

TSE_exam

Kovalev Evgeny

2016, December 24

1 Предобработка данных и графики рядов

Заметим, что дубликатов нет совсем, а пропусков в коротких рядах нет, и есть только там, где имеются значения длинных рядов.

Сначала я добавил в исходную таблицу строки для пропущенных наблюдений. Затем я заполнил года и кварталы в данных строчках в соответствии с порядком. Потом я начал добавлять соответствующие значения целевых показателей.

1. Денежные средства и денежные эквиваленты

Видно, что почти нигде не бывает поквартальных отчетов. Поэтому для каждого года заполним пропуски одними и теми же значениями, соответствующими этому году. Исключения: 2004 год (нет никакой информации, заполняем средним по 2003/2005 годам), 2009 год (там происходит что-то странное, похожее на выброс. От греха подальше заполним этот год значением за 4 квартал).

2. Заемные средства (долгосрочные/краткосрочные)

То же самое, 2004 год заполняем средним по 2003/2005, в случае 2000 года, где отличаются значения за 1-2 и 4 квартал, заполним значение за 3 квартал средним.

3. Выручка/валовая прибыль/чистая прибыль

Заменять средним по такому же кварталу прошлого/следующего года - идея неплохая, но пропусков для этого слишком много. Поэтому было решено делать это при малейшей возможности, а перед этим заполнять пропуски значениями прямой, соединяющей два ближайших квартала без пропусков. Например, пусть B - значение признака за 4 квартал одного года, A - значение признака за 4 квартал прошлого года, и за 1-2-3 кварталы данного года значений нет. Тогда эти пропуски заполнятся значениями $(B + 3A)/4$, $(B + A)/2$, $(3B + A)/4$, соответственно.

После всего заполнения пропусков в выручке, валовой прибыли и чистой прибыли каждое значение за 4 квартал было заменено на (значение за 4 квартал - значение за 3 квартал), после этого значение за 3 квартал на (значение за 3 квартал - значение за 2 квартал), после этого значение за 2 квартал на (значение за 2 квартал - значение за 1 квартал).

В ходе подобной обработки данных возникла новая таблица Evroplan_new.xlsx, уже без пропусков и с переходом к поквартальным значениям - с ней и будем работать.

```
library(forecast)
library(readxl)
library(plyr)
library(tseries)
library(MASS)
```

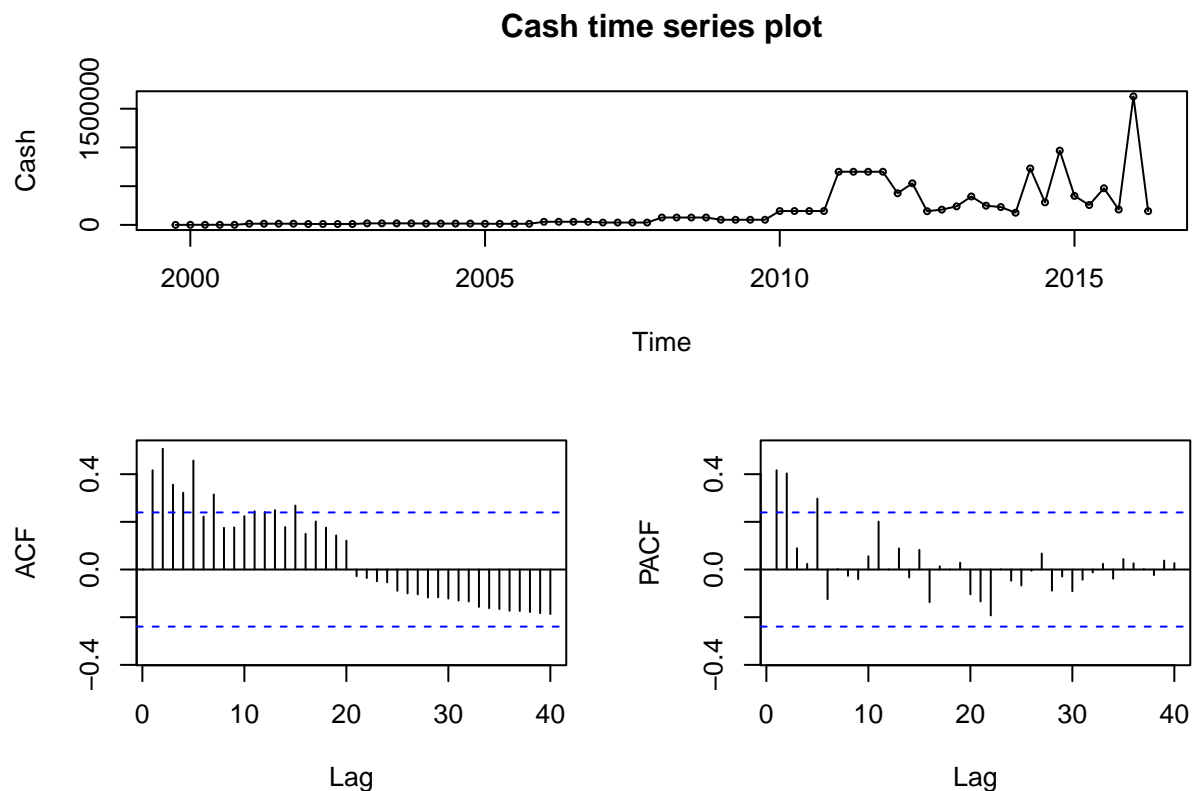
```

data <- read_excel("C:/Users/Вредный Я/Desktop/Data Analysis/CMF/Time Series Econometrics/Exam (Financial indi
# убираем столбцы, заполненные только NA
data <- data[colSums(!is.na(data)) > 0]
# убираем первые две строки
data <- data[-c(1, 2), ]
# переименовываем некоторые столбцы
names(data)[1] <- "Год"
names(data)[2] <- "Квартал"
names(data)[33] <- "Заемные средства.2"
# сбрасываем индексацию строк
rownames(data) <- seq(length=nrow(data))

cash <- as.numeric(rev(data$`Денежные средства и денежные эквиваленты`))
ltb <- as.numeric(rev(data$`Заемные средства`))
stb <- as.numeric(rev(data$`Заемные средства.2`))
rv <- as.numeric(rev(data$`Выручка`))
gp <- as.numeric(rev(data$`Валовая прибыль (убыток)`))
np <- as.numeric(rev(data$`Чистая прибыль (убыток)`))

tsdisplay(ts(cash, frequency=4, start=c(1999, 4)), lag.max=40, main="Cash time series plot",
          xlab="Time", ylab="Cash")

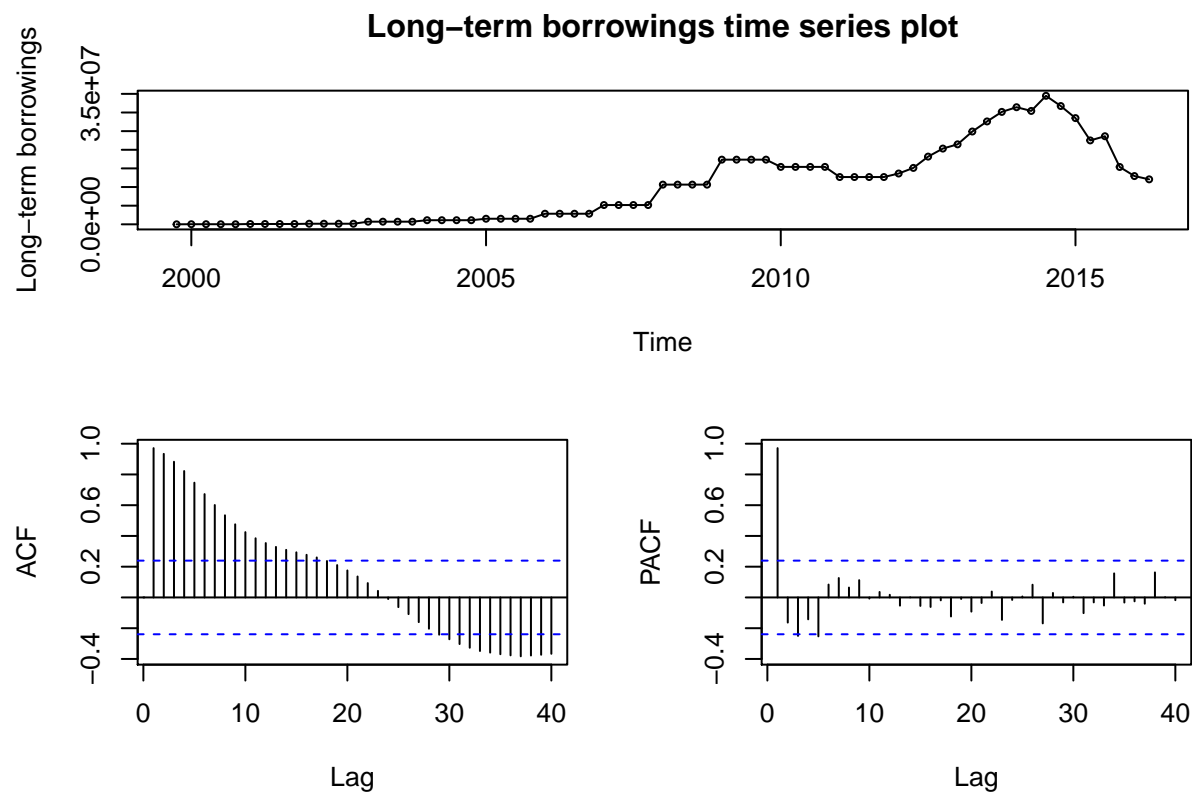
```



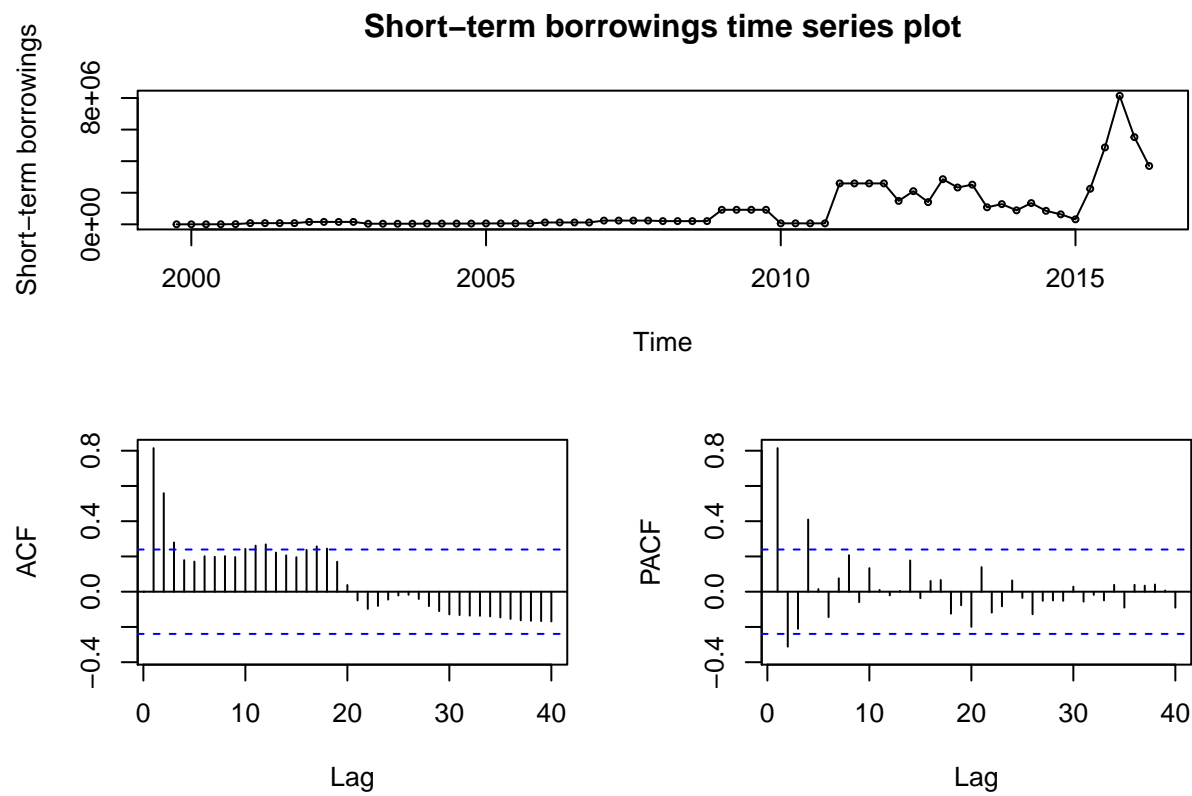
```

tsdisplay(ts(ltb, frequency=4, start=c(1999, 4)), lag.max=40,
          main="Long-term borrowings time series plot", xlab="Time", ylab="Long-term borrowings")

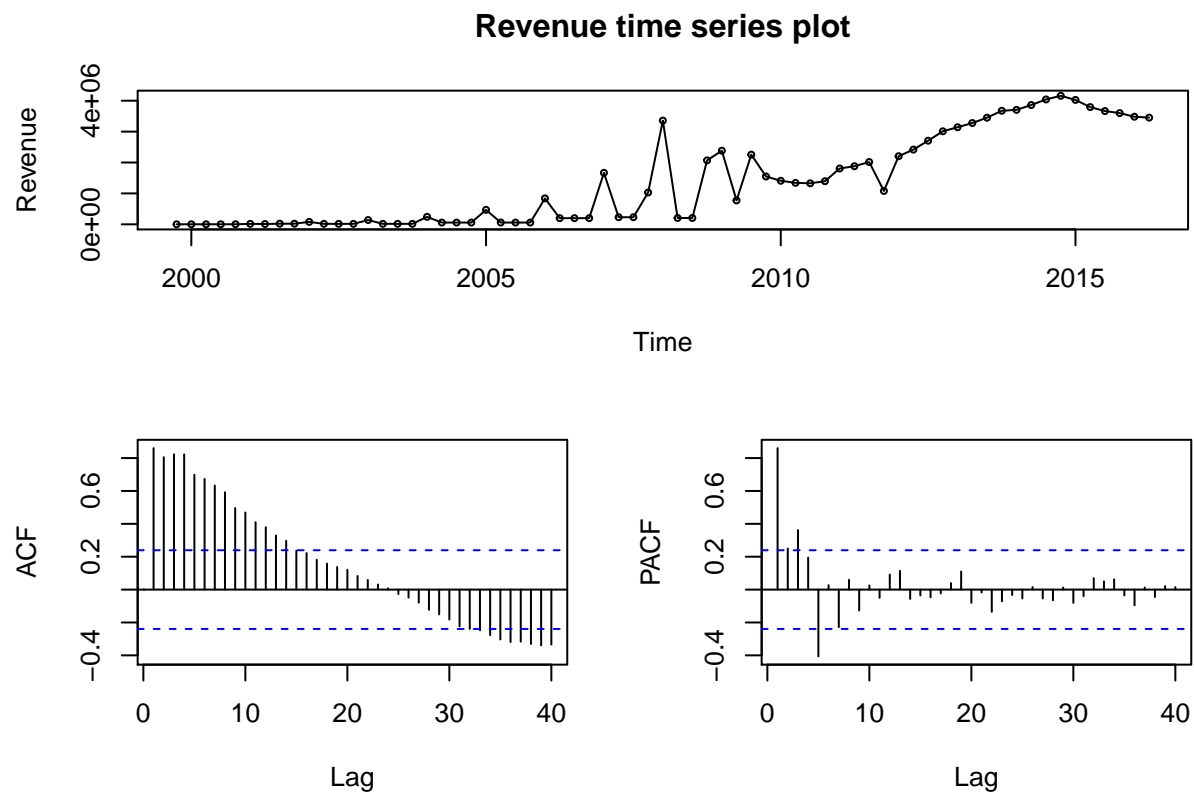
```



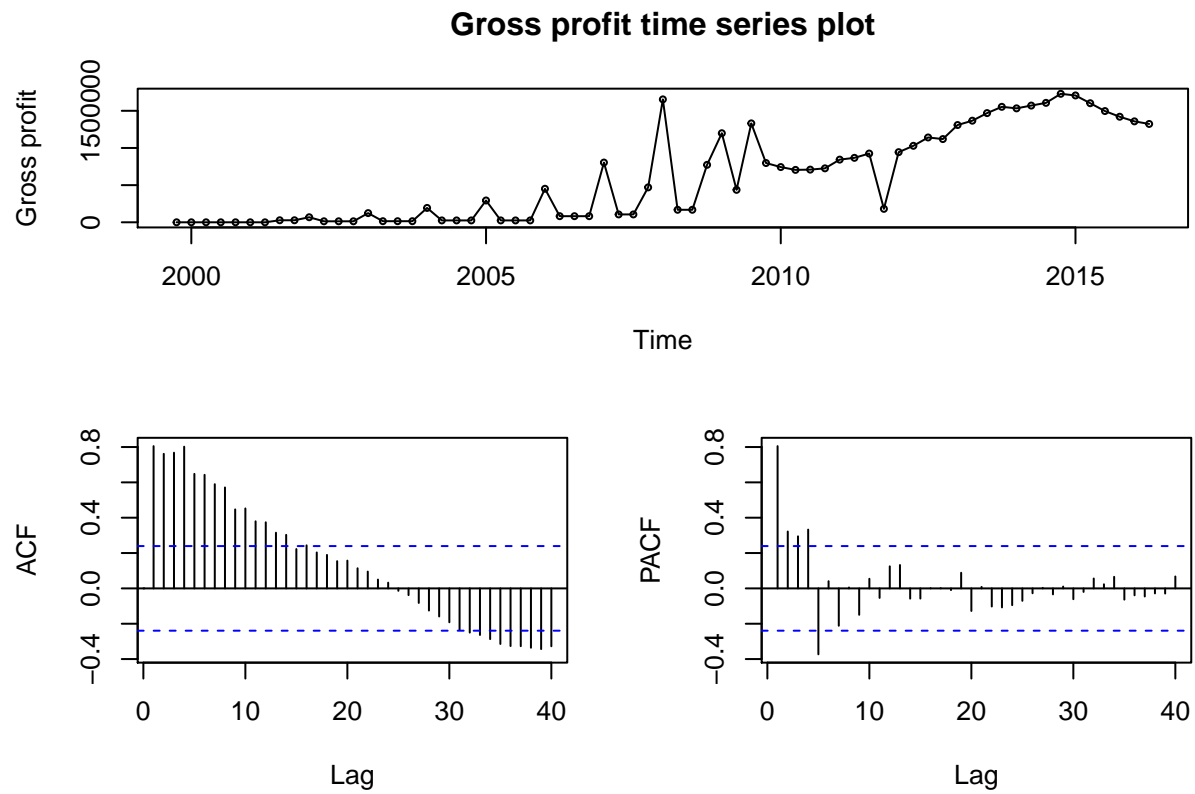
```
tsdisplay(ts(stb, frequency=4, start=c(1999, 4)), lag.max=40,
          main="Short-term borrowings time series plot", xlab="Time", ylab="Short-term borrowings")
```



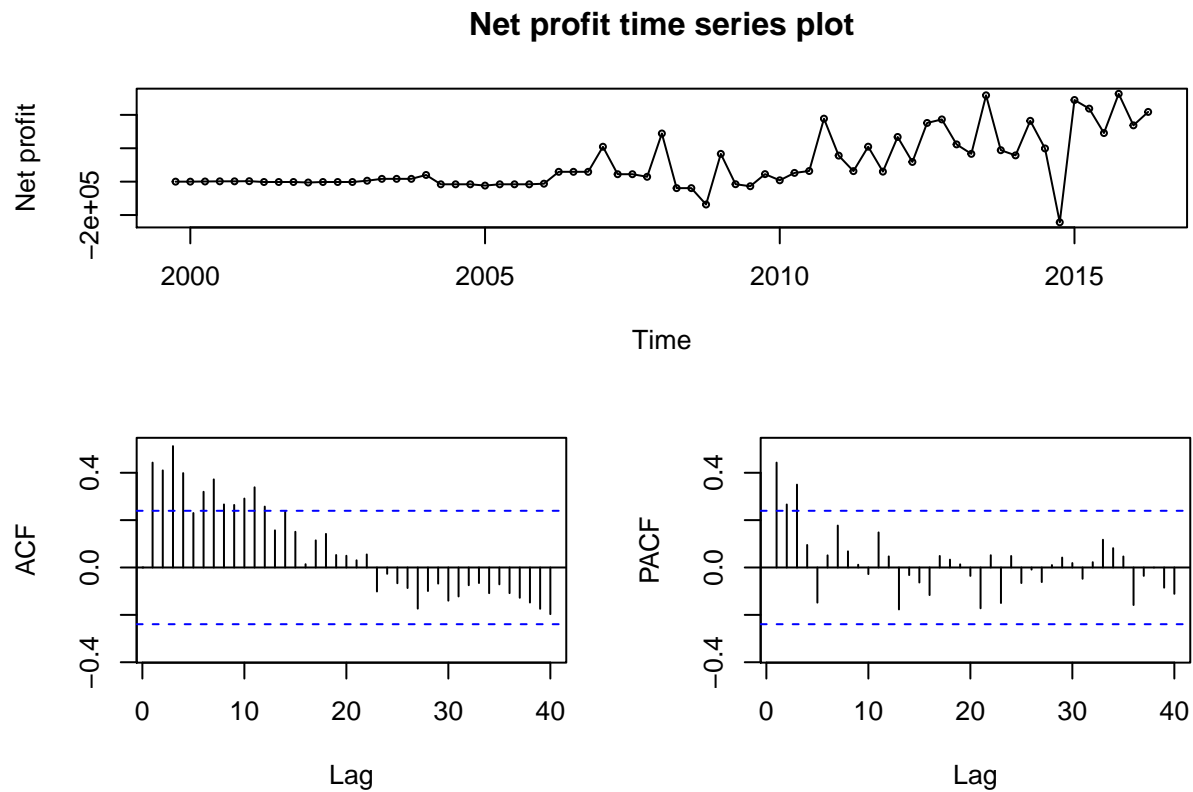
```
tsdisplay(ts(rv, frequency=4, start=c(1999, 4)), lag.max=40, main="Revenue time series plot",
          xlab="Time", ylab="Revenue")
```



```
tsdisplay(ts(gp, frequency=4, start=c(1999, 4)), lag.max=40, main="Gross profit time series plot",
          xlab="Time", ylab="Gross profit")
```

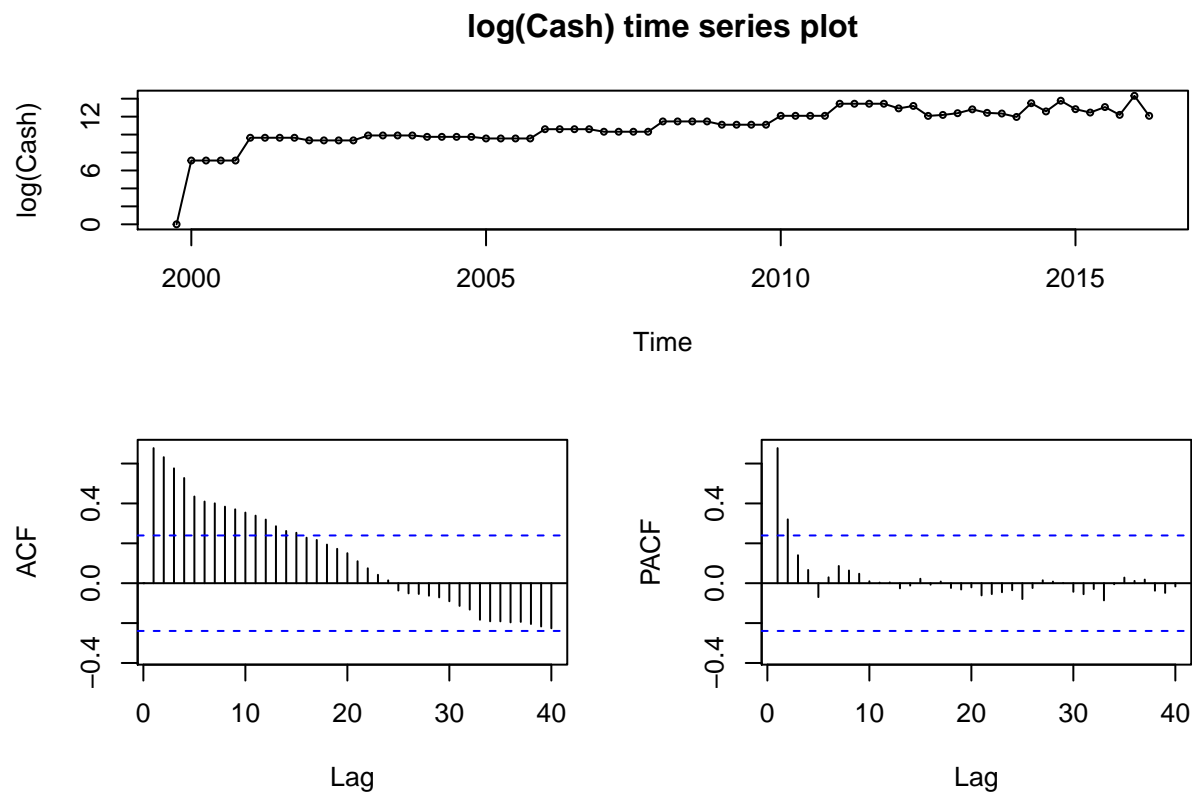


```
tsdisplay(ts(np, frequency=4, start=c(1999, 4)), lag.max=40, main="Net profit time series plot",
          xlab="Time", ylab="Net profit")
```

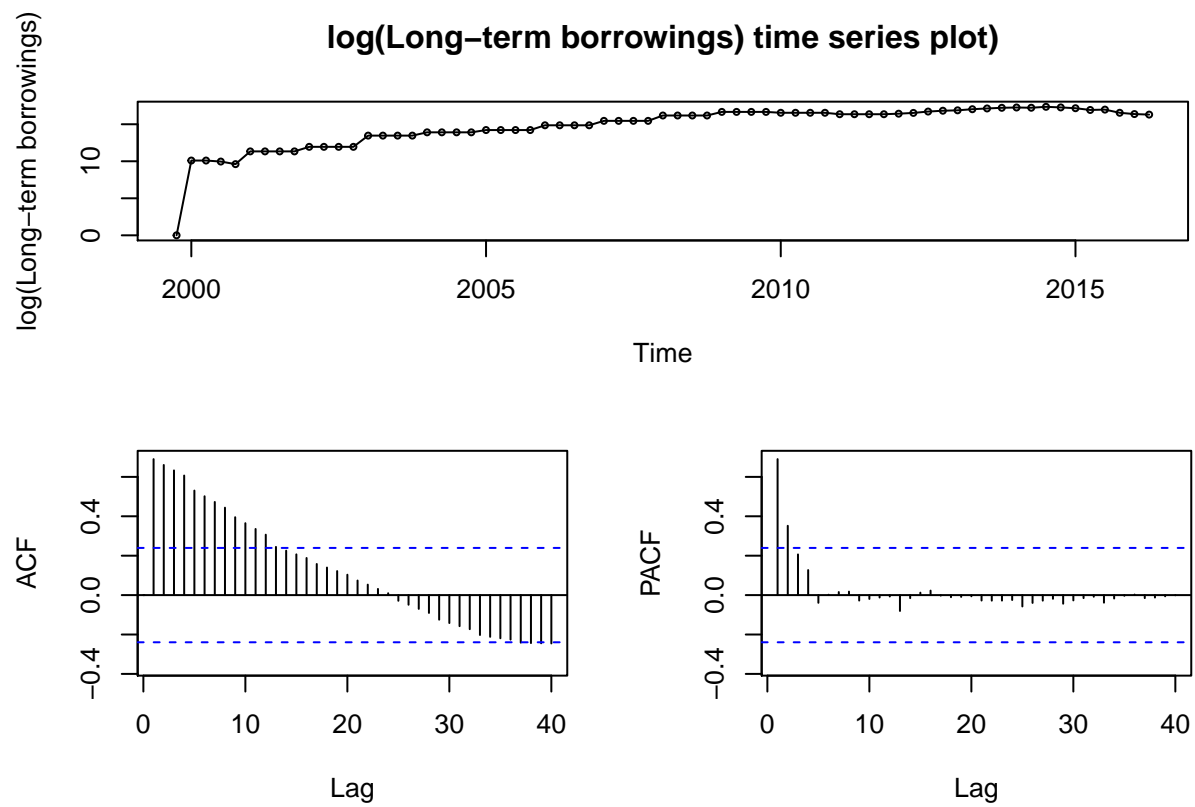


Видно, что масштабы значений очень сильно различаются. Поэтому нарисуем те же самые графики, но в логарифмическом масштабе (также это поможет стабилизировать дисперсию).

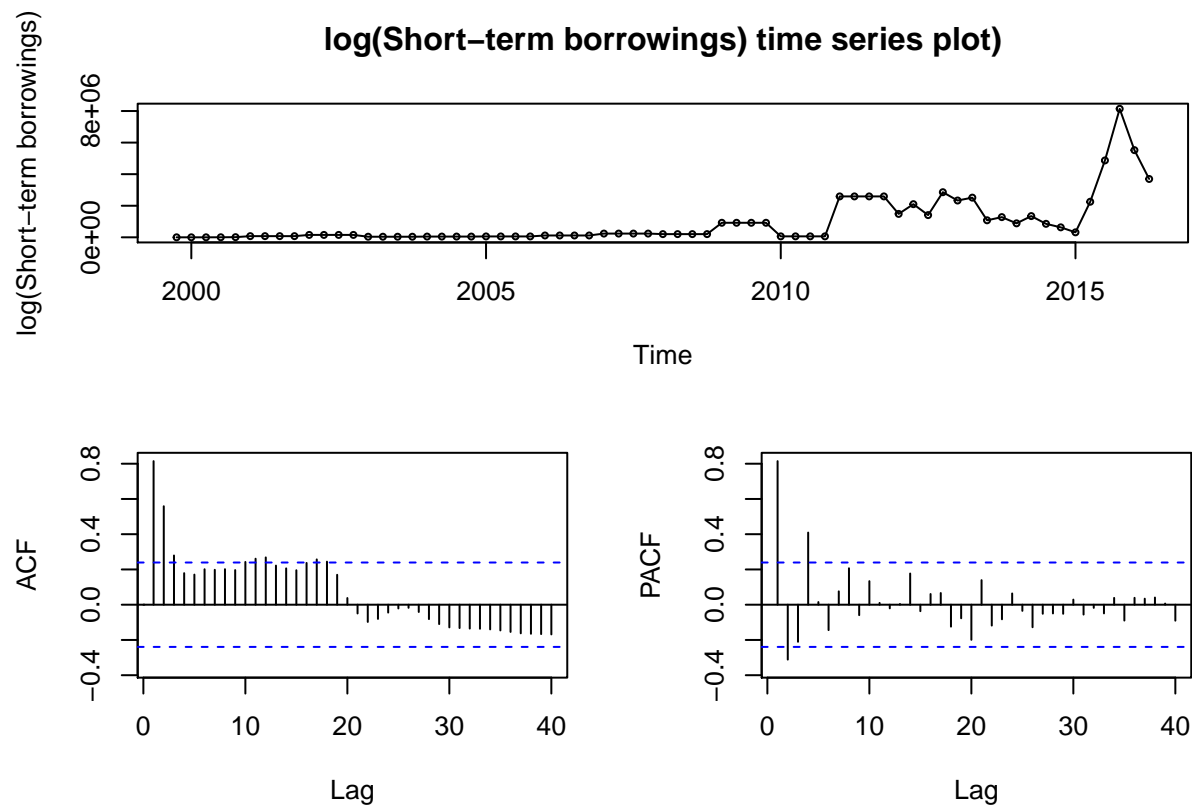
```
# во избежание проблем с логарифмированием переводим все неположительные значения в 1
cash[cash <= 0] <- 1
ltb[ltb <= 0] <- 1
stb[stb <= 0] <- 1
rv[rv <= 0] <- 1
gp[gp <= 0] <- 1
np[np <= 0] <- 1
tsdisplay(ts(log(cash), frequency=4, start=c(1999, 4)), lag.max=40, main="log(Cash) time series plot",
          xlab="Time", ylab="log(Cash)")
```



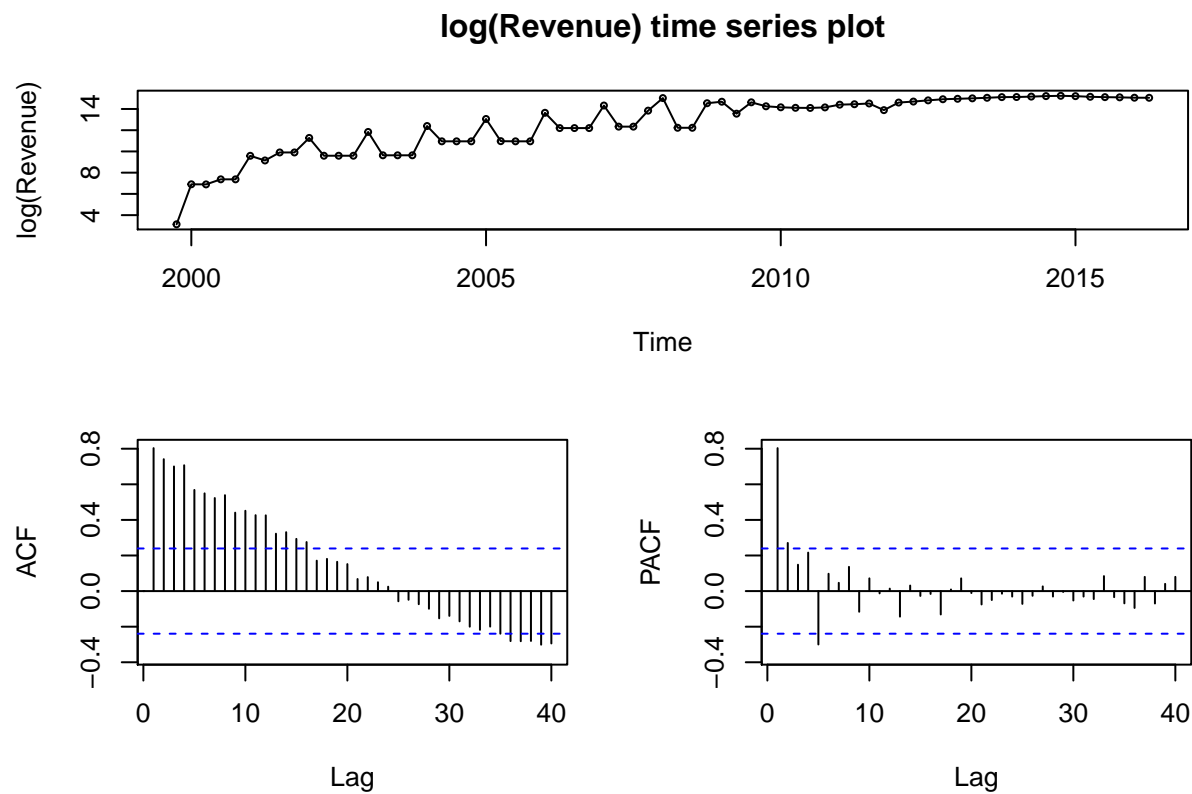
```
tsdisplay(ts(log(ltb), frequency=4, start=c(1999, 4)), lag.max=40,
          main="log(Long-term borrowings) time series plot", xlab="Time", ylab="log(Long-term borrowings)")
```

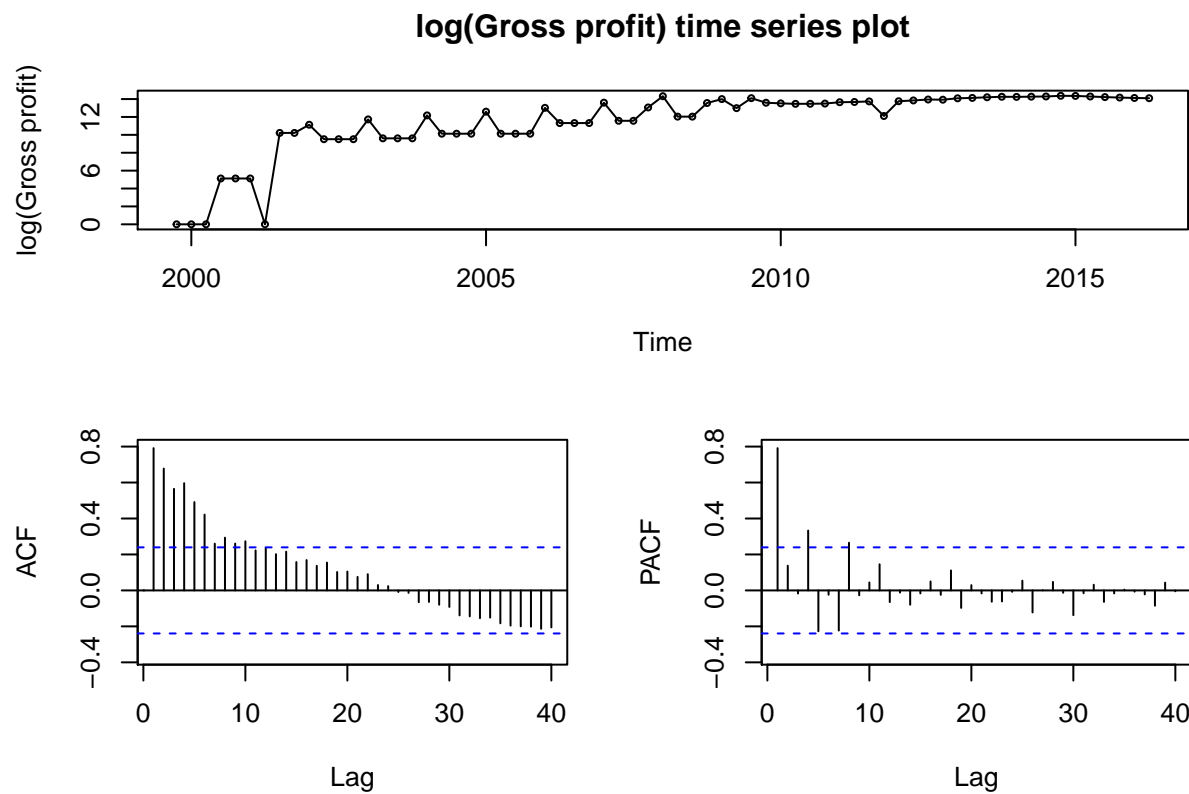
```
tsdisplay(ts(stb, frequency=4, start=c(1999, 4)), lag.max=40,
          main="log(Short-term borrowings) time series plot", xlab="Time", ylab="log(Short-term borrowings)")
```



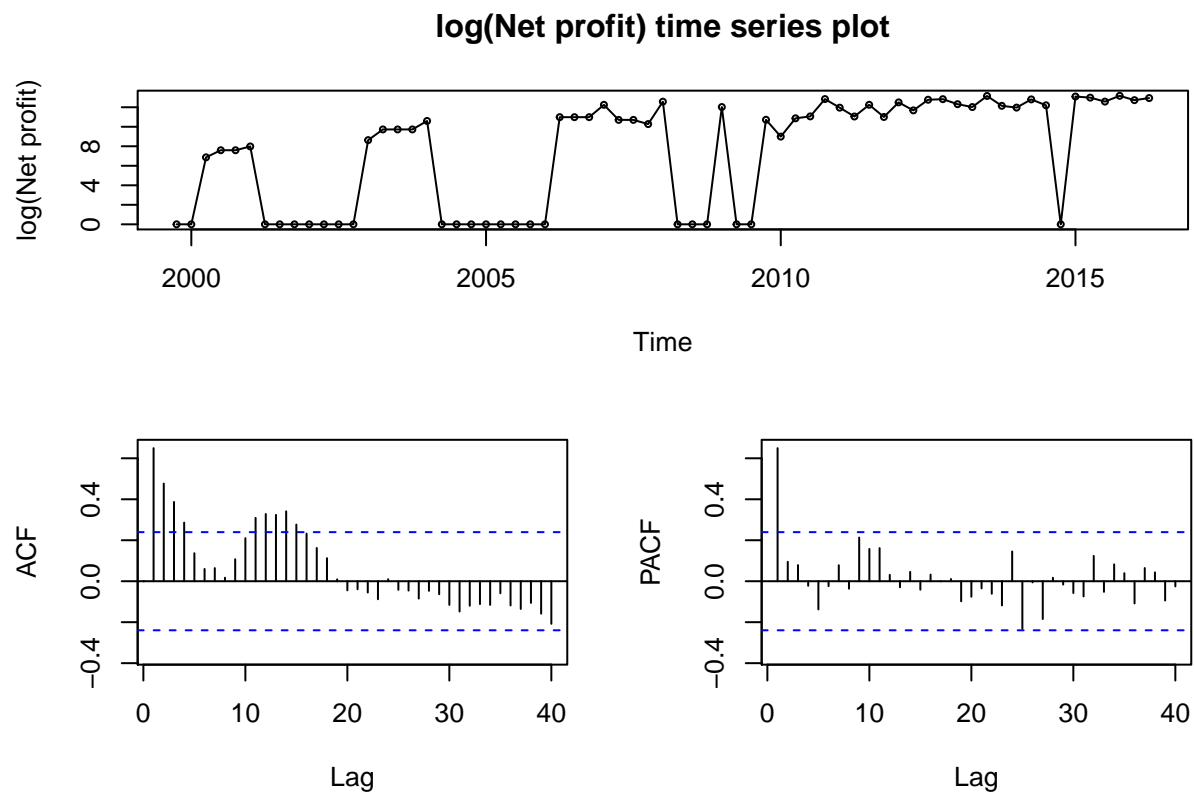
```
tsdisplay(ts(log(rv), frequency=4, start=c(1999, 4)), lag.max=40, main="log(Revenue) time series plot",
          xlab="Time", ylab="log(Revenue)")
```



```
tsdisplay(ts(log(gp), frequency=4, start=c(1999, 4)), lag.max=40, main="log(Gross profit) time series plot",
          xlab="Time", ylab="log(Gross profit)")
```



```
tsdisplay(ts(log(np), frequency=4, start=c(1999, 4)), lag.max=40, main="log(Net profit) time series plot",
          xlab="Time", ylab="log(Net profit)")
```

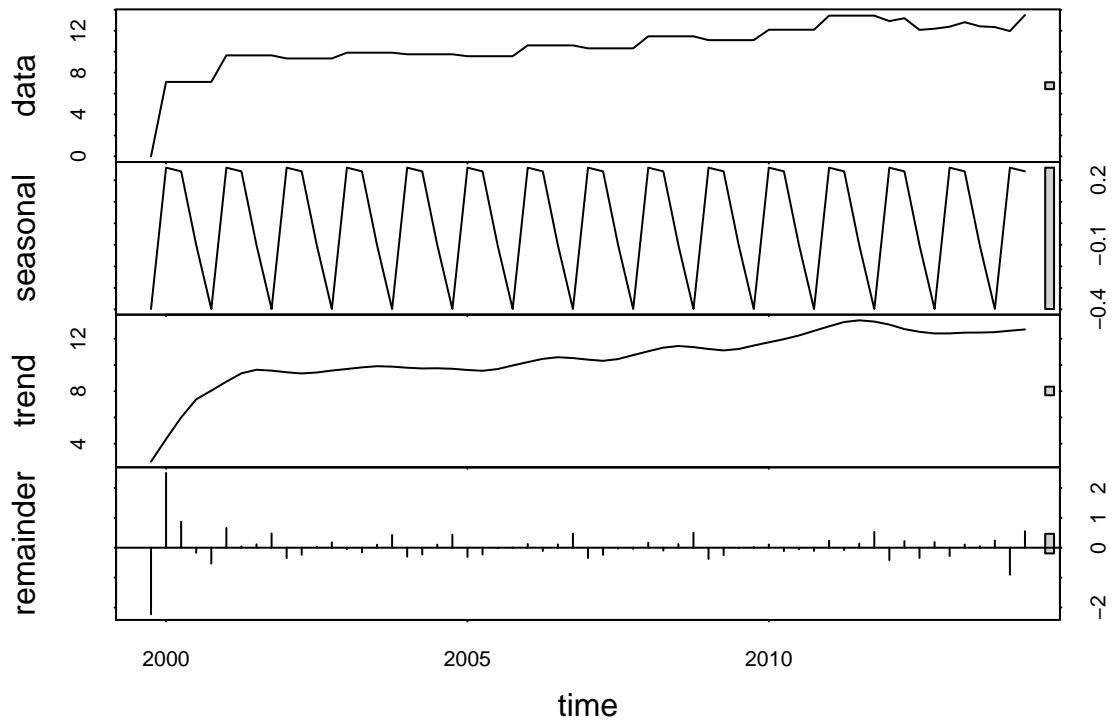


2 Ретроспективный прогноз

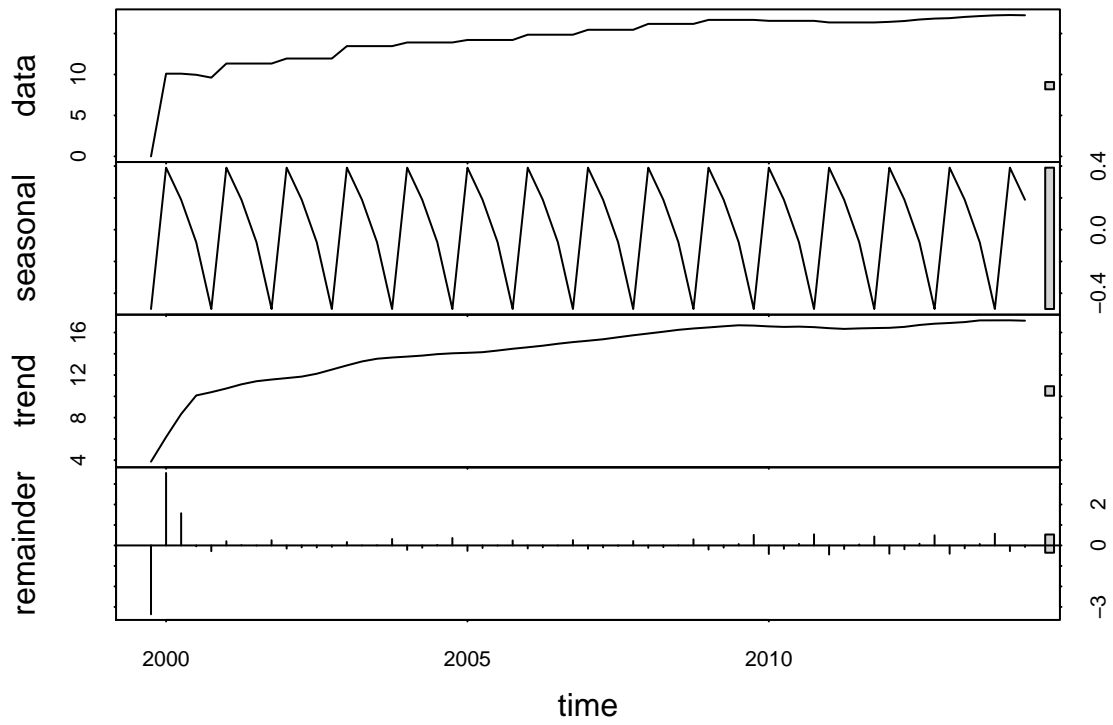
2.1 STL

STL-декомпозиция рядов:

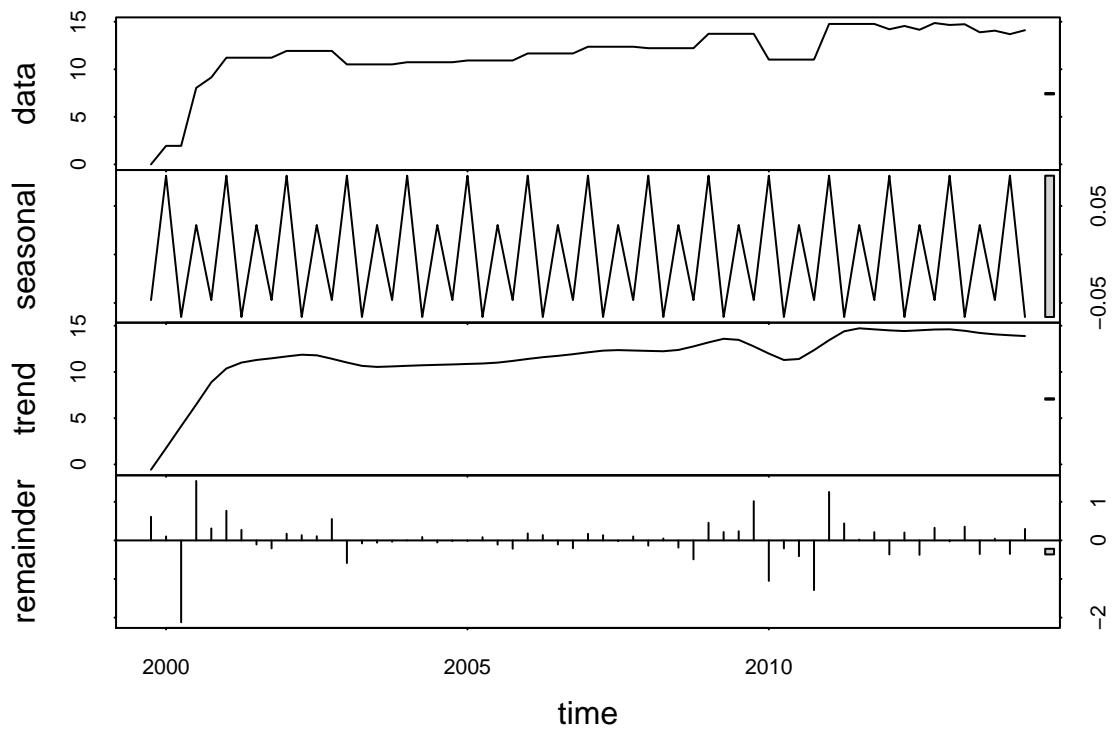
```
plot(stl(ts(log(cash[1:59])), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



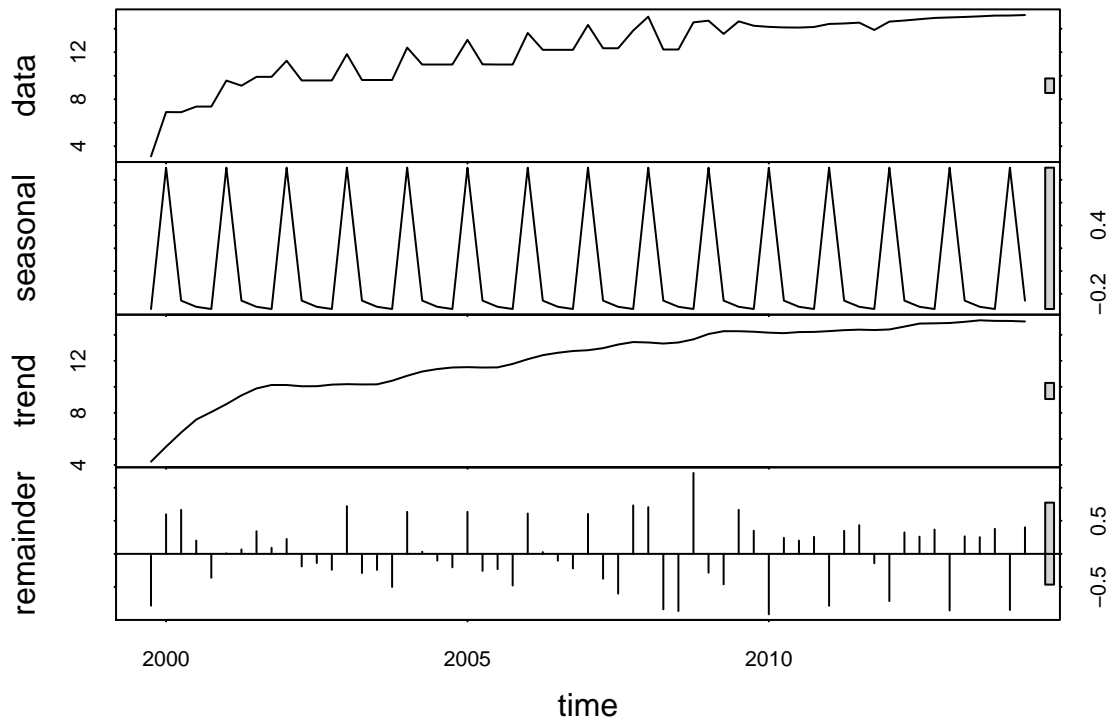
```
plot(stl(ts(log(ltb[1:59])), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



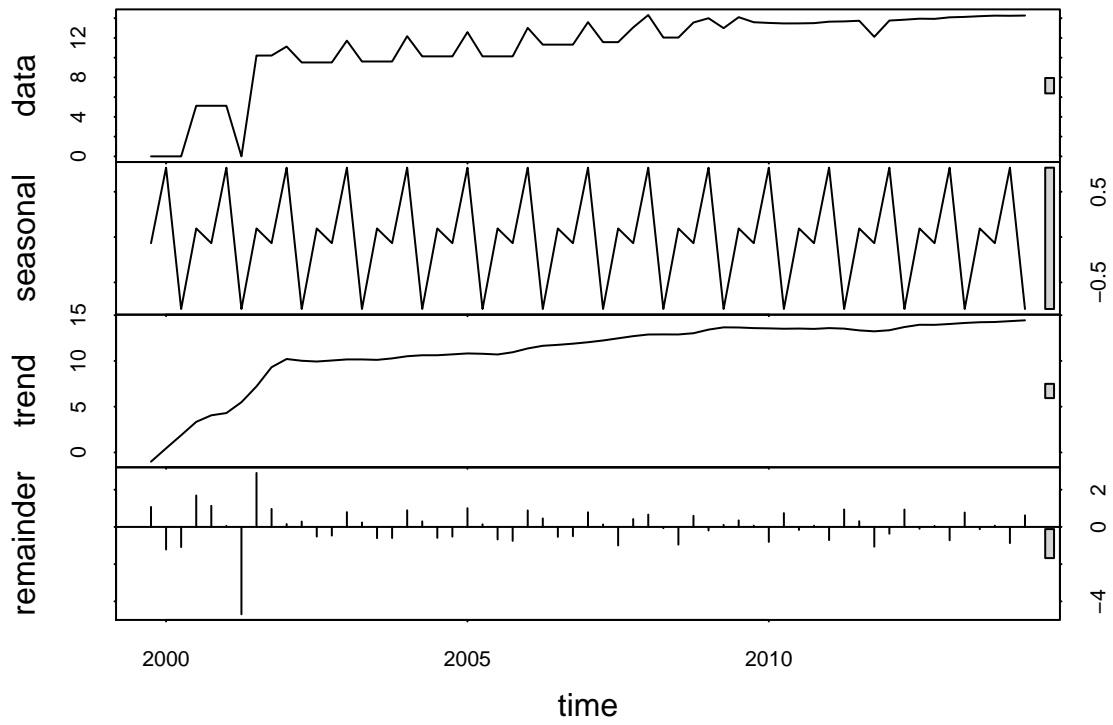
```
plot(stl(ts(log(stb[1:59])), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



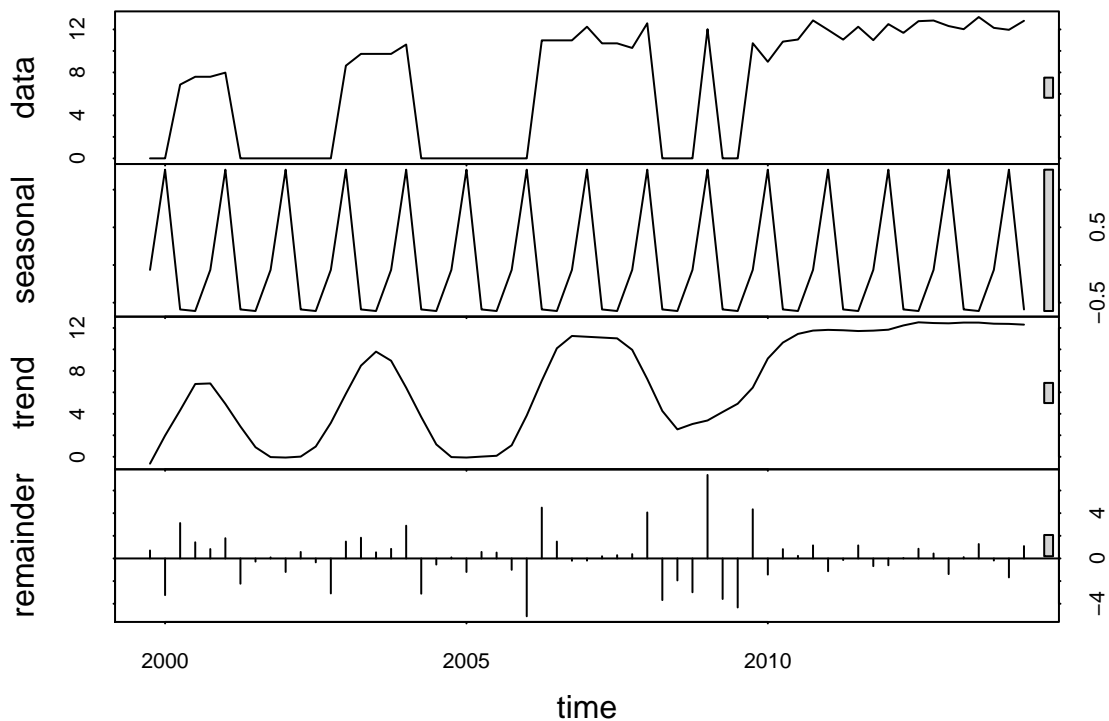
```
plot(stl(ts(log(rv[1:59])), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```

```
plot(stl(ts(log(gp[1:59]), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



```
plot(stl(ts(log(np[1:59])), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



2.2 Дифференцирование и стационарность

Применим сезонное/обычное дифференцирование по необходимости. Проверим стационарность с помощью критерия Дики-Фуллера.

```
cash.diff <- diff(log(cash[1:59]), 4)
adf.test(cash.diff)
```

```
## Warning in adf.test(cash.diff): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: cash.diff
## Dickey-Fuller = -4.5548, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
ltb.diff <- diff(log(ltb[1:59]), 1)
adf.test(ltb.diff)
```

```
## Warning in adf.test(ltb.diff): p-value smaller than printed p-value
```

```
##
```

```
## Augmented Dickey-Fuller Test
##
## data: ltb.diff
## Dickey-Fuller = -6.073, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(log(stb[1:59]))
```

```
## Warning in adf.test(log(stb[1:59])): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: log(stb[1:59])
## Dickey-Fuller = -4.5022, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(log(rv[1:59]))
```

```
## Warning in adf.test(log(rv[1:59])): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: log(rv[1:59])
## Dickey-Fuller = -4.8571, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
gp.diff <- diff(log(gp[1:59]), 1)
adf.test(gp.diff)
```

```
## Warning in adf.test(gp.diff): p-value smaller than printed p-value
```

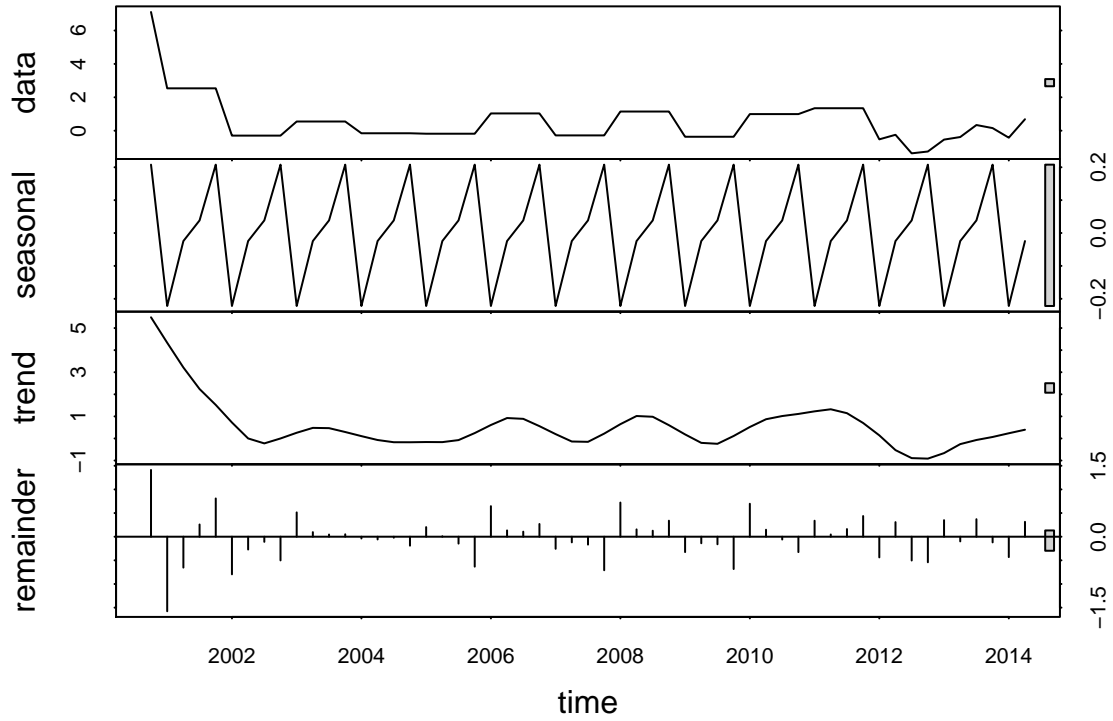
```
##
## Augmented Dickey-Fuller Test
##
## data: gp.diff
## Dickey-Fuller = -5.2501, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
np.diff <- diff(log(np[1:59]), 4)
adf.test(np.diff)
```

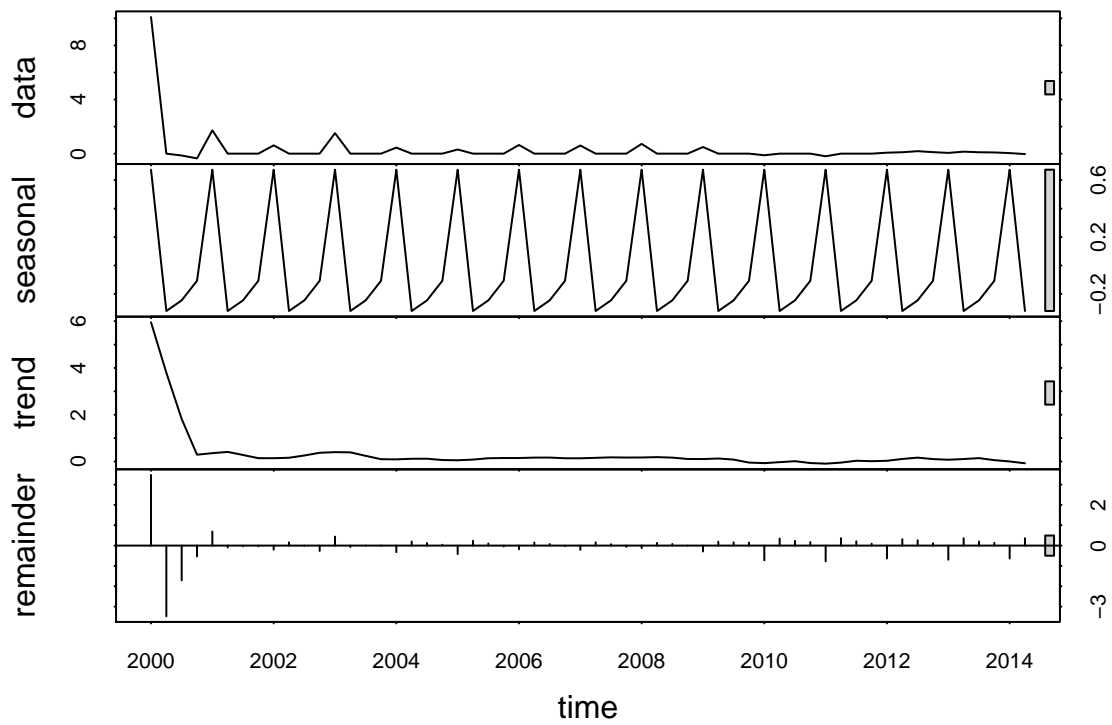
```
## Warning in adf.test(np.diff): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: np.diff
## Dickey-Fuller = -4.6441, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

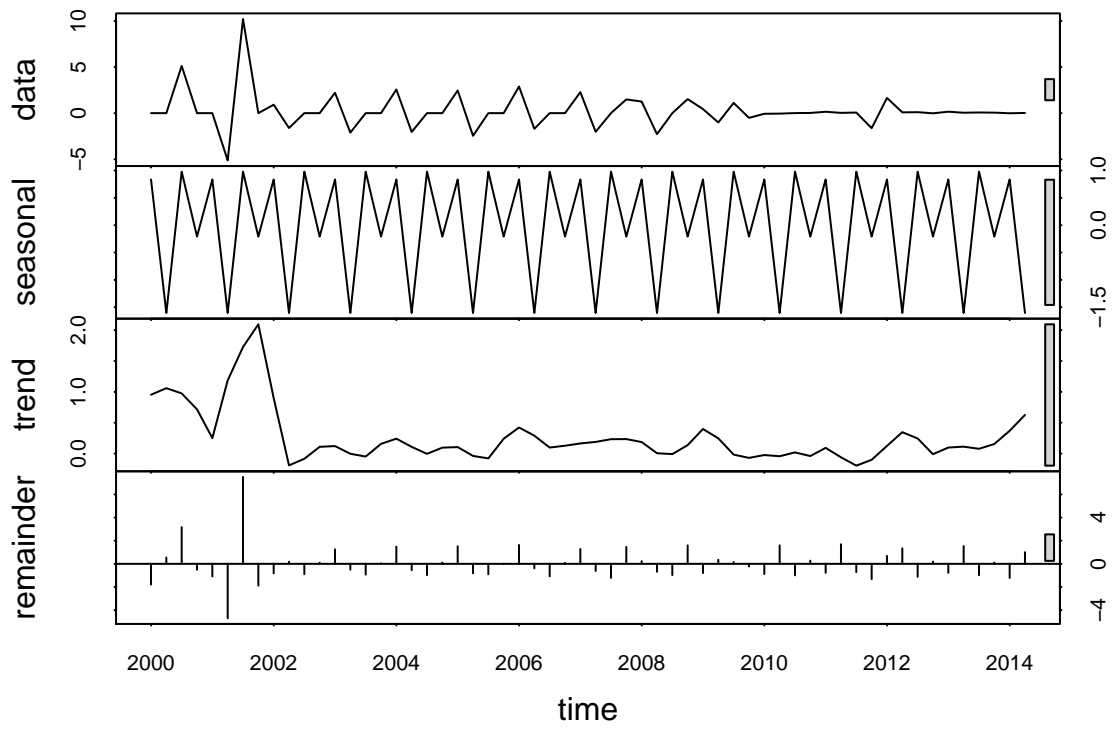
```
plot(stl(ts(cash.diff, frequency=4, start=c(2000, 4)), s.window="periodic", robust=FALSE))
```



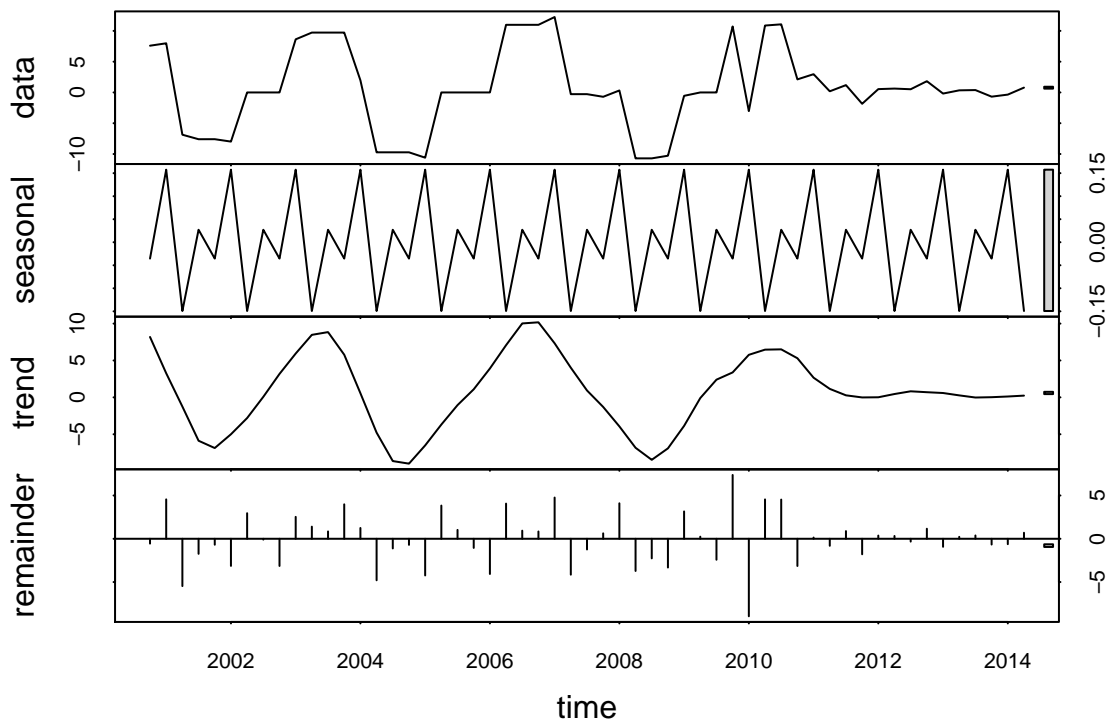
```
plot(stl(ts(ltb.diff, frequency=4, start=c(2000, 1)), s.window="periodic", robust=FALSE))
```



```
plot(stl(ts(gp.diff, frequency=4, start=c(2000, 1)), s.window="periodic", robust=FALSE))
```



```
plot(stl(ts(np.diff, frequency=4, start=c(2000, 4)), s.window="periodic", robust=FALSE))
```



2.3 Выбор лучшей модели, анализ остатков и прогноз

Далее нам понадобятся три функции:

- `best.arma.model`: ищет лучшую ARIMA-модель, перебирая модели с разными параметрами p , q , P , Q и сравнивая их по критерию Акаике;
- `arma.model.analysis`: анализирует остатки модели (проверяет несмещенность с помощью теста Стьюдента, стационарность с помощью теста Дики-Фуллера и неавтокоррелированность с помощью теста Бокса-Пирса);
- `arma.model.predict`: делает предсказание.

```
best.arma.model <- function(data, p0=4, q0=4, P0=1, Q0=1, d=0, D=0) {
  # все возможные комбинации параметров
  params <- expand.grid(0:p0, 0:q0, 0:P0, 0:Q0)
  models <- vector("list", nrow(params))
  aics <- numeric(nrow(params))
  for (i in 1:nrow(params)) {
    # tryCatch нужен, потому что на некоторых наборах параметров модель может не обучаться
    tryCatch({
      models[[i]] <- arima(data, order=c(params[i, 1], d, params[i, 2]),
                           seasonal=list(order=c(params[i, 3], D, params[i, 4]), period=4))
      aics[[i]] <- models[[i]]$aic
    }, error=function(e) {
```



```

    cat("Wrong parameters:", params[i, 1], params[i, 2], params[i, 3], params[i, 4], "\n")
  })
}
# там, где модель не обучается, будет стоять AIC=0
# нам этого не нужно, поэтому все такие значения поменяем на какое-нибудь очень большое
aics[aics==0] <- 100000000
# лучшая модель и лучшие параметры
best.model <- arima(data, order=c(params[which.min(aics), 1], d, params[which.min(aics), 2]),
                    seasonal=list(order=c(params[which.min(aics), 3], D, params[which.min(aics), 4]),
                                   period=4))

print(best.model)
cat("Best parameters:", params[which.min(aics), 1], params[which.min(aics), 2],
    params[which.min(aics), 3], params[which.min(aics), 4])
return(best.model)
}

arima.model.analysis <- function(model, data, ylab) {
  tsdisplay(ts(model$residuals, frequency=4, start=c(1999, 4)), lag.max=40,
    main=paste(ylab, "residuals time series plot", sep=" "), xlab="Time",
    ylab=paste(ylab, "residuals", sep=" "))
  # анализ остатков
  print(t.test(model$residuals))
  print(adf.test(model$residuals))
  print(Box.test(model$residuals))
}

arima.model.predict <- function(model, data, ylab, start=c(2014, 3)) {
  data.predict <- as.numeric(predict(model, n.ahead=8)$pred)
  plot(ts(data, frequency=4, start=c(1999, 4)), ylab=ylab, xlim=c(1999, start[1] + 2),
    ylim=c(0, max(data, exp(data.predict))))
  title(paste(ylab, "time series plot", sep=" "))
  lines(ts(exp(data.predict), frequency=4, start=start), col="red")
  return(exp(data.predict))
}

```

2.3.1 Денежные средства и денежные эквиваленты

Найдем лучшую модель для денежных средств и денежных эквивалентов.

```
best.cash.model <- best.arima.model(log(cash[1:59]), D=1)
```

```

##
## Call:
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),
##    2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),
##    4]), period = 4))
##
## Coefficients:
##      ar1
##    0.9283
## s.e. 0.0796
##

```

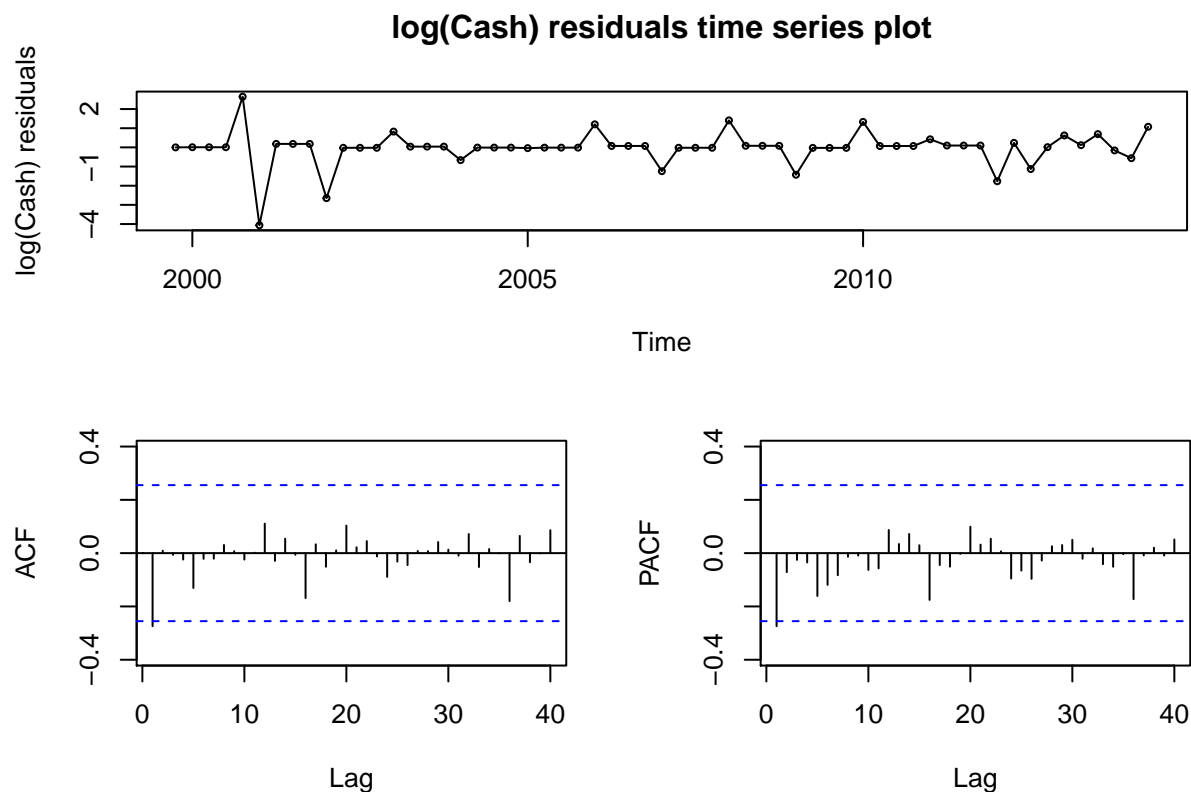
```
## sigma^2 estimated as 0.8658: log likelihood = -75.07, aic = 154.14
## Best parameters: 1 0 0 0
```

Проанализируем остатки.

```
arima.model.analysis(best.cash.model, log(cash[1:59]), "log(Cash)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = -0.24942, df = 58, p-value = 0.8039
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.2654140 0.2065992
## sample estimates:
## mean of x
## -0.02940744
```

```
## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```



```
##
## Augmented Dickey-Fuller Test
##
```

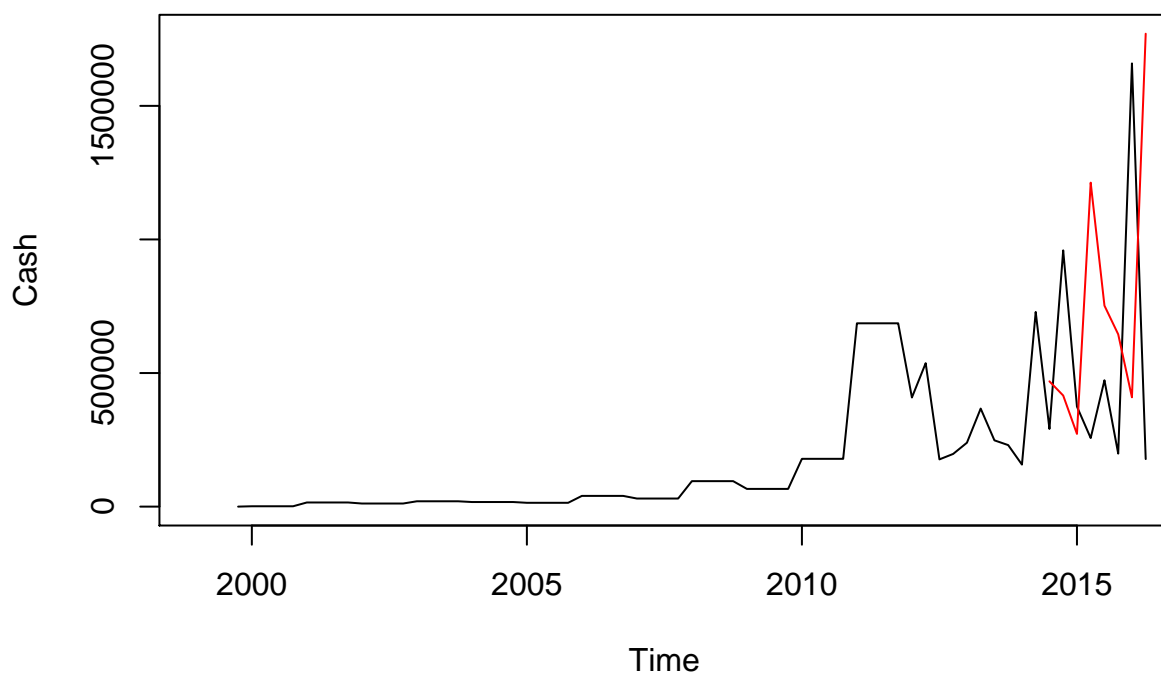
```
## data: model$residuals
## Dickey-Fuller = -4.1785, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 4.4299, df = 1, p-value = 0.03531
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
cash.predict <- arima.model.predict(best.cash.model, cash, "Cash")
```

Cash time series plot



2.3.2 Долгосрочные заемные средства

Найдем лучшую модель для долгосрочных заемных средств.

```
best.ltb.model <- best.arima.model(log(ltb[1:59]), d=1)
```

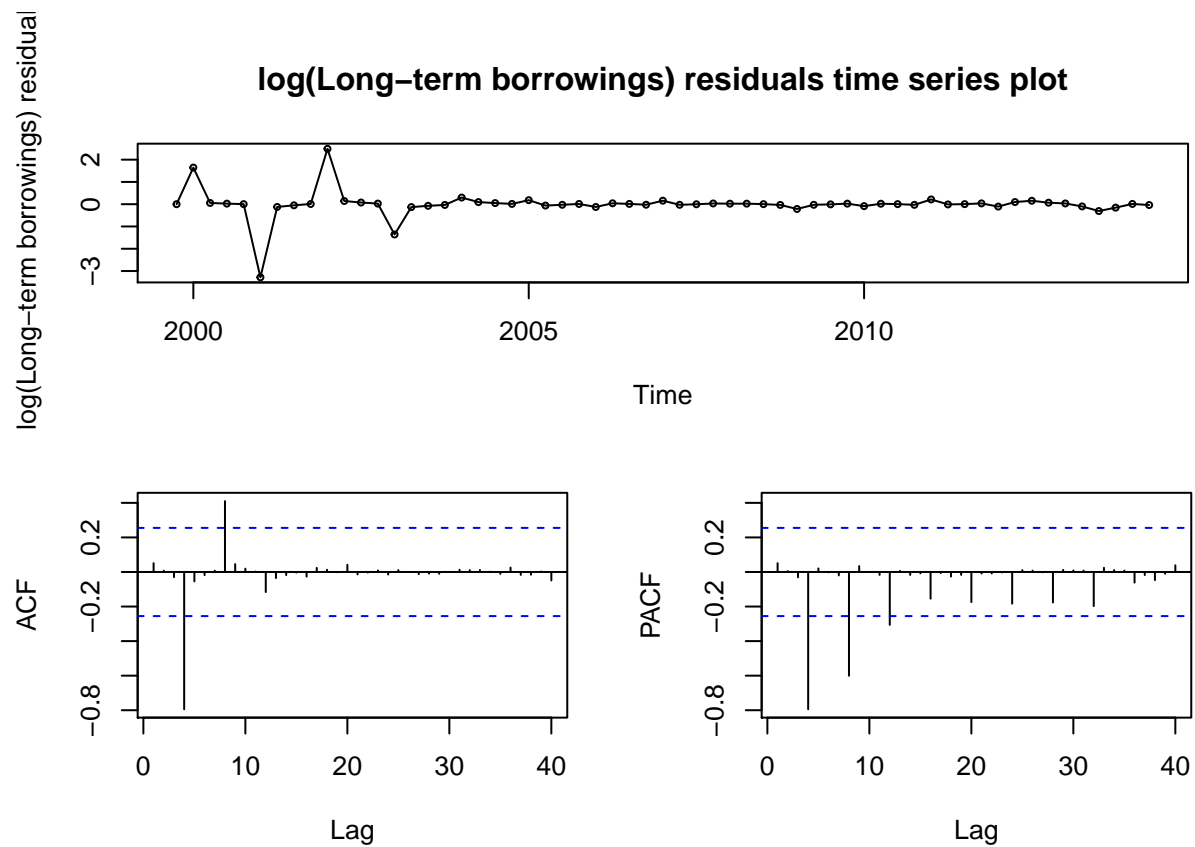
```
##
## Call:
```

```
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),
## 2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),
## 4]), period = 4))
##
## Coefficients:
##      ma1      ma2      ma3      ma4      sar1      sma1
##    -0.0348 -0.0285 -0.0348  1.0000  0.7985  1.0000
## s.e.  0.1535  0.1243  0.1536  0.1776  0.1276  0.1759
##
## sigma^2 estimated as 0.3802: log likelihood = -77.93, aic = 169.85
## Best parameters: 0 4 1 1
```

Проанализируем остатки.

```
arma.model.analysis(best.ltb.model, log(ltb[1:59]), "log(Long-term borrowings)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = -0.074885, df = 58, p-value = 0.9406
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.1666904  0.1546682
## sample estimates:
##      mean of x
## -0.006011093
##
## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```



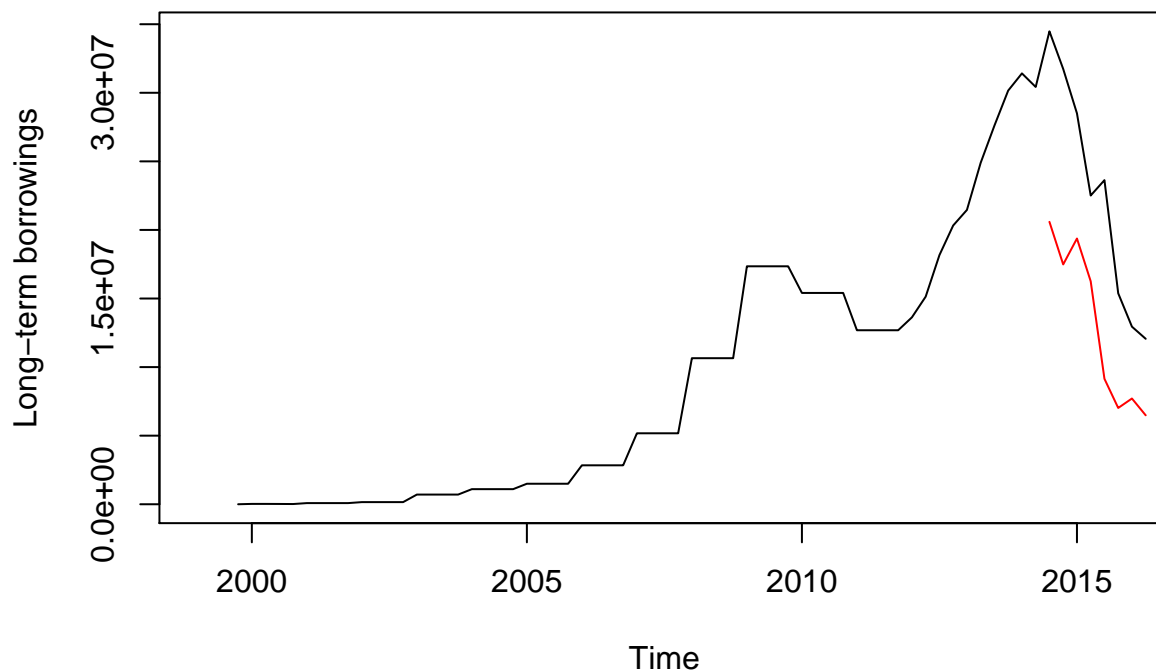
```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -12.776, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 0.1538, df = 1, p-value = 0.6949
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
ltb.predict <- arima.model.predict(best.ltb.model, ltb, "Long-term borrowings")
```

Long-term borrowings time series plot



2.3.3 Краткосрочные заемные средства

Найдем лучшую модель для краткосрочных заемных средств.

```
best.stb.model <- best.arima.model(log(stb[1:59]))
```

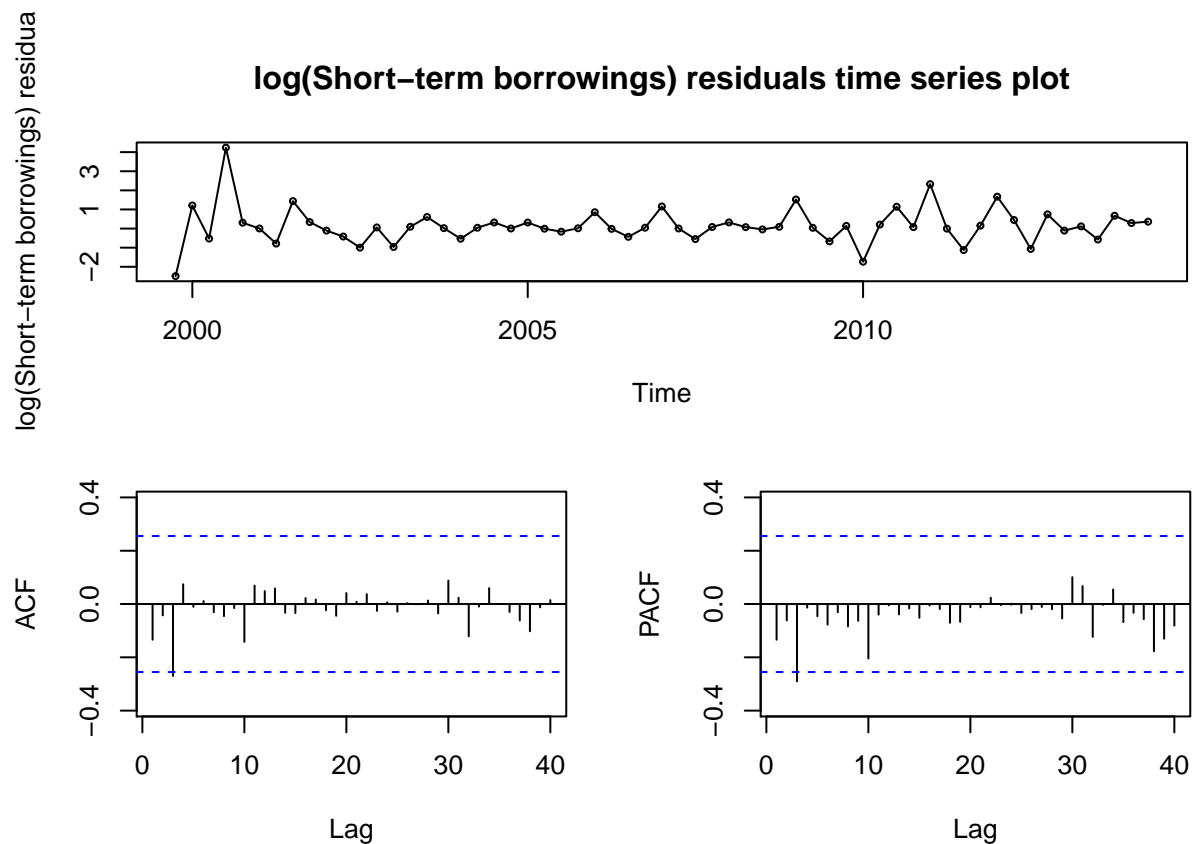
```
##
## Call:
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),
## 2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),
## 4]), period = 4))
##
## Coefficients:
##      ar1    ar2    ar3    sar1 intercept
##      1.0359 0.4974 -0.5683 -0.5529  10.5927
## s.e. 0.1157 0.1827 0.1284 0.1438   2.3034
##
## sigma^2 estimated as 0.9152: log likelihood = -83.36, aic = 178.71
## Best parameters: 3 0 1 0
```

Проанализируем остатки.

```
arima.model.analysis(best.stb.model, log(stb[1:59]), "log(Short-term borrowings)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = 1.1227, df = 58, p-value = 0.2662
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.1092358 0.3882729
## sample estimates:
## mean of x
## 0.1395186

## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```

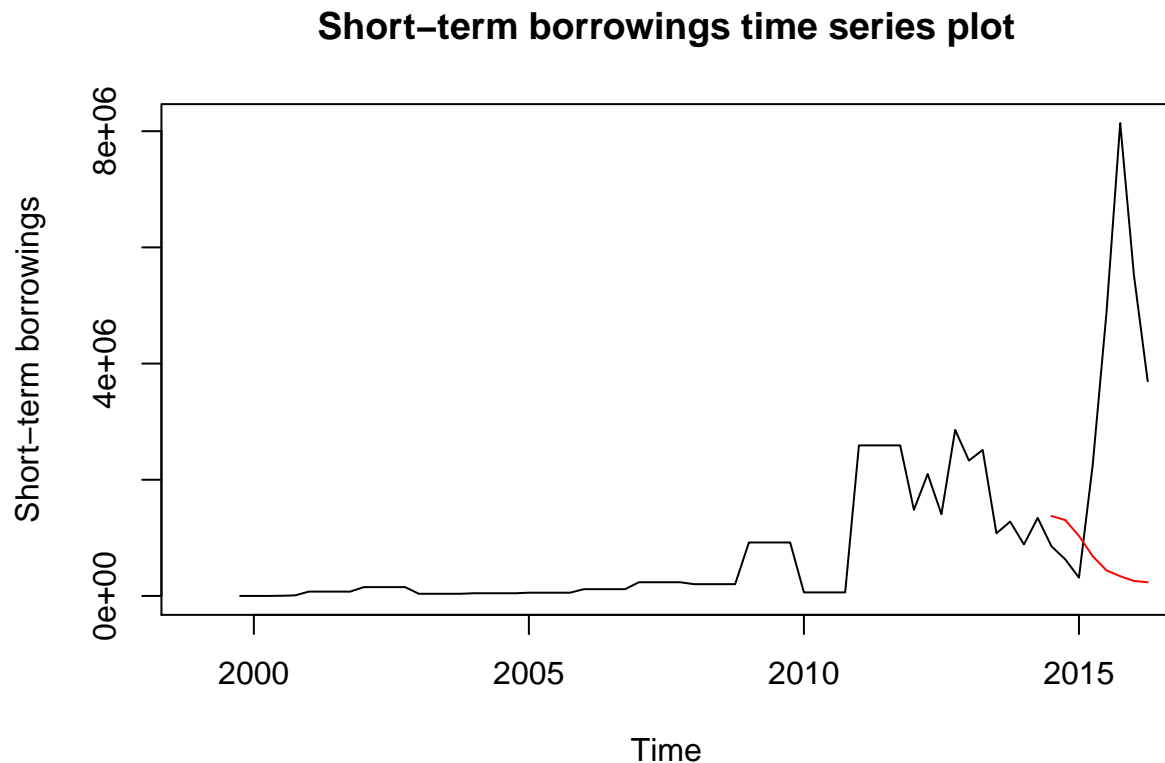


```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -4.6279, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 1.0581, df = 1, p-value = 0.3036
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
stb.predict <- arima.model.predict(best.stb.model, stb, "Short-term borrowings")
```



2.3.4 Выручка

Найдем лучшую модель для выручки.

```
best.rv.model <- best.arima.model(log(rv[1:59]))
```

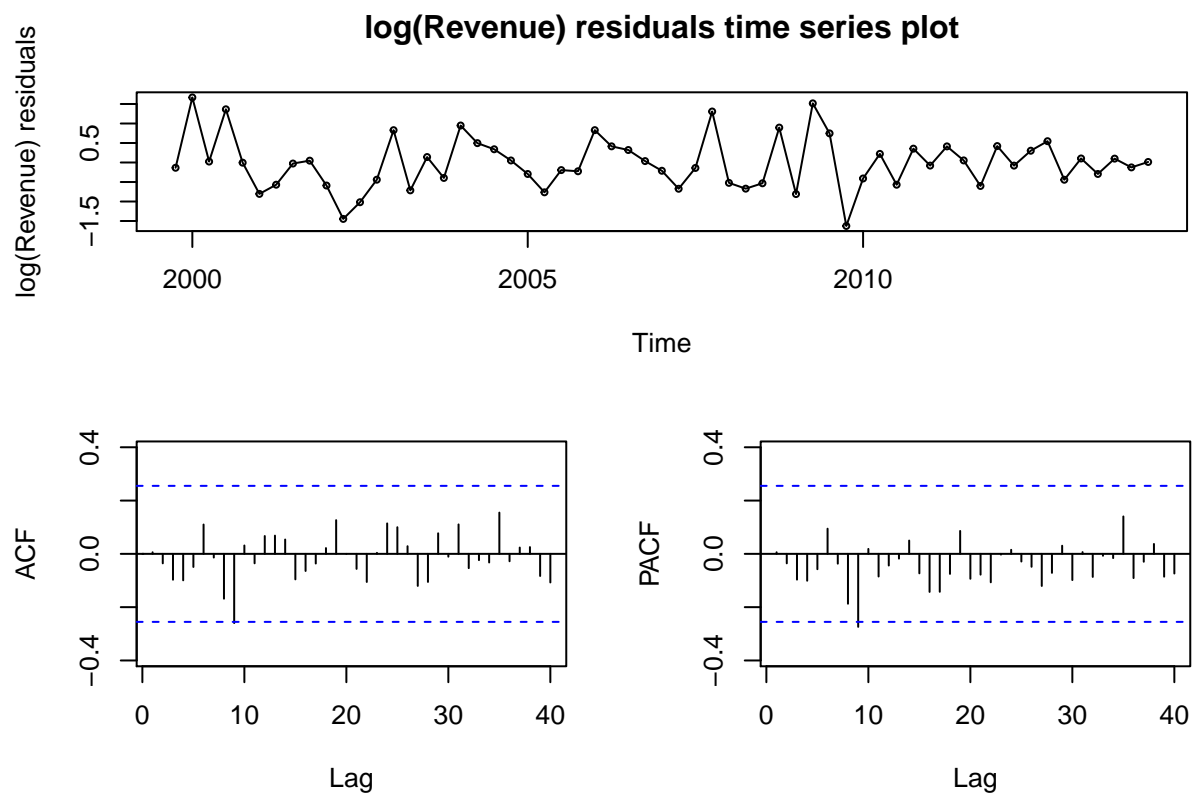
```
##  
## Call:  
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),  
##    2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),  
##    4]), period = 4))  
##  
## Coefficients:  
##      ar1    ar2    ar3    ma1    sar1    sma1 intercept  
##    0.5649 0.0565 0.3751 0.3263 0.8990 -0.1143 16.6468  
## s.e. 0.0016 0.0006 0.0005   NaN 0.0002 0.1452 348.2821  
##  
## sigma^2 estimated as 0.4396: log likelihood = -63.05, aic = 142.11  
## Best parameters: 3 1 1 1
```


Проанализируем остатки.

```
arima.model.analysis(best.rv.model, log(rv[1:59]), "log(Revenue)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = -0.17675, df = 58, p-value = 0.8603
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.1896756 0.1588975
## sample estimates:
## mean of x
## -0.01538904

## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```



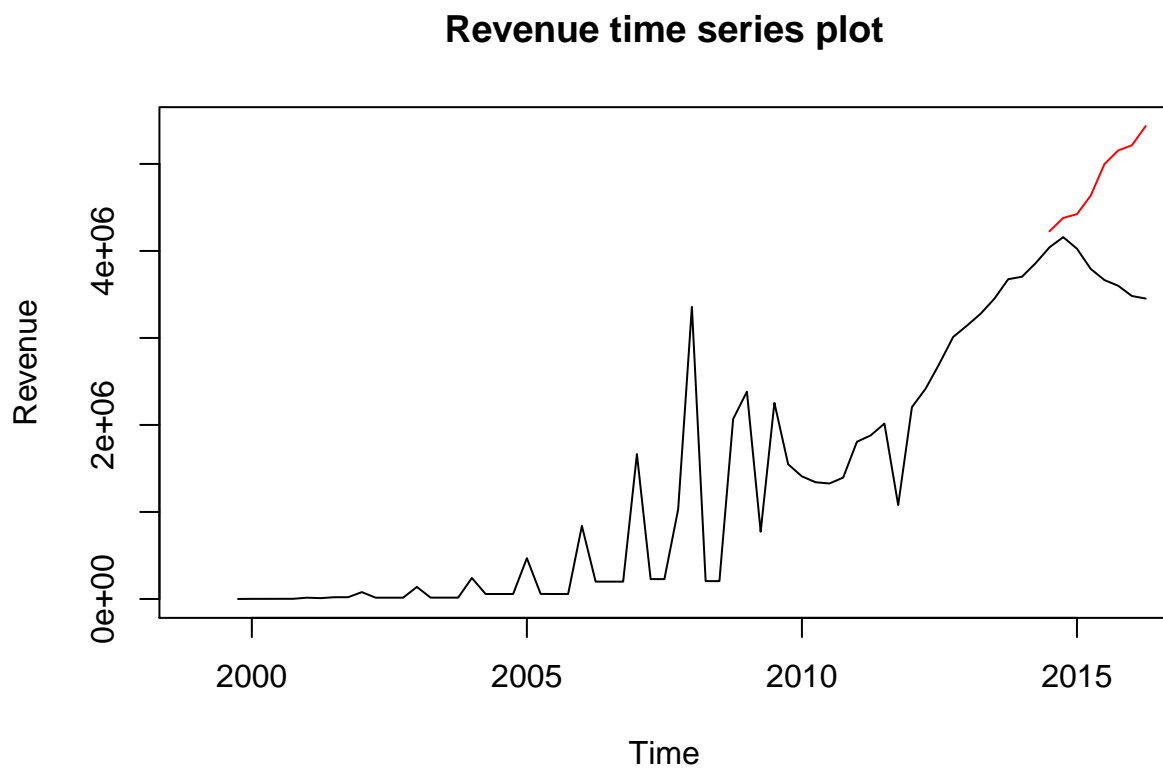
```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -5.1736, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 0.0021421, df = 1, p-value = 0.9631
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
rv.predict <- arima.model.predict(best.rv.model, rv, "Revenue")
```



2.3.5 Валовая прибыль

Найдем лучшую модель для валовой прибыли.

```
best.gp.model <- best.arima.model(log(gp[1:59]), d=1)
```

```
##
## Call:
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),
## 2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),
## 4]), period = 4))
```

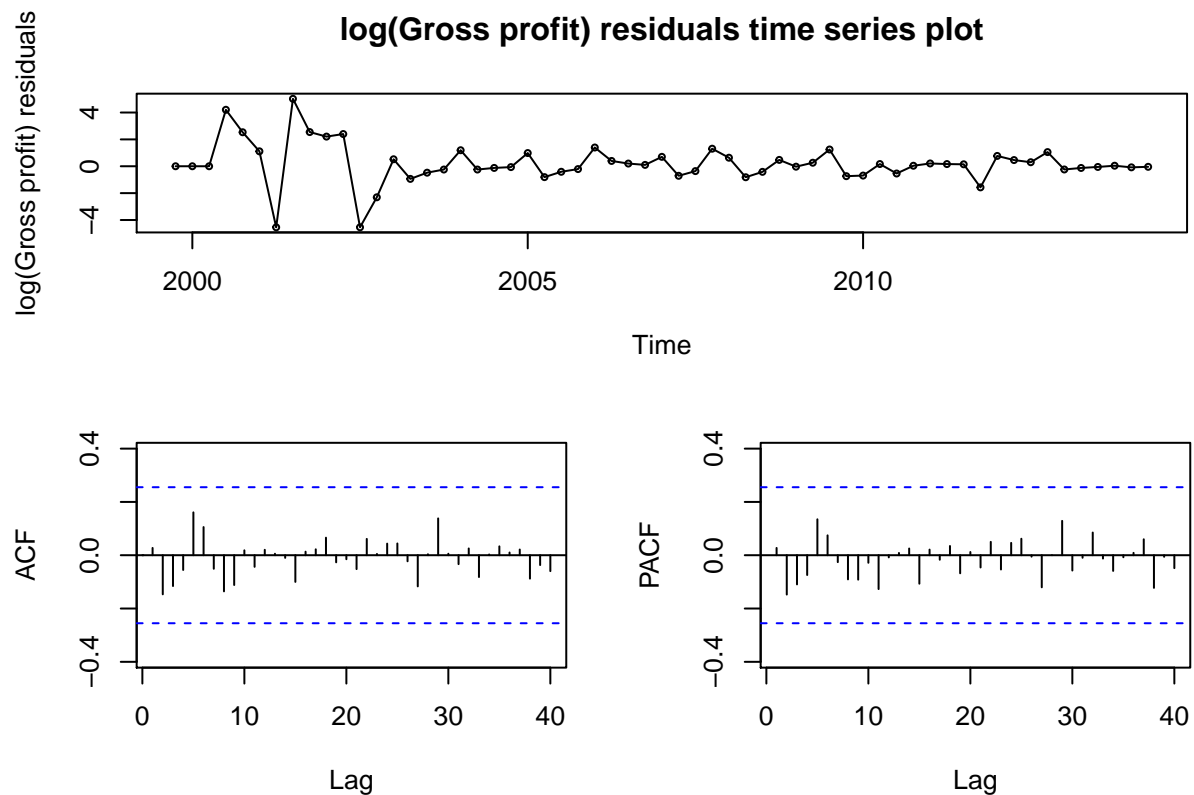
```
##
## Coefficients:
##      ma1    sar1
##    -0.5082 0.5642
## s.e. 0.1195 0.1148
##
## sigma^2 estimated as 2.295: log likelihood = -107.26, aic = 220.53
## Best parameters: 0 1 1 0
```

Проанализируем остатки.

```
arima.model.analysis(best.gp.model, log(gp[1:59]), "log(Gross profit)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = 0.99263, df = 58, p-value = 0.325
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.1973357 0.5855666
## sample estimates:
## mean of x
## 0.1941155

## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```

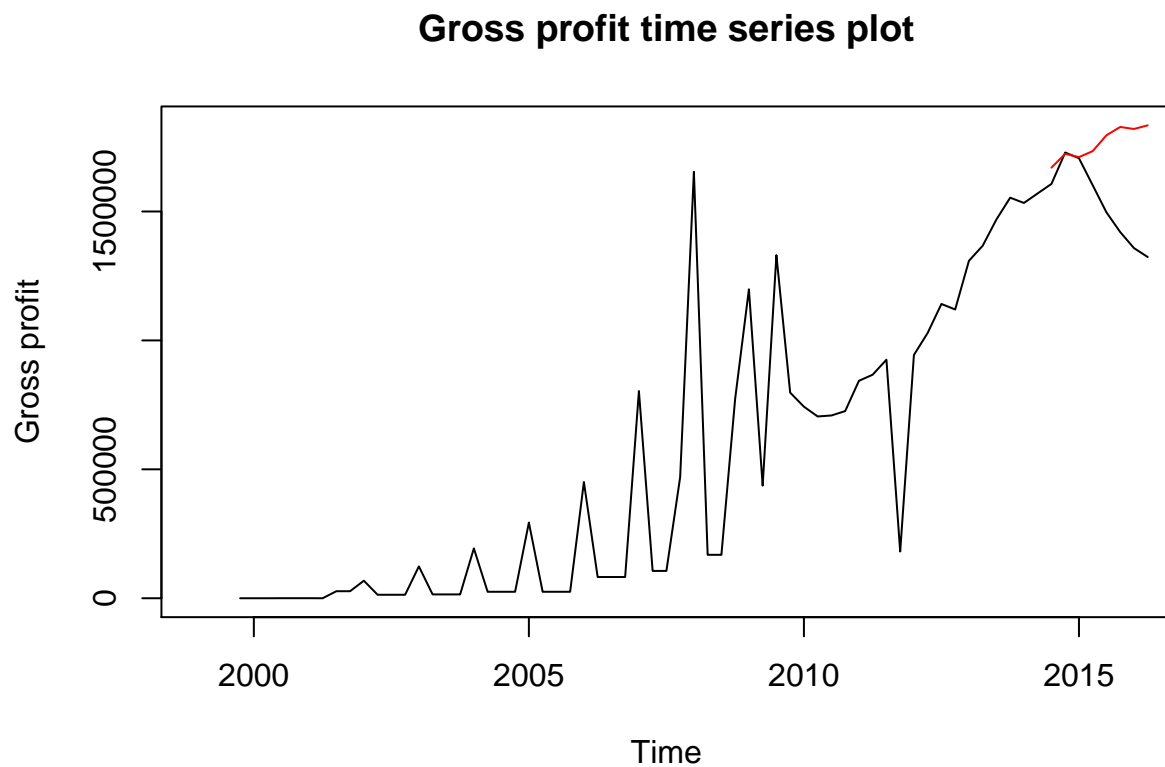


```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -4.7068, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 0.044161, df = 1, p-value = 0.8336
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
gp.predict <- arima.model.predict(best.gp.model, gp, "Gross profit")
```



2.3.6 Чистая прибыль

Найдем лучшую модель для чистой прибыли.

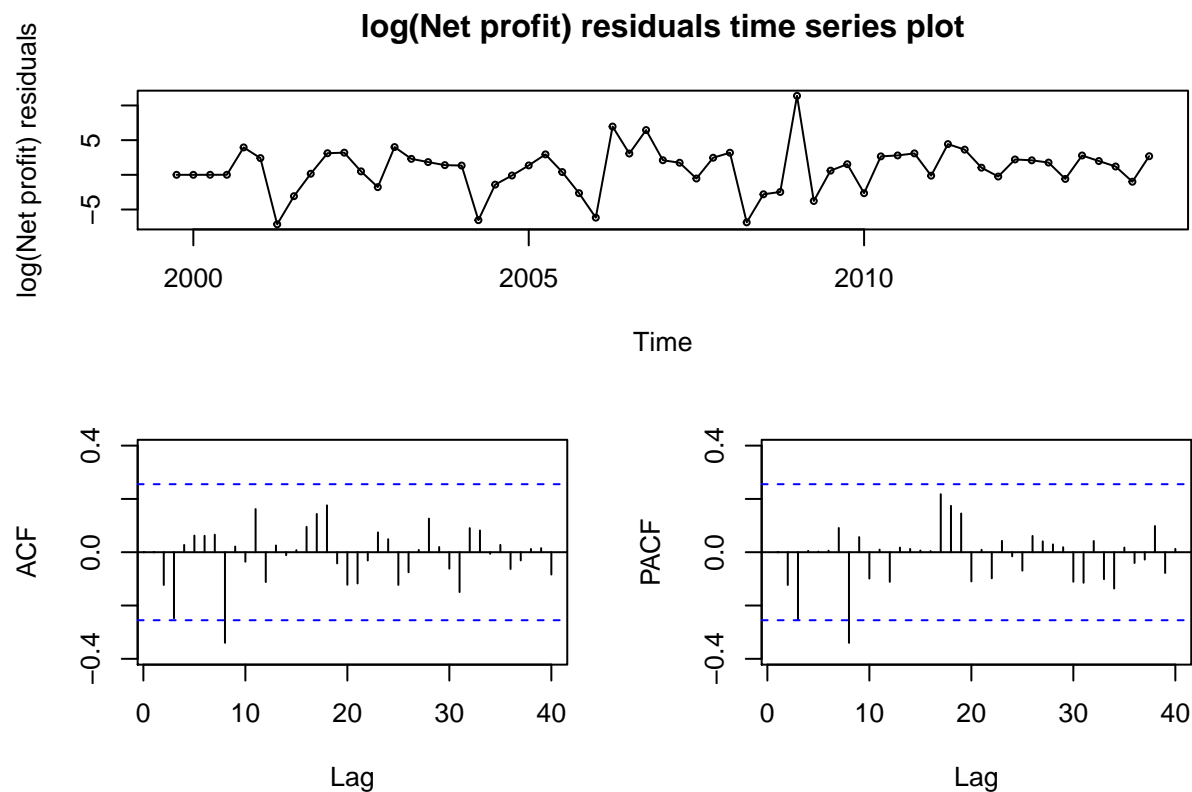
```
best.np.model <- best.arma.model(log(np[1:59]), D=1)
```

```
##  
## Call:  
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),  
## 2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),  
## 4]), period = 4))  
##  
## Coefficients:  
##      ar1    ar2    ar3    ma1    ma2    ma3    sma1  
##    0.9149 0.1736 -0.5498 -0.4327 -0.4441 0.9885 -0.7207  
## s.e. 0.1270 0.2007 0.1387 0.4062 0.2154 0.5803 0.1269  
##  
## sigma^2 estimated as 12.08: log likelihood = -152.45, aic = 320.9  
## Best parameters: 3 3 0 1
```

Проанализируем остатки.

```
arma.model.analysis(best.np.model, log(np[1:59]), "log(Net profit)")
```

```
##  
## One Sample t-test  
##  
## data: model$residuals  
## t = 2.0288, df = 58, p-value = 0.04707  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 0.01155239 1.71625185  
## sample estimates:  
## mean of x  
## 0.8639021  
  
## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```



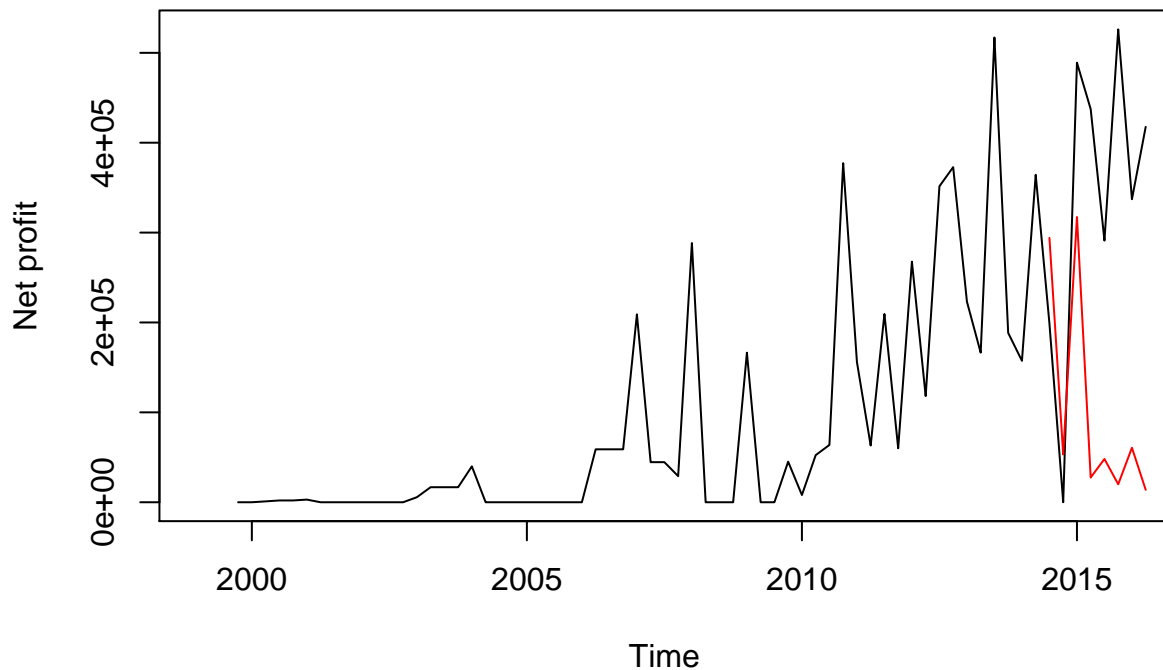
```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -4.7233, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 5.8479e-09, df = 1, p-value = 0.9999
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
np.predict <- arima.model.predict(best.np.model, np, "Net profit")
```

Net profit time series plot



Исходя из анализа остатков, модели построены достаточно хорошие, но видно, что в некоторых местах прогноз сильно расходится с реальностью. Скорее всего, это происходит из-за непредсказуемых изменений переменных: например, в выручке с 2012 по 2015 год был стабильный подъем, который повлиял на прогноз модели, хотя в 2015 году начался спад; в 2015-2016 годах был огромный скачок значения краткосрочных заемных средств, и предсказать его было нельзя. Также не стоит забывать и об обилии пропусков в начальных данных, которые в большинстве своем пришлось заполнять по линейности, что могло сильно повлиять на качество моделей.

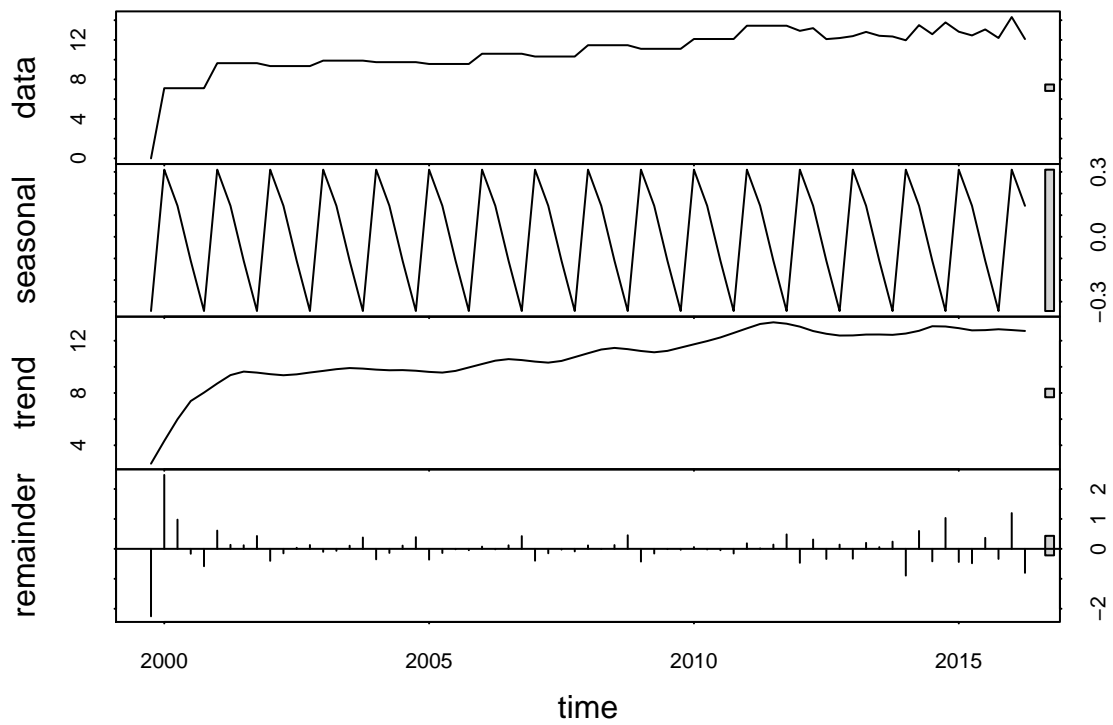
3 Перспективный прогноз

Перспективный прогноз будем делать по той же схеме, что и ретроспективный.

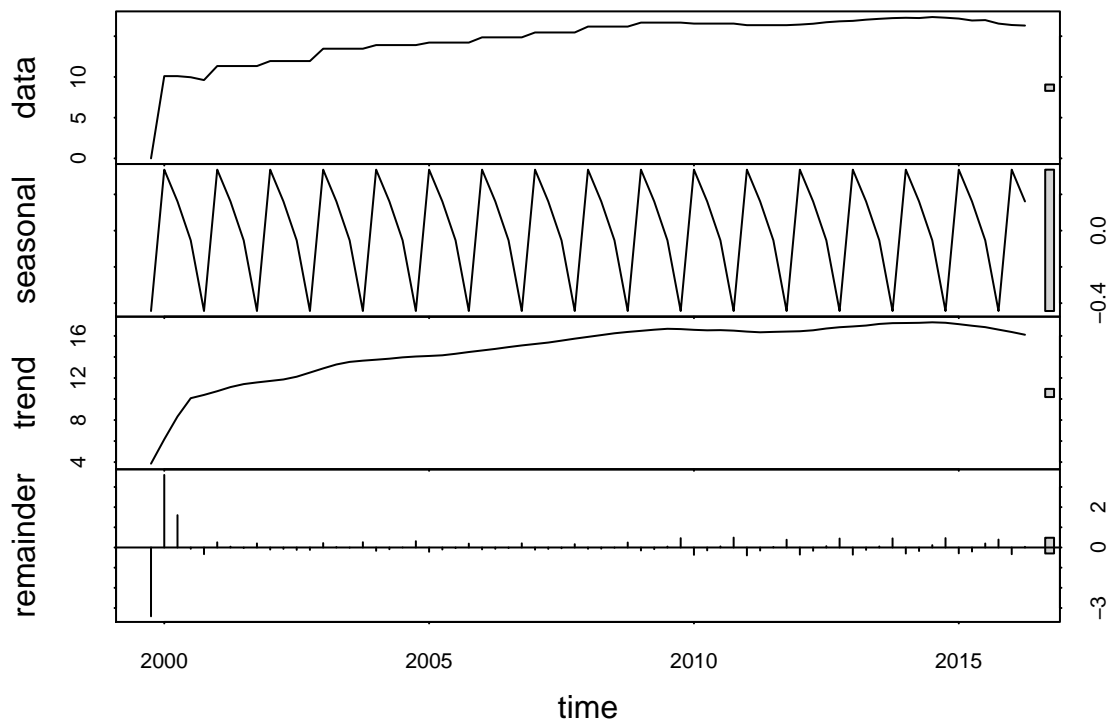
3.1 STL

STL-декомпозиция рядов:

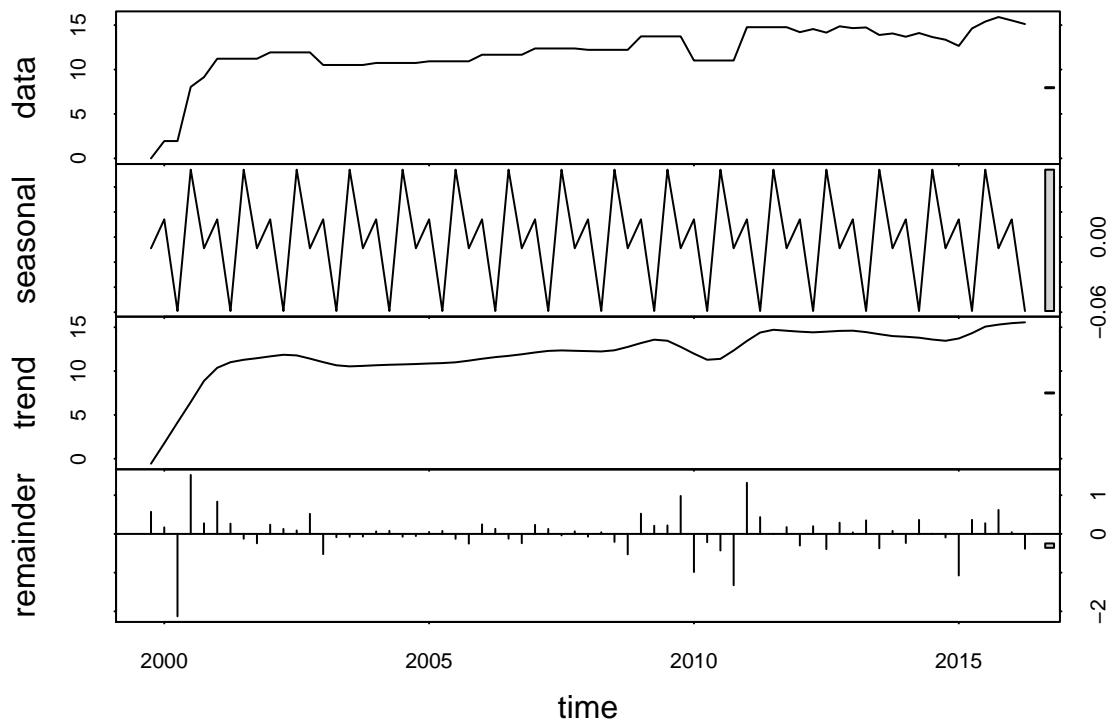
```
plot(stl(ts(log(cash), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



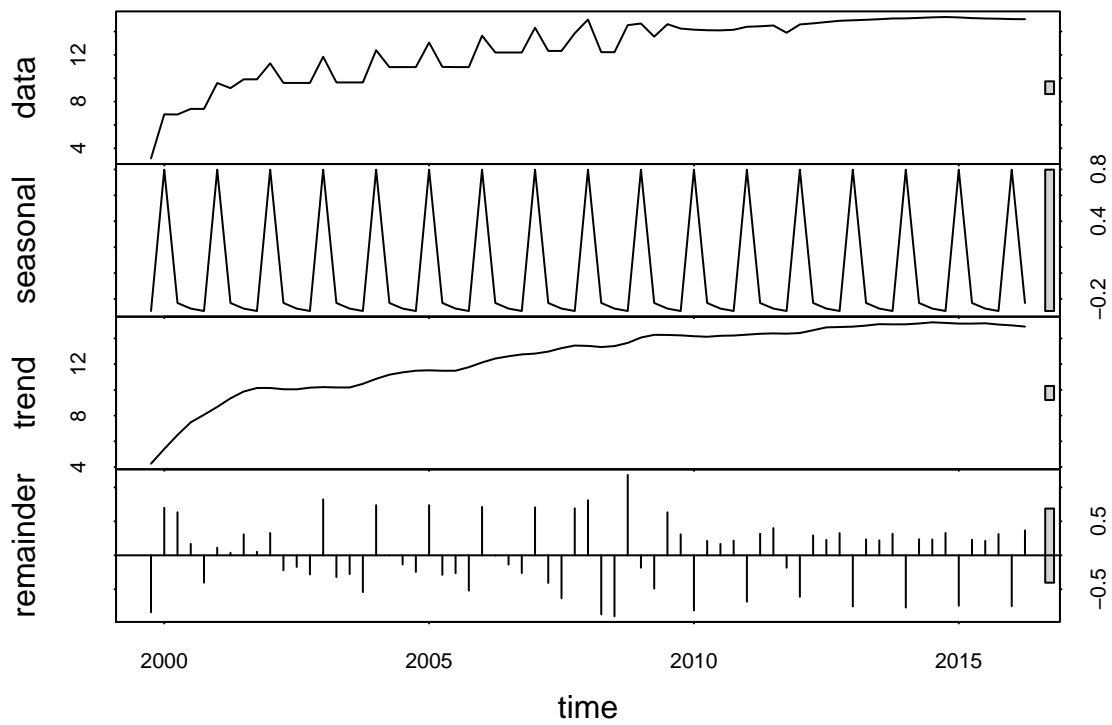
```
plot(stl(ts(log(ltb), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```

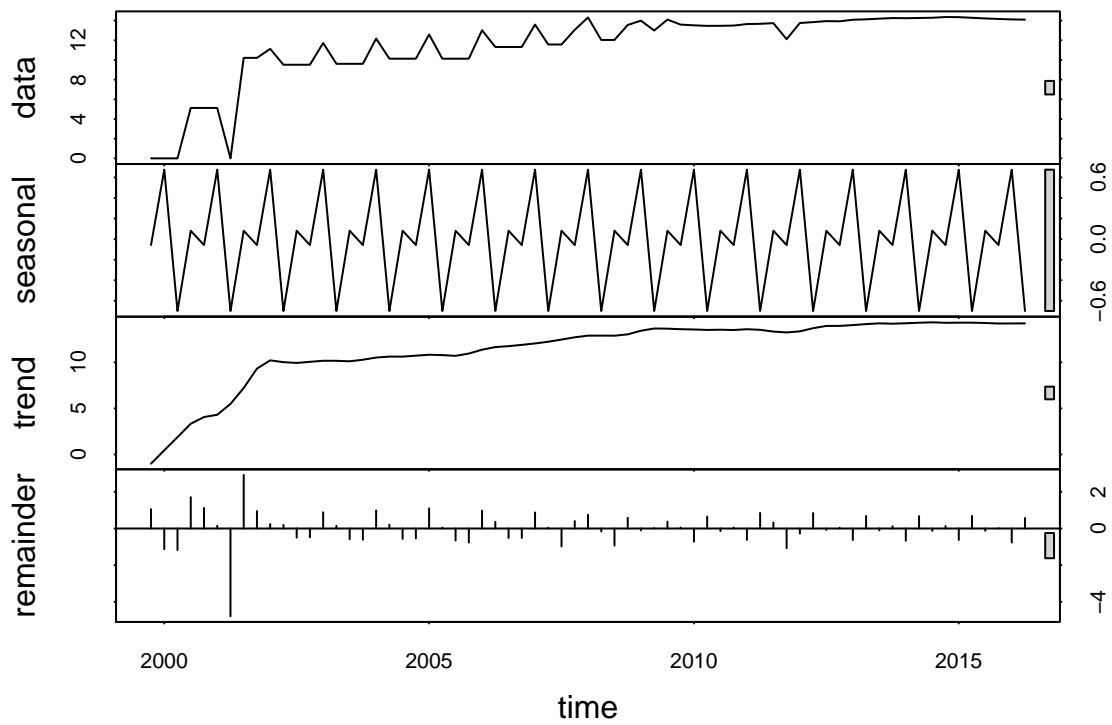
```
plot(stl(ts(log(stb), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



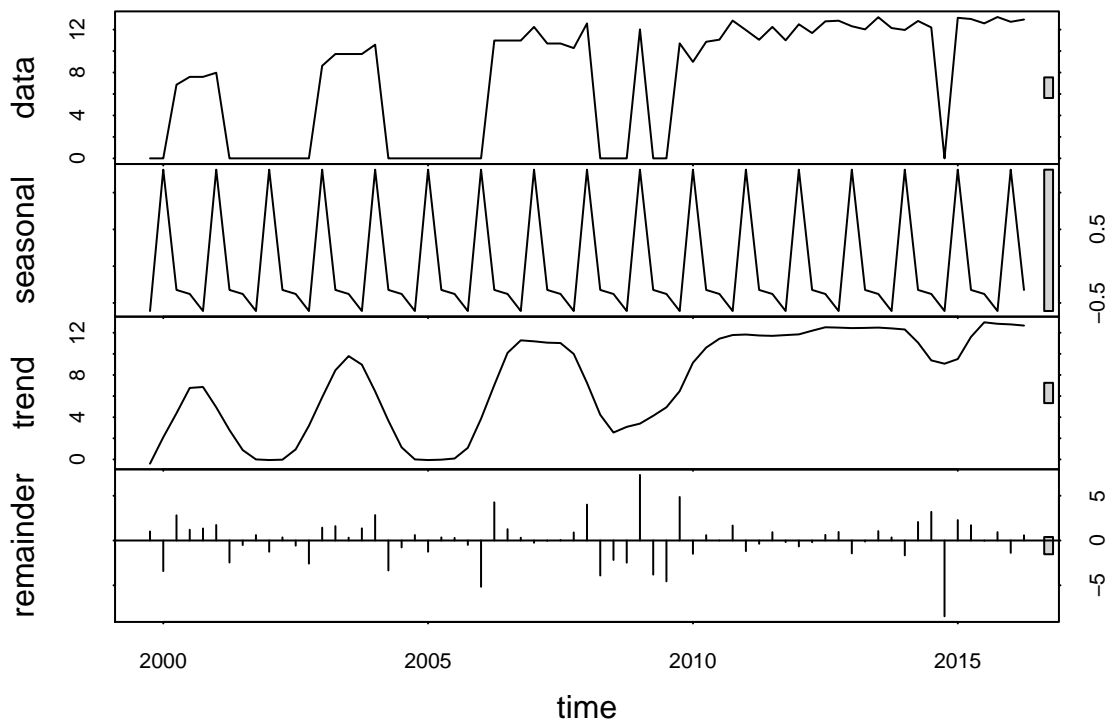
```
plot(stl(ts(log(rv), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



```
plot(stl(ts(log(gp), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



```
plot(stl(ts(log(np), frequency=4, start=c(1999, 4)), s.window="periodic", robust=FALSE))
```



3.2 Дифференцирование и стационарность

Применим сезонное/обычное дифференцирование по необходимости. Проверим стационарность с помощью критерия Дики-Фуллера.

```
persp.cash.diff <- diff(log(cash), 4)
adf.test(persp.cash.diff)
```

```
## Warning in adf.test(persp.cash.diff): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: persp.cash.diff
## Dickey-Fuller = -5.0429, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
persp.ltb.diff <- diff(log(ltb), 4)
persp.ltb.diff <- diff(persp.ltb.diff, 2)
adf.test(persp.ltb.diff)
```

```
## Warning in adf.test(persp.ltb.diff): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: persp.ltb.diff
## Dickey-Fuller = -4.4703, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(log(stb))
```

```
## Warning in adf.test(log(stb)): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: log(stb)
## Dickey-Fuller = -4.1364, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
persp.rv.diff <- diff(log(rv), 4)
adf.test(persp.rv.diff)
```

```
## Warning in adf.test(persp.rv.diff): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: persp.rv.diff
## Dickey-Fuller = -4.6463, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
persp.gp.diff <- diff(log(gp), 2)
adf.test(persp.gp.diff)
```

```
## Warning in adf.test(persp.gp.diff): p-value smaller than printed p-value
```

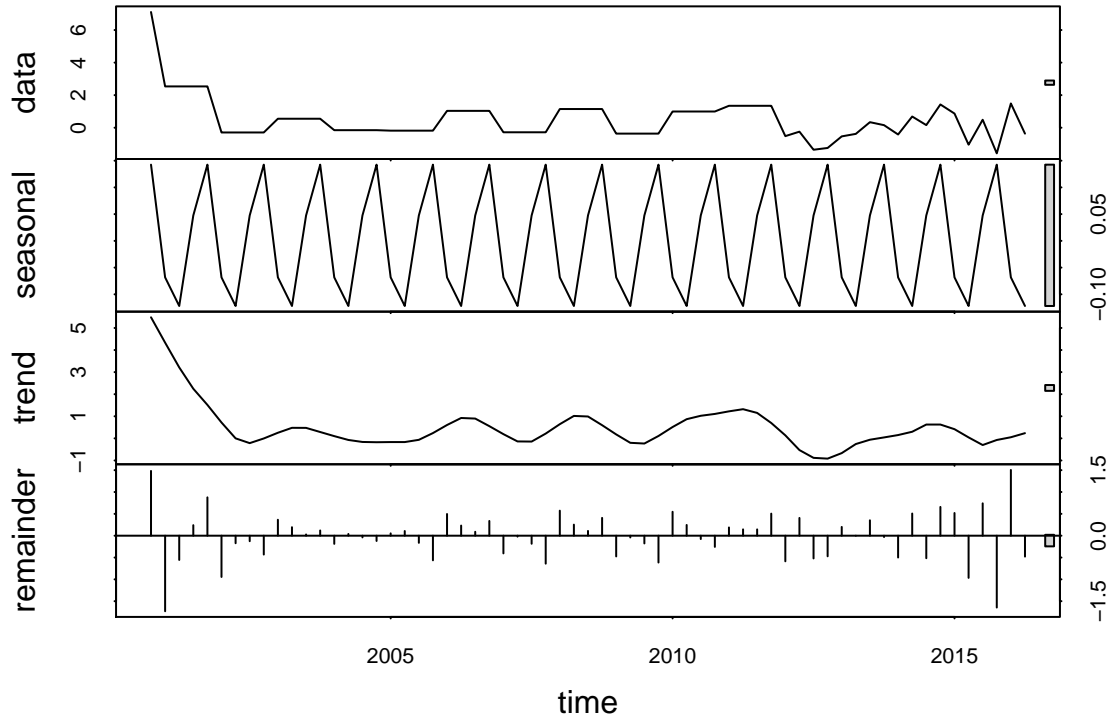
```
##
## Augmented Dickey-Fuller Test
##
## data: persp.gp.diff
## Dickey-Fuller = -4.5982, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(log(np))
```

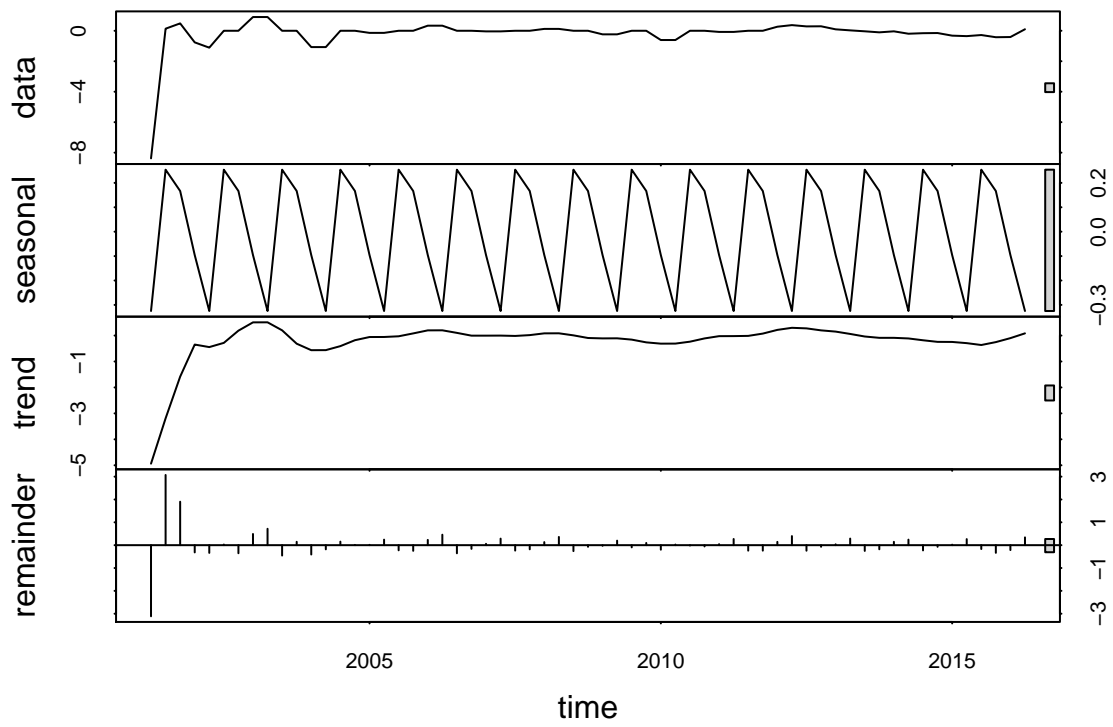
```
## Warning in adf.test(log(np)): p-value smaller than printed p-value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: log(np)
## Dickey-Fuller = -4.4729, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

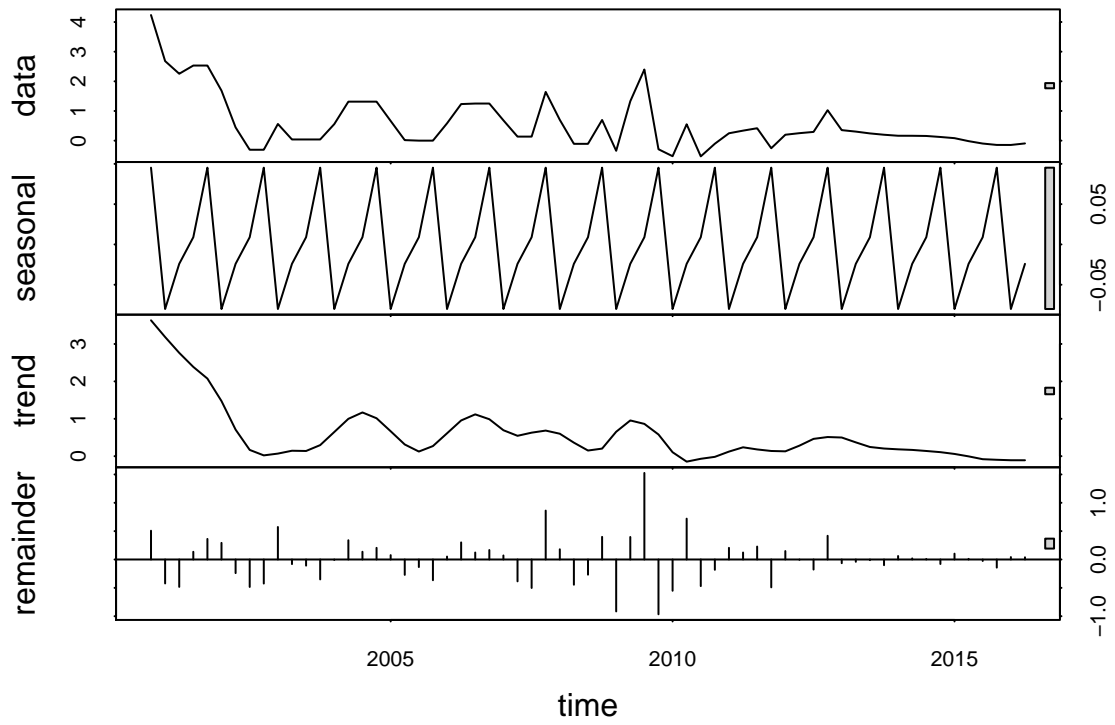
```
plot(stl(ts(persp.cash.diff, frequency=4, start=c(2000, 4)), s.window="periodic", robust=FALSE))
```



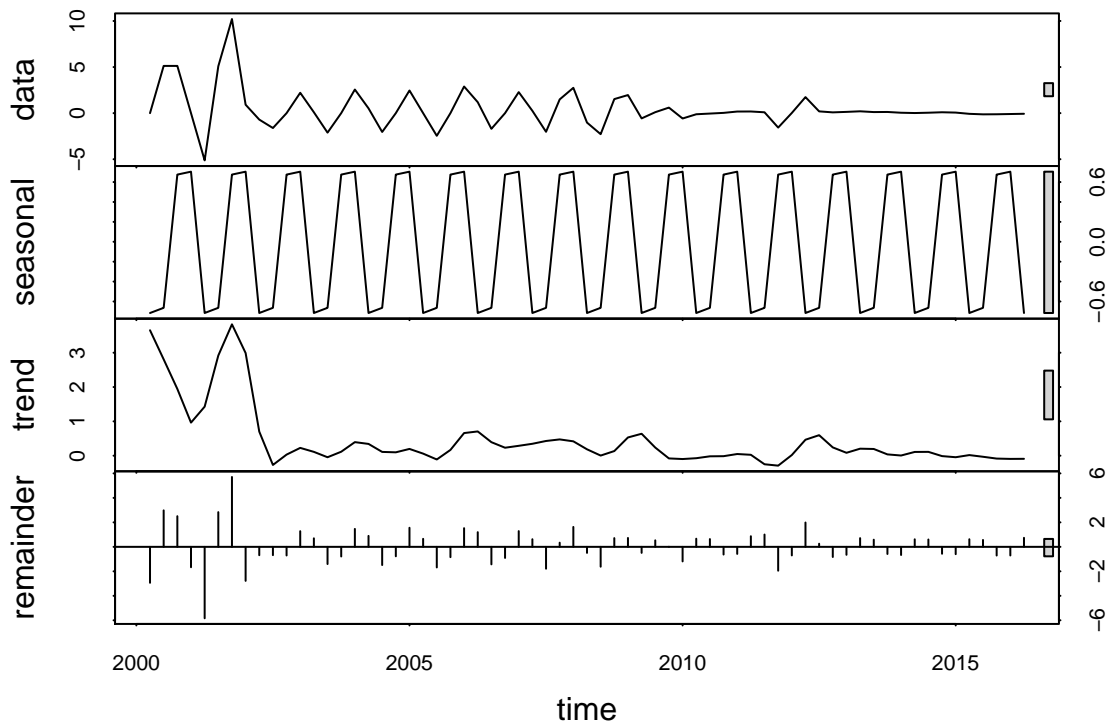
```
plot(stl(ts(persp.ltb.diff, frequency=4, start=c(2001, 2)), s.window="periodic", robust=FALSE))
```



```
plot(stl(ts(persp.rv.diff, frequency=4, start=c(2000, 4)), s.window="periodic", robust=FALSE))
```

```
plot(stl(ts(persp.gp.diff, frequency=4, start=c(2000, 2)), s.window="periodic", robust=FALSE))
```



3.3 Выбор лучшей модели, анализ остатков и прогноз

Для поиска лучшей ARIMA-модели, анализа ее остатков и предсказания будем использовать написанные ранее функции.

3.3.1 Денежные средства и денежные эквиваленты

Найдем лучшую модель для денежных средств и денежных эквивалентов.

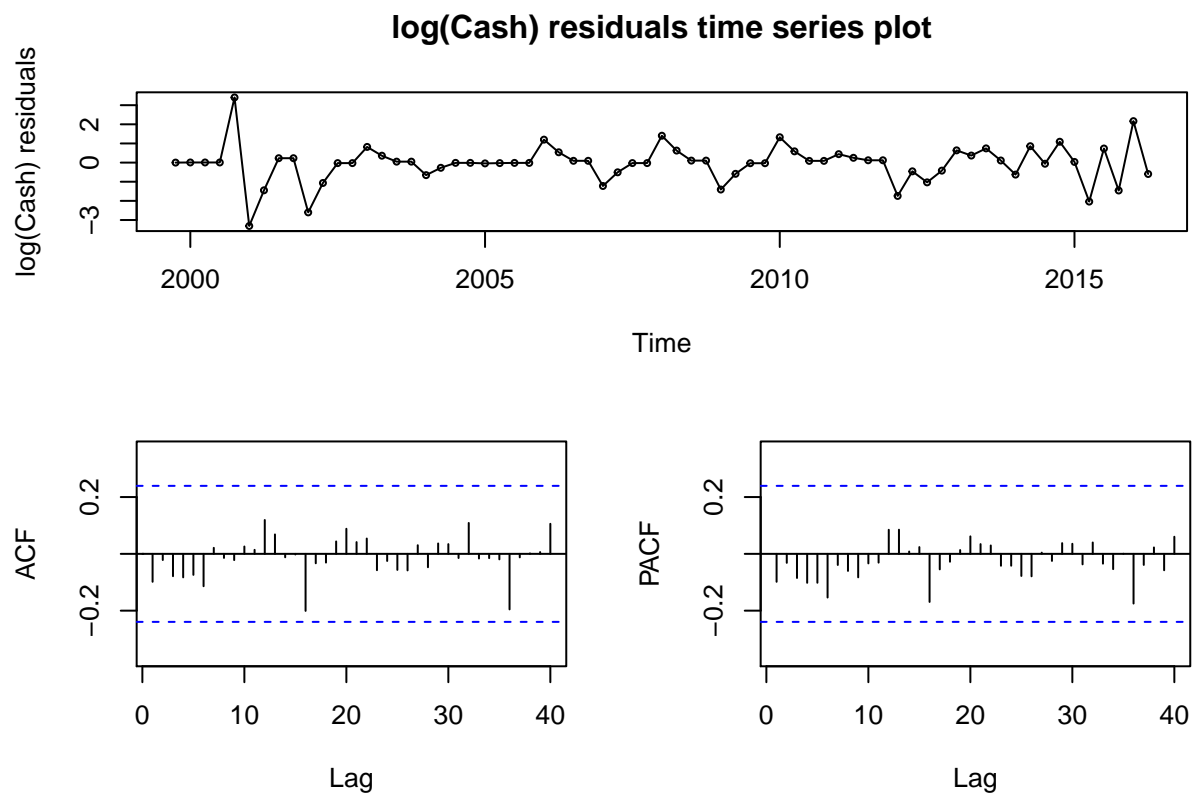
```
best.persp.cash.model <- best.arma.model(log(cash), D=1)
```

```
##
## Call:
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),
## 2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),
## 4]), period = 4))
##
## Coefficients:
##      ar1      ar2
##    0.5430  0.3667
## s.e. 0.1485  0.1536
##
## sigma^2 estimated as 1.022: log likelihood = -90.88, aic = 187.75
## Best parameters: 2 0 0 0
```

Проанализируем остатки.

```
arima.model.analysis(best.persp.cash.model, log(cash), "log(Cash)")
```

```
##  
## One Sample t-test  
##  
## data: model$residuals  
## t = -0.33652, df = 66, p-value = 0.7375  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.2812303 0.2001028  
## sample estimates:  
## mean of x  
## -0.04056376  
  
## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```



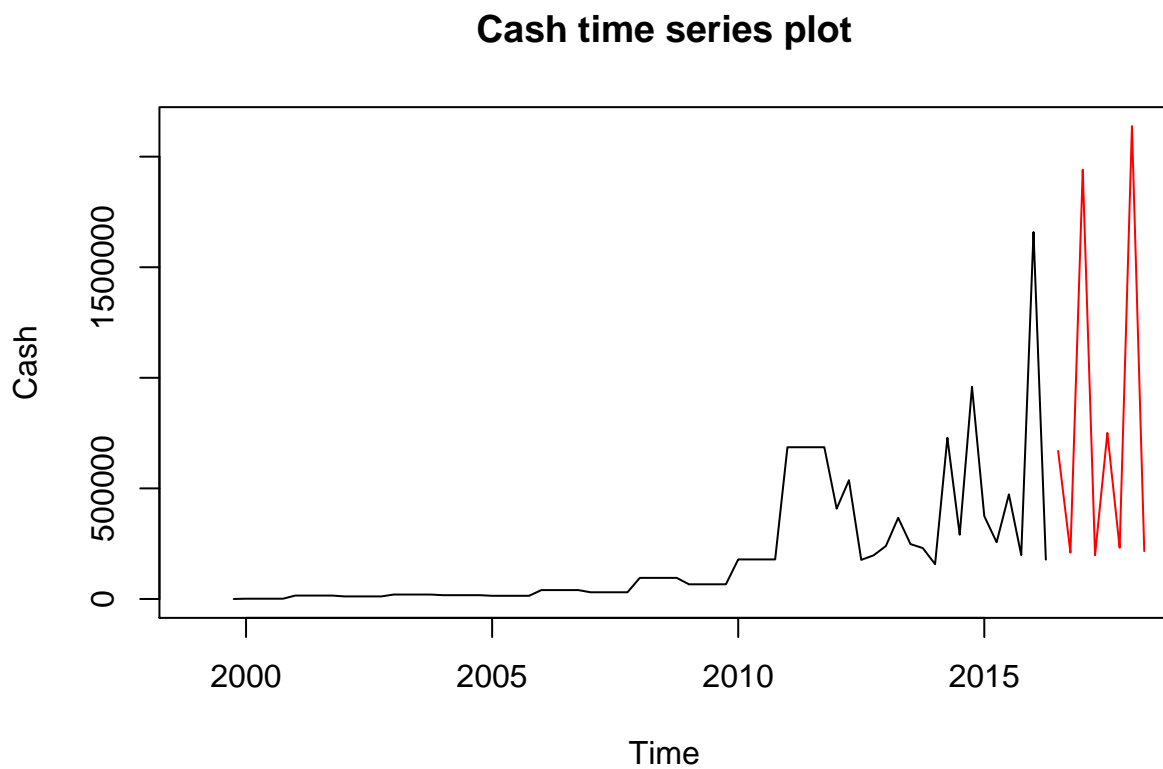
```
##  
## Augmented Dickey-Fuller Test  
##  
## data: model$residuals  
## Dickey-Fuller = -5.3289, Lag order = 4, p-value = 0.01  
## alternative hypothesis: stationary
```

```
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 0.6478, df = 1, p-value = 0.4209
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
persp.cash.predict <- arima.model.predict(best.persp.cash.model, cash, "Cash", start=c(2016, 3))
```



3.3.2 Долгосрочные заемные средства

Найдем лучшую модель для долгосрочных заемных средств.

```
best.persp.ltb.model <- best.arima.model(log(ltb), d=2, D=1)
```

```
## Wrong parameters: 1 1 0 0
##
## Call:
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),
## 2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),
```

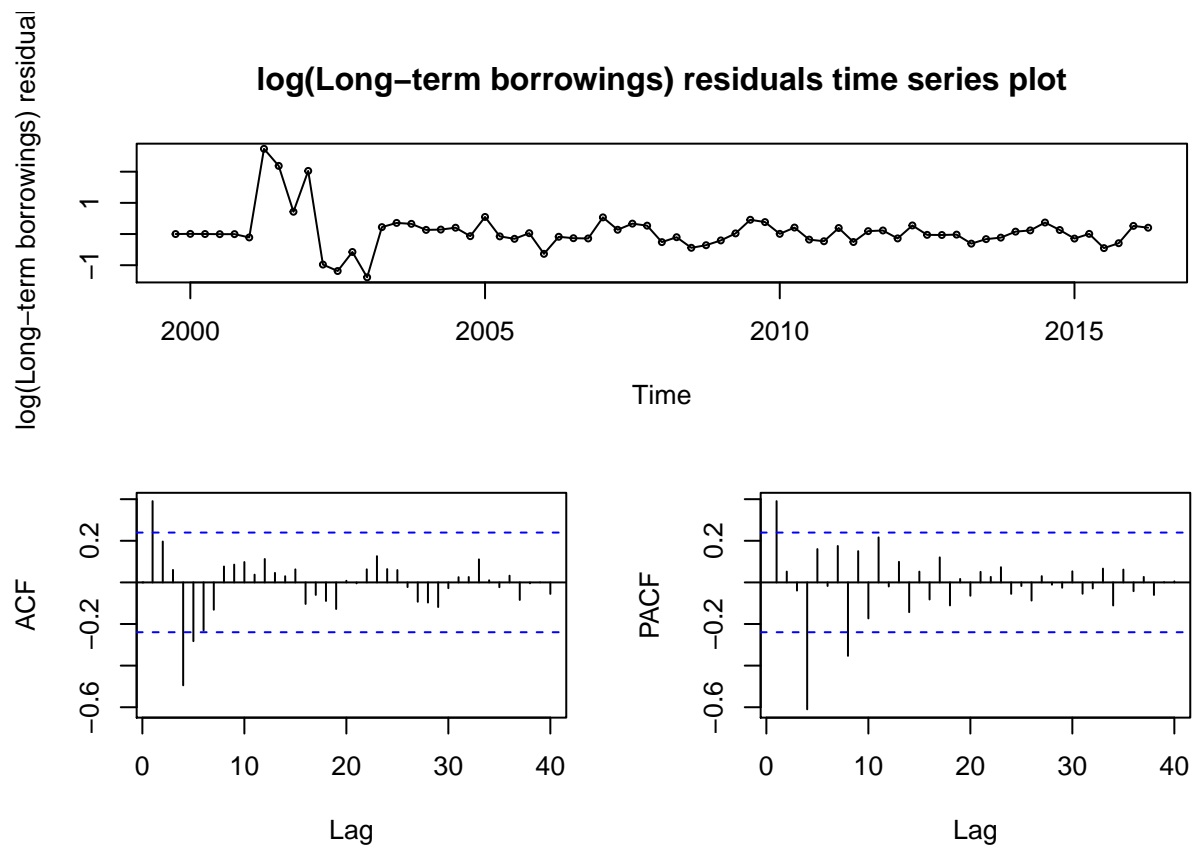
```
## 4]), period = 4))
##
## Coefficients:
##      ar1      ar2      ma1      ma2      ma3      ma4      sma1
##    -0.6954 -0.1249 -0.0624 -0.3729 -0.2068  0.8892  0.9917
## s.e.  0.2474  0.1910  0.1564  0.1262  0.1318  0.1321  0.1284
##
## sigma^2 estimated as 0.4102: log likelihood = -76.38, aic = 168.77
## Best parameters: 2 4 0 1
```

Проанализируем остатки.

```
arima.model.analysis(best.persp.ltb.model, log(ltb), "log(Long-term borrowings)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = 0.90574, df = 66, p-value = 0.3684
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.08155177  0.21698101
## sample estimates:
## mean of x
## 0.06771462

## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```



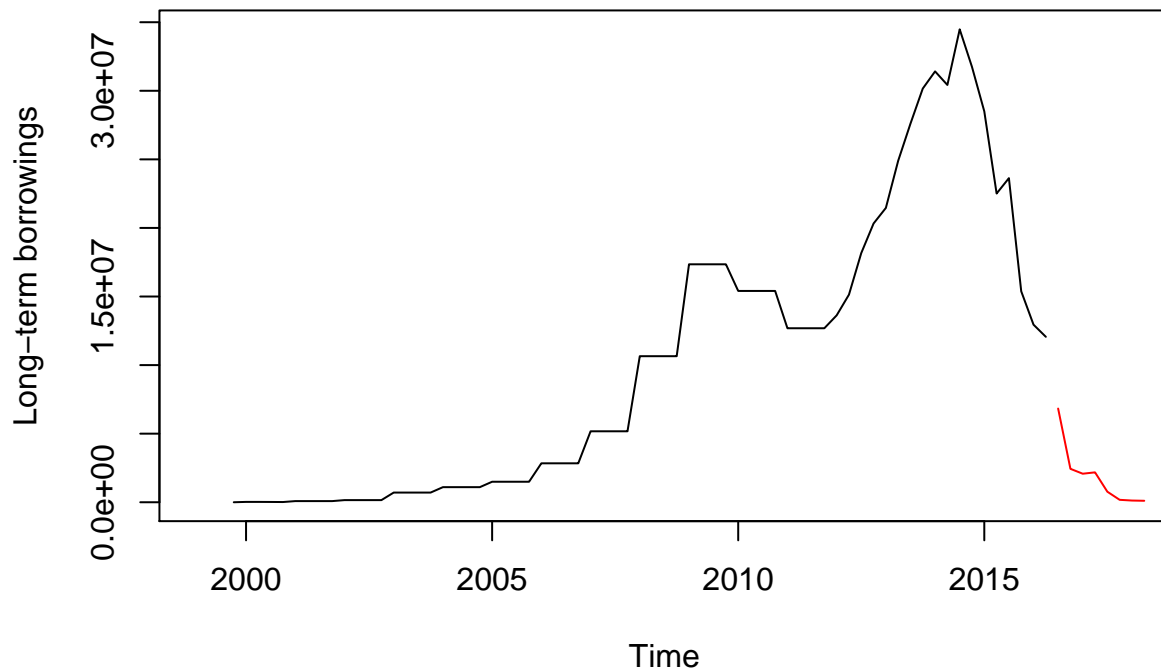
```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -4.6462, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 10.223, df = 1, p-value = 0.001387
```

Остатки несмещены и стационарны. Несмотря на то, что критерий Бокса-Пирса указывает на возможный неучет всех особенностей данных (то есть, наверное, можно найти модель чуть лучше), из коррелограмм видно, что остатки в целом не так уж сильно автокоррелированы. Поэтому можно не заморачиваться по поводу идеального качества остатков и остановиться на данной модели.

Сделаем прогноз.

```
persp.ltb.predict <- arima.model.predict(best.persp.ltb.model, ltb, "Long-term borrowings",
                                          start=c(2016, 3))
```

Long-term borrowings time series plot



3.3.3 Краткосрочные заемные средства

Найдем лучшую модель для краткосрочных заемных средств.

```
best.persp.stb.model <- best.arima.model(log(stb))
```

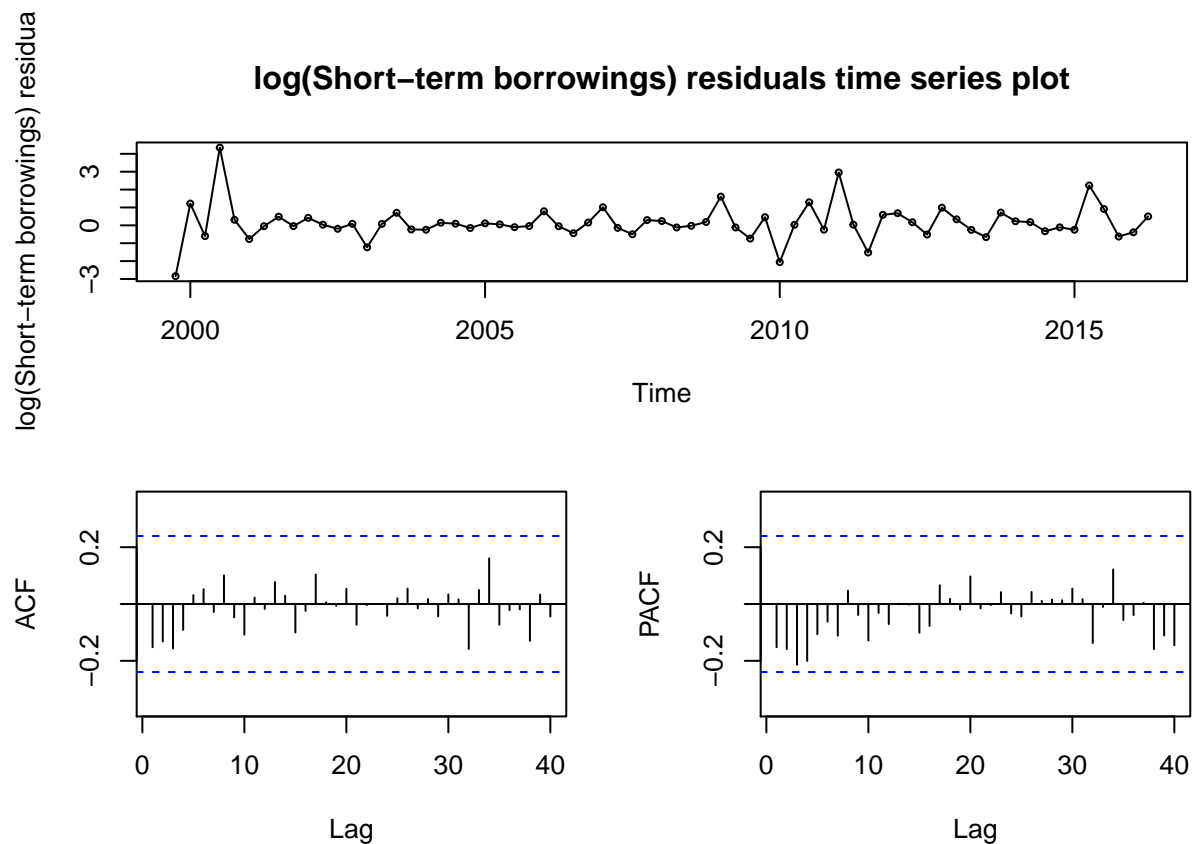
```
##  
## Call:  
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),  
## 2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),  
## 4]), period = 4))  
##  
## Coefficients:  
##      ar1    ma1    ma2 intercept  
##    0.8995 0.1628 0.6358 11.1405  
## s.e. 0.0717 0.1050 0.1208 1.9681  
##  
## sigma^2 estimated as 0.9689: log likelihood = -95.87, aic = 201.74  
## Best parameters: 1 2 0 0
```

Проанализируем остатки.

```
arima.model.analysis(best.persp.stb.model, log(stb), "log(Short-term borrowings)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = 1.1348, df = 66, p-value = 0.2606
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.1034096 0.3757523
## sample estimates:
## mean of x
## 0.1361713

## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```

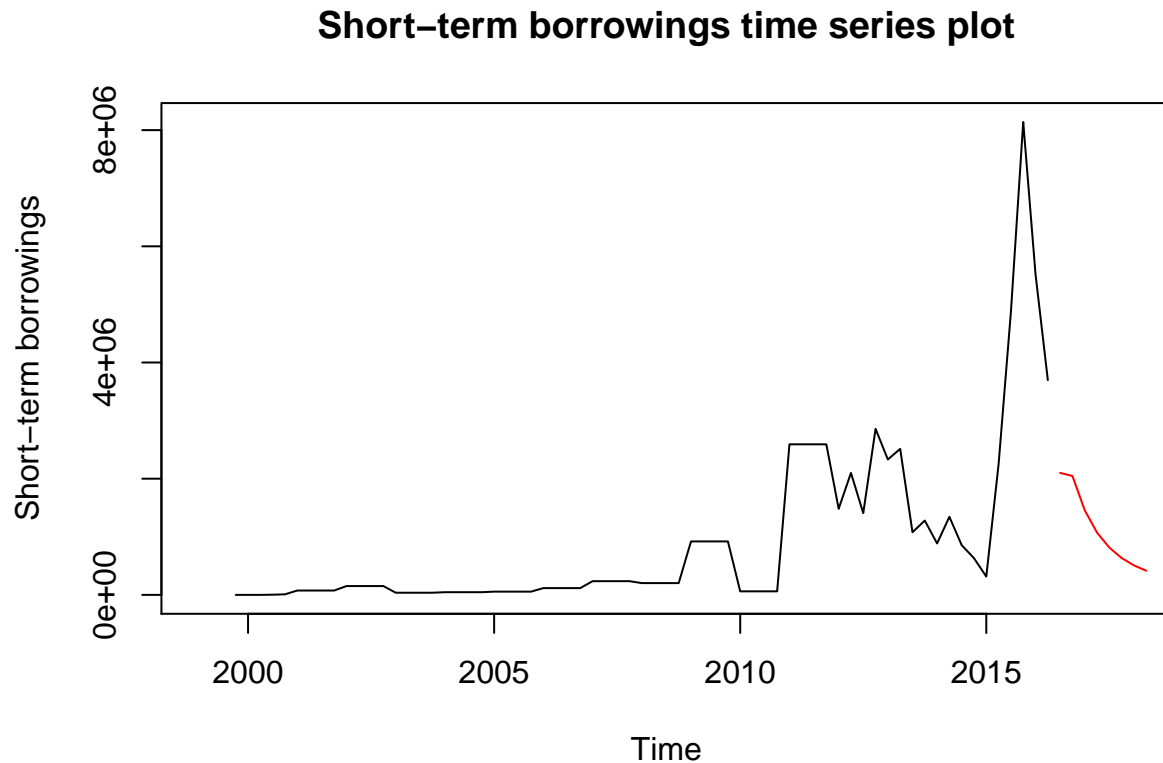


```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -5.3425, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 1.5513, df = 1, p-value = 0.2129
```


Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
persp.stb.predict <- arima.model.predict(best.persp.stb.model, stb, "Short-term borrowings",  
                                         start=c(2016, 3))
```



3.3.4 Выручка

Найдем лучшую модель для выручки.

```
best.persp.rv.model <- best.arima.model(log(rv), D=1)
```

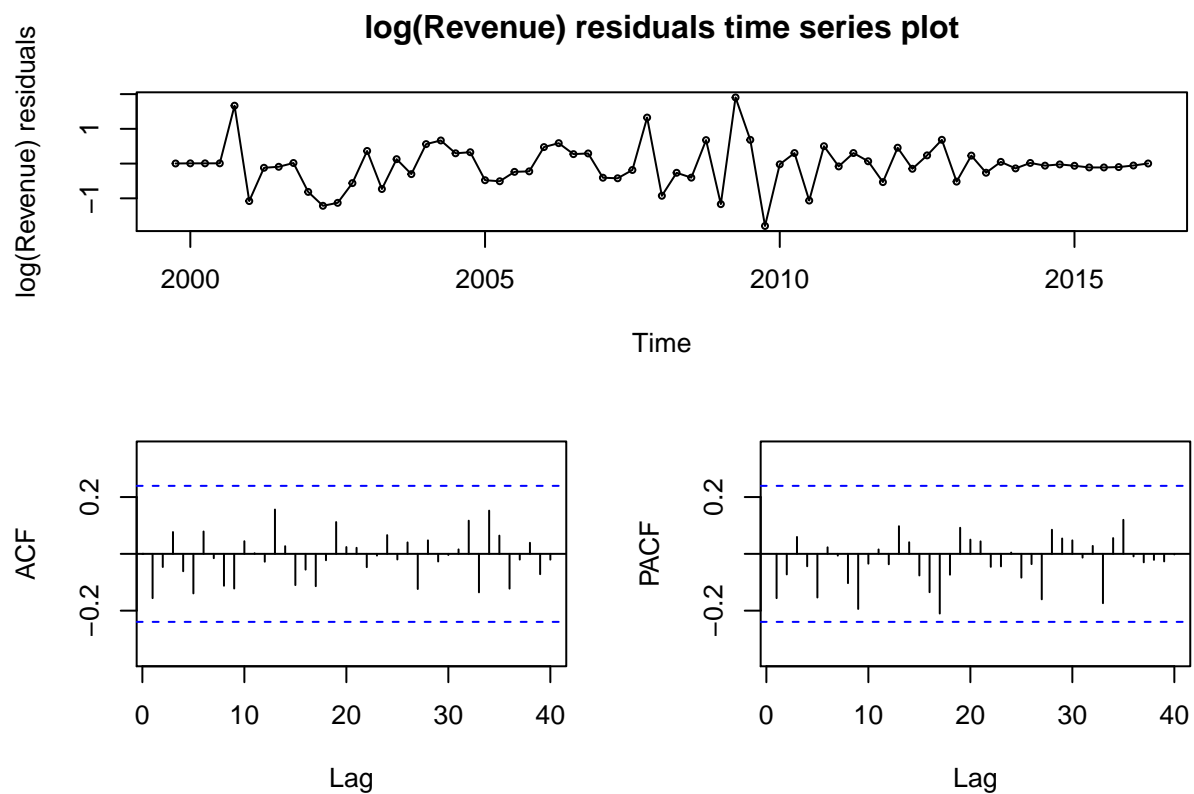
```
##  
## Call:  
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),  
##    2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),  
##    4]), period = 4))  
##  
## Coefficients:  
##      ar1      ma1      ma2  
##    0.9735 -0.0372 -0.4224  
## s.e. 0.0391  0.1356  0.1315  
##  
## sigma^2 estimated as 0.4158: log likelihood = -62.85, aic = 133.7  
## Best parameters: 1 2 0 0
```

Проанализируем остатки.

```
arima.model.analysis(best.persp.rv.model, log(rv), "log(Revenue)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = -0.63212, df = 66, p-value = 0.5295
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.2017076 0.1046987
## sample estimates:
## mean of x
## -0.04850449

## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```



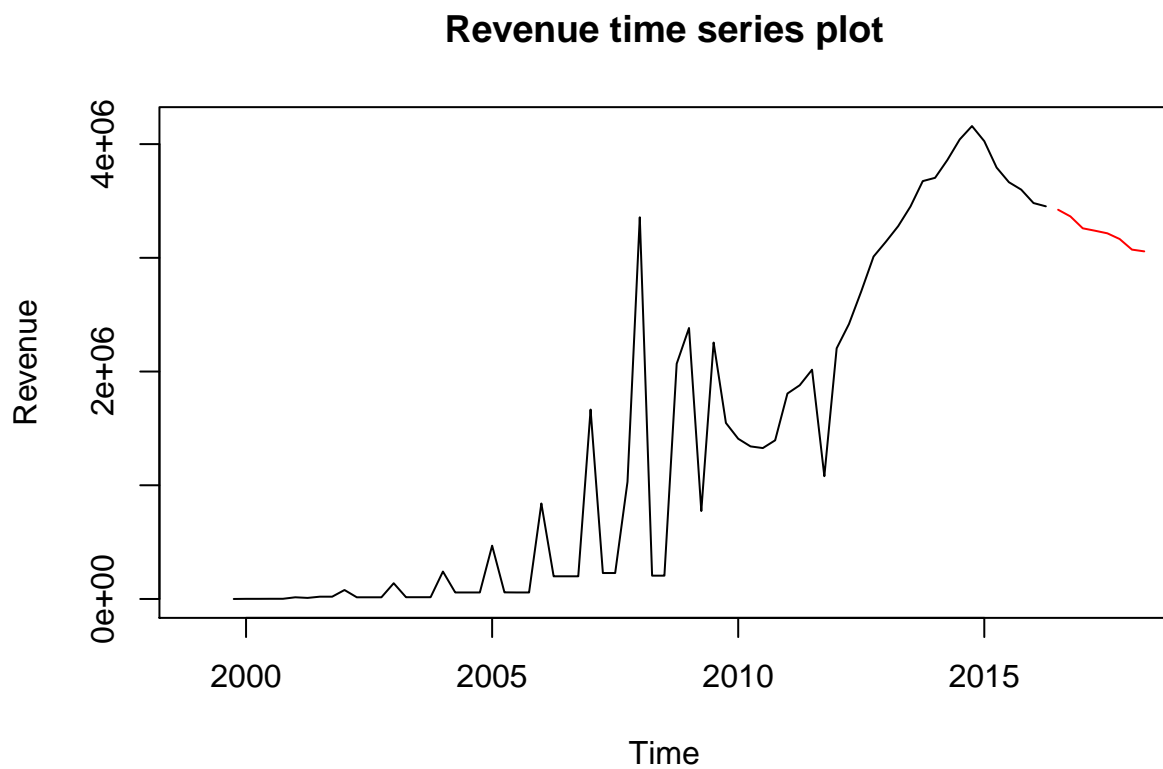
```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -4.7093, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 1.629, df = 1, p-value = 0.2018
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
persp.rv.predict <- arima.model.predict(best.persp.rv.model, rv, "Revenue", start=c(2016, 3))
```



3.3.5 Валовая прибыль

Найдем лучшую модель для валовой прибыли.

```
best.persp.gp.model <- best.arima.model(log(gp), d=2)
```

```
##
## Call:
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),
## 2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),
## 4]), period = 4))
```

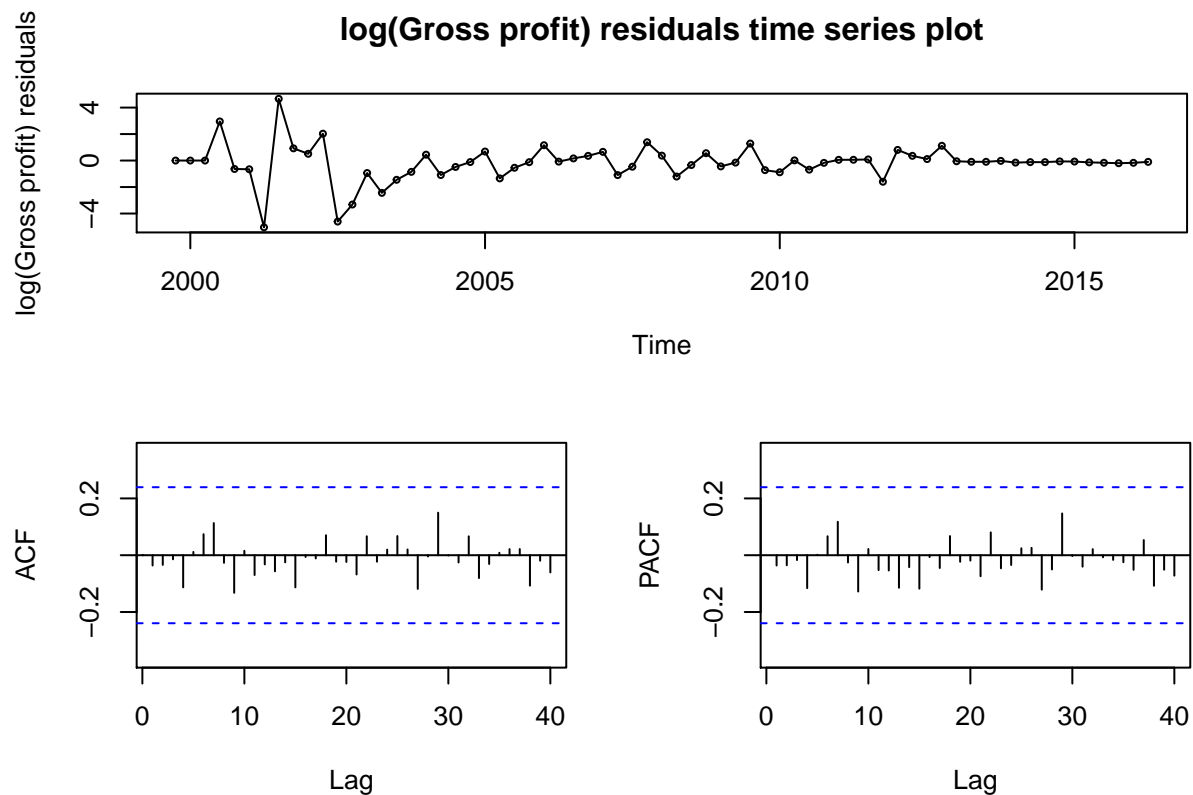
```
##
## Coefficients:
##      ar1      ar2      ar3      ma1
##    -0.7098 -0.6158 -0.6407 -0.7630
## s.e.  0.1017  0.1121  0.0980  0.0936
##
## sigma^2 estimated as 1.905: log likelihood = -115.36, aic = 240.72
## Best parameters: 3 1 0 0
```

Проанализируем остатки.

```
arima.model.analysis(best.persp.gp.model, log(gp), "log(Gross profit)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = -1.1111, df = 66, p-value = 0.2705
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.5152280 0.1467939
## sample estimates:
## mean of x
## -0.1842171

## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```

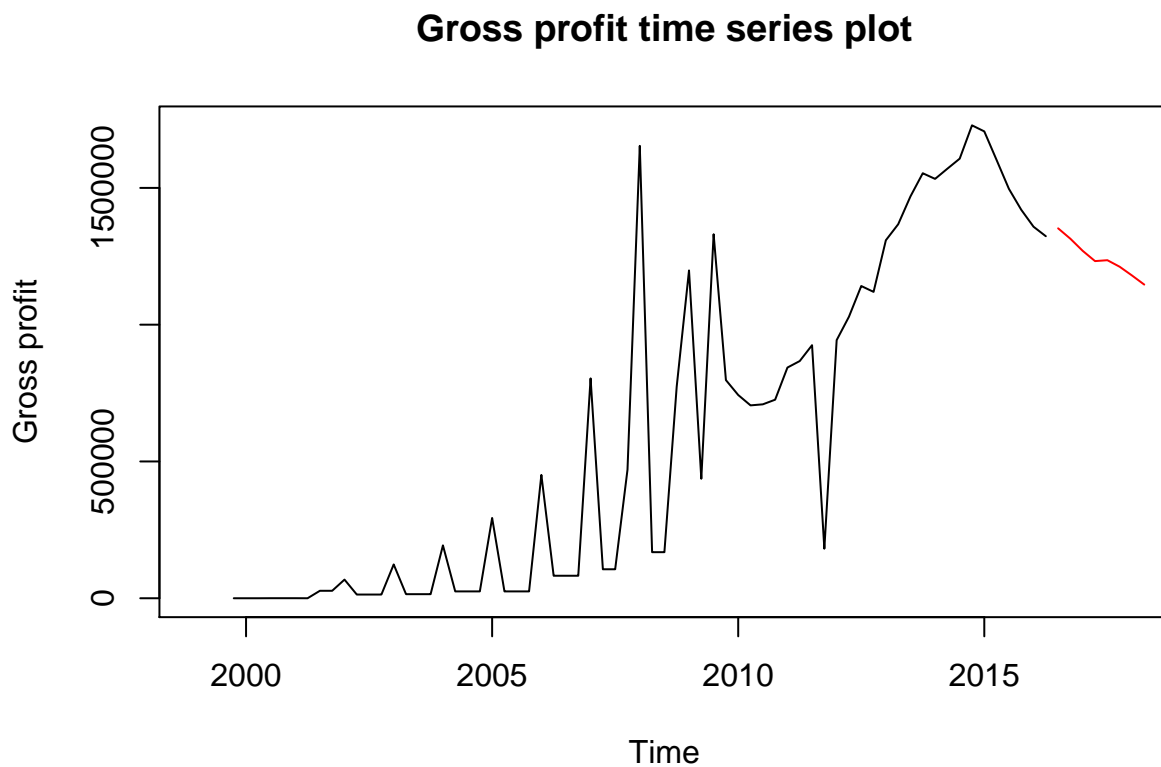


```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -4.216, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 0.087246, df = 1, p-value = 0.7677
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
persp.gp.predict <- arima.model.predict(best.persp.gp.model, gp, "Gross profit", start=c(2016, 3))
```



3.3.6 Чистая прибыль

Найдем лучшую модель для чистой прибыли.

```
best.persp.np.model <- best.arima.model(log(np))
```

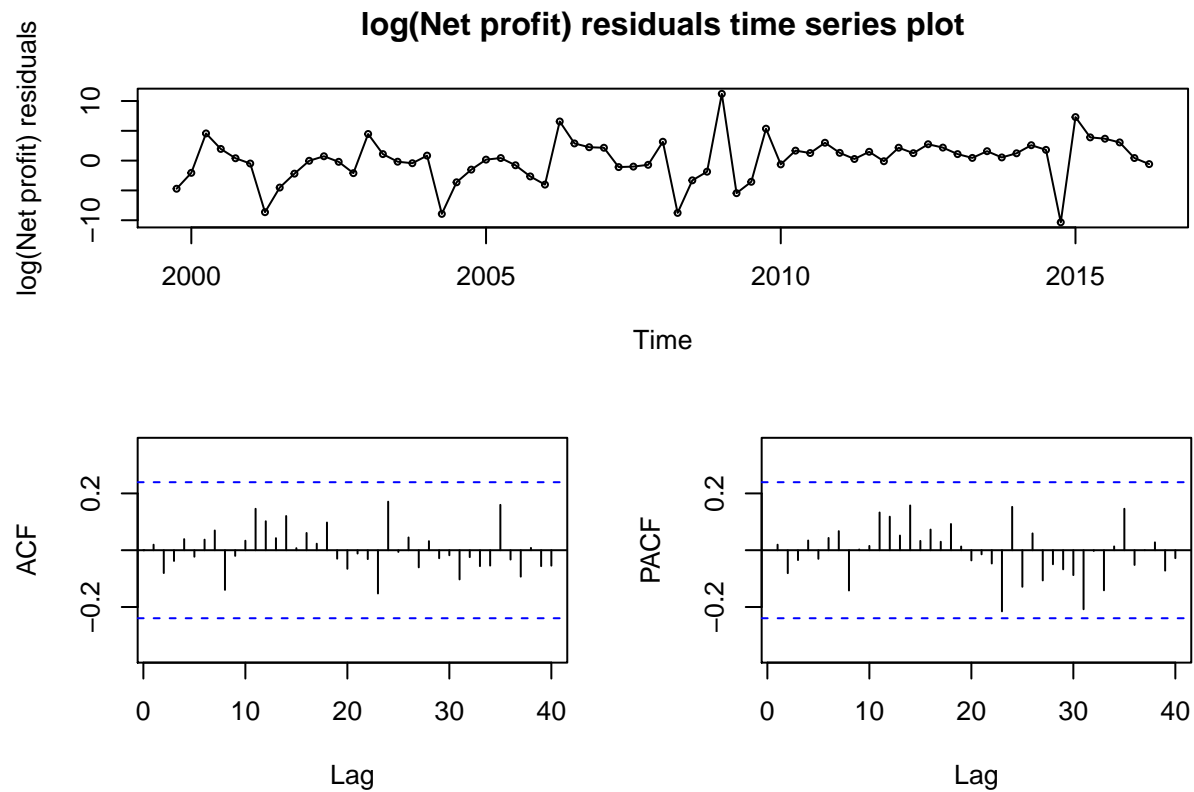
```
## Wrong parameters: 2 2 1 0
##
## Call:
## arima(x = data, order = c(params[which.min(aics), 1], d, params[which.min(aics),
## 2]), seasonal = list(order = c(params[which.min(aics), 3], D, params[which.min(aics),
## 4]), period = 4))
##
## Coefficients:
##      ar1      ar2      ar3      ma1      ma2 intercept
##      1.7967 -1.3394  0.3705 -1.3787  1.0000   7.2114
## s.e.  0.1253   0.1981  0.1245   0.0611  0.0631   1.5752
##
## sigma^2 estimated as 13.59: log likelihood = -185.07, aic = 384.15
## Best parameters: 3 2 0 0
```

Проанализируем остатки.

```
arima.model.analysis(best.persp.np.model, log(np), "log(Net profit)")
```

```
##
## One Sample t-test
##
## data: model$residuals
## t = 0.28709, df = 66, p-value = 0.7749
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.7753139 1.0357293
## sample estimates:
## mean of x
## 0.1302077

## Warning in adf.test(model$residuals): p-value smaller than printed p-value
```

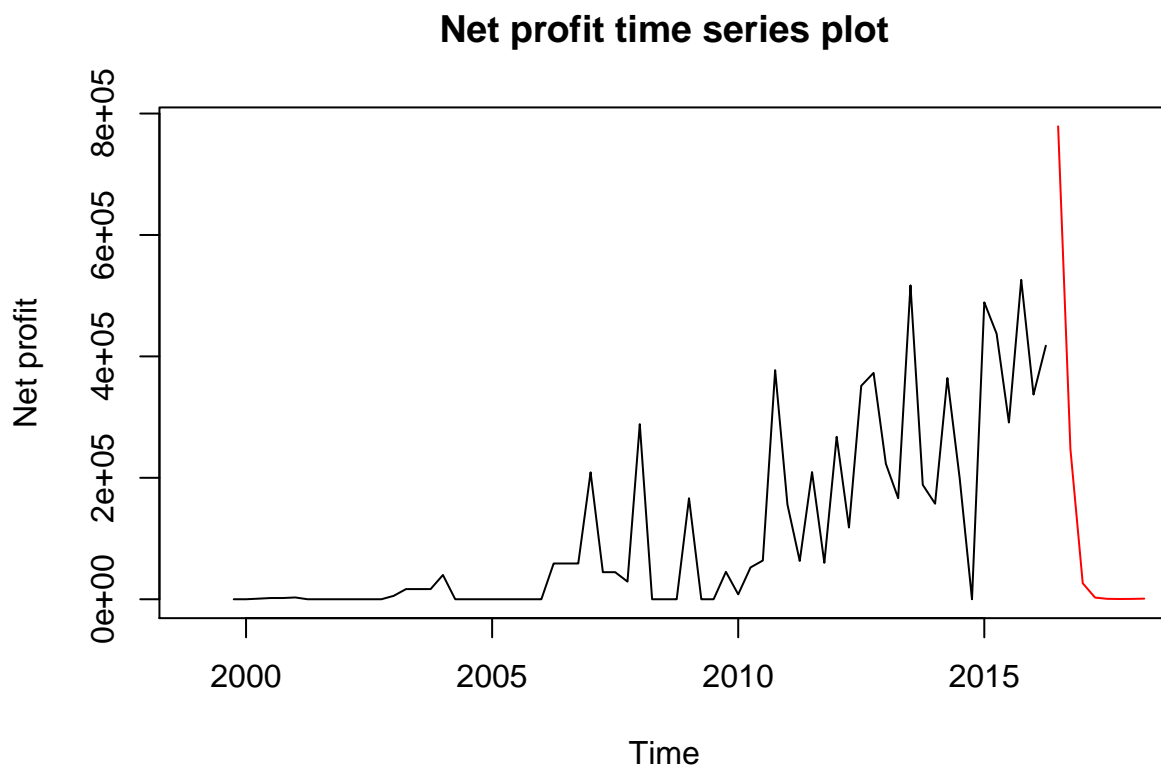


```
##
## Augmented Dickey-Fuller Test
##
## data: model$residuals
## Dickey-Fuller = -4.787, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
##
##
## Box-Pierce test
##
## data: model$residuals
## X-squared = 0.025415, df = 1, p-value = 0.8733
```

Остатки несмещены, стационарны и неавтокоррелированы.

Сделаем прогноз.

```
persp.np.predict <- arima.model.predict(best.persp.np.model, np, "Net profit", start=c(2016, 3))
```



4 Сохранение результатов

```
# таблица с прогнозами
forecast.df <- data.frame(rev(persp.cash.predict), rev(persp.ltb.predict), rev(persp.stb.predict),
                          rev(persp.rv.predict), rev(persp.gp.predict), rev(persp.np.predict))
names(forecast.df) <- c("Cash", "Long-term borrowings", "Short-term borrowings",
                       "Revenue", "Gross profit", "Net profit")
forecast.df <- data.frame(Quarter=c(2, 1, 4, 3, 2, 1, 4, 3), forecast.df)
forecast.df <- data.frame(Year=c(2018, 2018, 2017, 2017, 2017, 2017, 2016, 2016), forecast.df)
forecast.df
```

```
## Year Quarter Cash Long.term.borrowings Short.term.borrowings
## 1 2018 2 216231.4 105734.5 415417.6
## 2 2018 1 2138247.4 123374.4 507744.4
## 3 2017 4 231557.9 177088.0 634661.4
## 4 2017 3 750659.4 767285.4 813323.7
## 5 2017 2 197712.2 2179589.5 1071564.9
## 6 2017 1 1940969.3 2080410.6 1455966.6
## 7 2016 4 209348.1 2435446.3 2047183.4
## 8 2016 3 669337.4 6831219.6 2099604.8
## Revenue Gross.profit Net.profit
## 1 3056860 1146544 1013.3937
## 2 3072196 1179960 539.5318
```



```
## 3 3165497    1211803    396.0106
## 4 3215634    1235715    630.2740
## 5 3238245    1232430    2763.7021
## 6 3259594    1269720    26359.6236
## 7 3363996    1313975    247344.7369
## 8 3422931    1352180    778907.2941
```

```
write.csv(forecast.df, "C:/Users/Вредный Я/Desktop/Data Analysis/CMF/Time Series Econometrics/Exam (Financial ind
```