

# Time series - project

## Вариант 12

Рахманова Амина, Клечикова Маргарита БЭК-196

## 1. Задание

Взять дневные или недельные цены акций 5-7 компаний за период 2015-2021 годы включительно. Из них 2 компании должны быть те, кто имеет привилегированные акции.

Построить одномерные модели временных рядов, оценить волатильность.

Построить модель векторной авторегрессии для этих акций, проверить каждое уравнение VAR модели на структурный сдвиг в период начала пандемии (в США - 24 февраля) и в момент выхода новости о создании вакцины (в США - 4 ноября).

Выяснить, наблюдается ли коинтеграция между рассматриваемыми акциями? Продемонстрировать, как коинтеграция влияет на прогноз.

Для расчета ожидаемой доходности финансовых инструментов использовать модель CAPM.

## 2. Введение

Для данного исследования мы выбрали акции следующих компаний:

- ПАО “ТАТНЕФТЬ” (TATN\_p) - привилегированные акции
- ПАО “СУРГУТНЕФТЕГАЗ” (SNGS\_p) - привилегированные акции
- ОАО АК “АЛРОСА” (ALRS)
- ПАО “ГАЗПРОМ НЕФТЬ” (SIBN)
- ПАО “НОВОЛИПЕЦКИЙ МЕТАЛЛУРГИЧЕСКИЙ КОМБИНАТ” (NLMK)
- ПАО “КАМАЗ” (KMAZ)
- ПАО “АЭРОФЛОТ-РОССИЙСКИЕ АВИАЛИНИИ” (AFLT)

## Наши данные

Мы взяли дневные данные с 05.01.2015 по 30.12.2021 (исключая выходные на бирже). Так они выглядят в исходном виде:

```
head(data, 5)
```

date	TATN_p	SNGS_p	ALRS	SIBN	NLMK	KMAZ	AFLT
<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2015-01-05	134.7	29.600	60.38	142.0	66.28	26.25	33.21
2015-01-06	135.3	30.195	61.28	144.6	66.70	27.05	33.07
2015-01-08	136.0	31.600	60.20	148.1	67.41	28.55	35.17

<b>date</b> <date>	<b>TATN_p</b> <dbl>	<b>SNGS_p</b> <dbl>	<b>ALRS</b> <dbl>	<b>SIBN</b> <dbl>	<b>NLMK</b> <dbl>	<b>KMAZ</b> <dbl>	<b>AFLT</b> <dbl>
2015-01-09	134.5	32.010	61.91	146.6	67.90	28.75	34.00
2015-01-12	133.3	31.360	63.00	143.1	70.12	28.50	34.45

5 rows

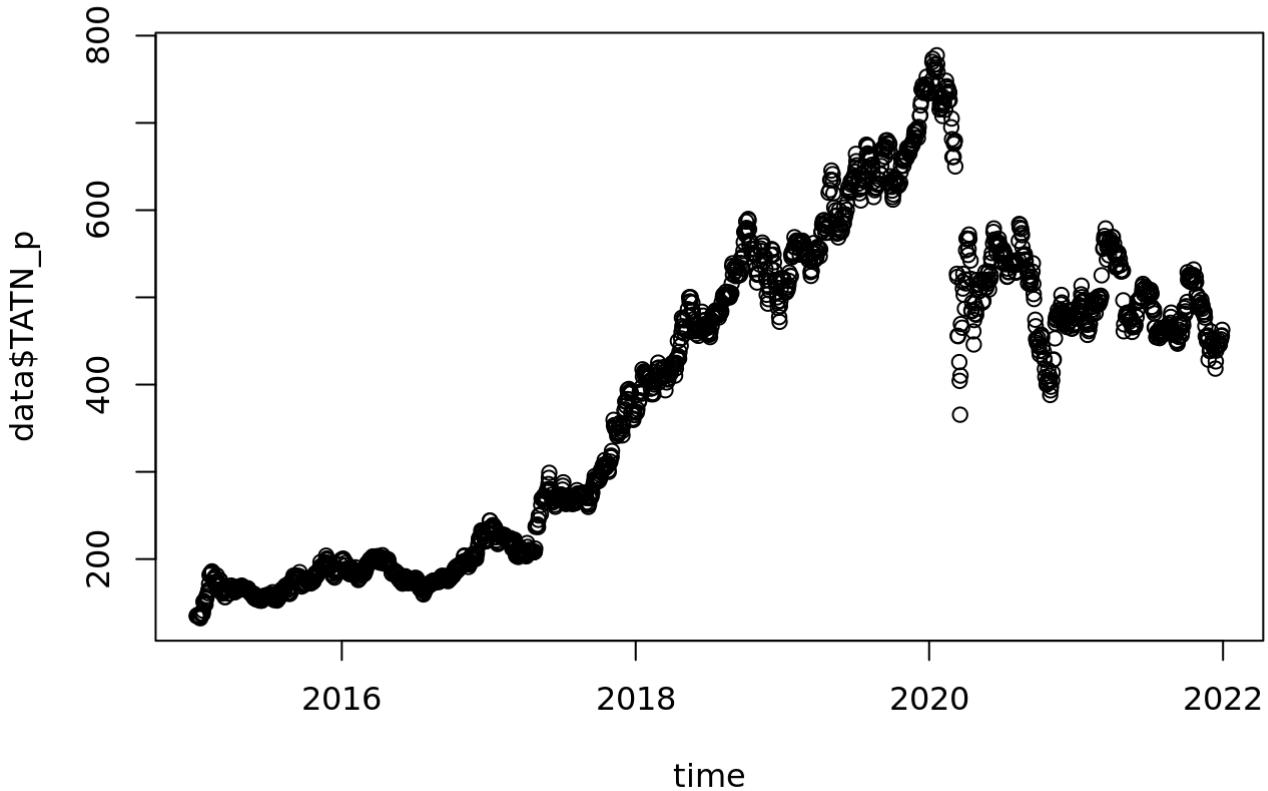
```
tail(data, 5)
```

<b>date</b> <date>	<b>TATN_p</b> <dbl>	<b>SNGS_p</b> <dbl>	<b>ALRS</b> <dbl>	<b>SIBN</b> <dbl>	<b>NLMK</b> <dbl>	<b>KMAZ</b> <dbl>	<b>AFLT</b> <dbl>
2021-12-24	446.1	38.005	115.30	562.00	212.40	106.0	58.96
2021-12-27	451.9	38.410	116.55	526.25	214.60	105.0	59.08
2021-12-28	455.7	38.150	119.98	540.70	215.76	104.6	58.92
2021-12-29	456.4	38.340	121.25	539.00	215.02	104.7	58.98
2021-12-30	462.8	38.570	122.34	544.80	217.04	105.0	59.16

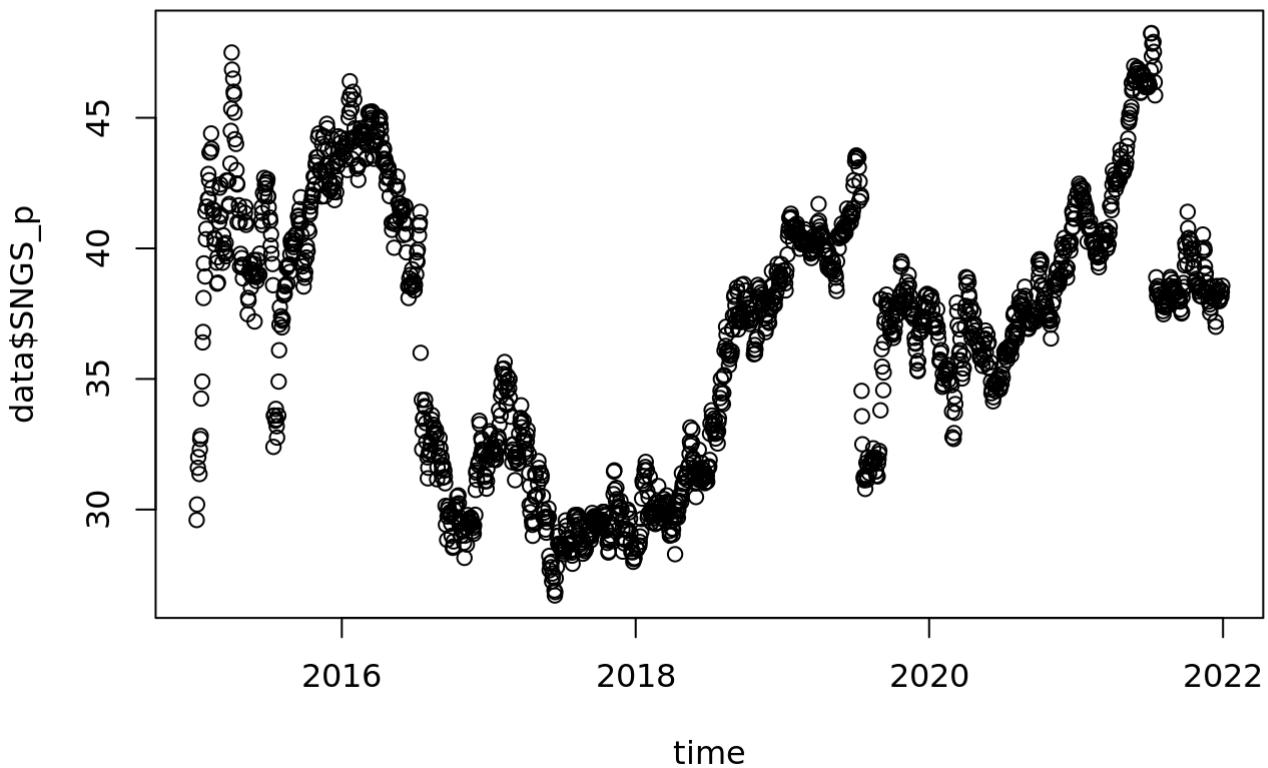
5 rows

### 3. Проверка данных на сезонность

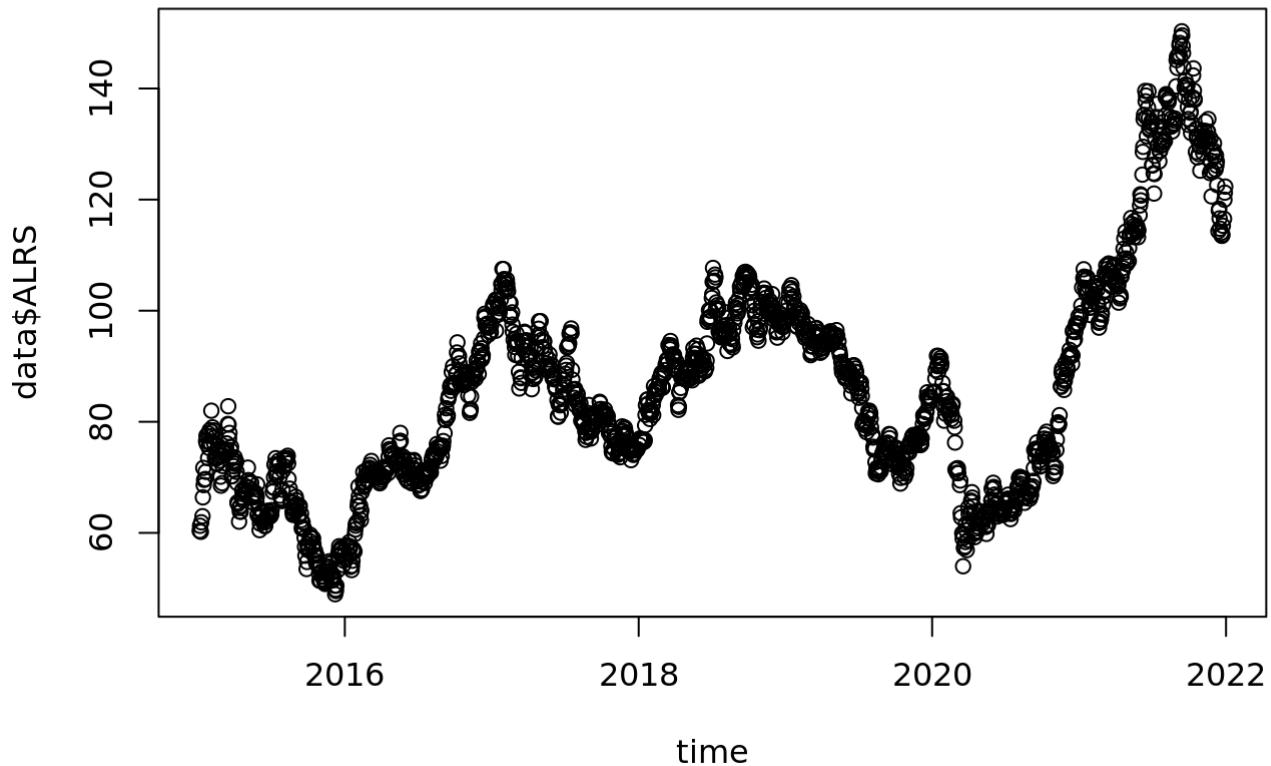
```
time = data$date  
time = as.Date(time, "%m/%d/%y")  
  
plot(time, data$TATN_p)
```



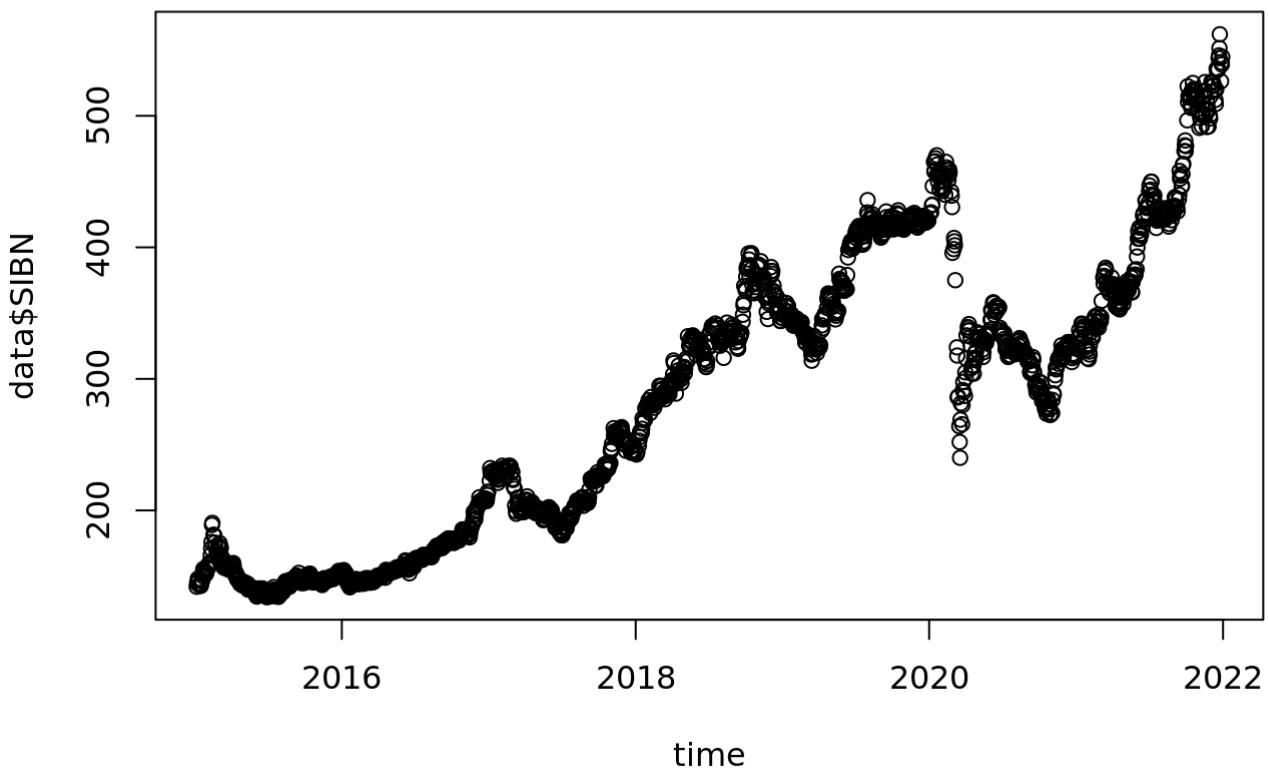
```
plot(time, data$SNGS_p)
```



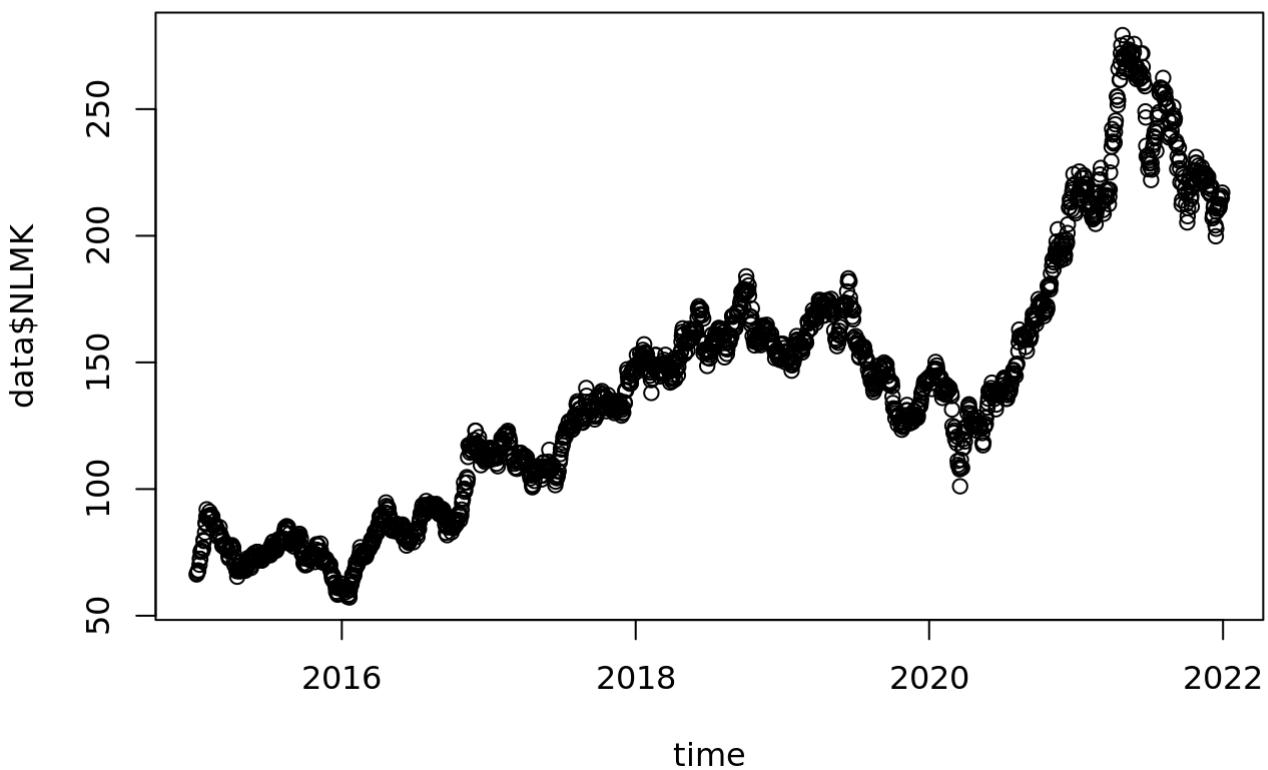
```
plot(time, data$ALRS)
```



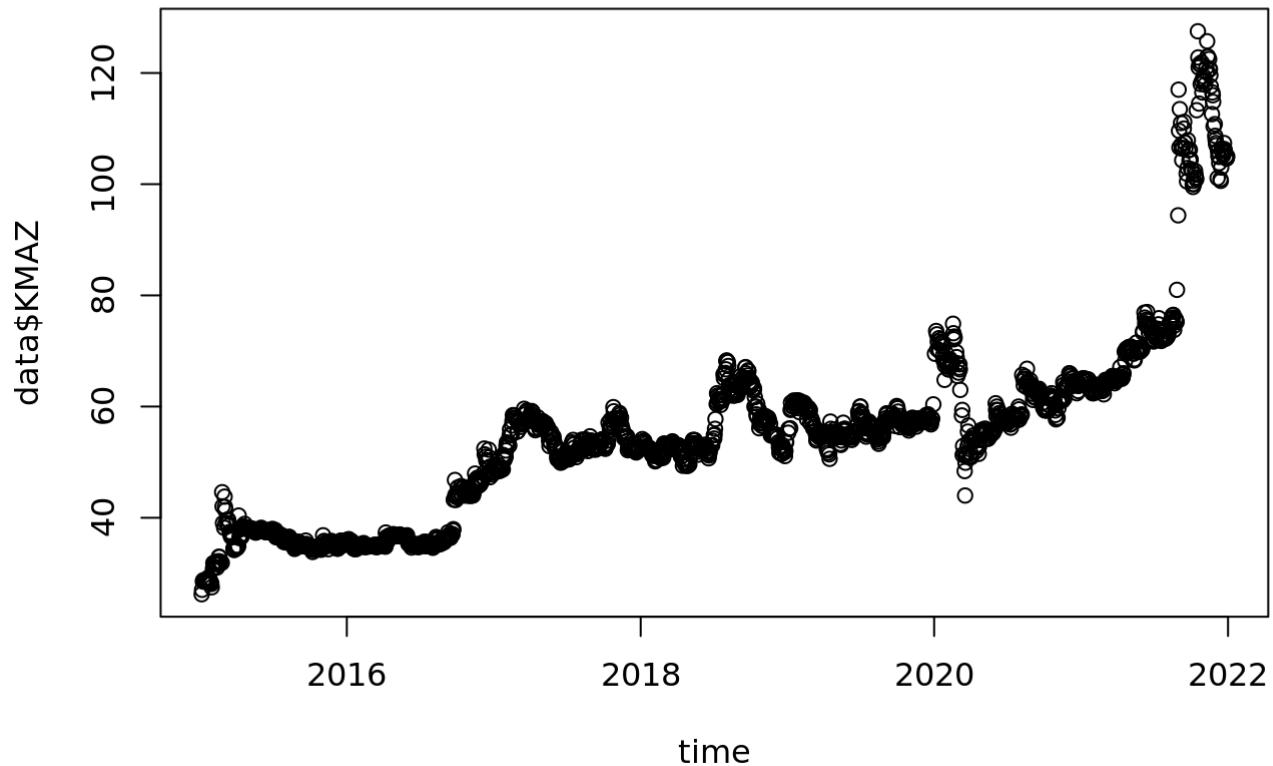
```
plot(time, data$SIBN)
```



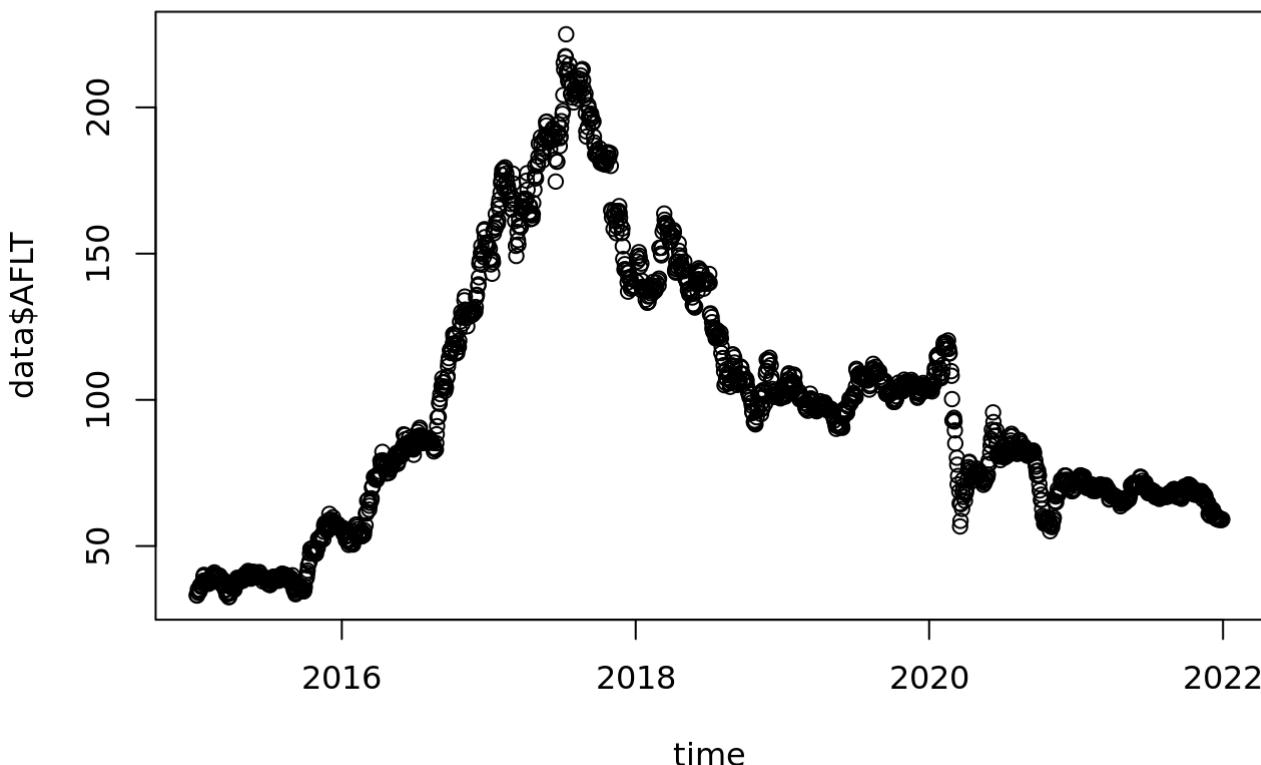
```
plot(time, data$NLMK)
```



```
plot(time, data$KMAZ)
```



```
plot(time, data$AFLT)
```



На графиках сезонность не наблюдается, что ожидаемо, так как цены на акции меняются стихийно, вследствие прочих внешних факторов.

## 4. Проверка данных на стационарность

Критическое значение для проверки на 1% уровне значимости = -3.43

### 4.1. TATN\_p

```
#Pacf(data$TATN_p) #lags = 1
stTANT = ur.df(data$TATN_p, type="drift", lags = 1, selectlags = "Fixed")
stTANT@teststat
```

```
##          tau2      phil
## statistic -1.556173 1.577026
```

Значение t статистики  $-1.556173 > -3.43$ .

### 4.2. SNGS\_p

```
#Pacf(data$SNGS_p) #lags = 1
stsNGS = ur.df(data$SNGS_p, type="drift", lags = 1, selectlags = "Fixed")
stsNGS@teststat
```

```
##          tau2      phil
## statistic -2.913332 4.291275
```

Значение t статистики -2.913332 > -3.43.

## 4.3. ALRS

```
#Pacf(data$ALRS) #lags = 1
stALRS = ur.df(data$ALRS, type="drift", lags = 1, selectlags = "Fixed")
stALRS@teststat
```

```
##          tau2      phil
## statistic -1.520594 1.532856
```

Значение t статистики -1.520594 > -3.43.

## 4.4. SIBN

```
#Pacf(data$SIBN) #lags = 1
stSIBN = ur.df(data$SIBN, type="drift", lags = 1, selectlags = "Fixed")
stSIBN@teststat
```

```
##          tau2      phil
## statistic -0.001731526 1.700032
```

Значение t статистики -0.001731526 > -3.43.

## NLMK

```
#Pacf(data$NLMK) #lags = 1
stNLMK = ur.df(data$NLMK, type="drift", lags = 1, selectlags = "Fixed")
stNLMK@teststat
```

```
##          tau2      phil
## statistic -0.9758007 1.515227
```

Значение t статистики -0.9758007 > -3.43.

## 4.5. KMAZ

```
#Pacf(data$KMAZ) #lags = 1
stKMAZ = ur.df(data$KMAZ, type="drift", lags = 1, selectlags = "Fixed")
stKMAZ@teststat
```

```
##          tau2      phil
## statistic -0.9685645 1.289144
```

Значение t статистики -0.9685645 > -3.43.

## 4.6. AFLT

```
#Pacf(data$AFLT) #lags = 1  
stAFLT = ur.df(data$AFLT, type="drift", lags = 1, selectlags = "Fixed")  
stAFLT@teststat
```

```
## tau2 phi1  
## statistic -1.397104 1.014872
```

Значение t статистики -1.397104 > -3.43.

## 4.7. Вывод:

Гипотеза о нестационарности рядов не отклоняется => ряды нестационарны, применяем тест ЭнглаГренджера на коинтеграцию для начальных данных, а для построения ARMA-модели будем использовать разности.

# 5. Проверка первых разностей на стационарность

## 5.1. Расчет первых разностей

```
d1TATN = diff(data$TATN_p, differences=1)  
d1SNGS = diff(data$SNGS_p, differences=1)  
d1ALRS = diff(data$ALRS, differences=1)  
d1SIBN = diff(data$SIBN, differences=1)  
d1NLMK = diff(data$NLMK, differences=1)  
d1KMAZ = diff(data$KMAZ, differences=1)  
d1AFLT = diff(data$AFLT, differences=1)
```

## 5.2. d1TATN

```
#Pacf(d1TATN) #lags = 2  
std1TATN = ur.df(d1TATN, type="drift", lags = 2, selectlags = "Fixed")  
std1TATN@teststat
```

```
## tau2 phi1  
## statistic -21.80503 237.73
```

Значение t статистики -21.80503 < -3.43.

## 5.3. d1SNGS

```
#Pacf(d1SNGS) #lags = 7  
std1SNGS = ur.df(d1SNGS, type="drift", lags = 7, selectlags = "Fixed")  
std1SNGS@teststat
```

```
##          tau2      phil
## statistic -14.47239 104.7288
```

Значение t статистики -14.47239 < -3.43.

## 5.4. d1ALRS

```
#Pacf(d1ALRS) #lags = 27
std1ALRS = ur.df(d1ALRS, type="drift", lags = 27, selectlags = "Fixed")
std1ALRS@teststat
```

```
##          tau2      phil
## statistic -7.792801 30.36435
```

Значение t статистики -7.792801 < -3.43.

## 5.5. d1SIBN

```
#Pacf(d1SIBN) #lags = 1
std1SIBN = ur.df(d1SIBN, type="drift", lags = 1, selectlags = "Fixed")
std1SIBN@teststat
```

```
##          tau2      phil
## statistic -28.52974 406.9731
```

Значение t статистики -28.52974 < -3.43.

## 5.6. d1NLMK

```
#Pacf(d1NLMK) #lags = 7
std1NLMK = ur.df(d1NLMK, type="drift", lags = 7, selectlags = "Fixed")
std1NLMK@teststat
```

```
##          tau2      phil
## statistic -13.70413 93.90189
```

Значение t статистики -13.70413 < -3.43.

## 5.7. d1KMAZ

```
#Pacf(d1KMAZ) #lags = 2
std1KMAZ = ur.df(d1KMAZ, type="drift", lags = 2, selectlags = "Fixed")
std1KMAZ@teststat
```

```
##          tau2      phil
## statistic -26.38805 348.1647
```

Значение t статистики -26.38805 < -3.43.

## 5.8. d1AFLT

```
#Pacf(d1AFLT) #lags = 1  
std1AFLT = ur.df(d1AFLT, type="drift", lags = 1, selectlags = "Fixed")  
std1AFLT@teststat
```

```
## tau2 phil  
## statistic -28.56271 407.9146
```

Значение t статистики -28.56271 < -3.43.

## 5.9. Вывод:

Гипотеза о нестационарности отклоняется на 1% уровне => разности стационарны, можно строить ARMA модель на первых разностях.

# 6. ARMA model и проверка на автокорреляцию

lag = ln(1765) = 7

Параметры ARMA(p, k) выбирались на основе сравнения BIC и AIC нескольких моделей ARMA.

## 6.1. d1TATN

```
#eacf(d1TATN) # (p, k) = (0,2)  
ARMAmodelTANT = Arima(data$TATN_p, c(0,1,2), include.constant=TRUE, method = c("CSS-ML"))  
  
#Acf(residuals(ARMAmodelTANT)) # автокорреляция на графике  
Box.test(residuals(ARMAmodelTANT), lag = 7, type = c("Ljung-Box"), fitdf = 2) # тест на автокорреляцию
```

```
##  
## Box-Ljung test  
##  
## data: residuals(ARMAmodelTANT)  
## X-squared = 7.0018, df = 5, p-value = 0.2205
```

```
shapiro.test(residuals(ARMAmodelTANT)) # проверка на нормальное распределение
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: residuals(ARMAmodelTANT)  
## W = 0.83457, p-value < 2.2e-16
```

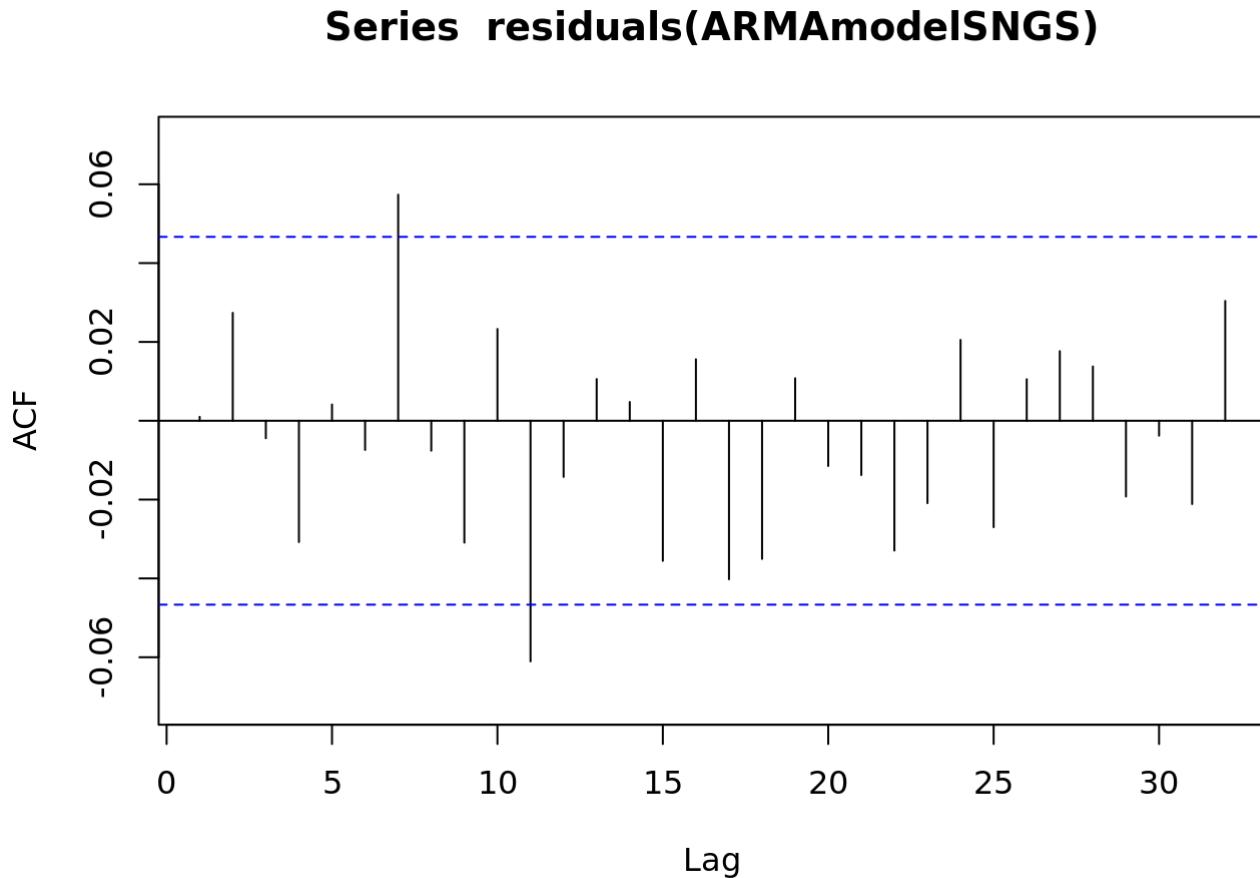
## 6.2. d1SNGS

```

ARMAmodelSNGS = Arima(data$SNGS_p, c(0,1,1), include.constant=TRUE, method = c("CSS-M
L"))

Acf(residuals(ARMAmodelSNGS))

```



```
Box.test(residuals(ARMAmodelSNGS), lag = 7, type = c("Ljung-Box"), fitdf = 1)
```

```

## 
## Box-Ljung test
## 
## data: residuals(ARMAmodelSNGS)
## X-squared = 9.002, df = 6, p-value = 0.1735

```

```
shapiro.test(residuals(ARMAmodelSNGS))
```

```

## 
## Shapiro-Wilk normality test
## 
## data: residuals(ARMAmodelSNGS)
## W = 0.81328, p-value < 2.2e-16

```

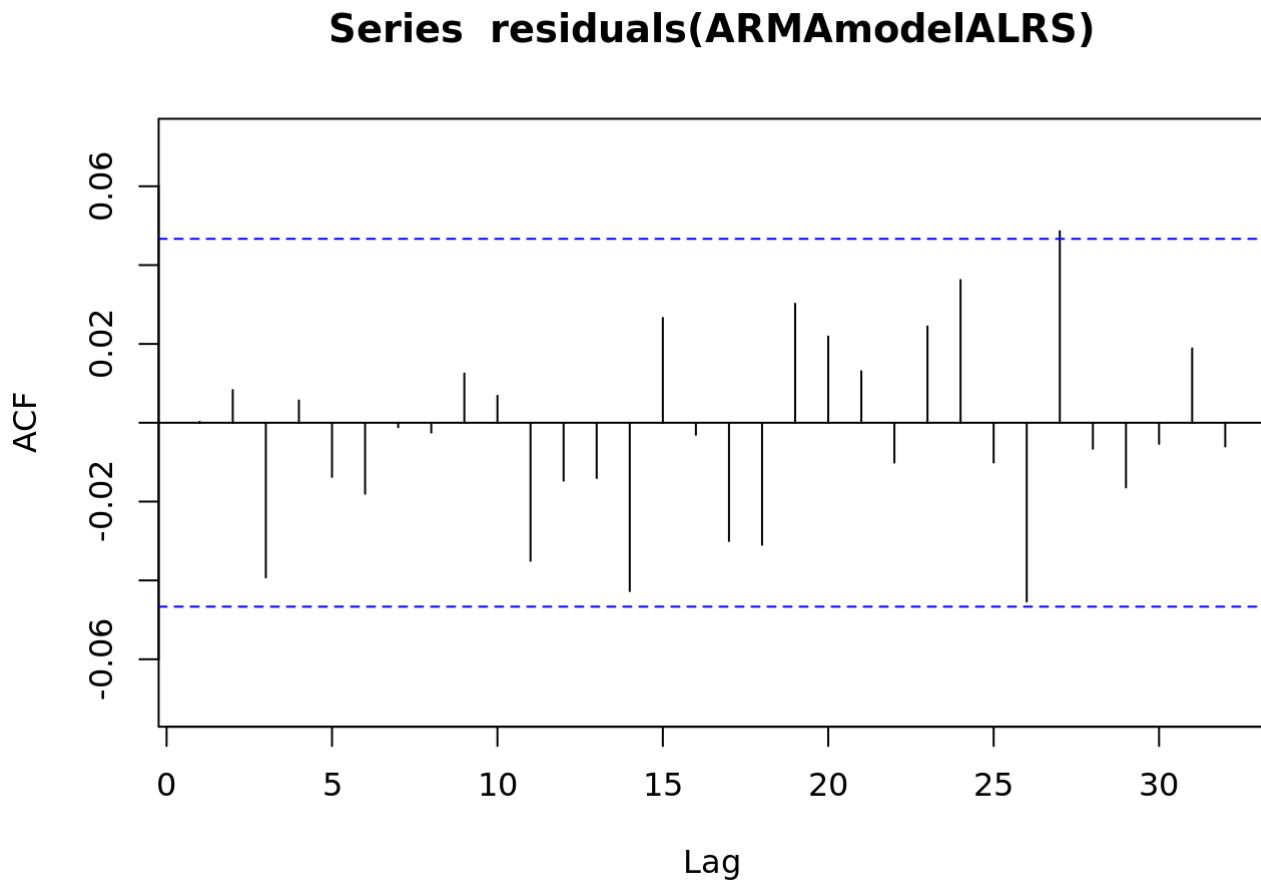
## 6.3. d1ALRS

```

ARMAmodelALRS = Arima(data$ALRS, c(0,1,1), include.constant=TRUE, method = c("CSS-ML"))
)

Acf(residuals(ARMAmodelALRS))

```



```
Box.test(residuals(ARMAmodelALRS), lag = 7, type = c("Ljung-Box"), fitdf = 1)
```

```

## 
## Box-Ljung test
## 
## data: residuals(ARMAmodelALRS)
## X-squared = 3.8173, df = 6, p-value = 0.7014

```

```
shapiro.test(residuals(ARMAmodelALRS))
```

```

## 
## Shapiro-Wilk normality test
## 
## data: residuals(ARMAmodelALRS)
## W = 0.97255, p-value < 2.2e-16

```

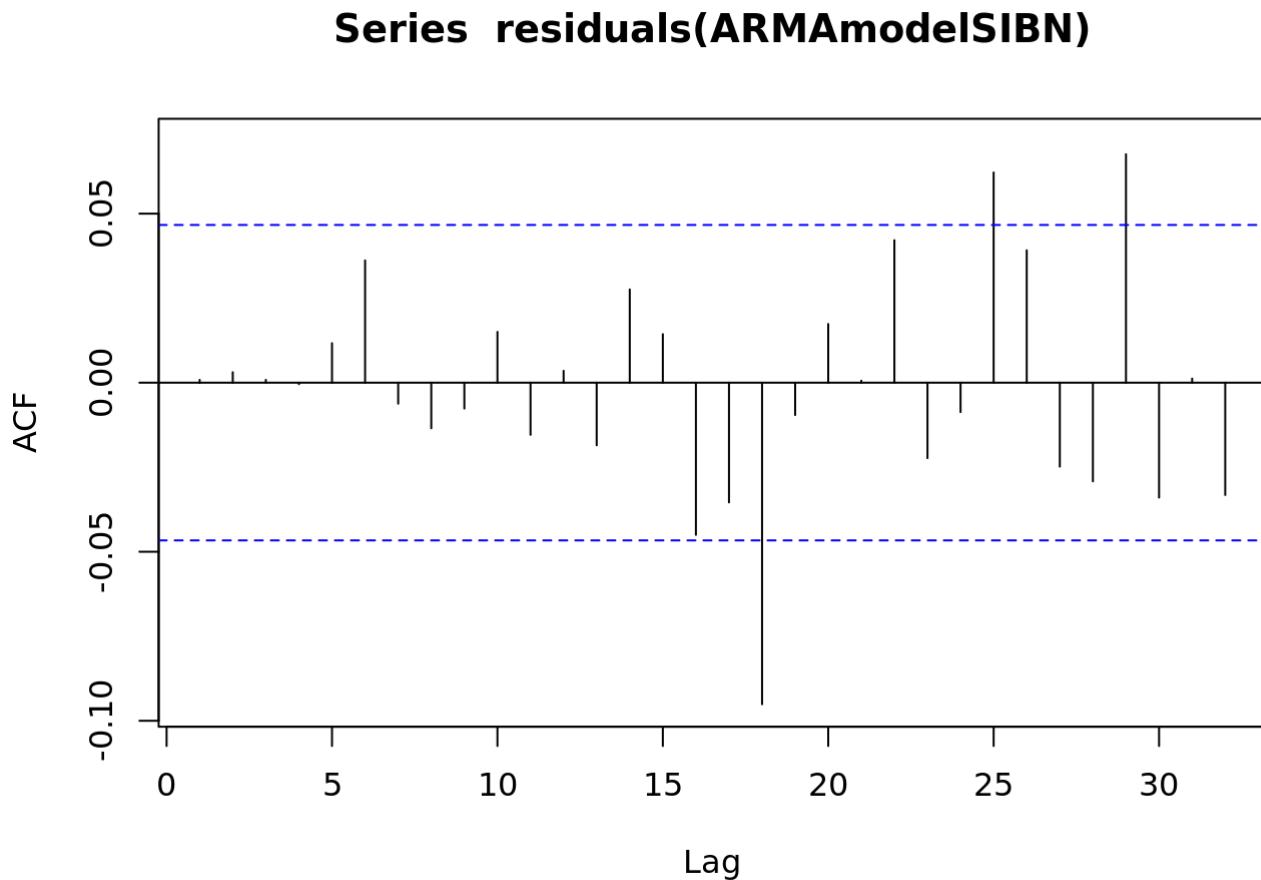
## 6.4. d1SIBN

```

ARMAmodelSIBN = Arima(data$SIBN, c(0,1,4), include.constant=TRUE, method = c("CSS-ML"))
)

Acf(residuals(ARMAmodelSIBN))

```



```
Box.test(residuals(ARMAmodelSIBN), lag = 7, type = c("Ljung-Box"), fitdf = 4)
```

```

## 
## Box-Ljung test
## 
## data: residuals(ARMAmodelSIBN)
## X-squared = 2.6499, df = 3, p-value = 0.4488

```

```
shapiro.test(residuals(ARMAmodelSIBN))
```

```

## 
## Shapiro-Wilk normality test
## 
## data: residuals(ARMAmodelSIBN)
## W = 0.86721, p-value < 2.2e-16

```

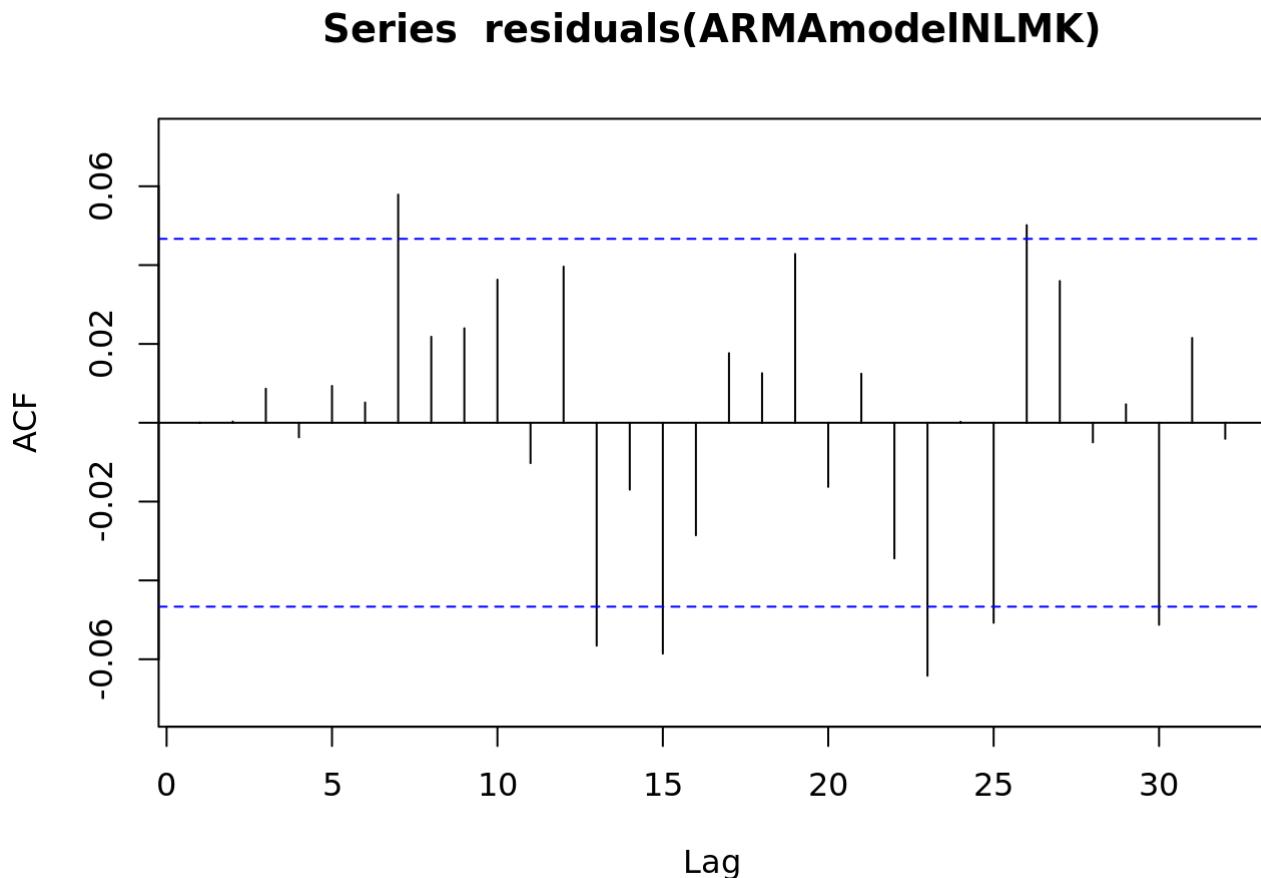
## 6.5. d1NLMK

```

ARMAmodelNLMK = Arima(data$NLMK, c(0,1,1), include.constant=TRUE, method = c("CSS-ML"))
)

Acf(residuals(ARMAmodelNLMK))

```



```
Box.test(residuals(ARMAmodelNLMK), lag = 7, type = c("Ljung-Box"), fitdf = 1)
```

```

## 
## Box-Ljung test
## 
## data: residuals(ARMAmodelNLMK)
## X-squared = 6.3029, df = 6, p-value = 0.3901

```

```
shapiro.test(residuals(ARMAmodelNLMK))
```

```

## 
## Shapiro-Wilk normality test
## 
## data: residuals(ARMAmodelNLMK)
## W = 0.96844, p-value < 2.2e-16

```

## 6.6. d1KMAZ

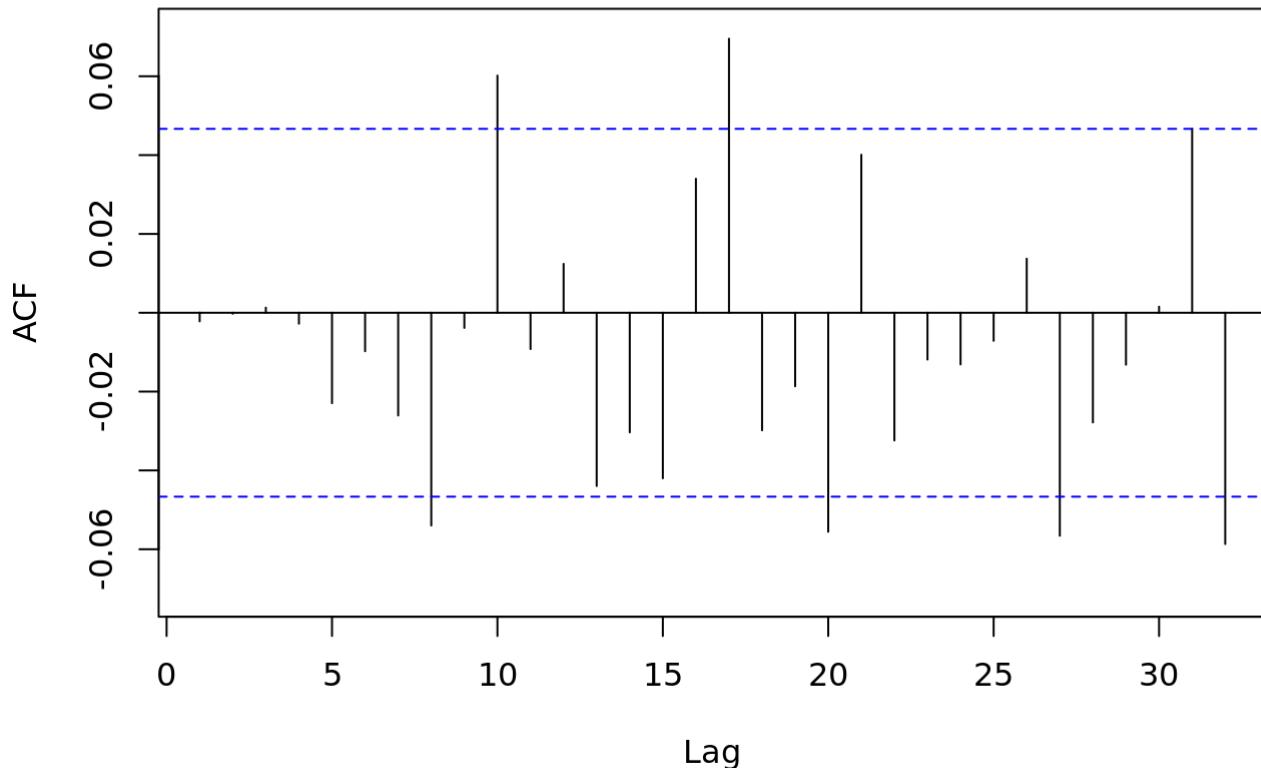
```

ARMAmodelKMAZ = Arima(data$KMAZ, c(0,1,4), include.constant=TRUE, method = c("CSS-ML"))
)

Acf(residuals(ARMAmodelKMAZ))

```

## Series residuals(ARMAmodelKMAZ)



```
Box.test(residuals(ARMAmodelKMAZ), lag = 7, type = c("Ljung-Box"), fitdf = 4)
```

```

## 
## Box-Ljung test
## 
## data: residuals(ARMAmodelKMAZ)
## X-squared = 2.3316, df = 3, p-value = 0.5065

```

```
shapiro.test(residuals(ARMAmodelKMAZ))
```

```

## 
## Shapiro-Wilk normality test
## 
## data: residuals(ARMAmodelKMAZ)
## W = 0.6537, p-value < 2.2e-16

```

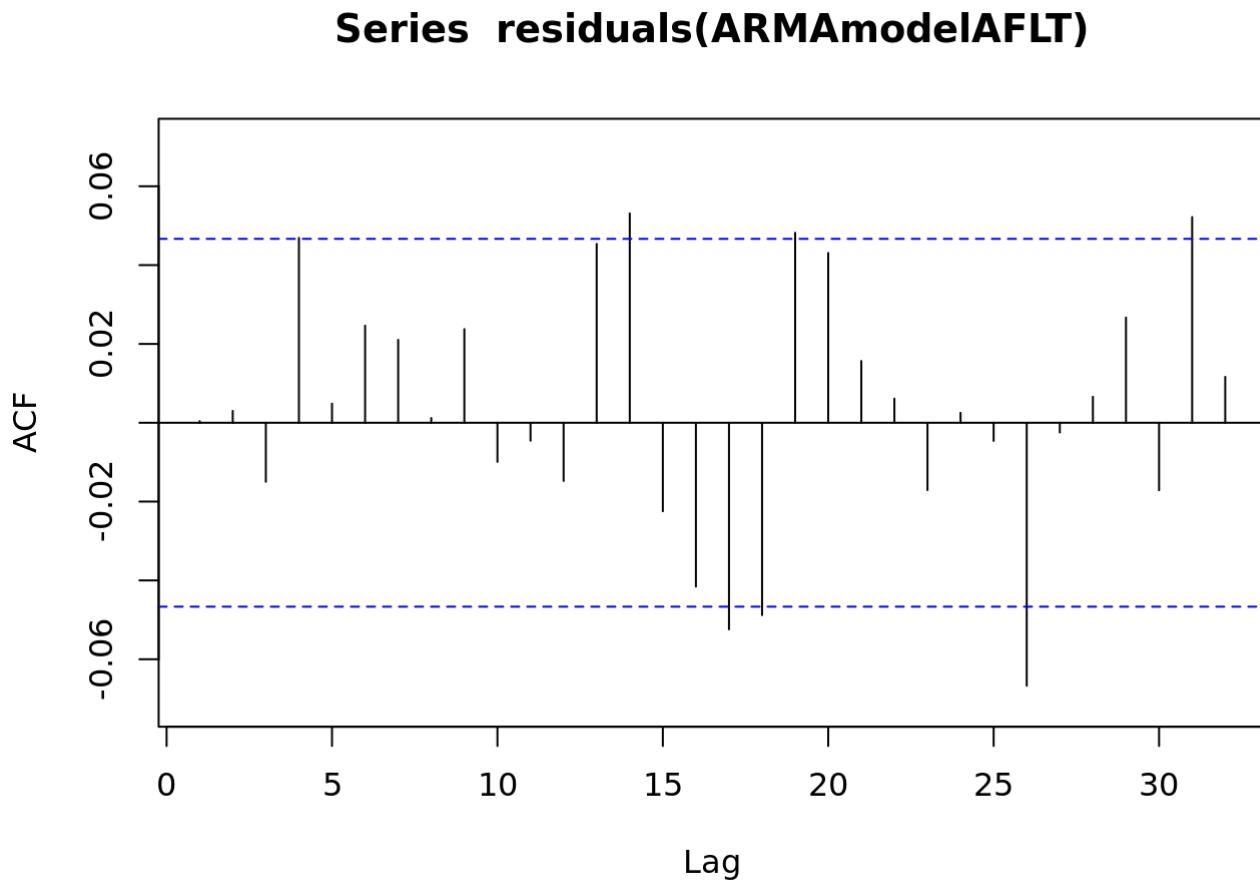
## 6.7. d1AFLT

```

ARMAmodelAFLT = Arima(data$AFLT, c(0,1,1), include.constant=TRUE, method = c("CSS-ML"))
)

Acf(residuals(ARMAmodelAFLT))

```



```
Box.test(residuals(ARMAmodelAFLT), lag = 7, type = c("Ljung-Box"), fitdf = 1)
```

```

## 
## Box-Ljung test
## 
## data: residuals(ARMAmodelAFLT)
## X-squared = 6.212, df = 6, p-value = 0.3999

```

```
shapiro.test(residuals(ARMAmodelAFLT))
```

```

## 
## Shapiro-Wilk normality test
## 
## data: residuals(ARMAmodelAFLT)
## W = 0.89919, p-value < 2.2e-16

```

## 6.8. Вывод:

Box-Ljung test: p-value > 0.05 - автокорреляция остатков отсутствует на уровне значимости 5%, на графиках можно увидеть аналогичный результат.

Shapiro-Wilk normality test: p-value < 2.2e-16 - гипотеза о нормальном распределении отвергается.

## 7. Проверка данных на волатильность через GARCH

Параметры и тип модели GARCH выбирались опытным путем через сравнение моделей по показателям LogLikelihood и Information Criteria.

При расчетах используется первая разность.

Также каждая модель проверяется на автокорреляцию стандартизованных остатков и остатков в квадрате с помощью Box-Ljung теста.

### 7.1. d1TATN

```
spec_TATN = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1, 1)),
                        mean.model = list(armaOrder = c(0, 2), include.mean = TRUE),
                        distribution.model = "norm")
garch.TATN = ugarchfit(spec_TATN, d1TATN)

Box.test(residuals(garch.TATN, standardize="TRUE"), lag = 7, type = c("Ljung-Box"),
         fitdf = 2)
```

```
##  
## Box-Ljung test  
##  
## data: residuals(garch.TATN, standardize = "TRUE")  
## X-squared = 4.9887, df = 5, p-value = 0.4173
```

```
Box.test(residuals(garch.TATN, standardize="TRUE")^2, lag = 7, type = c("Ljung-Box"),
         fitdf = 2)
```

```
##  
## Box-Ljung test  
##  
## data: residuals(garch.TATN, standardize = "TRUE")^2  
## X-squared = 5.668, df = 5, p-value = 0.3399
```

### 7.2. d1SNGS

```
spec_SNGS = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1, 1)),
                        mean.model = list(armaOrder = c(0, 1), include.mean = TRUE),
                        distribution.model = "std")
garch.SNGS = ugarchfit(spec_SNGS, d1SNGS)

Box.test(residuals(garch.SNGS, standardize="TRUE"), lag = 7, type = c("Ljung-Box"),
         fitdf = 1)
```

```

## 
## Box-Ljung test
## 
## data: residuals(garch.SNGS, standardize = "TRUE")
## X-squared = 3.561, df = 6, p-value = 0.7358

```

```

Box.test(residuals(garch.SNGS, standardize="TRUE")^2, lag = 7, type = c("Ljung-Box"),
fitdf = 1)

```

```

## 
## Box-Ljung test
## 
## data: residuals(garch.SNGS, standardize = "TRUE")^2
## X-squared = 0.84517, df = 6, p-value = 0.9908

```

## 7.3. d1ALRS

```

spec_ALRS = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1, 1)),
                        mean.model = list(armaOrder = c(0, 1), include.mean = TRUE), distribution.model = "norm")
garch.ALRS = ugarchfit(spec_ALRS, d1ALRS)

Box.test(residuals(garch.ALRS, standardize="TRUE"), lag = 7, type = c("Ljung-Box"),
fitdf = 1)

```

```

## 
## Box-Ljung test
## 
## data: residuals(garch.ALRS, standardize = "TRUE")
## X-squared = 6.9212, df = 6, p-value = 0.3282

```

```

Box.test(residuals(garch.ALRS, standardize="TRUE")^2, lag = 7, type = c("Ljung-Box"),
fitdf = 1)

```

```

## 
## Box-Ljung test
## 
## data: residuals(garch.ALRS, standardize = "TRUE")^2
## X-squared = 7.4364, df = 6, p-value = 0.2824

```

## 7.4. d1SIBN

```

spec_SIBN = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1, 3)),
                      mean.model = list(armaOrder = c(0, 4), include.mean = TRUE),
                      distribution.model = "std")
garch.SIBN = ugarchfit(spec_SIBN, d1SIBN)

Box.test(residuals(garch.SIBN, standardize="TRUE"), lag = 7, type = c("Ljung-Box"),
         fitdf = 4)

```

```

## 
## Box-Ljung test
##
## data: residuals(garch.SIBN, standardize = "TRUE")
## X-squared = 5.5013, df = 3, p-value = 0.1386

```

```

Box.test(residuals(garch.SIBN, standardize="TRUE")^2, lag = 7, type = c("Ljung-Box"),
         fitdf = 4)

```

```

## 
## Box-Ljung test
##
## data: residuals(garch.SIBN, standardize = "TRUE")^2
## X-squared = 4.5245, df = 3, p-value = 0.2101

```

## 7.5. d1NLMK

```

spec_NLMK = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1, 1)),
                       mean.model = list(armaOrder = c(0, 1), include.mean = TRUE),
                       distribution.model = "std")
garch.NLMK = ugarchfit(spec_NLMK, d1NLMK)

Box.test(residuals(garch.NLMK, standardize="TRUE"), lag = 7, type = c("Ljung-Box"),
         fitdf = 1)

```

```

## 
## Box-Ljung test
##
## data: residuals(garch.NLMK, standardize = "TRUE")
## X-squared = 3.7148, df = 6, p-value = 0.7152

```

```

Box.test(residuals(garch.NLMK, standardize="TRUE")^2, lag = 7, type = c("Ljung-Box"),
         fitdf = 1)

```

```

## 
## Box-Ljung test
##
## data: residuals(garch.NLMK, standardize = "TRUE")^2
## X-squared = 6.6174, df = 6, p-value = 0.3577

```

## 7.6. d1KMAZ

```
spec_KMAZ = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1, 3)),
                        mean.model = list(armaOrder = c(0, 4), include.mean = TRUE), distribution.model = "std")
garch.KMAZ = ugarchfit(spec_KMAZ, d1KMAZ)

Box.test(residuals(garch.KMAZ, standardize="TRUE"), lag = 7, type = c("Ljung-Box"),
         fitdf = 4)
```

```
## 
## Box-Ljung test
##
## data: residuals(garch.KMAZ, standardize = "TRUE")
## X-squared = 13.125, df = 3, p-value = 0.004373
```

```
Box.test(residuals(garch.KMAZ, standardize="TRUE")^2, lag = 7, type = c("Ljung-Box"),
         fitdf = 4)
```

```
## 
## Box-Ljung test
##
## data: residuals(garch.KMAZ, standardize = "TRUE")^2
## X-squared = 0.52356, df = 3, p-value = 0.9137
```

## 7.7. d1AFLT

```
spec_AFLT = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1, 3)),
                        mean.model = list(armaOrder = c(0, 1), include.mean = TRUE), distribution.model = "std")
garch.AFLT = ugarchfit(spec_AFLT, d1AFLT)

Box.test(residuals(garch.AFLT, standardize="TRUE"), lag = 7, type = c("Ljung-Box"),
         fitdf = 1)
```

```
## 
## Box-Ljung test
##
## data: residuals(garch.AFLT, standardize = "TRUE")
## X-squared = 11.503, df = 6, p-value = 0.07403
```

```
Box.test(residuals(garch.AFLT, standardize="TRUE")^2, lag = 7, type = c("Ljung-Box"),
         fitdf = 1)
```

```
## 
## Box-Ljung test
##
## data: residuals(garch.AFLT, standardize = "TRUE")^2
## X-squared = 2.0451, df = 6, p-value = 0.9155
```

## 7.8. Вывод:

Во всех случаях в стандартизированных остатках в квадрате отсутствует автокорреляция, что является признаком хорошей модели.

# 8. Тест Энглера-Гренджера на парную коинтеграцию

На коинтеграцию в тесте Энглера-Гренджера проверяются все нестационарные данные. В нашем случае - все начальные ряды.

Критическое значение на уровне 5% без тренда = -3.34, на уровне 1% без тренда = -3.90. Данные коинтегрированы при значении t-статистики < критического.

## 8.1. TATN\_p

```
coint_TATN_SNGS = lm(data$TATN_p ~ data$SNGS_p)
coint_TATN_ALRS = lm(data$TATN_p ~ data$ALRS)
coint_TATN_SIBN = lm(data$TATN_p ~ data$SIBN)
coint_TATN_NLMK = lm(data$TATN_p ~ data$NLMK)
coint_TATN_KMAZ = lm(data$TATN_p ~ data$KMAZ)
coint_TATN_AFLT = lm(data$TATN_p ~ data$AFLT)

ur.df(coint_TATN_SNGS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -1.5013 1.4453
```

```
ur.df(coint_TATN_ALRS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -1.446 1.1099
```

```
ur.df(coint_TATN_SIBN$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: 0.2291 0.4253
```

```
ur.df(coint_TATN_NLMK$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -1.4745 1.0874
```

```
ur.df(coint_TATN_KMAZ$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -1.3097 0.9066
```

```
ur.df(coint_TATN_AFLT$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -1.4967 1.4806
```

С TATN\_p коинтеграции нет.

## 8.2. SNGS\_p

```
coint_SNGS_TANT = lm(data$SNGS_p ~ data$TATN_p)  
coint_SNGS_ALRS = lm(data$SNGS_p ~ data$ALRS)  
coint_SNGS_SIBN = lm(data$SNGS_p ~ data$SIBN)  
coint_SNGS_NLMK = lm(data$SNGS_p ~ data$NLMK)  
coint_SNGS_KMAZ = lm(data$SNGS_p ~ data$KMAZ)  
coint_SNGS_AFLT = lm(data$SNGS_p ~ data$AFLT)
```

```
ur.df(coint_SNGS_TANT$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -2.8917 4.2211
```

```
ur.df(coint_SNGS_ALRS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -2.895 4.23
```

```
ur.df(coint_SNGS_SIBN$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -2.8964 4.2235
```

```
ur.df(coint_SNGS_NLMK$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -2.897 4.2225
```

```
ur.df(coint_SNGS_KMAZ$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -2.9049 4.2601
```

```
ur.df(coint_SNGS_AFLT$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -5.4669 15.0145
```

С SNGS\_р коинтегрированна только переменная AFLT.

## 8.3. ALRS

```
coint_ALRS_TANT = lm(data$ALRS ~ data$TATN_p)
coint_ALRS_SNGS = lm(data$ALRS ~ data$SNGS_p)
coint_ALRS_SIBN = lm(data$ALRS ~ data$SIBN)
coint_ALRS_NLMK = lm(data$ALRS ~ data$NLMK)
coint_ALRS_KMAZ = lm(data$ALRS ~ data$KMAZ)
coint_ALRS_AFLT = lm(data$ALRS ~ data$AFLT)

ur.df(coint_ALRS_TANT$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
## 
## ##### # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### # The value of the test statistic is: -1.4831 1.3459
```

```
ur.df(coint_ALRS_SNGS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
## 
## ##### # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### # The value of the test statistic is: -1.4893 1.4717
```

```
ur.df(coint_ALRS_SIBN$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
## 
## ##### # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### # The value of the test statistic is: -2.2661 2.59
```

```
ur.df(coint_ALRS_NLMK$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
## 
## ##### # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### # The value of the test statistic is: -2.7277 3.7439
```

```
ur.df(coint_ALRS_KMAZ$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -3.0689 4.7165

```

```
ur.df(coint_ALRS_AFLT$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -1.3729 1.3105

```

С ALRS коинтеграции нет.

## 8.4. SIBN

```

coint_SIBN_TANT = lm(data$SIBN ~ data$TATN_p)
coint_SIBN_SNGS = lm(data$SIBN ~ data$SNGS_p)
coint_SIBN_ALRS = lm(data$SIBN ~ data$ALRS)
coint_SIBN_NLMK = lm(data$SIBN ~ data$NLMK)
coint_SIBN_KMAZ = lm(data$SIBN ~ data$KMAZ)
coint_SIBN_AFLT = lm(data$SIBN ~ data$AFLT)

ur.df(coint_SIBN_TANT$residuals, type="drift", lags = 1, selectlags = "Fixed")

```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: 1.1736 1.5028

```

```
ur.df(coint_SIBN_SNGS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: 0.0343 1.5387

```

```
ur.df(coint_SIBN_ALRS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -1.4087 1.2166

```

```
ur.df(coint_SIBN_NLMK$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -1.3987 1.157

```

```
ur.df(coint_SIBN_KMAZ$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -2.9036 4.2163

```

```
ur.df(coint_SIBN_AFLT$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -0.0155 1.6982

```

С SIBN коинтеграции нет.

## 8.5. NLMK

```

coint_NLMK_TANT = lm(data$NLMK ~ data$TATN_p)
coint_NLMK_SNGS = lm(data$NLMK ~ data$SNGS_p)
coint_NLMK_ALRS = lm(data$NLMK ~ data$ALRS)
coint_NLMK_SIBN = lm(data$NLMK ~ data$SIBN)
coint_NLMK_KMAZ = lm(data$NLMK ~ data$KMAZ)
coint_NLMK_AFLT = lm(data$NLMK ~ data$AFLT)

ur.df(coint_NLMK_TANT$residuals, type="drift", lags = 1, selectlags = "Fixed")

```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -1.0397 0.8438
```

```
ur.df(coint_NLMK_SNGS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -0.9367 1.2336
```

```
ur.df(coint_NLMK_ALRS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -2.471 3.0671
```

```
ur.df(coint_NLMK_SIBN$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -1.9031 1.8112
```

```
ur.df(coint_NLMK_KMAZ$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -2.4753 3.0933
```

```
ur.df(coint_NLMK_AFLT$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -0.9753 1.5147

```

С NLMK коинтеграции нет.

## 8.6. KMAZ

```

coint_KMAZ_TANT = lm(data$KMAZ ~ data$TATN_p)
coint_KMAZ_SNGS = lm(data$KMAZ ~ data$SNGS_p)
coint_KMAZ_ALRS = lm(data$KMAZ ~ data$ALRS)
coint_KMAZ_SIBN = lm(data$KMAZ ~ data$SIBN)
coint_KMAZ_NLMK = lm(data$KMAZ ~ data$NLMK)
coint_KMAZ_AFLT = lm(data$KMAZ ~ data$AFLT)

ur.df(coint_KMAZ_TANT$residuals, type="drift", lags = 1, selectlags = "Fixed")

```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -1.0175 1.0046

```

```
ur.df(coint_KMAZ_SNGS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -0.9542 1.2625

```

```
ur.df(coint_KMAZ_ALRS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -2.7516 3.9455

```

```
ur.df(coint_KMAZ_SIBN$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -3.1226 4.9635

```

```
ur.df(coint_KMAZ_NLMK$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -2.416 3.1114

```

```
ur.df(coint_KMAZ_AFLT$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -0.8838 1.2057

```

С KMAZ коинтеграции нет.

## 8.7. AFLT

```

coint_AFLT_TANT = lm(data$AFLT ~ data$TATN_p)
coint_AFLT_SNGS = lm(data$AFLT ~ data$SNGS_p)
coint_AFLT_ALRS = lm(data$AFLT ~ data$ALRS)
coint_AFLT_SIBN = lm(data$AFLT ~ data$SIBN)
coint_AFLT_NLMK = lm(data$AFLT ~ data$NLMK)
coint_AFLT_KMAZ = lm(data$AFLT ~ data$KMAZ)

```

```
ur.df(coint_AFLT_TANT$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```

## 
## ##### #####
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ##### #####
## 
## The value of the test statistic is: -1.3352 0.9198

```

```
ur.df(coint_AFLT_SNGS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -4.8218 11.7055
```

```
ur.df(coint_AFLT_ALRS$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -1.1566 0.6728
```

```
ur.df(coint_AFLT_SIBN$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -1.4083 1.0331
```

```
ur.df(coint_AFLT_NLMK$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -1.3966 1.0141
```

```
ur.df(coint_AFLT_KMAZ$residuals, type="drift", lags = 1, selectlags = "Fixed")
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #  
## #####  
##  
## The value of the test statistic is: -1.2597 0.8071
```

С AFLT коинтегрированна только переменная SNGS\_p.

## 8.8. Вывод:

Из всех данных коинтегрированными на 1% уровне является только пара SNGS\_p и AFLT. Данную коинтеграцию можно использовать в качестве основной модели для построения совместного прогноза.

Для остальных переменных - переходить на первую разность и строить вектор авторегрессионной модели.

## 9. Тест Johansen на коинтеграцию всех данных

Проверим все акции на коинтеграцию:

```
df_all = data %>% dplyr::select(-date)
coint_jo_all = ca.jo(df_all, ecdet = "const", type = "eigen", K=2, spec = "transitor
y", season = NULL)
coint_jo_all
```

```
## 
## ##### Johansen-Procedure Unit Root / Cointegration Test #
## #####
## The value of the test statistic is: 2.7395 3.8481 8.4134 24.8913 30.5814 34.892 4
3.147
```

На уровне  $r = 0: 43.15 < 51.91 \Rightarrow$  нулевая гипотеза (коинтеграция между 7 переменными отсутствует) не отвергается, то есть коинтеграции между выбранными нами акциями не наблюдается.

Таким образом, для совместного прогноза берем разности по всем данным и строим VAR model.

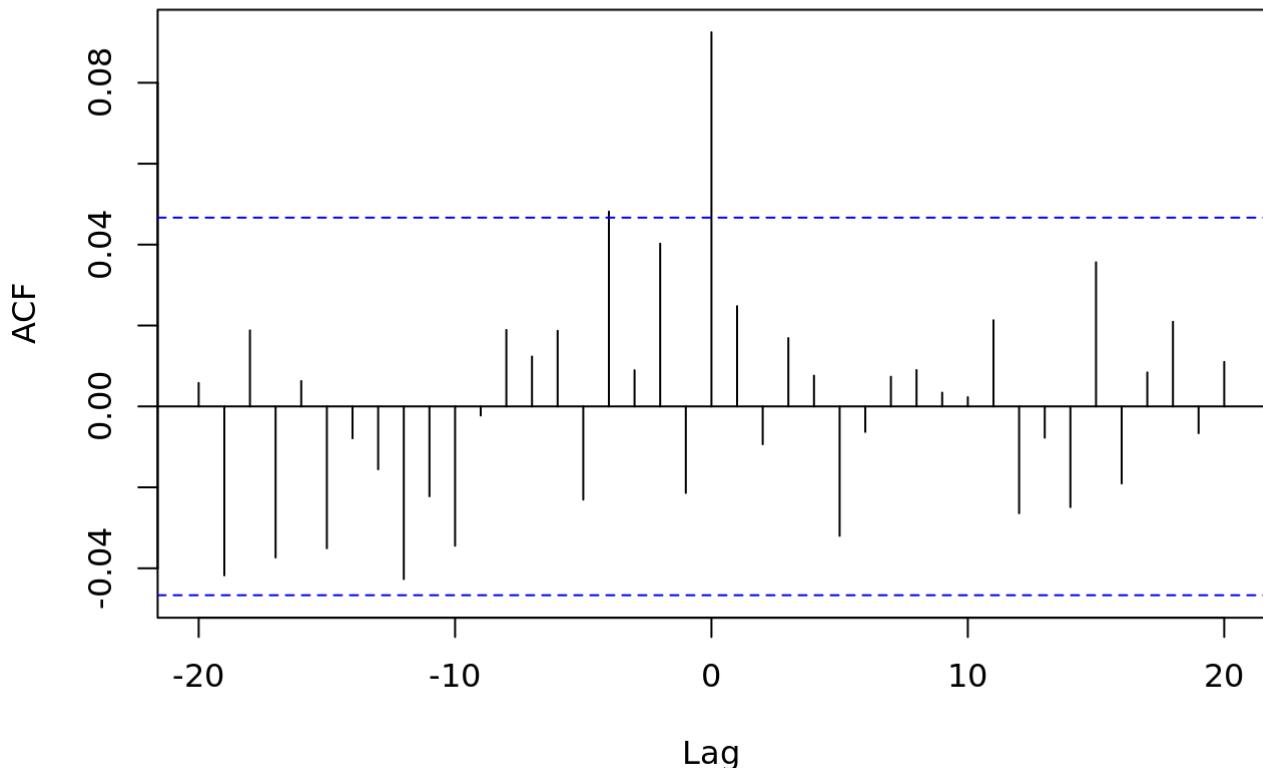
## 10. Анализ кросс-корреляции между переменными, причинность по Грэнджеру

Целевая переменная - TATN\_p.

### 10.1. TATN & SNGS

```
ccf(d1TATN, d1SNGS, lag.max = 20, type = c("correlation"), plot = TRUE)
```

## d1TATN & d1SNGS



```
grangertest(d1SNGS, d1TATN, order = 1)
```

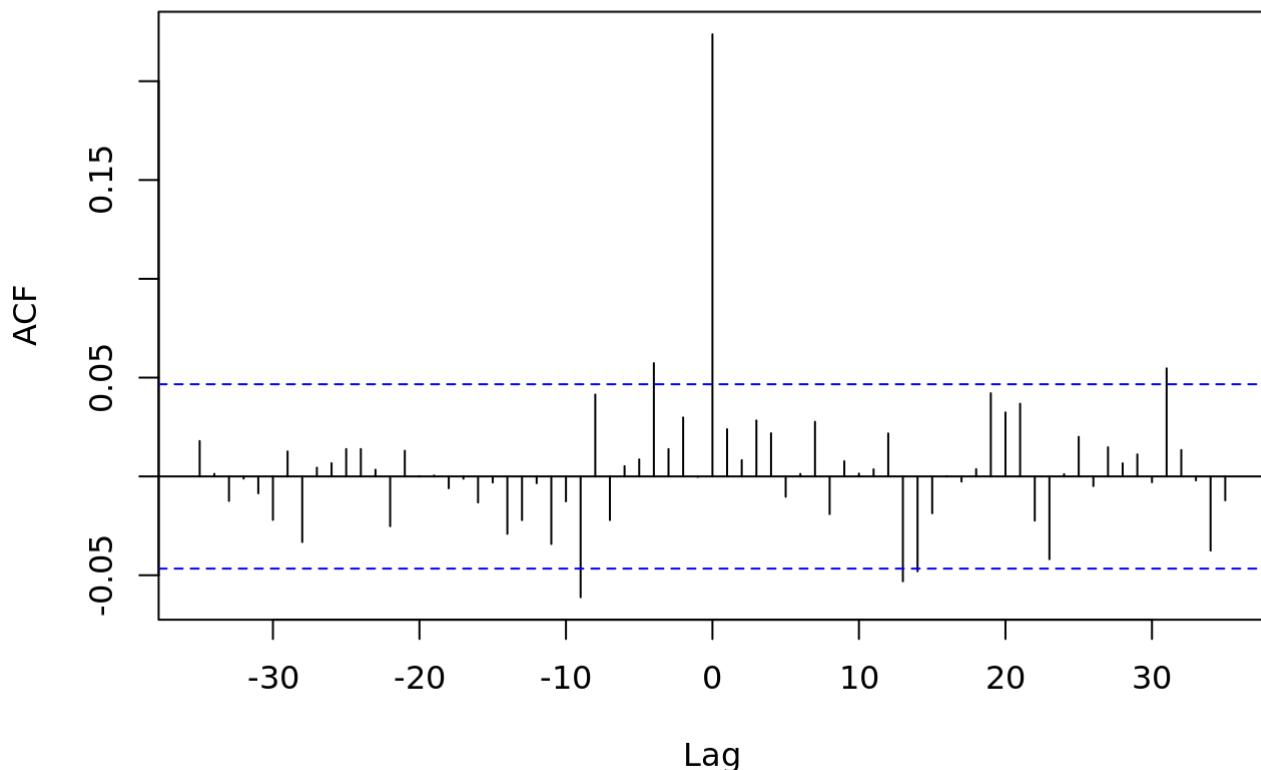
	Res.Df <dbl>	Df <dbl>	F <dbl>	Pr(>F) <dbl>
1	1760	NA	NA	NA
2	1761	-1	0.486094	0.4857669
2 rows				

p-value = 0.4858 => влияния нет

## 10.2. TATN & ALRS

```
ccf(d1TATN, d1ALRS, lag.max = 35, type = c("correlation"), plot = TRUE)
```

## d1TATN & d1ALRS



```
grangertest(d1ALRS, d1TATN, order = 31)
```

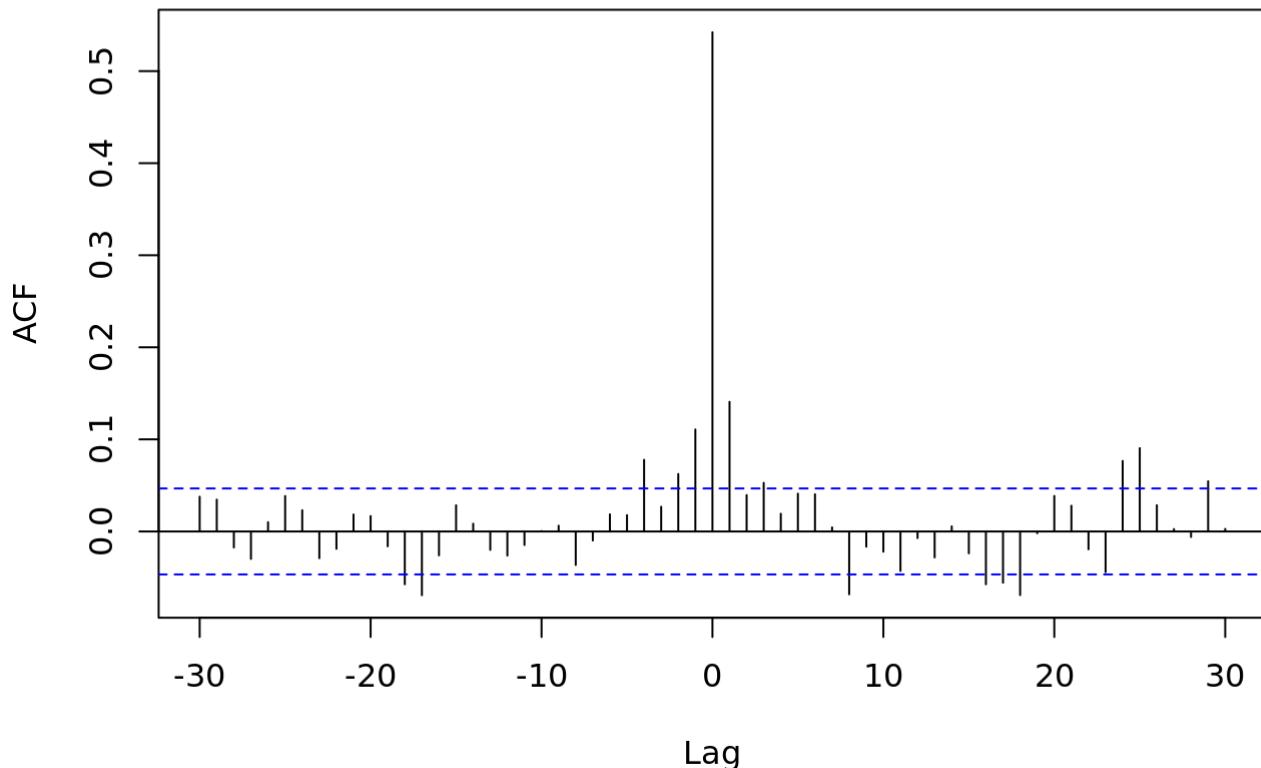
	Res.Df <dbl>	Df <dbl>	F <dbl>	Pr(>F) <dbl>
1	1670	NA	NA	NA
2	1701	-31	1.072281	0.3604539
2 rows				

p-value = 0.3605 => влияния нет

## 10.3. TATN & SIBN

```
ccf(d1TATN, d1SIBN, lag.max = 30, type = c("correlation"), plot = TRUE)
```

## d1TATN & d1SIBN



```
grangertest(d1SIBN, d1TATN, order = 29)
```

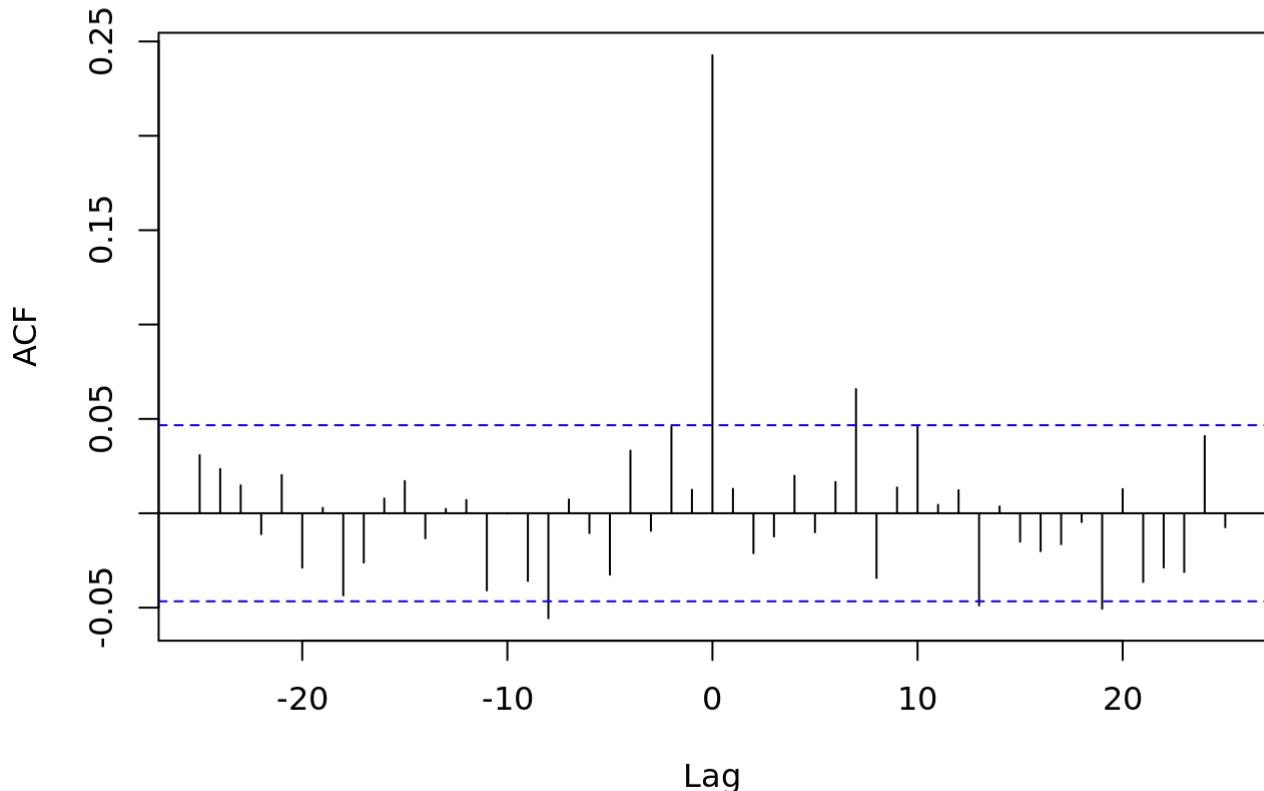
	Res.Df <dbl>	Df <dbl>	F <dbl>	Pr(>F) <dbl>
1	1676	NA	NA	NA
2	1705	-29	2.711155	2.717852e-06
2 rows				

p-value = 2.718e-06 => влияние есть

## 10.4. TATN & NLMK

```
ccf(d1TATN, d1NLMK, lag.max = 25, type = c("correlation"), plot = TRUE)
```

## d1TATN & d1NLMK



```
grangertest(d1NLMK, d1TATN, order = 19)
```

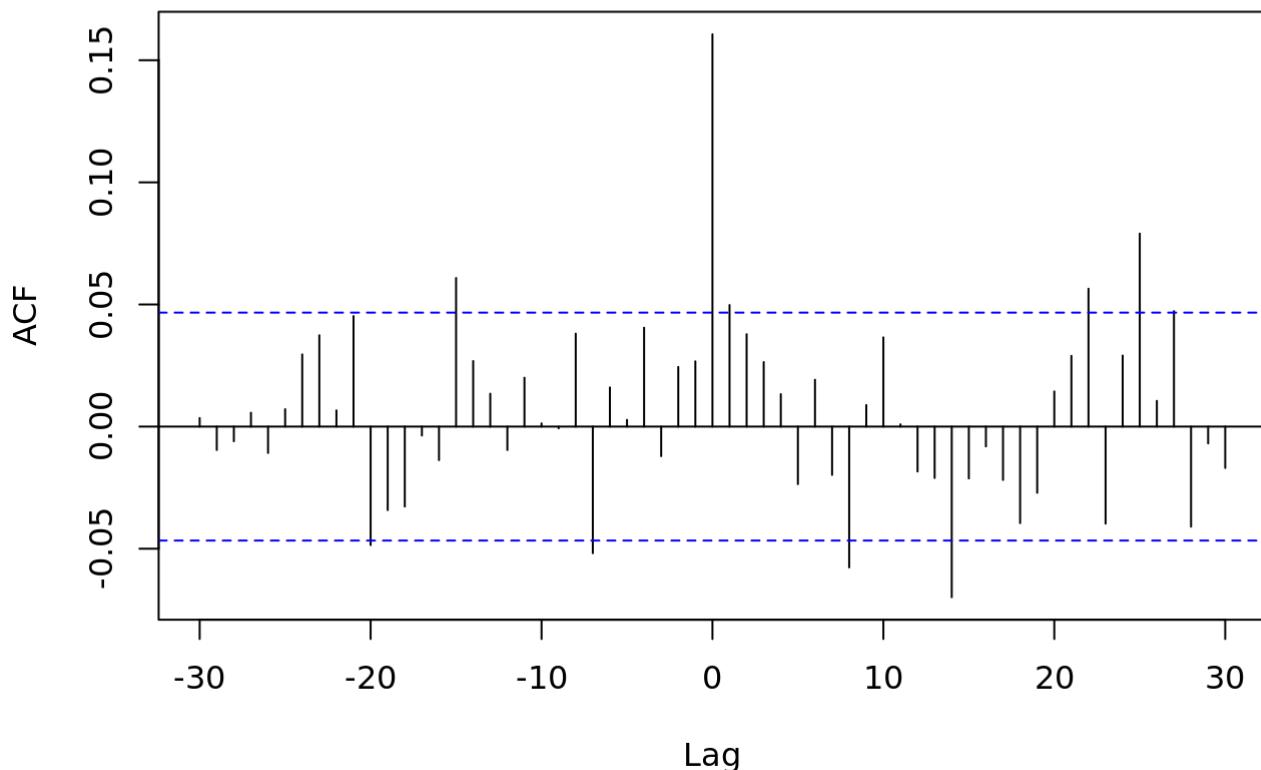
	Res.Df <dbl>	Df <dbl>	F <dbl>	Pr(>F) <dbl>
1	1706	NA	NA	NA
2	1725	-19	1.926688	0.009423283
2 rows				

p-value = 0.009423 => влияние есть

## 10.5. TATN & KMAZ

```
ccf(d1TATN, d1KMAZ, lag.max = 30, type = c("correlation"), plot = TRUE)
```

## d1TATN & d1KMAZ



```
grangertest(d1KMAZ, d1TATN, order = 25)
```

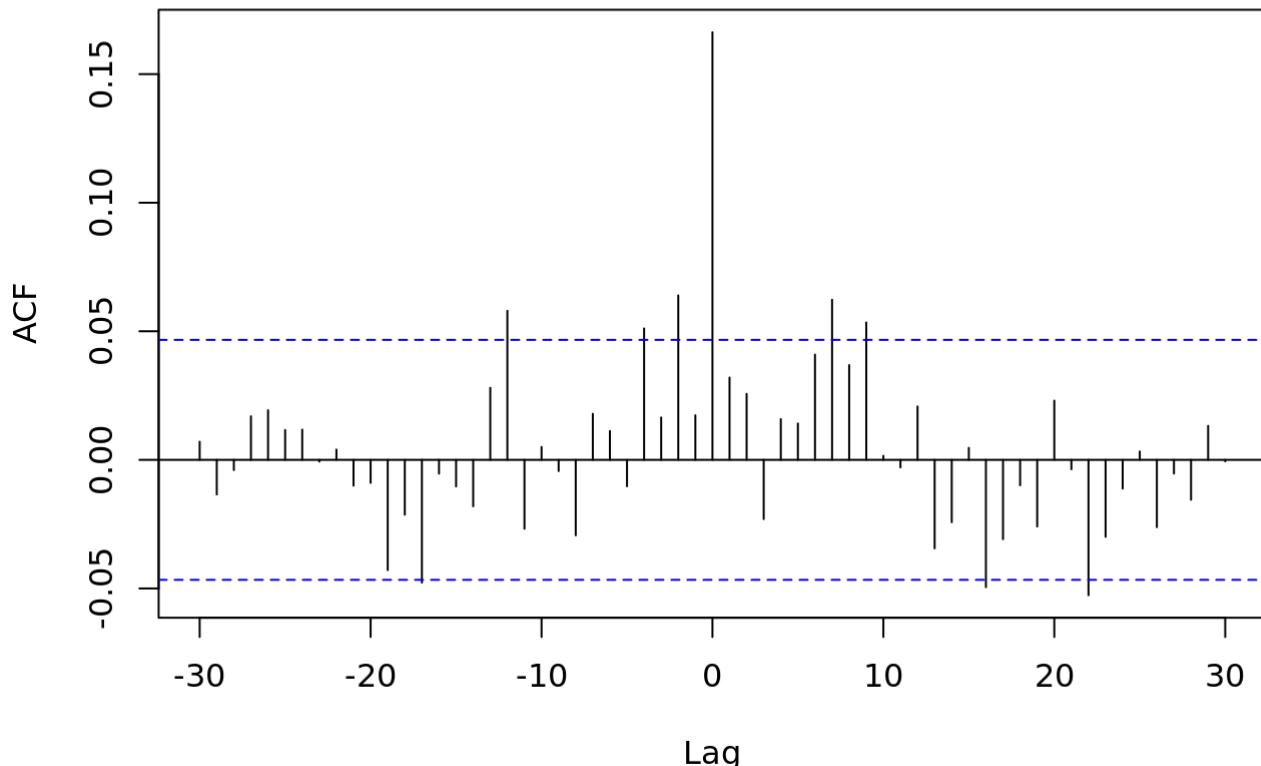
	Res.Df <dbl>	Df <dbl>	F <dbl>	Pr(>F) <dbl>
1	1688	NA	NA	NA
2	1713	-25	1.873928	0.005577233
2 rows				

p-value = 0.005577 => влияние есть

## 10.6. TATN & AFLT

```
ccf(d1TATN, d1AFLT, lag.max = 30, type = c("correlation"), plot = TRUE)
```

## d1TATN & d1AFLT



```
grangertest(d1AFLT, d1TATN, order = 22)
```

	Res.Df <dbl>	Df <dbl>	F <dbl>	Pr(>F) <dbl>
1	1697	NA	NA	NA
2	1719	-22	1.515455	0.05897031
2 rows				

p-value = 0.05897 => влияние есть

## 10.7. Вывод:

Наиболее сильное влияние имеют SIBN, KMAZ, NLMK и AFLT.

## 11. VAR-модель

Целевая переменная анализа - привилегированные акции компании ПАО “Татнефть”.

### 11.1. Модель с 2-мя переменными: SIBN cause TATN

```
df = data.frame(d1TATN, d1SIBN)
VARselect(df, lag.max = 23, type = 'const')$selection
```

```
## AIC(n)  HQ(n)  SC(n)  FPE(n)
##      10       2       1      10
```

```
var = VAR(df, p = 10, type = 'const')
Hosking(var, lags=1.5*var$p)
```

```
##   lags statistic df    p-value
##     15   20.99677 20  0.3973226
```

```
LiMcLeod(var, lags=1.5*var$p)
```

```
##   lags statistic df    p-value
##     15   21.11511 20  0.3903855
```

Авто- и кросс-корреляции нет

### 11.1.1. Модель без ограничений (в сравнении с ARMA)

```
n = length(coef(ARMAmodelTANT)) -1
x = var$p-n+1
rss = sum(ARMAmodelTANT$residuals[x:length(residuals(ARMAmodelTANT))] ^2)
n1 = var$varresult$d1TATN$rank-1
ess = sum(var$varresult$d1TATN$residuals^2)

Fstat = ((rss-ess) / (n1-n)) / (ess / (length(d1TATN)-2*n1-1))
Fstat
```

```
## [1] 4.565615
```

```
qf(0.99, df1=n1-n, df2=length(d1TATN)-2*n1-1)
```

```
## [1] 1.944196
```

```
pf(Fstat, n1-n, length(d1TATN)-2*n1-1, lower.tail=F)
```

```
## [1] 6.587105e-10
```

p-value = 6.587105e-10 => VAR(2) model превосходит ARMA model.

### 11.1.2. Модель с ограничениями (в сравнении с ARMA)

```
resvar = restrict(var, method = c("ser"), thresh = 0.8)
n1 = resvar$varresult$d1TATN$rank-1
if(n1<n+1) {n1=n+1}
ess = sum(resvar$varresult$d1TATN$residuals^2)

Fstat = ((rss-ess)/(n1-n))/(ess/(length(d1TATN)-2*n1-1))
Fstat
```

```
## [1] 7.328169
```

```
qf(0.99, df1=n1-n, df2=length(d1TATN)-2*n1-1)
```

```
## [1] 2.257918
```

```
pf(Fstat, n1-n, length(d1TATN)-n-n1-1, lower.tail=F)
```

```
## [1] 2.29406e-12
```

```
Hosking(resvar, lags=1.5*var$p)
```

```
##   lags statistic df  p-value
##     15    25.81503 20  0.172008
```

```
LiMcLeod(resvar, lags=1.5*var$p)
```

```
##   lags statistic df  p-value
##     15    25.9219 20  0.1684068
```

F-критерий увеличился, при этом модель все еще лучше ARMA и не имеет авто- и кросс-корреляции остатков => превосходит модель без ограничений.

## 11.2. Модель с 3-мя переменными: SIBN, KMAZ cause TATN

```
df = data.frame(d1TATN, d1SIBN, d1KMAZ)
VARselect(df, lag.max = 21, type = 'const')$selection
```

```
## AIC(n)  HQ(n)  SC(n)  FPE(n)
##      10       2       1      10
```

```
var = VAR(df, p = 10, type = 'const')
Hosking(var, lags=1.3*var$p)
```

```
##   lags statistic df  p-value
##     13    26.19231 27  0.5079481
```

```
LiMcLeod(var, lags=1.3*var$p)
```

```
##   lags statistic df    p-value
##     13    26.48937 27 0.4915874
```

Авто- и кросс-корреляции нет

## 11.2.1. Модель без ограничений (в сравнении с ARMA)

```
n = length(coef(ARMAmodelTANT)) -1
x = var$p-n+1
rss = sum(ARMAmodelTANT$residuals[x:length(residuals(ARMAmodelTANT))] ^2)
n1 = var$varresult$d1TATN$rank-1
ess = sum(var$varresult$d1TATN$residuals^2)

Fstat = ((rss-ess) / (n1-n)) / (ess / (length(d1TATN)-2*n1-1))
Fstat
```

```
## [1] 3.456849
```

```
qf(0.99, df1=n1-n, df2=length(d1TATN)-2*n1-1)
```

```
## [1] 1.735509
```

```
pf(Fstat, n1-n, length(d1TATN)-2*n1-1, lower.tail=F)
```

```
## [1] 3.398337e-09
```

p-value = 3.398337e-09 => VAR(3) model превосходит ARMA model.

## 11.2.2. Модель с ограничениями (в сравнении с ARMA)

```
resvar = restrict(var, method = c("ser"), thresh = 0.7)
n1 = resvar$varresult$d1TATN$rank-1
if(n1<n1) {n1=n1}
ess = sum(resvar$varresult$d1TATN$residuals^2)

Fstat = ((rss-ess) / (n1-n)) / (ess / (length(d1TATN)-2*n1-1))
Fstat
```

```
## [1] 5.069553
```

```
qf(0.99, df1=n1-n, df2=length(d1TATN)-2*n1-1)
```

```
## [1] 1.915418
```

```
pf(Fstat, n1-n, length(d1TATN)-n-n1-1, lower.tail=F)
```

```
## [1] 5.914694e-12
```

```
Hosking(resvar, lags=1.3*var$p)
```

```
##   lags statistic df   p-value
##     13    34.46076 27 0.1530583
```

```
LiMcLeod(resvar, lags=1.3*var$p)
```

```
##   lags statistic df   p-value
##     13    34.73685 27 0.145658
```

F-критерий увеличился, при этом модель все еще лучше ARMA и не имеет авто- и кросс-корреляции остатков => превосходит модель без ограничений.

### 11.2.3. Сравнение модели с 2-мя и 3-мя переменными

```
#модель с 2-мя переменными
df = data.frame(d1TATN, d1SIBN)
var = VAR(df, p = 10, type = 'const')
resvar = restrict(var, method = c("ser"), thresh = 0.8)
n = resvar$varresult$d1TATN$rank-1
x = 10-10+1
rss = sum(resvar$varresult$d1TATN$residuals[x:length(resvar$varresult$d1TATN$residuals)]^2)

#модель с 3-мя переменными
df = data.frame(d1TATN, d1SIBN, d1KMAZ)
var = VAR(df, p = 10, type = 'const')
resvar = restrict(var, method = c("ser"), thresh = 0.7)
n1 = resvar$varresult$d1TATN$rank-1
ess = sum(resvar$varresult$d1TATN$residuals^2)

Fstat = ((rss-ess) / (n1-n)) / (ess / (length(d1TATN)-2*n1-1))
Fstat
```

```
## [1] 1.965555
```

```
qf(0.99, df1=n1-n, df2=length(d1TATN)-2*n1-1)
```

```
## [1] 2.52158
```

```
pf(Fstat, n1-n, length(d1TATN)-2*n1-1, lower.tail=F)
```

```
## [1] 0.04719159
```

F-критерий уменьшился, однако значение p-value на 5% уровне значимости говорит о том, что модель с тремя переменными корректна и лучше для объяснения d1TATN, чем с двумя.

## 11.3. Модель с 4-мя переменными: SIBN, KMAZ, NLMK cause TATN

```
df = data.frame(d1TATN, d1SIBN, d1KMAZ, d1NLMK)
VARselect(df, lag.max = 24, type = 'const')$selection
```

```
## AIC(n)  HQ(n)  SC(n)  FPE(n)
##      8       1       1       8
```

```
var = VAR(df, p = 8, type = 'const')
Hosking(var, lags=1.2*var$p)
```

```
##  lags statistic   df   p-value
##  9.6  29.88831 25.6 0.2543419
```

```
LiMcLeod(var, lags=1.2*var$p)
```

```
##  lags statistic   df   p-value
##  9.6  30.20763 25.6 0.2416713
```

Авто- и кросс-корреляции нет

### 11.3.1. Модель без ограничений (в сравнении с ARMA)

```
n = length(coef(ARMAmodelTANT)) - 1
x = var$p - n + 1
rss = sum(ARMAmodelTANT$residuals[x:length(residuals(ARMAmodelTANT))] ^ 2)
n1 = var$varresult$d1TATN$rank - 1
ess = sum(var$varresult$d1TATN$residuals ^ 2)

Fstat = ((rss - ess) / (n1 - n)) / (ess / (length(d1TATN) - 2 * n1 - 1))
Fstat
```

```
## [1] 3.148388
```

```
qf(0.99, df1 = n1 - n, df2 = length(d1TATN) - 2 * n1 - 1)
```

```
## [1] 1.707842
```

```
pf(Fstat, n1 - n, length(d1TATN) - 2 * n1 - 1, lower.tail = F)
```

```
## [1] 2.569228e-08
```

p-value = 2.569228e-08 => VAR(4) model превосходит ARMA model.

### 11.3.2. Модель с ограничениями (в сравнении с ARMA)

```
resvar = restrict(var, method = c("ser"), thresh = 0.6)
n1 = resvar$varresult$d1TATN$rank-1
if(n1<n+1) {n1=n+1}
ess = sum(resvar$varresult$d1TATN$residuals^2)

Fstat = ((rss-ess)/(n1-n))/(ess/(length(d1TATN)-2*n1-1))
Fstat
```

```
## [1] 4.512438
```

```
qf(0.99, df1=n1-n, df2=length(d1TATN)-2*n1-1)
```

```
## [1] 1.864688
```

```
pf(Fstat, n1-n, length(d1TATN)-n-n1-1, lower.tail=F)
```

```
## [1] 5.348773e-11
```

```
Hosking(resvar, lags=1.2*var$p)
```

```
##   lags statistic   df   p-value
##   9.6   34.52153 25.6 0.1118634
```

```
LiMcLeod(resvar, lags=1.2*var$p)
```

```
##   lags statistic   df   p-value
##   9.6   34.8303 25.6 0.1052958
```

F-критерий увеличился, при этом модель все еще лучше ARMA и не имеет авто- и кросс-корреляции остатков => превосходит модель без ограничений.

### 11.3.3. Сравнение модели с 3-мя и 4-мя переменными

```

#модель с 3-мя переменными
df = data.frame(d1TATN, d1SIBN, d1KMAZ)
var = VAR(df, p = 10, type = 'const')
resvar = restrict(var, method = c("ser"), thresh = 0.7)
n = resvar$varresult$d1TATN$rank-1
x = 10-8+1
rss = sum(resvar$varresult$d1TATN$residuals[x:length(resvar$varresult$d1TATN$residuals)]^2)

#модель с 4-мя переменными
df = data.frame(d1TATN, d1SIBN, d1KMAZ, d1NLMK)
var = VAR(df, p = 8, type = 'const')
resvar = restrict(var, method = c("ser"), thresh = 0.6)
n1 = resvar$varresult$d1TATN$rank-1
ess = sum(resvar$varresult$d1TATN$residuals^2)

Fstat = ((rss-ess) / (n1-n)) / (ess / (length(d1TATN)-2*n1-1))
Fstat

```

```
## [1] -1.447424
```

```
qf(0.99, df1=n1-n, df2=length(d1TATN)-2*n1-1)
```

```
## [1] 4.617544
```

```
pf(Fstat, n1-n, length(d1TATN)-2*n1-1, lower.tail=F)
```

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

Модель с 4-мя переменными хуже, чем модель с 3-мя переменными для объяснения d1TATN.

## 11.4. Итог:

Конечная модель - VAR model с 3-мя переменными с ограничениями.

```

df_var = data.frame(d1TATN, d1SIBN, d1KMAZ)
var = VAR(df_var, p = 10, type = 'const')
resvar = restrict(var, method = c("ser"), thresh = 0.7)

```

## 12. Структурные сдвиги на VAR модели

Нас интересуют конкретные разрывы в 2020 году, а именно 31 января 2020 года - появились первые зараженные коронавирусом в России и 11 августа 2020 - регистрация в России первой в мире вакции от COVID-19. Поэтому рассмотрим разрывы на промежутке всего 2020 года.

Для новых “кусков” данных нам понадобятся новые модели ARMA для проверки структурных сдвигов.  
 $\ln(250) = 5$

Мы проводим Sup-F тест для поиска структурных разрывов, а также тест Bai Perron - для множественных разрывов.

## 12.1. d1TATN

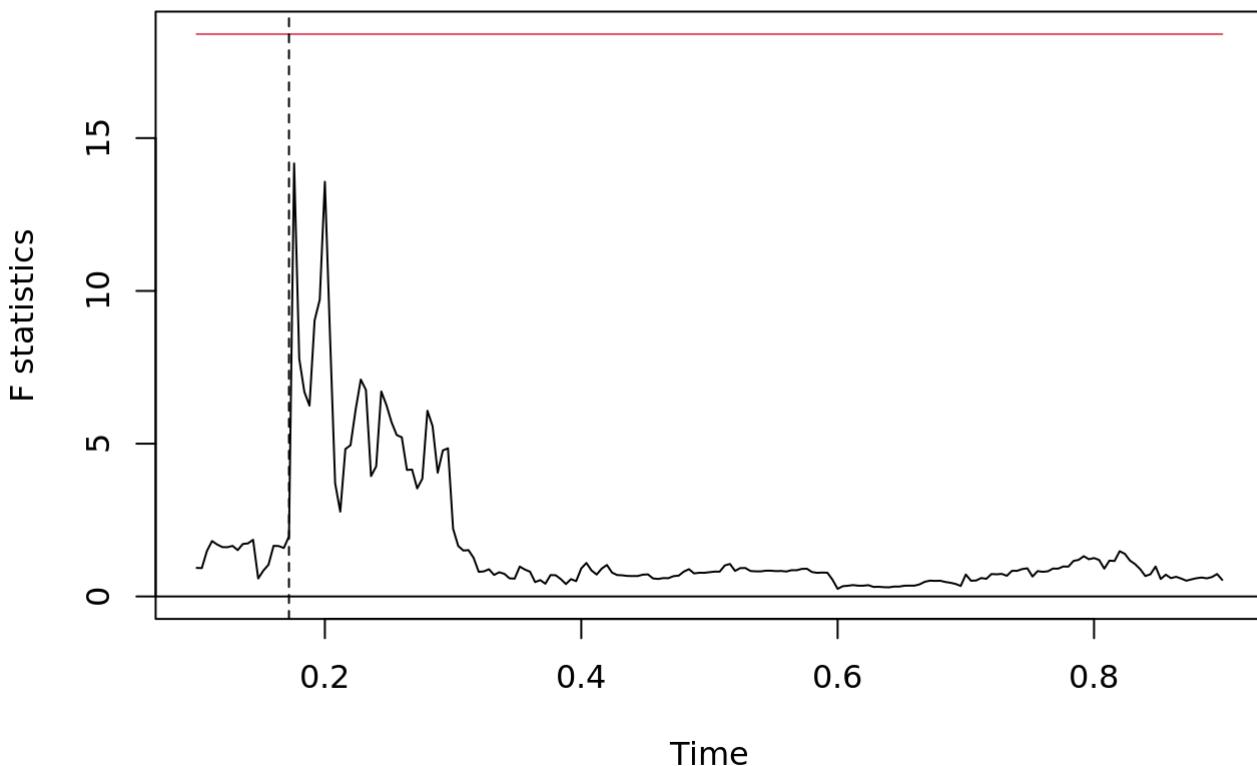
```
d1TATN_sdvig = df_var$d1TATN[1261:1510]
ARMAmodelTATN_sdvig = Arima(d1TATN_sdvig, c(2,0,1), include.constant=TRUE, method = c
("CSS-ML"))
Box.test(residuals(ARMAmodelTATN_sdvig), lag = 5, type = c("Ljung-Box"), fitdf = 3)
```

```
## 
## Box-Ljung test
##
## data: residuals(ARMAmodelTATN_sdvig)
## X-squared = 0.10832, df = 2, p-value = 0.9473
```

Автокорреляции нет - модель годная.

```
d1TATN_sdvig = d1TATN_sdvig[1:length(d1TATN_sdvig) ]
d1TATN_sdvig_11 = c(0,d1TATN_sdvig[1:length(d1TATN_sdvig)-1])
d1TATN_sdvig_12 = c(0,0,d1TATN_sdvig[2:length(d1TATN_sdvig)-2])

stat_TATN = Fstats(d1TATN_sdvig ~ d1TATN_sdvig_11 + d1TATN_sdvig_12, from = 0.1, to =
NULL)
plot(stat_TATN, alpha = 0.01)
lines(breakpoints(stat_TATN))
```



```
a_TATN = breakpoints(stat_TATN)
a_TATN$breakpoints
```

```
## [1] 44
```

```
sctest(stat_TATN, type = "supF")
```

```
##  
## supF test  
##  
## data: stat_TATN  
## sup.F = 14.172, p-value = 0.05631
```

```
data$date[1261+44]
```

```
## [1] "2020-03-10"
```

Структурный разрыв есть на 10% уровне - 10 марта 2020 года.

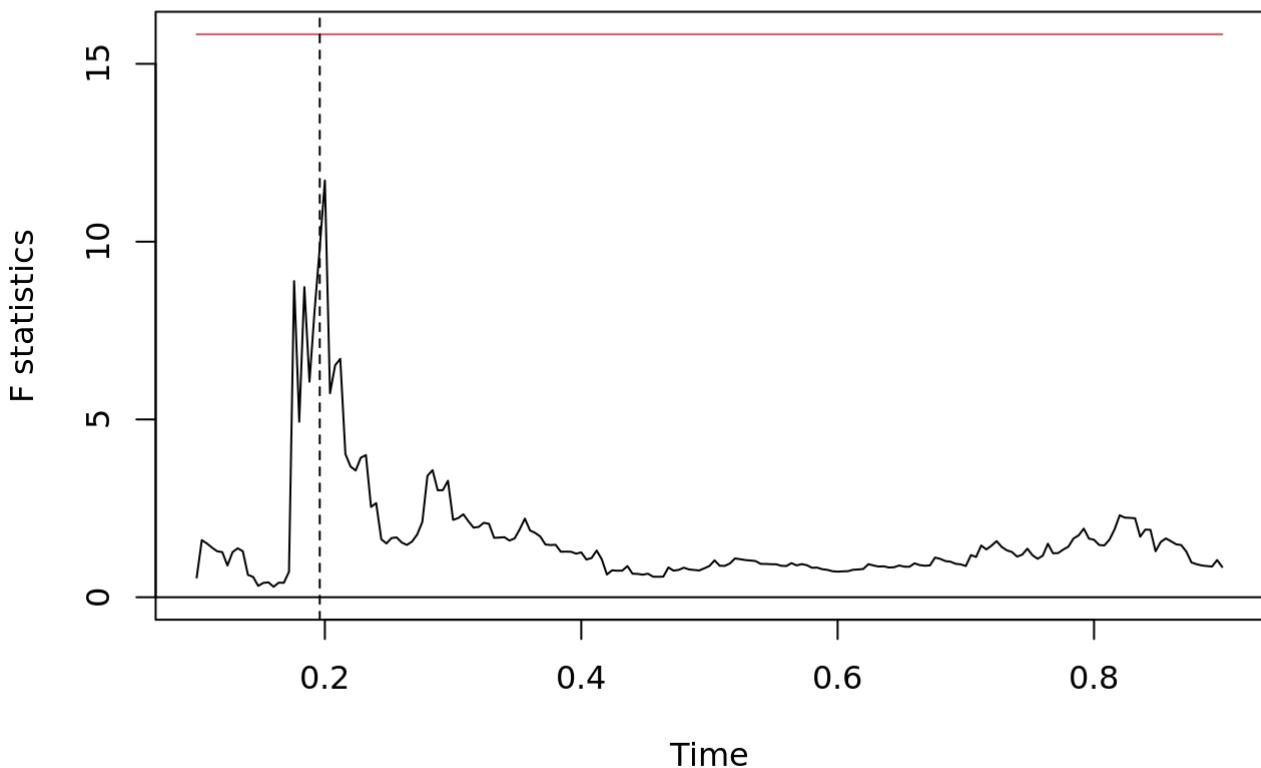
## 12.2. d1SIBN

```
d1SIBN_sdvig = df_var$d1SIBN[1261:1510]  
ARMAmodelSIBN_sdvig = Arima(d1SIBN_sdvig, c(1,0,1), include.constant=TRUE, method = c  
("CSS-ML"))  
Box.test(residuals(ARMAmodelSIBN_sdvig), lag = 5, type = c("Ljung-Box"), fitdf = 2)
```

```
##  
## Box-Ljung test  
##  
## data: residuals(ARMAmodelSIBN_sdvig)  
## X-squared = 2.3194, df = 3, p-value = 0.5088
```

Автокорреляции нет - модель годная.

```
d1SIBN_sdvig = d1SIBN_sdvig[1:length(d1SIBN_sdvig)]  
d1SIBN_sdvig_11 = c(0,d1SIBN_sdvig[1:length(d1SIBN_sdvig)-1])  
  
stat_SIBN = Fstats(d1SIBN_sdvig ~ d1SIBN_sdvig_11, from = 0.1, to = NULL)  
plot(stat_SIBN, alpha = 0.01)  
lines(breakpoints(stat_SIBN))
```



```
a_SIBN = breakpoints(stat_SIBN)
a_SIBN$breakpoints
```

```
## [1] 50
```

```
sctest(stat_SIBN, type = "supF")
```

```
##
## supF test
##
## data: stat_SIBN
## sup.F = 11.721, p-value = 0.05807
```

```
data$date[1261+50]
```

```
## [1] "2020-03-18"
```

Присутствует сдвиг на 10% уровне - 18 марта 2020 года.

## 12.3. d1KMAZ

```

d1KMAZ_sdvig = df_var$d1KMAZ[1261:1510]
ARMAmodelKMAZ_sdvig = Arima(d1KMAZ_sdvig, c(2,0,2), include.constant=TRUE, method = c
("CSS-ML"))
Box.test(residuals(ARMAmodelKMAZ_sdvig), lag = 5, type = c("Ljung-Box"), fitdf = 4)

```

```

## 
## Box-Ljung test
##
## data: residuals(ARMAmodelKMAZ_sdvig)
## X-squared = 2.6541, df = 1, p-value = 0.1033

```

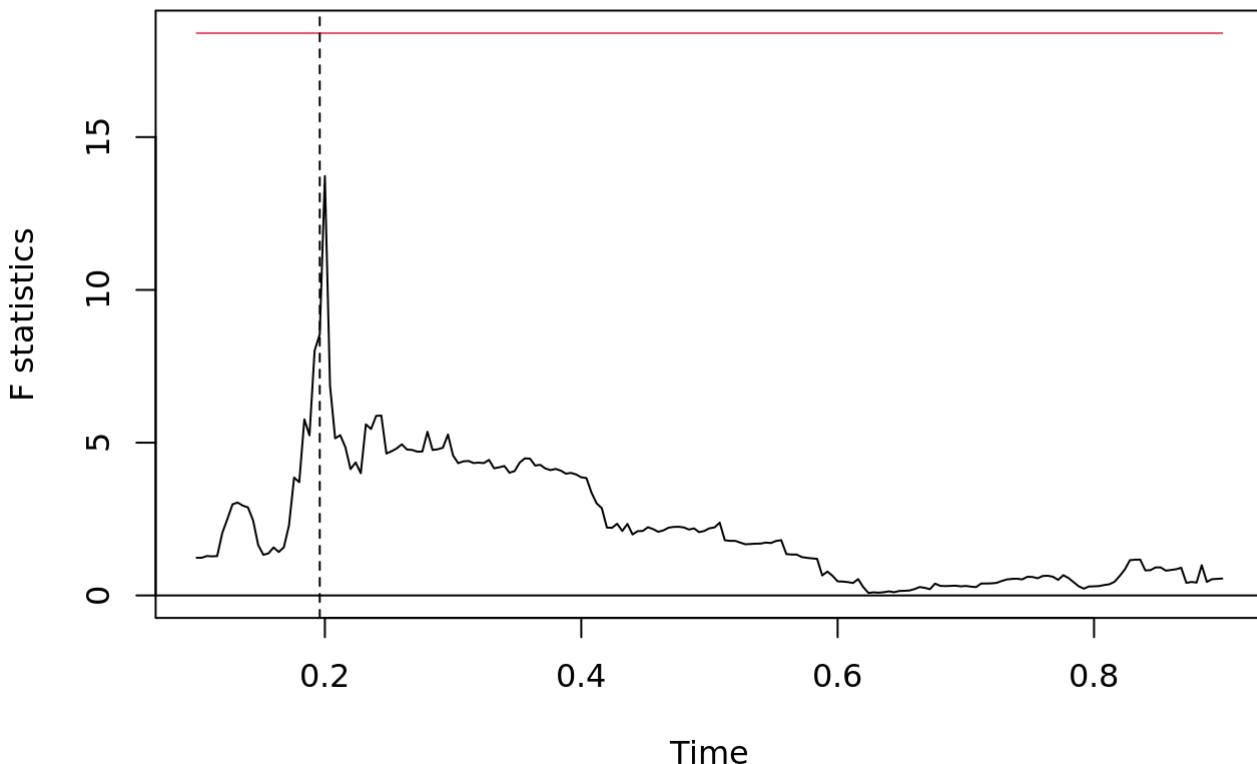
Автокорреляции нет - модель годная.

```

d1KMAZ_sdvig = d1KMAZ_sdvig[1:length(d1KMAZ_sdvig) ]
d1KMAZ_sdvig_11 = c(0,d1KMAZ_sdvig[1:length(d1KMAZ_sdvig)-1])
d1KMAZ_sdvig_12 = c(0,0,d1KMAZ_sdvig[2:length(d1KMAZ_sdvig)-2])

stat_KMAZ = Fstats(d1KMAZ_sdvig ~ d1KMAZ_sdvig_11 + d1KMAZ_sdvig_12, from = 0.1, to =
NULL)
plot(stat_KMAZ, alpha = 0.01)
lines(breakpoints(stat_KMAZ))

```



```

a_KMAZ = breakpoints(stat_KMAZ)
a_KMAZ$breakpoints

```

```

## [1] 50

```

```
sctest(stat_KMAZ, type = "supF")
```

```
##  
## supF test  
##  
## data: stat_KMAZ  
## sup.F = 13.726, p-value = 0.06689
```

```
data$date[1261+50]
```

```
## [1] "2020-03-18"
```

Присутствует сдвиг на 10% уровне - 18 марта 2020 года.

## 12.4. Вывод:

На 10% уровне значимости наблюдаются структурные разрывы в марте (10 и 18 числа). Тогда начали вступать в силу первые серьезные ограничения из-за COVID-19, а заболеваемость в стране стремительно поднималась вверх.

## 14. Прогнозы

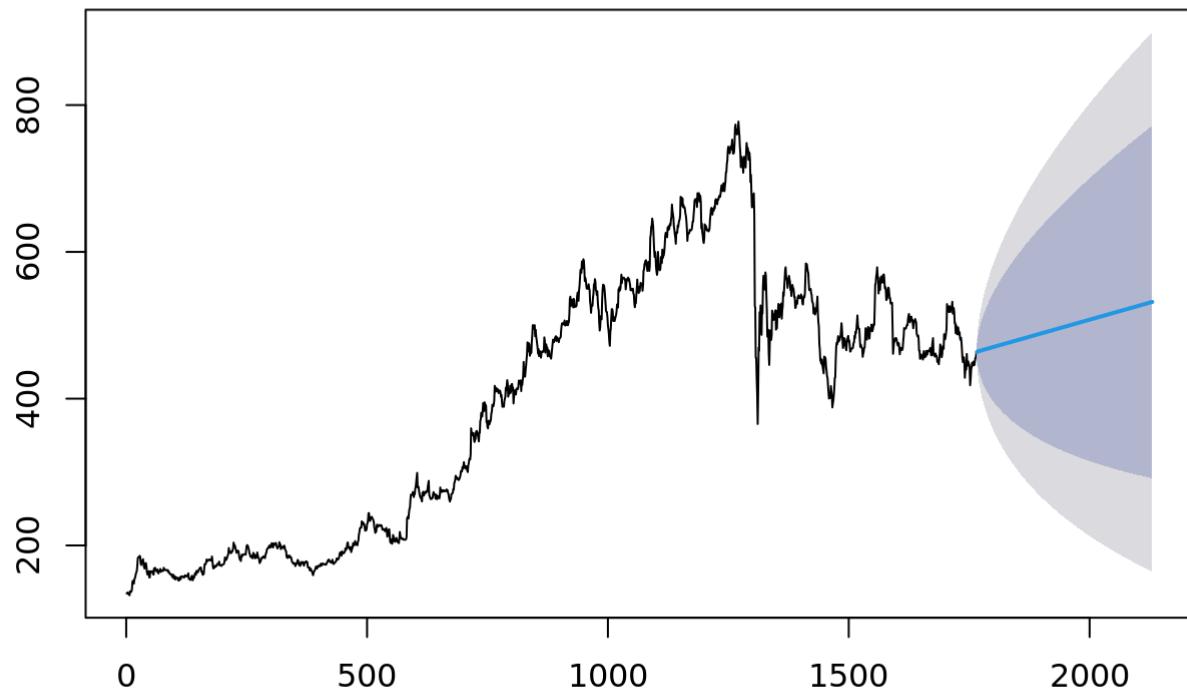
Для сравнения моделей мы построили несколько разных прогнозов на год:

### 14.1. Прогнозы на основе модели ARMA

Прогноз на основе модели ARMA дает представление о колебании будущих цен на акции. Мы можем видеть, что цены везде идут вверх, но сильно медленнее поднимаются у акций Сургутнефтегаза и Аэрофлота.

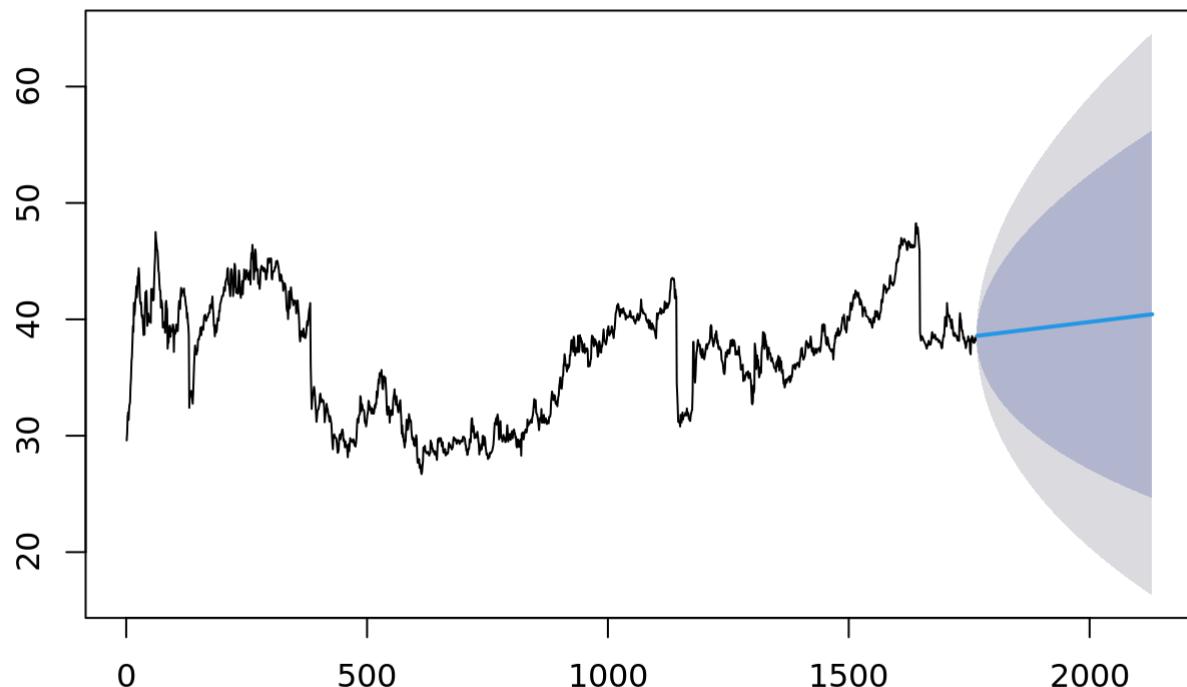
```
forecast_TATN = forecast(ARMAmodelTANT, h=365)  
plot(forecast_TATN)
```

## Forecasts from ARIMA(0,1,2) with drift



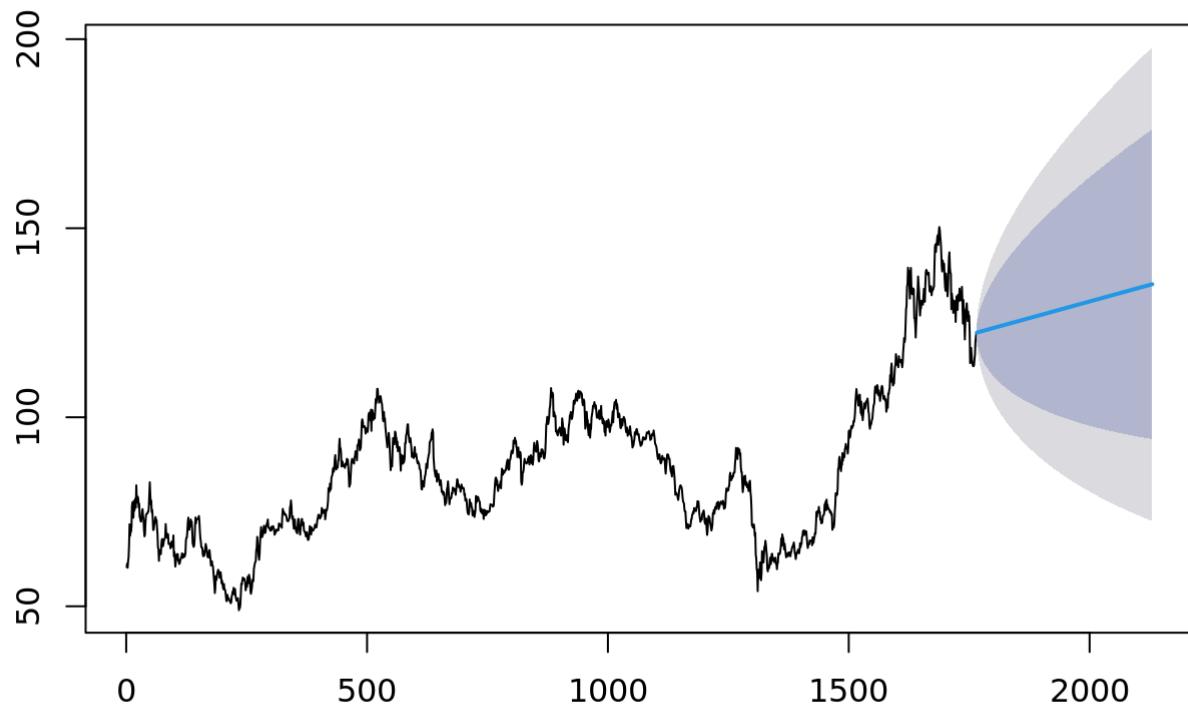
```
forecast_SNGS = forecast(ARMAmodelSNGS, h=365)
plot(forecast_SNGS)
```

## Forecasts from ARIMA(0,1,1) with drift



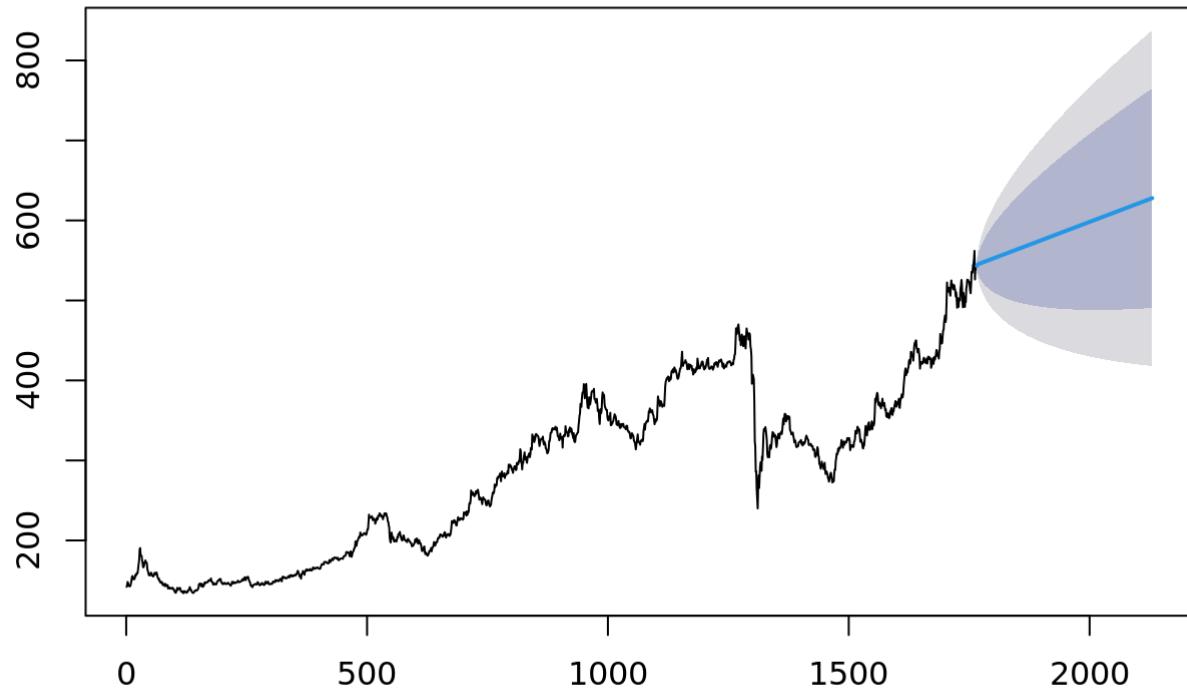
```
forecast_ALRS = forecast(ARMAmodelALRS, h=365)
plot(forecast_ALRS)
```

## Forecasts from ARIMA(0,1,1) with drift



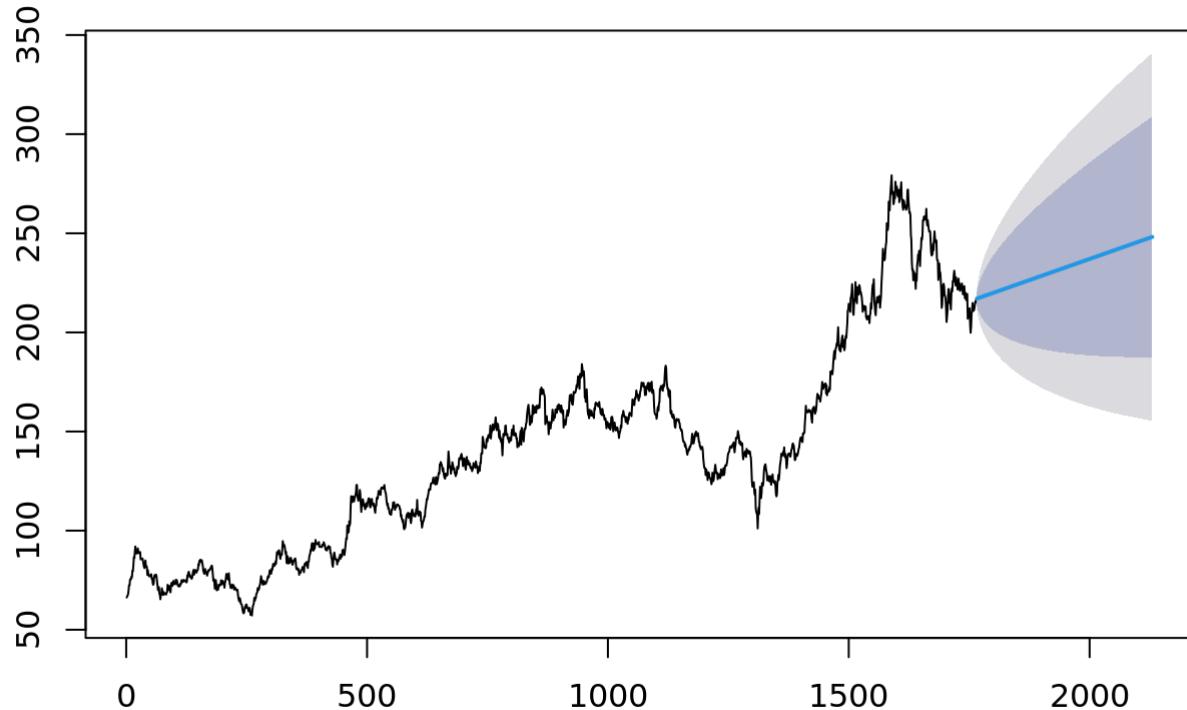
```
forecast_SIBN = forecast(ARMAmodelSIBN, h=365)
plot(forecast_SIBN)
```

## Forecasts from ARIMA(0,1,4) with drift



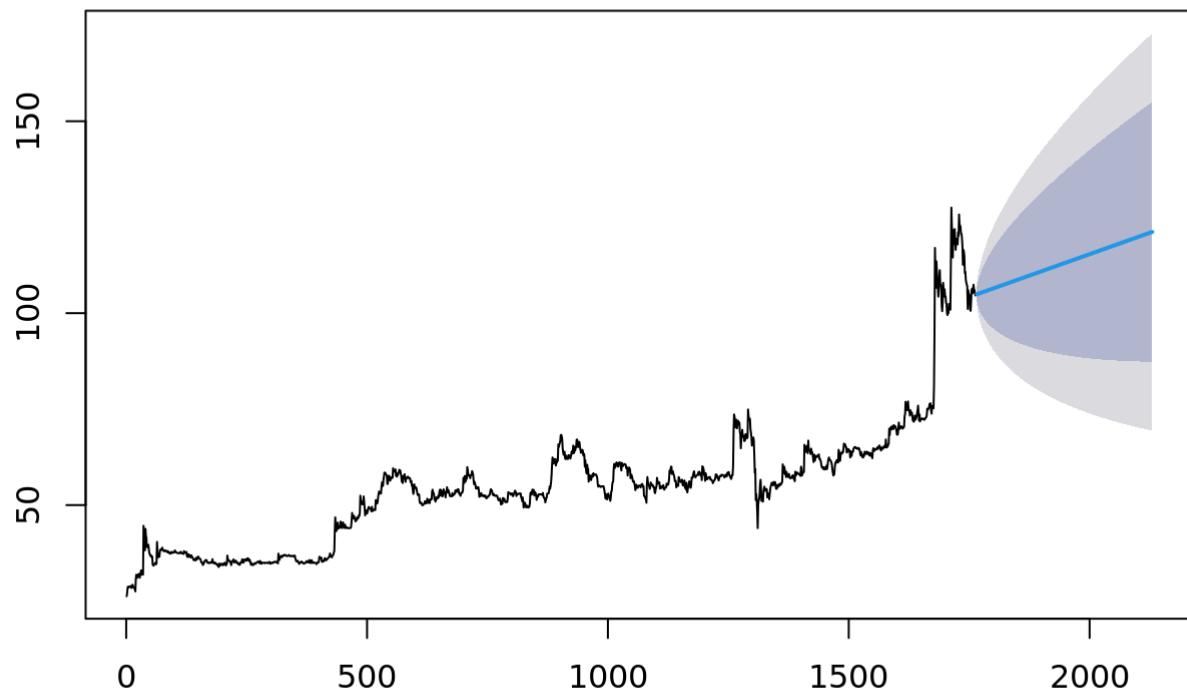
```
forecast_NLMK = forecast(ARMAmodelNLMK, h=365)
plot(forecast_NLMK)
```

## Forecasts from ARIMA(0,1,1) with drift



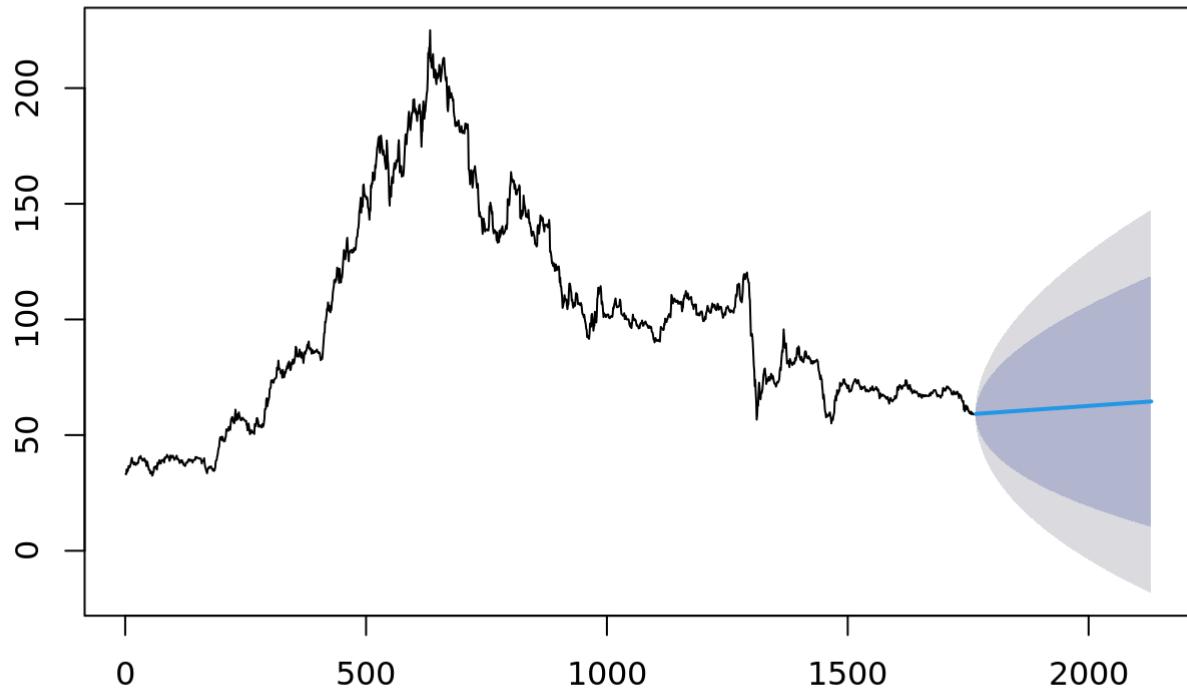
```
forecast_KMAZ = forecast(ARMAmodelKMAZ, h=365)
plot(forecast_KMAZ)
```

## Forecasts from ARIMA(0,1,4) with drift



```
forecast_AFLT = forecast(ARMAmodelAFLT, h=365)
plot(forecast_AFLT)
```

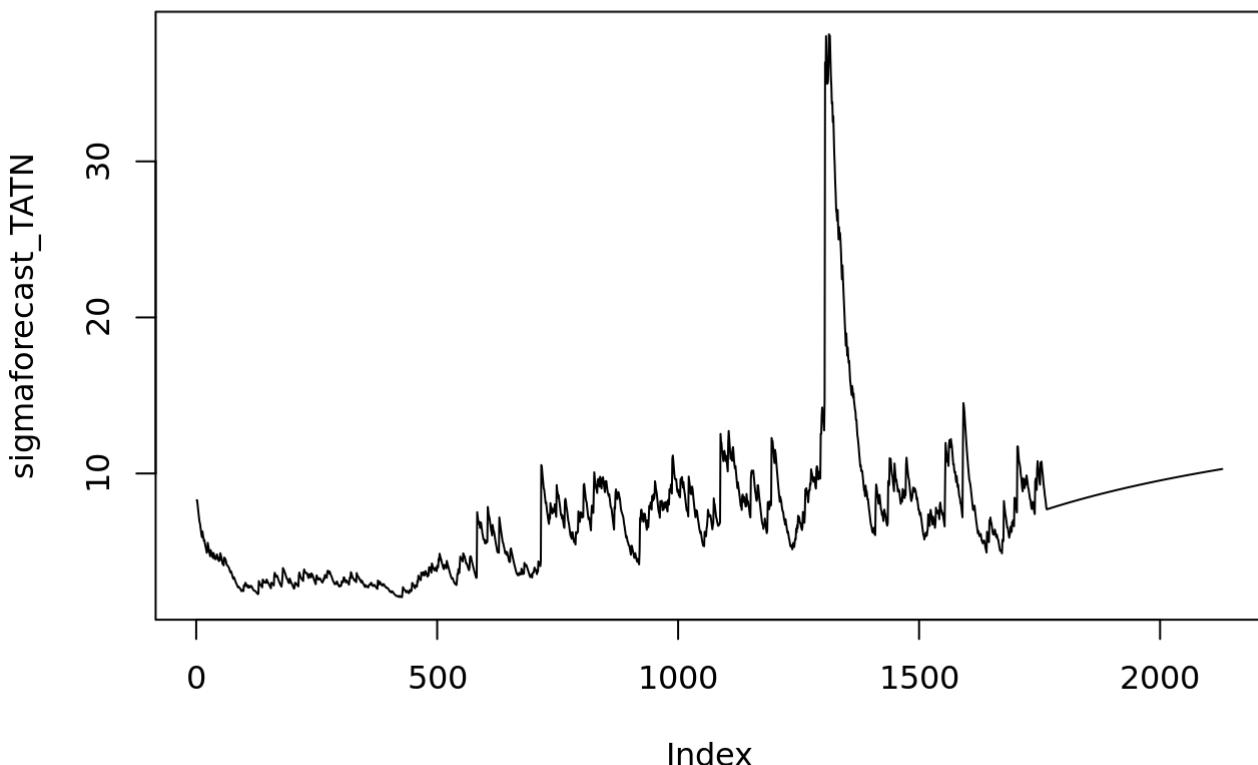
## Forecasts from ARIMA(0,1,1) with drift



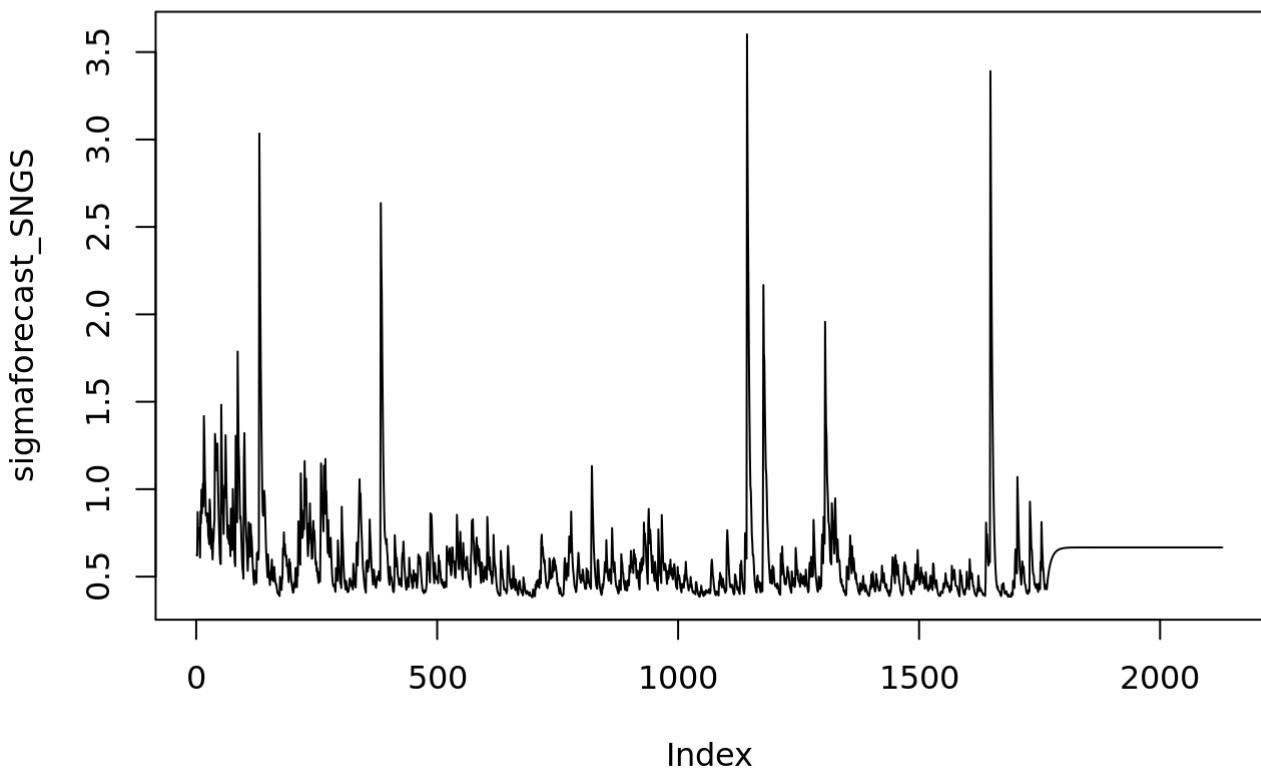
## 14.2. Прогнозы на основе модели simple GARCH

Что касается прогноза на основе simple GARCH, он показывает прогноз именно изменения цен на акции. В целом, можно сказать, что прогнозы двух моделей дополняют друг друга.

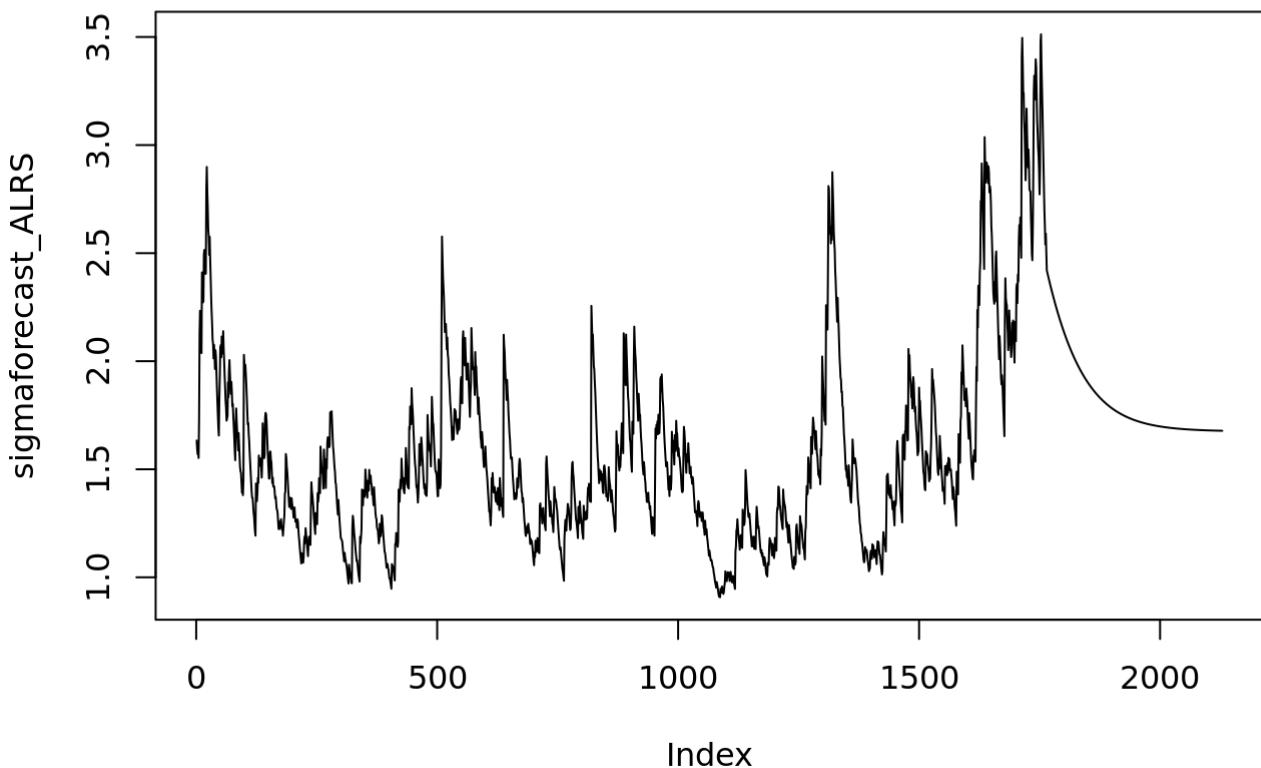
```
prognoz_TATN = ugarchforecast(garch.TATN, n.ahead = 365)
sigmaforecast_TATN = c(garch.TATN@fit$sigma, sigma(prognoz_TATN))
plot(sigmaforecast_TATN, type = 'l')
```



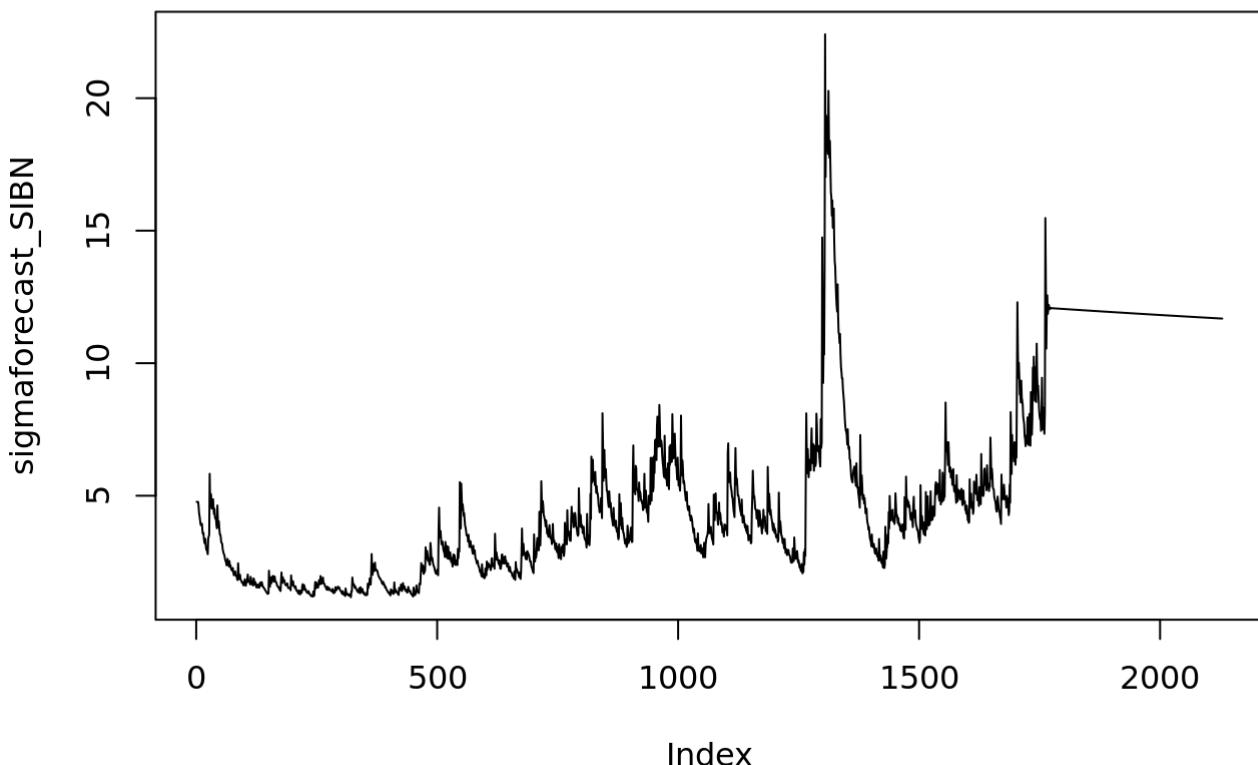
```
prognoz_SNGS = ugarchforecast(garch.SNGS, n.ahead = 365)
sigmaforecast_SNGS = c(garch.SNGS@fit$sigma, sigma(prognoz_SNGS))
plot(sigmaforecast_SNGS, type = 'l')
```



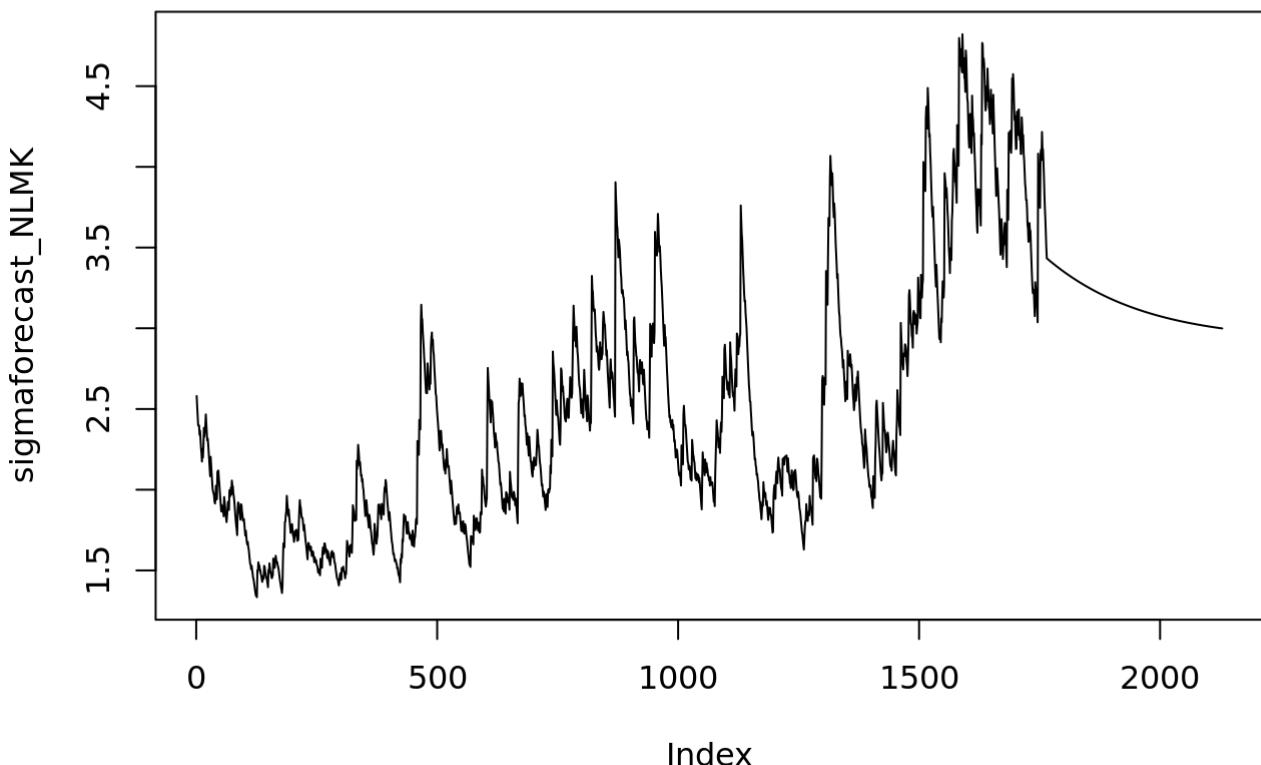
```
prognоз_ALRS = ugarchforecast(garch.ALRS, n.ahead = 365)
sigmaforecast_ALRS = c(garch.ALRS@fit$sigma, sigma(prognоз_ALRS))
plot(sigmaforecast_ALRS, type = 'l')
```



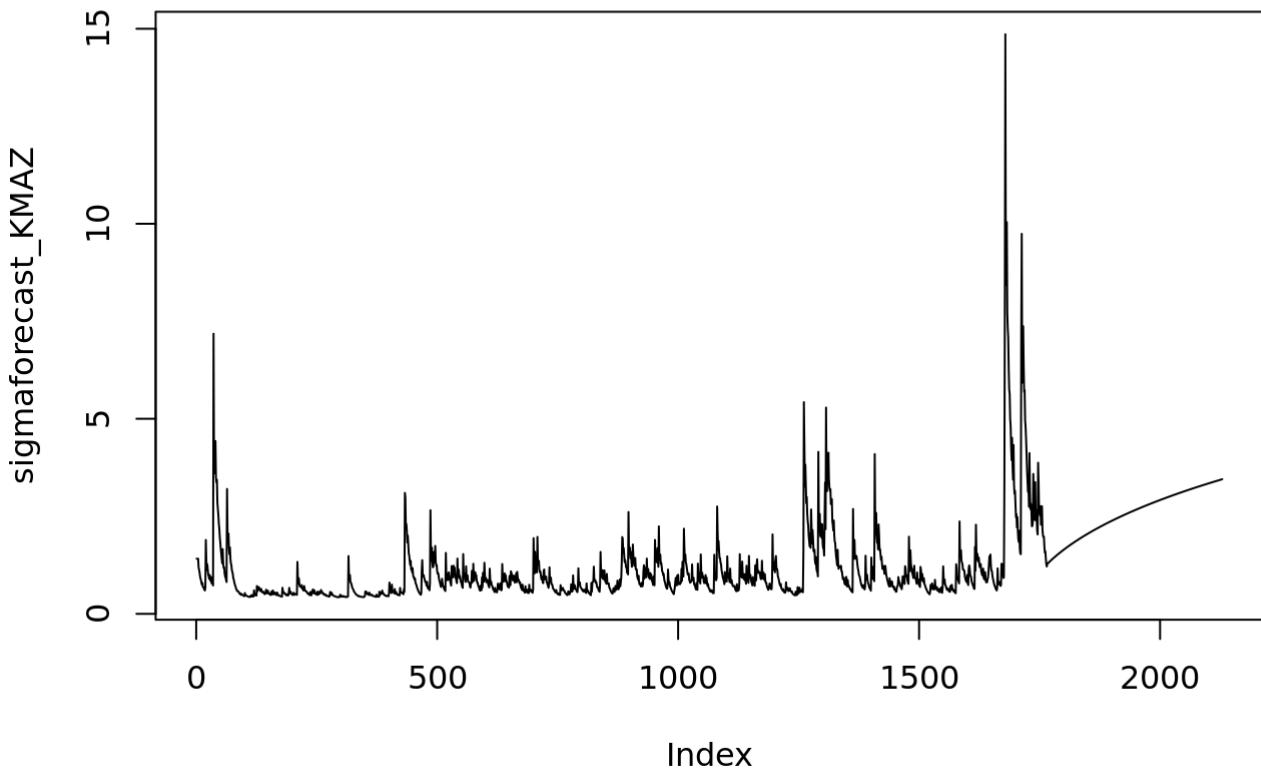
```
prognoz_SIBN = ugarchforecast(garch.SIBN, n.ahead = 365)
sigmaforecast_SIBN = c(garch.SIBN@fit$sigma, sigma(prognoz_SIBN))
plot(sigmaforecast_SIBN, type = 'l')
```



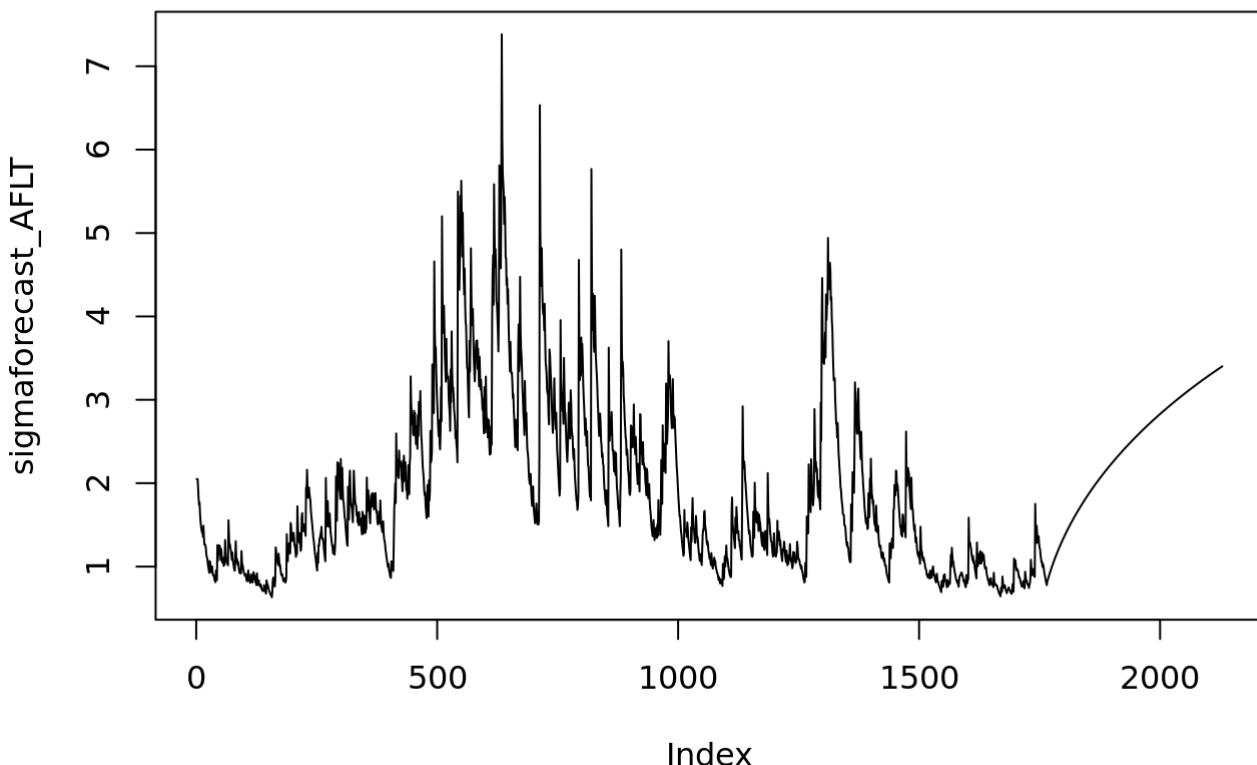
```
prognoz_NLMK = ugarchforecast(garch.NLMK, n.ahead = 365)
sigmaforecast_NLMK = c(garch.NLMK@fit$sigma, sigma(prognoz_NLMK))
plot(sigmaforecast_NLMK, type = 'l')
```



```
prognoz_KMAZ = ugarchforecast(garch.KMAZ, n.ahead = 365)
sigmaforecast_KMAZ = c(garch.KMAZ@fit$sigma, sigma(prognoz_KMAZ))
plot(sigmaforecast_KMAZ, type = 'l')
```



```
prognoz_AFLT = ugarchforecast(garch.AFLT, n.ahead = 365)
sigmaforecast_AFLT = c(garch.AFLT@fit$sigma, sigma(prognoz_AFLT))
plot(sigmaforecast_AFLT, type = 'l')
```



### 14.3. Прогноз VAR модели

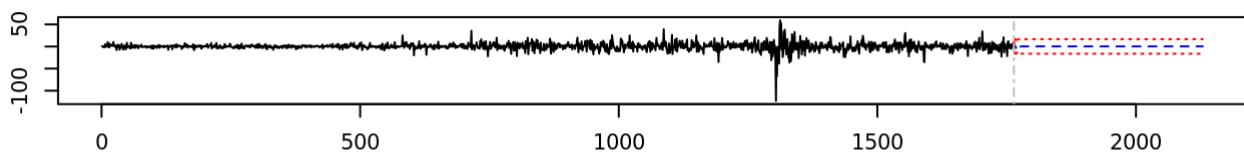
Прогноз по VAR модели показывает прогноз целевой переменной – цен акций компании Татнефть - на основе предыдущих данных цен акций Газпром нефти и Камаз - влияющих на нее переменных.

На остатках:

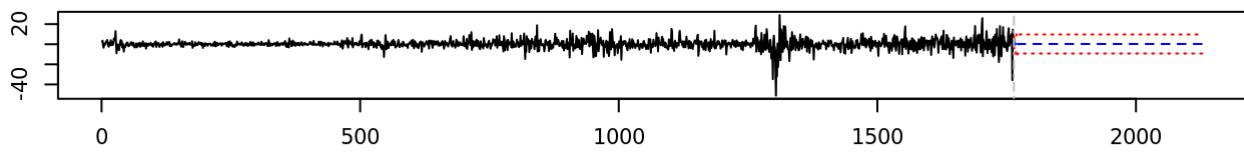
```
df_var = data.frame(d1TATN, d1SIBN, d1KMAZ)
#VARselect(df_var, lag.max = 20, type = "const")
VAR_final = VAR(df_var, p = 10, type = "const")

prognoz_var = predict(VAR_final, n.ahead = 365)
plot(prognoz_var, nc = 1)
```

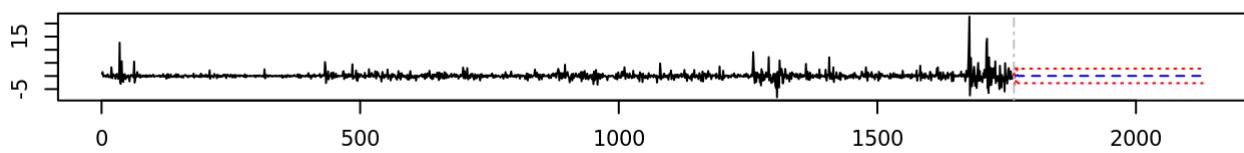
### Forecast of series d1TATN



### Forecast of series d1SIBN



### Forecast of series d1KMAZ

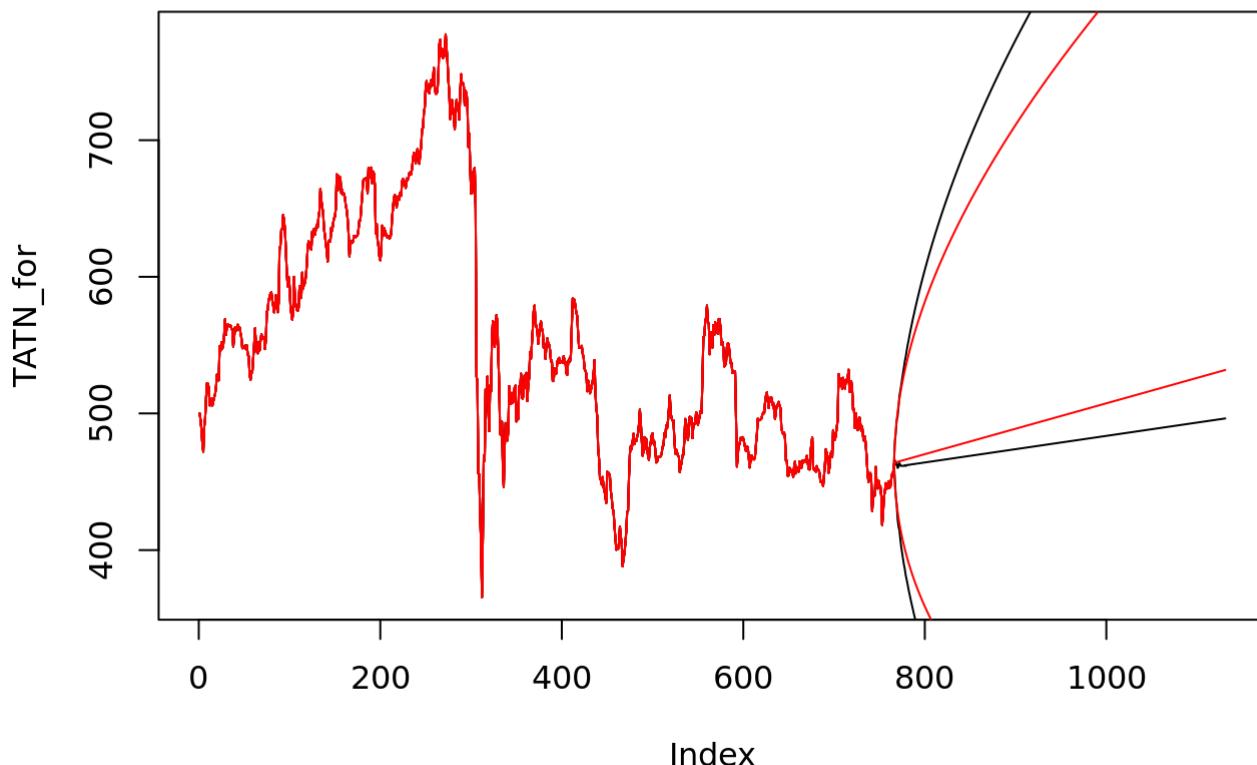


Прогноз основных данных:

```
Price_forARMA = c(data$TATN_p, forecast_TATN$mean)
Price_forARMA = Price_forARMA[1000:2130]
Price_upARMA = c(data$TATN_p, forecast_TATN$upper[,2])
Price_upARMA = Price_upARMA[1000:2130]
Price_lowARMA = c(data$TATN_p, forecast_TATN$lower[,2])
Price_lowARMA = Price_lowARMA[1000:2130]

plot(TATN_for, type = "l")
lines(TATN_up)
lines(TATN_low)

lines(Price_forARMA, col='red')
lines(Price_upARMA, col='red')
lines(Price_lowARMA, col='red')
```



Красная линия - прогноз по модели ARMA для цен на акции компании Татнефть.

Черная линия - прогноз на основе VAR(3) модели

На графике видно, что VAR прогноз немного ниже, чем ARMA, так как основано на трех переменных, а не одной, целевой.

## 13. Ожидаемая доходность портфеля

Ожидаемая доходность:  $R_i = R_f + \beta(R_m - R_f)$

- $R_f$  - безрисковая ставка доходности (доходность государственных облигаций - мы взяли годовые)
- $R_m$  - ожидаемая рыночная доходность (индекс Мосбиржи)
- $R_m - R_f$  - премия за инвестиционный риск
- $\beta$  - коэффициент чувствительности (источник - Тинькофф Инвестиции)

Бета-коэффициенты: TATN\_p = 1, SNGS\_p = 0.99, ALRS = 0.93, SIBN = 0.86, NLMK = 0.27, KMAZ = 0.81, AFLT = 1.12

Мы считаем реальную доходность акций через процент изменения от дня ко дню, а доходность по модели CAPM – по формуле через линейную регрессию.

```

tatn = read_excel("stocks.xlsx", sheet = "TATN_p") %>% dplyr::select("Дата", "Изм. %")
) %>%
  rename("date" = "Дата", "TATN_return" = "Изм. %")
sngs = read_excel("stocks.xlsx", sheet = "SNGS_p") %>% dplyr::select("Дата", "Изм. %")
) %>%
  rename("date" = "Дата", "SNGS_return" = "Изм. %")
alrs = read_excel("stocks.xlsx", sheet = "ALRS") %>% dplyr::select("Дата", "Изм. %")
%>%
  rename("date" = "Дата", "ALRS_return" = "Изм. %")
sibn = read_excel("stocks.xlsx", sheet = "SIBN") %>% dplyr::select("Дата", "Изм. %")
%>%
  rename("date" = "Дата", "SIBN_return" = "Изм. %")
nlmk = read_excel("stocks.xlsx", sheet = "NLMK") %>% dplyr::select("Дата", "Изм. %")
%>%
  rename("date" = "Дата", "NLMK_return" = "Изм. %")
kmaz = read_excel("stocks.xlsx", sheet = "KMAZ") %>% dplyr::select("Дата", "Изм. %")
%>%
  rename("date" = "Дата", "KMAZ_return" = "Изм. %")
aflt = read_excel("stocks.xlsx", sheet = "AFLT") %>% dplyr::select("Дата", "Изм. %")
%>%
  rename("date" = "Дата", "AFLT_return" = "Изм. %")

imoex = read_excel("IMOEX.xlsx") %>% dplyr::select("Дата", "Изм. %") %>%
  rename("date" = "Дата", "R_m" = "Изм. %")
imoex$date = ymd(imoex$date)
imoex$R_m = as.numeric(imoex$R_m)
bonds = read_excel("annual_bonds.xlsx") %>% dplyr::select("Дата", "Цена") %>%
  rename("date" = "Дата", "R_f" = "Цена")
bonds$date = ymd(bonds$date)

capm = data %>% dplyr::select(date) %>% left_join(tatn, by = "date") %>% left_join(sngs, by = "date") %>%
  left_join(alrs, by = "date") %>% left_join(sibn, by = "date") %>% left_join(nlmk, by = "date") %>%
  left_join(kmaz, by = "date") %>% left_join(aflt, by = "date") %>%
  left_join(bonds, by = "date") %>% left_join(imoex, by = "date")

capm$TATN_return = capm$TATN_return/100
capm$SNGS_return = capm$SNGS_return/100
capm$ALRS_return = capm$ALRS_return/100
capm$SIBN_return = capm$SIBN_return/100
capm$NLMK_return = capm$NLMK_return/100
capm$KMAZ_return = capm$KMAZ_return/100
capm$AFLT_return = capm$AFLT_return/100
capm$R_m = capm$R_m/100
capm$R_f = capm$R_f/100

capm = capm %>% mutate(risk_premium = R_m - R_f)

capm = capm %>% mutate(TATN_rp = 1 * risk_premium, SNGS_rp = 0.99 * risk_premium,
                        ALRS_rp = 0.93 * risk_premium, SIBN_rp = 0.86 * risk_premium,
                        NLMK_rp = 0.27 * risk_premium, KMAZ_rp = 0.81 * risk_premium,
                        AFLT_rp = 1.12 * risk_premium)

# считаем ожидаемую доходность
TATN fit = lm(TATN return ~ R_f + TATN rp, data = capm)

```

```

SNGS_fit = lm(SNGS_return ~ R_f + SNGS_rp, data = capm)
ALRS_fit = lm(ALRS_return ~ R_f + ALRS_rp, data = capm)
SIBN_fit = lm(SIBN_return ~ R_f + SIBN_rp, data = capm)
NLMK_fit = lm(NLMK_return ~ R_f + NLMK_rp, data = capm)
KMAZ_fit = lm(KMAZ_return ~ R_f + KMAZ_rp, data = capm)
AFLT_fit = lm(AFLT_return ~ R_f + AFLT_rp, data = capm)

TATN_predict = predict.lm(TATN_fit)
SNGS_predict = predict.lm(SNGS_fit)
ALRS_predict = predict.lm(ALRS_fit)
SIBN_predict = predict.lm(SIBN_fit)
NLMK_predict = predict.lm(NLMK_fit)
KMAZ_predict = predict.lm(KMAZ_fit)
AFLT_predict = predict.lm(AFLT_fit)

capm$TATN_predict = TATN_predict
capm$SNGS_predict = SNGS_predict
capm$ALRS_predict = ALRS_predict
capm$SIBN_predict = SIBN_predict
capm$NLMK_predict = NLMK_predict
capm$KMAZ_predict = KMAZ_predict
capm$AFLT_predict = AFLT_predict

```

## 13.1. Графики доходности

Красная линия - реальная доходность Синяя линия - доходность по модели CAPM

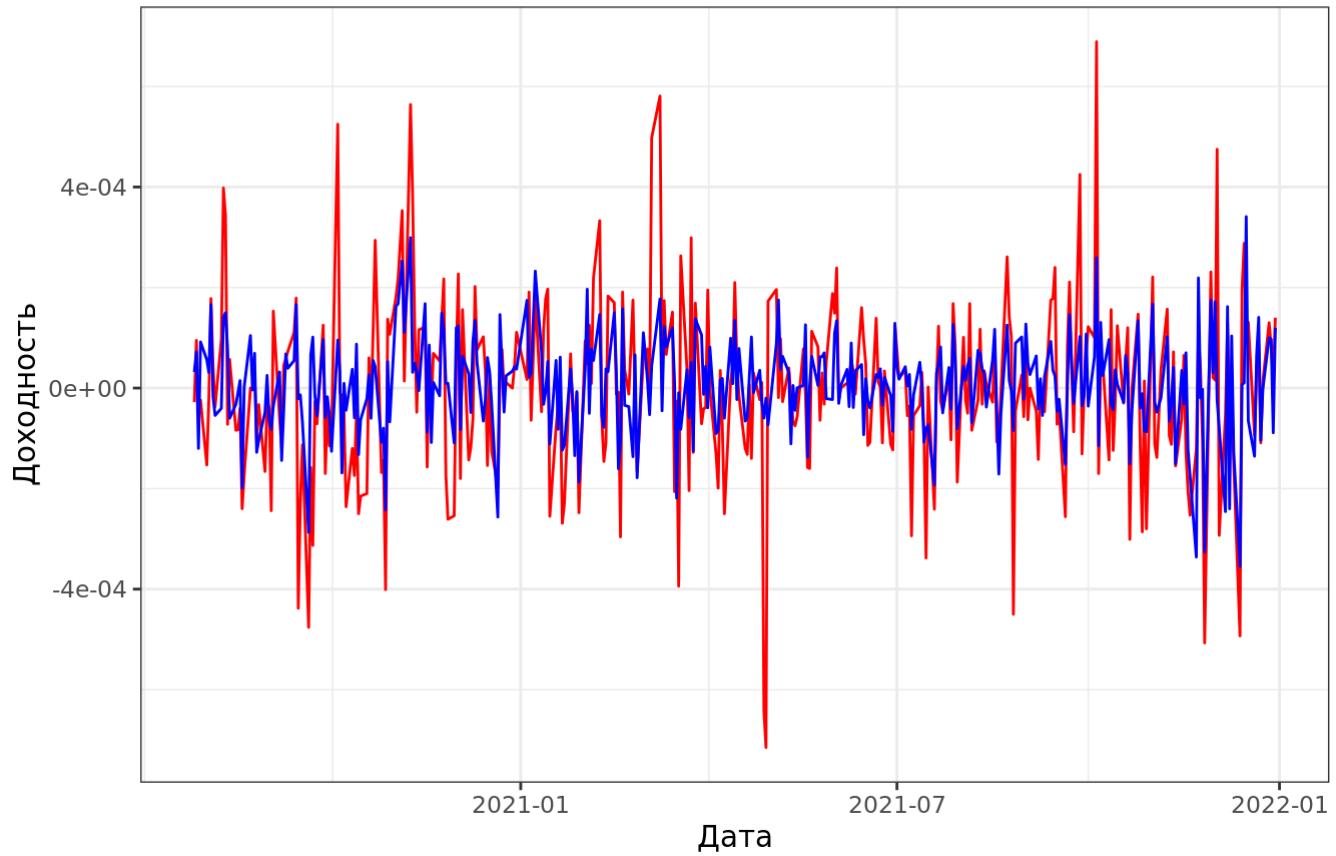
```

date_d = c(1:1765)
capm = capm %>% mutate(date_d = date_d) %>% filter(date_d > 1399)
capm$date = ymd(capm$date)

ggplot(capm) +
  geom_line(aes(x = date, y = TATN_return), color = "red") +
  geom_line(aes(x = date, y = TATN_predict), color = "blue") +
  ggtitle('График доходности по модели CAPM и реальной доходности \n Татнефть') +
  xlab('Дата') +
  ylab('Доходность') +
  theme_bw()

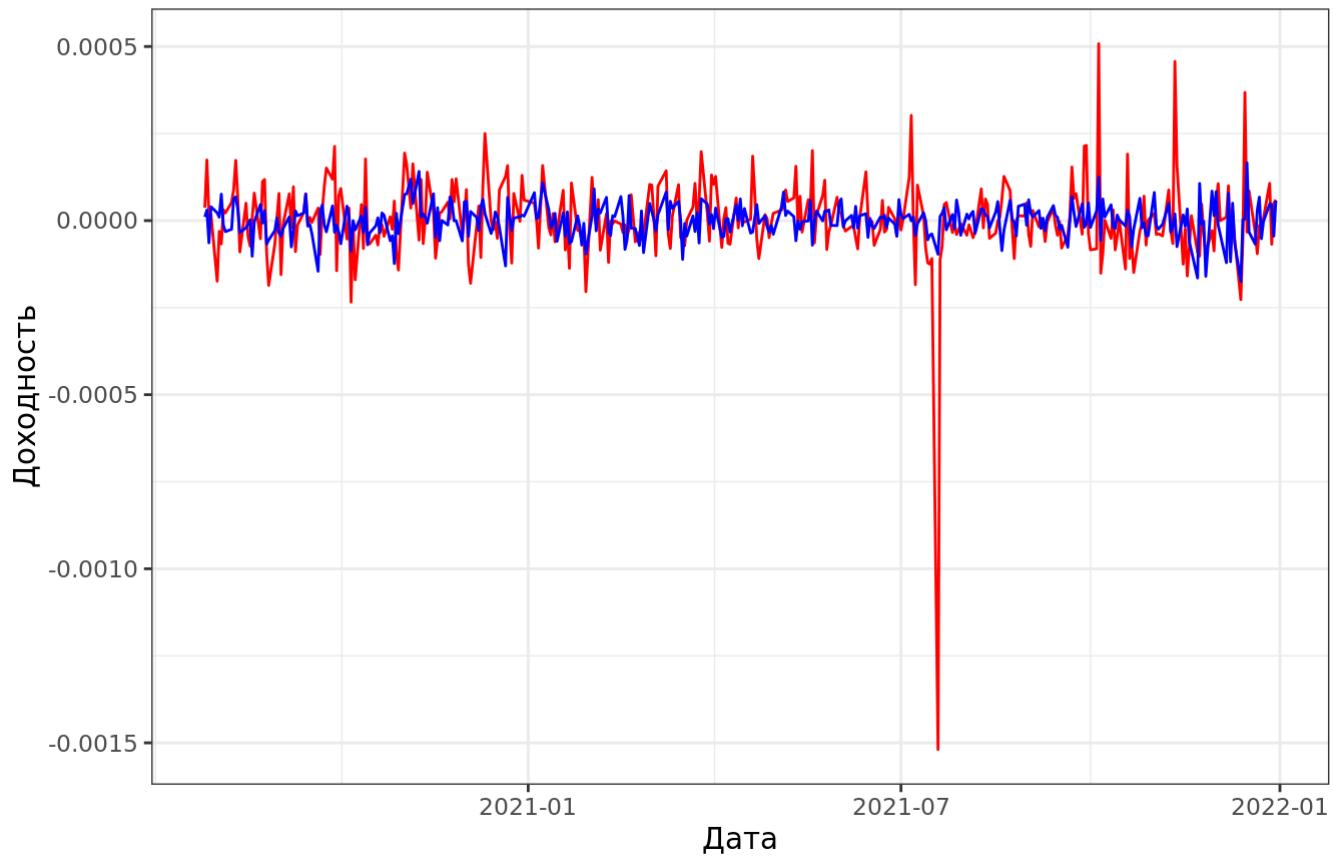
```

## График доходности по модели CAPM и реальной доходности Татнефть



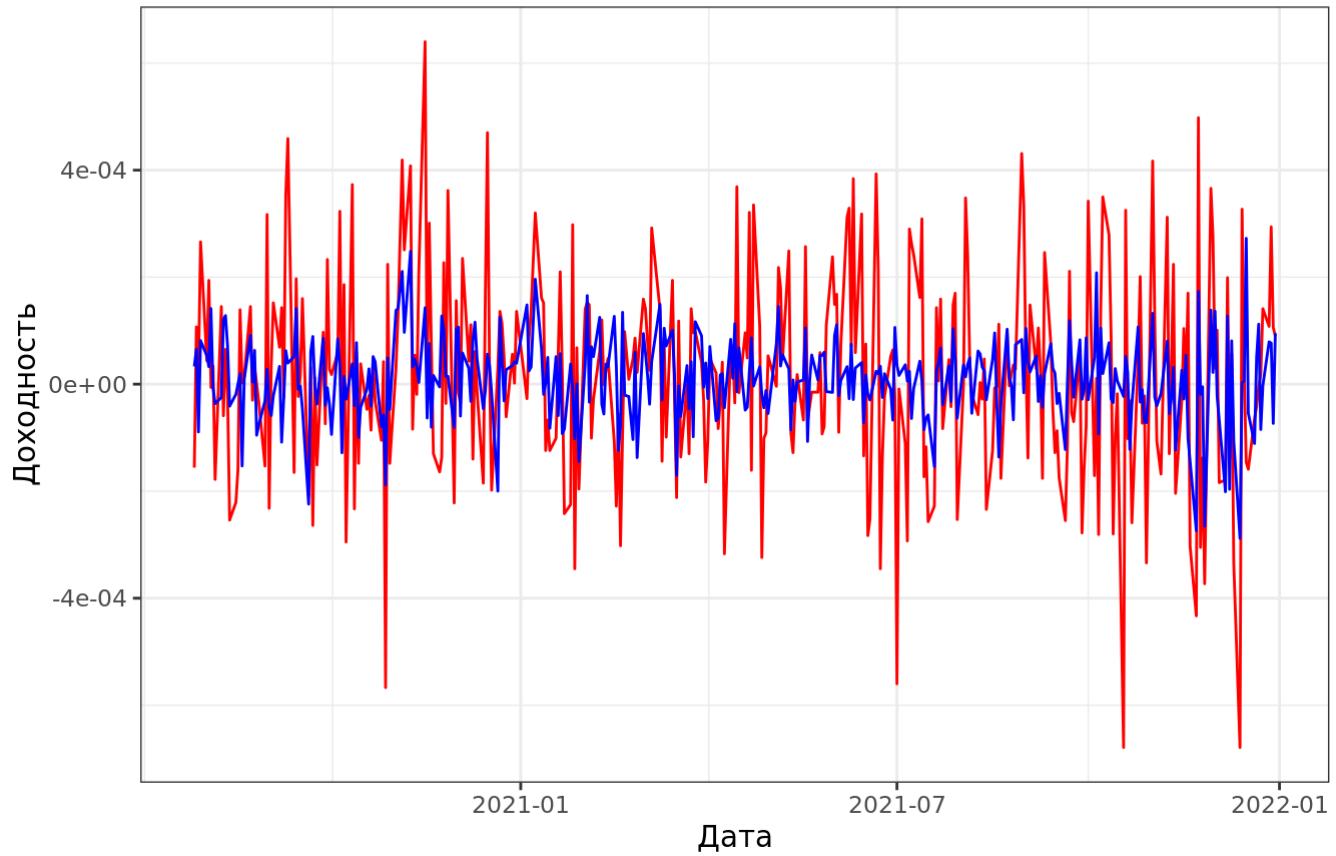
```
ggplot(capm) +  
  geom_line(aes(x = date, y = SNGS_return), color = "red") +  
  geom_line(aes(x = date, y = SNGS_predict), color = "blue") +  
  ggtitle('График доходности по модели CAPM и реальной доходности \n Сургутнефтегаз')  
+  
  xlab('Дата') +  
  ylab('Доходность') +  
  theme_bw()
```

## График доходности по модели CAPM и реальной доходности Сургутнефтегаз



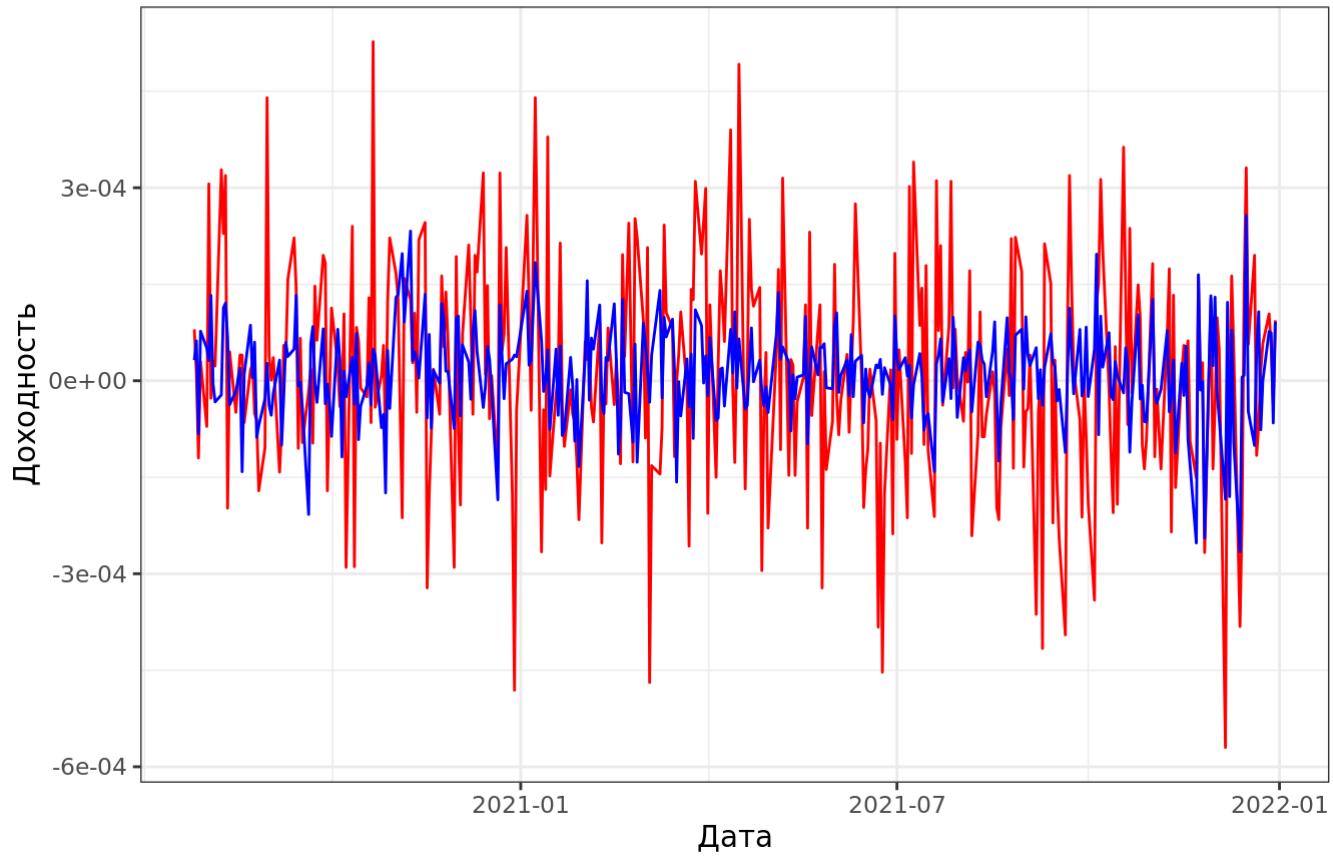
```
ggplot(capm) +  
  geom_line(aes(x = date, y = ALRS_return), color = "red") +  
  geom_line(aes(x = date, y = ALRS_predict), color = "blue") +  
  ggtitle('График доходности по модели CAPM и реальной доходности \n Алроса') +  
  xlab('Дата') +  
  ylab('Доходность') +  
  theme_bw()
```

## График доходности по модели CAPM и реальной доходности Алроса



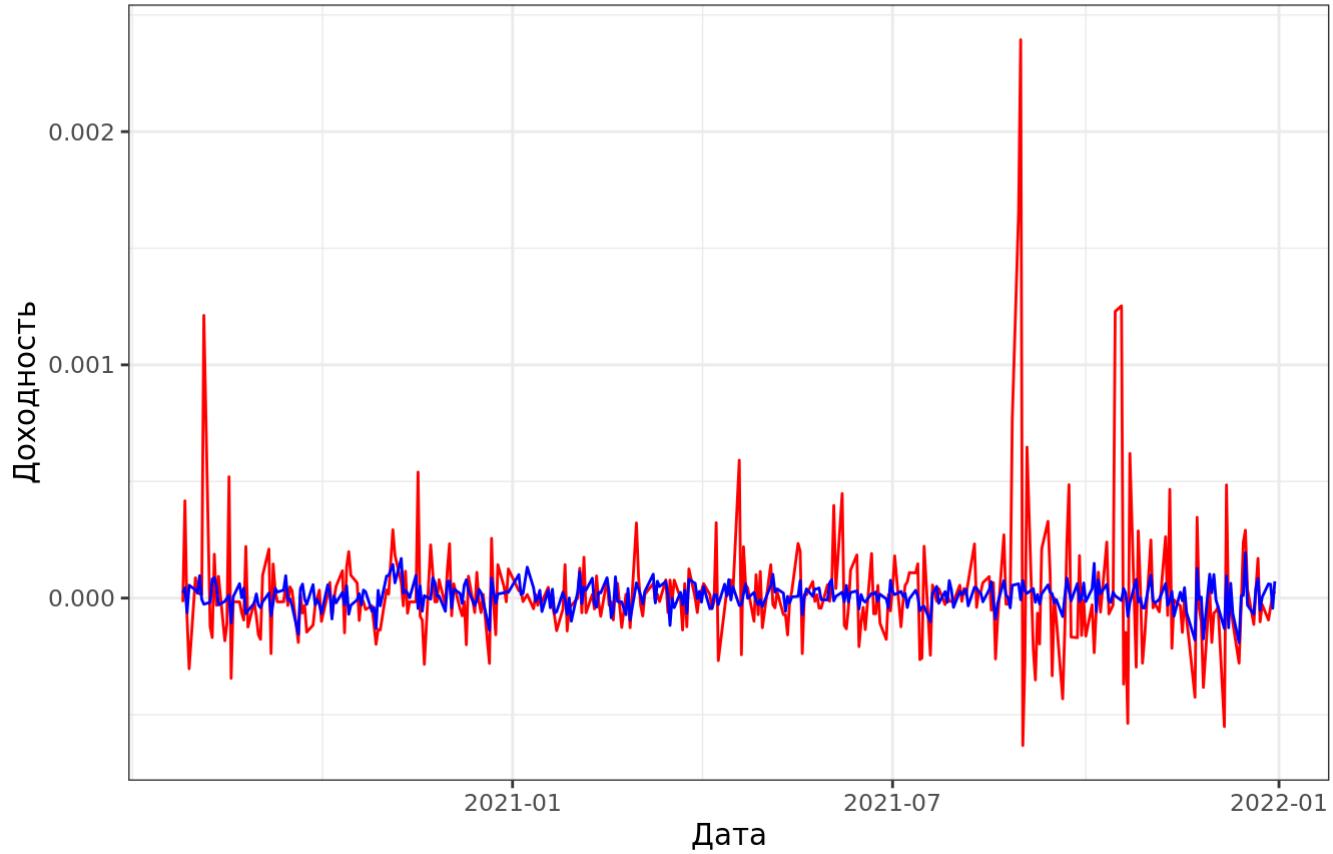
```
ggplot(capm) +  
  geom_line(aes(x = date, y = NLMK_return), color = "red") +  
  geom_line(aes(x = date, y = NLMK_predict), color = "blue") +  
  ggtitle('График доходности по модели CAPM и реальной доходности \n Газпром нефть')  
+  
  xlab('Дата') +  
  ylab('Доходность') +  
  theme_bw()
```

## График доходности по модели CAPM и реальной доходности Газпром нефть



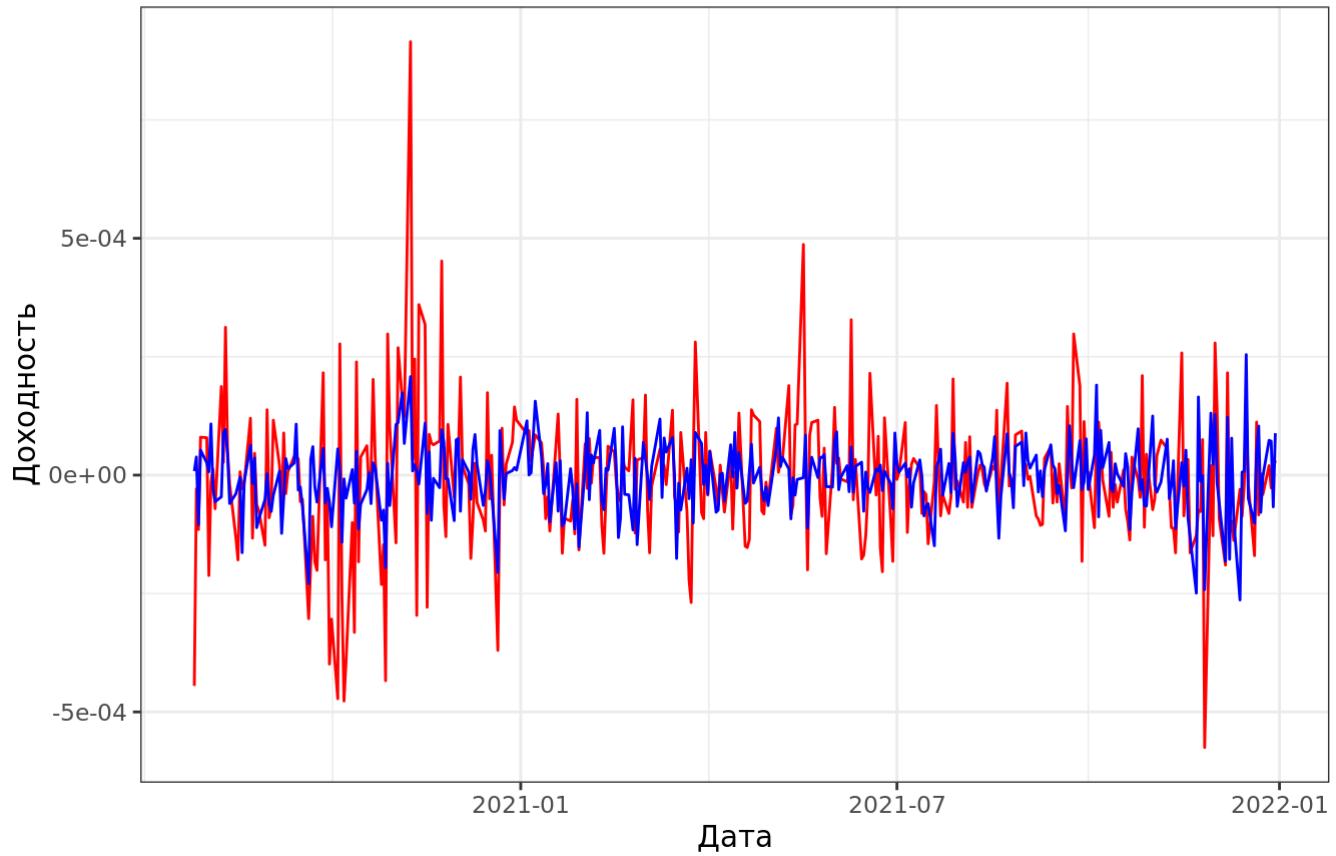
```
ggplot(capm) +  
  geom_line(aes(x = date, y = KMAZ_return), color = "red") +  
  geom_line(aes(x = date, y = KMAZ_predict), color = "blue") +  
  ggtitle('График доходности по модели CAPM и реальной доходности \n Новолипецкий мега  
аллургический завод') +  
  xlab('Дата') +  
  ylab('Доходность') +  
  theme_bw()
```

## График доходности по модели CAPM и реальной доходности Новолипецкий металлургический завод



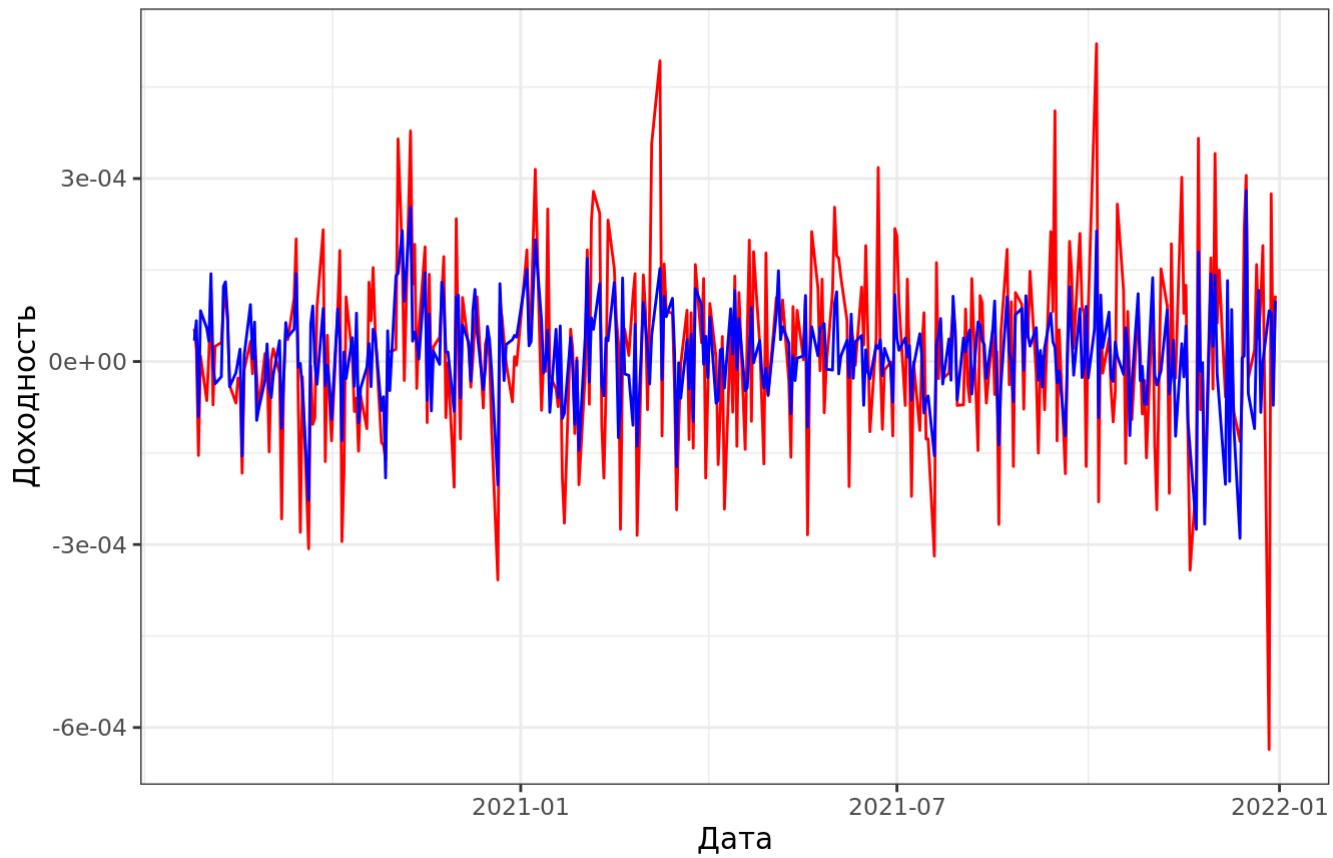
```
ggplot(capm) +  
  geom_line(aes(x = date, y = AFLT_return), color = "red") +  
  geom_line(aes(x = date, y = AFLT_predict), color = "blue") +  
  ggtitle('График доходности по модели CAPM и реальной доходности \n Камаз') +  
  xlab('Дата') +  
  ylab('Доходность') +  
  theme_bw()
```

## График доходности по модели CAPM и реальной доходности Камаз



```
ggplot(capm) +  
  geom_line(aes(x = date, y = SIBN_return), color = "red") +  
  geom_line(aes(x = date, y = SIBN_predict), color = "blue") +  
  ggtitle('График доходности по модели CAPM и реальной доходности \n Аэрофлот') +  
  xlab('Дата') +  
  ylab('Доходность') +  
  theme_bw()
```

## График доходности по модели CAPM и реальной доходности Аэрофлот



Таким образом, мы построили графики сравнения реальной доходности и доходности, прогнозируемой моделью. В целом, можно увидеть, что модель довольно четко описывает реальную доходность, имея лишь меньше выбросов.