# Проект по временным рядам: Оценка мультиплакатора расходов Москвы

2024-05-26

```
library(lubridate)
library(haven)
library(dplyr)
library(openxlsx)
library(forecast)
library(vars)
library(tseries)
library(seasonal)
library(base)
library(BVAR)
library(urca)
library(modelsummary)
library(reshape2)
library(ggplot2)
library(tsDyn)
```

##Обзор литературы на тему векторных авторегрессий

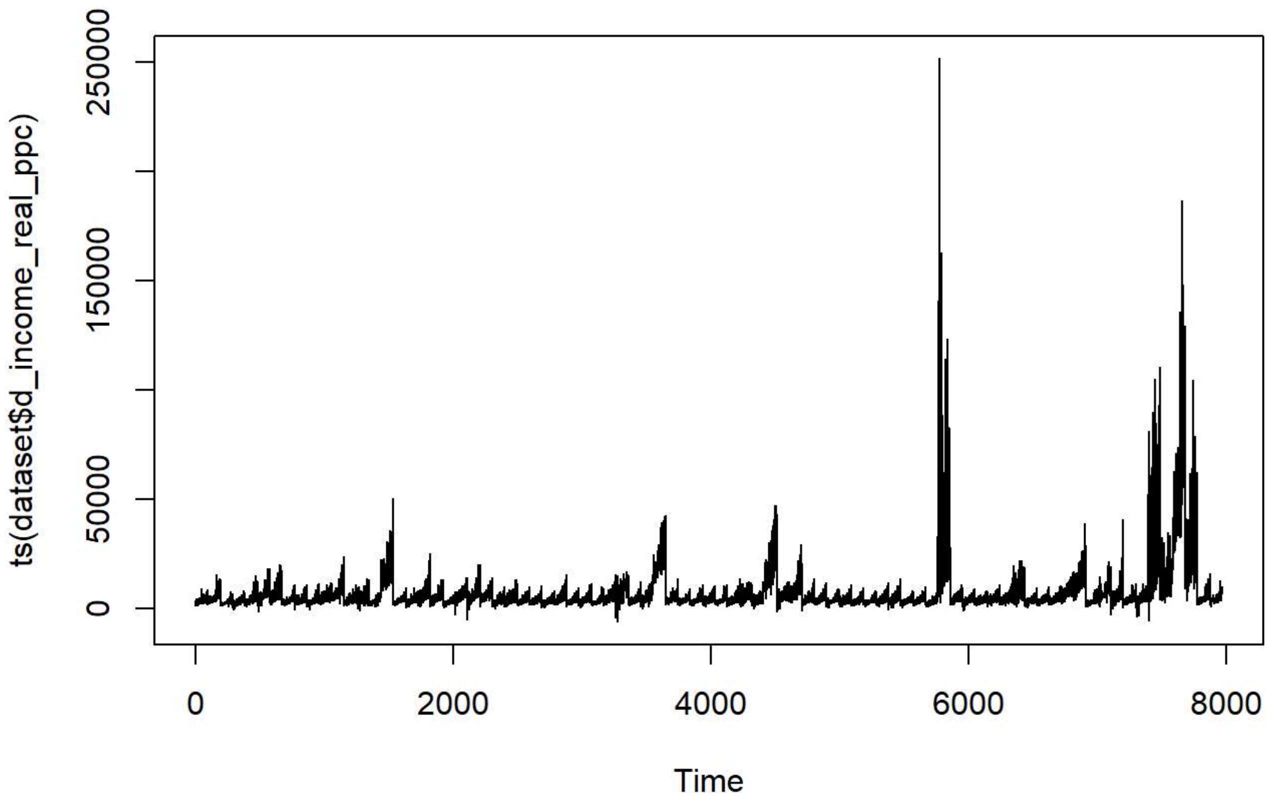
#VAR/ECV/VECV:: 1.А.Ю. Завьялов, Е.В. Нилова, Д.Н. Шульц, “ФИСКАЛЬНЫЕ МУЛЬТИПЛИКАТОРЫ РАСШИРЕННОГО БЮДЖЕТА РФ И СПОСОБЫ ИХ ОЦЕНИВАНИЯ” Строится модель VAR и ECM на квартальных данных по России с 2000 по 2013 гг.

#SVAR: 1. Зяблицкий И.Е 2020, “Оценка фискальных мультипликаторов в российской экономике” В статье строится SVAR для ВВП(расходы расширенного правительства, доходы расширенного правительства, ставка MIACR до 1 дня и рублевая цена на нефть Urals). В исследовании используются квартальные данные с 2004 по 2019 гг. 2. Алексей БАЛАЕВ, 2018, “ВЛИЯНИЕ СТРУКТУРЫ БЮДЖЕТНЫХ РАСХОДОВ НА ЭКОНОМИЧЕСКИЙ РОСТ В РОССИИ” В основе работы построение SVAR модели на ежегодных данных 2000-2017 гг. #BVAR: 1. А. И. Вотинов, И. П. Станкевич, 2017, “VAR-подход к оценке эффективности мер фискального стимулирования экономики” Модели SVAR, BVAR

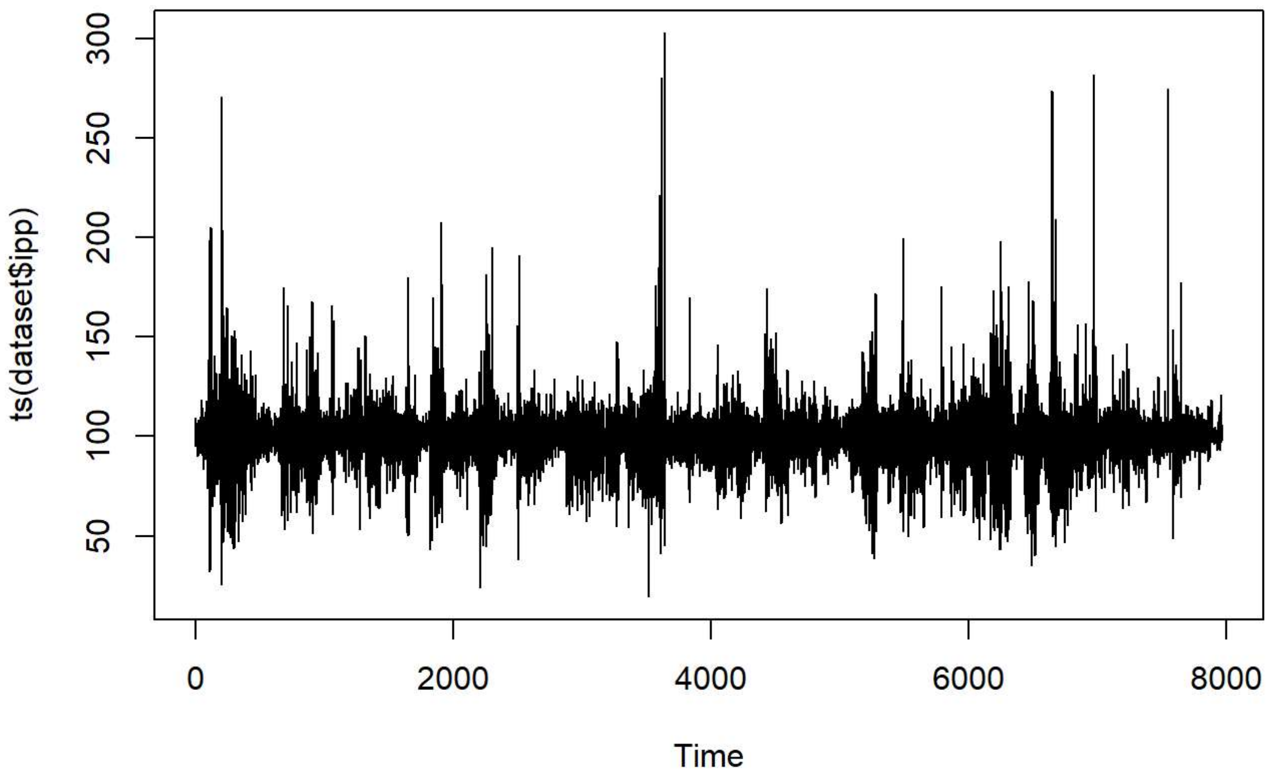
2. С.А. Власов, Е.Б. Дерюгина, 2018, “Фискальные мультипликаторы в России” байесовская векторная авторегрессионная модель на основе идентификации нулевыми и знаковыми ограничениями

```
setwd("C:/Users/annaz/Documents/R/Times series")
dataset <- read.xlsx("data.xlsx")
```

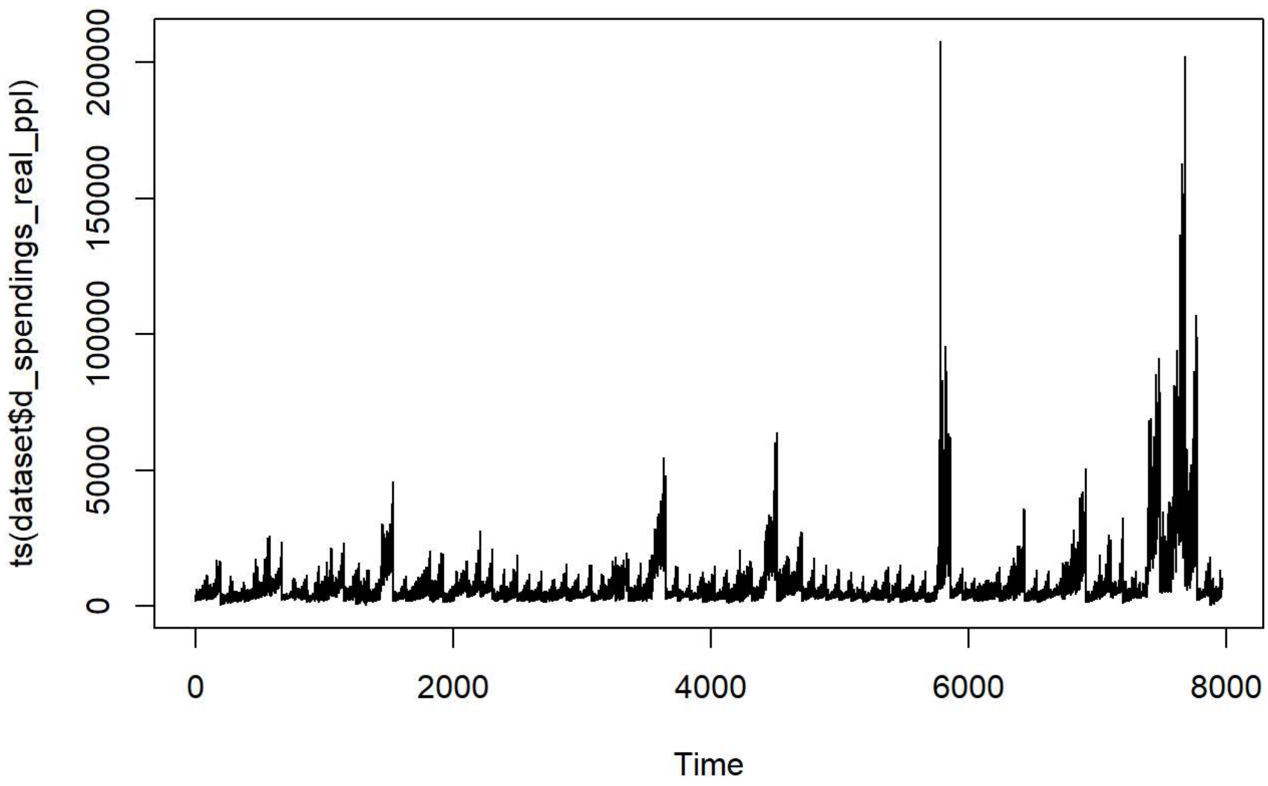
```
plot(ts(dataset$d_income_real_ppc))
```




```
plot(ts(dataset$ipp))
```



```
plot(ts(dataset$d_spendings_real_ppl))
```




## Seasonal adjustment

Месячные данные по расходам и доходам получены из отчетов регионов и могут иметь разные паттерны в определенные месяцы. Ниже мы пишем функцию для избавления от сезонности определенных переменных в панельном датасете. В экзогенных переменных включаем показатели ставок межбанковского кредитного рынка группы MIACR и цену на нефть марки Brent, основываясь на литературе.

```
## Model used in SEATS is different: (2 1 2)(0 1 1)
```

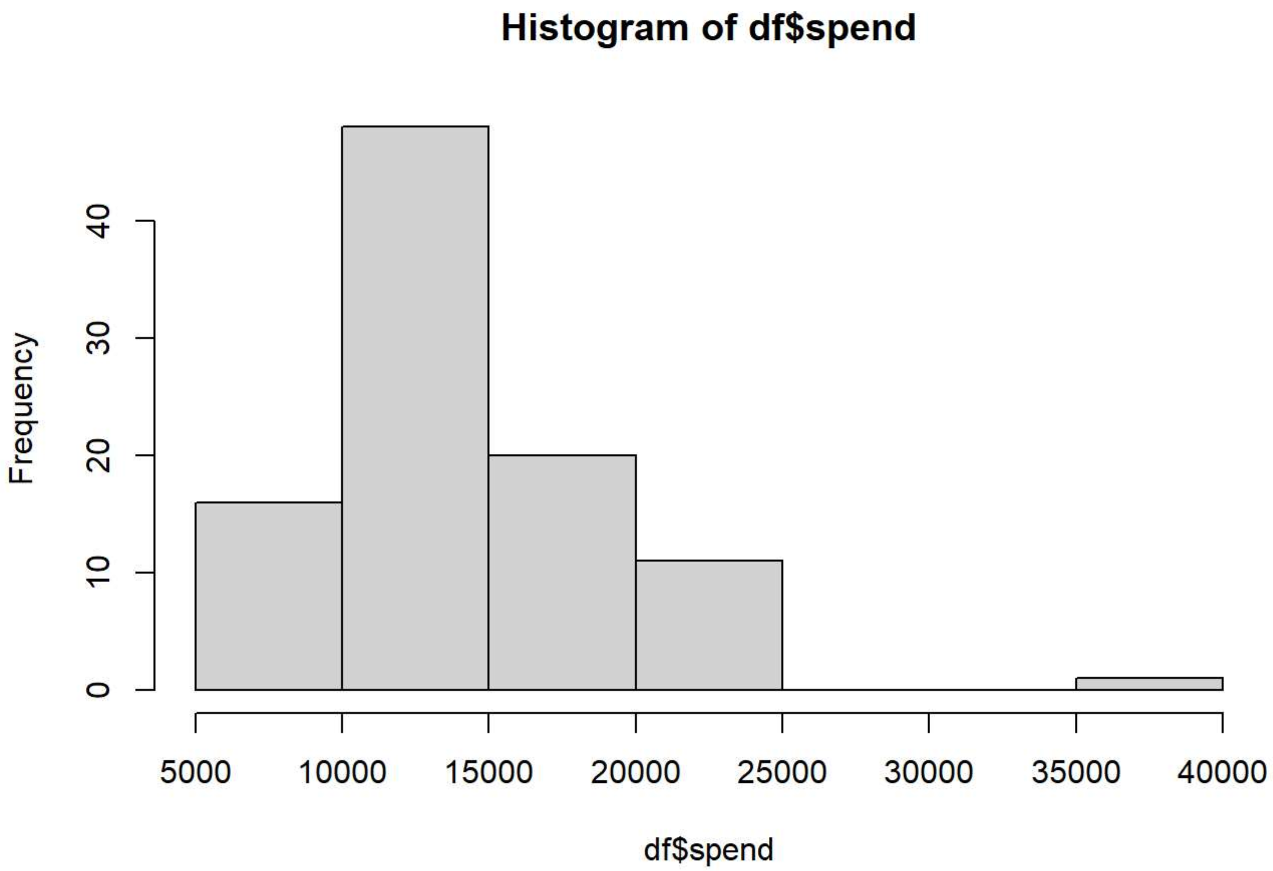
Возьмем в качестве изучаемого региона г. Москву.

```
#dataset <- read.xlsx("data.xlsx")
df = df %>% dplyr::filter(id == 73) %>%
            arrange(year, month, id)

TSspend = ts(df$spend, start = c(2014,1), frequency = 12)
TSincome = ts(df$income, start = c(2014,1), frequency = 12)
TSipp = ts(df$ipp, start = c(2014,1), frequency = 12)
```

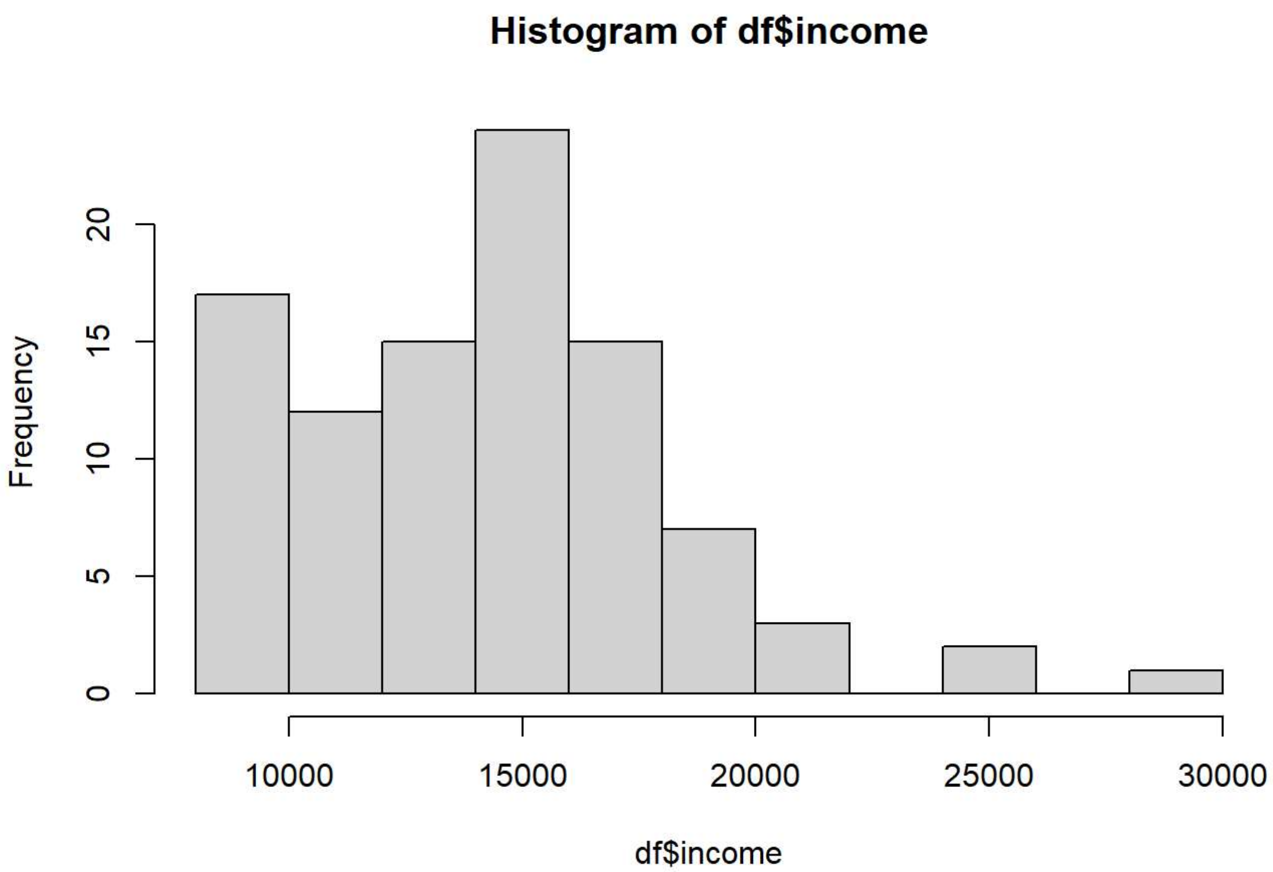
## Visualisation

Для устойчивости данных возмем показатели расходов и доходов в логарифм. На первый взгляд по графическому анализу ряд в логарифмах имеет тренд и не может быть стационарным

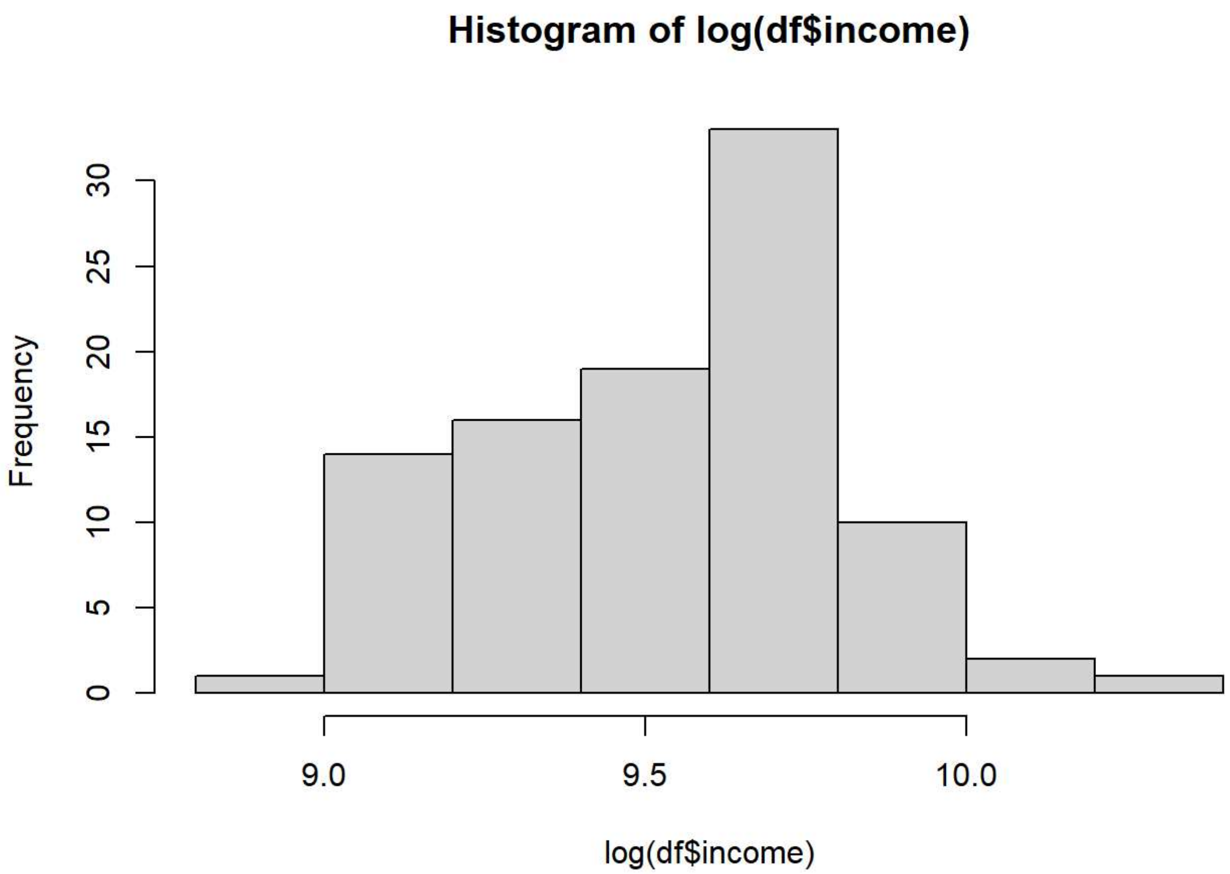
```
hist(df$spend)
```



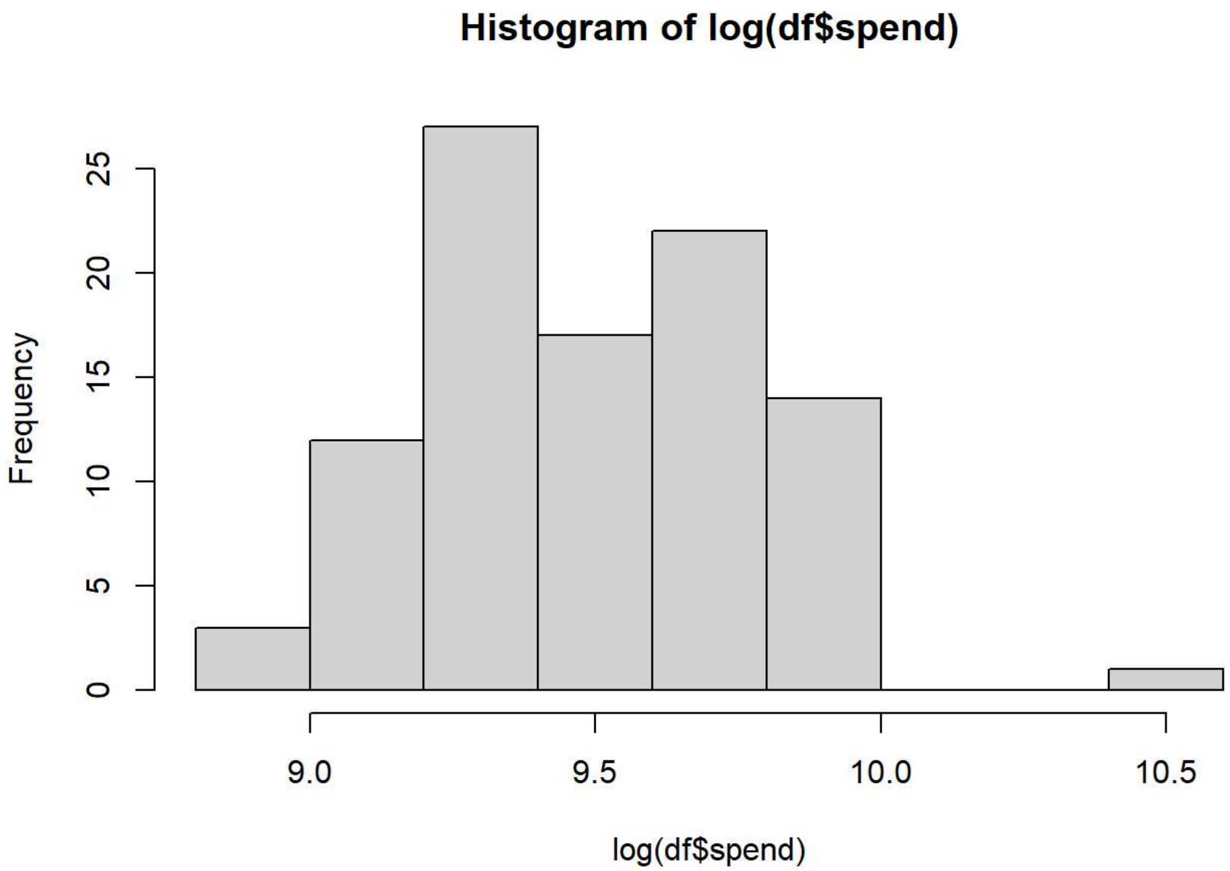

```
hist(df$income)
```



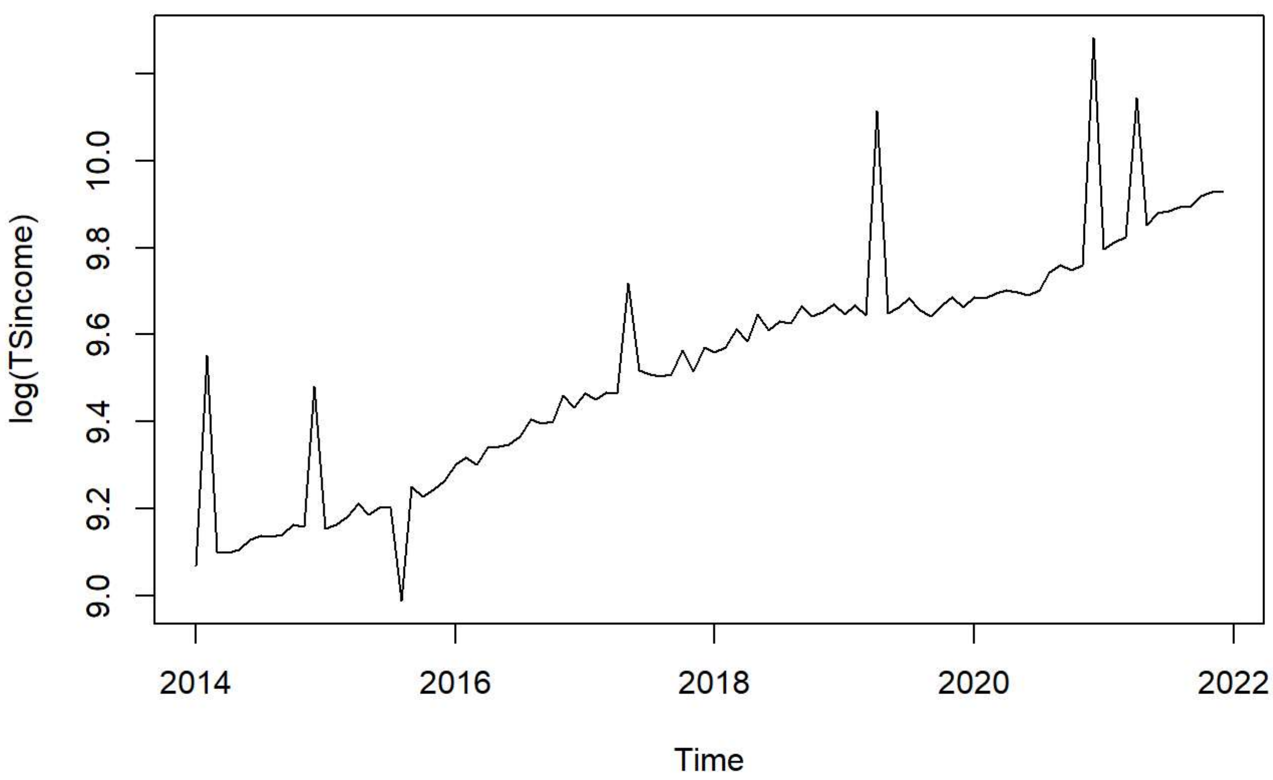
```
hist(log(df$income))
```



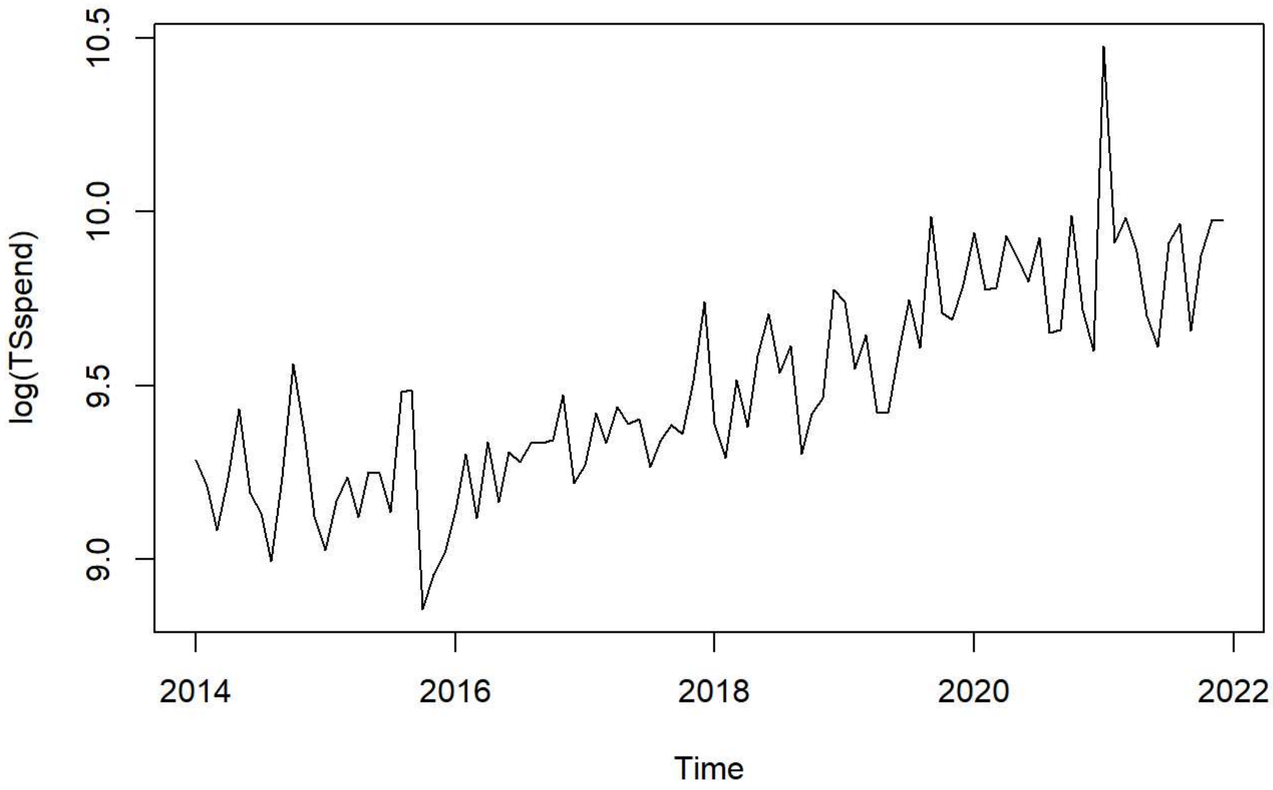

```
hist(log(df$spend))
```



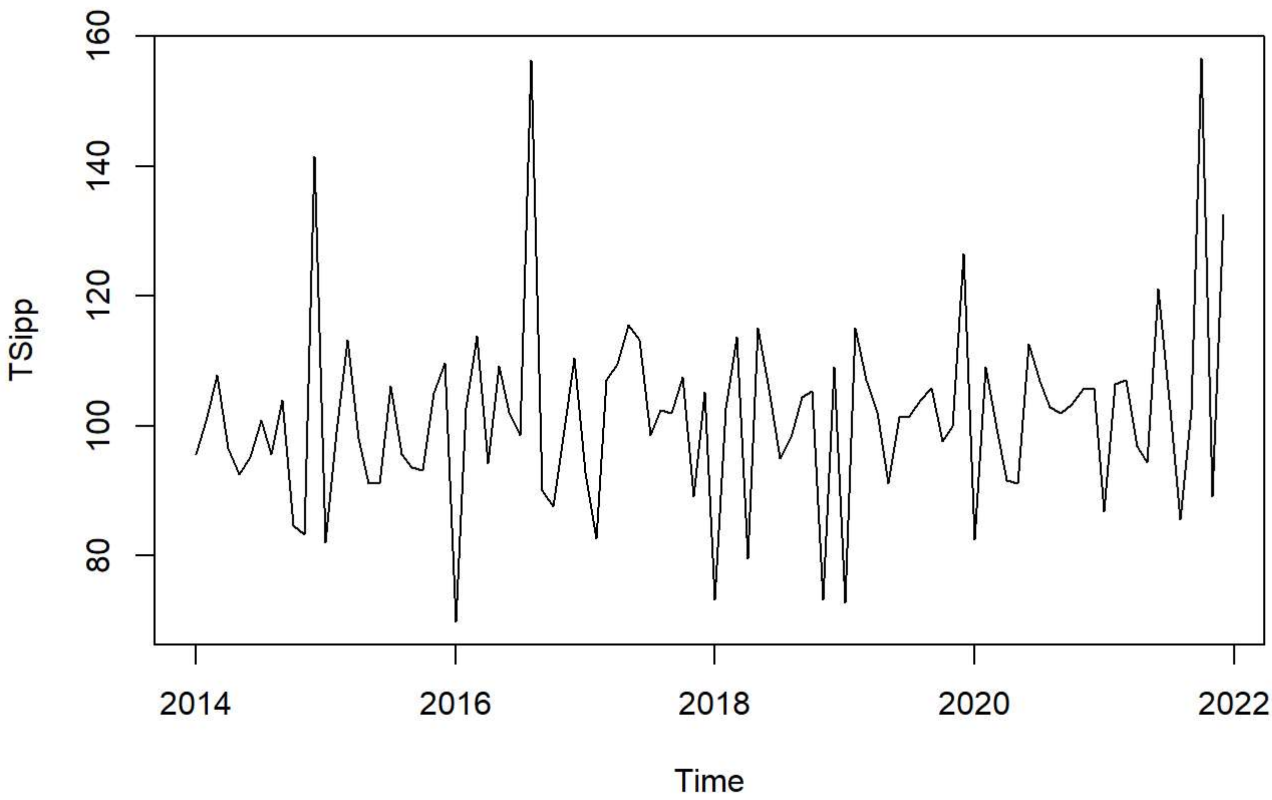

```
plot(log(TSincome))
```



```
plot(log(TSspend))
```




```
plot(TSipp)
```




### Stationarity check

Проверяем стационарность переменной ИПП и первых разниц логарифмов расходов и доходов.

```
adf.test(df$ipp) #p-value = 0.0439 - stationary on 5% significance
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  df$ipp
## Dickey-Fuller = -3.5239, Lag order = 4, p-value = 0.0439
## alternative hypothesis: stationary
```

```
pp.test(df$ipp) #p-value = 0.01 - stationary
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  df$ipp
## Dickey-Fuller Z(alpha) = -115.05, Truncation lag parameter = 3, p-value
## = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(df$ipp) #p-value = 0.1 - stationary on 5% significance
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  df$ipp
## KPSS Level = 0.32383, Truncation lag parameter = 3, p-value = 0.1
```

```
adf.test(diff(df$ipp)) #p-value = 0.01 - stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(df$ipp)
## Dickey-Fuller = -6.6242, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
pp.test(diff(df$ipp)) #p-value = 0.01 - stationary
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  diff(df$ipp)
## Dickey-Fuller Z(alpha) = -124.16, Truncation lag parameter = 3, p-value
## = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(df$ipp)) #p-value = 0.1 - stationary on 5% significance
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(df$ipp)
## KPSS Level = 0.047713, Truncation lag parameter = 3, p-value = 0.1
```

```
adf.test(diff(log(TSspend))) #p-value = 0.01 - stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(log(TSspend))
## Dickey-Fuller = -8.5227, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
pp.test(diff(log(TSspend))) #p-value = 0.01 - stationary
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  diff(log(TSspend))
## Dickey-Fuller Z(alpha) = -103.95, Truncation lag parameter = 3, p-value
## = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(log(TSspend))) #p-value = 0.1 - stationary on 5% significance
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(log(TSspend))
## KPSS Level = 0.034804, Truncation lag parameter = 3, p-value = 0.1
```

```
adf.test(diff(log(TSincome))) #p-value = 0.01 - stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(log(TSincome))
## Dickey-Fuller = -6.6554, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
pp.test(diff(log(TSincome))) #p-value = 0.01 - stationary
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  diff(log(TSincome))
## Dickey-Fuller Z(alpha) = -126.12, Truncation lag parameter = 3, p-value
## = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(log(TSincome))) #p-value = 0.1 - stationary on 5% significance
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(log(TSincome))
## KPSS Level = 0.028824, Truncation lag parameter = 3, p-value = 0.1
```

В итоге мы добились стационарности через взяти первой разности у логарифмов доходов и расходов. Для лучшей интерпретации возьмем первую разницу у ИПП тоже. Мы отобрали стационарные и отмасштабированные временные ряды для дальнейшего анализа

```
d_TSspend = diff(log(TSspend))
d_TSincome = diff(log(TSincome))
d_TSipp = diff(TSipp/100)
```

Очередность переменных выбираем по их важности, также для оценки мультипликатора расходов мы поместим из в порядке: ИПП, доходы, расходы

```
dataForVAR = data.frame(d_income = d_TSincome,
                     d_spend = d_TSspend,
                     d_ipp = d_TSipp)
dataExog = data.frame(BRENT = df$BRENT[2:96]/100,
                      MIACR = df$MIACR[2:96])

summary(dataForVAR)
```

```
##     d_income             d_spend              d_ipp          
##  Min.   :-0.484653   Min.   :-0.630068   Min.   :-0.674000  
##  1st Qu.:-0.009354   1st Qu.:-0.121583   1st Qu.:-0.098000  
##  Median : 0.009156   Median : 0.007806   Median : 0.001000  
##  Mean   : 0.009069   Mean   : 0.007248   Mean   : 0.003874  
##  3rd Qu.: 0.022075   3rd Qu.: 0.136100   3rd Qu.: 0.109500  
##  Max.   : 0.522296   Max.   : 0.873373   Max.   : 0.582000  
```

```
summary(dataExog)
```

```
##      BRENT            MIACR       
##  Min.   :0.2663   Min.   : 4.060  
##  1st Qu.:0.4922   1st Qu.: 6.680  
##  Median :0.6271   Median : 7.720  
##  Mean   :0.6572   Mean   : 8.243  
##  3rd Qu.:0.7614   3rd Qu.:10.185  
##  Max.   :1.2284   Max.   :16.960  
```

Models

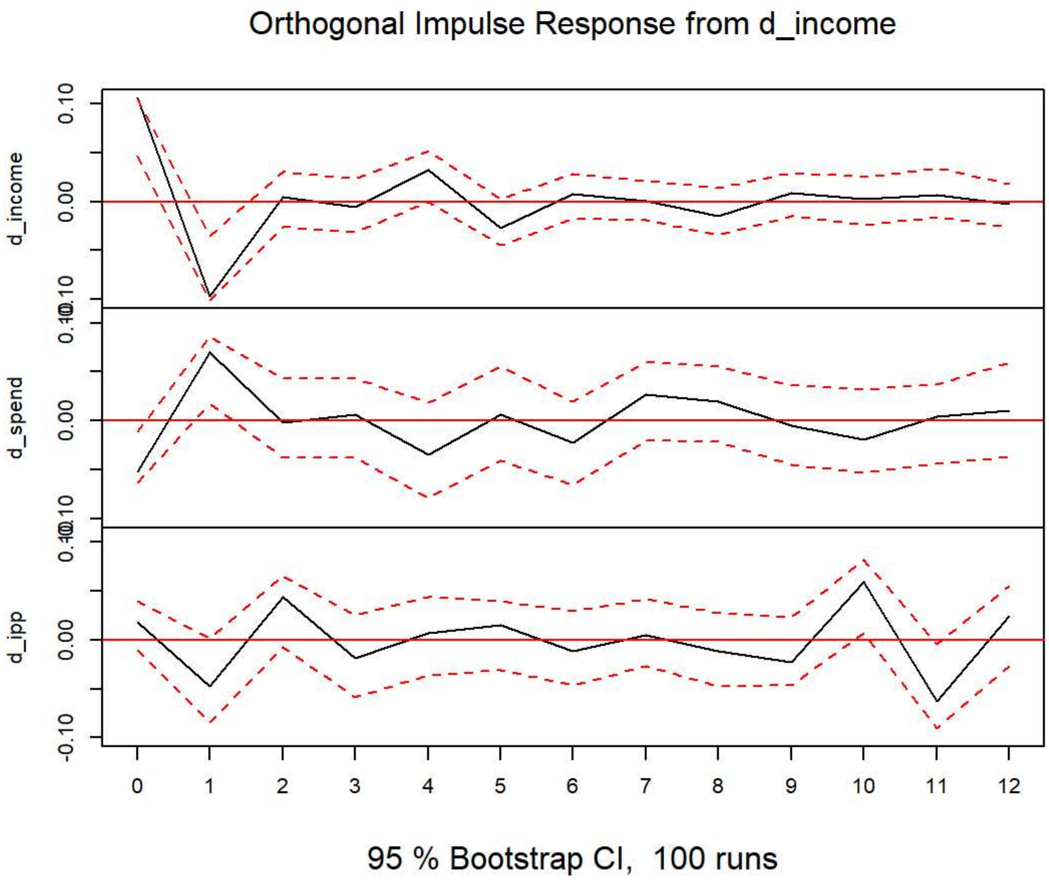
# VAR

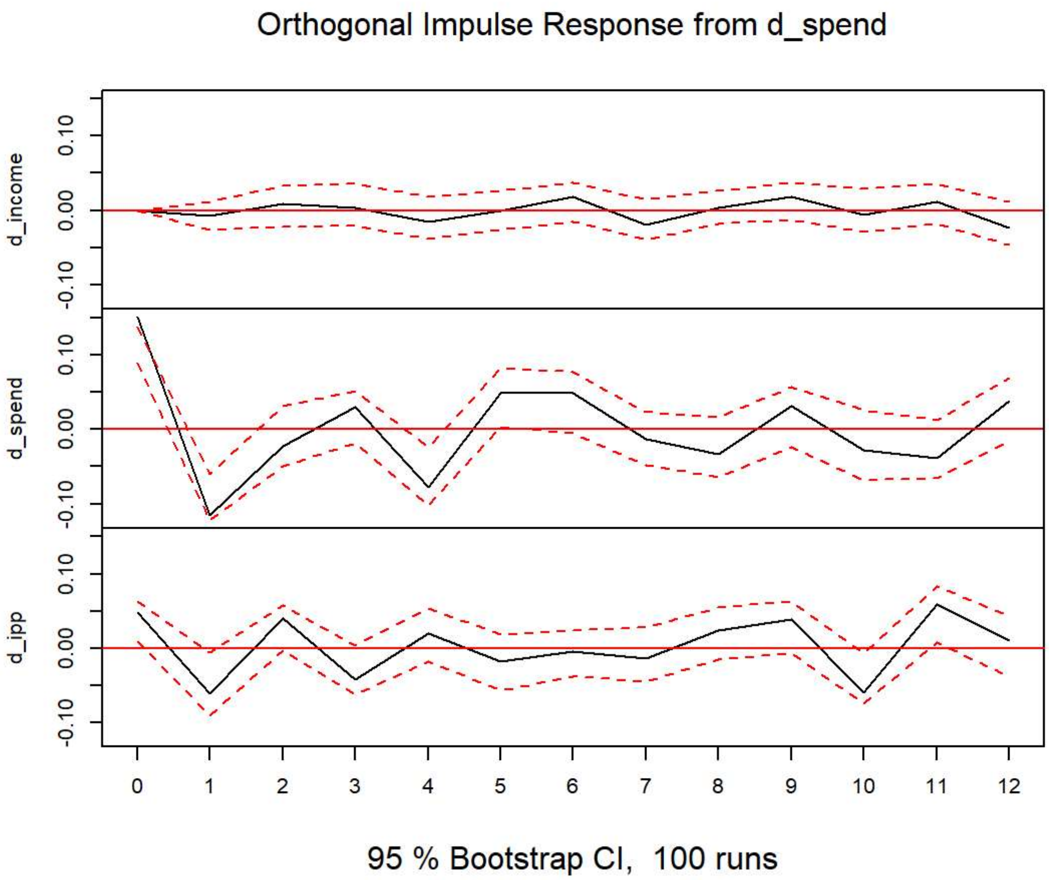
Проверим модель обычной векторной авторегрессии, для этого оценим по информационным критериям какой лаг лучше взять.

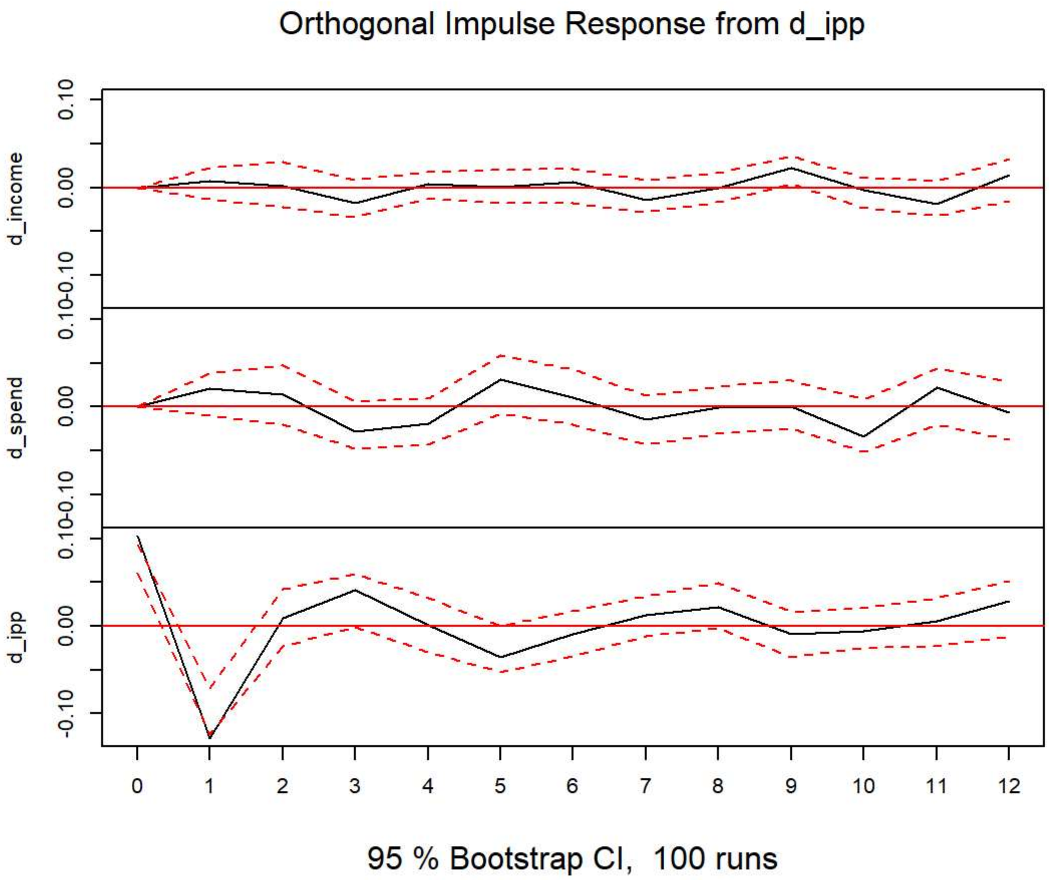
```
m0 <- VARselect(dataForVAR,lag.max = 12, type = "both", exogen = dataExog)
m0$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n) 
##     11      3      2     11 
```

```
m1 <- VAR(dataForVAR, p=11, type = "both", exogen = dataExog)
ORF_o <- vars::irf(m1, n.ahead = 12)
plot(ORF_o)
```

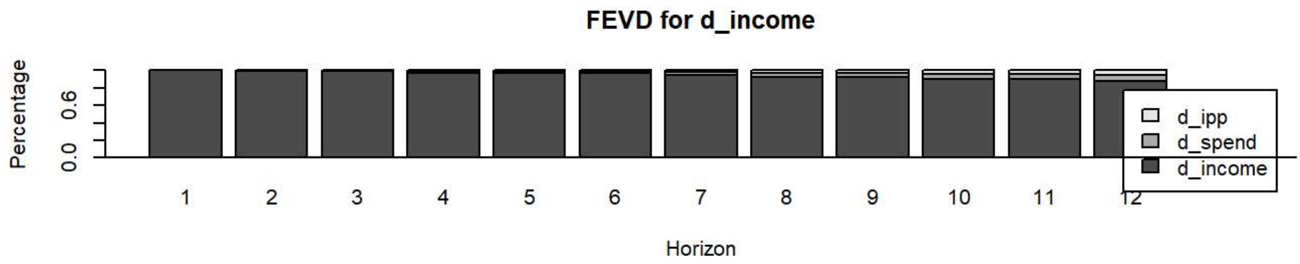

Orthogonal Impulse Response from d_income
d_income
d_spend
d_ipp
0.10
0.00
-0.10
0 1 2 3 4 5 6 7 8 9 10 11 12
95 % Bootstrap CI, 100 runs

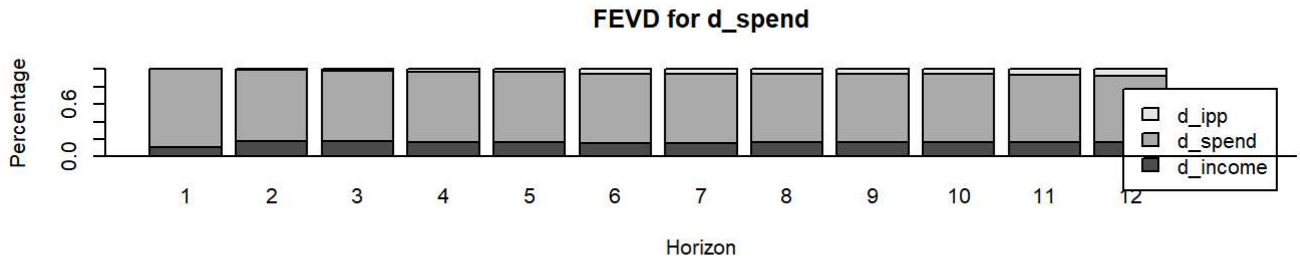

Orthogonal Impulse Response from d_spend
d_income
d_spend
d_ipp
0.10
0.00
-0.10
0 1 2 3 4 5 6 7 8 9 10 11 12
95 % Bootstrap CI, 100 runs

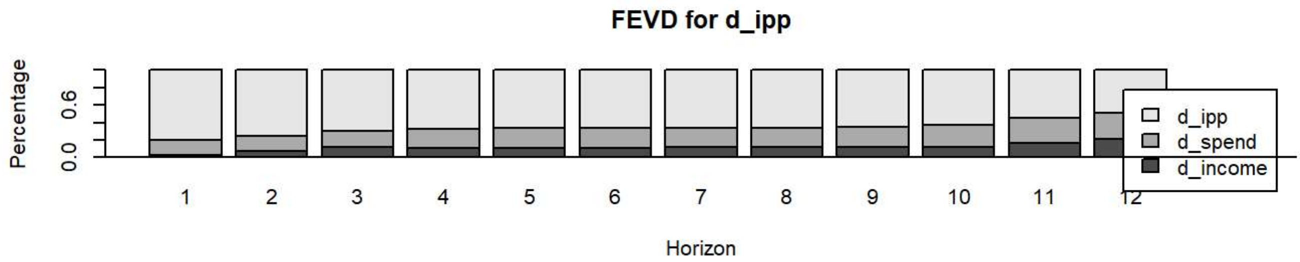



Все переменные влияют на себя в первом периоде, к тому же есть эффект изменения ИПП на спад процентного изменения расходов в 4 периоде и эффект доходов на ИПП в конце года, а на расходы в 1 периоде

```
FEVD_o <- vars::fevd(m1, n.ahead = 12)
plot(FEVD_o)
```




Процентное влияние каждой переменной на себя уменьшается со временем. Для ИПП влияние расходов проявляется со второго периода, а доходов с третьего, но суммарно эффекты меньше 40% к 12му периоду. Для Доходов влияние расходов проявляется со третьего периода, а ИПП с первого и поднимается до 10% к десятому периоду, но суммарно эффекты меньше 20% Для Расходов влияние доходов и ИПП проявляется с первого периода, но эффект доходов постоянно около 15%, а ИПП ближе к 10% только с 6го периода

## SVAR

В литературе в основном используют структурные авторегрессии для оценки мультипликаторов расходов государств. Главными переменными являются расходы, доходы и ввп(врп), который мы заменили на ИПП из-за отсутсвия месячных региональных данных. Авторы использую модели SVAR с структурными шоками и нулевой матрицей А.

Матрица мгновенных откликов с коэффициентами в В, включает в себя: реакцию роста ИПП на неожиданный шок изменения доли доходов в модели, реакцию роста изменения доли расходов на неожиданный шок изменения ИПП в модели, реакцию роста изменения доли доходов на неожиданный шок изменения доли расходов в модели,

```
##B model SVAR, using for shocks
B_matrix_1 <- diag(1, nrow = 3)
B_matrix_1[2,1]<- NA   ### immediate effect of ipp on d_income
B_matrix_1[3,2] <- NA  ### immediate effect of d_income on d_spend
B_matrix_1[1,3] <- NA  ### immediate effect of d_spend on ipp
B_matrix_1
```

```
##      [,1] [,2] [,3]
## [1,]    1    0   NA
## [2,]   NA    1    0
## [3,]    0   NA    1
```

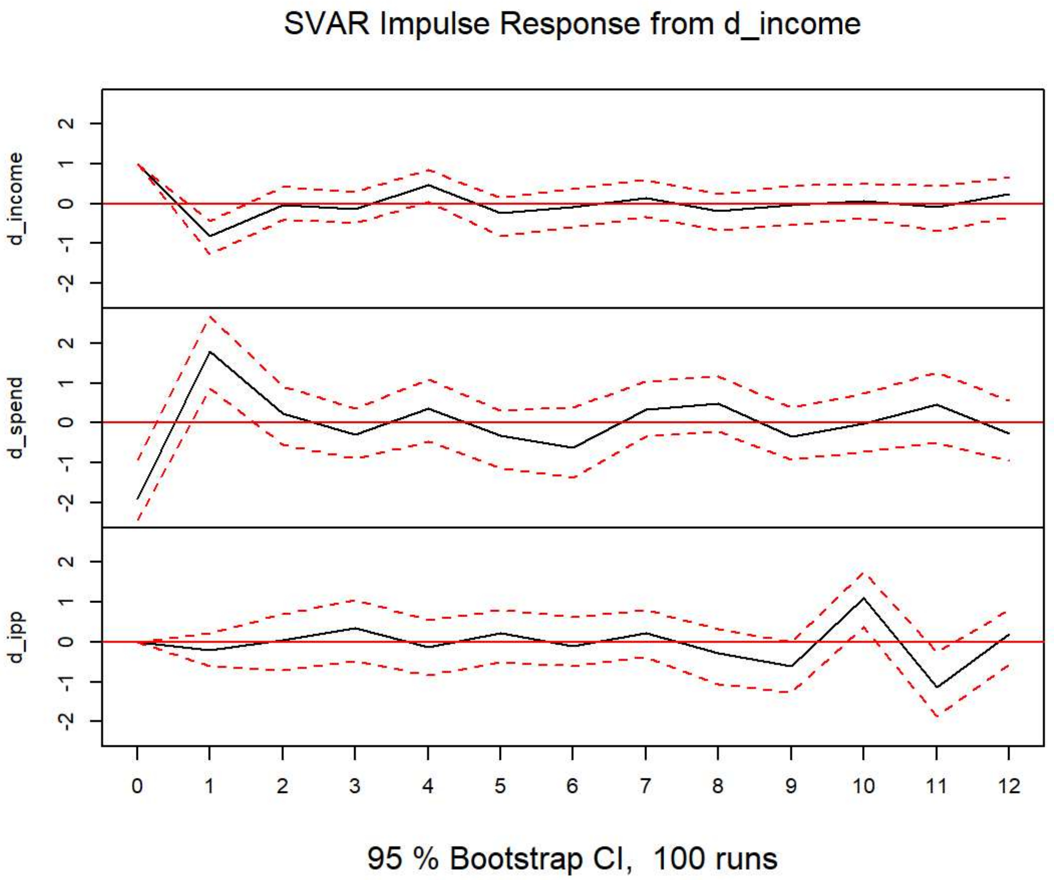
```
m3 <- SVAR(m1, Bmat = B_matrix_1, estmethod = "direct", max.iter = 1000)

summary(m3)
```

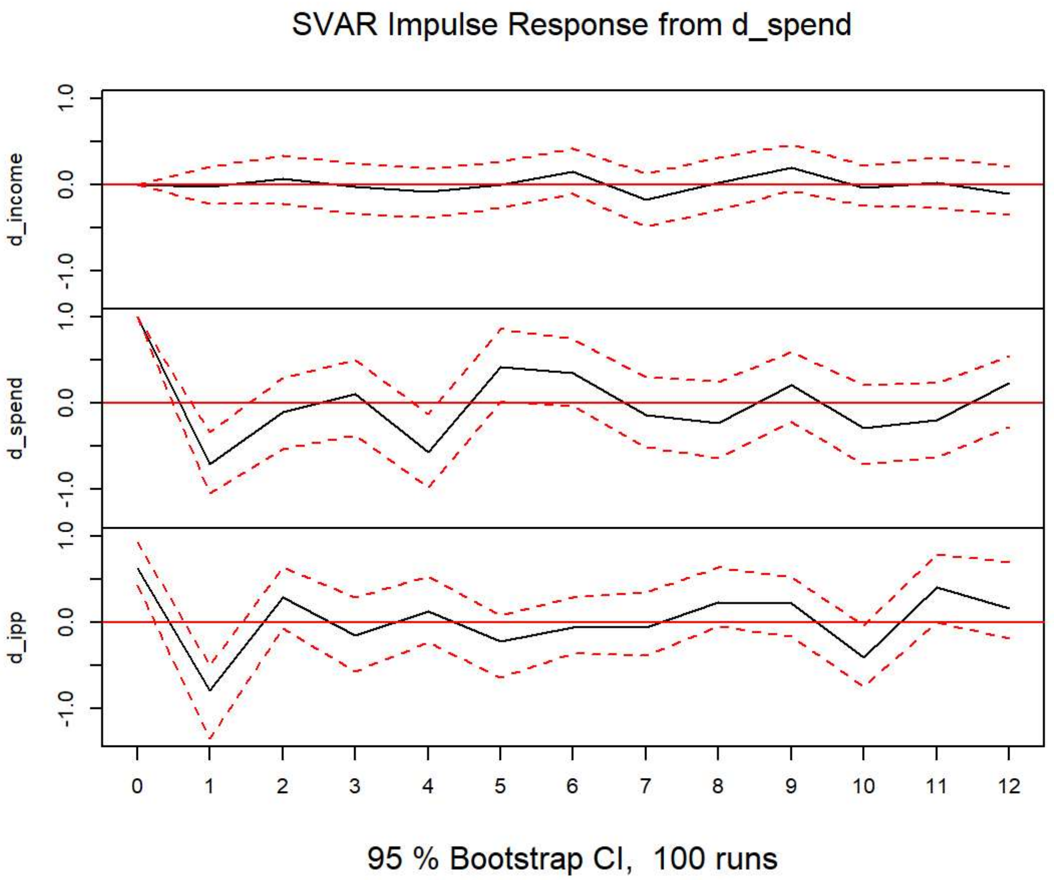
```
##
## SVAR Estimation Results:
## ========================
##
## Call:
## SVAR(x = m1, estmethod = "direct", Bmat = B_matrix_1, max.iter = 1000)
##
## Type: B-model
## Sample size: 84
## Log Likelihood: -268.225
## Method: direct
## Number of iterations: 42
## Convergence code: 10
##
## LR overidentification test:
##
##  LR overidentification
##
## data:  dataForVAR
## Chi^2 = 897, df = 3, p-value <2e-16
##
##
## Estimated A matrix:
##          d_income d_spend d_ipp
## d_income        1       0     0
## d_spend         0       1     0
## d_ipp           0       0     1
##
## Estimated B matrix:
##          d_income d_spend  d_ipp
## d_income    1.000  0.0000 0.5463
## d_spend    -1.901  1.0000 0.0000
## d_ipp       0.000  0.6304 1.0000
##
## Covariance matrix of reduced form residuals (*100):
##          d_income d_spend  d_ipp
## d_income   129.84 -190.15  54.63
## d_spend   -190.15  461.56  63.04
## d_ipp       54.63   63.04 139.74
```

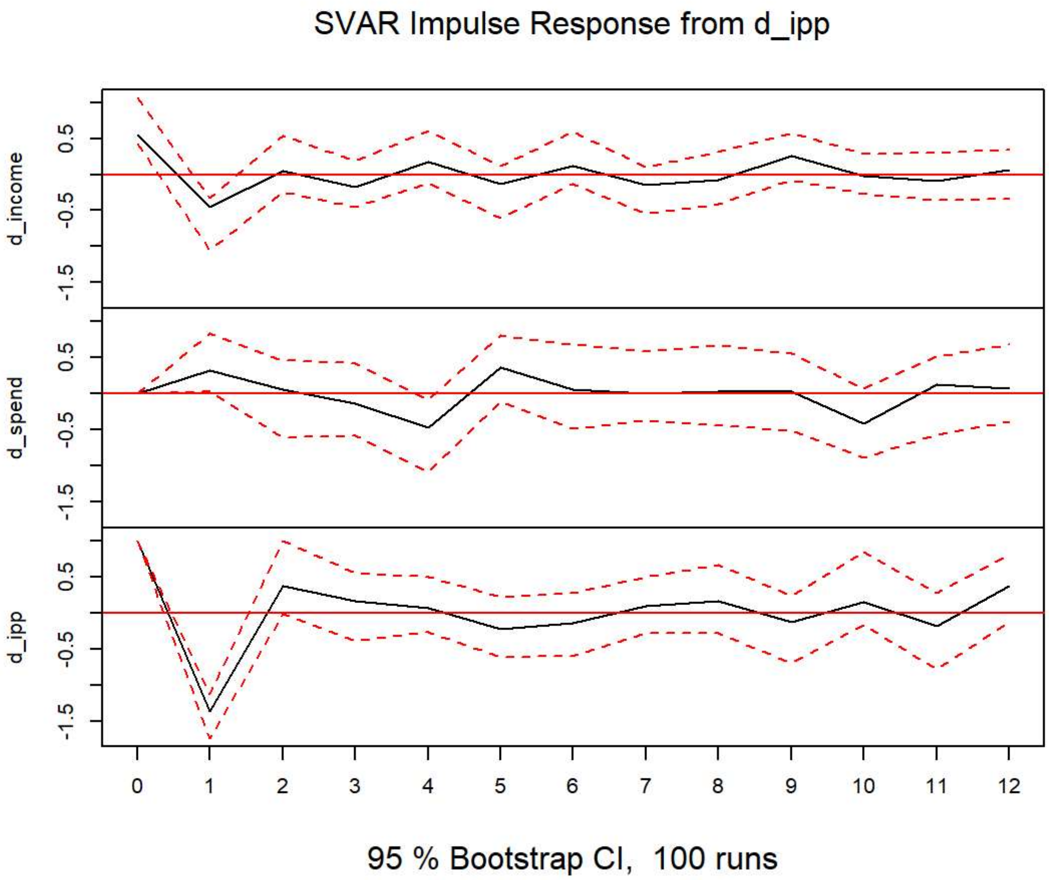
```
m3$B
```

```
##           d_income   d_spend     d_ipp
## d_income  1.000000 0.0000000 0.5462963
## d_spend  -1.901481 1.0000000 0.0000000
## d_ipp     0.000000 0.6303704 1.0000000
```

```
IRF_o_B <- vars::irf(m3, n.ahead = 12)
plot(IRF_o_B)
```

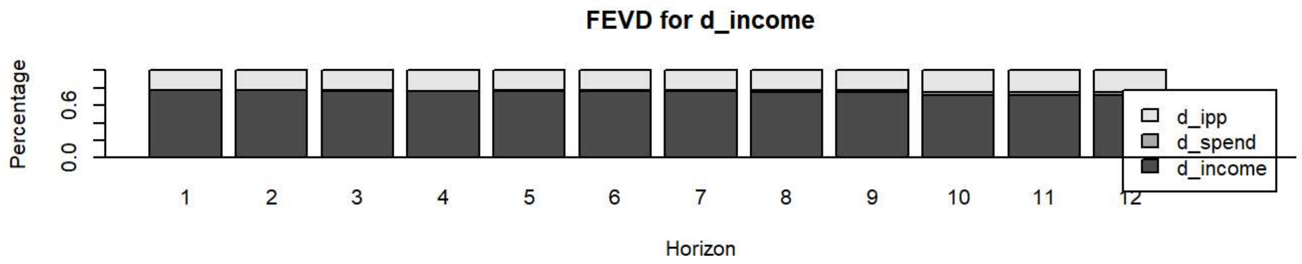

SVAR Impulse Response from d_income
d_income
d_spend
d_ipp
2
1
0
-1
-2
0 1 2 3 4 5 6 7 8 9 10 11 12
95 % Bootstrap CI, 100 runs

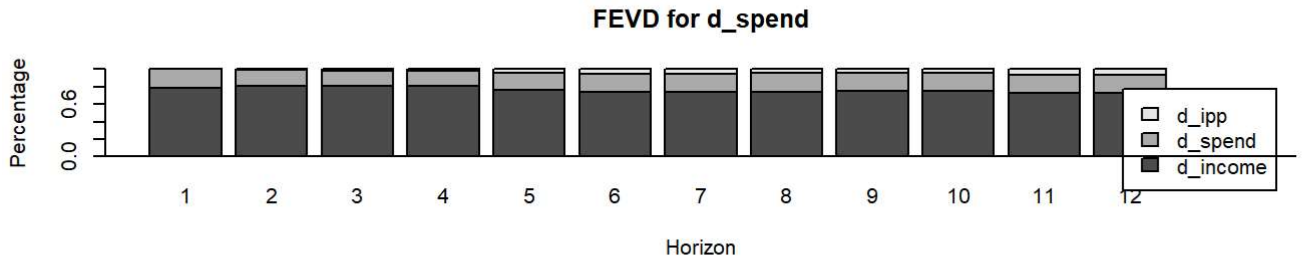

SVAR Impulse Response from d_spend
d_income
d_spend
d_ipp
1.0
0.0
-1.0
0 1 2 3 4 5 6 7 8 9 10 11 12
95 % Bootstrap CI, 100 runs

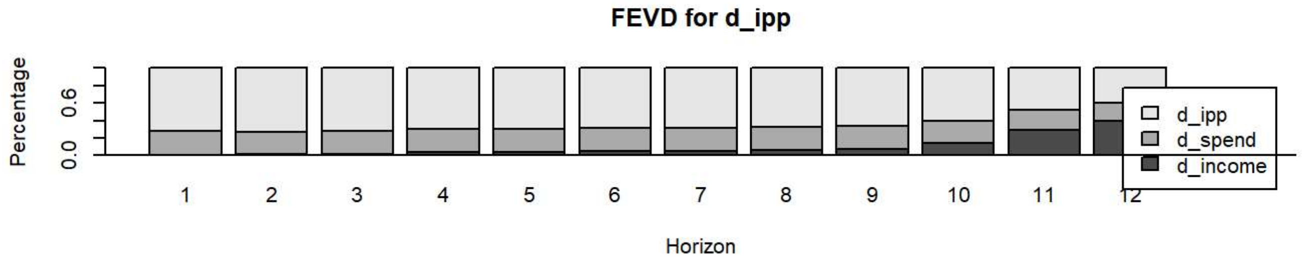



Эффект переменных на себя в первом лаге всё так же значим, к тому же изменение ИПП влияет на изменение доходов для первого лага, доходы положительно влияют на расходы для первого лага и ИПП за 12 месяцев назад, а расходы отрицательно влияют на ИПП для первого лага.

```
plot(vars::fevd(m3, n.ahead = 12))
```




В процентном соотношении влияние изменения расходов на 1% и изменения Индекса производства на изменение самого Индекса промышленного производства соотносятся как 3:7 в 1м периоде, далее со второго появляется эффект расходов и увеличивается до 20% к 12му периоду, а эффект ИПП и доходов будет по 40%

Влияние промышленного индекса на процентное изменение доходов 40% практически не меняется за год, в то время как влияние расходов появлется на третьем периоде и увеличивается до 5% к 12му периоду

На расходы сами на себя влияют одинаково около 20% в течение года, доходы же влияют значительно, а влияние ИПП появляется со второго периода и смещает влияние дохода на 5%

###VECM

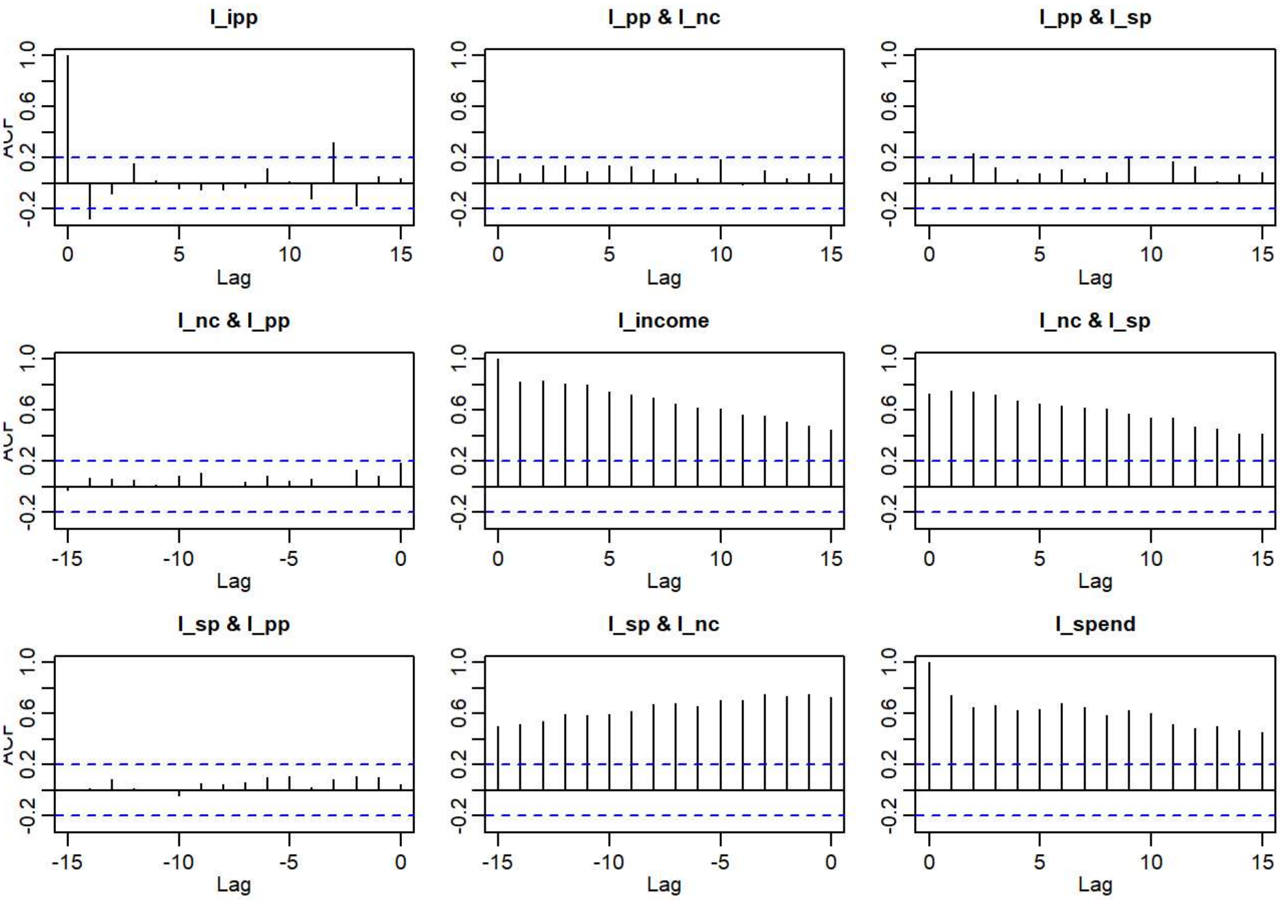
```
##Не используем разницы переменных, т.к. нам нужен I(0)0, но используем логарифмы для простот
ы интерпретации результатов
dataForVECM = data.frame(l_ipp = log(TSipp),
                         l_income = log(TSincome),
                         l_spend = log(TSspend))

j_obj <- ca.jo(dataForVECM, K = 9)
summary(j_obj)
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: maximal eigenvalue statistic (lambda max) , with linear trend
##
## Eigenvalues (lambda):
## [1] 0.2266025842 0.0820593961 0.0009602809
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 2 |  0.08  6.50  8.18 11.65
## r <= 1 |  7.45 12.91 14.90 19.19
## r = 0  | 22.36 18.90 21.07 25.75
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                l_ipp.l9 l_income.l9 l_spend.l9
## l_ipp.l9     1.00000000   1.0000000  1.0000000
## l_income.l9  0.09214857  -0.7552717 -0.1994325
## l_spend.l9  -0.13200575   0.5814179  0.6326183
##
## Weights W:
## (This is the loading matrix)
##
##             l_ipp.l9 l_income.l9   l_spend.l9
## l_ipp.d    -1.915811  -0.2106596 -0.009566829
## l_income.d -0.332210   0.2301702 -0.018466415
## l_spend.d   1.618609  -0.4867604 -0.009776248
```

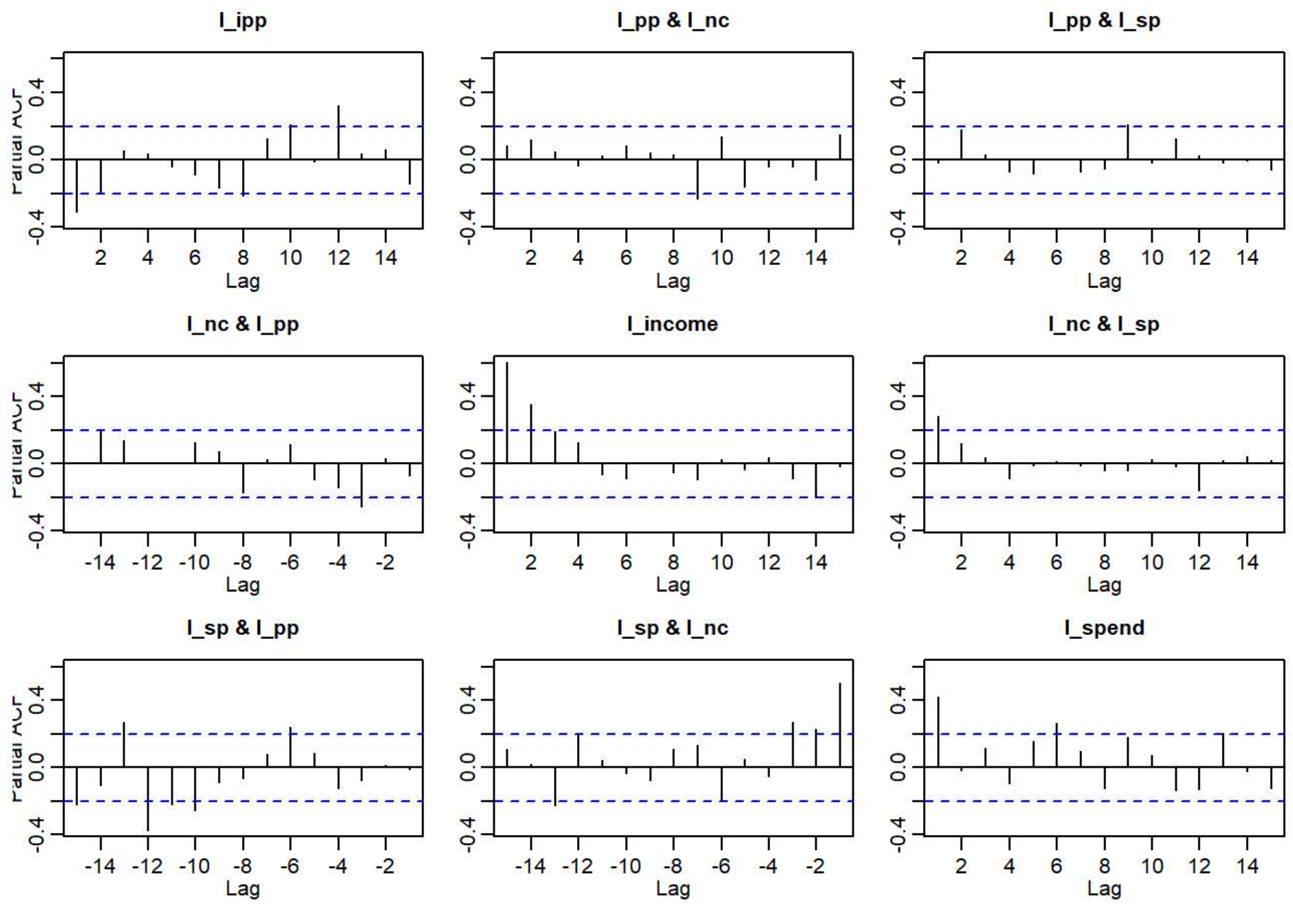
```
## Т.к.значение тестовой статистики при r<=2 равно 0.08, что меньше любого критического значе
ния при всех уровнях значимости, то мы принимаем нулевую гипотезу, т.е. ранг матрицы <=2, но
т.к. значение тестовой статистики при r<=1 равно 7.45, то мы принимаем гипотезу, что ранг <=
1,а при r=0, нулевая гипотеза отвергается на 5% уровне значимости, следовательно ранг матрицы
П = 1, значит в наших данных есть одно коинтеграционное соотношение (1 долгосрочная связь)
# При наличии коинтеграционной связи между переменными необходимо использовать VECM модель

cointRel <- ts(dataForVECM$l_ipp + 0.09214857*dataForVECM$l_income - 0.13200575 *dataForVECM
$l_spend)
adf.test(cointRel) ##Стационарны на 5% уровне значимости
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  cointRel
## Dickey-Fuller = -3.9587, Lag order = 4, p-value = 0.01441
## alternative hypothesis: stationary
```

```
##Изучив литературу по оценки мультипликаторов гос расходов в России на квартальных данных, м
ы увидели, что чаще всего используется лаг равный 3 и 4 кварталам. На ежемесячных данных это
аналогично лагу 9-12 месяцев.
## Построив графики автокорреляционной функции и частной автокорреляционной функции для наших
данных, мы больше склоняемся к лагу 12
acf(dataForVECM)
```




```
pacf(dataForVECM)
```




```
##Однако опираясь на информационные критерии, расчитанные автоматически из модели VAR, можно сделать вывод о влиянии лага 11
m0$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     11      3      2     11
```

```
## Для VECM необходимо использовать лаг равный лагу VAR модели - 1,т.к. VECM модель считает разницы, т.е. будем использовать лаг = 10
##Сформируем датафрейм с экзогенными переменными для модели VECM
dataExogen = data.frame(BRENT = df$BRENT/100,
                        MIACR = df$MIACR)


##VECM-модель, оцененная ML, с лагом 11, с одной коинтеграцией и энзогенными переменными:
m3 <-  VECM(dataForVECM,lag=10, r=1, estim = "ML", exogen = dataExogen)
summary(m3)
```

```
## #############
## ###Model VECM
## #############
## Full sample size: 96     End sample size: 85
## Number of variables: 3   Number of estimated slope parameters 102
## AIC -983.4845    BIC -729.4488   SSR 2.662685
## Cointegrating vector (estimated by ML):
##    l_ipp   l_income    l_spend
## r1     1 -0.2383506 -0.1029086
##
##
##                   ECT                Intercept          l_ipp -1
## Equation l_ipp    -0.8629(0.4033)*   1.1322(0.5132)*    -0.4079(0.3885)
## Equation l_income 0.8078(0.3539)*    -0.9983(0.4503)*   -0.7463(0.3409)*
## Equation l_spend  0.0526(0.5378)     0.0608(0.6844)     0.0561(0.5181)
##                   l_income -1        l_spend -1          l_ipp -2
## Equation l_ipp    -0.4279(0.1642)*   -0.1384(0.1102)     -0.4836(0.3625)
## Equation l_income -0.8363(0.1441)*** 0.0103(0.0967)      -0.5554(0.3180).
## Equation l_spend  0.3033(0.2190)     -0.7484(0.1470)***  0.1893(0.4833)
##                   l_income -2         l_spend -2         l_ipp -3
## Equation l_ipp    -0.1283(0.2132)     0.0636(0.1262)     -0.2737(0.3363)
## Equation l_income -0.7195(0.1871)***  0.0162(0.1107)     -0.5188(0.2950).
## Equation l_spend  0.3761(0.2844)      -0.7152(0.1683)*** 0.1266(0.4484)
##                   l_income -3         l_spend -3          l_ipp -4
## Equation l_ipp    -0.1981(0.2319)     -0.0233(0.1416)     -0.0888(0.3111)
## Equation l_income -0.5924(0.2034)**   0.0288(0.1243)      -0.5117(0.2730).
## Equation l_spend  0.6407(0.3092)*     -0.4842(0.1889)*    -0.0252(0.4148)
##                   l_income -4         l_spend -4          l_ipp -5
## Equation l_ipp    -0.2390(0.2504)     -0.0744(0.1435)     -0.0558(0.3003)
## Equation l_income -0.3092(0.2197)     -0.0565(0.1259)     -0.6018(0.2635)*
## Equation l_spend  0.2998(0.3339)      -0.7890(0.1914)***  0.1655(0.4004)
##                   l_income -5         l_spend -5          l_ipp -6
## Equation l_ipp    0.0064(0.2454)      -0.2277(0.1643)     -0.2149(0.2984)
## Equation l_income -0.2869(0.2154)     -0.1353(0.1441)     -0.5744(0.2618)*
## Equation l_spend  0.1311(0.3273)      -0.6549(0.2190)**   0.5722(0.3979)
##                   l_income -6         l_spend -6          l_ipp -7
## Equation l_ipp    0.2045(0.2408)      -0.1391(0.1680)     -0.4781(0.2928)
## Equation l_income -0.2632(0.2112)     -0.1108(0.1474)     -0.5281(0.2569)*
## Equation l_spend  -0.0847(0.3210)     -0.3894(0.2240).    0.7229(0.3904).
##                   l_income -7         l_spend -7          l_ipp -8
## Equation l_ipp    0.2557(0.2457)      -0.2852(0.1501).    -0.5696(0.2731)*
## Equation l_income -0.2178(0.2156)     -0.1113(0.1317)     -0.4977(0.2396)*
## Equation l_spend  0.1183(0.3277)      -0.1144(0.2002)     0.6543(0.3642).
##                   l_income -8         l_spend -8          l_ipp -9
## Equation l_ipp    0.1733(0.2386)      -0.4066(0.1424)**   -0.3355(0.2193)
## Equation l_income -0.3467(0.2093)     -0.0894(0.1249)     -0.2284(0.1924)
## Equation l_spend  0.1582(0.3181)      -0.2707(0.1898)     0.4563(0.2925)
##                   l_income -9         l_spend -9          l_ipp -10
## Equation l_ipp    -0.1839(0.2037)     -0.1930(0.1326)     -0.0395(0.1324)
## Equation l_income -0.3783(0.1788)*    -0.1123(0.1164)     -0.0178(0.1162)
## Equation l_spend  0.3951(0.2717)      -0.0942(0.1769)     0.1203(0.1766)
##                   l_income -10        l_spend -10         BRENT
## Equation l_ipp    0.0827(0.1616)      -0.1800(0.1076)     -0.1123(0.0996)
## Equation l_income -0.1276(0.1418)     -0.0581(0.0944)     0.1510(0.0874).
## Equation l_spend  0.2345(0.2155)      0.0567(0.1434)      -0.0719(0.1328)
```

```
##                     MIACR
## Equation l_ipp    0.0150(0.0094)
## Equation l_income -0.0158(0.0082).
## Equation l_spend  -0.0089(0.0125)
```

```
## VECM, как VAR в уровнях, оцененный OLS:
m4 <- vec2var(j_obj, r=1)
summary(m4)
```

```
##               Length Class  Mode
## deterministic    3   -none- numeric
## A                9   -none- list
## p                1   -none- numeric
## K                1   -none- numeric
## y              288   -none- numeric
## obs              1   -none- numeric
## totobs           1   -none- numeric
## call             3   -none- call
## vecm             1   ca.jo  S4
## datamat       2697   -none- numeric
## resid          261   -none- numeric
## r                1   -none- numeric
```

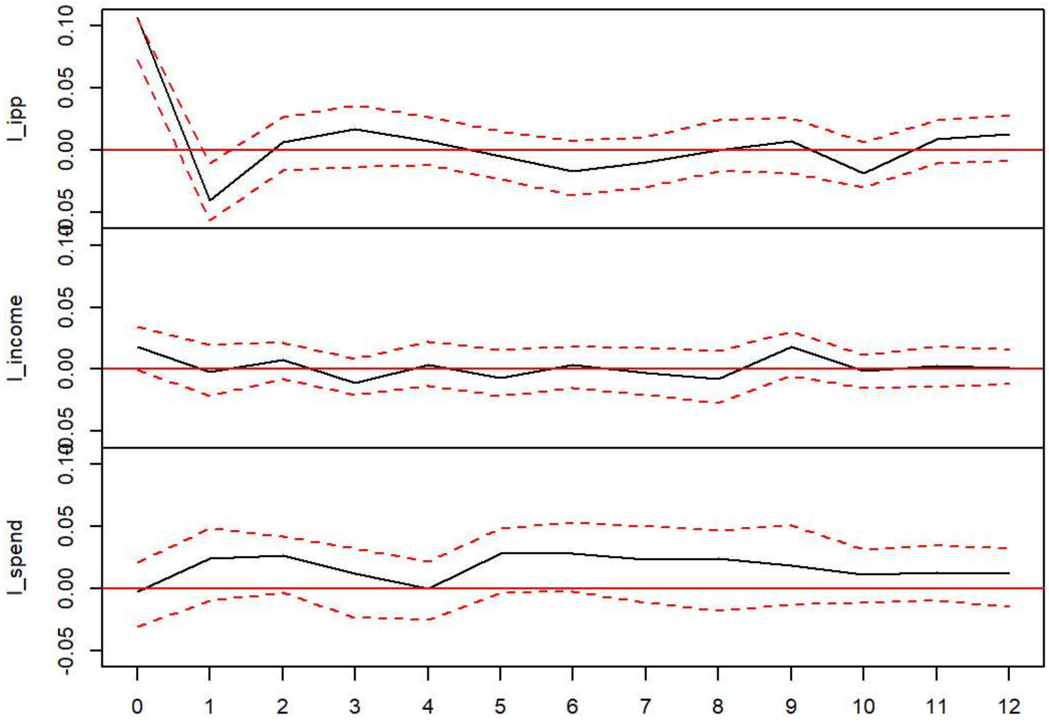
```
m4
```

```
##
## Coefficient matrix of lagged endogenous variables:
##
## A1:
##             l_ipp.l1 l_income.l1    l_spend.l1
## l_ipp    -0.34942767 -0.19706056 -0.0005648042
## l_income -0.04415842  0.14786897  0.0144597930
## l_spend   0.21786075  0.09911659  0.2637936358
##
##
## A2:
##             l_ipp.l2 l_income.l2  l_spend.l2
## l_ipp    -0.09092049  0.11715685  0.24916034
## l_income  0.02610759  0.15728601  0.06889394
## l_spend   0.26326134  0.05515834 -0.08942909
##
##
## A3:
##            l_ipp.l3 l_income.l3  l_spend.l3
## l_ipp     0.1094384 -0.05280654 -0.02319072
## l_income -0.1453852  0.19822275  0.08422090
## l_spend   0.1132690  0.26692681  0.18993929
##
##
## A4:
##              l_ipp.l4 l_income.l4  l_spend.l4
## l_ipp     0.097937304 -0.07145434 -0.01033625
## l_income -0.096981945  0.28623952 -0.08481645
## l_spend   0.005065957 -0.36536580 -0.30981836
##
##
## A5:
##              l_ipp.l5 l_income.l5  l_spend.l5
## l_ipp    -0.007494008  0.19070032 -0.04368052
## l_income -0.100140550 -0.01132374 -0.01728714
## l_spend   0.223656062 -0.03427236  0.15569786
##
##
## A6:
##             l_ipp.l6 l_income.l6 l_spend.l6
## l_ipp    -0.23527794  0.28390359 0.08784254
## l_income  0.02119255  0.07972582 0.01773697
## l_spend   0.38602793 -0.31553307 0.19375528
##
##
## A7:
##              l_ipp.l7  l_income.l7  l_spend.l7
## l_ipp    -0.337783440 0.0006167535 -0.08900790
## l_income  0.006829966 0.1102378087  0.01734541
## l_spend   0.231703877 0.1580940442  0.25332286
##
##
## A8:
##            l_ipp.l8 l_income.l8   l_spend.l8
## l_ipp    -0.2242233 -0.04529507 -0.113664657
```

```
## l_income -0.1056944 -0.02573523 -0.004330903
## l_spend   0.1424558  0.10532143 -0.148583044
##
##
## A9:
##            l_ipp.l9 l_income.l9  l_spend.l9
## l_ipp    0.12193971 -0.40230027  0.19634010
## l_income 0.10602041  0.02686541 -0.05236889
## l_spend  0.03530792  0.17970649  0.27765593
##
##
## Coefficient matrix of deterministic regressor(s).
##
##           constant
## l_ipp     8.102561
## l_income  1.434225
## l_spend  -6.803095
```

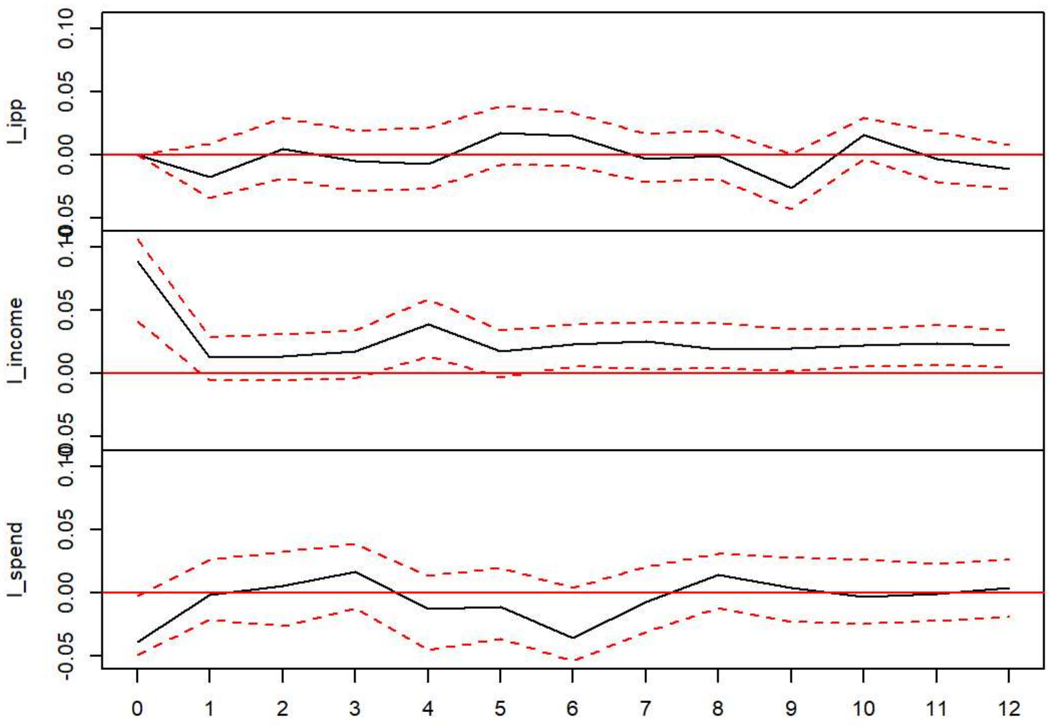
```
## Impulse response functions
IRF <- vars::irf(m4, n.ahead = 12, boot = TRUE, ci = 0.95, runs = 100)
plot(IRF)
```

Orthogonal Impulse Response from l_ipp

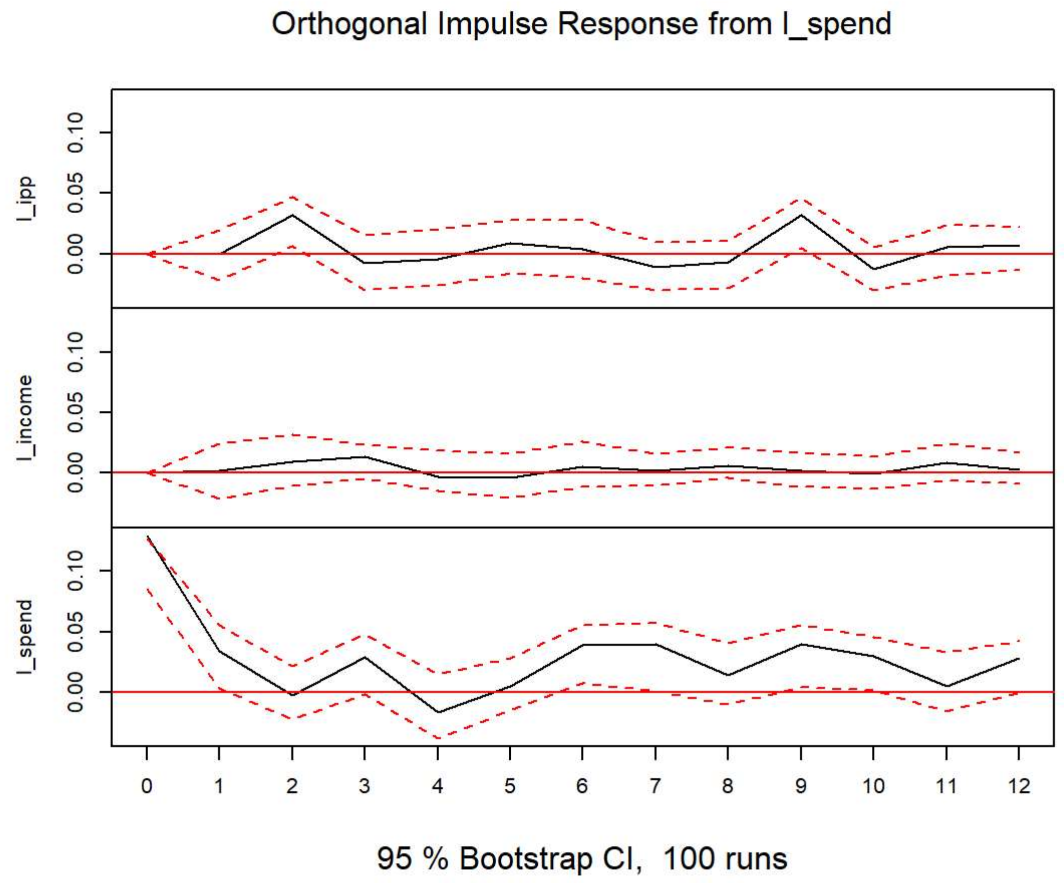



95 % Bootstrap CI, 100 runs

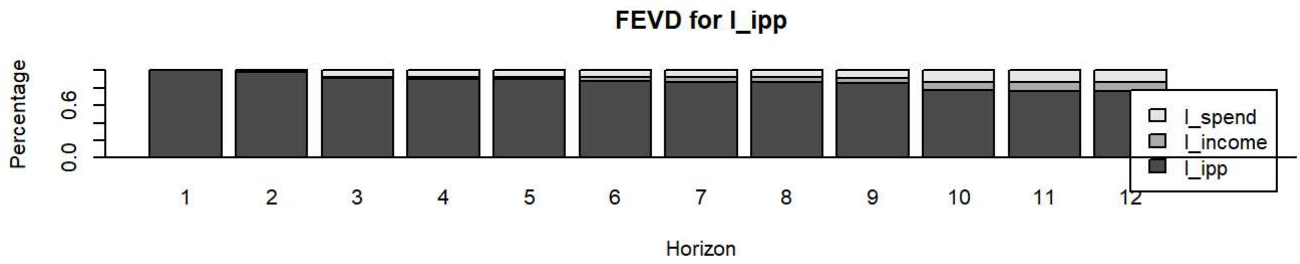
Orthogonal Impulse Response from l_income

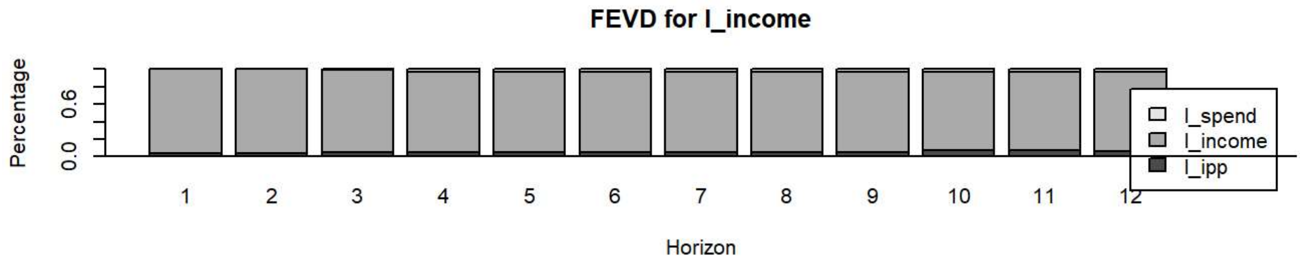


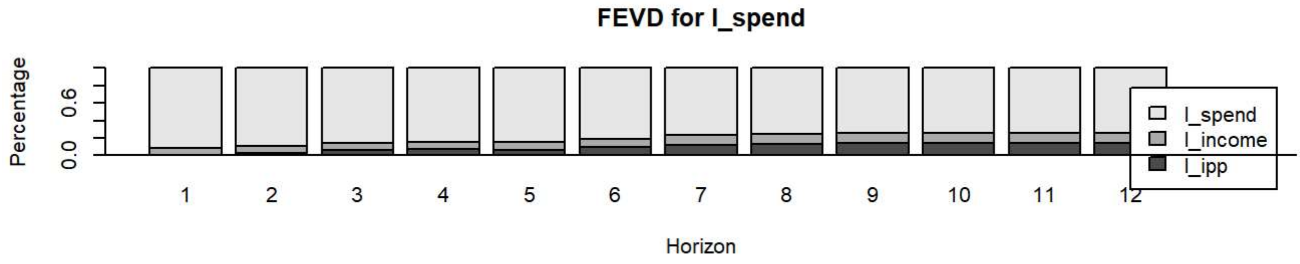

95 % Bootstrap CI, 100 runs




```
## FEVD
FEVD_o <- vars::fevd(m4, n.ahead=12)
plot(FEVD_o)
```




# Интерпретация Impulse response functions

Изменение логарифма индекса промышленного производства (ipp) в соответствии с полученными IRF повлияет только на свое же значение в первом периоде #Изменение логарифма дохода региона согласно IRF влияет на логарифм расходов в 6-ом периоде и на сами себя в 4. А изменение логарифма государственных #расходов влияет на себя в 6, 7 и 9 периодах.

# Интерпретация FEVD

На ИПП влияние государственных расходов и доходов региона появляется в третьем периоде, но их наибольшее влияние (около 20% совокупно) проявляется в 10-12 периодах. Причем большее влияние оказывают расходы. На доходы региона в каждом периоде ИПП влияет мало и примерно одинаково.Влияние гос расходов проявляется только в 4 периоде, что соизмеримо с кварталом. Однако влияние и гос расходов, и ИПП крайне мало, каждый менее 5%. На расходы уже в первом периоде влияют доходы, что экономически логично. Во втором периоде появляется влияние ИПП. Наибольшее влияние проявляется через полгода в 7 периоде и сохраняется до 12.

# BVAR

```
setwd("C:/Users/annaz/Documents/R/Times series")
dd <- read.xlsx("TS_data.xlsx",sheet = "Sheet1")

dd=dd[with(dd, order(id,year,month)), ]

dd_grouped_by_region = (dd %>% group_by(id,region) %>%
                summarize(month_count=n()))
```

Анализ BVAR будет произведен на основании данных о доходах и расходах бюджета, а также ИПП по г. Москва.

```
df = dd[dd$region=='г. Москва',]

TSspend = ts(df$spend, start = c(2014,1), frequency = 12)
TSincome = ts(df$income, start = c(2014,1), frequency = 12)
TSipp = ts(df$ipp, start = c(2014,1), frequency = 12)


d_TSspend = diff(TSspend)
d_TSincome = diff(TSincome)
d_TSipp = diff(TSipp)

# После перехода к разницам все данные стацонарны

# Подготавливаем данные для VAR

dataForVAR = data.frame(d_income = d_TSincome/1000,
                        d_spend = d_TSspend/1000,
                        d_ipp = d_TSipp)
dataExog = data.frame(MIACR = df$MIACR[2:96],
                      BRENT = df$BRENT[2:96]/100)

summary(dataForVAR)
```

```
##     d_income           d_spend             d_ipp
##  Min.   :-11.2107   Min.   :-15.2262   Min.   :-67.4000
##  1st Qu.: -0.1236   1st Qu.: -1.4733   1st Qu.: -9.8000
##  Median :  0.1433   Median :  0.1207   Median :  0.1000
##  Mean   :  0.1249   Mean   :  0.1126   Mean   :  0.3874
##  3rd Qu.:  0.3183   3rd Qu.:  1.6702   3rd Qu.: 10.9500
##  Max.   : 11.8748   Max.   : 20.6250   Max.   : 58.2000
```

```
summary(dataExog)
```

```
##      MIACR            BRENT
##  Min.   : 4.060   Min.   :0.2663
##  1st Qu.: 6.680   1st Qu.:0.4922
##  Median : 7.720   Median :0.6271
##  Mean   : 8.243   Mean   :0.6572
##  3rd Qu.:10.185   3rd Qu.:0.7614
##  Max.   :16.960   Max.   :1.2284
```

```
sum(is.na(dataForVAR))
```

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

```
sum(is.na(dataExog))
```

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

```
dataForVAR <- scale(dataForVAR)
dataExog <- scale(dataExog)

dataTotal = cbind(dataForVAR,dataExog)
```

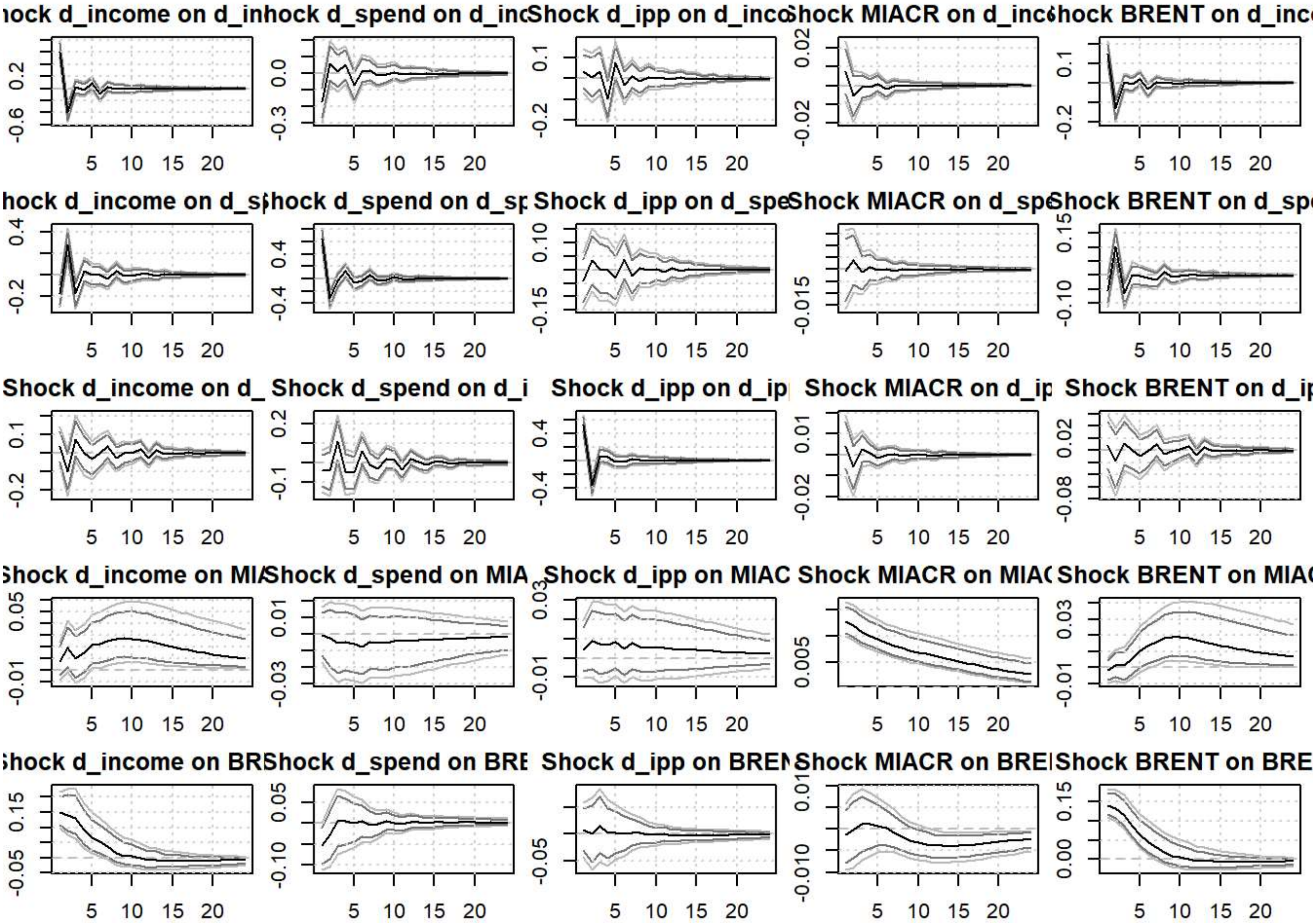
# Оценим BVAR со стартовыми значениями гиперпараметров. Используется априорное распределение Миннесота.

```
minnPrior <- bv_minnesota(
  lambda = bv_lambda(mode = 0.5, sd = 0.00001),
  alpha = bv_alpha(mode = 3, sd = 0.00001),
  psi = bv_psi(scale = 0.001, shape = 0.002))

bvPrior_Object <- bv_priors(mn = minnPrior)

m <- bvar(data = dataTotal, lags = 12,
          priors = bvPrior_Object, verbose = FALSE)
m
```

```
## Bayesian VAR consisting of 83 observations, 5 variables and 12 lags.
## Time spent calculating: 15.65 secs
## Hyperparameters: lambda
## Hyperparameter values after optimisation: 0.5
## Iterations (burnt / thinning): 10000 (5000 / 1)
## Accepted draws (rate): 8 (0.002)
```

# Построим IRF и FEVD полученной модели без введения ограничений на импульсные отклики.

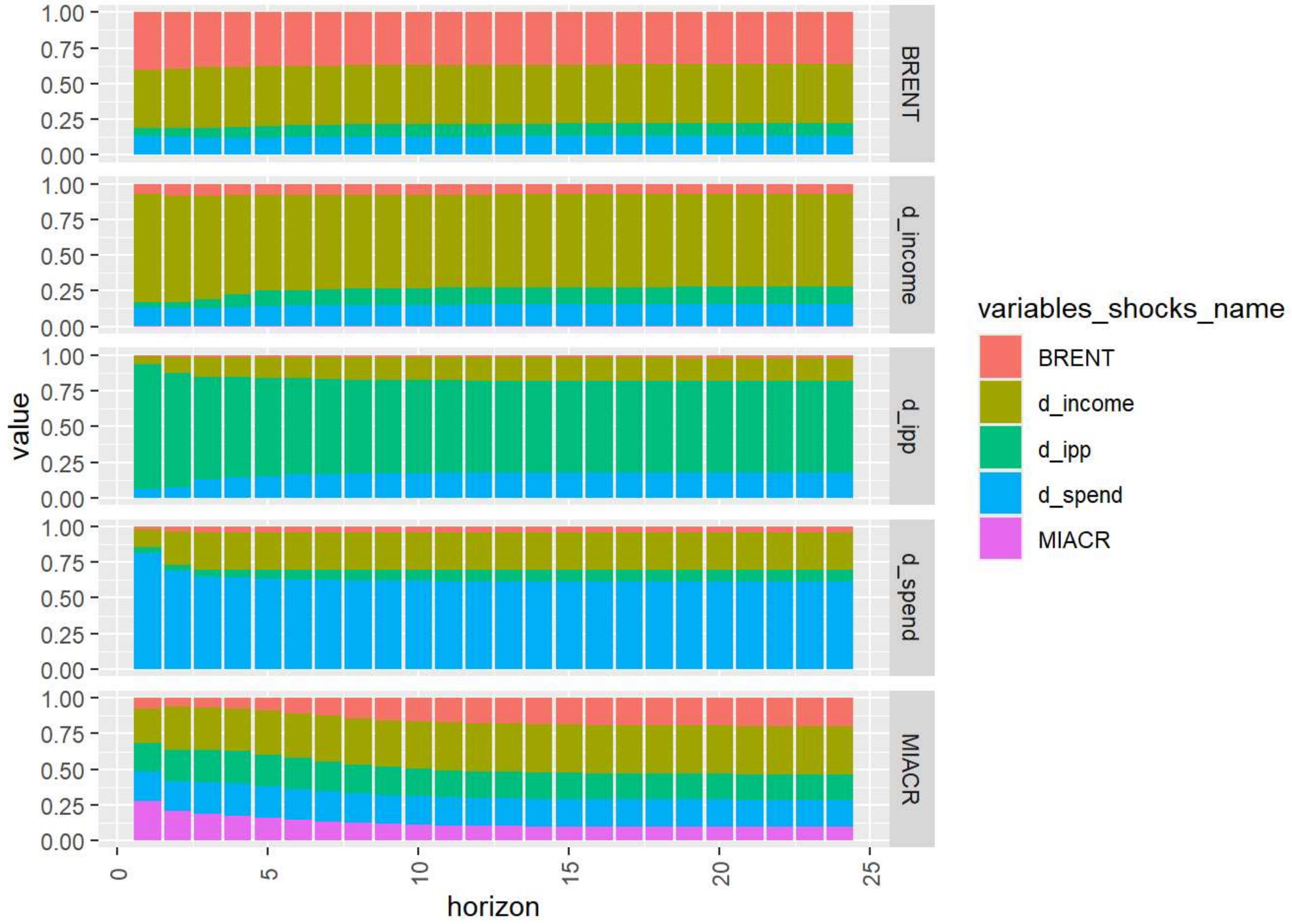
```
bv_irf_m__no_restrictionts=bv_irf(
  horizon = 24,
  fevd = TRUE,
  identification = FALSE,
  sign_restr = NULL,
  sign_lim = 1000
)

plot(irf(m,value=bv_irf_m__no_restrictionts,conf_bands = c(0.05, 0.1)))
```




Как можно заметить для Индекса промышленного производства (d_ipp) значимыми для 3го месяца (квартал) являются шоки госрасходов (d_spend). Шоки налоговых доходов (d_income) являются значимыми только для второго месяца. Прочие импульсные отклики представляются либо не значимыми, либо нелогичными, что обусловлено отсутствием введенных ограничений (шоки доходов на нефть, фактические ставки по кредитам).

Рассмотрим FEVD данной модели используя значения входящие в 95% доверительный интервал:

```
fevd_m__no_restrictionts=fevd(m,value=bv_irf_m__no_restrictionts, n_thin = 1L,conf_bands = c
(0.05, 0.1))
df_fevd_m__no_restrictionts=data.frame(melt(fevd_m__no_restrictionts$quants[5,,,])) # 5 - 95%
colnames(df_fevd_m__no_restrictionts)=c('variables','horizon','variables_shocks','value')
df_fevd_m__no_restrictionts=merge(df_fevd_m__no_restrictionts,
                                  cbind(variables = rownames(melt(fevd_m__no_restrictionts$va
riables)),melt(fevd_m__no_restrictionts$variables,value.name='variables_name')), by = "variab
les")
df_fevd_m__no_restrictionts=merge(df_fevd_m__no_restrictionts,
                                  cbind(variables_shocks=rownames(melt(fevd_m__no_restriction
ts$variables)),melt(fevd_m__no_restrictionts$variables,value.name='variables_shocks_name'))
                                  , by = "variables_shocks")
ggplot(df_fevd_m__no_restrictionts,aes(x = horizon, y = value, fill = variables_shocks_name))
+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +  geom_col(position = "stack")
+ facet_grid(variables_name ~ .)
```




Анализируя FEVD, можем заметить, что для d_ipp со временем возрастает вклад как расходов, так и доходов. При этом эффекта от шоков цены на нефть и ставок по кредитам для d_ipp не наблюдается. Также можно заметить, что шоки доходов сильнее влияют на d_spend, чем шоки расходов на d_income, что представляется логичным и согласуется с теорией.

В то же время в дисперсию экзогенных переменных (BRENT,MIACR) значимый вклад вносят и расходы, доходы и ИПП, что является следствием отсутствий ограничений на отклики.

# Рассмотрим IRF и FEVD полученной модели с учетом введения ограничений на импульсные отклики.

В рассмотренных статьях (в частности - Власов С., Дерюгина Е. Фискальные мультипликаторы в России / Банк России) предлагается следующие ограничения на отклики:

| Шок | ВВП | Доходы | Расходы | ВВП Евросоюза | Рублевая цена нефти |
|---|---|---|---|---|---|
| Доходов | - | + | >=0 | 0 | 0 |
| Расходов | + | >= 0 | + | 0 | 0 |

Исходя из приведенных выше ограничений и ограничений пакет вводим матрицу ограничений:

| Шок | ИПП | Доходы | Расходы | Ставки по кредитам | Цена нефти |
|---|---|---|---|---|---|
| Доходов | - | + | >=0 | 0 | 0 |
| Расходов | + | >= 0 | + | 0 | 0 |

| Шок | ИПП | Доходы | Расходы | Ставки по кредитам | Цена нефти |
|---|---|---|---|---|---|
| ИПП | + | + | 0 | >=0 | 0 |
| Ставок | - | - | >=0 | + | >=0 |
| Цены нефти | + | + | >=0 | - | + |

Однако с подобными ограничениями матрица не может быть подобрана в указанное число итераций. При увеличении числа итераций в 5 раз так же IRF и FEVD так же не могут быть построены.

Попробуем уменьшить число переменных для формирования IRF и FEVD с ограничениями;

```
dataForVAR = data.frame(
                        d_income = d_TSincome/1000,
                        d_ipp = d_TSipp,
                        d_spend = d_TSspend/1000
                        )

minnPrior <- bv_minnesota(
  lambda = bv_lambda(mode = 0.5, sd = 0.00001),
  alpha = bv_alpha(mode = 3, sd = 0.00001),
  psi = bv_psi(scale = 0.001, shape = 0.002))

bvPrior_Object <- bv_priors(mn = minnPrior)

m2 <- bvar(data = dataForVAR, lags = 12,
          priors = bvPrior_Object,verbose = FALSE)
```

Предполагаем, что d_spend влияет на d_ipp и d_income, d_income влияет только на себя, d_ipp влияет на d_income

```
struct_matrix_v1 = matrix(c(1, 1, NA,
                            0, 1, 1,
                            0, 0, 1
), byrow = TRUE,ncol = 3)

bv_irf_m__with_restrictionts_v1=bv_irf(
  horizon = 24,
  fevd = TRUE,
  identification = TRUE,
  sign_restr = struct_matrix_v1,
  sign_lim = 50000
)

# plot(irf(m2,value=bv_irf_m__with_restrictionts_v1,conf_bands = c(0.05, 0.1)))
# fevd_m__with_restrictionts_v1=fevd(m2,value=bv_irf_m__with_restrictionts_v1,conf_bands = c
(0.05, 0.1))
```

Получаем аналогичный неутешительный результат.

Пробуем иную комбинацию ограничений: Предполагаем, что d_spend влияет на d_ipp и d_income, d_income влияет только на себя и негативно на d_ipp, d_ipp влияет на d_income

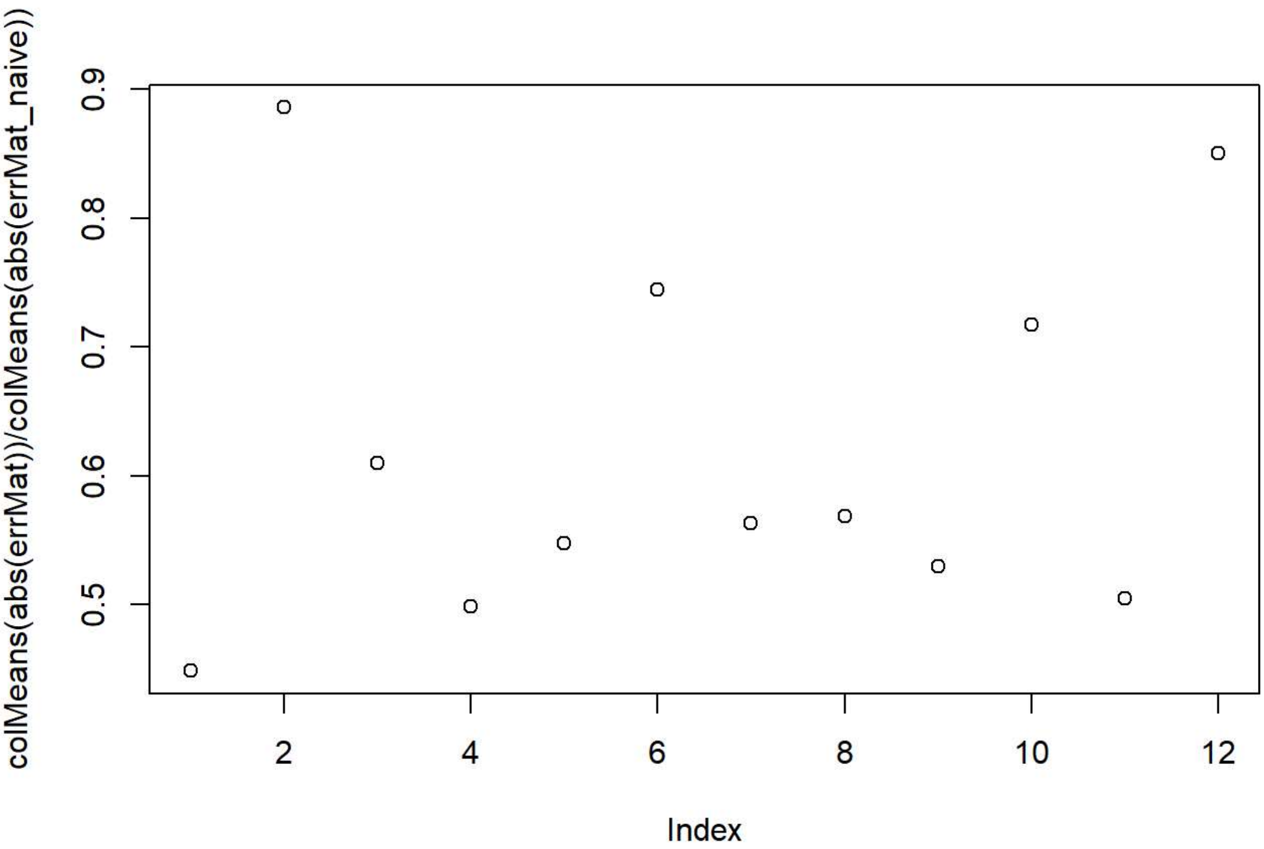
```
struct_matrix_v1 = matrix(c(1, 1, 1,
                            -1, 1, 1,
                            0, 0, 1
), byrow = TRUE,ncol = 3)

bv_irf_m__with_restrictionts_v1=bv_irf(
  horizon = 24,
  fevd = TRUE,
  identification = TRUE,
  sign_restr = struct_matrix_v1,
  sign_lim = 50000
)

# plot(irf(m2,value=bv_irf_m__with_restrictionts_v1,conf_bands = c(0.05, 0.1)))
# fevd_m__with_restrictionts_v1=fevd(m2,value=bv_irf_m__with_restrictionts_v1,conf_bands = c
(0.05, 0.1))
```

Аналогично построить IRF и FEVD с учетом этого ограничения не удалось.

# Произведем настройку гиперпараметров на

# кроссвалидации на 12 точках (год)

```
errMat <- matrix(nrow = 12, ncol = 12)
dataTotalCV = data.frame(dataTotal[1:84,])

for (i in 1:12){
  minnPrior <- bv_minnesota(
    lambda = bv_lambda(mode = 0.5, sd = 0.00001),
    alpha = bv_alpha(mode = 3, sd = 0.00001),
    psi = bv_psi(scale = 0.001, shape = 0.002, mode = rep(1, ncol(dataTotalCV))))

  bvPrior_Object <- bv_priors(mn = minnPrior)

  m <- bvar(data = dataTotalCV[1:(nrow(dataTotalCV) - i - 11), ], lags = 4,
            priors = bvPrior_Object,verbose = FALSE)
  bvarForecasts <- predict(m)

  bvarForecasts$fcast[,,3]

  ipp_allForecasts <- bvarForecasts$fcast[,,3]
  median_ipp_forecast <- apply(ipp_allForecasts, 2, median)

  errMat[i, ] <- median_ipp_forecast -
    dataTotalCV$d_ipp[(nrow(dataTotalCV) - i - 10):(nrow(dataTotalCV) - i + 1)]
}

errMat_naive <- matrix(nrow = 12, ncol = 12)
for (i in 1:12){
  errMat_naive[i, ] <- dataTotalCV$d_ipp[(nrow(dataTotalCV) - i - 11)] -
    dataTotalCV$d_ipp[(nrow(dataTotalCV) - i - 10):(nrow(dataTotalCV) - i + 1)]
}

#MASE
plot(colMeans(abs(errMat)) / colMeans(abs(errMat_naive)))
```




Исходя из значения метрики MASE, можем видеть, что прогноз по BVAR лучше наивного.