HW1 TS1

Alexey Sek

26.02.2023

1 Задача 1

Рассмотрим $MA(1)$ процесс $y_t = 10 + u_t + 3u_{t-1}$, где величины $u_t \sim N(0;4)$ и независимы

1.1 a)

Найти теоретическую автокорреляционную функцию ACF_k

$\operatorname{Var}(y_t) = \operatorname{Var}(u_t + 3u_{t-1} = \operatorname{Var}(u_t) + 9\operatorname{Var}(u_{t-1}) = 10\operatorname{Var}(u_t) = 10 * 4 = 40$

Если $k \geqslant 2$, то $\operatorname{Cov}(y_t, y_{t+k}) = 0$

Потому что у "игреков"не будет совпадающих "ушек а разные "ушки"независимы. Очень похожий пример разбирался в одной из первых лекций

Если $k = 1 : \operatorname{Cov}(y_t, y_{t+k}) = \operatorname{Cov}(y_t, y_{t+1}) = \operatorname{Cov}(u_t + 3u_{t-1}, u_{t+1} + 3u_t) = 3\operatorname{Var}(u_t) = 3 * 4 = 12$

Чтобы найти автокорреляционную функцию - необходимо поделить ковариацию на корень из произведения дисперсий, но дисперсии "ушек"равны, поэтому можно поделить просто на дисперсию

Ответ: $ACF_k = \begin{cases} 12/40 = 0.3, & \text{k=1} \\ 0, & \text{k} \geqslant 2 \end{cases}$

1.2 b)

Найти первые два значения частной автокорреляционной функции $PACF_1 = \phi_{11}$ и $PACF_2 = \phi_{22}$

Первое значение частной автокорреляционной функции совпадает со значением автокорреляционной функции, т.к. здесь нет промежуточных наблюдений

$PACF_1 = pCorr(y_t, y_{t+1}; \varnothing) = ACF_1 = 0.3 = \phi_{11}$

$y_t = const + \phi_{21}y_{t-1} + \phi_{22}y_{t-2} + v_t$

$v_t = y_t - const - \phi_{21}y_{t-1} - \phi_{22}y_{t-2}$

$\begin{cases} \mathrm{Cov}(v_t, y_{t-1}) = 0 \\ \mathrm{Cov}(v_t, y_{t-2}) = 0 \end{cases} \quad \begin{cases} \mathrm{Cov}(y_t - \phi_{21}y_{t-1} - \phi_{22}y_{t-2}, y_{t-1}) = 0 \\ \mathrm{Cov}(y_t - \phi_{21}y_{t-1} - \phi_{22}y_{t-2}, y_{t-2}) = 0 \end{cases}$

$\begin{cases} \mathrm{Cov}(y_{t-1}, y_t) = \phi_{21}\,\mathrm{Cov}(y_{t-1}, y_{t-1}) + \phi_{22}\,\mathrm{Cov}(y_{t-1}, y_{t-2}) \\ \mathrm{Cov}(y_{t-2}, y_t) = \phi_{21}\,\mathrm{Cov}(y_{t-2}, y_{t-1}) + \phi_{22}\,\mathrm{Cov}(y_{t-2}, y_{t-2}) \end{cases} \quad \begin{cases} \gamma_1 = \phi_{21}\gamma_0 + \phi_{22}\gamma_1 \\ \gamma_2 = \phi_{21}\gamma_1 + \phi_{22}\gamma_0 \end{cases}$

Поделим левые и правые части обеих уравнений на γ_0

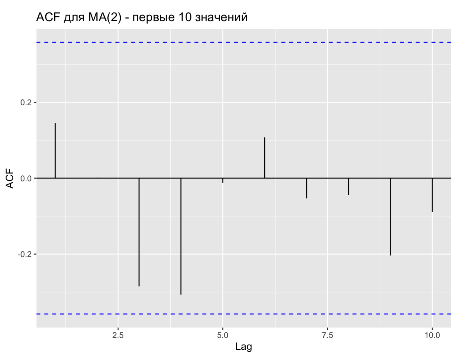
$\begin{cases} ACF_1 = \phi_{21} * 1 + \phi_{22} * ACF_1 \\ ACF_2 = \phi_{21} * ACF_1 + \phi_{22} * 1 \end{cases} \quad \begin{cases} \phi_{21} + 0.3\phi_{22} = 0.3 \\ 0.3\phi_{21} + \phi_{22} = 0 \end{cases} \quad \begin{cases} \phi_{22} = -0.3\phi_{21} \\ \phi_{21} - 0.09\phi_{21} = 0.3 \end{cases}$

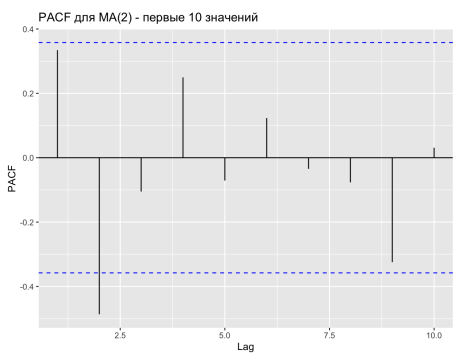
$\begin{cases} \phi_{21} = 0.3/0.91 \approx 0.33 \\ \phi_{22} = -0.3 * 0.3/0.91 = -9/91 \approx -0.1 \end{cases}$

Ответ: $\begin{cases} \mathrm{PACF}_1 = \phi_{11} \approx 0.33 \\ \mathrm{PACF}_2 = \phi_{22} = -9/91 \approx -0.1 \end{cases}$

1.3 c)

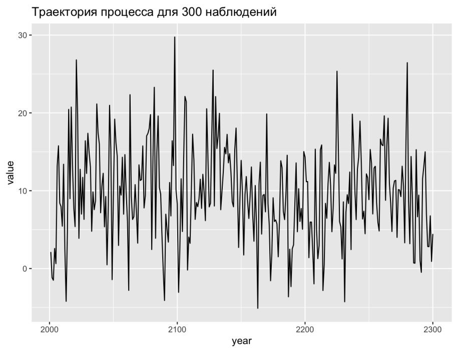
Сгенерировать 30 наблюдений, построить график ряда, графики первых 10 значений $PACF$ и ACF

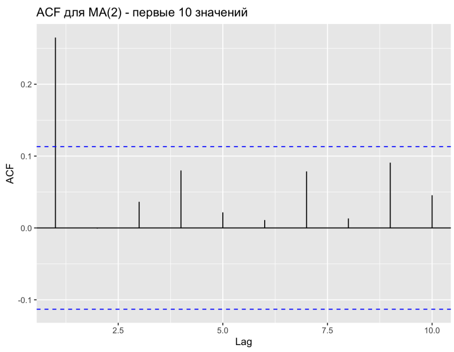


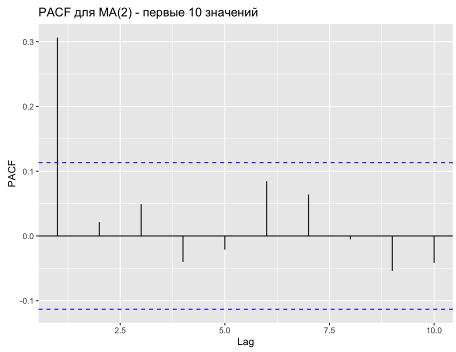


1.4 d)

Сгенерировать 300 наблюдений, построить график ряда, графики первых 10 значений $PACF$ и ACF







Вывод: действительно, при увеличении количества наблюдений до 300 - выборочные ACF и $PACF$ стали более похожи на истинные

2 Задача 2

Рассмотрим случайное блуждание $y_t = 1 + y_{t-1} + u_t$, где величины $u_t \sim N(0; 4)$ и независимы, а $y_0 = 10$

2.1 a)

Найти $\mathbb{E}(y_t)$**,** $\text{Var}(y_t)$ **и** $\text{Cov}(y_{10}, y_{20})$

$\mathbb{E}(y_0) = \mathbb{E}(10) = 10$
$\mathbb{E}(y_1) = \mathbb{E}(1 + y_0 + u_1) = \mathbb{E}(1) + \mathbb{E}(y_0) + \mathbb{E}(u_1) = 1 + 10 + 0 = 11$
$\mathbb{E}(y_2) = \mathbb{E}(1 + y_1 + u_2) = \mathbb{E}(1) + \mathbb{E}(y_1) + \mathbb{E}(u_2) = 1 + 11 + 0 = 12$

Можно заметить, что: $\mathbb{E}(y_t) = 10 + t$

$\text{Var}(y_0) = \text{Var}(10) = 0$
$\text{Var}(y_1) = \text{Var}(1 + y_0 + u_1) = \text{Var}(1) + \text{Var}(y_0) + \text{Var}(u_1) = 0 + 0 + 4 = 4$
$\text{Var}(y_2) = \text{Var}(1 + y_1 + u_2) = \text{Var}(1) + \text{Var}(y_1) + \text{Var}(u_2) = 0 + 4 + 4 = 8$
$\text{Var}(y_3) = \text{Var}(1 + y_2 + u_3) = \text{Var}(1) + \text{Var}(y_2) + \text{Var}(u_3) = 0 + 8 + 4 = 12$

Можно заметить, что: $\text{Var}(y_t) = 4t$

$\text{Cov}(y_{10}, y_{20}) = \text{Cov}(1 + y_9 + u_{10}, 1 + y_{19} + u_{20}) = \text{Cov}(y_9, y_{19}) + \text{Cov}(y_9, u_{20}) + \text{Cov}(u_{10}, y_{19}) + \text{Cov}(u_{10}, u_{20}) = 9\,\text{Var}(u_t) + 0 + \text{Var}(u_t) + 0 = 10\,\text{Var}(u_t) = 10 * 4 = 40$

Таким образом: $\text{Cov}(y_{10}, y_{20}) = 40$

2.2 b)

Сравнить $\text{Corr}(y_{10}, y_{20})$ **и** $\text{Corr}(y_{110}, y_{120})$

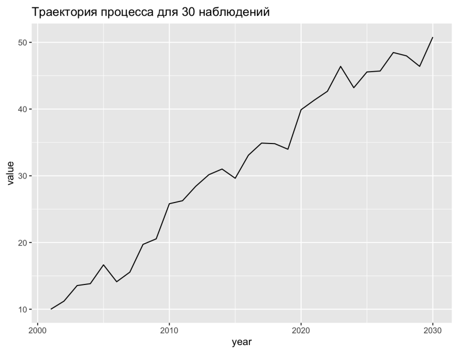
$\text{Corr}(y_{10}, y_{20}) = \frac{\text{Cov}(y_{10}, y_{20})}{\sqrt{\text{Var}(y_{10})\,\text{Var}(y_{20})}} = \frac{40}{\sqrt{40*80}} \approx 70.7\%$

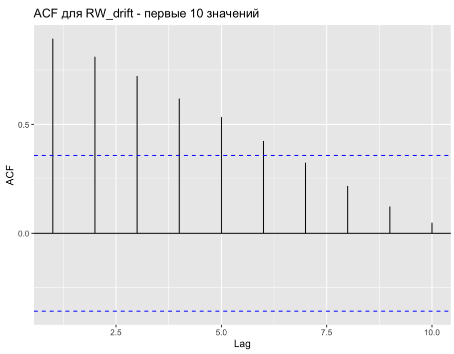
$\text{Corr}(y_{110}, y_{120}) = \frac{\text{Cov}(y_{110}, y_{120})}{\sqrt{\text{Var}(y_{110})\,\text{Var}(y_{120})}} = \frac{40}{\sqrt{440*480}} \approx 8.7\%$

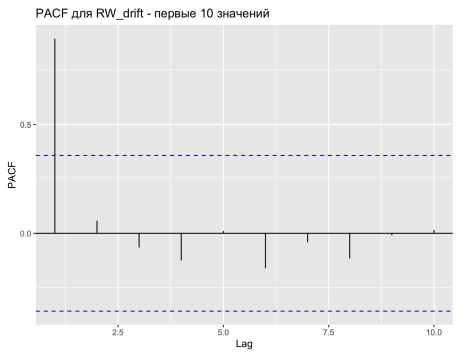
Таким образом: $\text{Corr}(y_{10}, y_{20}) > \text{Corr}(y_{110}, y_{120})$

2.3 c)

Сгенерировать 30 наблюдений, построить график ряда, графики первых 10 значений $PACF$ и ACF

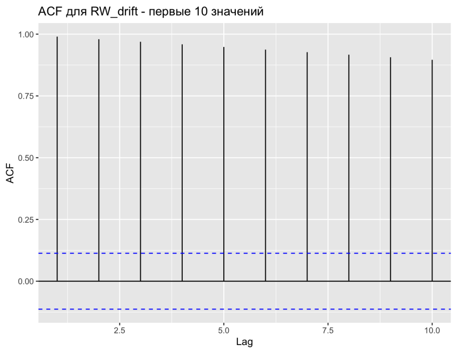


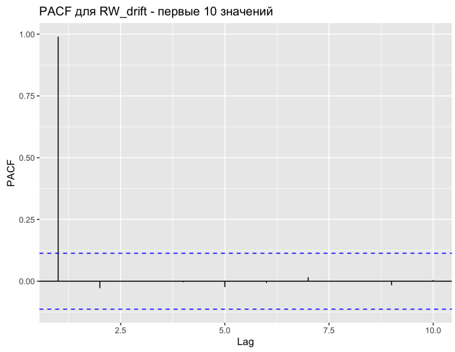




2.4 d)

Сгенерировать 300 наблюдений, построить график ряда, графики первых 10 значений $PACF$ и ACF





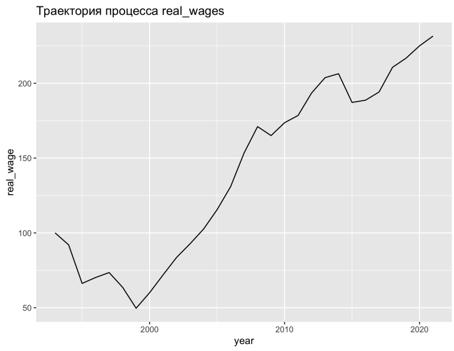
Вывод: действительно, при увеличении количества наблюдений до 300 - выборочные ACF и $PACF$ стали более похожи на истинные

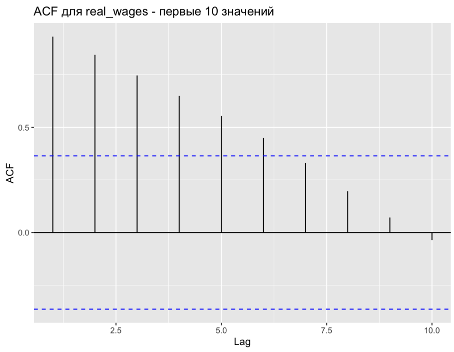
3 Задача 3

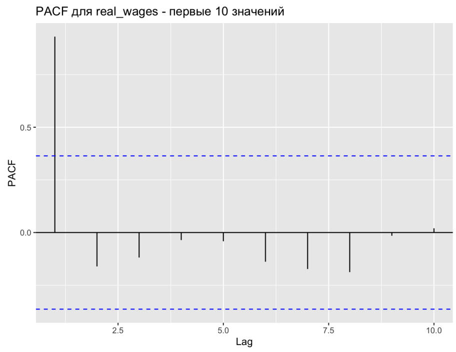
Возьмем любой несезонный ряд годовой периодичности - у меня уровень реальной зп в РФ по годам

3.1 a)

Построить график ряда, выборочные ACF и $PACF$







3.2 b)

Визуально оценить наличие тренда и стационарность

Вывод: Тренд явно присутствует - график заметно возрастает со временем. Процесс не похож на стационарный: для стационарности необходимо, чтобы математическое ожидание оставалось одинаковым, а здесь оно, как кажется по графику, растет

3.3 c) d)

Оценить модель ETS(AAN) и выписать оцененочные значения параметров

```
Series: real_wage
Model: ETS(A,A,N)
  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.0001000268

  Initial states:
     l[0]     b[0]
 88.02125 4.67952

  sigma^2:  142.4486

     AIC      AICc       BIC
247.1579 249.7666 253.9943
```

Тогда модель примет следующий вид:

$$\begin{cases} y_t = l_{t-1} + b_{t-1} + u_t \\ l_t = l_{t-1} + b_{t-1} + 0.9999u_t, & l_0 = 88.02 \\ u_t \sim N(0; 142.45) \text{ и независимы} \\ b_t = b_{t-1} + 0.0001u_t, & b_0 = 4.68 \end{cases}$$

3.4 e)

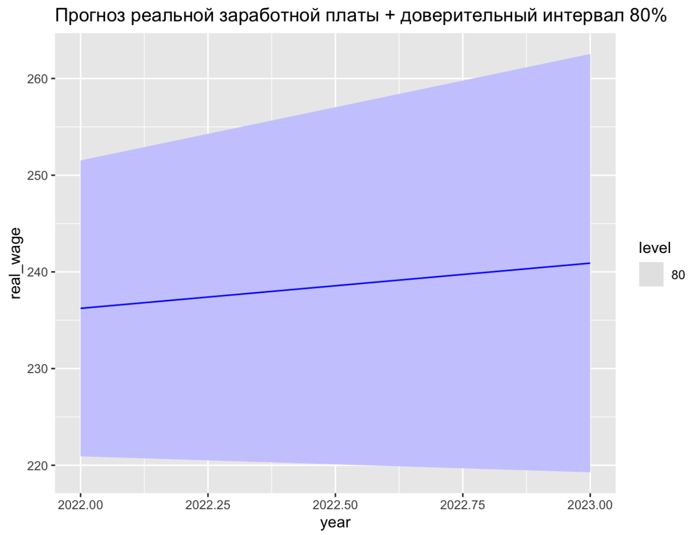
Оценить 80% доверительные интервалы на один и два шага вперед лапками

Пока не очень понял, как это сделать

3.5 f)

Оценить 80% доверительные интервалы на один и два шага вперед с помощью R

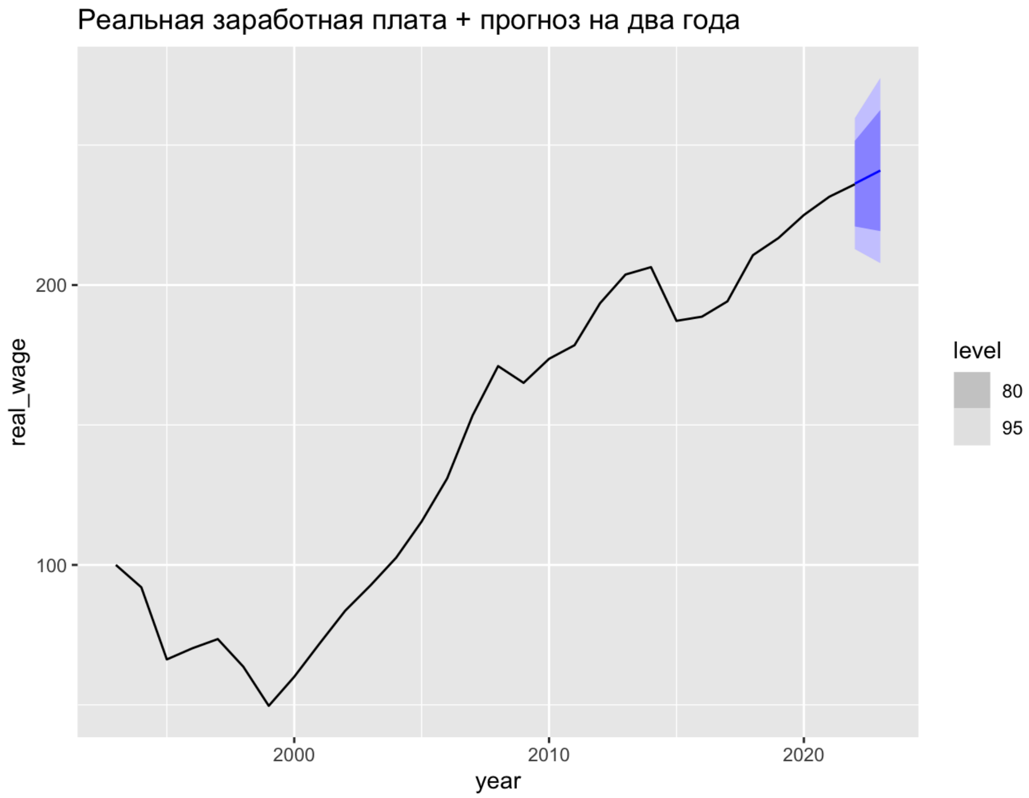
Ниже график для 80% доверительного интервала + значения границ доверительных интервалов на 1 и 2 шага вперед




```
> # Параметры брал из fcst - там есть mean и var
> print('CI_80% Lower bound - 1 step forward:')
[1] "CI_80% Lower bound - 1 step forward:"
> print(236 - 1.282 * sqrt(142))
[1] 220.7232
> print('CI_80% Upper bound - 1 step forward:')
[1] "CI_80% Upper bound - 1 step forward:"
> print(236 + 1.282 * sqrt(142))
[1] 251.2768
>
> print('CI_80% Lower bound - 2 steps forward:')
[1] "CI_80% Lower bound - 2 steps forward:"
> print(241 - 1.282 * sqrt(285))
[1] 219.3573
> print('CI_80% Upper bound - 2 steps forward:')
[1] "CI_80% Upper bound - 2 steps forward:"
> print(241 + 1.282 * sqrt(285))
[1] 262.6427
```

3.6 g)

Построить график прогноза и график самого ряда

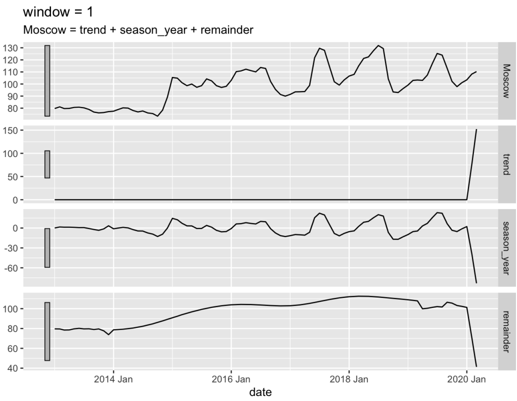


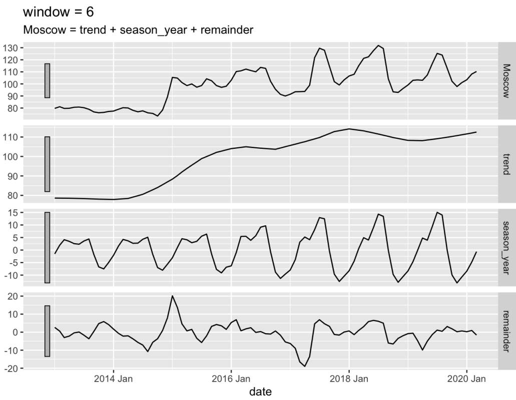
4 Задача 4

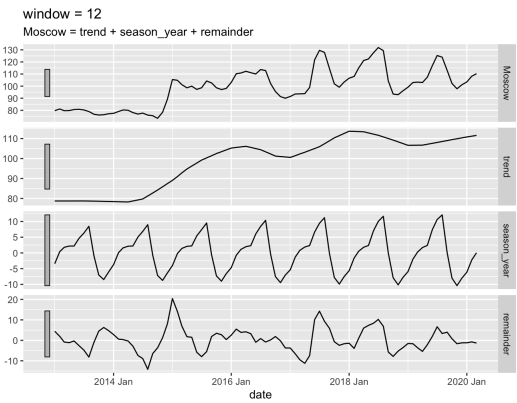
Взять любой сезонный ряд с квартальной или месячной периодичностью - у меня цены на яблоки в Москве

4.1 a)

STL разложение ряда + Визуализация для 3 уровней силы сглаживания сезонности



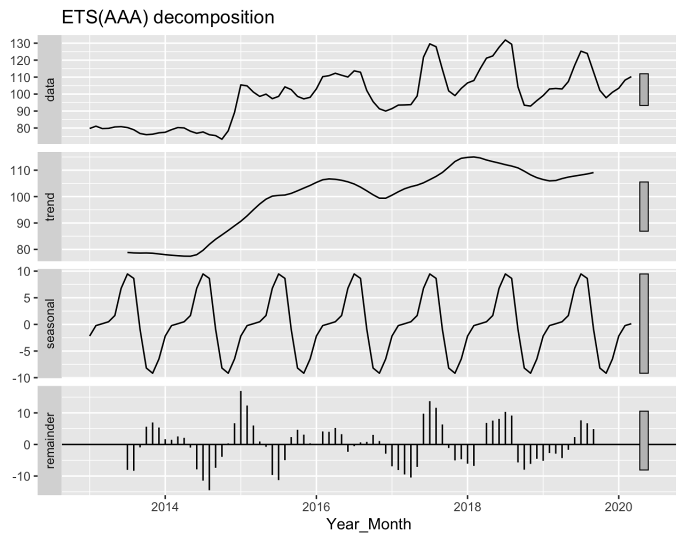




Выводы: чем выше параметр window для сезонности (сила сглаживания сезнности) - тем более похожи компоненты сезонности между собой для разных периодов. Также заметно, что компонента Remainder заметно уменьшается при повышении window для сезонности

4.2 b)

Разложение ряда на составляющие с помощью ETS(AAA) модели (аддитивный метод Хольта-Винтерса)



4.3 c) d)

Разделить даныне на обучающую и тестовую выборки + обучить модели

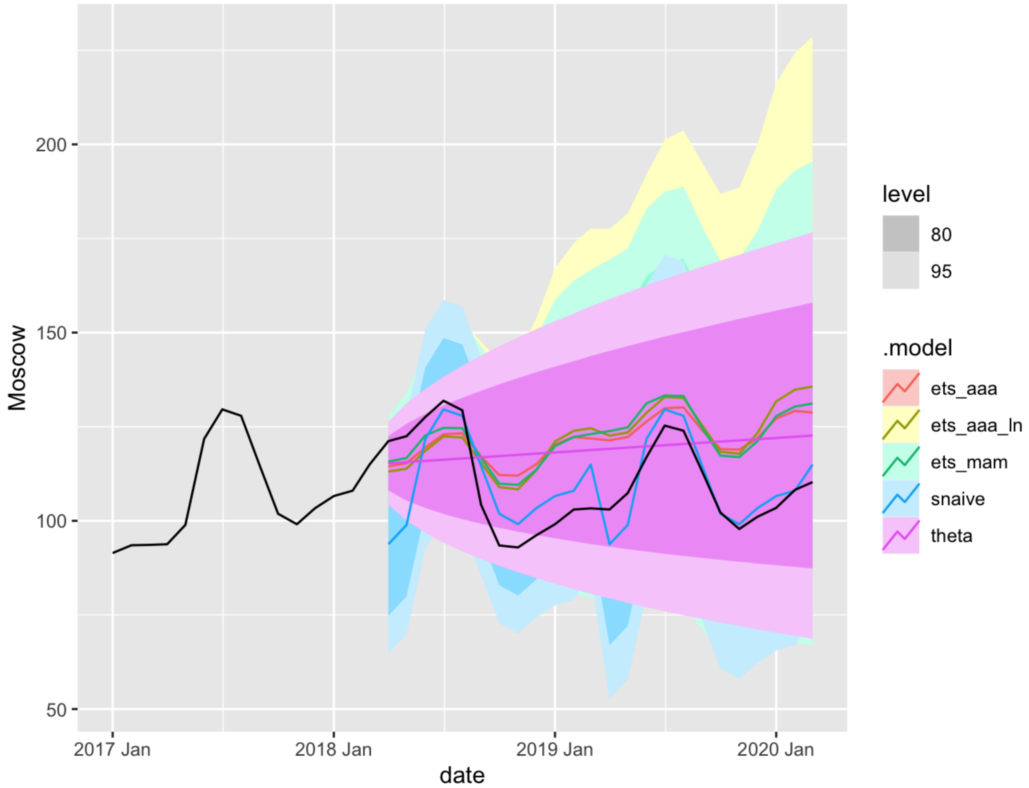
```
# Поделим на тестовую и тренировочную выборки
# В тестовой выборке 2 последних года
train <- filter(apple_prices, date >= yearmonth(ymd('2013-04-01')))
train <- filter(train, date <= yearmonth(ymd('2018-03-01')))
test <- filter(apple_prices, date > yearmonth(ymd('2018-03-01')))
print(train, n=100)
print(test, n=100)


# Обучим указанные в задании модели
mods = model(train,
             ets_aaa = ETS(Moscow ~ error('A') + trend('A') + season('A')),
             ets_mam = ETS(Moscow ~ error('M') + trend('A') + season('M')),
             snaive = SNAIVE(Moscow),
             theta = THETA(Moscow),
             ets_aaa_ln = ETS(log(Moscow) ~ error('A') + trend('A') + season('A'))
             )

```

4.4 e)

Найти MASE для каждой модели + добавил графики прогнозов разных моделей




```
> accuracy(fcst, apple_prices)
# A tibble: 5 × 10
  .model     .type      ME  RMSE   MAE     MPE  MAPE  MASE RMSSE  ACF1
  <chr>      <chr>   <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
1 ets_aaa    Test  -11.6    15.8  14.7 -11.6    14.0 1.23  1.10  0.828
2 ets_aaa_ln Test  -12.3    17.1  15.9 -12.2    15.0 1.33  1.19  0.834
3 ets_mam    Test  -12.4    15.9  14.7 -12.2    14.0 1.24  1.11  0.823
4 snaive     Test   -0.143   9.32  6.71  -0.541  6.09 0.565 0.649 0.525
5 theta      Test   -9.08   15.6  14.2   -9.53  13.6 1.20  1.08  0.797
```

Два лучших подхода: 1) сезонная наивная модель; 2) тета-метод

4.5 f)

Усреднить две лучшие модели и сравнить полученную модель с усредняемыми

	.model	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	average_models	Test	-4.61	10.6	9.37	-5.03	8.84	0.789	0.735	0.746
2	ets_aaa	Test	-11.6	15.8	14.7	-11.6	14.0	1.23	1.10	0.828
3	ets_aaa_ln	Test	-12.3	17.1	15.9	-12.2	15.0	1.33	1.19	0.834
4	ets_mam	Test	-12.4	15.9	14.7	-12.2	14.0	1.24	1.11	0.823
5	snaive	Test	-0.143	9.32	6.71	-0.541	6.09	0.565	0.649	0.525
6	theta	Test	-9.08	15.6	14.2	-9.53	13.6	1.20	1.08	0.797

Вывод: новая модель, полученная усреднением, не смогла обыграть обе усредняемые модели - сезонная наивная все равно оказалась наилучшей

5 Код

```
# install.packages("tsutils")
# install.packages("forecast")
# install.packages("ggplot2")
# install.packages("ggfortify")
# install.packages("dplyr")
library(tidyverse) # обработка данных
library(fpp3) # куча плюшек для рядов
library(rio) # импорт данных
library(dplyr)
library(ggfortify)
library(ggplot2)
library(tsutils)
#library(forecast)
set.seed(777)

# ------------------
# ДЗ_1 по курсу АВР_1
# Дата выполнения 25.02.2023
# Студент: Сек Алексей



# ------------------
# Задание 1

# Начнем с 30 наблюдений

# По дефолту константа равна 0, поэтому сначала посчитаем с 0, потом добавим 10
# По итогу получим y_t = 10 + u_t + 3*u_{t-1}
data <- tibble(ma_2_data = 10 + arima.sim(n=30, list(ma=c(3)), sd=2))

# Добавим дату
data$year = 2001:2030
data <- as_tsibble(data, index = year)

# Посмотрим, корректная ли табличка получилась
data

# Посмотрим на график процесса
p <- ggplot(data, aes(x=year, y=ma_2_data)) +
  geom_line() +
  ggtitle('Траектория процесса для 30 наблюдений') +
  xlab('year') +
  ylab('value')
p

# Посмотроим ACF для данного процесса
```

```
ACF(data)
acf <- ACF(data, ma_2_data, lag_max=10) %>% autoplot()
acf +
  ggtitle('ACF для MA(2) - первые 10 значений') +
  xlab('Lag') +
  ylab('ACF')


# Посмотроим PACF для данного процесса
PACF(data)
pacf <- PACF(data, ma_2_data, lag_max=10) %>% autoplot()
pacf +
  ggtitle('PACF для MA(2) - первые 10 значений') +
  xlab('Lag') +
  ylab('PACF')



# ------------------

# Теперь посмотрим 300 наблюдений

# По дефолту константа равна 0, поэтому сначала посчитаем с 0, потом добавим 10
# По итогу получим y_t = 10 + u_t + 3*u_{t-1}
data <- tibble(ma_2_data = 10 + arima.sim(n=300, list(ma=c(3)), sd=2))

# Добавим дату
data$year = 2001:2300
data <- as_tsibble(data, index = year)

# Посмотрим, корректная ли табличка получилась
data

# Посмотрим на график процесса
p <- ggplot(data, aes(x=year, y=ma_2_data)) +
  geom_line() +
  ggtitle('Траектория процесса для 300 наблюдений') +
  xlab('year') +
  ylab('value')
p

# Посмотроим ACF для данного процесса
ACF(data)
acf <- ACF(data, ma_2_data, lag_max=10) %>% autoplot()
acf +
  ggtitle('ACF для MA(2) - первые 10 значений') +
  xlab('Lag') +
  ylab('ACF')
```

```r
# Посмотроим PACF для данного процесса
PACF(data)
pacf <- PACF(data, ma_2_data, lag_max=10) %>% autoplot()
pacf +
  ggtitle('PACF для MA(2) - первые 10 значений') +
  xlab('Lag') +
  ylab('PACF')




# ------------------
# Задание 2

# Начнем с 30 наблюдений
data <- tibble(random_walk_drift =
                 10 + arima.sim(
                   model=list(order = c(0,1,0)),n=29, mean=1, sd=2
                   )
               )

# Добавим дату
data$year = 2001:2030
data <- as_tsibble(data, index = year)

# Посмотрим, корректная ли табличка получилась
data

# Посмотрим на график процесса
p <- ggplot(data, aes(x=year, y=random_walk_drift)) +
  geom_line() +
  ggtitle('Траектория процесса для 30 наблюдений') +
  xlab('year') +
  ylab('value')
p

# Посмотроим ACF для данного процесса
ACF(data)
acf <- ACF(data, random_walk_drift, lag_max=10) %>% autoplot()
acf +
  ggtitle('ACF для RW_drift - первые 10 значений') +
  xlab('Lag') +
  ylab('ACF')


# Посмотроим PACF для данного процесса
PACF(data)
pacf <- PACF(data, random_walk_drift, lag_max=10) %>% autoplot()
pacf +
```

```
  ggtitle('PACF для RW_drift - первые 10 значений') +
  xlab('Lag') +
  ylab('PACF')



# ------------------
# Рассмотрим 300 наблюдений
data <- tibble(random_walk_drift =
                 10 + arima.sim(
                   model=list(order = c(0,1,0)),n=299, mean=1, sd=2))

# Добавим дату
data$year = 2001:2300
data <- as_tsibble(data, index = year)

# Посмотрим, корректная ли табличка получилась
data

# Посмотрим на график процесса
p <- ggplot(data, aes(x=year, y=random_walk_drift)) +
  geom_line() +
  ggtitle('Траектория процесса для 300 наблюдений') +
  xlab('year') +
  ylab('value')
p

# Посмотроим ACF для данного процесса
ACF(data)
acf <- ACF(data, random_walk_drift, lag_max=10) %>% autoplot()
acf +
  ggtitle('ACF для RW_drift - первые 10 значений') +
  xlab('Lag') +
  ylab('ACF')


# Посмотроим PACF для данного процесса
PACF(data)
pacf <- PACF(data, random_walk_drift, lag_max=10) %>% autoplot()
pacf +
  ggtitle('PACF для RW_drift - первые 10 значений') +
  xlab('Lag') +
  ylab('PACF')




# ------------------
# Задание 3
```

```r
# Данные по реальной заработной плате:
# http://sophist.hse.ru/hse/1/tables/WAG_Y.htm

# Загрузим данные
df_wages = import(
  '//Users/alexeysek/Downloads/2. TS_1_online/HW1/wages.csv', dec = ",")
df_wages = as_tsibble(df_wages, index = year)
glimpse(df_wages)


# Посмотрим на график процесса
p <- ggplot(df_wages, aes(x=year, y=real_wage)) +
  geom_line() +
  ggtitle('Траектория процесса real_wages') +
  xlab('year') +
  ylab('real_wage')
p

# Посмотроим ACF для данного процесса
ACF(df_wages)
acf <- ACF(df_wages, real_wage, lag_max=10) %>% autoplot()
acf +
  ggtitle('ACF для real_wages - первые 10 значений') +
  xlab('Lag') +
  ylab('ACF')


# Посмотроим PACF для данного процесса
PACF(df_wages)
pacf <- PACF(df_wages, real_wage, lag_max=10) %>% autoplot()
pacf +
  ggtitle('PACF для real_wages - первые 10 значений') +
  xlab('Lag') +
  ylab('PACF')


# Визуально тренд есть - очень заметно по графику, что он восходящий
# Тренд похоже есть -> процесс не похож на стационарный
# т.к. среднее должно быть постоянным для стационарности


# Оценим ETS(AAN)
ets_aan = model(
  df_wages, ets_aan = ETS(real_wage ~ error('A') + trend('A') + season('N')))
report(ets_aan)

# Получили следующие оценки параметров модели ETS(AAN):
# alpha = 0.9999
# beta  = 0.0001000268
# l_0 = 88.02125
```

```r
# b_0 = 4.67952
# sigma^2 = 142.4486

fcst <- forecast(ets_aan, h = '2 years', level=c(80))
fcst

# Построим график прогноза с 80% доверительным интервало
autoplot(fcst, level=c(80)) +
  ggtitle('Прогноз реальной заработной платы + доверительный интервал 80%')

# Почему-то стандартная функция forecast в fpp3 не выдает ДИ
# Такое есть только в https://otexts.com/fpp2/the-forecast-package-in-r.html

# Поэтому ручками посчитал доверительные интервалы
# Параметры брал из fcst - там есть mean и var
print('CI_80% Lower bound - 1 step forward:')
print(236 - 1.282 * sqrt(142))
print('CI_80% Upper bound - 1 step forward:')
print(236 + 1.282 * sqrt(142))

print('CI_80% Lower bound - 2 steps forward:')
print(241 - 1.282 * sqrt(285))
print('CI_80% Upper bound - 2 steps forward:')
print(241 + 1.282 * sqrt(285))

# Чтобы не было пробела между данными и прогнозом
# Какой-то баг библиотеки, как мне кажется, что появляется этот пробел
df_wages <- df_wages %>%
  add_row(year=2022, real_wage=236)

autoplot(fcst, df_wages) +
  ggtitle('Реальная заработная плата + прогноз на два года')




# ------------------
# Задание 4
# Сезонный ряд месячной периодичности
# Цены на яблоки в РФ:
# https://www.kaggle.com/datasets/kapatsa/apple-prices-in-russian-regions

# Загрузим данные
apple_prices <- import(
  '//Users/alexeysek/Downloads/2. TS_1_online/HW1/apple_prices.csv')
# Преобразуем дату в нормальный формат
apple_prices$date <- yearmonth(ymd(apple_prices$date))
apple_prices <- as_tsibble(apple_prices, index=date)
# Возьмем ряд только по Москве
```

```r
apple_prices <- apple_prices[ ,c('date', 'Moscow')]
apple_prices


# Построим графики STL разложений с разной силой сглаживания сезонности
model(apple_prices,
  STL(Moscow ~ trend() + season(), robust = TRUE)) |> components() |>
  autoplot() + ggtitle('Дефолтные параметры')

model(apple_prices,
      STL(Moscow ~ trend() + season(window = 1), robust = TRUE)) |>
  components() |> autoplot() + ggtitle('window = 1')

model(apple_prices,
      STL(Moscow ~ trend() + season(window = 2), robust = TRUE)) |>
  components() |> autoplot() + ggtitle('window = 2')

model(apple_prices,
      STL(Moscow ~ trend() + season(window = 4), robust = TRUE)) |>
  components() |> autoplot() + ggtitle('window = 4')

model(apple_prices,
      STL(Moscow ~ trend() + season(window = 6), robust = TRUE)) |>
  components() |> autoplot() + ggtitle('window = 6')

model(apple_prices,
      STL(Moscow ~ trend() + season(window = 12), robust = TRUE)) |>
  components() |> autoplot() + ggtitle('window = 12')



# ETS(AAA) разложение
# Другими словами - аддитивный метод Хольта-Винтерса
autoplot(decompose(as.ts(apple_prices, frequency=12))) +
  ggtitle('ETS(AAA) decomposition') +
  xlab('Year_Month')


# Поделим на тестовую и тренировочную выборки
# В тестовой выборке 2 последних года
train <- filter(apple_prices, date >= yearmonth(ymd('2013-04-01')))
train <- filter(train, date <= yearmonth(ymd('2018-03-01')))
test <- filter(apple_prices, date > yearmonth(ymd('2018-03-01')))
print(train, n=100)
print(test, n=100)


# Обучим указанные в задании модели
mods = model(train,
             ets_aaa = ETS(Moscow ~ error('A') + trend('A') + season('A')),
```

```r
             ets_mam = ETS(Moscow ~ error('M') + trend('A') + season('M')),
             snaive = SNAIVE(Moscow),
             theta = THETA(Moscow),
             ets_aaa_ln = ETS(log(Moscow) ~ error('A') + trend('A') +
                                season('A'))
             )


# Сделаем прогноз и сравним результаты
# Особенно нам интересно MASE
fcst = forecast(mods, h = 24)
accuracy(fcst, apple_prices)

# Построим график
autoplot(fcst, filter(apple_prices, date >= yearmonth(ymd('2017-01-01'))))

# Две лучшие по MASE модели (min MASE) - это snaive и theta
# Усредним их прогнозы
mod_aggr = mutate(mods, average_models = (snaive + theta) / 2)
fcst = forecast(mod_aggr, h = 24)
accuracy(fcst, apple_prices)

# Усредненная модель оказалось посередине между 1 и 2 местом
# То есть нам не удалось обыграть оба усредняемых подхода,
# т.к. snaive все равно остался лучшим
```