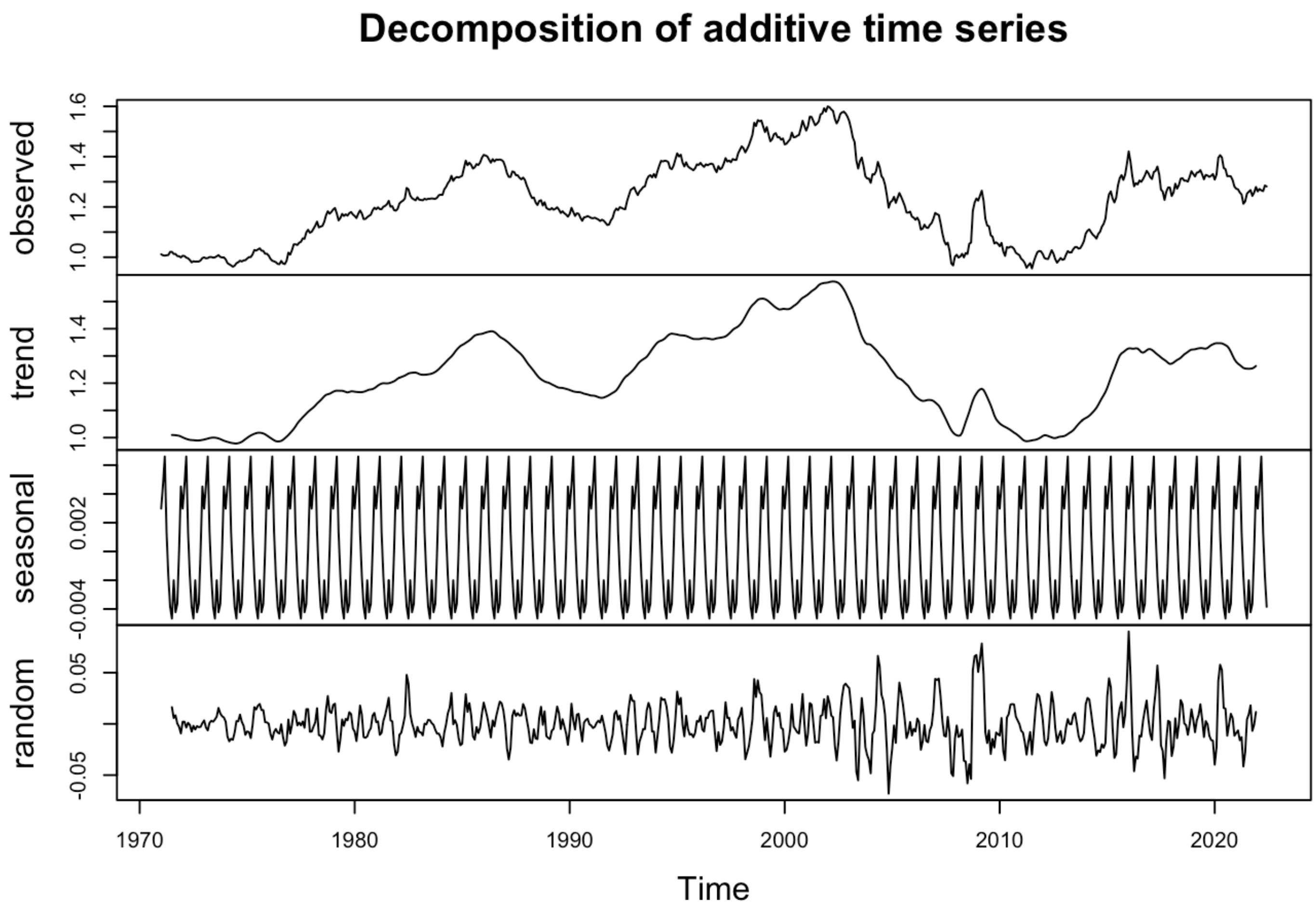
```
## Years with possible structural breaks: 1979 1994 2005 2015
```

We will keep in mind these years. Also, we create a matrix which reflects the four structural breaks to take them into account in the further analysis.

Furthermore, we should understand whether the series has a trend or a seasonal pattern. If it does, it is necessary to take it into account while running ARIMA model to avoid any inconsistency in the estimates.

Thus, the series was decomposed:

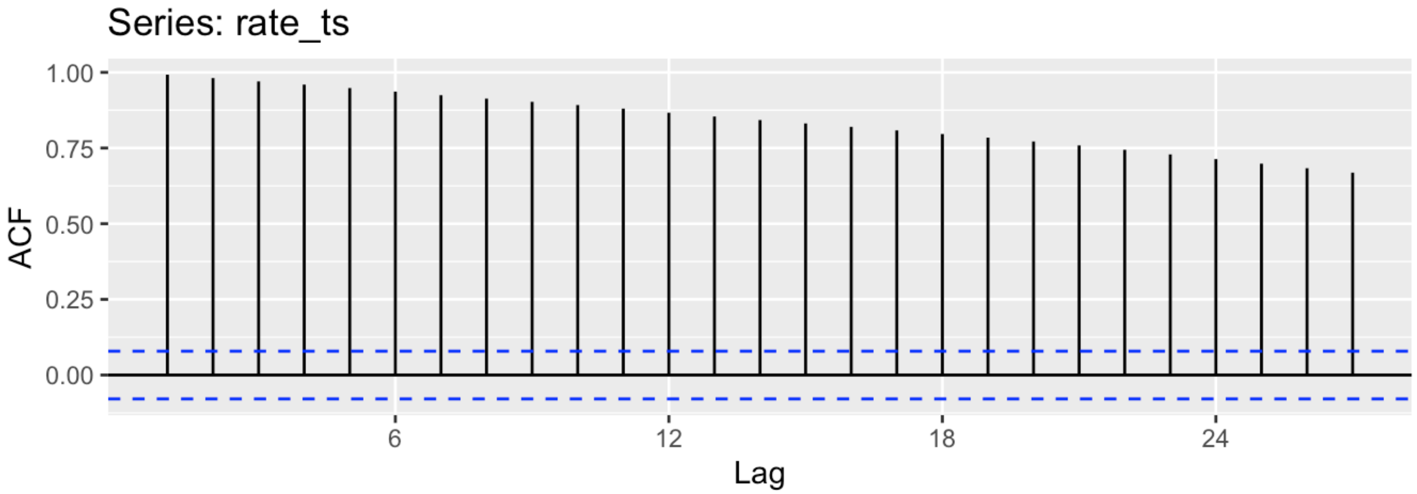
```
plot(decompose(rate_ts))
```

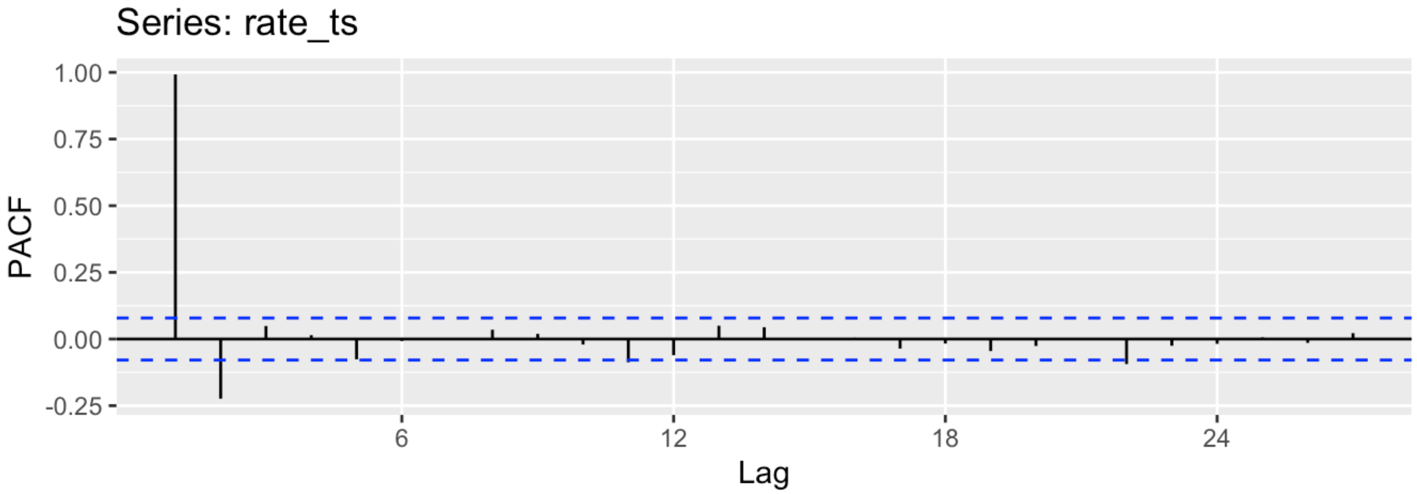


As seen, the series does not have a determined upward nor downward trend; the only thing we can see there is an uneven pattern and fluctuations. Apart from this, we can conclude that even if there is a seasonal component, it is totally negligible: the scale of the seasonality graph is too small (from -.004 to .002). To sum up, it is highly likely that we've got neither trend nor seasonal pattern.

After this, we check whether the original series is stationary or non-stationary. There are two plots, ACF and PACF, presented below:

```
grid.arrange(ggAcf(rate_ts), ggPacf(rate_ts), nrow = 2)
```





For stationary series, it is expected to see autocorrelations that would decay to zero at higher lags. Clearly, it is not the case here: autocorrelations only slightly decrease over lags. Therefore, we can assume nonstationarity. To make sure, we will perform two tests on stationarity: ADF test (which takes H0 as nonstationarity) and KPSS test (in which H0 stands for stationarity).

The result of the ADF test:

```
library(tseries)
adf.test(rate_ts, k = 10)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  rate_ts
## Dickey-Fuller = -2.0956, Lag order = 10, p-value = 0.5379
## alternative hypothesis: stationary
```

P-value is even greater than 0.1, then we can conclude that the null hypothesis of nonstationarity cannot be rejected at any appropriate level of confidence.

The result of the KPSS test:

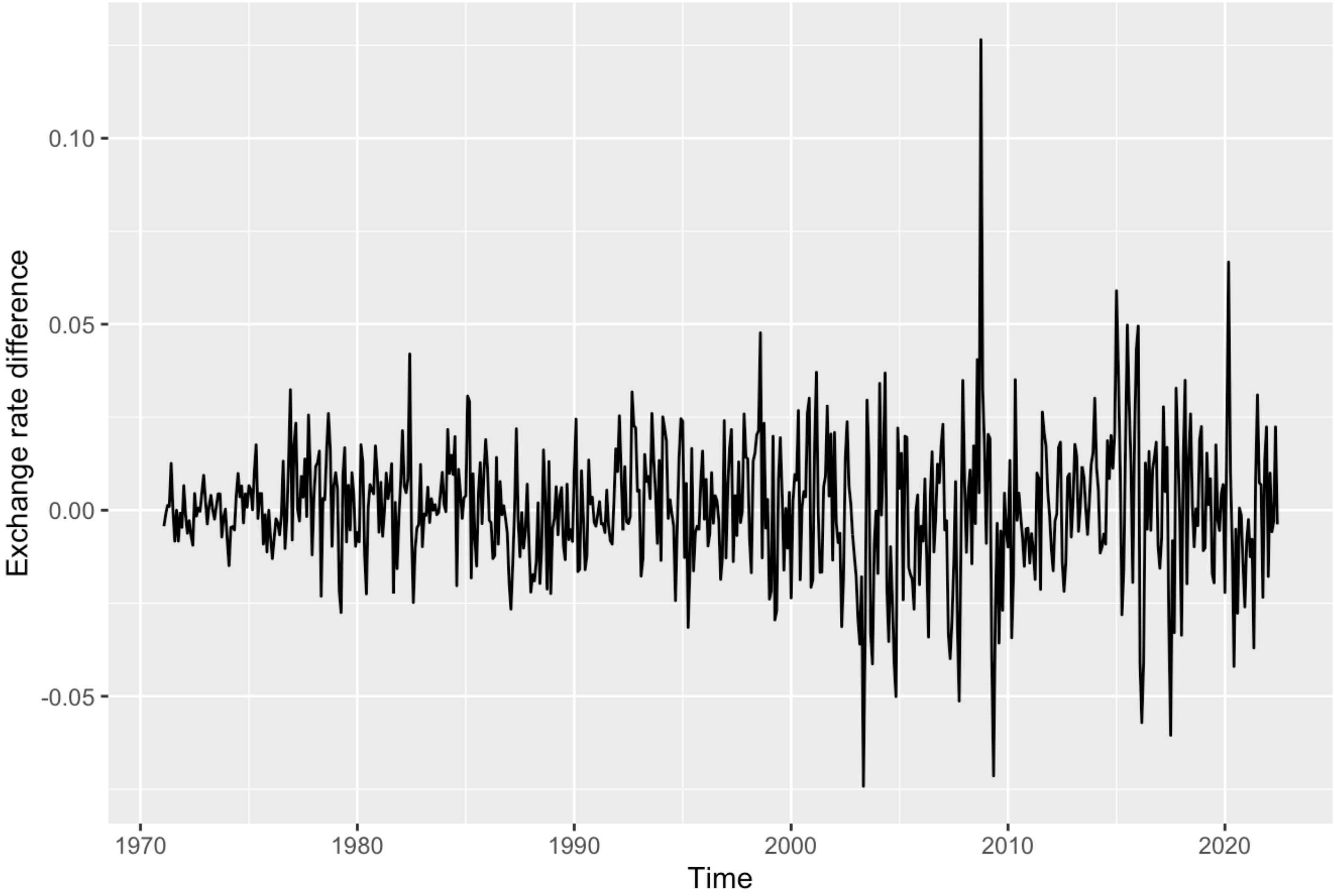
```
kpss.test(rate_ts, null = "Level")
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  rate_ts
## KPSS Level = 1.381, Truncation lag parameter = 6, p-value = 0.01
```

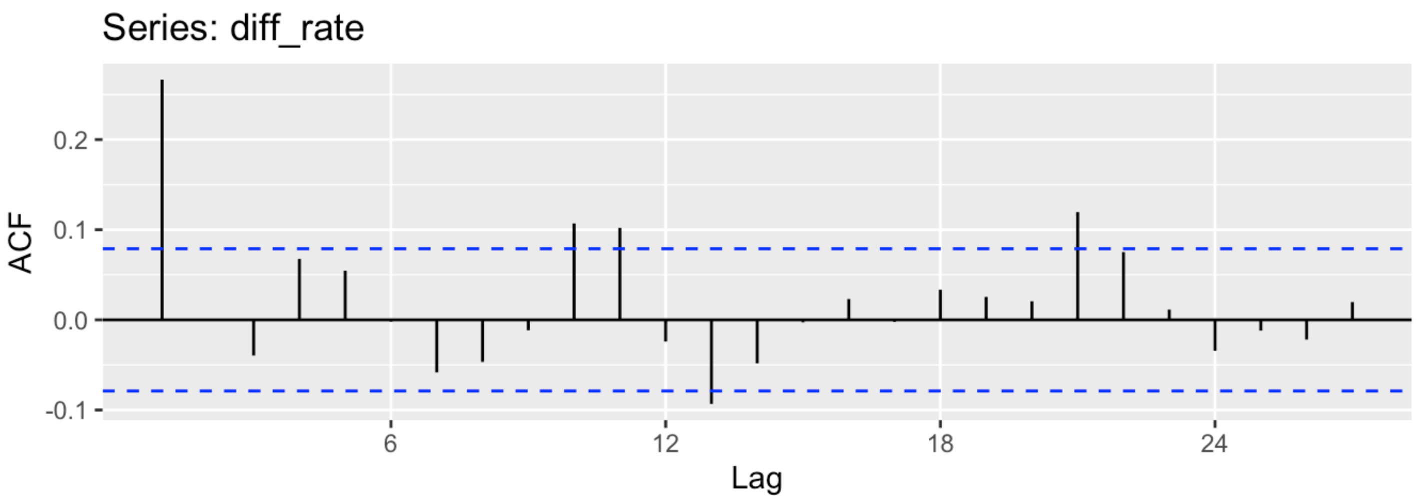
P-value is 0.01; therefore we can reject the null hypothesis of stationarity at the 95% level of confidence, which confirms the results of the ADF test. Thus, the original series is not stationary. To get rid of non-stationarity, the first difference was taken. The plot and the ACF/PACF graphs are below:

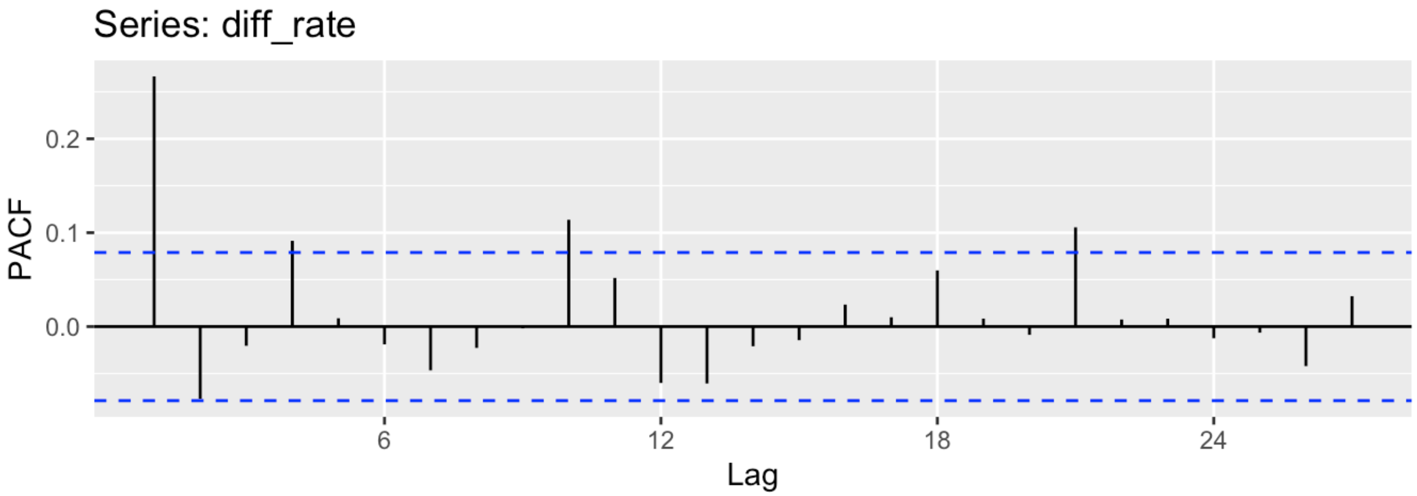
```
diff_rate <- diff(rate_ts)
autoplot(diff_rate) + ggtitle("Figure 4. Change in Exchange Rate, USD to CAD") + ylab
("Exchange rate difference")
```

Figure 4. Change in Exchange Rate, USD to CAD



```
grid.arrange(ggAcf(diff_rate), ggPacf(diff_rate), nrow = 2)
```





Now the data looks stationary. Getting back to seasonality, the absence of regular spikes at specific lags (for example, spikes at the 4th, 8th, 12th lag and so on means quarterly seasonality) supports the assumption that there is no significant seasonal components.

Again, we take a look at the following tests to confirm that the differenced series is stationary:

```
adf.test(diff_rate, k = 10)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_rate
## Dickey-Fuller = -6.3582, Lag order = 10, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff_rate, null = "Level", lshort = TRUE)
```

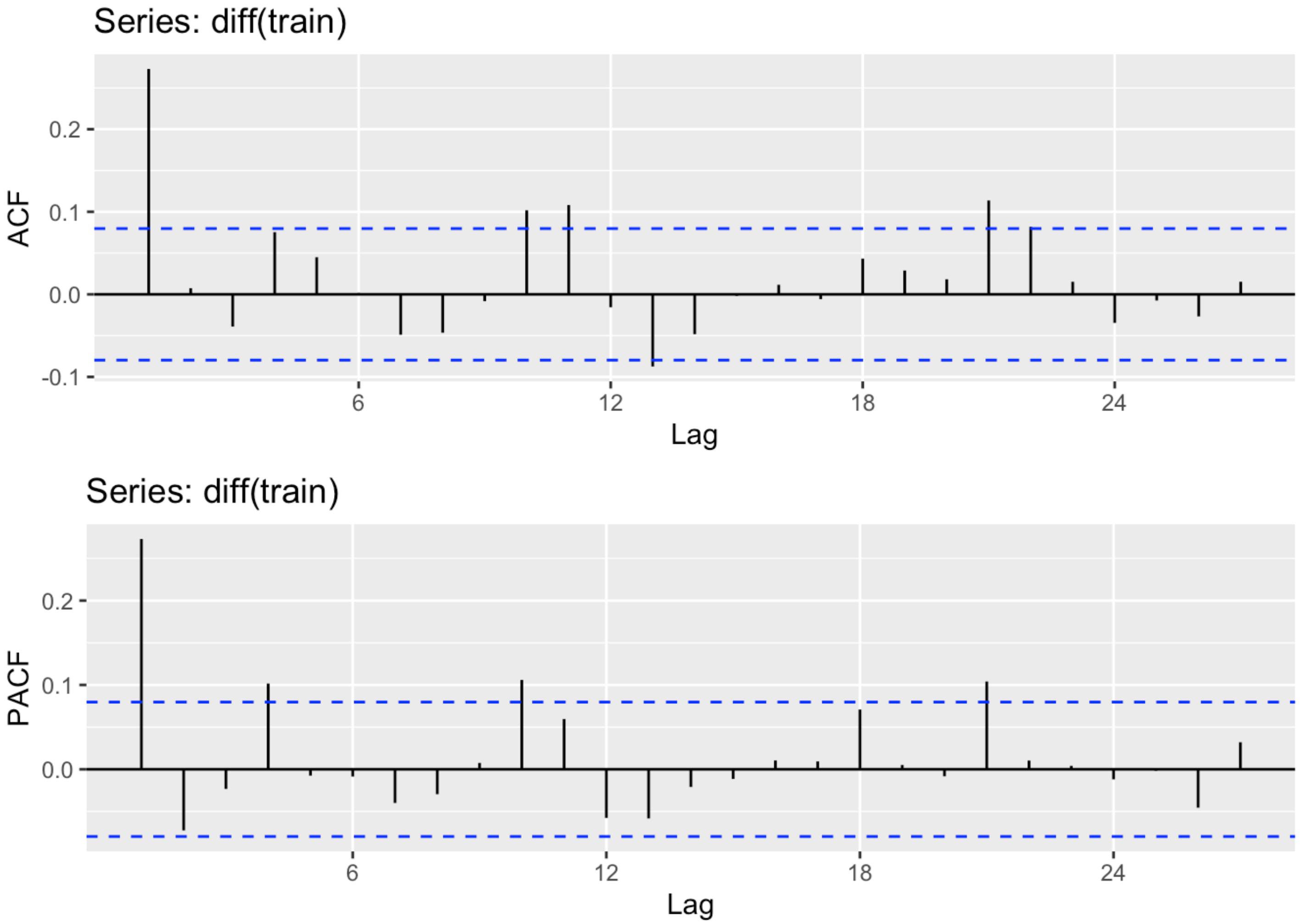
```
##
##  KPSS Test for Level Stationarity
##
## data:  diff_rate
## KPSS Level = 0.1065, Truncation lag parameter = 6, p-value = 0.1
```

So, the ADF test shows that H0 is rejected at 95% (p-value = 0.01), and the KPSS test states that H0 is not rejected at 95% level of confidence, which means that now the series is stationary.

To find out how accurate our forecast will be, we'll forecast the last available year at first. In case of time series, there is no point in predicting more than 12-24 months as the forecast will be highly inaccurate.

First of all, it is necessary to estimate ARIMA model. One of the possible ways to choose the best value for p and q lags it to take a look at ACF and PACF graphs of the train data:

```
grid.arrange(ggAcf(diff(train)), ggPacf(diff(train)), nrow = 2)
```



So, it can be assumed that the original data (not differenced) may be analyzed as in ARIMA(4,1,1). The number of p is determined by the PACF graph; the ACF plot determines the number of q. The d was already proven to be equal to 1 as the first difference appeared to be stationary. However, the best way is to iterate through all the possible models and choose the best one according to the AIC criteria. As the Box-Jenkins approach recommends that the model be parsimonious, we set the maximum number of lags at five. Apart from this, we include the matrix consisting of four vectors which represent four structural breaks that we discussed above. The results of the Arima estimation are presented below:

```
matr = matrix(0, nrow = 6, ncol = 6)
for (i in 0:5){
  for (j in 0:5) {
    model = Arima(train, order=c(i,1,j), include.constant = TRUE, xreg = matrr)
    matr[i+1,j+1] = model$aic
  }
}

matr <- data.frame(matr)
colnames(matr) <- c('q=0', 'q=1', 'q=2', 'q=3', 'q=4', 'q=5')
rownames(matr) <- c('p=0', 'p=1', 'p=2', 'p=3', 'p=4', 'p=5')
matr %>% kable() %>% kable_styling()
```

	q=0	q=1	q=2	q=3	q=4	q=5
p=0	-3141.516	-3183.747	-3182.693	-3184.525	-3184.176	-3183.397
p=1	-3182.043	-3182.356	-3184.142	-3183.350	-3182.963	-3181.410
p=2	-3183.197	-3181.279	-3183.893	-3182.033	-3193.533	-3192.540
p=3	-3181.507	-3182.414	-3188.142	-3180.822	-3193.039	-3191.132
p=4	-3185.643	-3183.658	-3181.664	-3191.187	-3191.134	-3189.153
p=5	-3183.660	-3181.657	-3179.659	-3191.062	-3189.132	-3187.152

```
sprintf("The minimum AIC is: %f", min(matr))
```

```
## [1] "The minimum AIC is: -3193.532511"
```

The minimum AIC is found at p = 2, q = 4. After the estimation, we can verify that all lags are significant at least at 95% level. However, neither the drift nor the structural breaks vectors are significant, which means we can exclude them:

```
coeftest(Arima(train, order = c(2,1,4), include.constant = TRUE, xreg = matrr))
```

```
##
## z test of coefficients:
##
##          Estimate  Std. Error  z value  Pr(>|z|)
## ar1    0.88031068  0.03786164  23.2507 < 2.2e-16 ***
## ar2   -0.93043526  0.03086043 -30.1498 < 2.2e-16 ***
## ma1   -0.60454194  0.05614563 -10.7674 < 2.2e-16 ***
## ma2    0.72742236  0.05238118  13.8871 < 2.2e-16 ***
## ma3    0.21160974  0.04630604   4.5698 4.882e-06 ***
## ma4    0.10899091  0.04208679   2.5897  0.009607 **
## drift  0.00039315  0.00094808   0.4147  0.678373
## xreg1  0.00493297  0.01622611   0.3040  0.761117
## xreg2  0.02195546  0.02309774   0.9505  0.341835
## xreg3  0.00050330  0.02854040   0.0176  0.985930
## xreg4 -0.01908705  0.03315679  -0.5757  0.564845
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit2 <- Arima(train, order = c(2,1,4))
```

Thus, we estimate the ARIMA(2,1,4) model. Before building a forecast, it is important to examine the residuals:

```
Box.test(residuals(fit2), lag = 20, type = 'Ljung')
```

```
##
##  Box-Ljung test
##
## data:  residuals(fit2)
## X-squared = 13.004, df = 20, p-value = 0.8772
```

```
jarque.bera.test(residuals(fit2))
```

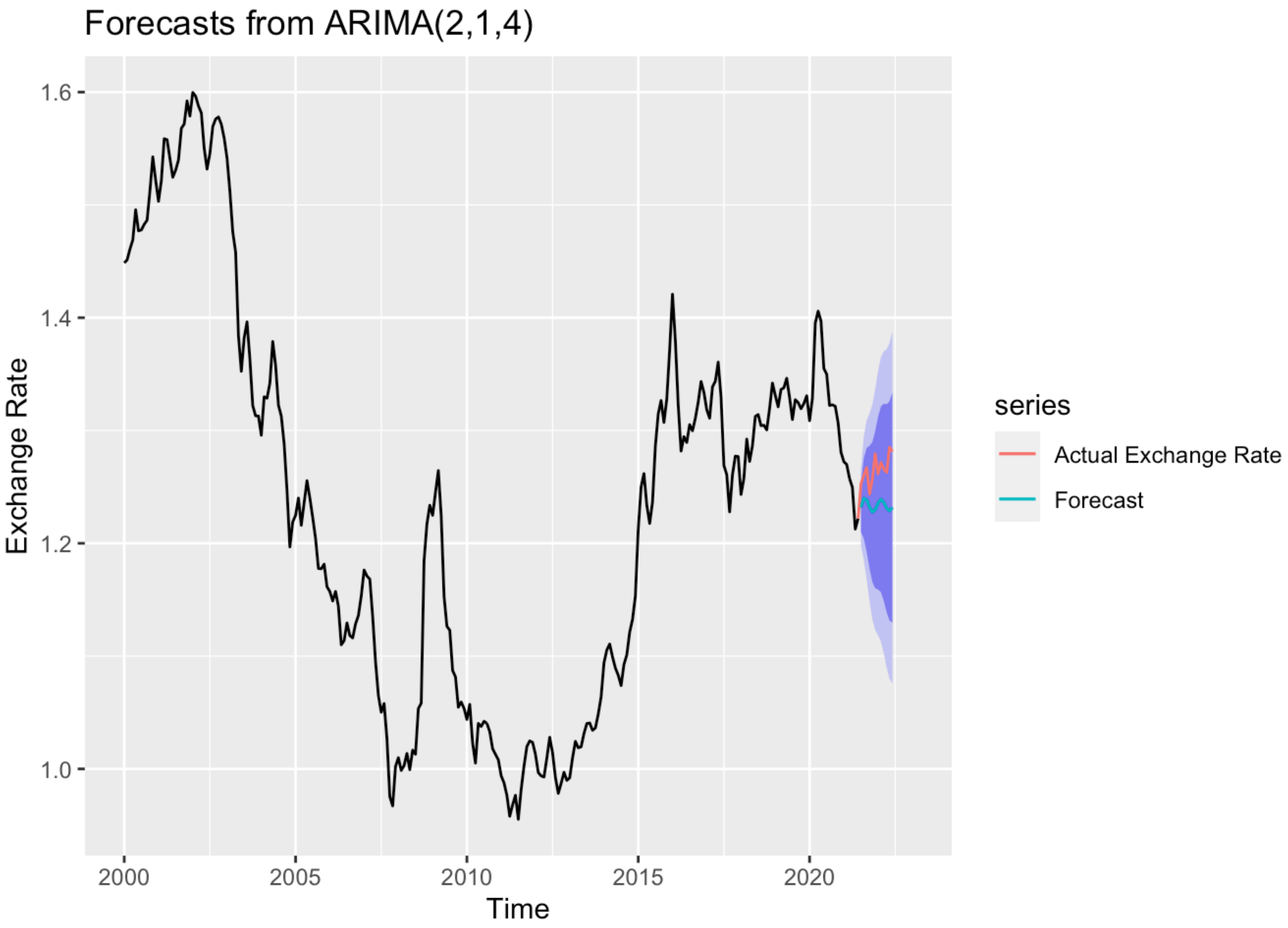
```
##
##  Jarque Bera Test
##
## data:  residuals(fit2)
## X-squared = 575.57, df = 2, p-value < 2.2e-16
```

First, the Box-Ljung autocorrelation test was performed. The null hypothesis is that there is no autocorrelation, and since p-value is greater than 0.05, it cannot be rejected. However, having conducted the Jarque Bera Test for normality, it turned out that residuals are not normal (H0 was rejected). Nevertheless, forecasting can still be done in that case.

```
fore2 <- forecast(fit2, h = 12)
```

Finally, the forecast is illustrated below. For convenience, not the whole series is shown - only from the year of 2000.

```
actual_data <- window(rate_ts, start = c(2021, 06), end = c(2022,06))
autoplot(fore2, xlim = c(2000,2023)) + autolayer(actual_data, series = "Actual Exchan
ge Rate") + autolayer(fore2$mean, series = "Forecast") + ylab('Exchange Rate')
```



From what we see, the forecast is not perfectly accurate, but the errors are not critical - the predicted values are relatively close to the real ones. Moreover, the actual values are within the 95% confidence interval.

There is also automatic auto.arima function which is useful for us since it can either support or contradict our choice of lags. The results for auto.arima are presented below:

```
fit0 <- auto.arima(train, d = 1, seasonal = TRUE)
fit0
```

```
## Series: train
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       0.2765
## s.e.  0.0366
##
## sigma^2 = 0.0002993:  log likelihood = 1596.54
## AIC=-3189.08   AICc=-3189.06   BIC=-3180.27
```

As seen, the model offered by computer differs from what we chose manually. It is more parsimonious though AIC is greater. However, the MA coefficient is still significant as we can say from the test of coefficients:

```
coeftest(Arima(train, order = c(0,1,1)))
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ma1 0.276511   0.036615  7.5518 4.294e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Then the autocorrelation and normality tests are performed:

```
Box.test(residuals(fit0), lag = 10, type = 'Ljung')
```

```
##
##  Box-Ljung test
##
## data:  residuals(fit0)
## X-squared = 14.608, df = 10, p-value = 0.147
```

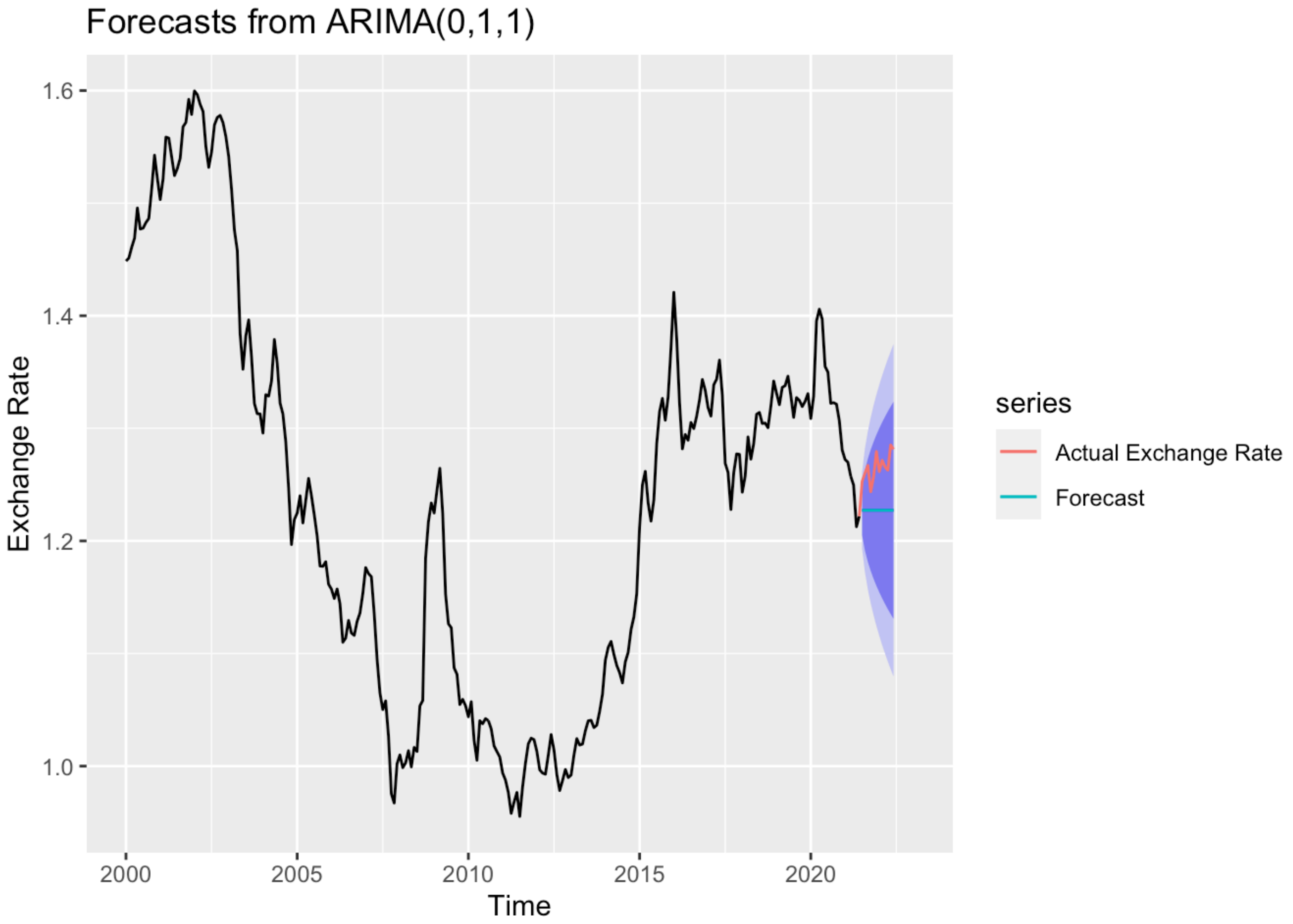
```
jarque.bera.test(residuals(fit0))
```

```
##
##  Jarque Bera Test
##
## data:  residuals(fit0)
## X-squared = 765.47, df = 2, p-value < 2.2e-16
```

Again, no autocorrelation is detected, but the residuals are not normal. Therefore, we can still make a forecast. It is presented graphically below:

```
fore0 <- forecast(fit0, h = 12)
```

```
autoplot(fore0, xlim = c(2000,2023)) + autolayer(actual_data, series = "Actual Exchan
ge Rate") + autolayer(fore0$mean, series = "Forecast") + ylab('Exchange Rate')
```



This forecast, although within the 95% confidence interval, looks less accurate (it does not reflect any trends and is constant during all 12 periods) than the previous one. To determine which forecast is better, it is worth taking a look at the accuracy metrics.

Accuracy metrics for ARIMA(2,1,4):

```
options(digits = 5)
acc01 <- data.frame(accuracy(fore0, actual_data))
acc01 %>% kable() %>% kable_styling()
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil.s.U
Training set	0.00028	0.01727	0.01276	0.01803	1.0334	0.21487	0.01283	NA
Test set	0.03853	0.04025	0.03853	3.03568	3.0357	0.64889	0.16733	2.8506

Accuracy metrics for ARIMA(0,1,1):

```
options(digits = 5)
acc00 <- data.frame(accuracy(fore2, actual_data))
acc00 %>% kable() %>% kable_styling()
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil.s.U
Training set	0.00027	0.01698	0.01262	0.01795	1.0226	0.21247	-0.00372	NA
Test set	0.03220	0.03464	0.03220	2.53528	2.5353	0.54229	0.28644	2.4574

On the test set, the errors are smaller in the first case. Therefore, it can be concluded that ARIMA(2,1,4) performs better in case of the 12-month univariate forecast of the USD/CAD exchange rate.

Lastly, the table of forecast and actual values for this model (ARIMA(2, 1, 4):

```
compare <- data.frame(cbind(fore2$mean, actual_data[-1]))
colnames(compare) <- c('Forecast', 'Real Data')
rownames(compare) <- c('Jul 2021', 'Aug 2021', 'Sep 2021', 'Oct 2021', 'Nov 2021', 'D
ec 2021', 'Jan 2022', 'Feb 2022', 'Mar 2022', 'Apr 2022', 'May 2022', 'Jun 2022')
compare$differ <- compare[,2] - compare[,1]
colnames(compare) <- c('Forecast', 'Real Data', 'Difference')
compare %>% kable() %>% kable_styling()
```

	Forecast	Real Data	Difference
Jul 2021	1.2319	1.2529	0.02099
Aug 2021	1.2399	1.2603	0.02039
Sep 2021	1.2385	1.2671	0.02856
Oct 2021	1.2321	1.2437	0.01161
Nov 2021	1.2277	1.2570	0.02932
Dec 2021	1.2298	1.2794	0.04964
Jan 2022	1.2357	1.2616	0.02593
Feb 2022	1.2389	1.2716	0.03265
Mar 2022	1.2364	1.2658	0.02943
Apr 2022	1.2311	1.2628	0.03173

May 2022	1.2288	1.2852	0.05641
Jun 2022	1.2317	1.2814	0.04972

Multivariate analysis

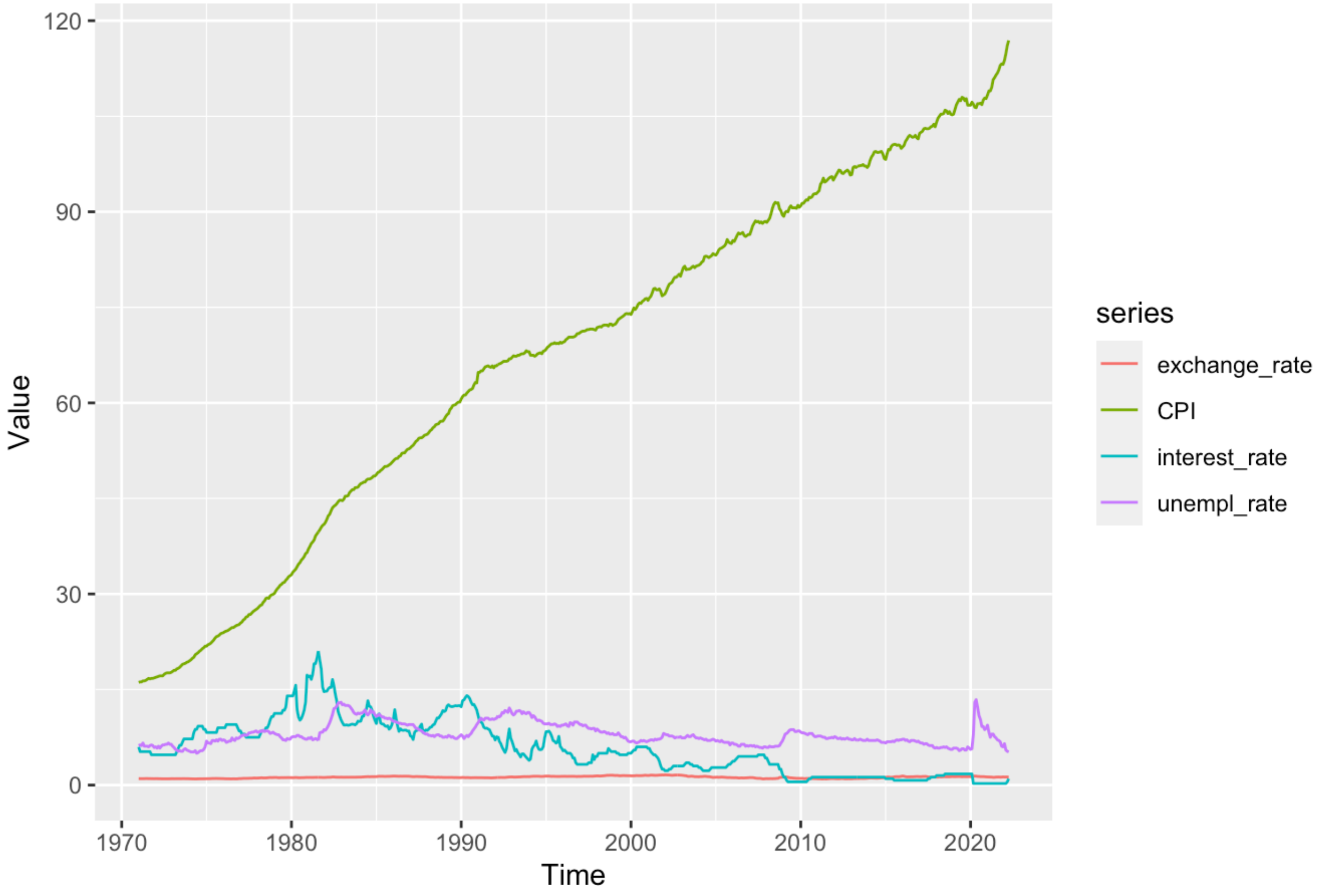
Now we use monthly data of the CAD/USD exchange rate, CPI (base year 2015), the interest rate and the unemployment rate, starting from January 1971 to April 2022 (most up-to-date data) with 616 observations in total.

```
ts_data <- read_excel("~/Desktop/ts_data2.xlsx")
series <- ts(ts_data, start = c(1971, 1), frequency = 12)
actual_data2 <- window(series, start = c(2021, 04), end = c(2022,04))
```

The plot of the series:

```
autoplot(series) + ggtitle("Figure 9. Series of Four Macroeconomic Indicators") + yla
b('Value')
```

Figure 9. Series of Four Macroeconomic Indicators



No tests needed to say this multivariate time series is not stationary: for example, CPI variable has a determined trend upwards. So, we can choose whether we will proceed with this series or take the first difference to make the series stationary. Regardless of the choice, it is still possible to make a forecast and analyze the series properly. We decided to stay with nonstationary series. Therefore, we should keep in mind that we have to take a rather big number of lags - this is necessary for all further actions, such as building IRF, estimating FEDV and so on. We decided to limit the maximum number of lags to 20 before proceeding to the automatic lag selection procedure.

The results of the lag selection:

```
train2 = window(series, end = c(2021,04))
VARselect(train2, lag.max = 20, type = "const")$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     13      2      2     13
```

Two criteria stated that two lags are at best, and the other two selected thirteen lags. We will estimate the model with 13 lags as we have to stick to the huge number of lags.

```
var1 <- VAR(train2, p = 13, type = "const")
#summary(var1)
```

To check the presence of autocorrelation in the VAR(13) model, we perform an asymptotic Portmanteau (a significant number of observations allows it):

```
serial.test(var1, lags.pt = 14, type = "PT.asymptotic")
```

```
## 
##  Portmanteau Test (asymptotic)
## 
## data:  Residuals of VAR object var1
## Chi-squared = 38.8, df = 16, p-value = 0.0011
```

The test shows that the autocorrelation is present (the null hypothesis is rejected), which makes it impossible to make an adequate forecast. Moreover, the residuals are not normal:

```
normality.test(var1, multivariate.only = TRUE)$jb.mul
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object var1
## Chi-squared = 84930, df = 8, p-value <2e-16
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object var1
## Chi-squared = 2119, df = 4, p-value <2e-16
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object var1
## Chi-squared = 82811, df = 4, p-value <2e-16
```

Since it is impossible to proceed with the forecast, we should change the model. In practice, it is usual to keep N small and include only those variables that are correlated and, therefore, useful in forecasting each other. Thus, it can be helpful to get rid of one variable that does not have as much impact on the exchange rate. Let's take a look at the correlation matrix:

```
correl <- data.frame(cor(series, y = NULL, use = "all.obs", method = c("pearson")))
correl %>% kable() %>% kable_styling()
```

	exchange_rate	CPI	interest_rate	unempl_rate
exchange_rate	1.00000	0.31005	-0.06815	0.30612
CPI	0.31005	1.00000	-0.72827	-0.12901
interest_rate	-0.06815	-0.72827	1.00000	0.31237
unempl_rate	0.30612	-0.12901	0.31237	1.00000

From this table, we can see a more or less significant correlation between the exchange rate and CPI, the exchange rate and unemployment, but no correlation between the exchange rate and the interest rate. If we drop it, the performance might be better (at least, we may be able to make predictions), so we will try this option.

With the new series consisted of the exchange rate, CPI and unemployment rate, it was suggested using VAR(13) or VAR(2) model based on the VAR selection:

```
data3 <- data.frame(train2)
data3 <- data3[-3]
series_3 <- ts(data3, start = c(1971, 1), frequency = 12)
VARselect(series_3, lag.max = 20, type = "const")$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     13      2      2     13
```

```
var3 <- VAR(series_3, p = 13, type = "const")
```

Again, since we have nonstationary series, we choose and estimate the model with a bigger number of lags. The asymptotic PT and normality tests were performed as well:

```
serial.test(var3, lags.pt = 20, type = "PT.asymptotic")
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var3
## Chi-squared = 66.6, df = 63, p-value = 0.36
```

```
normality.test(var3, multivariate.only = TRUE)
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object var3
## Chi-squared = 71324, df = 6, p-value <2e-16
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object var3
## Chi-squared = 1967, df = 3, p-value <2e-16
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object var3
## Chi-squared = 69358, df = 3, p-value <2e-16
```

The Portmanteau test did not reject the null hypothesis of no autocorrelation this time. Although the residuals are not normal, now we can still make a forecast

Beforehand, it would be helpful to check Granger Causality. There is no point in keeping a variable if it does not help to improve a forecast of the initial variable (in our case, exchange rate). The results are presented below:

```
causality(var3, "CPI")$Granger
```

```
##
##  Granger causality H0: CPI do not Granger-cause exchange_rate
##  unempl_rate
##
## data:  VAR object var3
## F-Test = 1.88, df1 = 26, df2 = 1653, p-value = 0.0046
```

```
causality(var3, "exchange_rate")$Granger
```

```
##
##  Granger causality H0: exchange_rate do not Granger-cause CPI
##  unempl_rate
##
## data:  VAR object var3
## F-Test = 1.95, df1 = 26, df2 = 1653, p-value = 0.0029
```

```
causality(var3, "unempl_rate")$Granger
```

```
## 
##  Granger causality H0: unempl_rate do not Granger-cause exchange_rate
##  CPI
## 
## data:  VAR object var3
## F-Test = 0.969, df1 = 26, df2 = 1653, p-value = 0.51
```

It appears that at the 90% level, CPI measure Granger-causes both the the exchange and unemployment rate (null hypothesis can be rejected). Likewise, at the 95% level of confidence, the exchange rate Granger-causes CPI and unemployment rate. This means that at least CPI can improve the forecast of the exchange rate (and vice-versa). Lastly, the unemployment rate does not Granger-cause the exchange rate and CPI together, but it is impossible to detect individual impact here.

To check the relationship between the unemployment rate and the exchange rate, we decided to limit the series to these two variables only. Meanwhile, it is possible to estimate the VAR(3) model for unemployment and exchange rate since there is no autocorrelation (refer to VAR selection procedure and autocorrelation test below). What we want to know now is the relationship between exchange rate and unemployment rate:

```
###VAR(6) w/ unemp and exch rate
data2 <- data.frame(train2)
data2 <- data2[-c(2,3)]
series_2 <- ts(data2, start = c(1971, 1), frequency = 12)
VARselect(series_2, lag.max = 20, type = "const")$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n) 
##      3      3      2      3
```

```
var2 <- VAR(series_2, p = 3, type = "const")
serial.test(var2, lags.pt = 10, type = "PT.asymptotic")
```

```
## 
##  Portmanteau Test (asymptotic)
## 
## data:  Residuals of VAR object var2
## Chi-squared = 44.7, df = 28, p-value = 0.024
```

```
causality(var2, 'exchange_rate')$Granger
```

```
##
##  Granger causality H0: exchange_rate do not Granger-cause unempl_rate
##
## data:  VAR object var2
## F-Test = 6.63, df1 = 3, df2 = 1188, p-value = 0.00019
```

```
causality(var2, 'unempl_rate')$Granger
```

```
##
##  Granger causality H0: unempl_rate do not Granger-cause exchange_rate
##
## data:  VAR object var2
## F-Test = 0.828, df1 = 3, df2 = 1188, p-value = 0.48
```

So, it turned out that the unemployment rate does not Granger-cause the exchange rate (all the coefficients on lagged values of the unemployment rate are zero in the equation for the exchange rate), which means it does not help to improve the forecast, however the exchange rate does Granger-cause the unemployment rate.

Since the unemployment rate variable does not seem to help improve the forecast of the exchange rate's future values, we also decided to build the third model which consisted of the vector of two variables: CPI and the exchange rate. In short, the model with 13 lags was estimated, no autocorrelation was found. What is surprising - the CPI indicator does not Granger-cause the exchange-rate, and the vice-versa is not true either:

```
#VAR(13) for cpi and unemp
data4 <- data.frame(train2)
data4 <- data4[c('CPI', 'exchange_rate')]
series_4 <- ts(data4, start = c(1971, 1), frequency = 12)
VARselect(series_4, lag.max = 20, type = "const")$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     13      2      2     13
```

```
var4 <- VAR(series_4, p = 13, type = "const")
serial.test(var4, lags.pt = 20, type = "PT.asymptotic") #no autocor
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var4
## Chi-squared = 36.6, df = 28, p-value = 0.13
```

```
causality(var4, 'exchange_rate')$Granger
```

```
##
##  Granger causality H0: exchange_rate do not Granger-cause CPI
##
## data:  VAR object var4
## F-Test = 1.18, df1 = 13, df2 = 1128, p-value = 0.29
```

```
causality(var4, 'CPI')$Granger
```

```
##
##  Granger causality H0: CPI do not Granger-cause exchange_rate
##
## data:  VAR object var4
## F-Test = 1.49, df1 = 13, df2 = 1128, p-value = 0.12
```

Anyway, we decided to make a forecast for all three models that we mentioned and compare their performance:

1. VAR(13) with three variables: exchange rate, CPI, unemployment rate;
2. VAR(3) with two variables: exchange rate, unemployment rate;
3. VAR(13) with two variables: exchange rate, CPI.

```
#FORECAST VAR(13) w/ three variables
varfore3 <- forecast(var3, h = 12)
plot3 <- autoplot(varfore3$forecast$exchange_rate, xlim = c(2000,2023),  main = "Exch
ange Rate Forecast from VAR(13)") + autolayer(actual_data2[,1], series = "Actual Valu
e") + autolayer(varfore3$forecast$exchange_rate$mean, series = "Forecast") + ylab('Ex
change Rate')
#plot4 <- autoplot(varfore3$forecast$unempl_rate, xlim = c(2000,2023),  main = "Unemp
loyment Rate Forecast") + autolayer(actual_data2[,'unempl_rate'], series = "Actual Va
lue") + autolayer(varfore3$forecast$unempl_rate$mean, series = "Forecast") + ylab('Un
employment Rate')
#plot44 <- autoplot(varfore3$forecast$CPI, xlim = c(2000,2023),  main = "CPI Forecast
") + autolayer(actual_data2[,'CPI'], series = "Actual Value") + autolayer(varfore3$fo
recast$CPI$mean, series = "Forecast") + ylab('CPI')
#grid.arrange(plot3, plot4, ncol = 1)
```

```
#Forecast VAR(3) w/ unemp and exchange
varfore <- forecast(var2, h = 12)
plot1 <- autoplot(varfore$forecast$exchange_rate, xlim = c(2000,2023),  main = "Excha
nge Rate Forecast from VAR(3)") + autolayer(actual_data2[,1], series = "Actual Value"
) + autolayer(varfore$forecast$exchange_rate$mean, series = "Forecast") + ylab('Excha
nge Rate')
#plot2 <- autoplot(varfore$forecast$unemp, xlim = c(2000,2018),  main = "Unemployment
Rate Forecast from VAR(3)") + autolayer(actual_data2[,4], series = "Actual Value") +
autolayer(varfore$forecast$unemp$mean, series = "Forecast") + ylab('Unemployment Rate
')
#grid.arrange(plot1, plot2, ncol = 1)
```

```
#Forecast VAR(13) w/ cpi and exchange
varfore4 <- forecast(var4, h = 12)
plot7 <- autoplot(varfore4$forecast$exchange_rate, xlim = c(2000,2023),  main = "Exch
ange Rate Forecast from VAR(13)") + autolayer(actual_data2[,1], series = "Actual Valu
e") + autolayer(varfore4$forecast$exchange_rate$mean, series = "Forecast") + ylab('Ex
change Rate')
#plot8 <- autoplot(varfore4$forecast$cpi, xlim = c(2000,2018),  main = "CPI Forecast
from VAR(13)") + autolayer(actual_data2[,2], series = "Actual Value") + autolayer(var
fore4$forecast$cpi$mean, series = "Forecast") + ylab(' CPI')
#grid.arrange(plot7, plot8, ncol = 1)
```

Accuracy measures of the exchange rate forecast for three multivariate models are presented below. We also added the results from the univariate forecast:

```
acc1 <- data.frame(accuracy(varfore$forecast$exchange_rate$mean, actual_data2[,1])[1,
])  #unemp and exchange
acc3 <- data.frame(accuracy(varfore3$forecast$exchange_rate$mean, actual_data2[,1])[1
,]) #best three variables
acc4 <- data.frame(accuracy(varfore4$forecast$exchange_rate$mean, actual_data2[,1])[1
,]) # cpi and exchange
acc2 <- data.frame(accuracy(fore2, actual_data)[2,])
acc2 <- acc2[-6,]
dfcomp <- cbind(acc3, acc1, acc4, acc2)
colnames(dfcomp) <- c("Multivariate1 (CPI+unemp+exchange)", "Multivariate2 (exchange+
unemp)", "Multivariate3 (CPI+exchange)", "Univariate")
dfcomp %>% kable() %>% kable_styling()
```

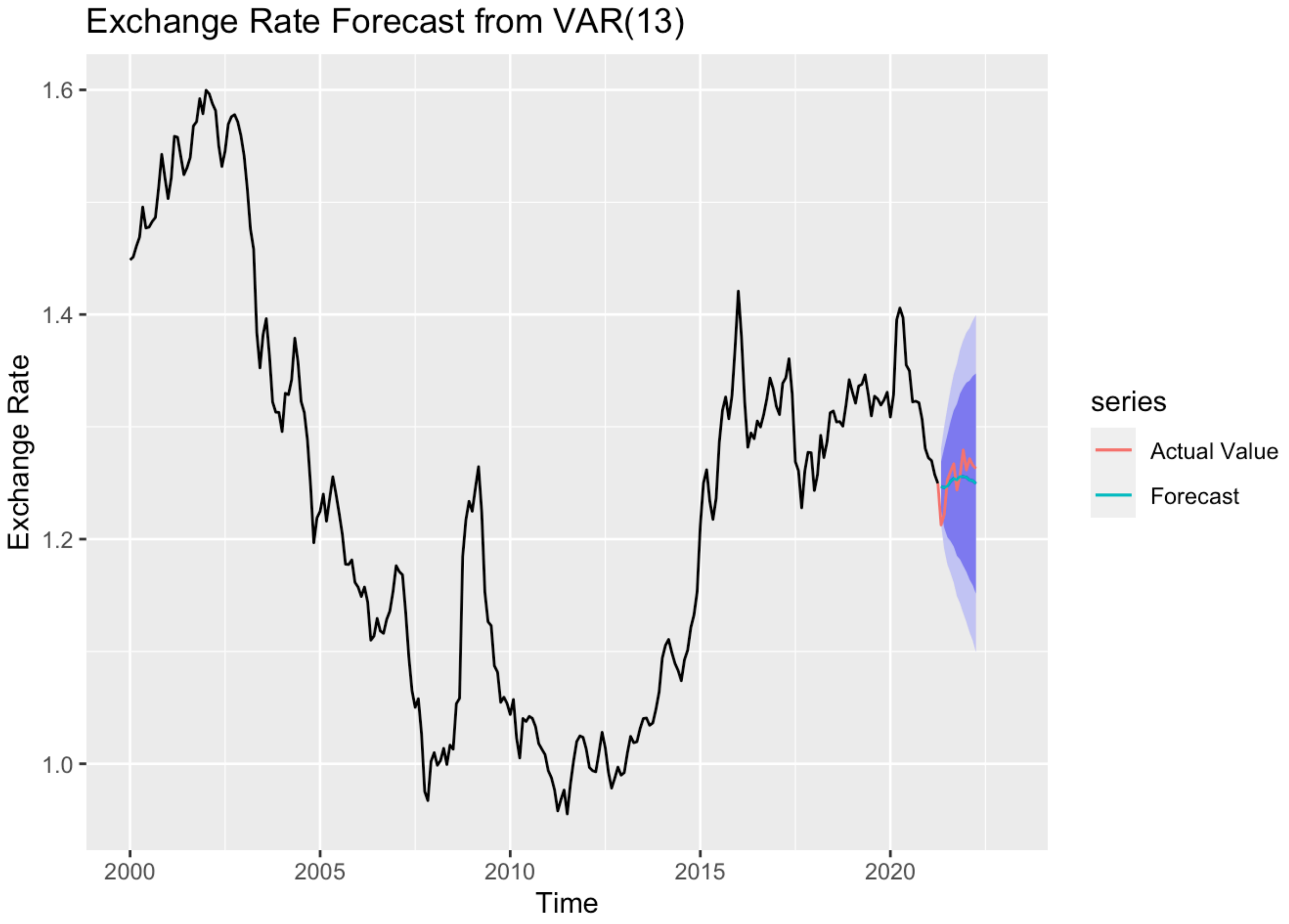
	Multivariate1 (CPI+unemp+exchange)	Multivariate2 (exchange+unemp)	Multivariate3 (CPI+exchange)	Univariate
ME	-0.04262	0.00373	0.00316	0.03220
RMSE	0.04382	0.01922	0.01695	0.03464
MAE	0.04262	0.01615	0.01436	0.03220

MPE	-3.40555	0.27393	0.23165	2.53528
MAPE	3.40555	1.29352	1.14960	2.53528
ACF1	-0.06154	0.44613	0.38413	0.28644
Theil’s U	2.68128	1.01992	0.89308	2.45736

As seen from the table, the first multivariate model (that has three variables - the exchange rate, cpi and the unemployment rate) has the largest error measures. It can be explained by the fact that the unemployment rate does not Granger-cause the exchange rate (that’s why the second model also performed relatively poorly. On the contrary, the third forecast performed much better and had the lowest error in each accuracy measure. The univariate model also showed good performance of the same level as the third multivariate model.

To sum up, below is the best exchange rate forecast (i.e. VAR(13) with CPI and exchange rate) plotted along with the table of actual and forecast values:

```
plot7
```



```
comparew <- data.frame(cbind(varfore4$forecast$exchange$mean, actual_data2[-1,1]))
colnames(comparew) <- c('Forecast', 'Real Data')
rownames(comparew) <- c('May 2021', 'Jun 2021', 'Jul 2021', 'Aug 2021', 'Sep 2021', '
Oct 2021', 'Nov 2021', 'Dec 2021', 'Jan 2022', 'Feb 2022', 'Mar 2022', 'Apr 2022')
comparew$differ2 <- comparew[,2] - comparew[,1]
colnames(comparew) <- c('Forecast', 'Real Data', 'Difference')
comparew %>% kable() %>% kable_styling()
```

	Forecast	Real Data	Difference
May 2021	1.2470	1.2126	-0.03438
Jun 2021	1.2457	1.2219	-0.02376
Jul 2021	1.2473	1.2529	0.00556
Aug 2021	1.2515	1.2603	0.00883
Sep 2021	1.2540	1.2671	0.01306
Oct 2021	1.2528	1.2437	-0.00905
Nov 2021	1.2558	1.2570	0.00119
Dec 2021	1.2556	1.2794	0.02375
Jan 2022	1.2551	1.2616	0.00654
Feb 2022	1.2525	1.2716	0.01906
Mar 2022	1.2520	1.2658	0.01376
Apr 2022	1.2494	1.2628	0.01335

What we can actually see is that VAR model managed to get more correct prediction in comparison to ARIMA. As VAR was trying to capture more complex dependencies, it was able to capture a sudden shock which made the real exchange rate go up sharply. ARIMA simply relied on the previous lags and therefore its predictions tended to the average of the previous values which was not enough in our case.

Moving to the structural analysis, the order of variables is vital here - the result will differ depending on the ordering. Therefore, we decided to reorder data so that CPI be ahead of the exchange rate as it is more powerful based on the previous tests. The unemployment rate was set in the last column as it does not have a substantial effect on either the exchange rate or CPI.

VAR is actually a reduced form model. Structural VAR reflects the implied structure among variables in the system. It is impossible to estimate SVAR directly, so we have to estimate a reduced-form VAR and then restore the parameters of the structural form. For example, we can compute $(B_0)^{-1}$ using Choleski decomposition:

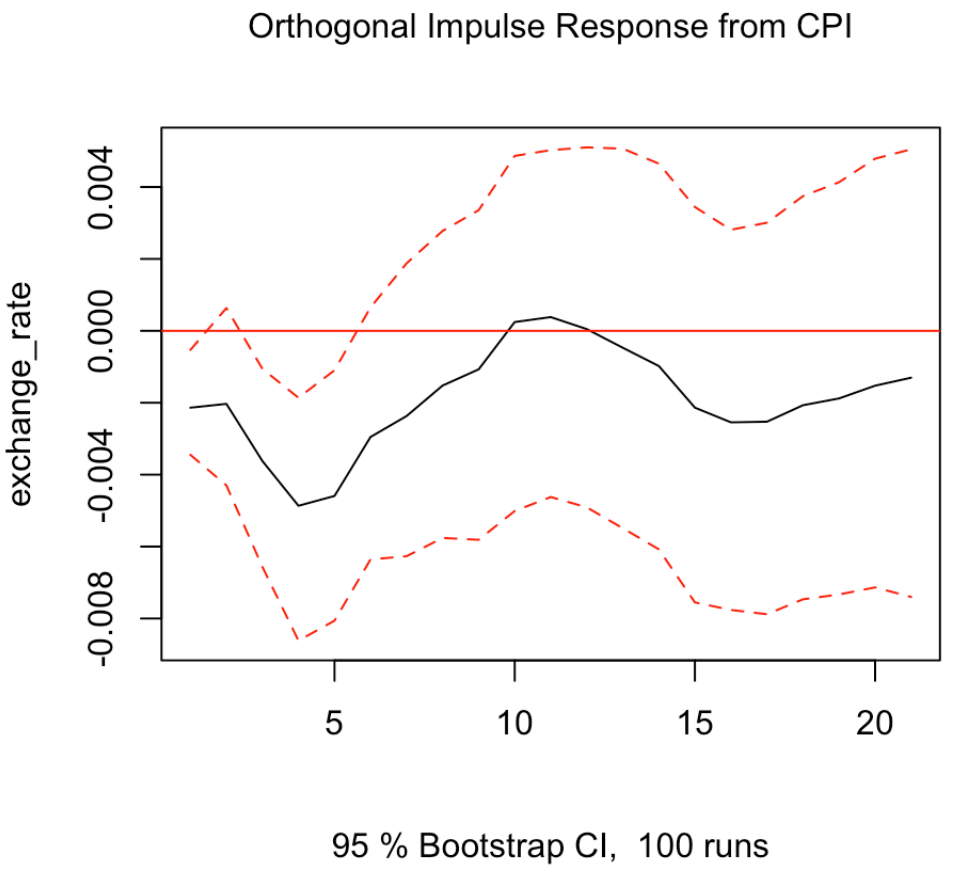
```
#another order
data_reorder <- series_3[, c('CPI', 'exchange_rate', 'unempl_rate')]
var3a <- VAR(data_reorder, p = 13, type = "const")
Omega_hat <- summary(var3a)$covres
chol(Omega_hat) %>% t()
```

```
##                      CPI exchange_rate unempl_rate
## CPI            0.2463783     0.0000000     0.00000
## exchange_rate -0.0021379     0.0169972     0.00000
## unempl_rate   -0.0349995     0.0090478     0.29373
```

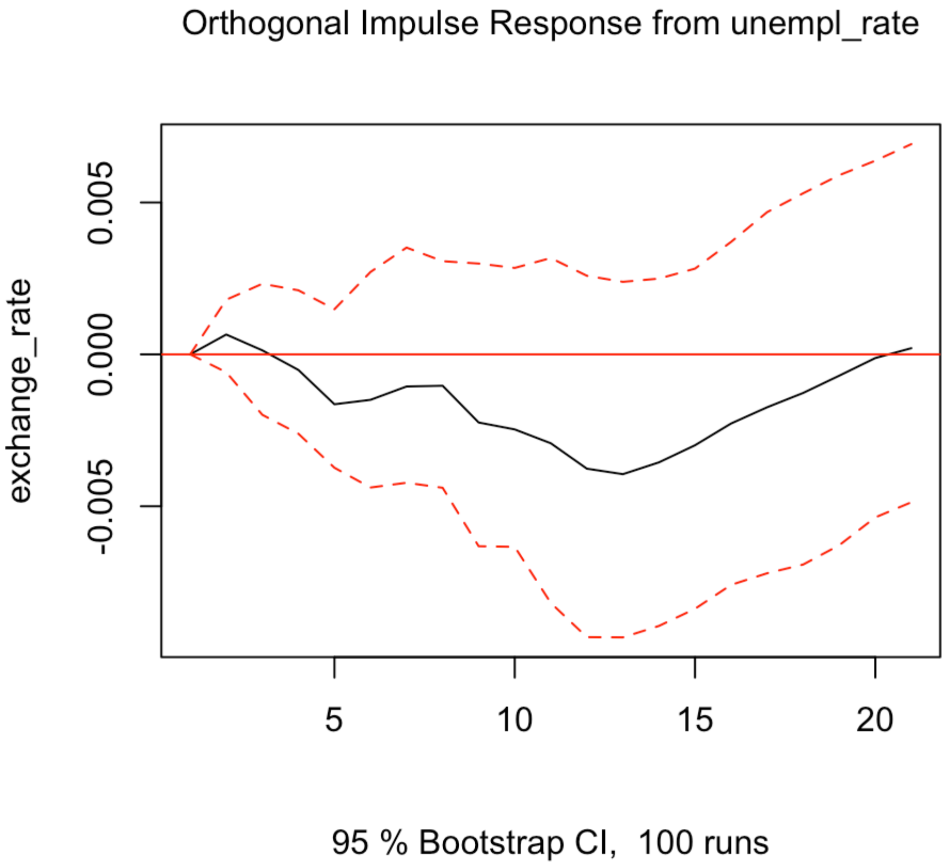
If we have a structural form, we can obtain impulse response functions and forward error variance decomposition. With the help of IRFs and FEVDs, we can examine the effect of other variables on the dependent variable (exchange rate), the sign of the impact and its length.

First, we investigate the IRF which means the responsiveness of the exchange rate to shocks on itself, CPI and unemployment rate. Figures below show the IRFs of CPI, unemployment rate and exchange rate respectively:

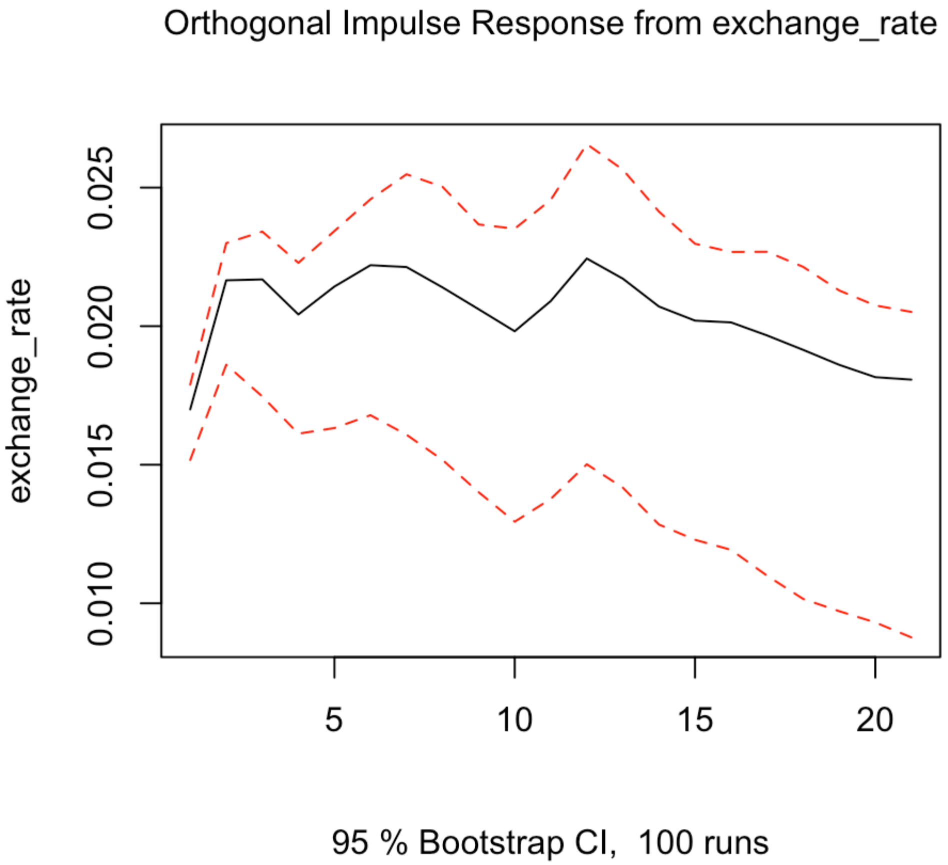
```
irf3a <- irf(var3a, impulse = c('CPI'), response = c("exchange_rate"), n.ahead = 20,
cumulative = FALSE)
plot(irf3a)
```



```
irf4a <- irf(var3a, impulse = "unempl_rate", response = c("exchange_rate"), n.ahead =
20, cumulative = FALSE)
plot(irf4a)
```



```
irf3c <- irf(var3a, impulse = c('exchange_rate'), response = c("exchange_rate"), n.ah
ead = 20, cumulative = FALSE)
plot(irf3c)
```



Starting with CPI, the price level increases are supposed to bring negative effect to the USD/CAD exchange rate in the short-run, having its peak in the 4th period. That was expected since the higher prices discourage agents from buying domestic goods, so the demand for the local (Canadian) currency is going down; therefore, the exchange rate USD/CAD decreases (the US dollar becomes more valuable). The impulse response is significant from appr. the second period to the sixth period after which the confidence interval started to have zero within its borders.

The sign of the unemployment rate effect was fluctuating; however, we can notice that the confidence interval always includes zero, that's why the response is insignificant.

In case of a structural shock to the exchange rate itself, it gives a positive permanent response for growth of the exchange rate. In the short and medium period the response fluctuates; in long-run, it decreases steadily.

A shock to the variable will not only have an impact on this variable, but it will also influence other variables. We can see that if we perform forecast error variance decomposition as well. The FEVD allows to analyze the contribution of variable j to the h-step forecast error variance of variable k. We are interested in the exchange rate, so we will only leave the results of FEVD of this variable:

```
fevd <- data.frame(fevd(var3a, n.ahead = 12)$exchange)
fevd[, c('exchange_rate', 'CPI','unempl_rate')]
```

```
##    exchange_rate      CPI unempl_rate
## 1        0.98443 0.015575  0.00000000
## 2        0.98810 0.011339  0.00055657
## 3        0.98220 0.017447  0.00035535
## 4        0.97269 0.026894  0.00041827
## 5        0.96781 0.030621  0.00156682
## 6        0.96978 0.028114  0.00210880
## 7        0.97238 0.025486  0.00213125
## 8        0.97496 0.022890  0.00215457
## 9        0.97609 0.020746  0.00316316
## 10       0.97685 0.018907  0.00424771
## 11       0.97717 0.017220  0.00560576
## 12       0.97674 0.015579  0.00767901
```

As seen, most of the error variance of the exchange rate can be explained by the shocks to their own series. Only up to 3% of the error variance is explained by CPI, while the unemployment rate does not explain anything at all. The same situation can be observed if we take a look at FEVD of CPI or unemployment rate - they can be mostly explained by their own shocks.

Conclusion

To conclude, with the help of a more complex methods of econometric analysis, the mulitvariate VAR model turned out to perform better than any other suggested specification of the VAR model and was able to predict the upward trend of the currency in 2021-2022. However, despite being a “simple” model, the ARIMA(2, 1, 4) model showed great results as well. Overall, the exchange rate was mostly explained by its own shocks.