Time Series 413, Assignment 7

Multivariate Time Series Models (TS7)

Reed Ballesteros

Northwestern University SPS, Fall 2022
MSDS-413-DL
Instructor: Dr. Jamie D. Riggs, Ph.D
2022-11-07

The file q-fdebt.txt contains the U.S. quarterly federal debts held by foreign and international investors, and federal reserve banks. The data are from the Federal Reserve Bank of St. Louis, from 1970 to 2012 for 171 observations, and not seasonally adjusted. The debts are in billions of dollars.

- year: year of the debts
- mon: starting month of the quarterly debts
- hbfin: debt held by foreign and international investors
- hbfrbn: debt held by federal reserve banks

Your objective is to explore the time series behavior of these data sets including EDA, modeling,model diagnostics, and interpretation.

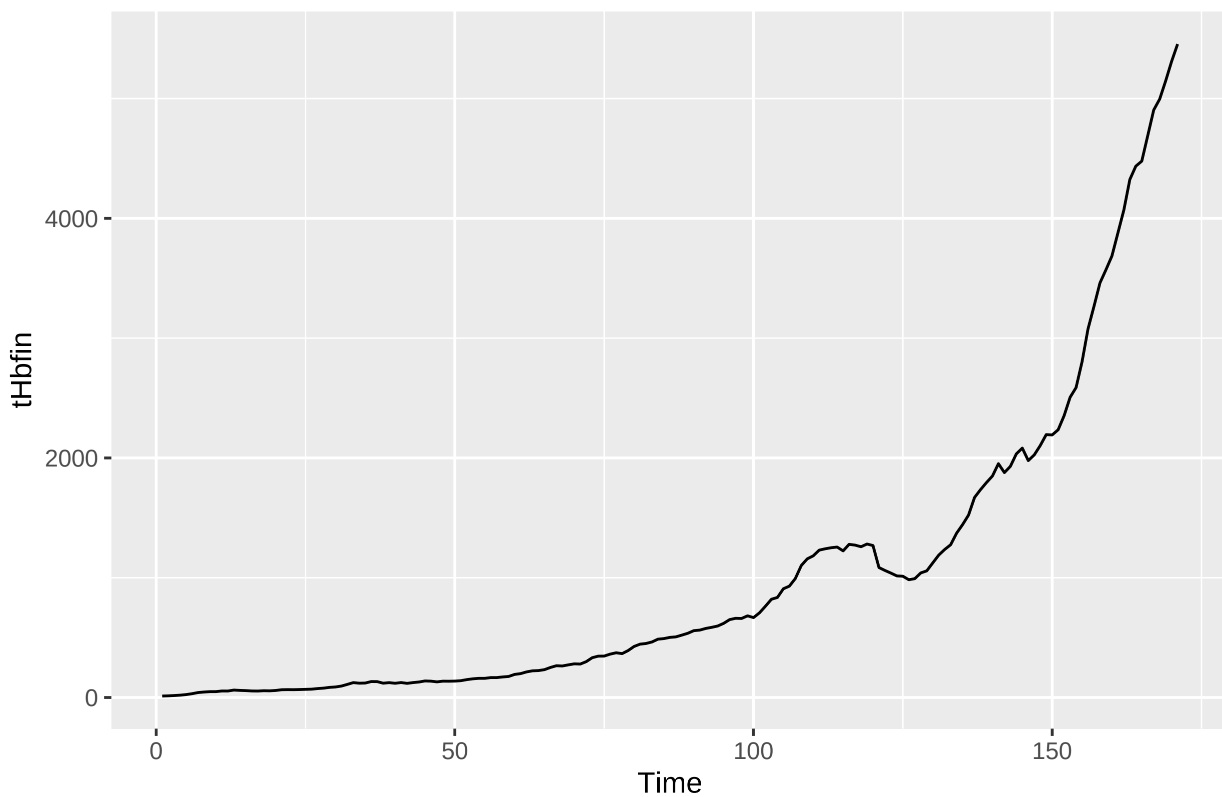
1. Debt (30 points)

Use the file q-fdebt.txt which contains the U.S. quarterly federal debts held by foreign and international investors, and federal reserve banks.

1.1. Use EDA to justify a log transformation and a first difference transformation, zit, of each time series for $i = 1, 2$ hbfin and hbfrbn, respectively.

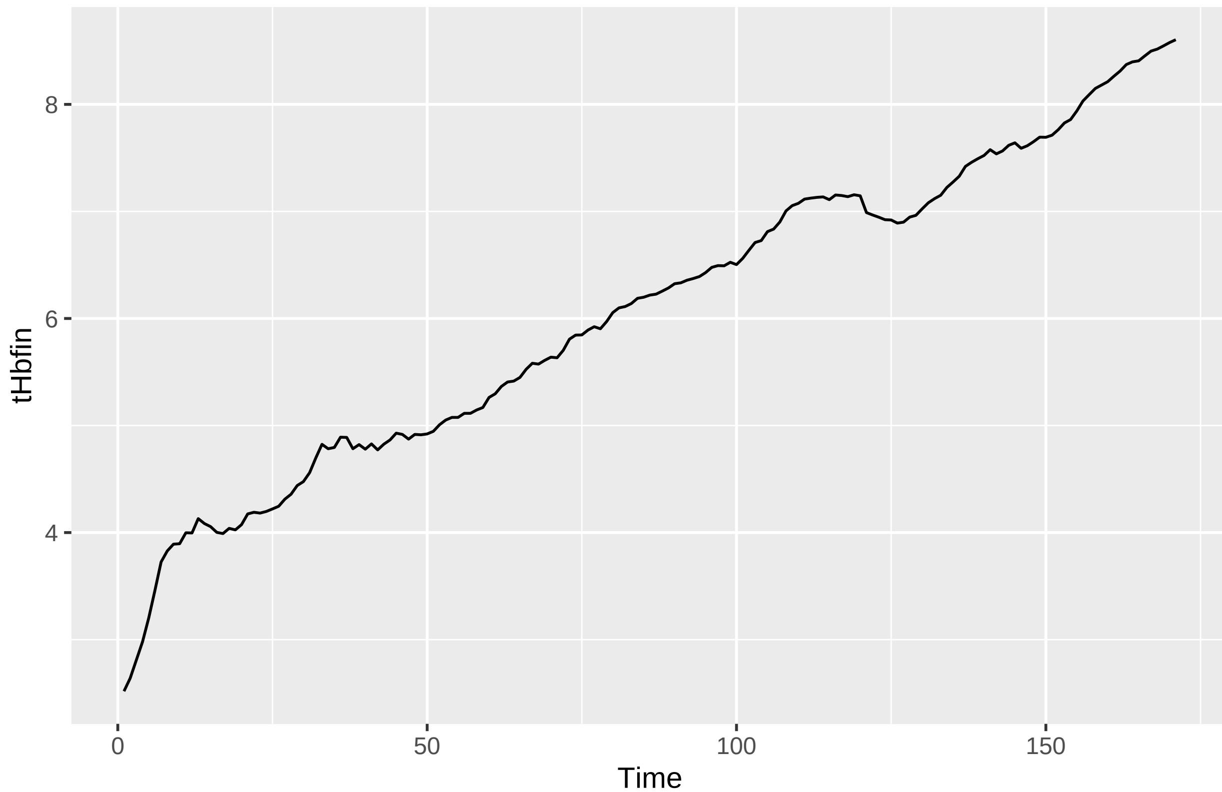
EDA: hbfin

Time series plot: hbfin



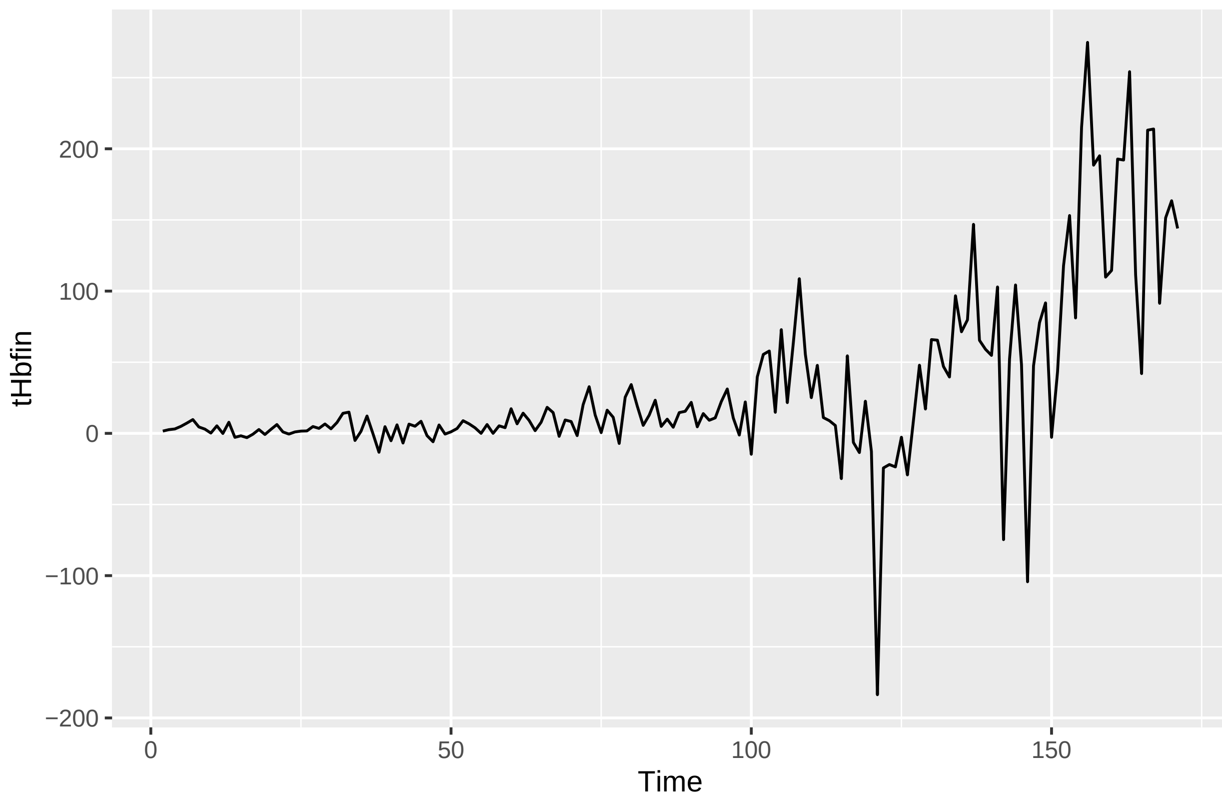
Plotting the hbfin data shows the data with a ‘slow’ curving upward trend for the first 36 years, followed by a sharp upward increase in the last 7 years. That being said, the data does not show a mean 0 nor constant variance, and we’ll need to transform the data to make it fit a proper VAR-based model.

Time series plot: log(hbfin)



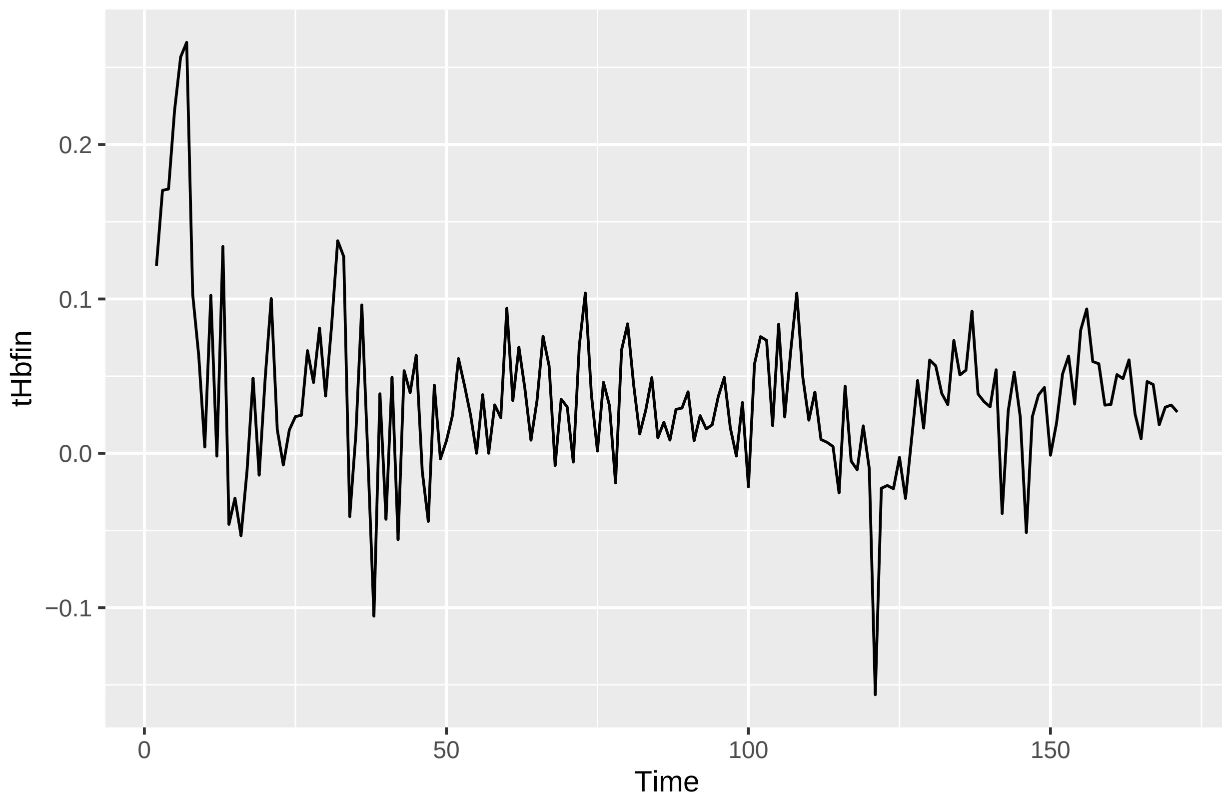
Attempting a log(hbfin) transformation does not meet Gaussian requirements such that the data still shows an upward trend in which the mean is not 0 and non-constant variance, thus showing a trend.

Time series plot: diff(hbfin)



While a diff(hbfin) transformation somewhat displays some non-constant variance and mean 0 in the first 28 years of data, it does not show the same for the last 14 years.

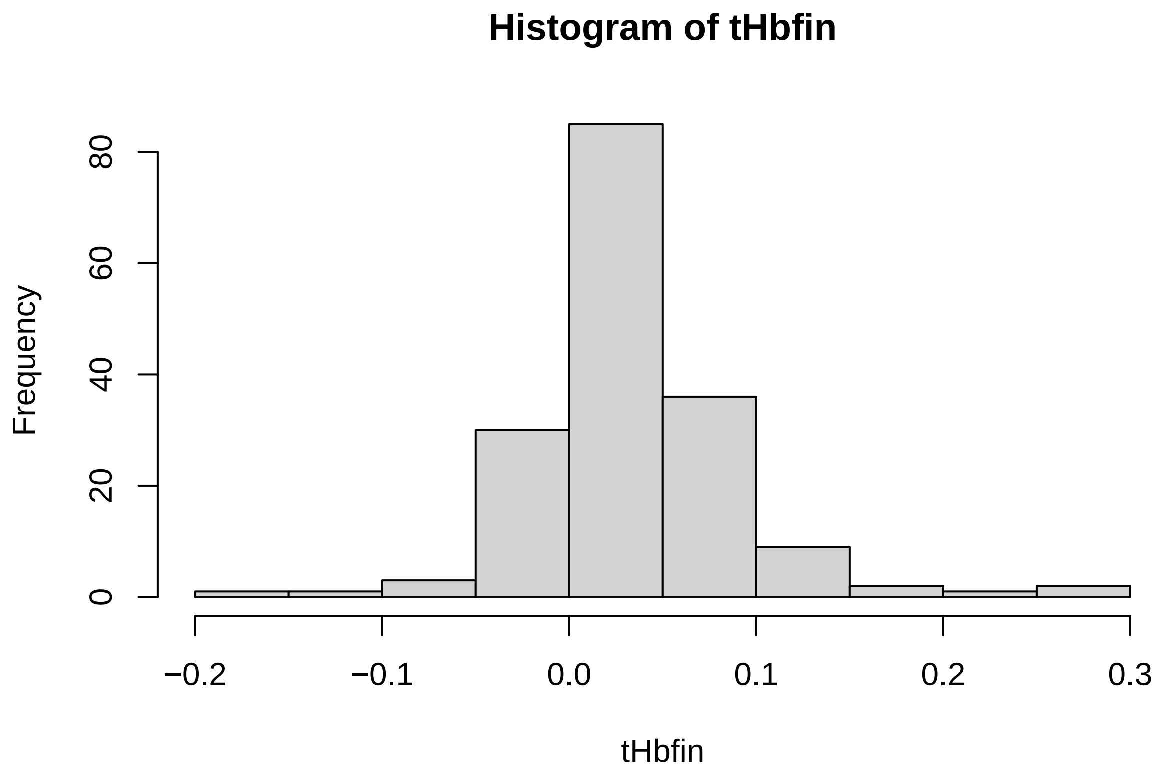
Time series plot: diff(log(hbfin))



While there might be some outliers around the first 3 years and year 32, the diff(log(hbfin)) transformation shows a just-above 0 mean and some pattern of constant variance. While it might not be exact, it could be close as the variance ranges from as low as -0.15 to as high as 0.27.

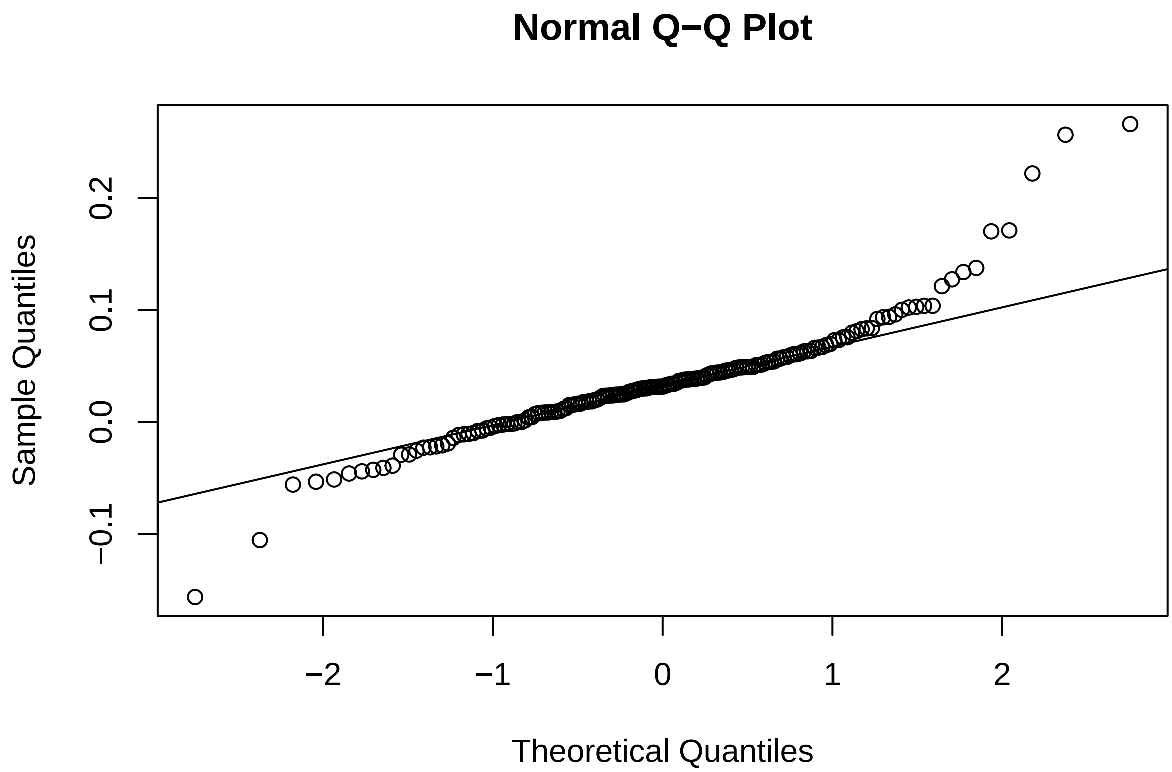
We will continue our EDA with the diff(log(hbfin)) transformation.

Histogram: diff(log(hbfin))



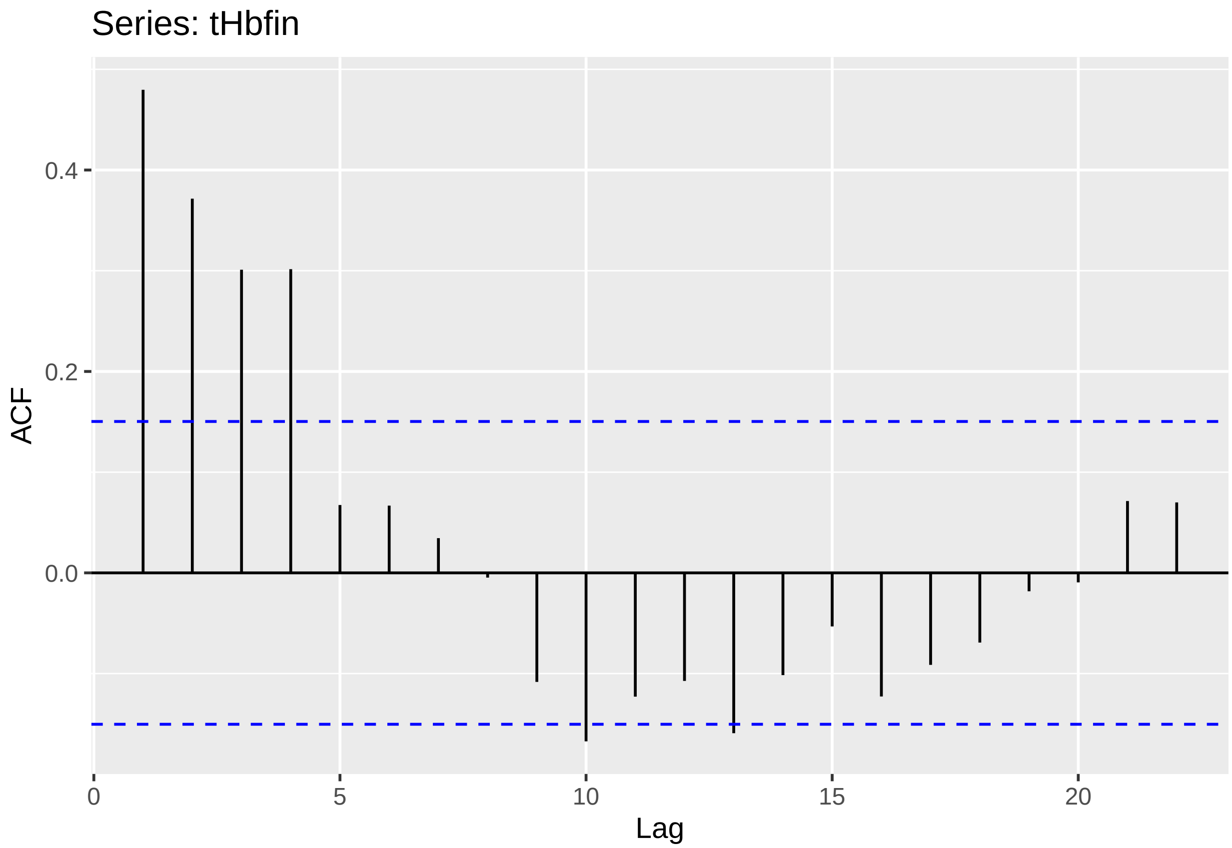
We see a tall right-skewed distribution, which does not conform to a normal Gaussian form.

Q-Q Plot: diff(log(hbfin))



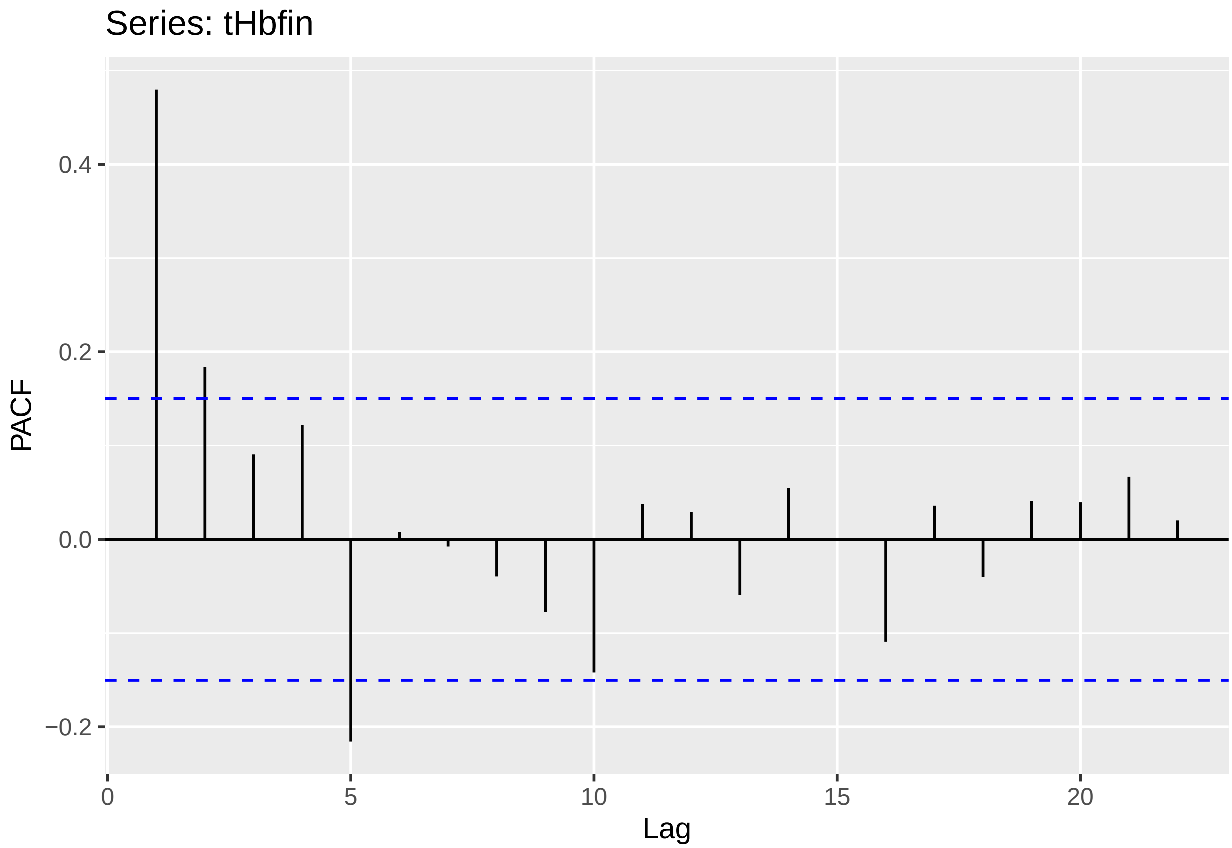
We observe very thick tails, indicating very tall Kurtosis, thus showing non-normalcy in respect to a Gaussian PDF. We can also observe right skewness due to the right tail looks thicker than the other.

ACF Plot: diff(log(hbfin))



We observe a fairly stationary ACF plot with some possible cycling, but most of it is contained within the 95% confidence interval (CI) threshold.

PACF Plot: diff(log(hbfin))



We observe a fairly stationary PACF plot, and from it we can use a possible AR(2) component.

T-Test for Mean 0: diff(log(hbfin))

```
##
##  One Sample t-test
##
## data:  data
## t = 8.9093, df = 169, p-value = 7.936e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.02787022 0.04373674
## sample estimates:
##  mean of x
## 0.03580348
##
## T-Test: mean *NOT* statistically zero, linear trend present ->
## reject H0
```

```
## [1] FALSE
```

While the 95% confidence interval (CI) range is not 0, the lower and upper bounds is relatively close to zero, as well as the calculated mean. In this case, we would call the expected mean of diff(log(hbfin)) to be very close to zero, in some ways accepted to be zero, with almost no signs of a linear trend.

Skewness: diff(log(hbfin))

```
##      skew    lwr.ci    upr.ci
## 0.9540442 0.9780590 1.0442407
## Skew: has *RIGHT* skewness,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(hbfin)) has a distribution with right skewness, thus showing non-normalcy in respect to a Gaussian PDF.

(excess) Kurtosis: diff(log(hbfin))

```
##     kurt   lwr.ci   upr.ci
## 4.724453 4.988993 5.135115
## Kurt: has *TALL thick-tailed* (excess) kurtosis,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(hbfin)) has a distribution with tall (excess) Kurtosis, thus showing non-normalcy in respect to a Gaussian PDF.

Constant variance: Breush-Pagan Test - diff(log(hbfin))

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(data ~ seq(1, length(data)))
## BP = 16.191, df = 1, p-value = 5.727e-05
##
## Breusch-Pagan: *NON*-constant variance, possible clustering,
## heteroscedastic -> reject H0
```

```
##    BP
## FALSE
```

Like what the plot shows, the Breusch-Pagan test confirms non-constant variance for diff(log(hbfin)).

Lag Dependency: diff(log(hbfin))

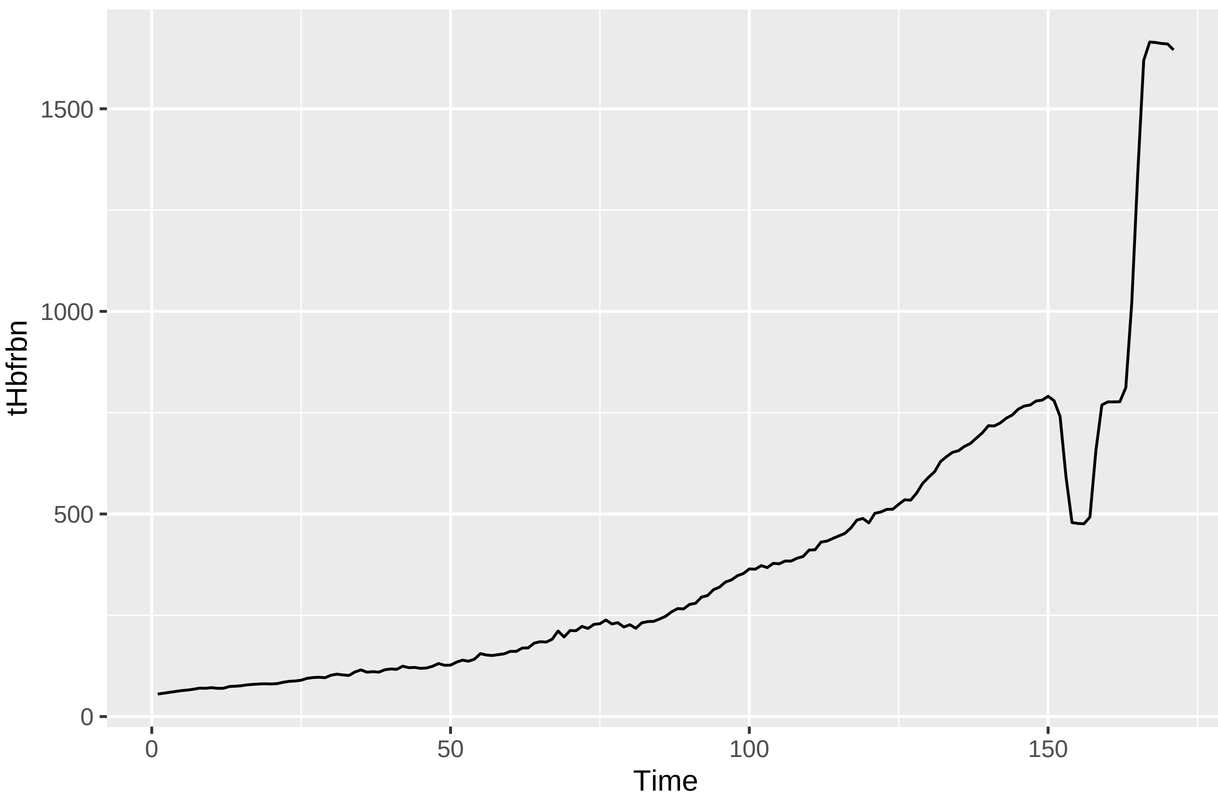
```
##
##  Box-Ljung test
##
## data:  data
## X-squared = 146.66, df = 30, p-value < 2.2e-16
##
## Box-Ljung: implies dependency present over 30 lags,
## autocorrelation present -> reject H0
```

```
## [1] FALSE
```

With a Box-Ljung test p-value < 0.05 for the diff(log(hbfin)) data, we observe lag dependency and thus serial autocorrelation.

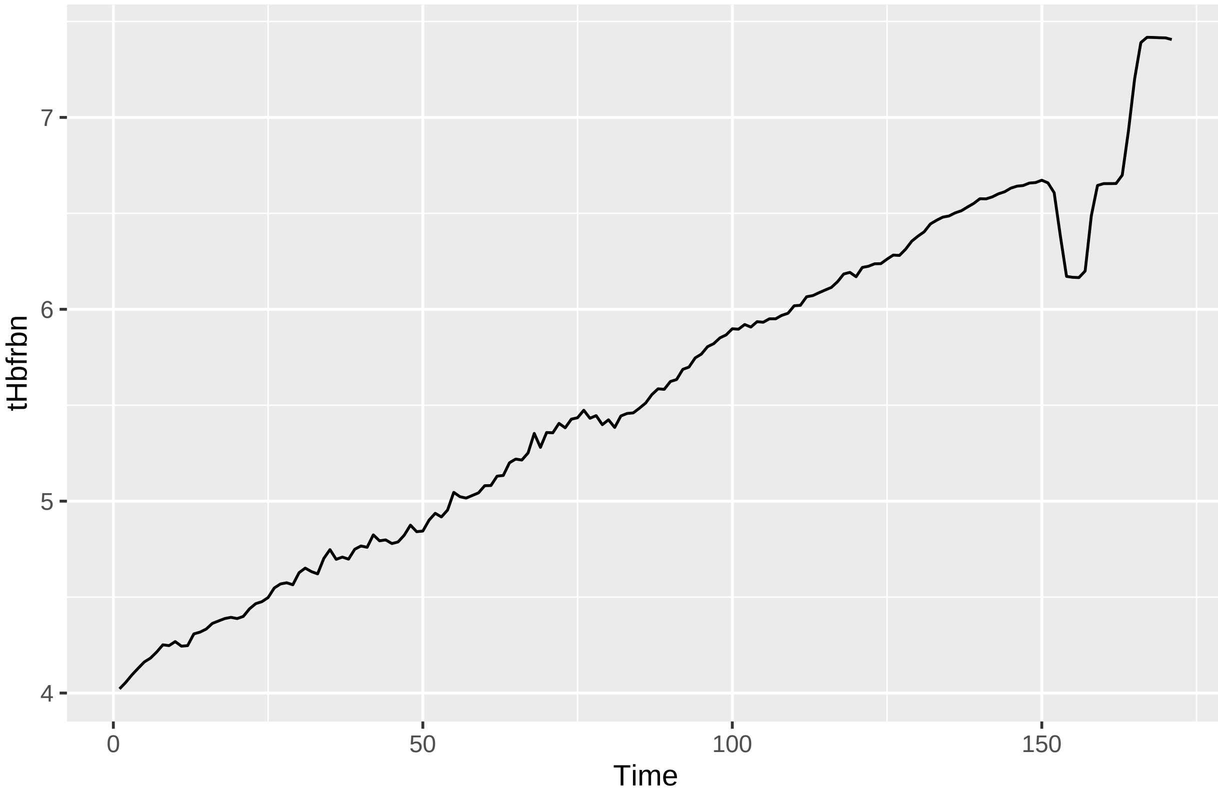
EDA: hbfrbn

Plot: hbfrbn



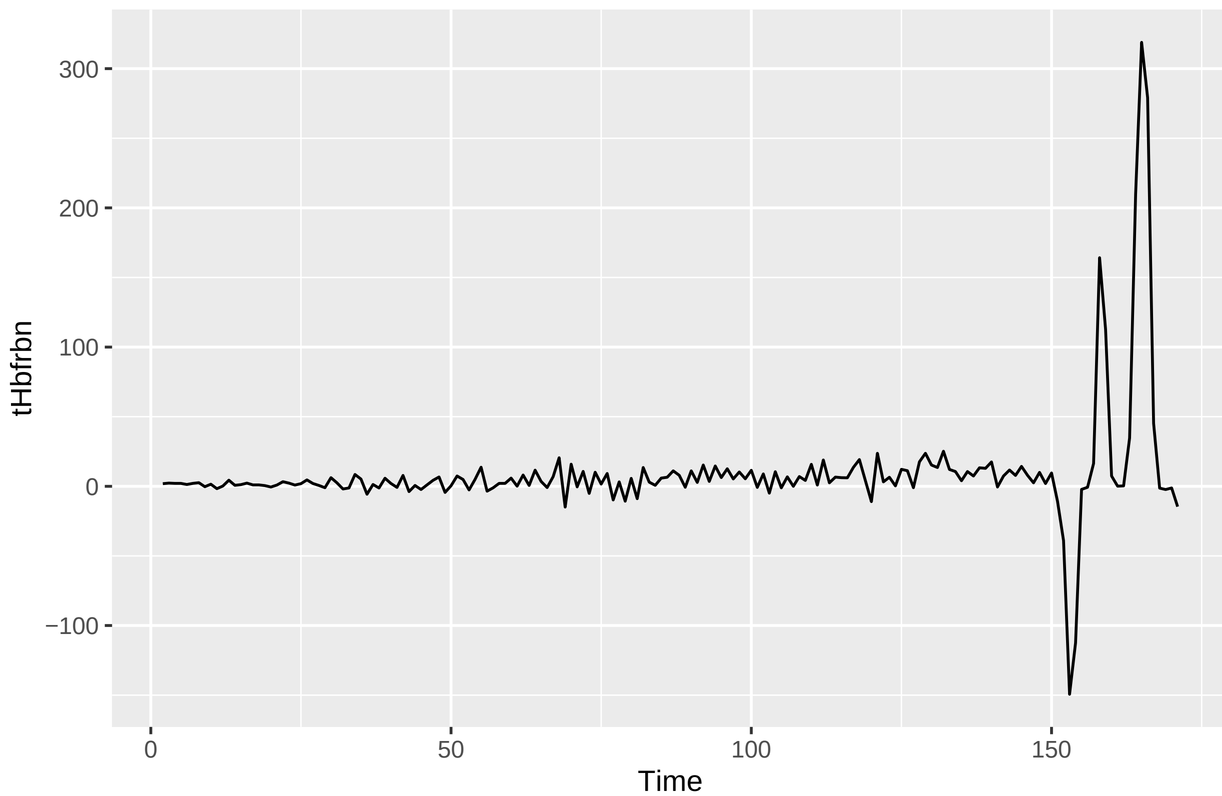
The plot of hbfrbn has a similar trend to that of hbfin, a 'slow' curving upward trend for the first 36 years, followed by a sharp upward increase in the last 7 years. The trend shows that there is no mean 0 and shows non-constant variance. We will conduct other transformations on the hbfrbn data to conform it towards a normal, Gaussian-like form.

Plot: log(hbfrbn)



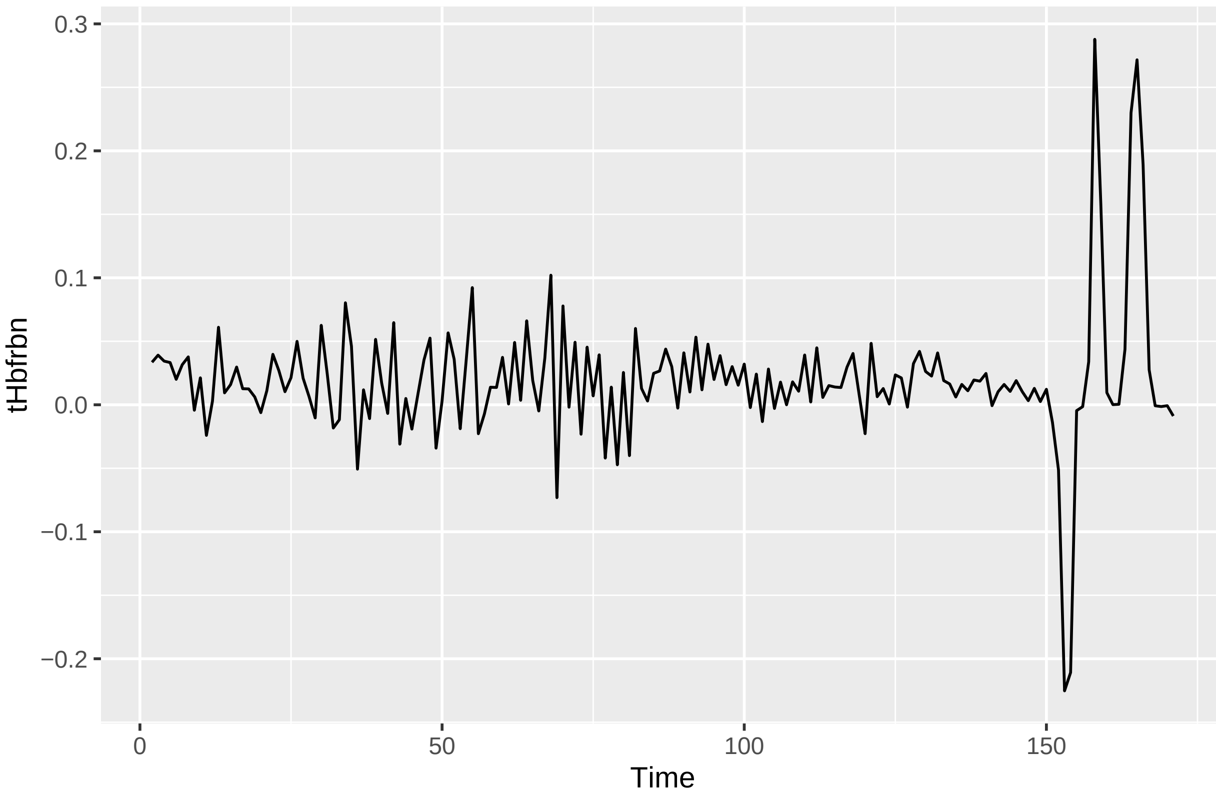
Taking the log of hbfrbn still shows an upward trend.

Plot: diff(hbfrbn)



Performing a diff(hbfrbn) might show somewhat a mean 0 and maybe some contstant variance for the first 35 years of data, but does not account for the last 7 years.

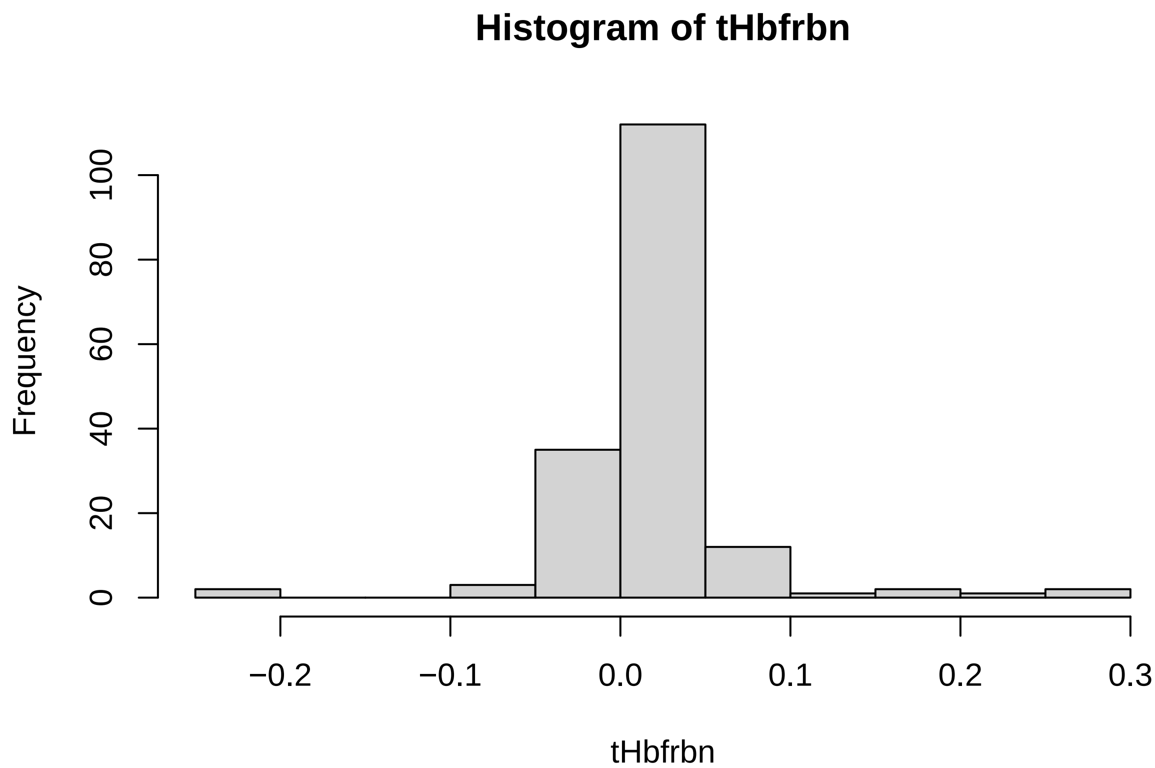
Plot: diff(log(hbfrbn))



Performing a diff(log(hbfrbn)) transformation displays a mean just above 0, and maybe some constant variance, despite outliers shown in the last 7 years of data. This might be the closest we can get the data to a normal Gaussian form.

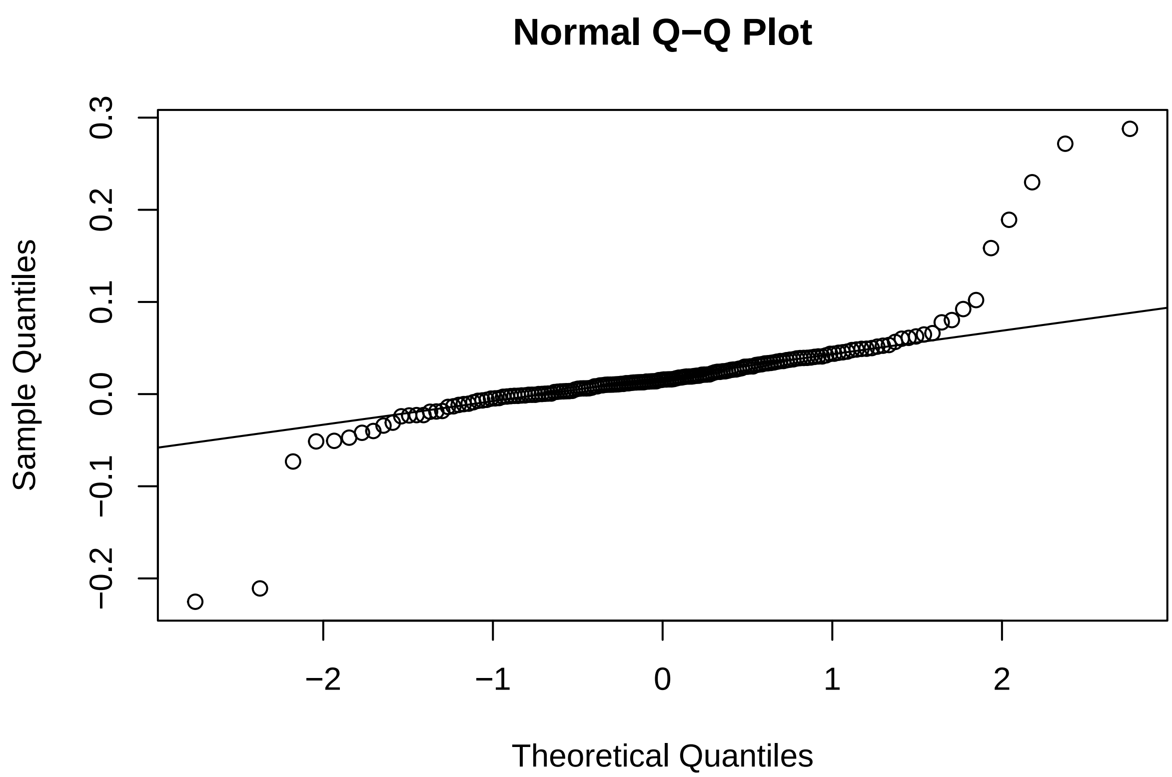
We will continue the EDA with the diff(log(hbfrbn)) transformation.

Histogram: diff(log(hbfrbn))



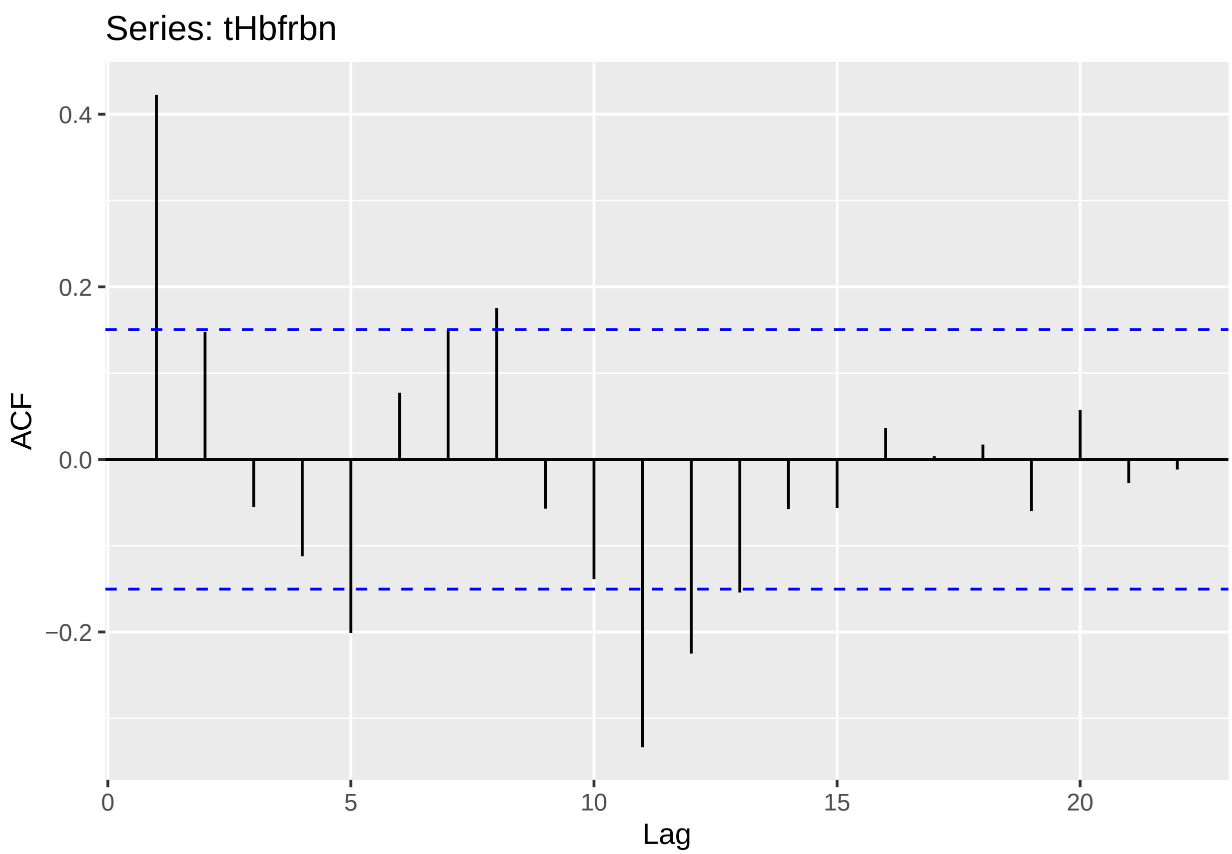
We can observe a slightly right-skewed, tall distribution, which does not conform to a normal Gaussian form.

Q-Q Plot: diff(log(hbfrbn))



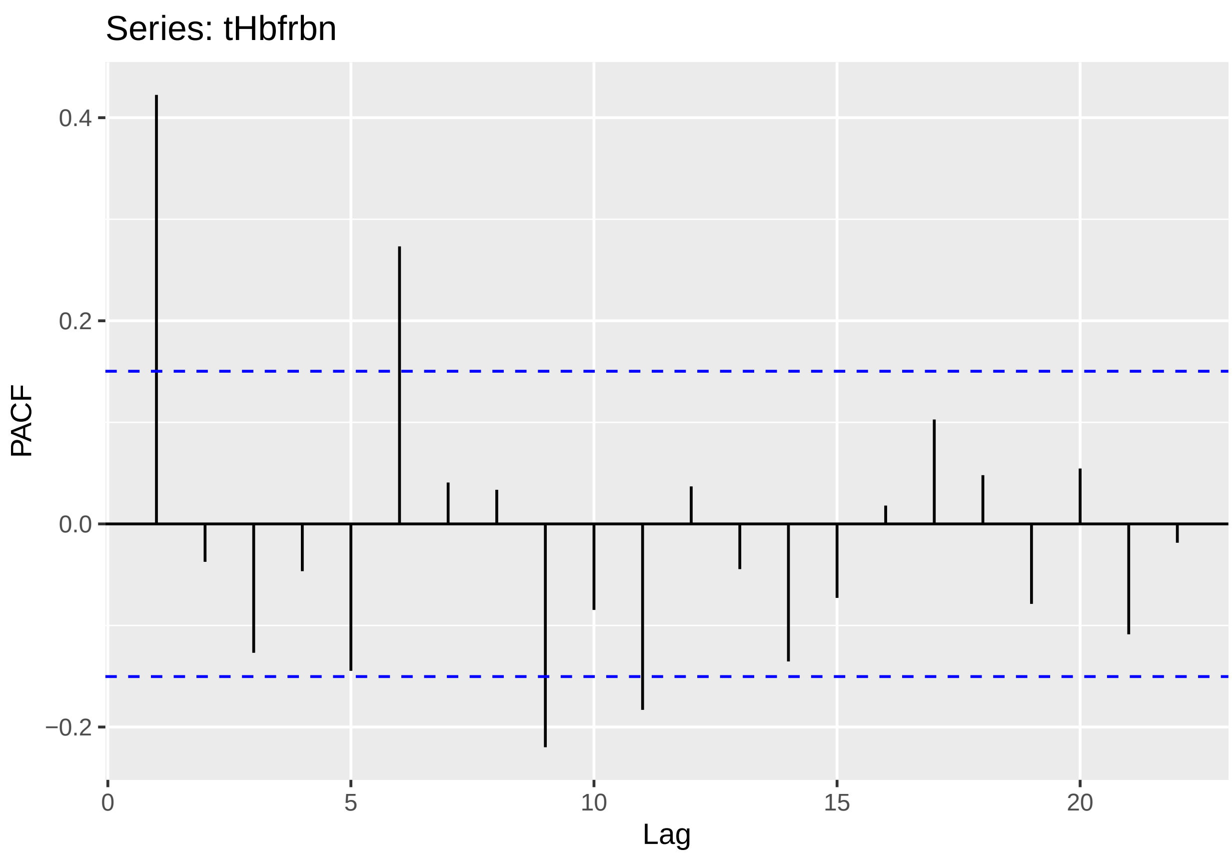
We observe very thick tails, indicating very tall Kurtosis, thus showing non-normalcy in respect to a Gaussian PDF. We can also observe right skewness due to the right tail looks thicker than the other.

ACF Plot: diff(log(hbfrbn))



We observe a fairly stationary ACF plot with some with some lags (5 out of 22) outside of the 95% confidence interval (CI) threshold. The max range of these exceeding lags are from -0.32 to 0.42, which could be considered relatively small.

PACF Plot: diff(log(hbfrbn))



We observe a fairly stationary PACF plot, and from it we can use a possible AR(1) or even AR(6) component.

T-Test for Mean 0: diff(log(hbfrbn))

```
##
##  One Sample t-test
##
## data:  data
## t = 4.9869, df = 169, p-value = 1.512e-06
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.01202558 0.02778505
## sample estimates:
##  mean of x
## 0.01990532
##
## T-Test: mean *NOT* statistically zero, linear trend present ->
## reject H0
```

```
## [1] FALSE
```

While the 95% confidence interval (CI) range is not 0, the lower and upper bounds is relatively close to zero, as well as the calculated mean. In this case, we would call the expected mean of diff(log(hbfrbn)) to be very close to zero, in some ways accepted to be zero, with almost no signs of a linear trend.

Skewness: diff(log(hbfrbn))

```
##      skew    lwr.ci    upr.ci
## 0.9848178 0.9339308 1.0879798
## Skew: has *RIGHT* skewness,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(hbfrbn)) has a distribution with right skewness, thus showing non-normalcy in respect to a Gaussian PDF.

(excess) Kurtosis: diff(log(hbfrbn))

```
##     kurt   lwr.ci   upr.ci
## 12.32073 12.36459 12.73997
## Kurt: has *TALL thick-tailed* (excess) kurtosis,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(hbfrbn)) has a distribution with tall (excess) Kurtosis, thus showing non-normalcy in respect to a Gaussian PDF.

Constant Variance: diff(log(hbfrbn))

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(data ~ seq(1, length(data)))
## BP = 12.379, df = 1, p-value = 0.0004341
##
## Breusch-Pagan: *NON*-constant variance, possible clustering,
## heteroscedastic -> reject H0
```

```
##    BP
## FALSE
```

Like what the plot shows, the Breusch-Pagan test confirms non-constant variance for diff(log(hbfin)).

Lag Dependency: diff(log(hbfrbn))

```
##
##  Box-Ljung test
##
## data:  data
## X-squared = 98.739, df = 30, p-value = 2.934e-09
##
## Box-Ljung: implies dependency present over 30 lags,
## autocorrelation present -> reject H0
```

```
## [1] FALSE
```

With a Box-Ljung test p-value < 0.05 for the diff(log(hbfin)) data, we observe lag dependency and thus serial autocorrelation.

Multivariate Normality: hbfin and hbfrbn

Let us test multivariate normality for hbfin and hbfrbn:

```
## $multivariateNormality
##              Test        Statistic              p value Result
## 1 Mardia Skewness 342.991869755551 5.71532167103068e-73     NO
## 2 Mardia Kurtosis  18.934513176286                    0     NO
## 3             MVN             <NA>                 <NA>     NO
##
## $univariateNormality
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling   hbfin     14.4332  <0.001      NO
## 2 Anderson-Darling  hbfrbn      8.8060  <0.001      NO
##
## $Descriptives
##          n      Mean   Std.Dev Median  Min    Max   25th    75th     Skew
## hbfin  171 1005.7579 1264.6363  502.0 12.4 5455.0 131.75 1263.75 1.822921
## hbfrbn 171  378.5731  346.1179  247.5 55.8 1664.7 120.35  517.80 1.893926
##        Kurtosis
## hbfin  2.771397
## hbfrbn 4.213803
```

The mvn test above shows that both hbfin and hbfrbn do not display multivariate or univariate normality, due to skewness and kurtosis for their respective data is no 0.

Multivariate Normality: diff(log(hbfin)) and diff(log(hbfrbn))

Let us test multivariate normality for diff(log(hbfin)) and diff(log(hbfrbn)).

```
## $multivariateNormality
##              Test        Statistic              p value Result
## 1 Mardia Skewness 56.9138236880574 1.28975562063145e-11     NO
## 2 Mardia Kurtosis 26.1692256527488                    0     NO
## 3             MVN             <NA>                 <NA>     NO
##
## $univariateNormality
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling   hbfin      3.5026  <0.001      NO
## 2 Anderson-Darling  hbfrbn     12.1673  <0.001      NO
##
## $Descriptives
##          n       Mean    Std.Dev     Median        Min       Max         25th
## hbfin  170 0.03580348 0.05239707 0.03170498 -0.1564128 0.2662271 0.0086420144
## hbfrbn 170 0.01990532 0.05204355 0.01562672 -0.2253063 0.2878344 0.0005951675
##              75th      Skew  Kurtosis
## hbfin  0.05595773 0.9540442  4.724453
## hbfrbn 0.03500505 0.9848178 12.320730
```

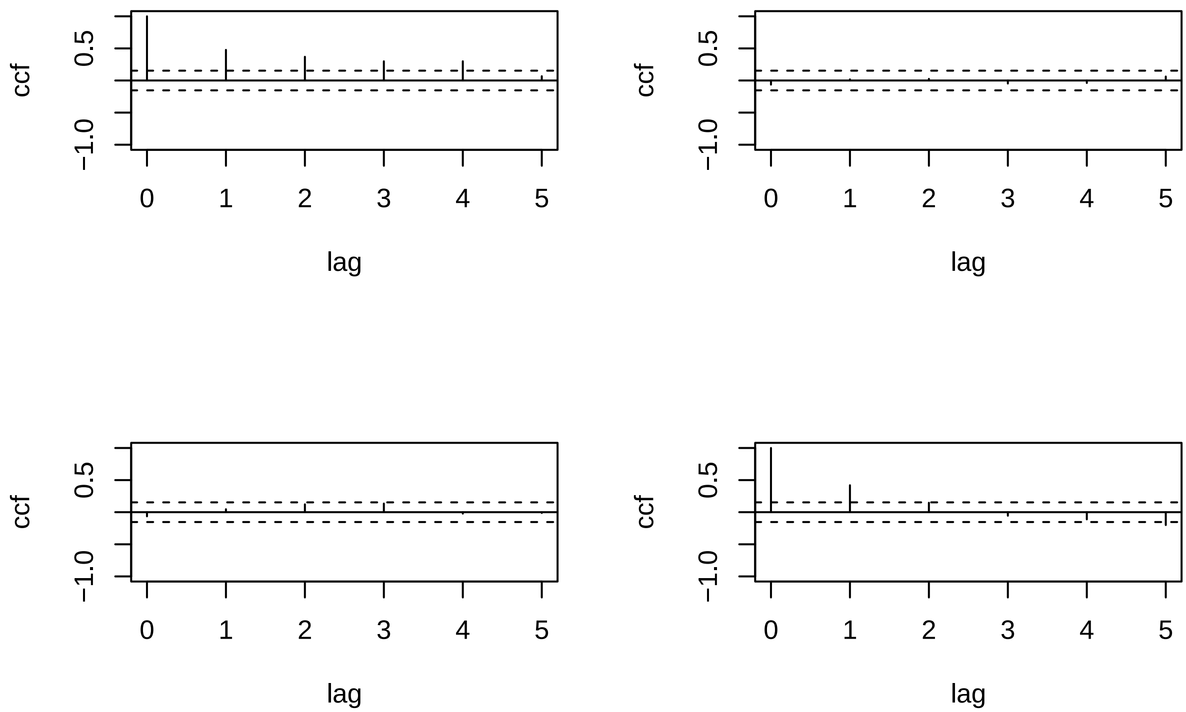
The mvn test above shows that both diff(log(hbfin)) and diff(log(hbfrbn)) do not display multivariate or univariate normality, due to skewness and kurtosis for their respective data is not 0.

Despite the strict non-normality, the means of diff(log(hbfin)) and diff(log(hbfrbn)) would be considered just above 0, not perfect but also not terrible to use for VAR() modelling.

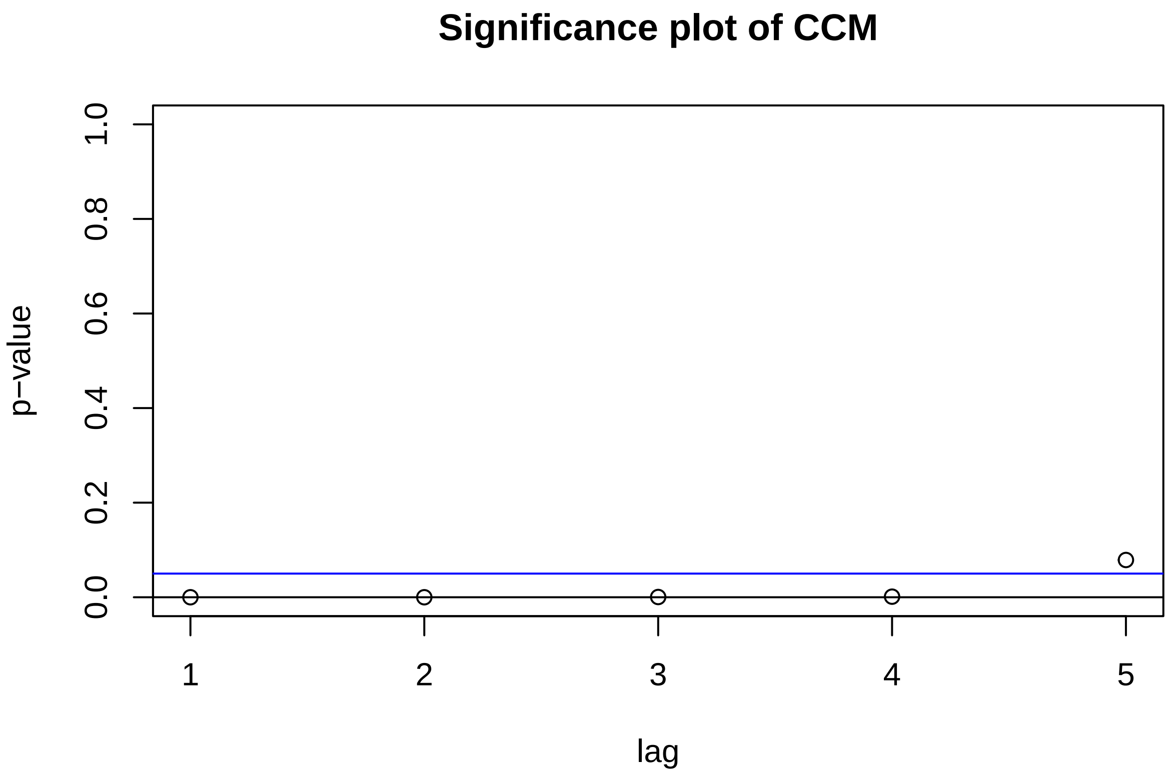
1.2. Obtain the first 5 lags of sample cross-correlation matrices of the z_{it}.

Let us run Cross-Correlation Matrices, or ccm(), on the z_{it} data:

```
## [1] "Covariance matrix:"
##            hbfin    hbfrbn
## hbfin   0.002745 -0.000175
## hbfrbn -0.000175  0.002709
## CCM at lag:  0
##         [,1]    [,2]
## [1,]  1.0000 -0.0642
## [2,] -0.0642  1.0000
## Simplified matrix:
## CCM at lag:  1
## + .
## . +
## CCM at lag:  2
## + .
## . .
## CCM at lag:  3
## + .
## . .
## CCM at lag:  4
## + .
## . .
## CCM at lag:  5
## . .
## . -
```




```
## Hit Enter for p-value plot of individual ccm:
```



While we see positive autocorrelation on the first 4 lags of diff(log(hbfin)) and on the 1st and and 5th lag of diff(log(hbfrbn)), the plots above indicate no significant cross-correlation between the two sets of data.

Let's use the Li-McLeod test to test for multivariate ARCH effects:

```
## Q(m) of squared series(LM test):
## Test statistic:  49.5067  p-value:  1.748288e-09
## Rank-based Test:
## Test statistic:  46.51283  p-value:  7.140752e-09
## Q_k(m) of squared series:
## Test statistic:  201.4092  p-value:  0
## Robust Test(5%) :  75.32885  p-value:  2.401036e-08
```

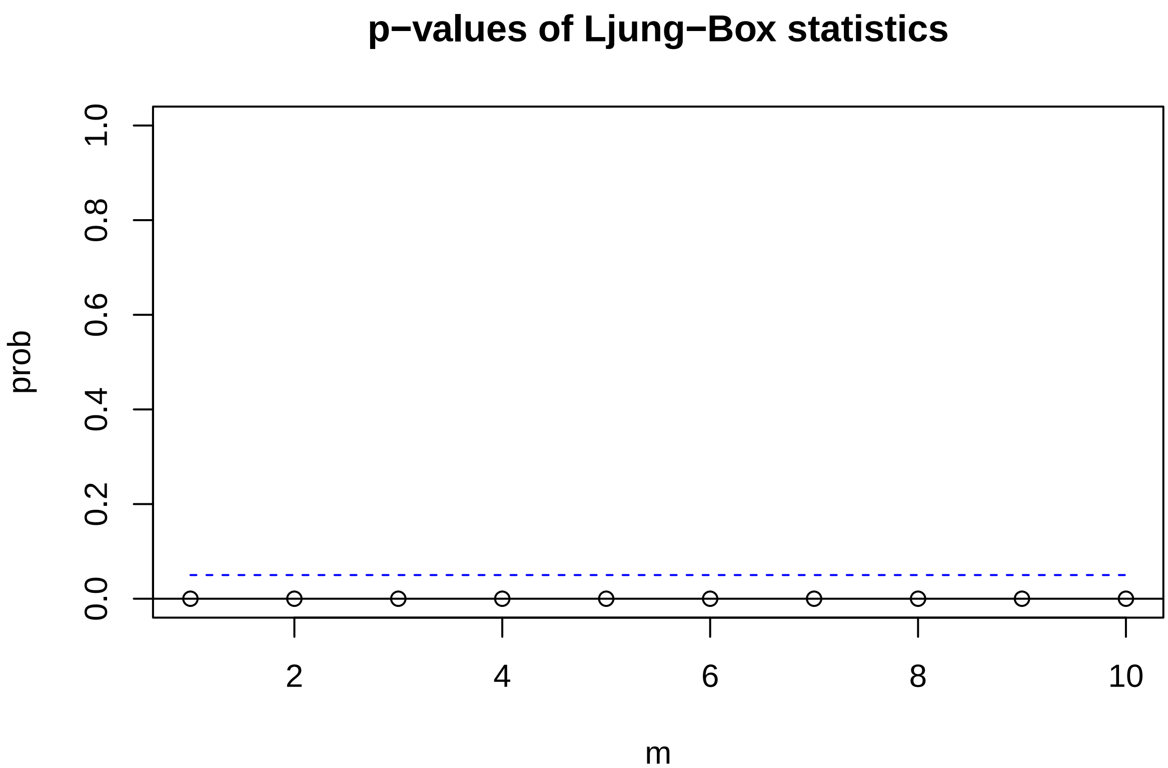
With the Li-McLeod test p-value < 0.05, we can say the diff(log(hbfin)) and diff(log(hbfrbn)) multivariate data contains ARCH effects.

1.3. Test $H0 : \rho_1 = ... = \rho_{10} = 0$ versus $H_a : \rho_j \neq 0$ for some j, where $j \in \{1, ..., 10\}$. Draw a conclusion using the 5% significance level.

We will run an mq() test on the diff(log(hbfin)) and diff(log(hbfrbn)) multivariate data:

```
## Ljung-Box Statistics:
##         m       Q(m)     df    p-value
##  [1,]   1.0      72.5     4.0        0
##  [2,]   2.0     104.8     8.0        0
```

```
##  [3,]   3.0     125.2    12.0         0
##  [4,]   4.0     143.4    16.0         0
##  [5,]   5.0     151.9    20.0         0
##  [6,]   6.0     154.6    24.0         0
##  [7,]   7.0     159.8    28.0         0
##  [8,]   8.0     165.6    32.0         0
##  [9,]   9.0     168.6    36.0         0
## [10,]  10.0     178.1    40.0         0
```



All lags with a multivariate Ljung-Box test p-value < 0.05 in the above plot exhibit serial cross-correlation.

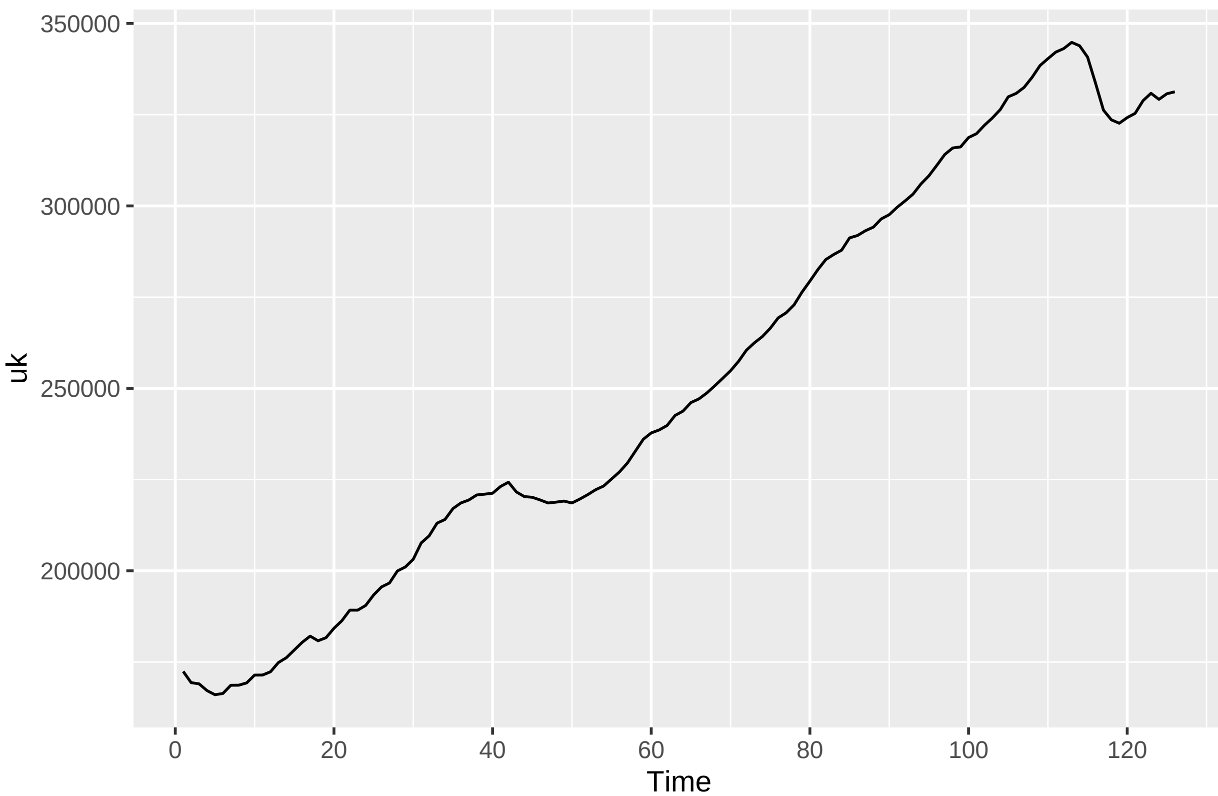
2. GDP (30 points)

Consider the growth rates, in percentages, of the quarterly real GDP of United Kingdom, Canada, and the United States located in the object **qgdp** in the **MTS R** package.

2.1. Use EDA to justify a VAR(4) model.

EDA: UK

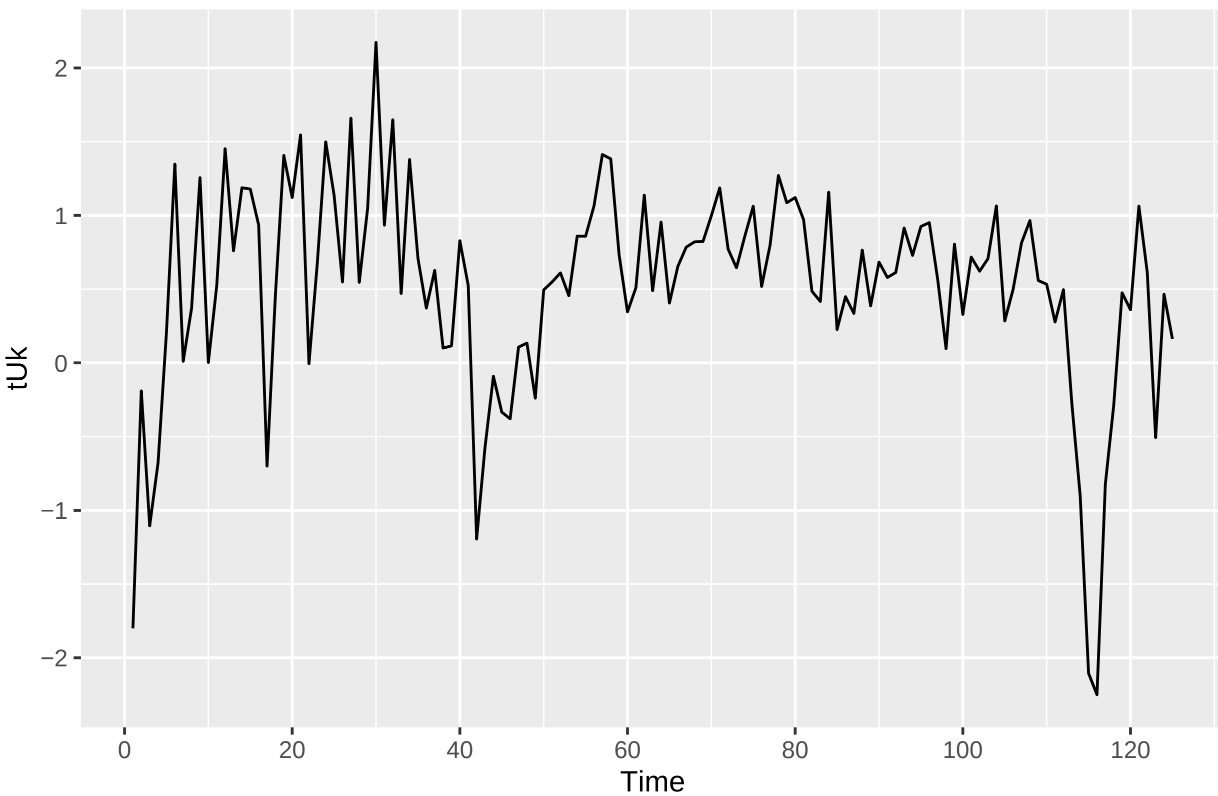
Time series plot: uk



Plotting the uk data shows a general upward trend. Because of this, the data does not show a mean 0 nor constant variance, and we'll need to transform the data to make it fit a proper VAR-based model.

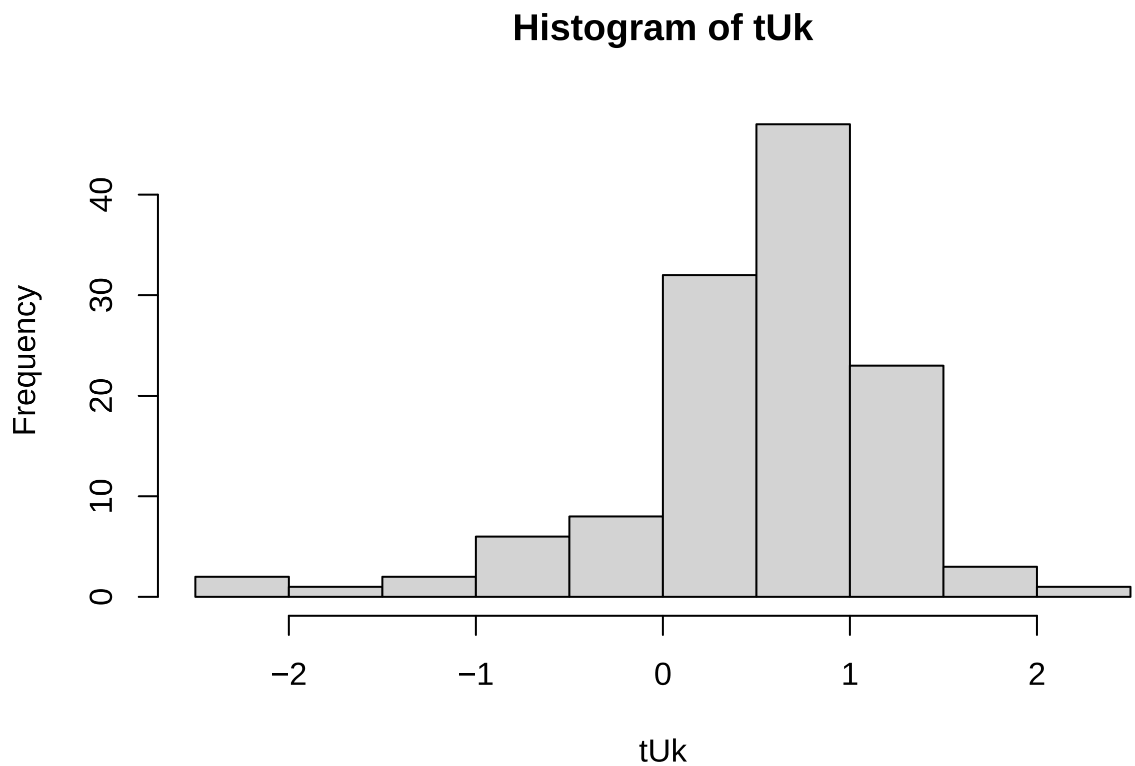
We will transform the uk data to diff(log(uk)) and perform an EDA.

Time series plot: diff(log(uk))



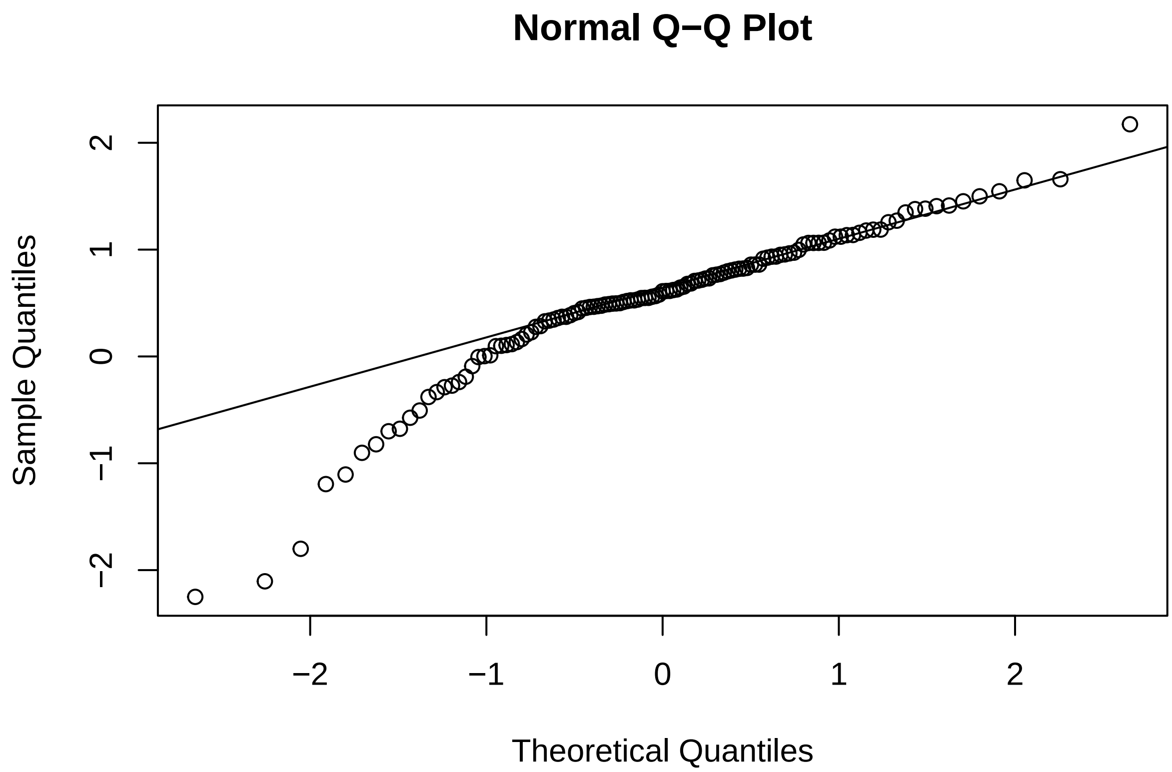
We get closer to a mean 0 using diff(log(uk)) but looks to be just above 0. The plot might not fully show constant variance but it's better than the upward trend of just using the original uk data.

Histogram: diff(log(uk))



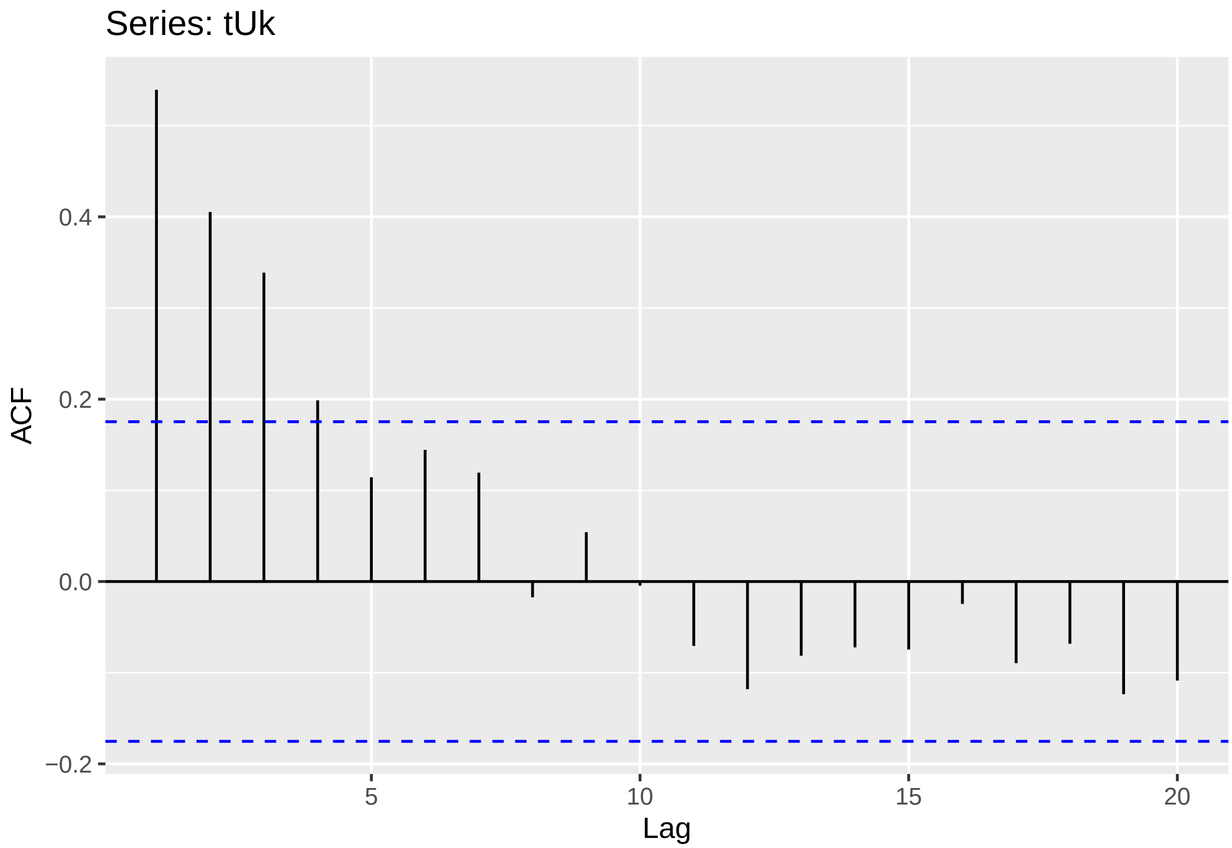
We see a tall left-skewed distribution, which does not conform to a normal Gaussian PDF.

Q-Q Plot: diff(log(uk))



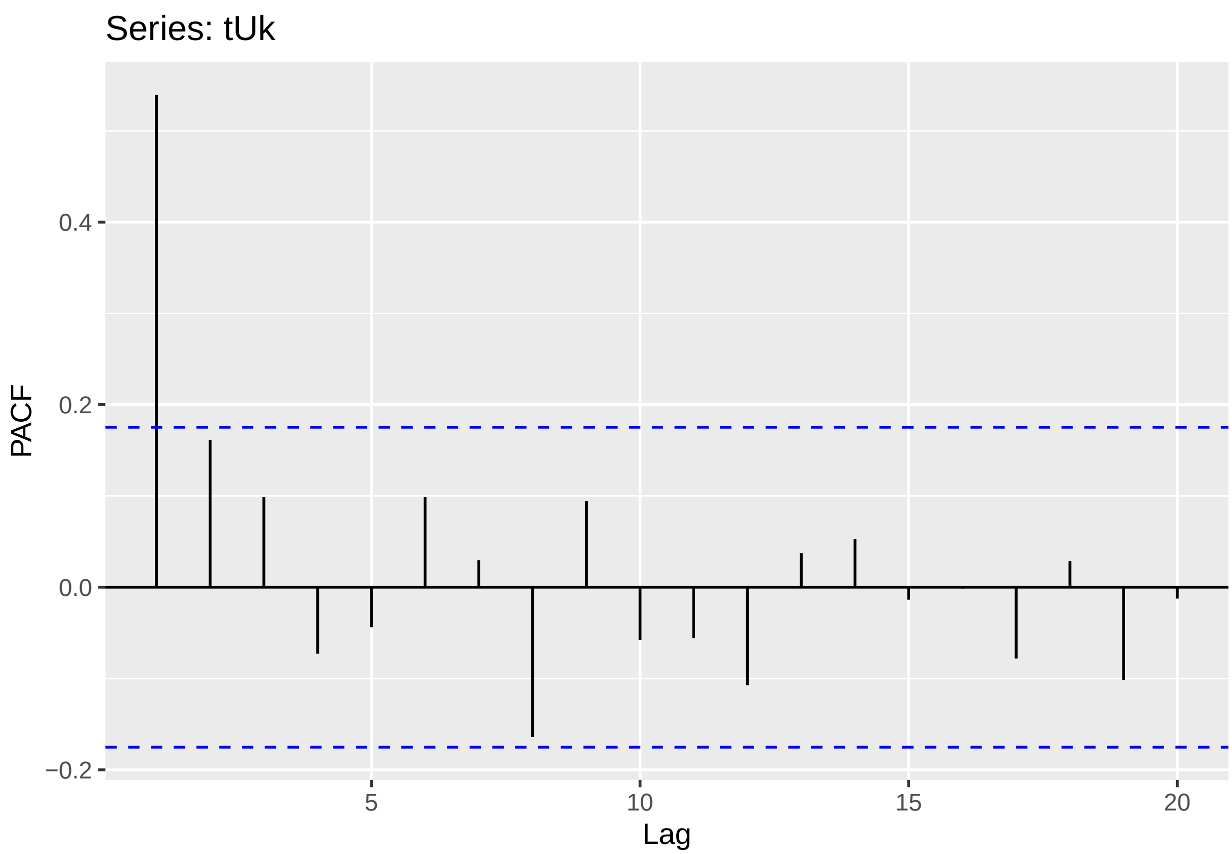
We observe a very thick left tail, indicating very tall Kurtosis, thus showing non-normalcy in respect to a Gaussian PDF. We can also observe left skewness due to the left tail looks much thicker.

ACF Plot: diff(log(uk))



We observe a fairly stationary ACF plot.

PACF Plot: diff(log(uk))



We observe a fairly stationary PACF plot, and from it we can use a possible AR(1) component.

T-Test for Mean 0: diff(log(uk))

```
##
##  One Sample t-test
##
## data:  data
## t = 8.2405, df = 124, p-value = 2.063e-13
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.3968563 0.6477620
## sample estimates:
## mean of x
## 0.5223092
##
## T-Test: mean *NOT* statistically zero, linear trend present ->
## reject H0
```

```
## [1] FALSE
```

The 95% CI might not contain 0 but the range is between 0.40 and 0.65, which is not far from zero.

Skewness: diff(log(uk))

```
##       skew    lwr.ci    upr.ci
```

```
## -1.319324 -1.410953 -1.372417
## Skew: has *LEFT* skewness,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(uk)) has a distribution with left skewness, thus showing non-normalcy in respect to a Gaussian PDF.

(excess) Kurtosis: diff(log(uk))

```
##     kurt   lwr.ci   upr.ci
## 2.966034 3.025796 3.167846
## Kurt: has *TALL thick-tailed* (excess) kurtosis,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(uk)) has a distribution with tall (excess) Kurtosis, thus showing non-normalcy in respect to a Gaussian PDF.

Constant Variance: diff(log(uk))

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(data ~ seq(1, length(data)))
## BP = 0.53897, df = 1, p-value = 0.4629
##
## Breusch-Pagan: constant variance, homoscedastic ->
## *FAIL* to reject H0
```

```
##   BP
## TRUE
```

Like what the plot shows, the Breusch-Pagan test confirms non-constant variance for diff(log(uk)).

Lag Dependency: diff(log(uk))

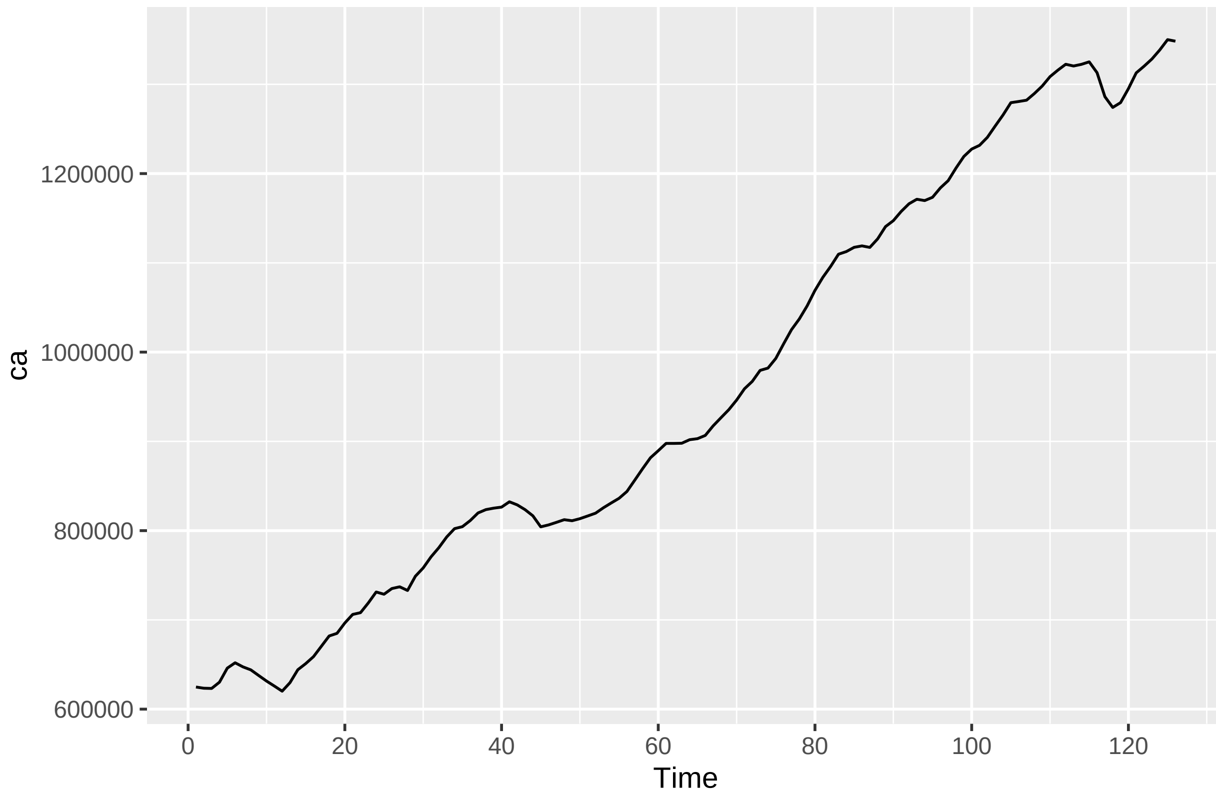
```
##
##  Box-Ljung test
##
## data:  data
## X-squared = 110.25, df = 30, p-value = 4.193e-11
##
## Box-Ljung: implies dependency present over 30 lags,
## autocorrelation present -> reject H0
```

```
## [1] FALSE
```

With a Box-Ljung test p-value < 0.05 for the diff(log(uk)) data, we observe lag dependency and thus serial autocorrelation.

EDA: CA

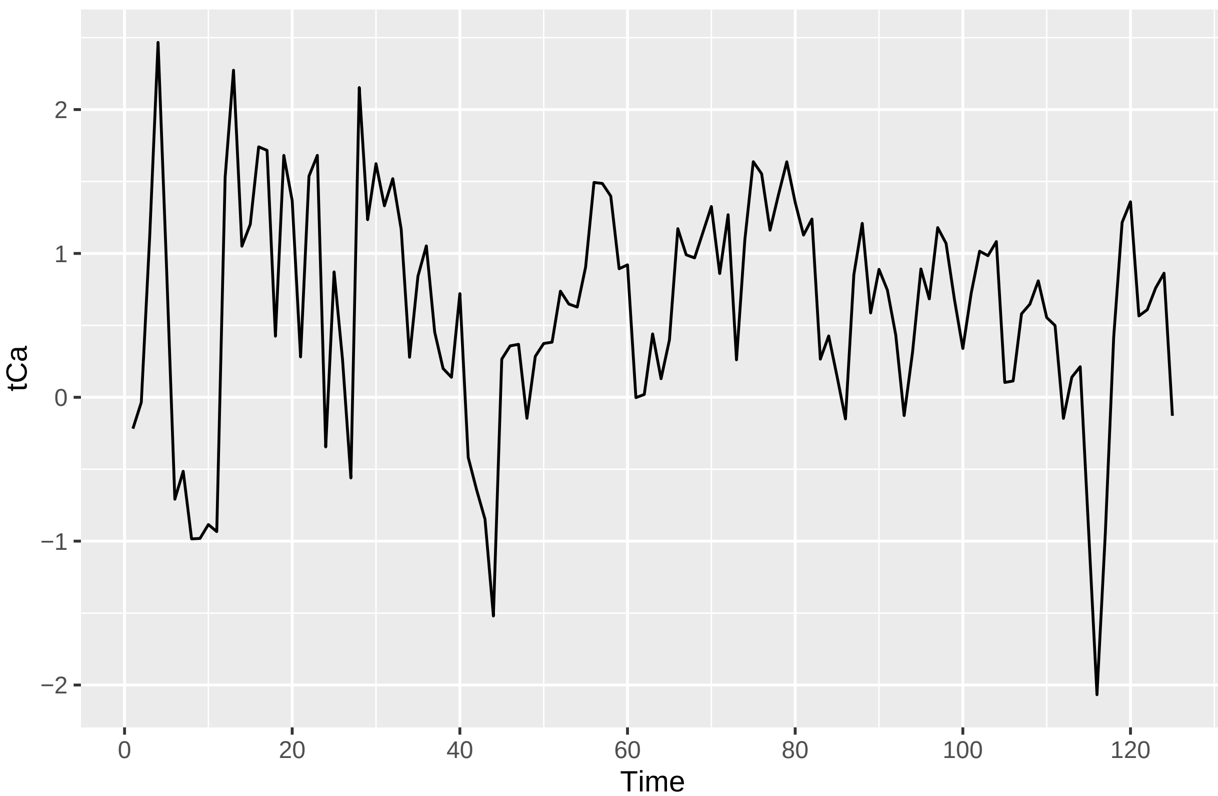
Time series plot: ca



Plotting the ca data shows a general upward trend. Because of this, the data does not show a mean 0 nor constant variance, and we'll need to transform the data to make it fit a proper VAR-based model.

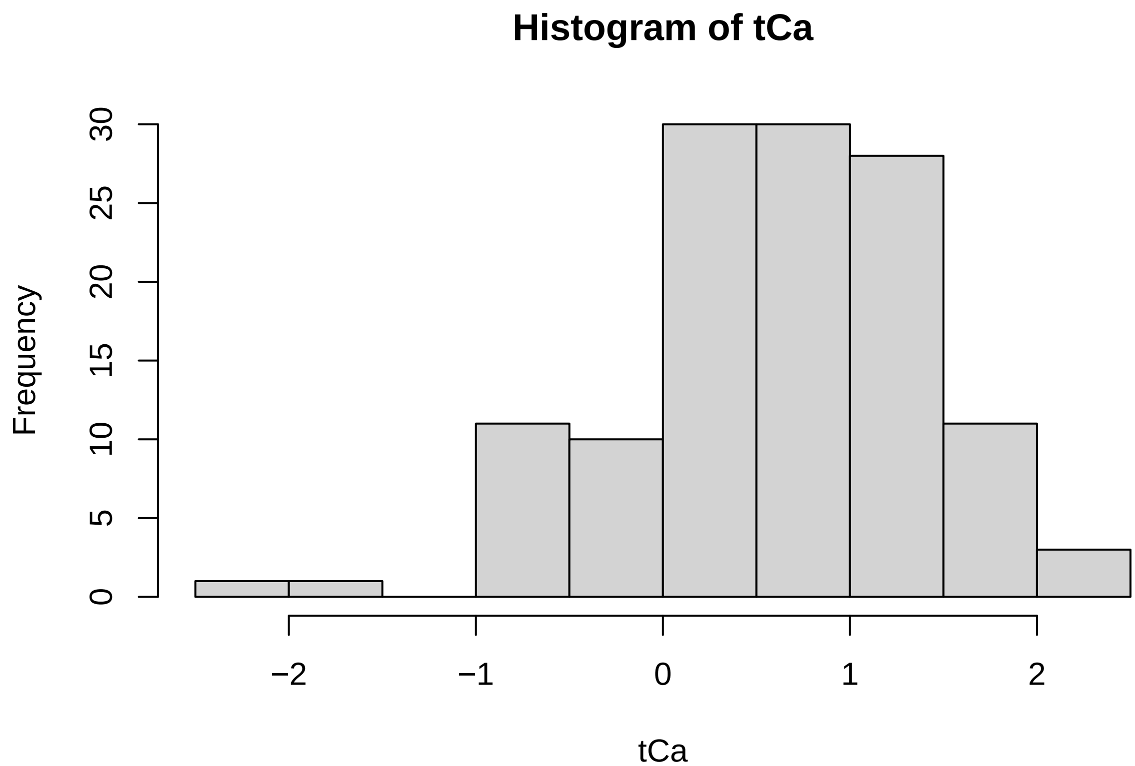
We will transform the ca data to diff(log(ca)) and perform an EDA.

Time series plot: diff(log(ca))



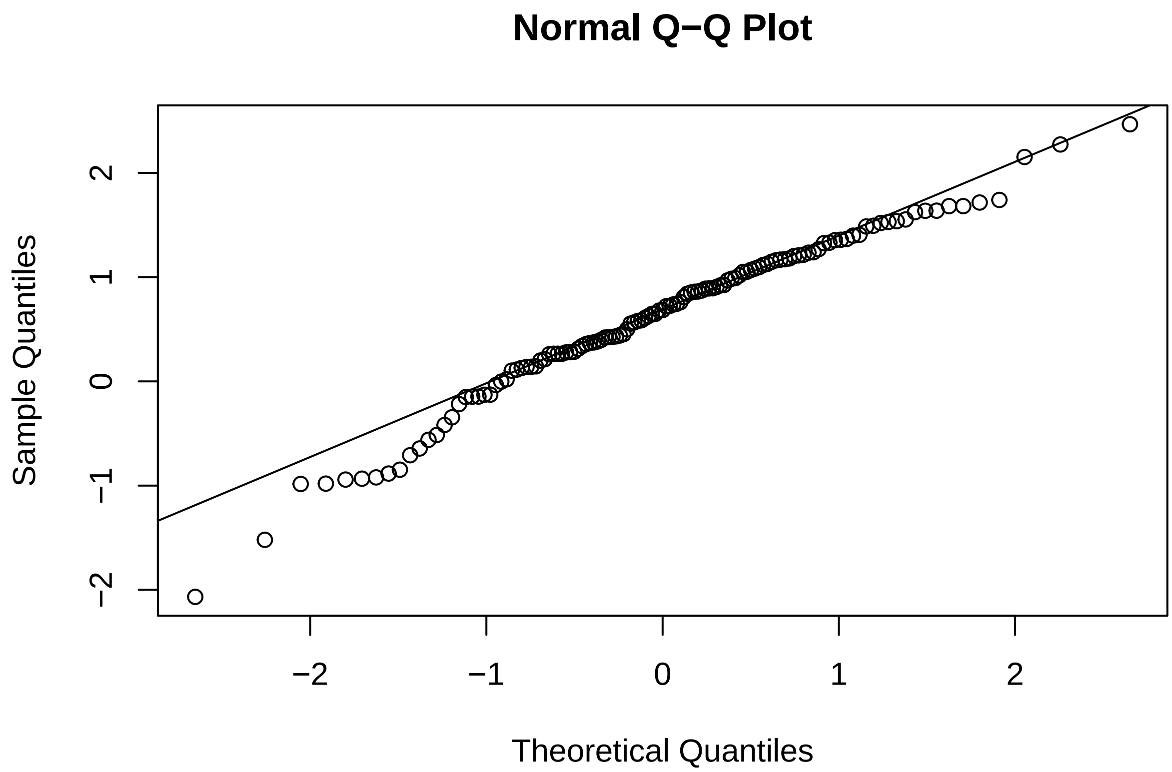
We get closer to a mean 0 using diff(log(ca)) but looks to be just above 0. The plot might not fully show constant variance but it's better than the upward trend of just using the original ca data.

Histogram: diff(log(ca))



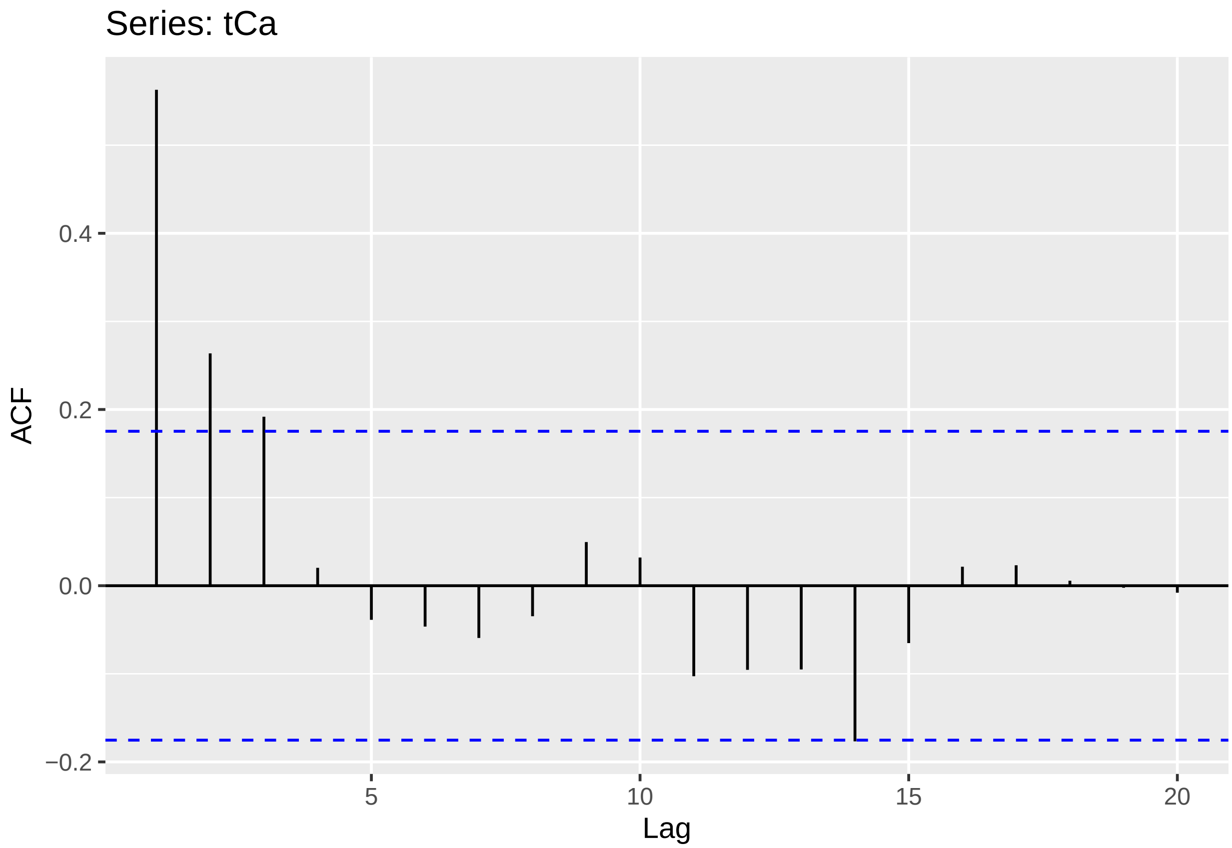
We see a tall left-skewed distribution, which does not conform to a normal Gaussian PDF.

Q-Q Plot: diff(log(ca))



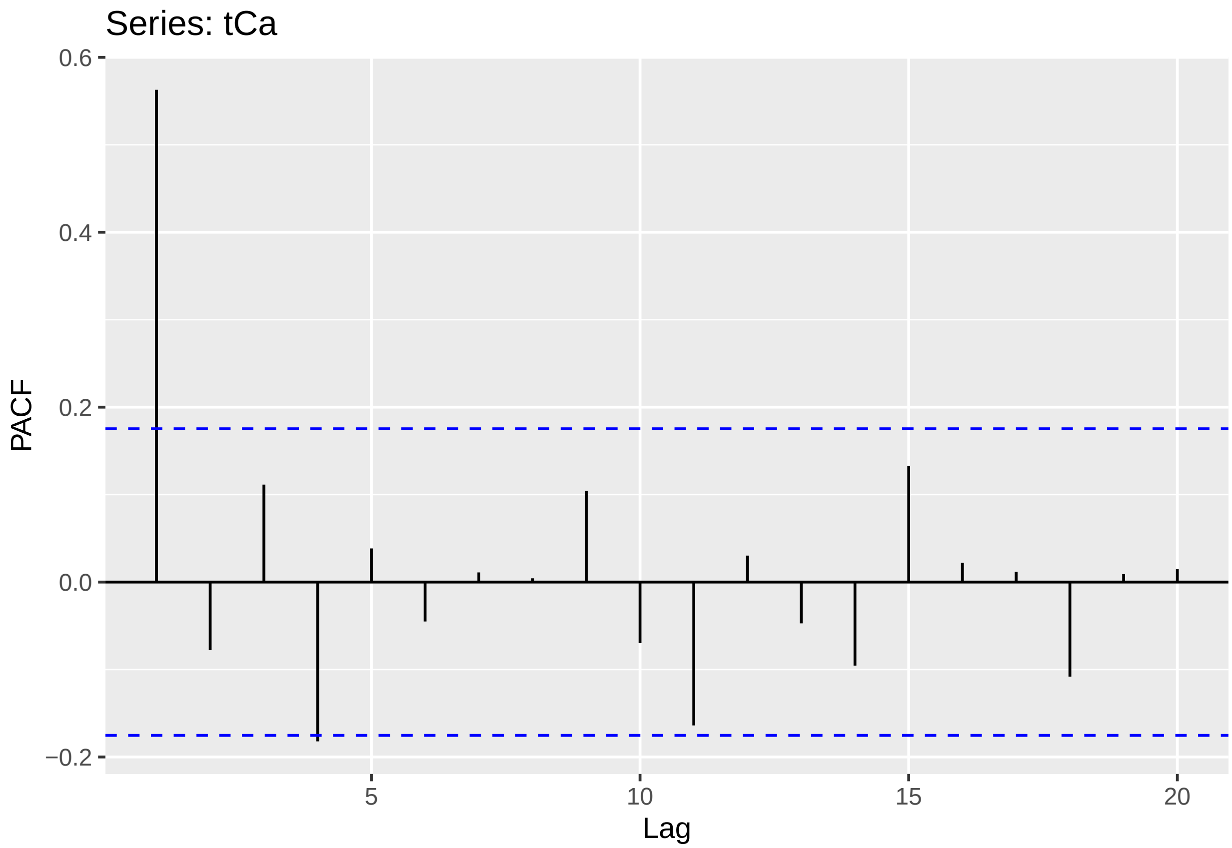
We observe a fairly straight Q-Q plot with most values lying along the ideal normal line, indicating a decent goodness-of-fit. The left tail does veer down with some outliers.

ACF Plot: diff(log(ca))



We observe a fairly stationary ACF plot.

PACF Plot: diff(log(ca))



We observe a fairly stationary PACF plot, and from it we can use a possible AR(1) component.

T-Test for Mean 0: diff(log(ca))

```
##
##  One Sample t-test
##
## data:  data
## t = 8.7622, df = 124, p-value = 1.207e-14
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.4763623 0.7543721
## sample estimates:
## mean of x
## 0.6153672
##
## T-Test: mean *NOT* statistically zero, linear trend present ->
## reject H0
```

```
## [1] FALSE
```

The 95% CI might not contain 0 but the range is between 0.48 and 0.75, which is not far from zero.

Skewness: diff(log(ca))

```
##       skew     lwr.ci     upr.ci
```

```
## -0.5829055 -0.6124209 -0.5843862
## Skew: has *LEFT* skewness,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(ca)) has a distribution with left skewness, thus showing non-normalcy in respect to a Gaussian PDF.

(excess) Kurtosis: diff(log(ca))

```
##      kurt    lwr.ci    upr.ci
## 0.5482180 0.6120922 0.6807891
## Kurt: has *TALL thick-tailed* (excess) kurtosis,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(ca)) has a distribution with tall (excess) Kurtosis, thus showing non-normalcy in respect to a Gaussian PDF.

Constant Variance: diff(log(ca))

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(data ~ seq(1, length(data)))
## BP = 6.8854, df = 1, p-value = 0.00869
##
## Breusch-Pagan: *NON*-constant variance, possible clustering,
## heteroscedastic -> reject H0
```

```
##    BP
## FALSE
```

Like what the plot shows, the Breusch-Pagan test confirms non-constant variance for diff(log(ca)).

Lag Dependency: diff(log(ca))

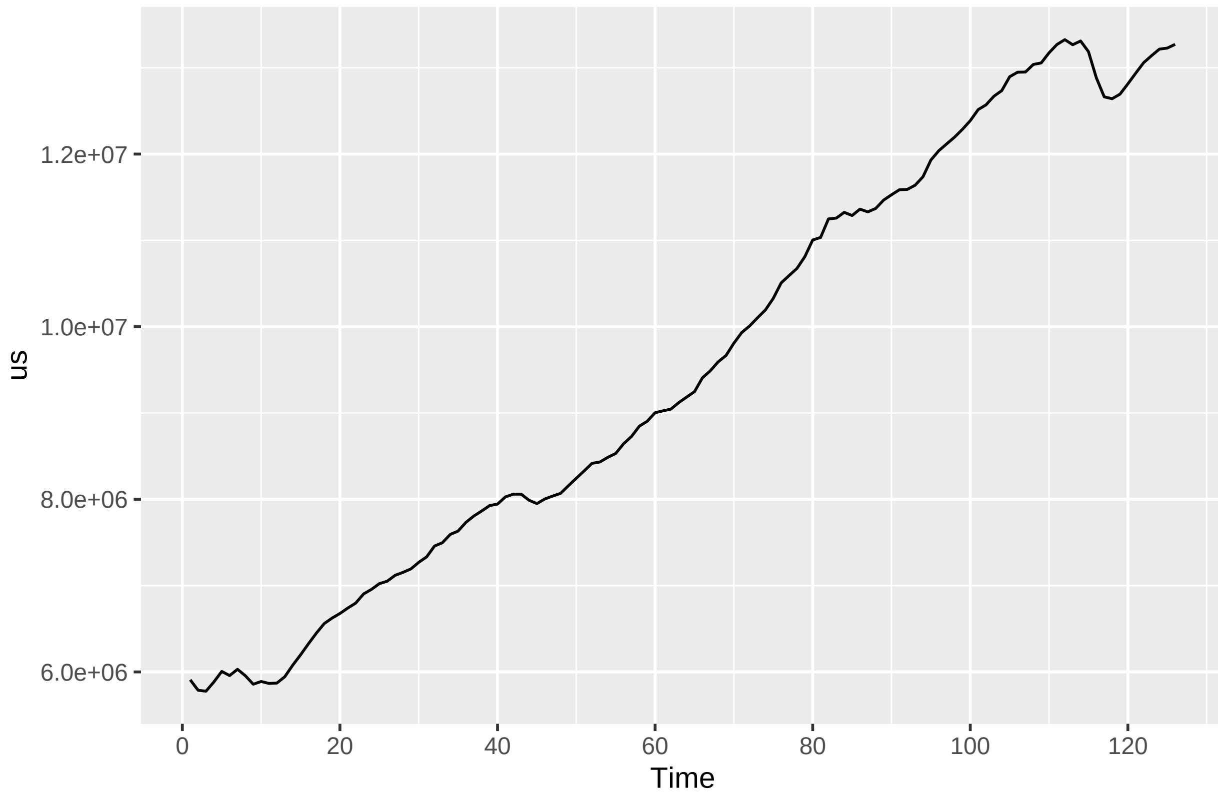
```
##
##  Box-Ljung test
##
## data:  data
## X-squared = 79.478, df = 30, p-value = 2.348e-06
##
## Box-Ljung: implies dependency present over 30 lags,
## autocorrelation present -> reject H0
```

```
## [1] FALSE
```

With a Box-Ljung test p-value < 0.05 for the diff(log(ca)) data, we observe lag dependency and thus serial autocorrelation.

EDA: US

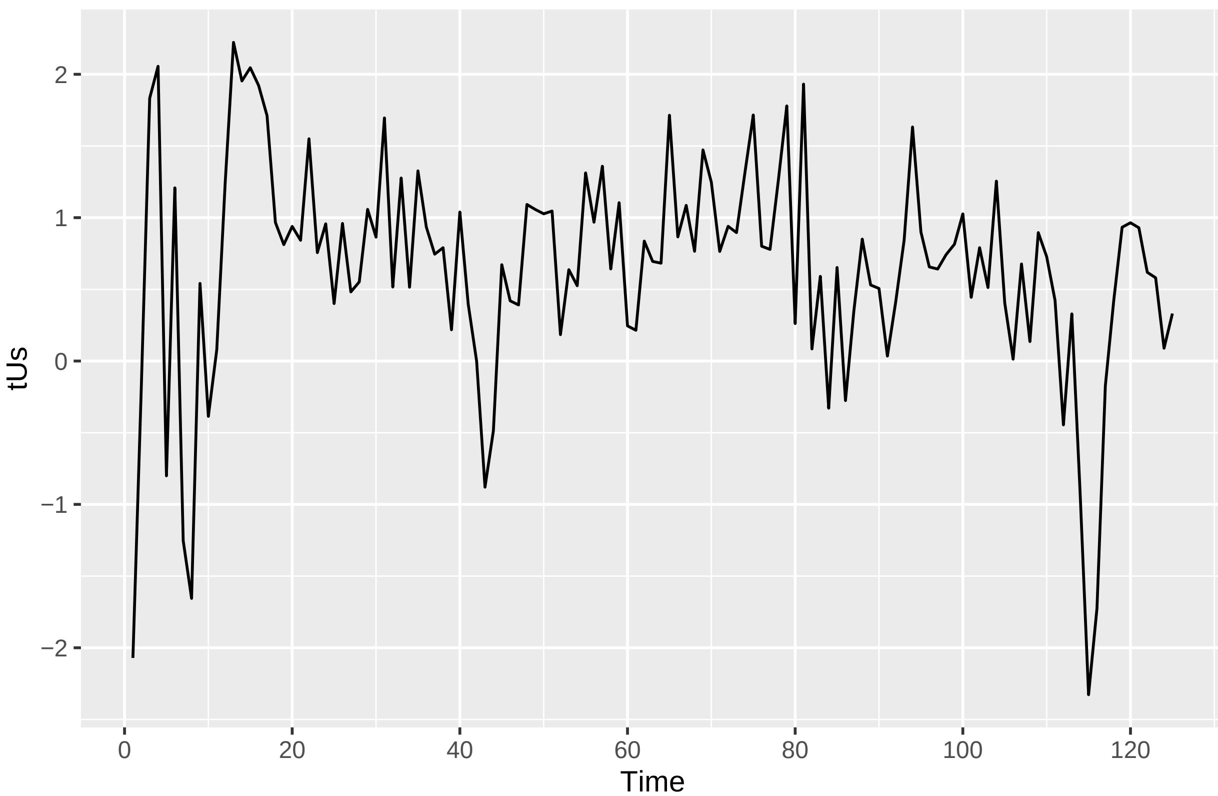
Time series plot: us



Plotting the ca data shows a general upward trend. Because of this, the data does not show a mean 0 nor constant variance, and we'll need to transform the data to make it fit a proper VAR-based model.

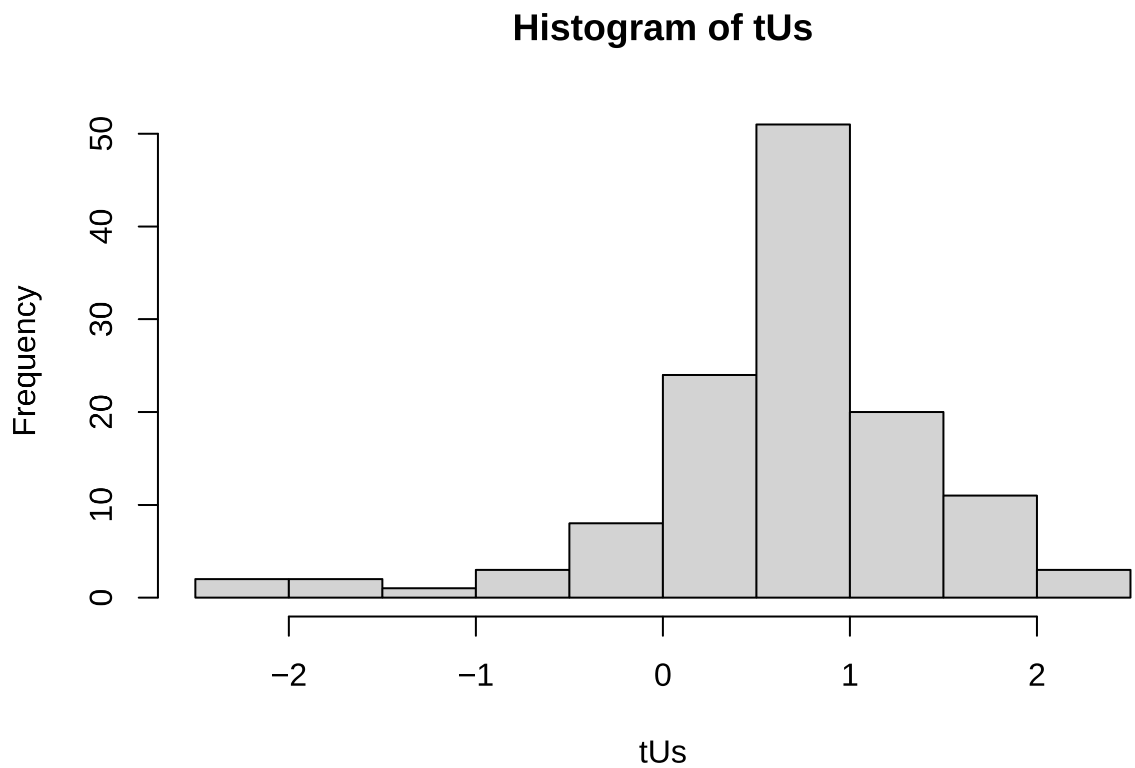
We will transform the us data to diff(log(us)) and perform an EDA.

Time series plot: diff(log(us))



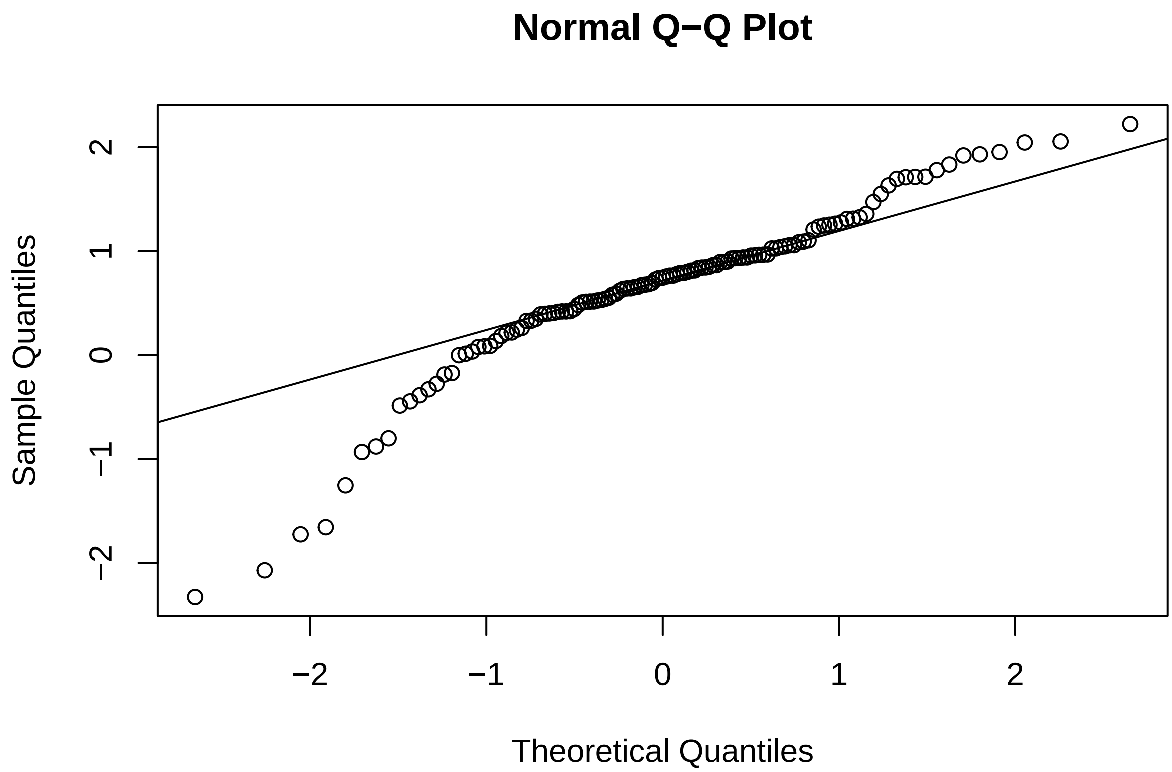
We get closer to a mean 0 using diff(log(us)) but looks to be just above 0. The plot might possibly exhibit constant variance but it's better than the upward trend of just using the original us data.

Histogram: diff(log(us))



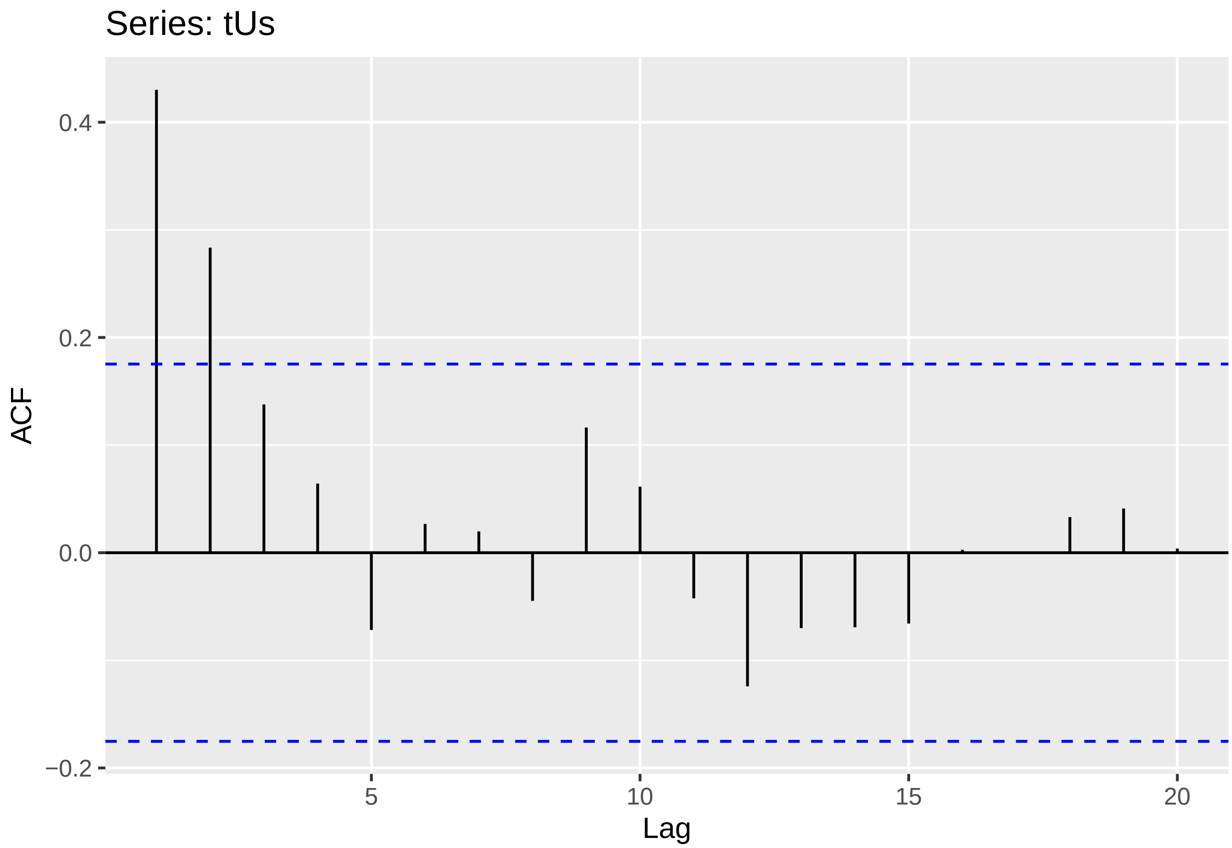
We see a tall left-skewed distribution, which does not conform to a normal Gaussian PDF.

Q-Q Plot: diff(log(us))



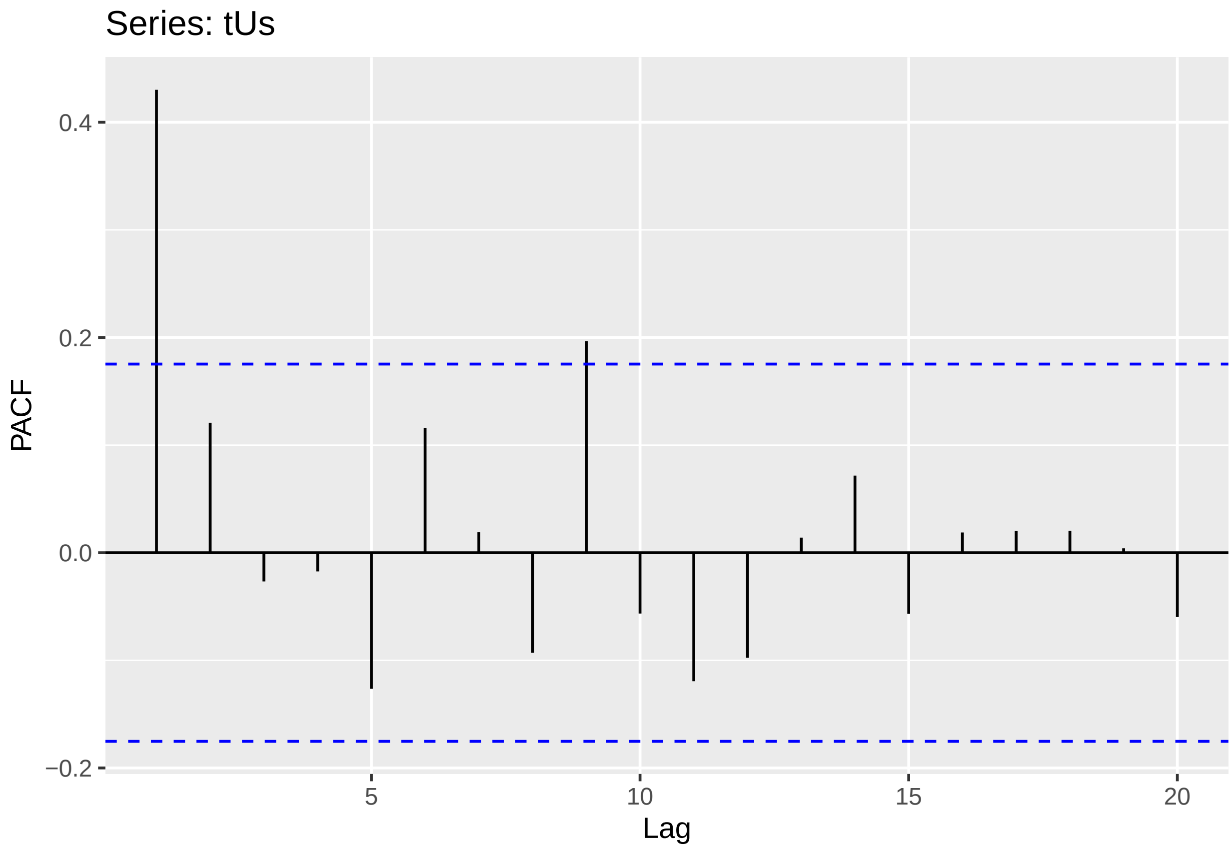
We observe a very thick left downward tail in the Q-Q plot, showing tall left skewness.

ACF Plot: diff(log(us))



We observe a fairly stationary ACF plot.

PACF Plot: diff(log(us))



We observe a fairly stationary PACF plot, and from it we can use a possible AR(1) component.

T-Test for Mean 0: diff(log(us))

```
##
##  One Sample t-test
##
## data:  data
## t = 9.1937, df = 124, p-value = 1.122e-15
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.5080237 0.7867755
## sample estimates:
## mean of x
## 0.6473996
##
## T-Test: mean *NOT* statistically zero, linear trend present ->
## reject H0
```

```
## [1] FALSE
```

The 95% CI might not contain 0 but the range is between 0.51 and 0.77, which is not far from zero.

Skewness: diff(log(us))

```
##       skew    lwr.ci    upr.ci
```

```
## -1.123478 -1.193105 -1.159997
## Skew: has *LEFT* skewness,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(us)) has a distribution with left skewness, thus showing non-normalcy in respect to a Gaussian PDF.

(excess) Kurtosis: diff(log(us))

```
##     kurt   lwr.ci   upr.ci
## 2.512260 2.598172 2.711739
## Kurt: has *TALL thick-tailed* (excess) kurtosis,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

diff(log(us)) has a distribution with tall (excess) Kurtosis, thus showing non-normalcy in respect to a Gaussian PDF.

Constant Variance: diff(log(us))

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(data ~ seq(1, length(data)))
## BP = 3.4806, df = 1, p-value = 0.06209
##
## Breusch-Pagan: constant variance, homoscedastic ->
## *FAIL* to reject H0
```

```
##   BP
## TRUE
```

The Breusch-Pagan test confirms constant variance for diff(log(us)).

Lag Dependency: diff(log(us))

```
##
##  Box-Ljung test
##
## data:  data
## X-squared = 50.267, df = 30, p-value = 0.01163
##
## Box-Ljung: implies dependency present over 30 lags,
## autocorrelation present -> reject H0
```

```
## [1] FALSE
```

With a Box-Ljung test p-value < 0.05 for the diff(log(us)) data, we observe lag dependency and thus serial autocorrelation.

Multivariate Normality: diff(log(uk)), diff(log(ca)), diff(log(us))

Let us test multivariate normality for diff(log(uk)), diff(log(ca)), and diff(log(us)):

```
## $multivariateNormality
##              Test        Statistic              p value Result
## 1 Mardia Skewness 86.9030579841158 2.19752328686866e-14     NO
## 2 Mardia Kurtosis 9.41120617972249                    0     NO
## 3             MVN             <NA>                 <NA>     NO
##
## $univariateNormality
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling    uk        3.0651  <0.001      NO
## 2 Anderson-Darling    ca        0.8930  0.0219      NO
## 3 Anderson-Darling    us        2.7748  <0.001      NO
##
## $Descriptives
##      n      Mean   Std.Dev    Median       Min      Max      25th      75th
## uk 125 0.5223092 0.7086442 0.6102259 -2.250303 2.173321 0.3285803 0.9510644
## ca 125 0.6153672 0.7851955 0.6839232 -2.067693 2.467191 0.2127979 1.1680803
## us 125 0.6473996 0.7872912 0.7453007 -2.327649 2.222757 0.3965866 1.0393021
##          Skew Kurtosis
## uk -1.3193243 2.966034
## ca -0.5829055 0.548218
## us -1.1234784 2.512260
```

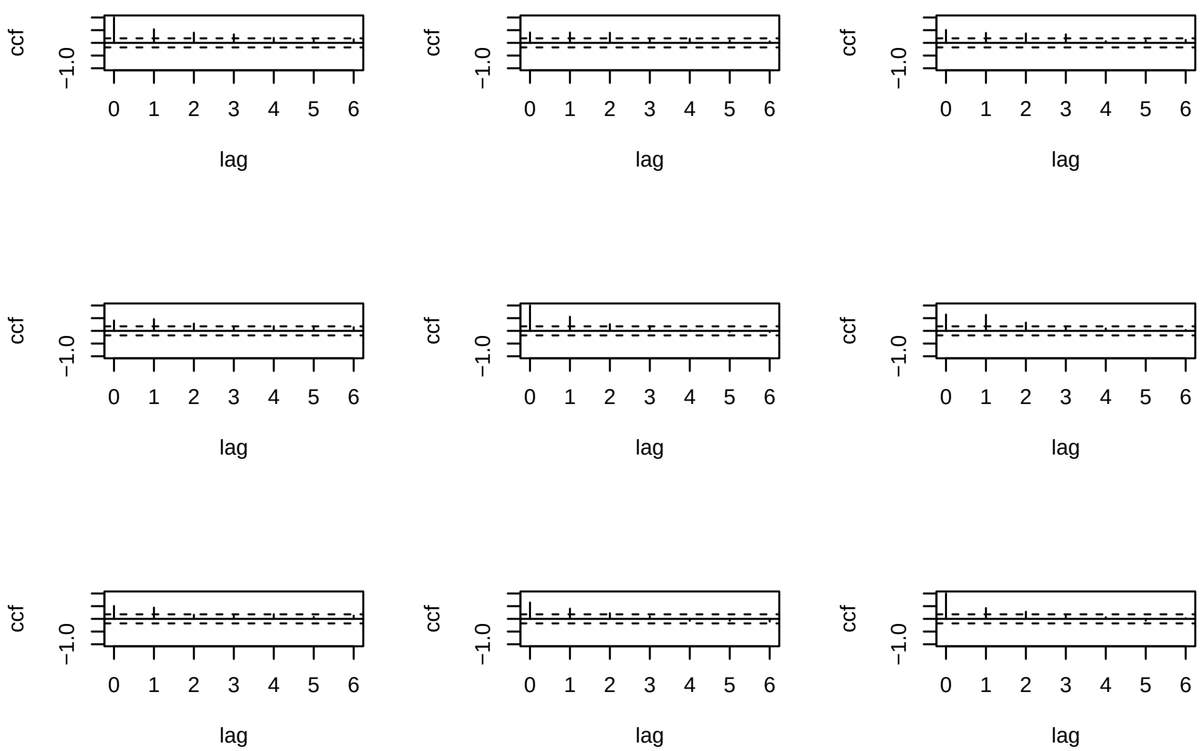
The mvn test above shows that the diff(log(uk)), diff(log(ca)), and diff(log(us)) datasets do not display multivariate or univariate normality, due to skewness and kurtosis for their respective data are not 0.

Despite the strict non-normality, the means of the diff(log(uk)), diff(log(ca)), and diff(log(us)) datasets hover just above 0, in a range from 0.52 to 0.65. not perfect but also not terrible to use for VAR() modelling. These means might be the closest we can get to 0.

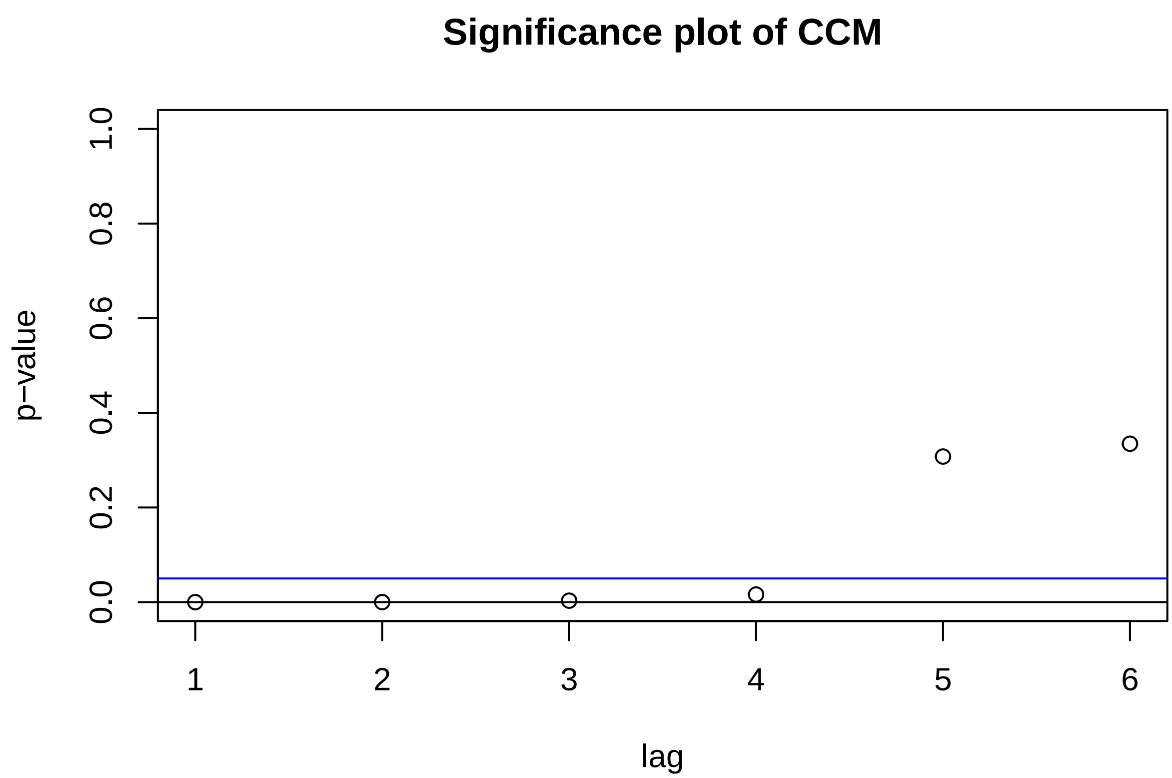
Let us run Cross-Correlation Matrices, or ccm(), on the transformed GDP data:

```
## [1] "Covariance matrix:"
##       uk    ca    us
## uk 0.502 0.228 0.283
## ca 0.228 0.617 0.399
## us 0.283 0.399 0.620
## CCM at lag:  0
##       [,1]  [,2]  [,3]
## [1,] 1.000 0.409 0.507
## [2,] 0.409 1.000 0.645
## [3,] 0.507 0.645 1.000
## Simplified matrix:
## CCM at lag:  1
## + + +
## + + +
## + + +
## CCM at lag:  2
## + + +
## + + +
## . + +
## CCM at lag:  3
## + . +
## . + .
## . . .
## CCM at lag:  4
```

```
## + . .
## + . .
## + . .
## CCM at lag:  5
## . . .
## . . .
## . . .
## CCM at lag:  6
## . . .
## . . .
## . . .
```




```
## Hit Enter for p-value plot of individual ccm:
```



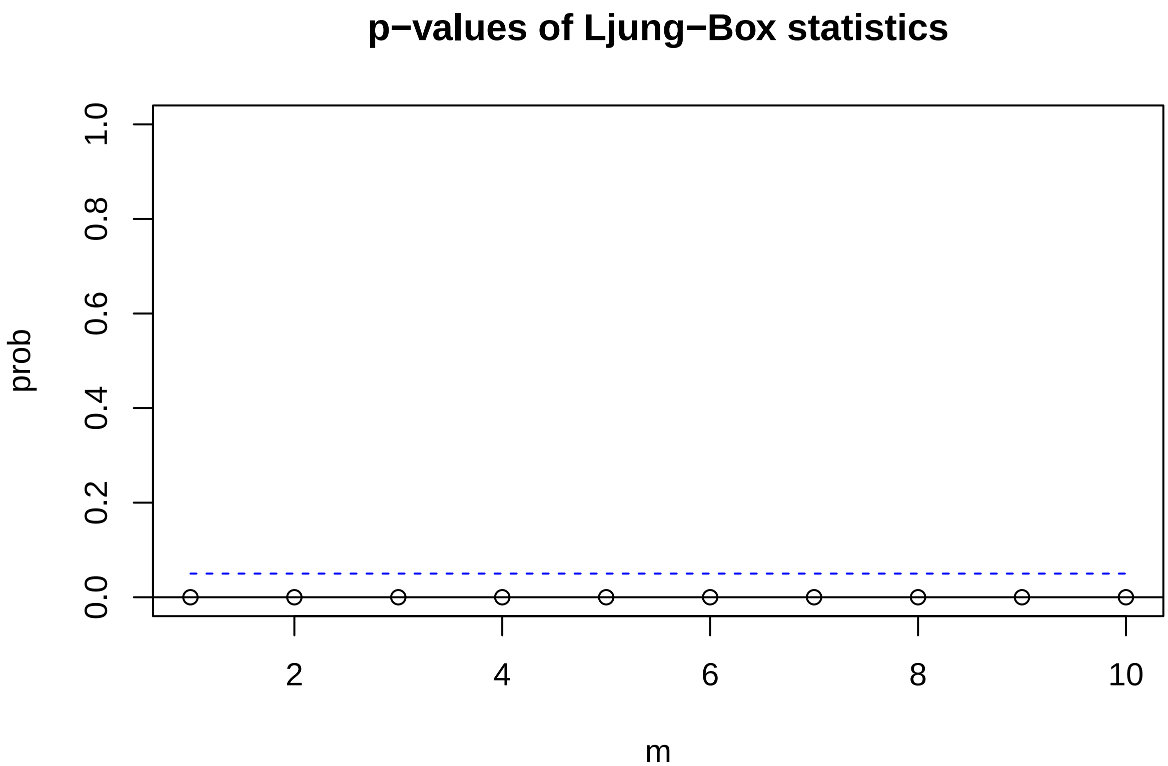
The farthest lag we observe cross correlation is with lag 4 which has the following CCM matrix:

```
# CCM at lag:  4
# + . .
# + . .
# + . .
```

This makes a viable case for a VAR(4) model, as it is the farthest matrix which displays cross-correlation interaction between all observations (in this case, the observations are countries).

We will run a Multivariate Ljung-Box Q Statistics, or mq() test on the transformed multivariate GDP data:

```
## Ljung-Box Statistics:
##          m       Q(m)     df    p-value
##  [1,]   1.0      79.1     9.0        0
##  [2,]   2.0     118.1    18.0        0
##  [3,]   3.0     143.3    27.0        0
##  [4,]   4.0     163.9    36.0        0
##  [5,]   5.0     174.7    45.0        0
##  [6,]   6.0     185.0    54.0        0
##  [7,]   7.0     199.8    63.0        0
##  [8,]   8.0     214.1    72.0        0
##  [9,]   9.0     219.7    81.0        0
## [10,]  10.0     226.7    90.0        0
```



All lags with a multivariate Ljung-Box test p-value < 0.05 in the above plot exhibit serial cross-correlation.

Let's use the Li-McLeod test to test for multivariate ARCH effects:

```
## Q(m) of squared series(LM test):
## Test statistic:  65.70814  p-value:  7.990275e-13
## Rank-based Test:
## Test statistic:  67.2266  p-value:  3.865797e-13
## Q_k(m) of squared series:
## Test statistic:  88.4113  p-value:  0.0001192077
## Robust Test(5%) :  158.1995  p-value:  1.64313e-14
```

With the Li-McLeod test p-value < 0.05, we can say the transformed GDP multivariate data contains ARCH effects.

Let us perform a VARselect() on the transformed GDP multivariate data:

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      4      1      1      4
##
## $criteria
##                  1           2           3           4           5           6
## AIC(n) -3.92828522 -3.99970466 -3.99099473 -4.01772671 -3.91444555 -3.84078641
## HQ(n)  -3.81326896 -3.79842620 -3.70345407 -3.64392386 -3.45438050 -3.29445917
## SC(n)  -3.64498533 -3.50392985 -3.28274500 -3.09700206 -2.78124598 -2.49511193
## FPE(n)  0.01967896  0.01832891  0.01850452  0.01804363  0.02005509  0.02166444
```

```
##                  7           8
## AIC(n) -3.7526434 -3.81944143
## HQ(n)  -3.1200539 -3.10058980
## SC(n)  -2.1944940 -2.04881712
## FPE(n)  0.0237765  0.02238471
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      4      1      1      4
```

Based on the lowest AIC value, the suggested lag to use on the transformed GDP multivariate data for a model would be VAR(4).

2.2. Fit a VAR(4) model to the series and perform model checking.

We create the following VAR(4) model from the transformed GDP data:

```
m1 <- MTS::VAR(growth,p=p)
```

```
## Constant term:
## Estimates:  0.1479567 0.07757439 0.2386772
## Std.Error:  0.07478564 0.08003665 0.08583514
## AR coefficient matrix
## AR( 1 )-matrix
##       [,1]   [,2]   [,3]
## [1,] 0.516 0.0719 0.0639
## [2,] 0.378 0.3160 0.4096
## [3,] 0.519 0.1730 0.1504
## standard error
##        [,1]   [,2]   [,3]
## [1,] 0.0953 0.0945 0.0887
## [2,] 0.1020 0.1011 0.0949
## [3,] 0.1094 0.1084 0.1018
## AR( 2 )-matrix
##         [,1]   [,2]     [,3]
## [1,] -0.0504  0.160 -0.00198
## [2,] -0.1740 -0.254  0.06295
## [3,] -0.2178 -0.159  0.22561
## standard error
##       [,1]   [,2]   [,3]
## [1,] 0.101 0.0986 0.0955
## [2,] 0.108 0.1055 0.1022
## [3,] 0.116 0.1132 0.1096
## AR( 3 )-matrix
##        [,1]    [,2]   [,3]
## [1,] 0.0524 -0.2788 0.1411
## [2,] 0.0962  0.1203 0.0137
## [3,] 0.0478 -0.0786 0.0738
## standard error
##       [,1]   [,2]   [,3]
## [1,] 0.103 0.0983 0.0937
## [2,] 0.110 0.1052 0.1003
## [3,] 0.118 0.1129 0.1076
```

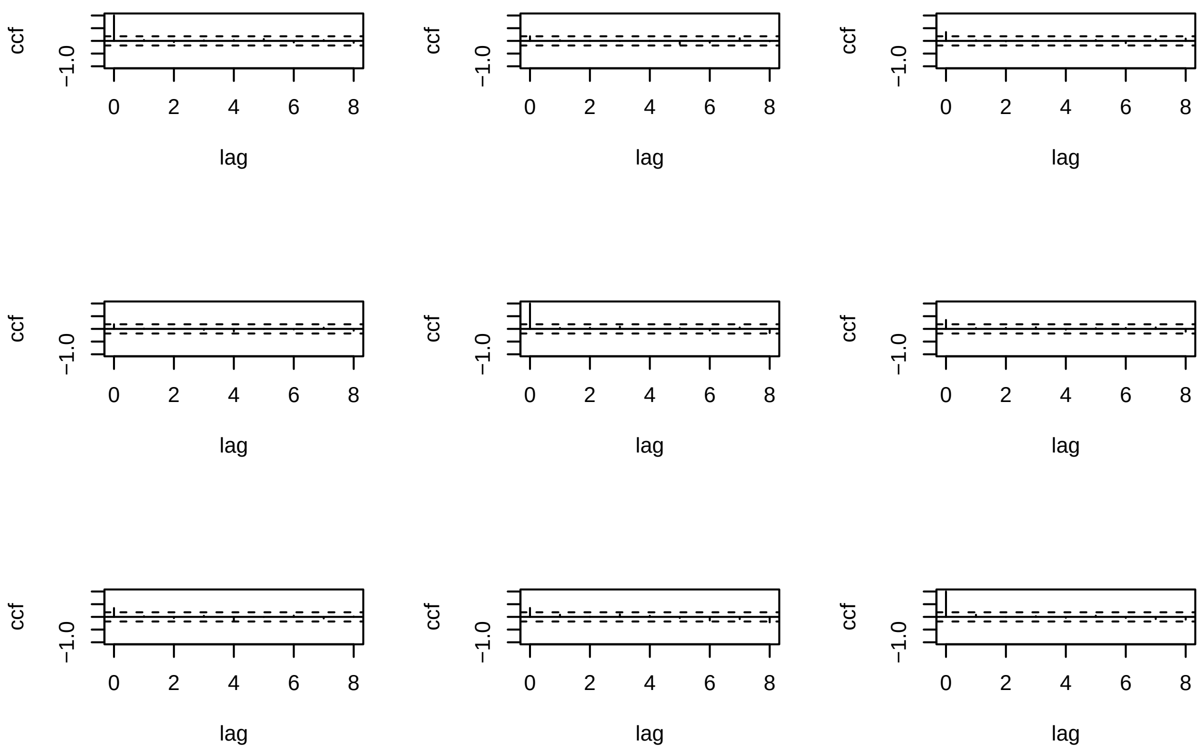
```
## AR( 4 )-matrix
##        [,1]    [,2]    [,3]
## [1,] 0.0401  0.2617 -0.2465
## [2,] 0.0747 -0.0903 -0.0978
## [3,] 0.1541 -0.1518 -0.0535
## standard error
##        [,1]   [,2]   [,3]
## [1,] 0.0910 0.0869 0.0882
## [2,] 0.0974 0.0931 0.0944
## [3,] 0.1045 0.0998 0.1013
##
## Residuals cov-mtx:
##            [,1]       [,2]       [,3]
## [1,] 0.22430413 0.04383870 0.08933612
## [2,] 0.04383870 0.25690861 0.09675468
## [3,] 0.08933612 0.09675468 0.29548204
##
## det(SSE) =  0.01306715
## AIC =  -3.761654
## BIC =  -2.9471
## HQ  =  -3.430743
```

From the model summary we get an AIC of -3.761654.

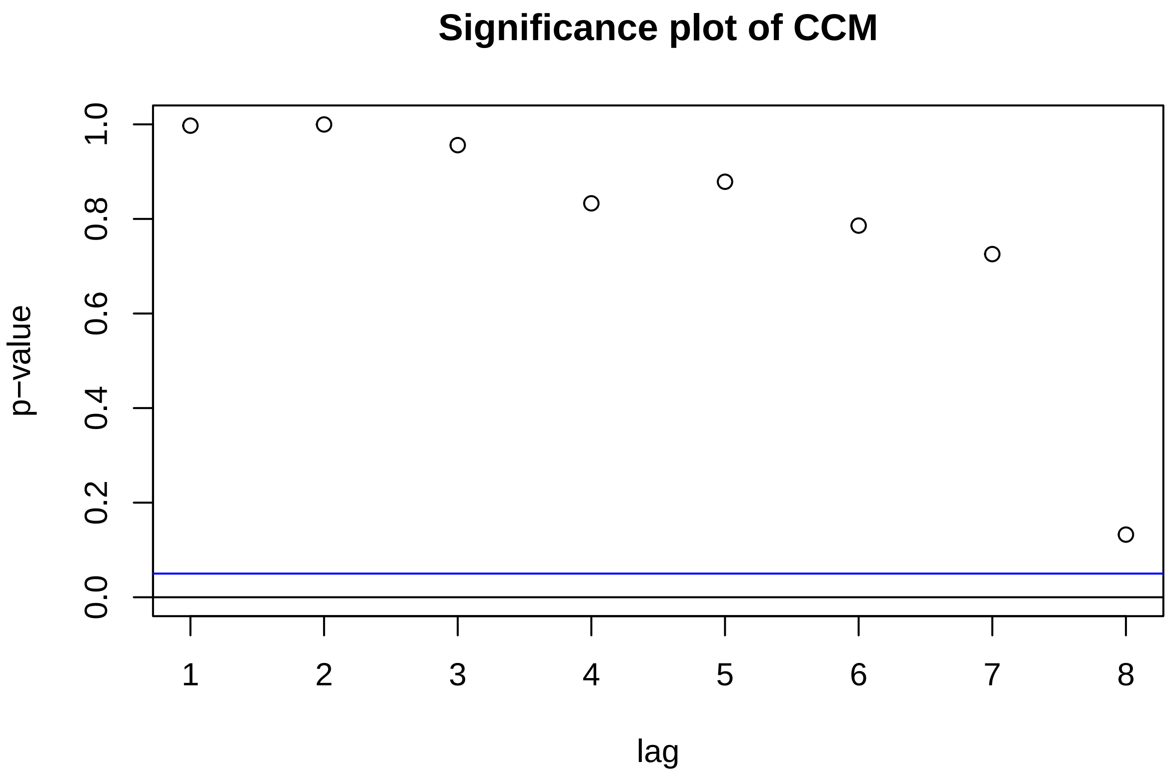
Model diagnostics: VAR(4) (m1)

```
## [1] "Covariance matrix:"
##        uk     ca     us
## uk 0.2262 0.0442 0.0901
## ca 0.0442 0.2590 0.0976
## us 0.0901 0.0976 0.2979
## CCM at lag:  0
##       [,1]  [,2]  [,3]
## [1,] 1.000 0.183 0.347
## [2,] 0.183 1.000 0.351
## [3,] 0.347 0.351 1.000
## Simplified matrix:
## CCM at lag:  1
## . . .
## . . .
## . . .
## CCM at lag:  2
## . . .
## . . .
## . . .
## CCM at lag:  3
## . . .
## . . .
## . . .
## CCM at lag:  4
## . . .
## . . .
## . . .
## CCM at lag:  5
```

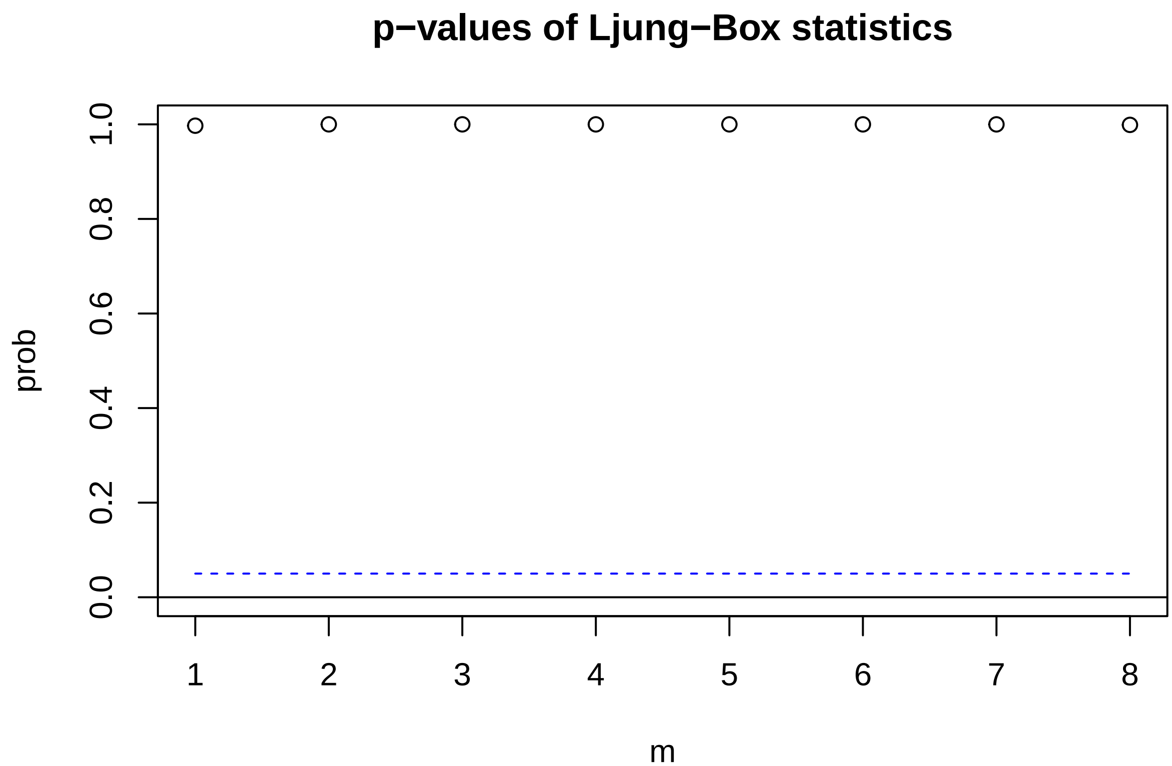
```
## . . .
## . . .
## . . .
## CCM at lag:  6
## . . .
## . . .
## . . .
## CCM at lag:  7
## . . .
## . . .
## . . .
## CCM at lag:  8
## . . .
## . . .
## . - .
```



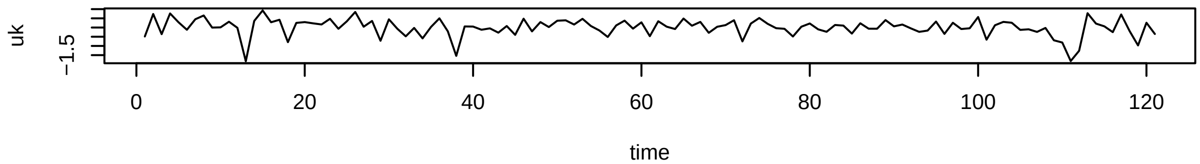

```
## Hit Enter for p-value plot of individual ccm:
```

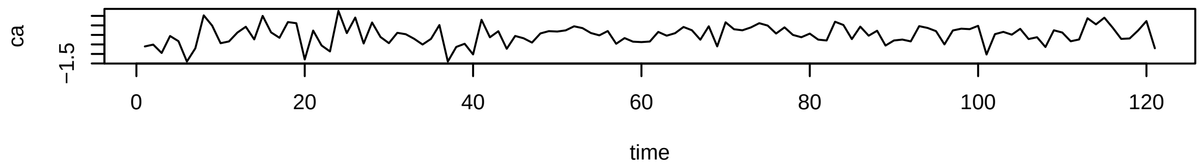


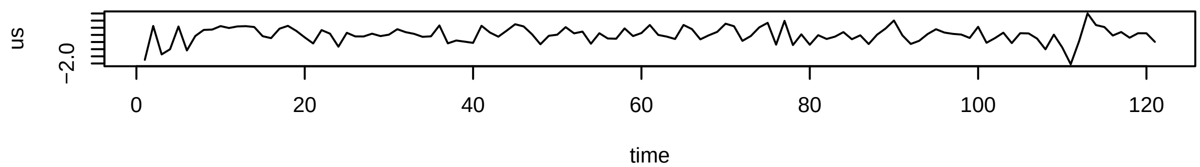

```
## Hit Enter to compute MQ-statistics:
##
## Ljung-Box Statistics:
##         m       Q(m)     df    p-value
## [1,]  1.00      1.50    9.00        1
## [2,]  2.00      2.32   18.00        1
## [3,]  3.00      5.58   27.00        1
## [4,]  4.00     10.75   36.00        1
## [5,]  5.00     15.29   45.00        1
## [6,]  6.00     20.88   54.00        1
## [7,]  7.00     27.09   63.00        1
## [8,]  8.00     40.83   72.00        1
```




```
## Hit Enter to obtain residual plots:
```







The model diagnostics show the model lags exhibit almost no serial cross-correlation, or very little, due to relatively stationary ccm plots and Ljung-Box p-values > 0.05, indicating lag independence.

2.3. Simplify the model by removing insignificant parameters with type-I error rates at $\alpha = 0.05$.

We create the following refined VAR(4) model:

```
m2 <- refVAR(m1, thres=1.96)
```

```
## Constant term:
## Estimates:  0.2044312 0 0.3232122
## Std.Error:  0.06873281 0 0.08050682
## AR coefficient matrix
## AR( 1 )-matrix
##       [,1]  [,2]  [,3]
## [1,] 0.565 0.000 0.000
## [2,] 0.376 0.299 0.408
## [3,] 0.508 0.268 0.000
## standard error
##        [,1]   [,2]   [,3]
## [1,] 0.0798 0.0000 0.0000
## [2,] 0.0852 0.0899 0.0887
## [3,] 0.0882 0.0751 0.0000
## AR( 2 )-matrix
```

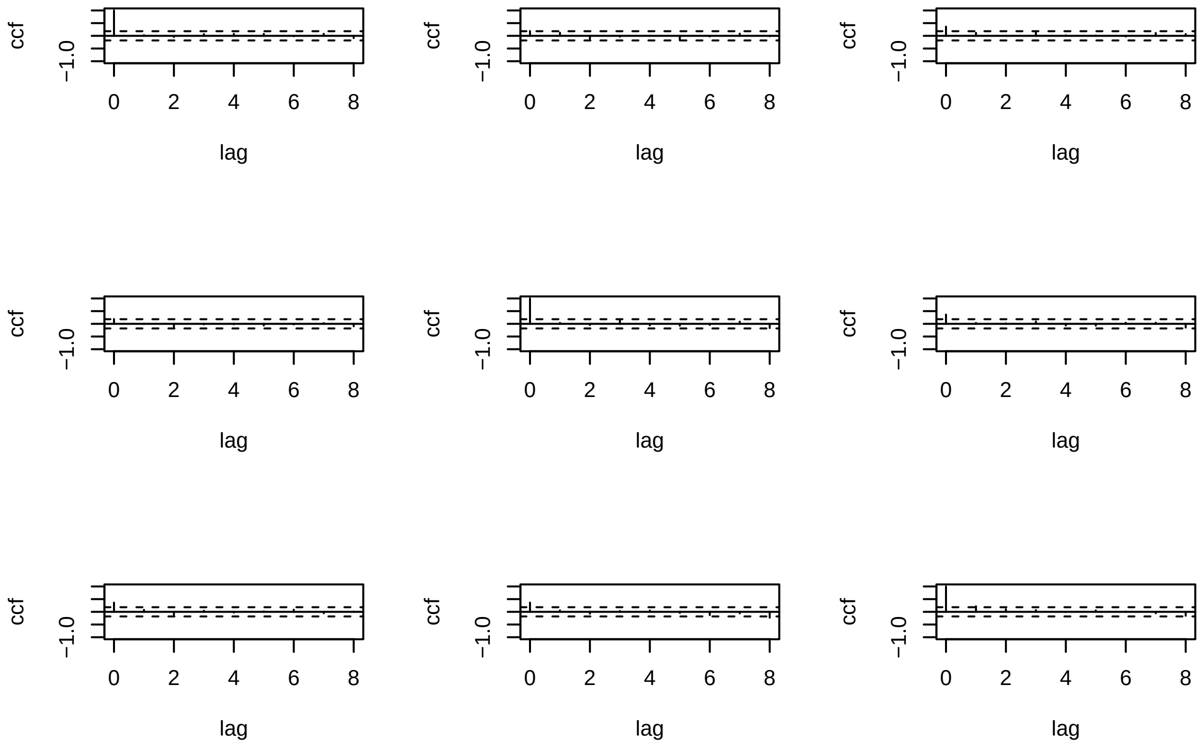
```
##      [,1]   [,2] [,3]
## [1,]    0  0.284    0
## [2,]    0 -0.153    0
## [3,]    0  0.000    0
## standard error
##      [,1]   [,2] [,3]
## [1,]    0 0.0728    0
## [2,]    0 0.0769    0
## [3,]    0 0.0000    0
## AR( 3 )-matrix
##      [,1]   [,2] [,3]
## [1,]    0 -0.249    0
## [2,]    0  0.000    0
## [3,]    0  0.000    0
## standard error
##      [,1] [,2] [,3]
## [1,]    0 0.09    0
## [2,]    0 0.00    0
## [3,]    0 0.00    0
## AR( 4 )-matrix
##      [,1]   [,2]   [,3]
## [1,]    0  0.296 -0.243
## [2,]    0  0.000  0.000
## [3,]    0 -0.199  0.000
## standard error
##      [,1]   [,2]   [,3]
## [1,]    0 0.0797 0.0849
## [2,]    0 0.0000 0.0000
## [3,]    0 0.0686 0.0000
##
## Residuals cov-mtx:
##            [,1]       [,2]      [,3]
## [1,] 0.23678534 0.04832432 0.1010461
## [2,] 0.04832432 0.27764069 0.1099519
## [3,] 0.10104613 0.10995186 0.3285638
##
## det(SSE) =  0.01620931
## AIC =  -3.930169
## BIC =  -3.658651
## HQ  =  -3.819866
```

From the model summary we get an AIC of -3.930169, which is lower than the original VAR(4) model (-3.761654).

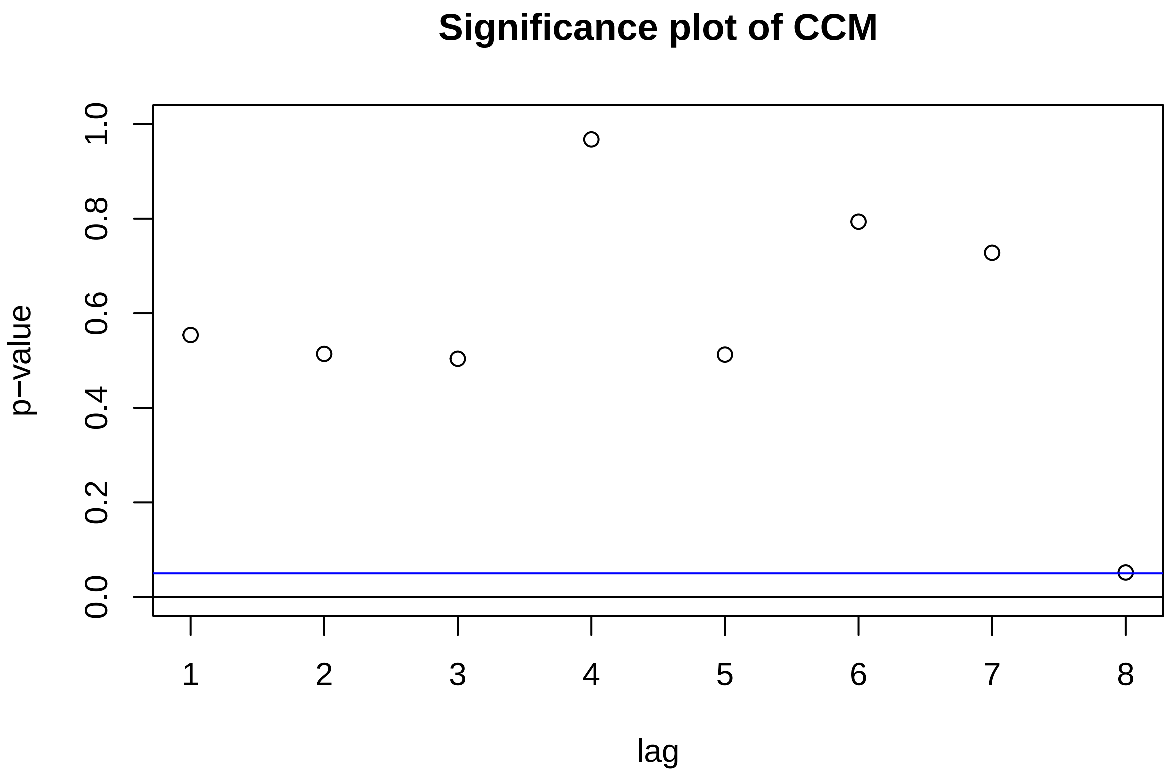
Model diagnostics: refined VAR(4) (m2)

```
## [1] "Covariance matrix:"
##        uk     ca    us
## uk 0.2388 0.0487 0.102
## ca 0.0487 0.2790 0.111
## us 0.1019 0.1109 0.331
## CCM at lag:  0
##       [,1]  [,2]  [,3]
## [1,] 1.000 0.189 0.362
```

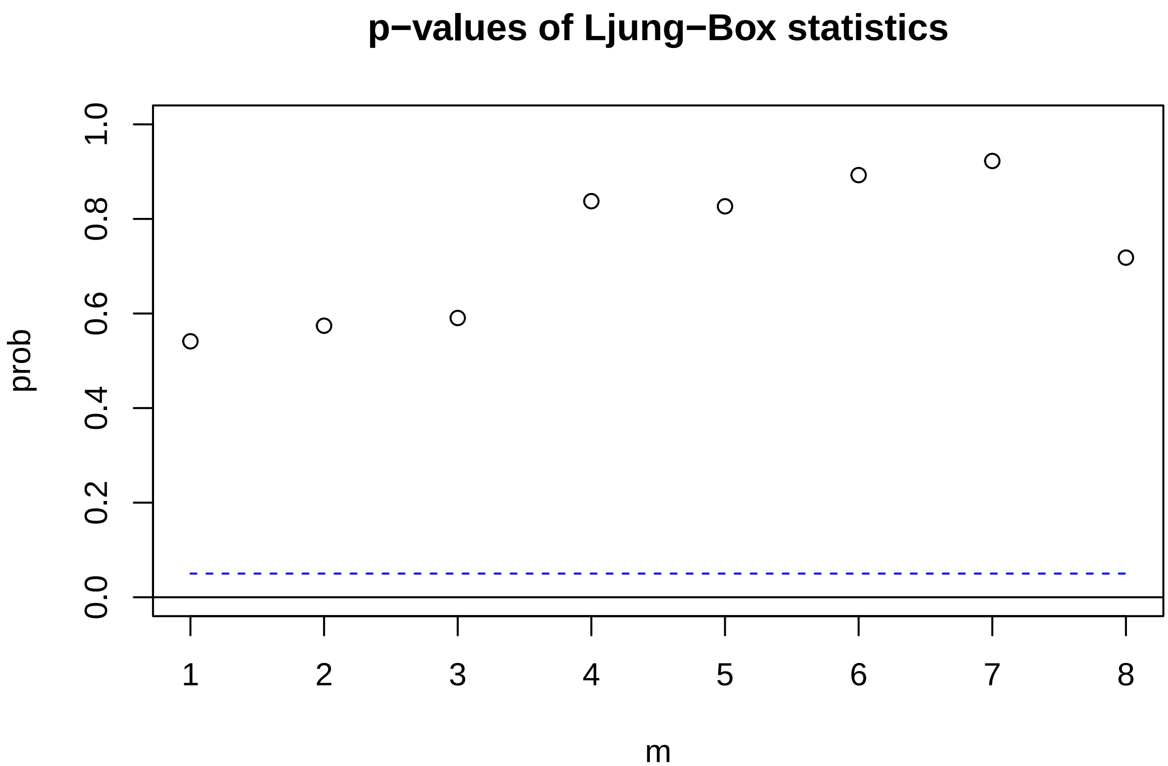
```
## [2,] 0.189 1.000 0.365
## [3,] 0.362 0.365 1.000
## Simplified matrix:
## CCM at lag:  1
## . . .
## . . .
## . . +
## CCM at lag:  2
## . . .
## . . .
## . . .
## CCM at lag:  3
## . . .
## . . .
## . . .
## CCM at lag:  4
## . . .
## . . .
## . . .
## CCM at lag:  5
## . . .
## . . .
## . . .
## CCM at lag:  6
## . . .
## . . .
## . . .
## CCM at lag:  7
## . . .
## . . .
## . . .
## CCM at lag:  8
## . . .
## . . .
## . - .
```



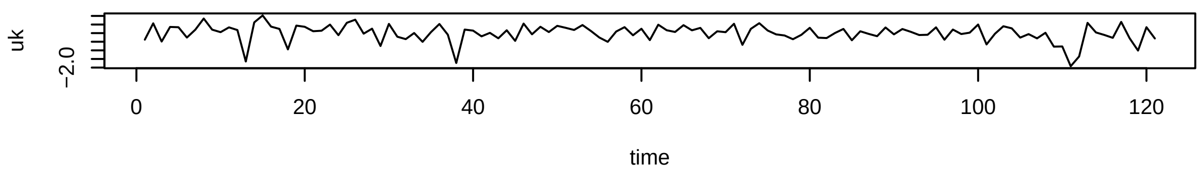

```
## Hit Enter for p-value plot of individual ccm:
```

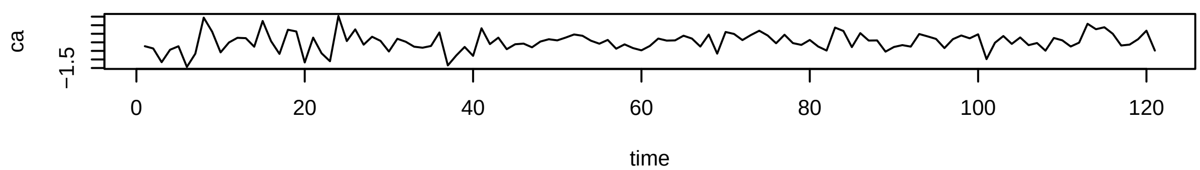


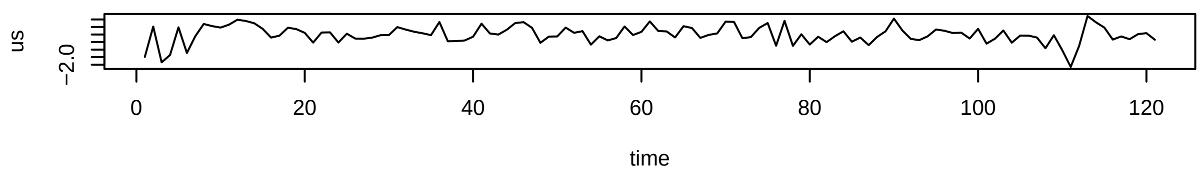

```
## Hit Enter to compute MQ-statistics:
##
## Ljung-Box Statistics:
##         m       Q(m)     df    p-value
## [1,]  1.00      7.93    9.00     0.54
## [2,]  2.00     16.26   18.00     0.57
## [3,]  3.00     24.71   27.00     0.59
## [4,]  4.00     27.71   36.00     0.84
## [5,]  5.00     36.06   45.00     0.83
## [6,]  6.00     41.54   54.00     0.89
## [7,]  7.00     47.78   63.00     0.92
## [8,]  8.00     64.66   72.00     0.72
```




```
## Hit Enter to obtain residual plots:
```







The model diagnostics show the model lags exhibit no serial cross-correlation, or very little, due to relatively stationary ccm plots and Ljung-Box p-values > 0.05, indicating lag independence.

2.4. From each model's diagnostics, compare the VAR(4) and the simplified models. Suggest and justify which model, if either, is best.

AIC comparison: VAR(4) (m1) vs refined VAR(4) (m2)

```
m1$aic # VAR(4)
```

```
## [1] -3.761654
```

```
m2$aic # refined VAR(4)
```

```
## [1] -3.930169
```

BIC comparison: VAR(4) (m1) vs refined VAR(4) (m2)

```
m1$bic # VAR(4)
```

```
## [1] -2.9471
```

```
m2$bic # refined VAR(4)
```

```
## [1] -3.658651
```

Based on the AIC and BIC scores above and what we've seen from their summaries and diagnostics in sections 2.2 and 2.3, the refined or simplified VAR(4) model would be preferred as it has lower scores compared to the original VAR(4) model. That being said, the refined VAR(4) just edges out as the differences between them are relatively small.

2.5. Generate a multivariate forecast from your best model.

Let's create a VAR(4) forecast model for the next 10 months:

```
p <- 4
m <- vars::VAR(X, p=p, type="const")
mv4 <- m
```

Serial Test: VAR(4) forecast:

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object m
## Chi-squared = 51.548, df = 54, p-value = 0.5695
```

The Portmanteau Test for the VAR(4) forecast model has p-value > 0.05, indicating the model has no multivariate serial cross-correlation.

Let's calculate forecasts with the VAR(4) model (m1) for the next 10 months:

```
title <- sprintf("GDP Forecasts VAR(%i)", p)
fm1 <- MTS::VARpred(m1,10)
```

```
## orig  125
## Forecasts at origin:  125
##            uk       ca      us
##  [1,] 0.23313 -0.07932 0.06016
##  [2,] 0.06873  0.17700 0.15142
##  [3,] 0.49487  0.15769 0.28147
##  [4,] 0.39105  0.38736 0.60488
##  [5,] 0.36524  0.53573 0.57553
##  [6,] 0.48612  0.53700 0.60477
##  [7,] 0.53168  0.60292 0.71000
##  [8,] 0.49005  0.61941 0.65772
##  [9,] 0.52962  0.56914 0.61732
## [10,] 0.54550  0.57123 0.63937
## Standard Errors of predictions:
##         [,1]   [,2]   [,3]
##  [1,] 0.4736 0.5069 0.5436
##  [2,] 0.5445 0.6537 0.6302
##  [3,] 0.5839 0.7012 0.6718
```

```
##  [4,] 0.6229 0.7257 0.6910
##  [5,] 0.6359 0.7476 0.7204
##  [6,] 0.6438 0.7579 0.7260
##  [7,] 0.6491 0.7602 0.7274
##  [8,] 0.6516 0.7618 0.7291
##  [9,] 0.6534 0.7627 0.7299
## [10,] 0.6544 0.7637 0.7309
## Root mean square errors of predictions:
##         [,1]   [,2]   [,3]
##  [1,] 0.4976 0.5326 0.5711
##  [2,] 0.6374 0.8285 0.7428
##  [3,] 0.6390 0.7679 0.7305
##  [4,] 0.6779 0.7615 0.7192
##  [5,] 0.6553 0.7796 0.7630
##  [6,] 0.6556 0.7733 0.7345
##  [7,] 0.6570 0.7637 0.7296
##  [8,] 0.6554 0.7642 0.7315
##  [9,] 0.6562 0.7640 0.7313
## [10,] 0.6560 0.7652 0.7325
```

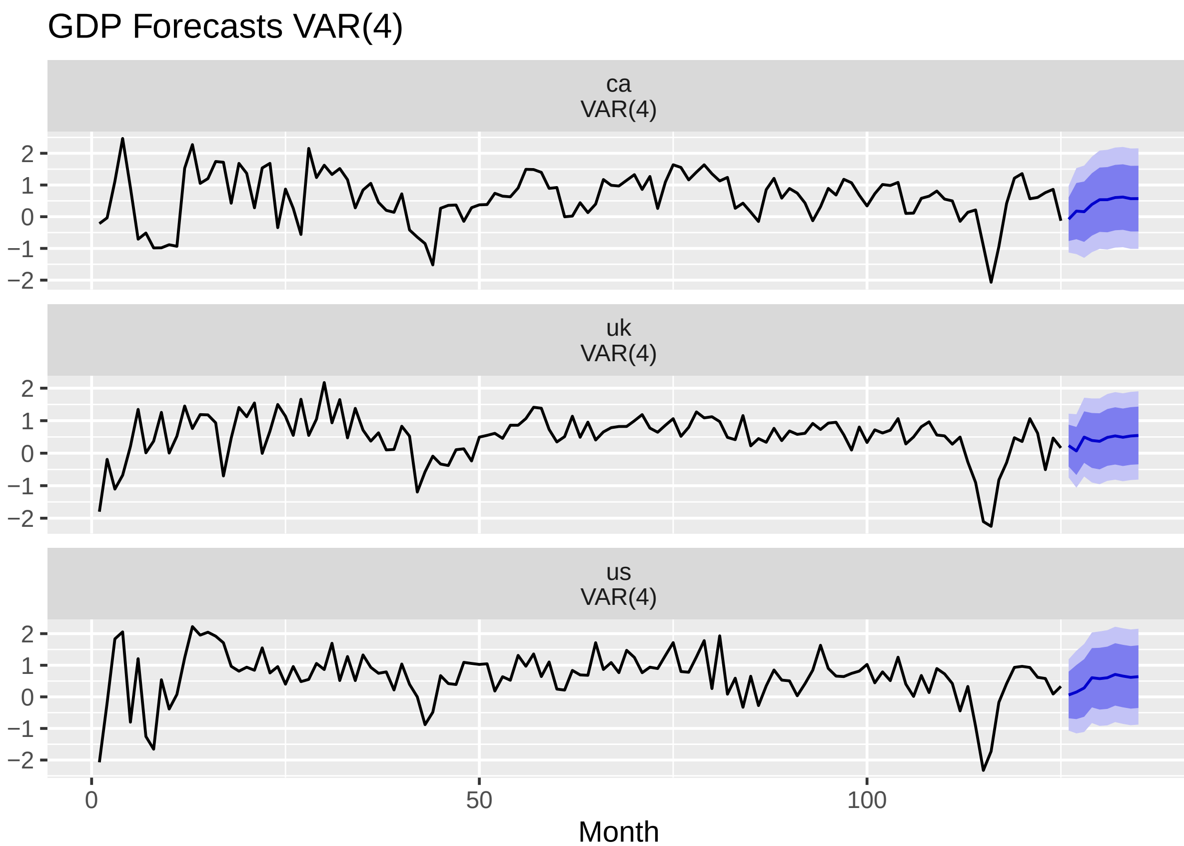
Let's calculate forecasts with the refined VAR(4) model (m2) for the next 10 months:

```r
title <- sprintf("GDP Forecasts VAR(%i)", p)
fm2 <- MTS::VARpred(m2,10)
```

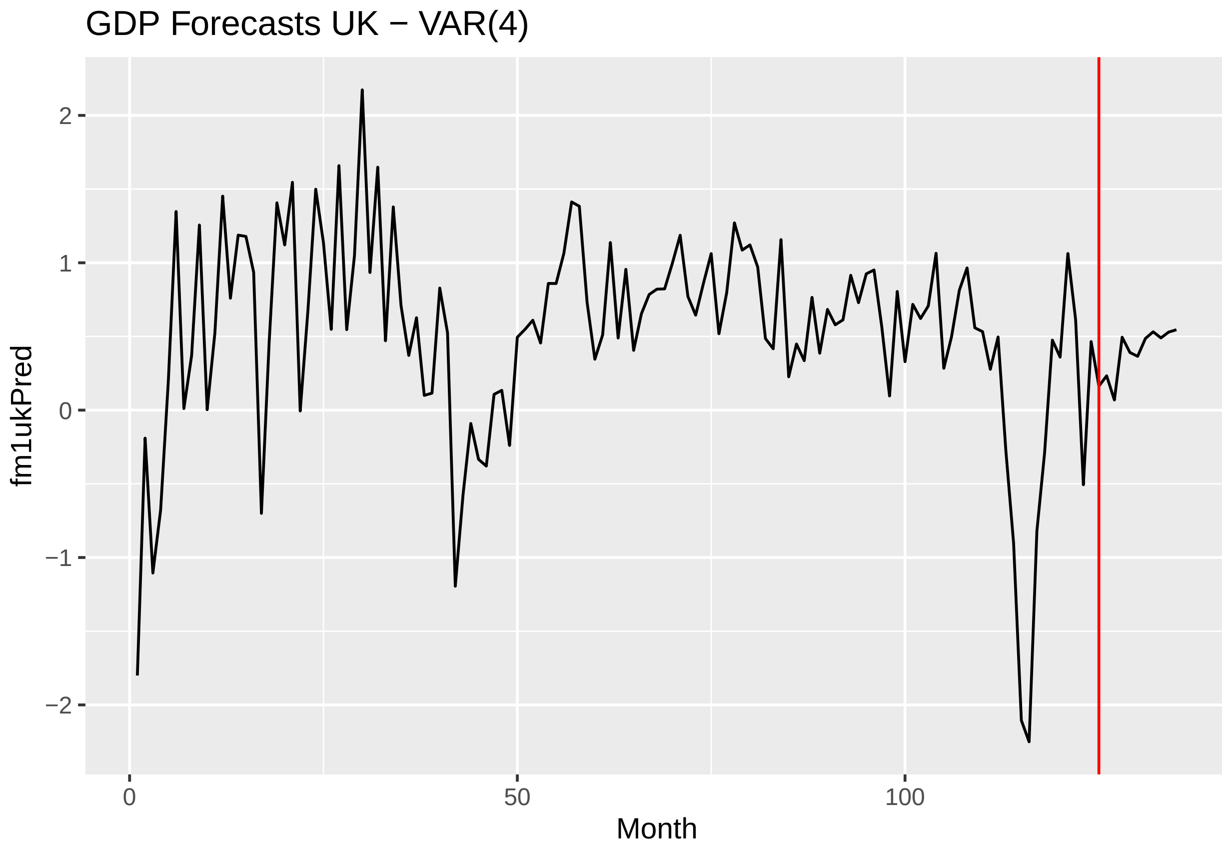
```
## orig  125
## Forecasts at origin:  125
##           uk      ca     us
##  [1,] 0.3820 0.02603 0.2504
##  [2,] 0.2533 0.27343 0.3731
##  [3,] 0.6210 0.32530 0.3535
##  [4,] 0.5078 0.43336 0.7515
##  [5,] 0.4625 0.57752 0.6922
##  [6,] 0.4982 0.56281 0.6585
##  [7,] 0.5525 0.53607 0.6624
##  [8,] 0.4785 0.55234 0.6613
##  [9,] 0.4899 0.53300 0.5994
## [10,] 0.5114 0.50375 0.6029
## Standard Errors of predictions:
##         [,1]   [,2]   [,3]
##  [1,] 0.4866 0.5269 0.5732
##  [2,] 0.5589 0.6778 0.6502
##  [3,] 0.6063 0.7407 0.6911
##  [4,] 0.6310 0.7694 0.7173
##  [5,] 0.6525 0.7854 0.7316
##  [6,] 0.6601 0.7911 0.7385
##  [7,] 0.6638 0.7945 0.7400
##  [8,] 0.6651 0.7960 0.7405
##  [9,] 0.6675 0.7965 0.7408
## [10,] 0.6690 0.7969 0.7414
## Root mean square errors of predictions:
##         [,1]   [,2]   [,3]
##  [1,] 0.5113 0.5536 0.6023
```

```
##  [2,] 0.6537 0.8578 0.7524
##  [3,] 0.6720 0.8273 0.7490
##  [4,] 0.6666 0.8111 0.7554
##  [5,] 0.6840 0.8090 0.7529
##  [6,] 0.6714 0.7998 0.7488
##  [7,] 0.6694 0.7997 0.7422
##  [8,] 0.6671 0.7984 0.7413
##  [9,] 0.6711 0.7971 0.7413
## [10,] 0.6712 0.7975 0.7423
```

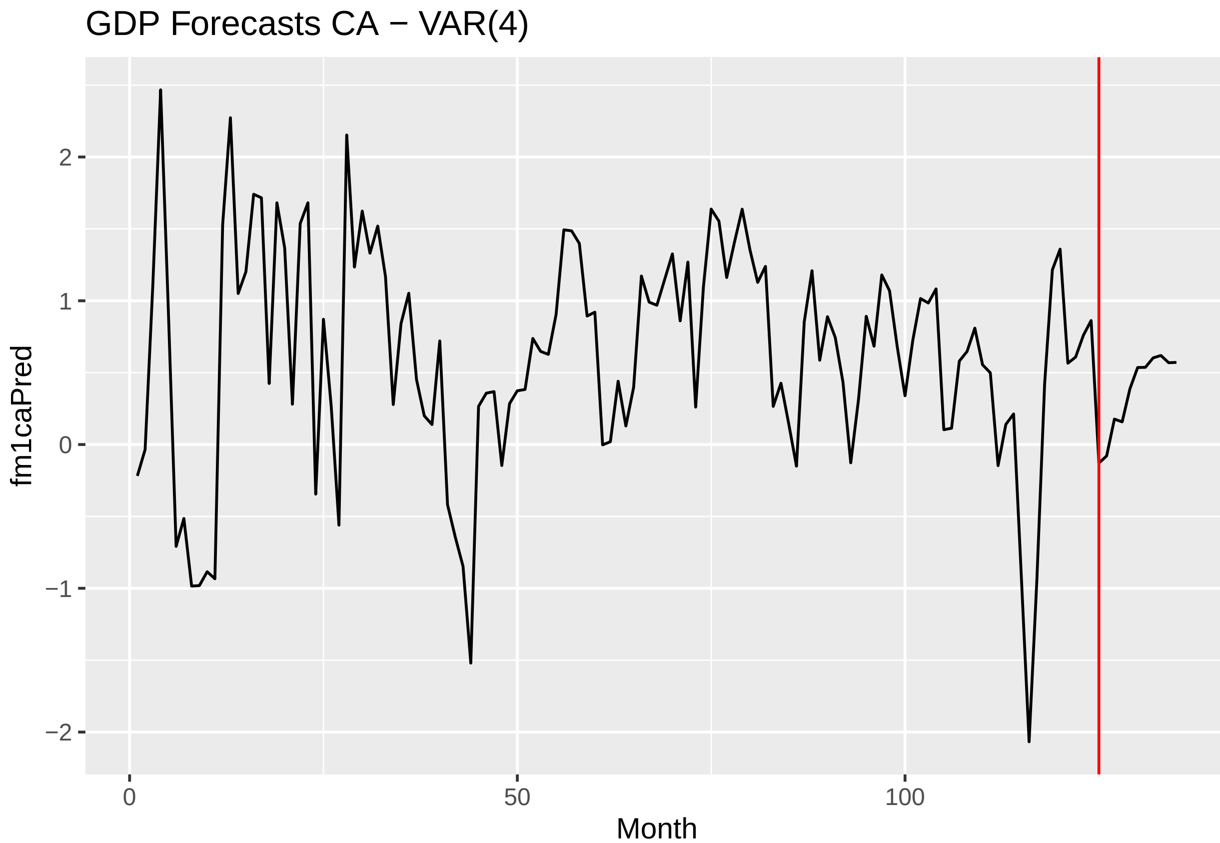
Plot the VAR(4) forecast:

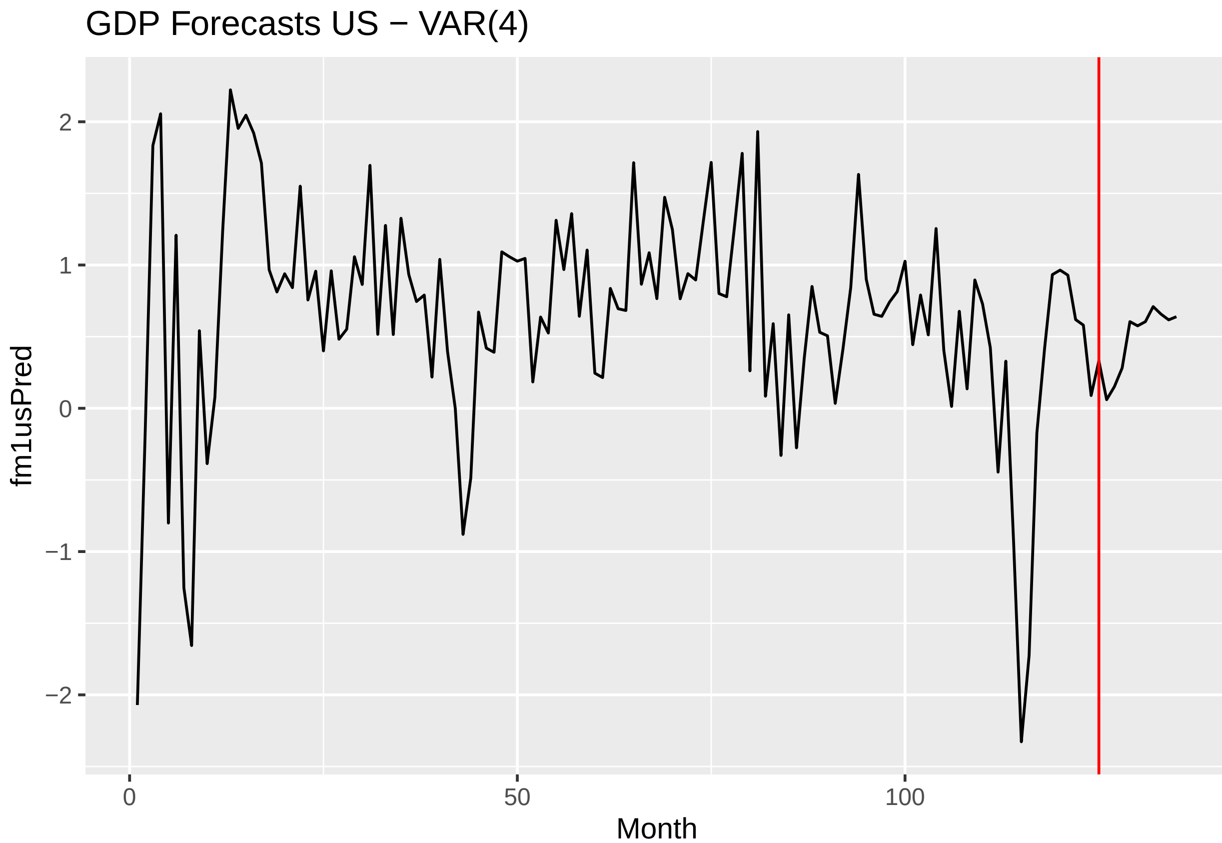


Individual plots of the VAR(4) forecasts:

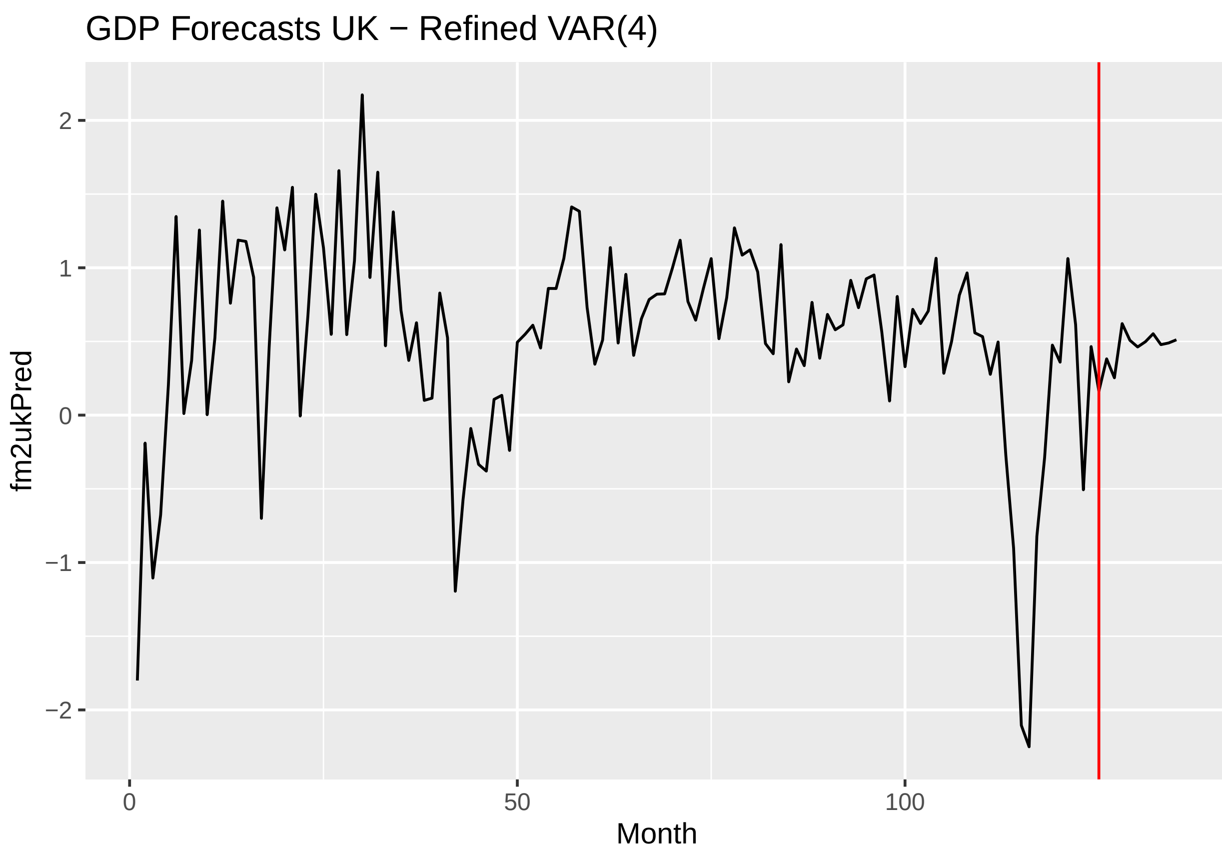


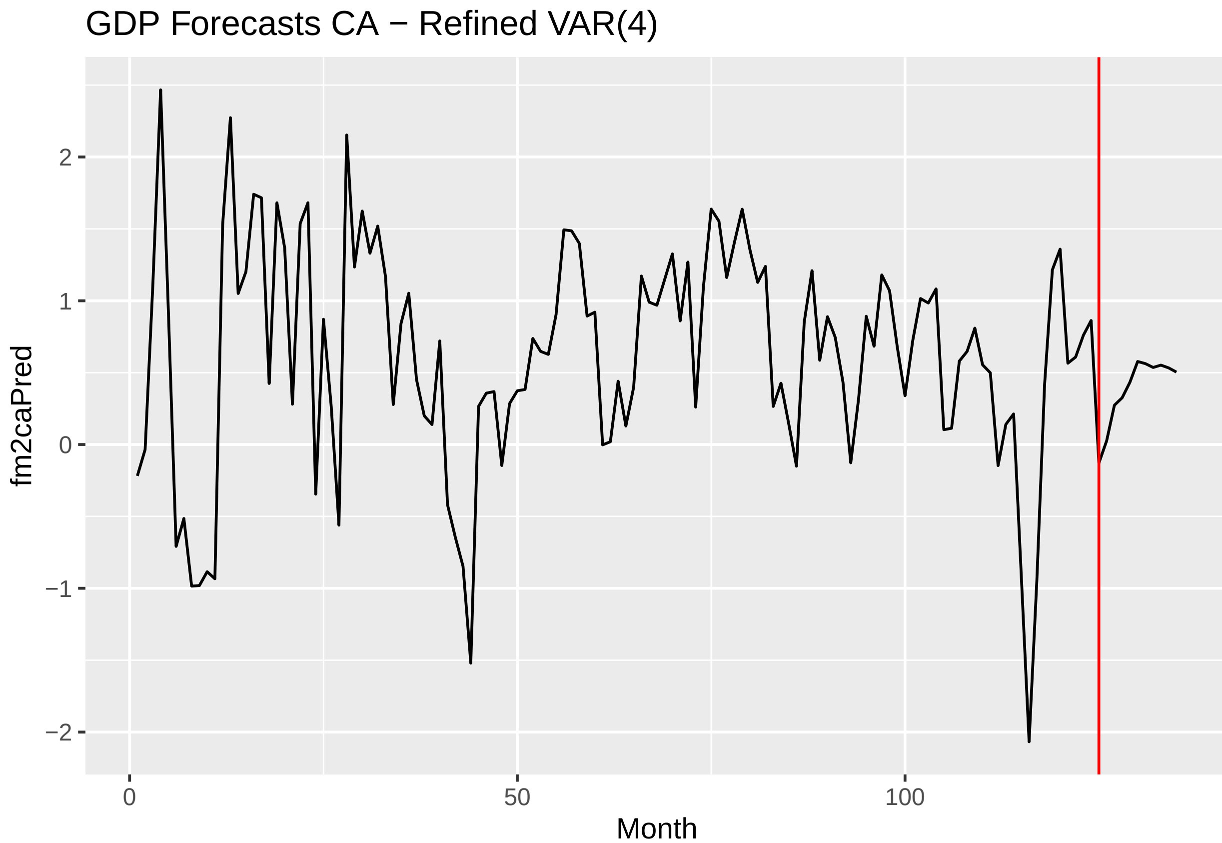
GDP Forecasts UK − VAR(4)
fm1ukPred
2
1
0
−1
−2
0
50
100
Month

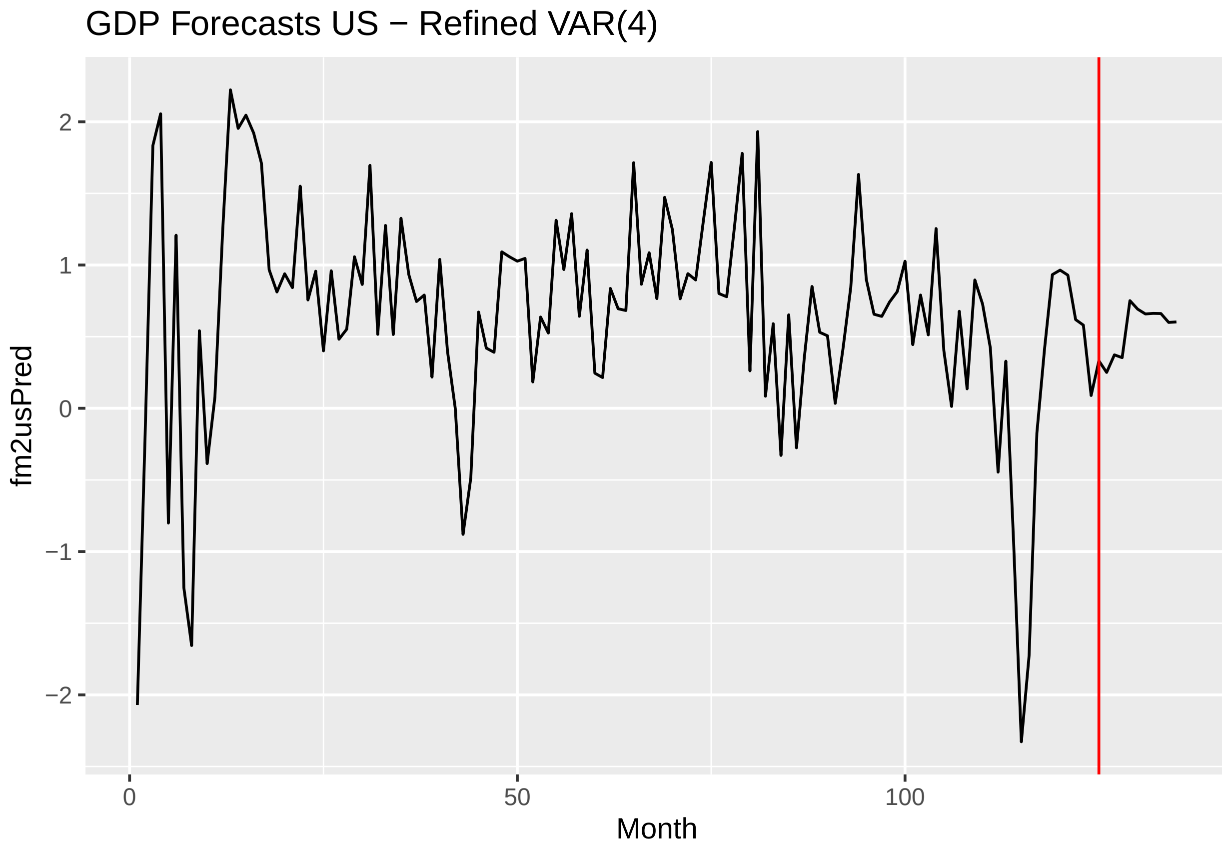

GDP Forecasts CA − VAR(4)
fm1caPred
2
1
0
−1
−2
0
50
100
Month



Plot the individual refined VAR(4) forecasts for each country:


GDP Forecasts UK − Refined VAR(4)
fm2ukPred
2
1
0
−1
−2
0
50
100
Month


GDP Forecasts CA − Refined VAR(4)
fm2caPred
2
1
0
−1
−2
0
50
100
Month



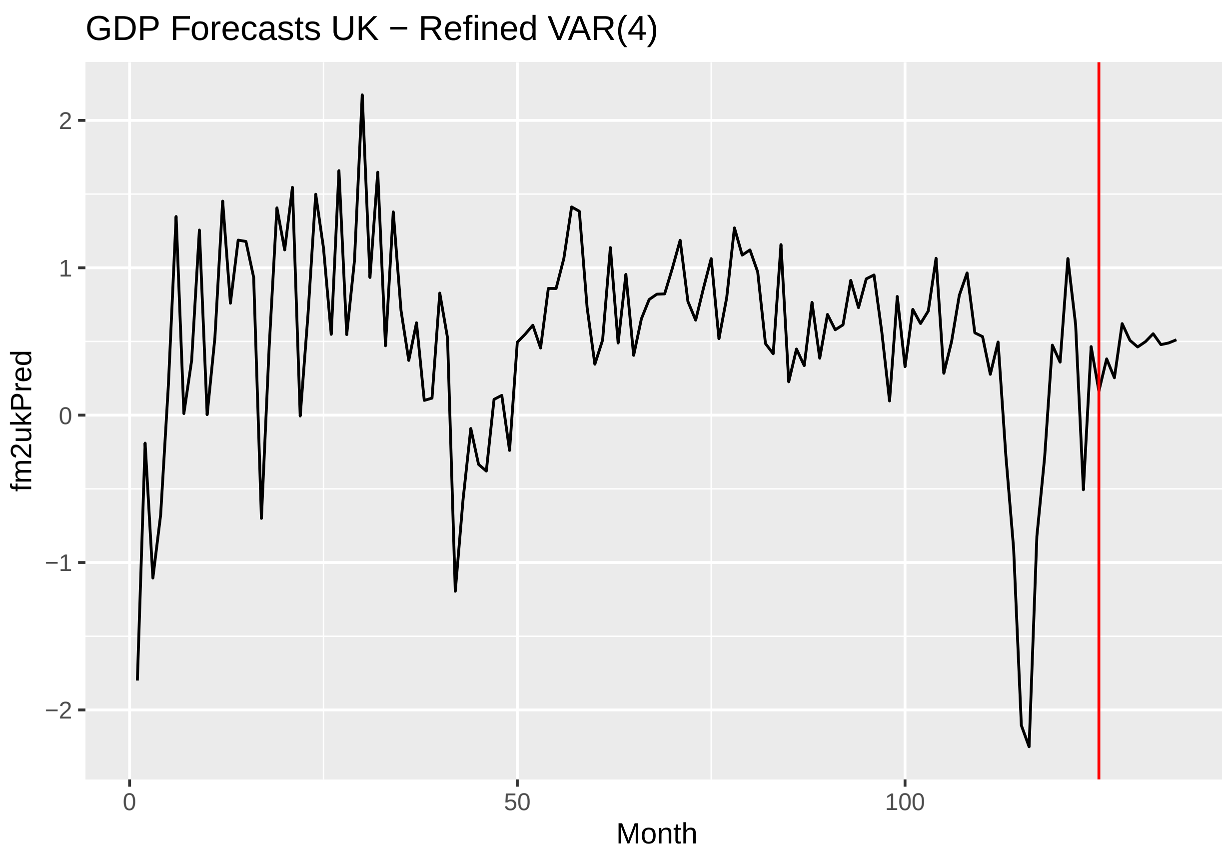
While we do not have 80% and 95% confidence interval forecasts for the refined VAR(4) model, we can compare the point forecasts for each country between the VAR(4) and refined VAR(4) models.

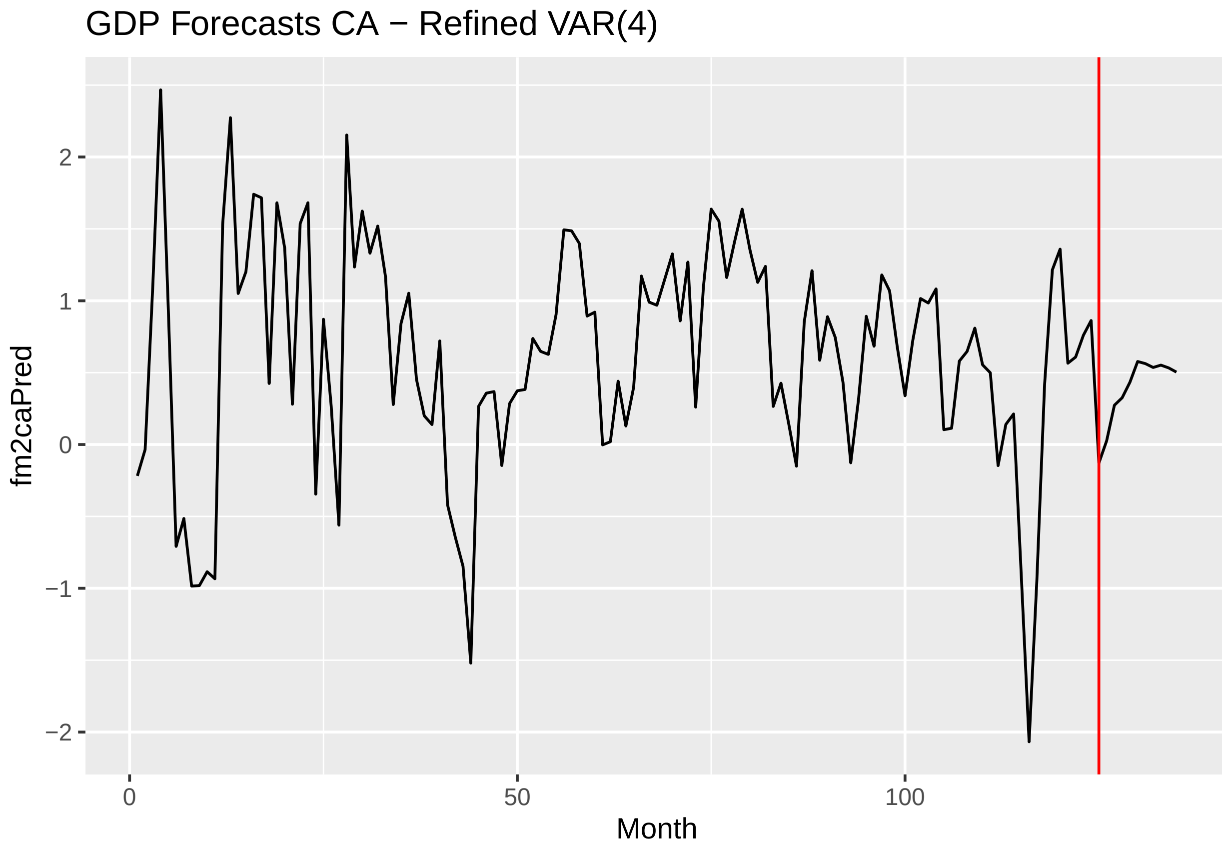
Generally speaking, when comparing the point forecasts between the VAR(4) model and the refined VAR(4) model, the patterns are very similar to each other, such that both sets of forecasts have a small upward spike in the first half of the forecast. But the VAR(4) forecasts have a somewhat general upward trend in the second half while the refined VAR(4) forecasts have a somewhat general downward trend.

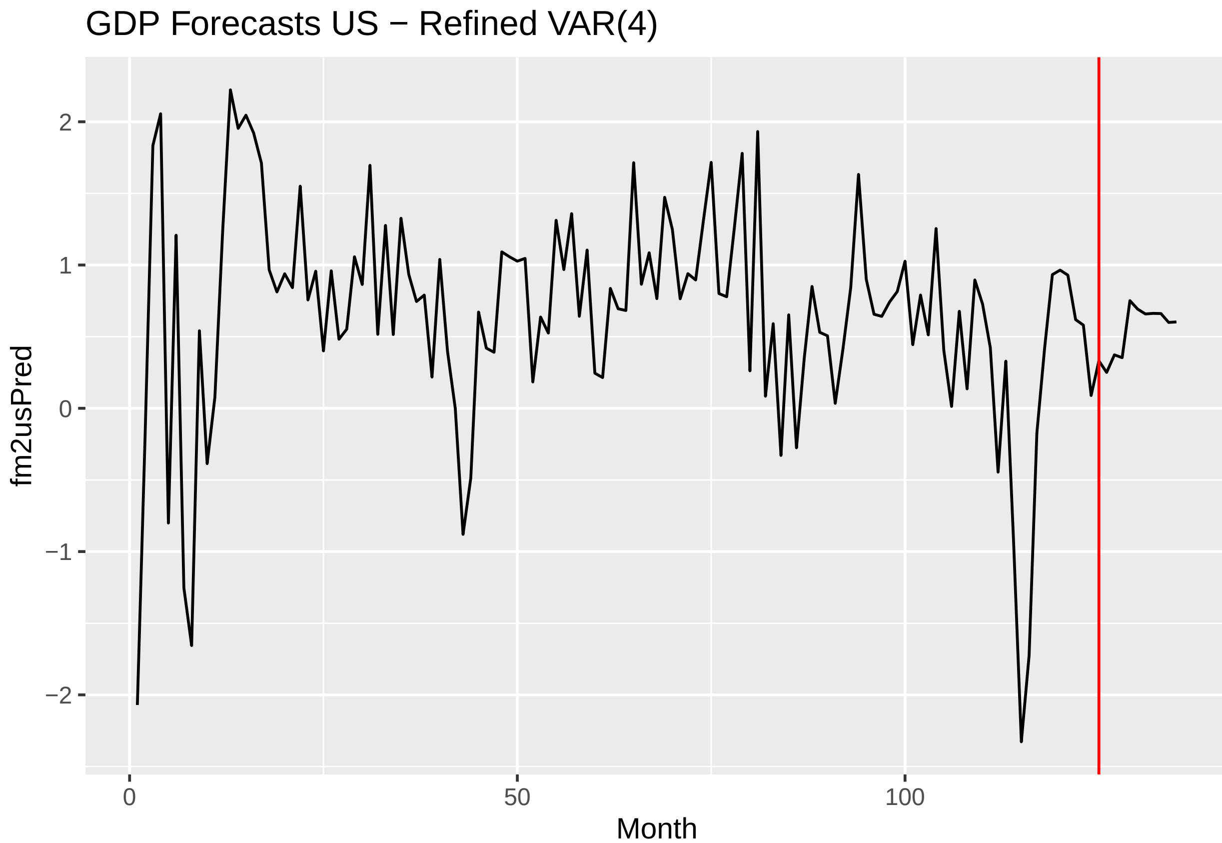
3. Report (20 points)

Write an executive summary of the outcomes of your GDP analysis.

(We will use the refined VAR(4) model for the executive summary, as we found it to have better/lower AIC and BIC scores.)

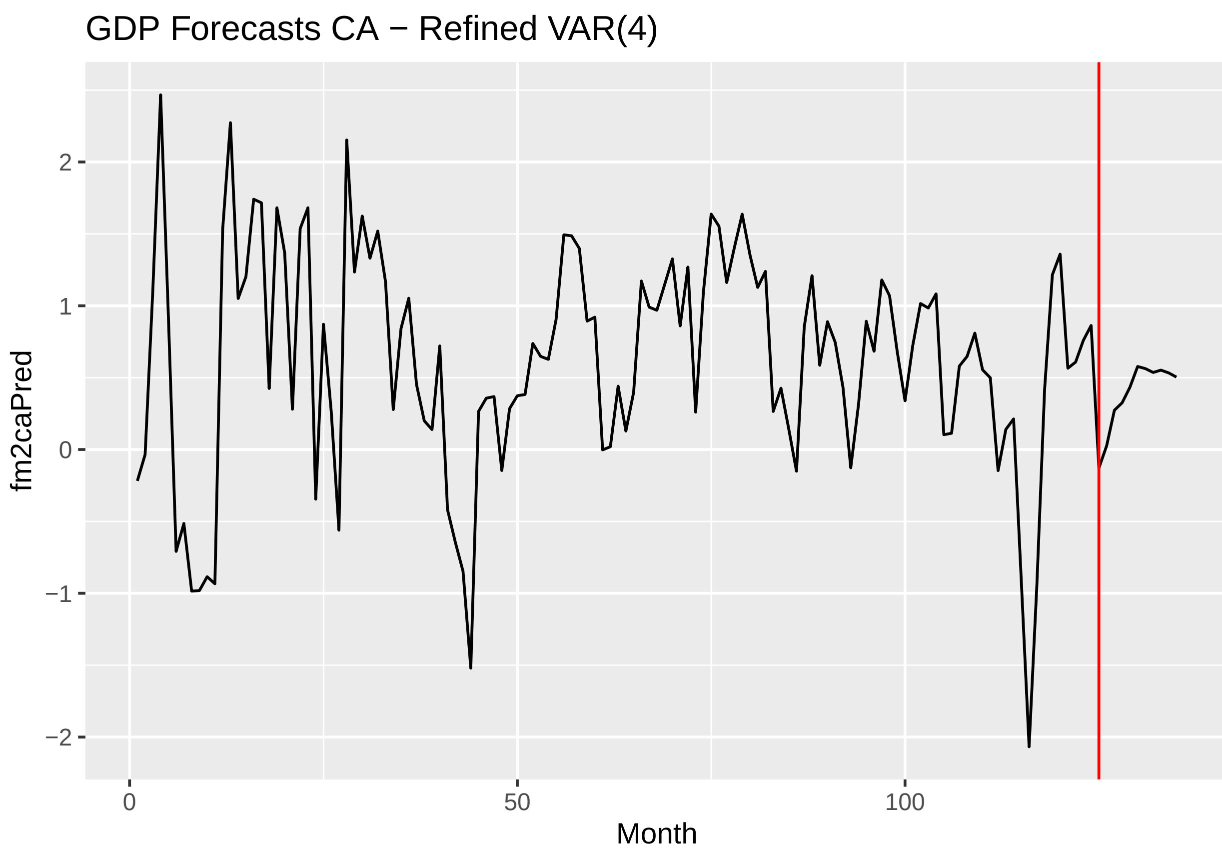

GDP Forecasts UK − Refined VAR(4)
fm2ukPred
Month
2
1
0
−1
−2
0
50
100

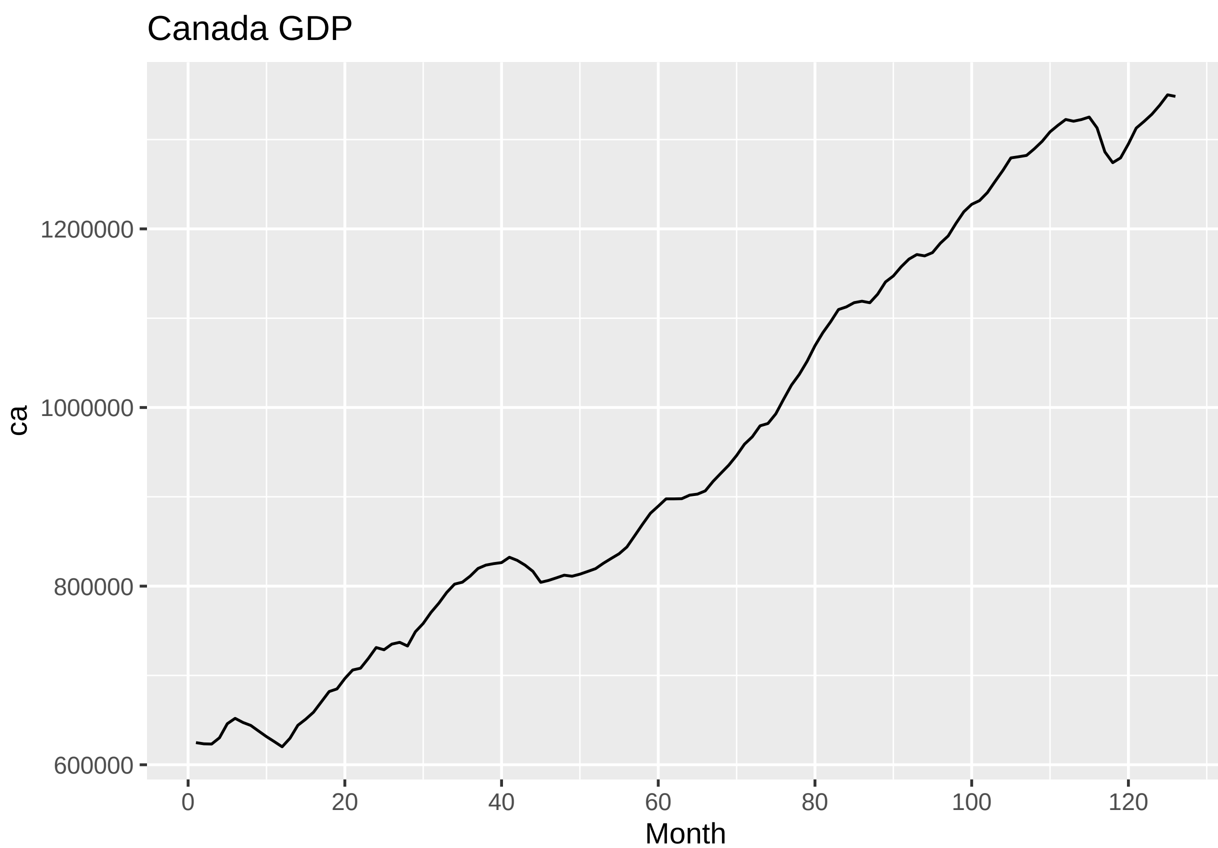

GDP Forecasts CA − Refined VAR(4)
fm2caPred
2
1
0
−1
−2
0
50
100
Month



The forecasting model we developed attempts to predict the gross domestic product (GDP) for the next 10 quarters in the United Kingdom, Canada, and the United States. The data is sourced from the the Multivariate Time Series (MTS) package available in the R programming language, which contains quarterly GDP data for the three countries from the first quarter of 1980 to the second quarter of 2011. The forecast is based on multivariate time series modelling which explores the possibility where the GDP of one country could be correlated to the GDP of other countries. While transforming the GDP data for each country to help fit our time series analysis, we present a forecast model based on the logged differenced transformation of each country's GDP data and the cross-correlation interactions found between the countries.

In each respective country's GDP plots above, the forecast is shown at the right side of the red vertical line. Generally speaking, the forecast for each country looks to have a small spike upwards within the first half of quarterly forecasts, then gently taper downwards towards the end of the second half. While we are forecasting 10 months of GDP predictions, we would recommend using this model as a short-term guide. That being said, this is a forecast of the transformed data and doesn't fully represent the original non-transformed data. But if we align the transformed data along with the GDP data, we can see large downward spikes in the transformed data coincide with dips in the GDP, while general volatility in the transformed data align with a general upward trend in the GDP. We can see this happen, for example, with the Canada GDP data below. As the forecast doesn't have large downward spikes we can at least not anticipate a dip in GDP in the short term.


GDP Forecasts CA − Refined VAR(4)
fm2caPred
2
1
0
−1
−2
0
50
100
Month



This is what we are able to present given the multivariate time series modelling tools we currently have available. We will continue to improve this model as our knowledge base expands to provide longer term and more accurate forecasts.