Национальный исследовательский университет

”ВЫСШАЯ ШКОЛА ЭКОНОМИКИ”

Факультет экономических наук

Курсовая работа

Применение моделей MIDAS к Российским макроэкономическим данным

Выполнила: Кузина Анна, БЭК132

Преподаватель: Демешев Борис Борисович

Москва

2016 г.

Содержание

1. Введение

Большинство моделей прогнозирования подразумевает, что используемые временные ряды имеют одинаковую частоту наблюдений. Однако, на практике данные по различным макроэкономическим или финансовым показателям собираются с разной частотой. Большинство макроэкономических показателей публикуются с частотой в квартал, а реже - в год. Так, данные по ВВП становятся доступны нам с максимальной частотой равной одному кварталу. В то же время, из экономической теории известно, что такие макроэкономические показатели как ВВП или экспорт могут быть подвержены влиянию со стороны более "высокочастотных" переменных, таких как безработица, ставки процента или валютный курс.

Одним из решений проблемы разночастотности данных является агрегирование низкочастотных данных, однако, оно может отрицательно сказаться на качестве полученных таким образом прогнозов, так как ведет к потере части информации о той или иной переменной. Более того, высокочастотные данные на практике появляются в открытом доступе гораздо раньше низкочастотных за тот же период: данные по безработице за первые три месяца года могут быть доступны, в то время как ВВП первого квартала еще не опубликован. В результате, появляется необходимость прогнозировать не только будущий, но текущий ВВП (now-casting), используя все доступные данные по высокочастотным переменным.

В основе данной работы лежат модели MIDAS(Mixed Data Sampling regression), предлагающие достаточно эффективный способ борьбы с проблемой разночастотности, и показавшие хорошие результаты при прогнозировании различных, в том числе и макроэкономических, данных. Помимо описания принципа работы моделей MIDAS большое внимание в работе будет уделено пакету `midasr` для `R`, предоставляющему широкие

возможности для построения моделей с использованием разночастотных данных. Практическая часть работы включает в себя прогнозирование различных макроэкономических показателей с использованием MIDAS на российских данных, а также сравнение полученных результатов с прогнозами более простых моделей, таких как модель без ограничений (Step weighting), модель с агрегированными данными (Time Averaging), ARIMA, AR.

Основные цели работы заключаются в изучении базовых спецификаций моделей MIDAS, способах их реализации в R, а также в сравнении их по прогнозной силе с другими, более распространенными на данный момент моделями для прогнозирования.

2. Теоретическая часть

2.1. Неограниченная и агрегированная модели

В моделях MIDAS предполагается, что исследователь строит регрессию низкочастотного ряда Y_t на лаги высокочастотных данных X_t (объясняющие данные могут состоять из одного или нескольких высокочастотных рядов). В более обобщенном виде в регрессию также включены лаги самого низкочастотного ряда.

Запишем уравнение регрессии в общем виде, используя p лагов (L) низкочастотного ряда, а также $j_{max} = m \times k$ лагов (L_{HF}) высокочастотного, m - количество высокочастотных наблюдений X за один низкочастотный период Y, k - количество "длинных" лагов для X. В рамках данной работы подобную спецификацию модели можно назвать неограниченной, так как нет никаких дополнительных условий, ограничивающих значения, которые могут принимать коэффициенты.

$$Y_t = \gamma + \sum_{i=1}^{p} \alpha_i L^i Y_t + \sum_{j=0}^{m \times k} \beta_j L_{HF}^j X_t + \epsilon_t \tag{1}$$

Если же в модели используется больше одной объясняющей переменной, то есть N высокочастотных рядов, неограниченная модель (1) приобретает более громоздкий вид:

$$Y_t = \gamma + \sum_{i=1}^{p} \alpha_i L^i Y_t + \sum_{n=1}^{N} \sum_{j=0}^{m \times k} \beta_n^{(j)} L_{HF}^j X_{t,n} + \epsilon_t \tag{2}$$

Такая неограниченная модель, которую в литературе также называют Step Weighting [1], предполагает построение регрессии и оценку всех коэффициентов методом наименьших квадратов. Однако, количество

коэффициентов, которое необходимо оценить, невероятно велико. Если быть точным, оно составляет $1 + p + N \cdot j_{max}$. Для примера, если мы строим регрессию ВВП (с ежегодными данными) на уровень безработицы и объем производства нефти (ежемесячные наблюдения) на основании данных последних пяти лет (5 низкочастотных лагов), то мы получим $1 + 5 + 2 \cdot 5 \cdot 12 = 126$ коэффициентов. Основная проблема, которая особенно актуальна для Российских исследований - недостаточное количество данных, чтобы получить оценки коэффициентов в данном случае.

Наиболее простой способ сокращения числа коэффициентов в модели — агрегирование высокочастотных данных, такие модели еще называют Time Averaging[1]. То есть можно просто взять среднее значение безработицы за 12 месяцев и получить ее годовое значение. Далее можно строить обычную неограниченную модель (2), но уже для данных одинаковой частоты.

$$\overline{X}_t = \frac{1}{m} \sum_{i=1}^{m} X_i$$

$$Y_t = \gamma + \sum_{i=1}^{p} \alpha_i L^i Y_t + \sum_{n=1}^{N} \sum_{j=0}^{k} \beta_n^{(j)} L_{HF}^j \overline{X}_{t,n} + \epsilon_t \tag{3}$$

В результате, количество коэффициентов может значительно сократиться. Если вернуться к примеру, рассмотренному выше, в аналогичной модели после агрегирования мы получим уже $1 + 5 + 2 \cdot 5 = 16$ коэффициентов, что на 110 оцениваемых параметров меньше, чем было изначально. Однако, нельзя гарантировать что каждое значение высокочастотного параметра имеет одинаковое влияние на зависимую переменную (а именно это и предполагается, когда мы усредняем значения за период). А значит, в результате подобного способа борьбы с разночастотностью мы теряем

часть доступной нам информации, что может отрицательно сказаться на качестве получаемых прогнозов.

2.2. Mixed data sampling regression

В Ghysels, Santa-Clara, and Valkanov (2004)[3] впервые были описаны модели MIDAS, в которых подразумевается наличие определенной функциональной связи между коэффициентами регрессии, и, как следствие, значительно снижается количество параметров, которые необходимо оценить. В Ghysels, Sinko, and Valkanov (2007)[6] MIDAS модели показали отличные результаты при прогнозировании макроэкономических показателей. В этой же работе было представлено расширение моделей MIDAS, включающее в себя авторегрессионную динамику. В общем виде модели MIDAS подразумевают модификацию неограниченной модели (2), которая теперь имеет следующий вид:

$$Y_t = \gamma + \sum_{i=1}^{p} \alpha_i L^i Y_t + \sum_{n=1}^{N} \beta_n B(L_{HF}, \theta_n) X_{t,n} + \epsilon_t, \tag{4}$$

где $B(L_{HF}, \theta_n) = \sum_{j=0}^{j_{max}} \Phi(\theta_n, k) L_{HF}^{j/m}$ — многочлен, определяющий веса лагов высокочастотных наблюдений в модели.

При введении таких ограничений в модель количество коэффициентов значительно снижается. Так как используемые функции не являются линейными, оценки коэффициентов могут быть получены с использованием нелинейного метода наименьших квадратов (NLS).

Получив оценки коэффициентов, мы можем строить прогноз для зависимой переменной. Пусть T_y - индекс последнего доступного наблюдения низкочастотного ряда, а T_x - последнего доступного наблюдения для

X. Оценка значения зависимой переменной через h периодов может быть найдена следующим образом:

$$Y_{T_y+h} = \widehat{\gamma} + \sum_{i=1}^{p} \widehat{\alpha}_i L^i Y_{T_y} + \widehat{\beta} B(L_{HF}, \widehat{\theta}_n) X_{T_x} \tag{5}$$

Из уравнения (5) можно сделать несколько важных выводов о специфике процесса прогнозирования с использованием моделей MIDAS.

- Во-первых, прогноз зависит от выбранного горизонта.

Так, если мы прогнозируем на 1 период вперед, необходимо учитывать это уже на этапе оценки коэффициентов, не включая в модель последние m наблюдений низкочастотного ряда, которые будут использованы исключительно при построении самого прогноза.

- Во-вторых, помимо того, что $m \cdot T_y$ может быть не равно T_x (обычно низкочастотные данные публикуются позже высокочастотных), разница между двумя этими значениями может изменяться со временем.

Например, мы хотим спрогнозировать ВВП страны в зависимости от уровня безработицы в ней. Находясь в апреле 2016 года, мы обладаем данными о квартальном ВВП вплоть до $Q_4 2015$ и ежемесячными данными по безработице до *февраля 2016*. Таким образом, мы можем, используя имеющуюся информацию, прогнозировать ВВП **1 квартала** 2016 года. Через месяц, в мае, доступная информация о ВВП никак не изменится, однако, опубликуют данные по безработице в *марте*, а значит, можно обновить прогноз по ВВП, используя новую информацию.

2.3. Функции ограничений

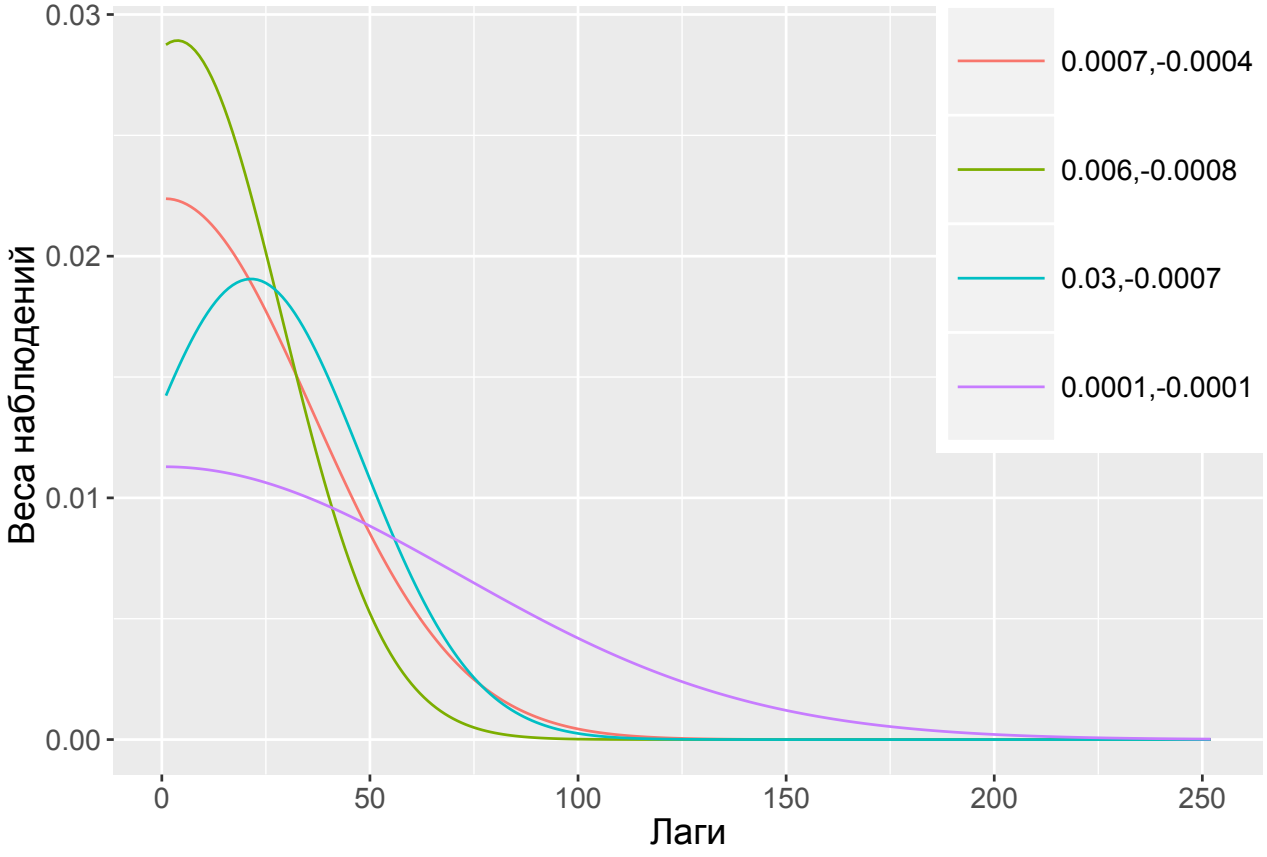
Для полной спецификации модели осталось определить вид функции $B(\theta_n, k)$. Она может принимать различные функциональные формы, которые в общем виде выглядят следующим образом:

$$\Phi(\theta_n, k) = \frac{f(\theta_n, k)}{\sum_{j=0}^{j_{max}} f(\theta_n, j)} \tag{6}$$

В данной работе используются ограничения, впервые изложенные в работах Ghysels et al. (2005, 2007) [4,6] Первая функция получила название **Экспоненциальный лаг Алмон** (Exponential Almon Lag):

$$f(\theta_n, k) = exp(\theta_n^{(1)} k + ... + \theta_n^{(q)} k^q) \tag{7}$$

Рисунок 1. Экспоненциальный лаг



Основным достоинством данной функции можно считать ее способность принимать разнообразные формы несмотря на небольшое количество параметров, которые за это отвечают. Во многих работах, в частности впервые в Ghysels, Santa-Clara, and Valkanov (2005), используется частный случай данной функции с двумя коэффициентами (при $q = 2$). Как видно из Рисунка 1[1], выбирая различные значения всего двух параметров, мы можем получать совершенно разные формы зависимостей: как убывающие с различными скоростями, так и "горбатые". Если же требуется, чтобы все коэффициенты имели равный вес, достаточно выбрать значения параметров $\theta_1 = \theta_2 = 0$. Этот случай не изображен на рисунке, однако это можно увидеть из уравнения (7)

Вторая спецификация функции ограничений также зависит только от двух параметров ($\theta_n = (\theta_n^{(1)}, \theta_n^{(2)})$) и названа авторами **Бета Лаг** (Beta Lag), так как основана на бета-функции.

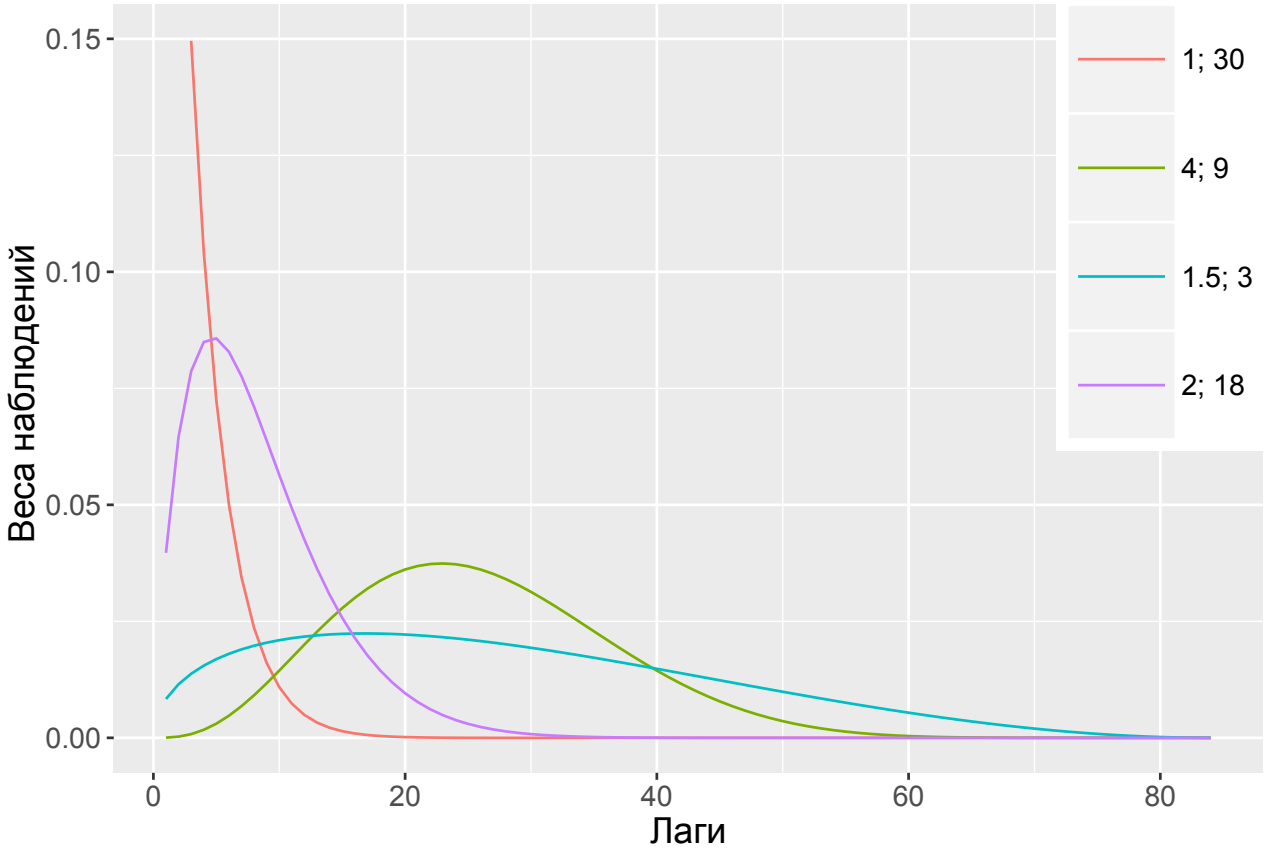
$$f(\theta_n^{(1)}, \theta_n^{(2)}, a) = \frac{a^{\theta_n^{(1)}-1}(1-a)^{\theta_n^{(2)}-1}\Gamma(\theta_n^{(1)} + \theta_n^{(2)})}{\Gamma(\theta_n^{(1)})\Gamma(\theta_n^{(2)})} \tag{8}$$

$$\Gamma(b) = \int_0^\infty e^{-x}x^{b-1}dx$$

$$a = \frac{k}{j_{max}}$$

[1]Приложение 1

Рисунок 2. Бета-функция



Также, как и в предыдущей спецификации ограничений, мы можем регулировать форму функции, изменяя значения параметров θ. Рисунок 2[2] демонстрирует несколько примеров того, какие формы может принимать данная функция ограничений. Здесь также возможен случай, когда все лаги имеют одинаковый вес. Как можно увидеть из уравнения (8), ему соответствуют значения параметров $\theta_1 = \theta_2 = 1$.

Данные функции обладают несколькими важными свойствами. Они всегда положительны и в сумме дают единицу, что позволяет использовать их именно в качестве весов для лагов высокочастотного ряда. Также можно заметить, что обе функции с определенного момента убывают и стремятся к 0 на бесконечности. Скорость убывания функции по сути определяет количество используемых лагов. Быстрое убывание соответствует небольшому количеству лагов, плавное — наоборот большому.

[2] Приложение 1

3. MIDAS в R

3.1. Первичная обработка данных

Для прогнозирования разночастотных данных с помощью моделей MIDAS можно использовать пакет *midasr*, в котором есть все необходимые для этого функции. Рассмотрим принцип их работы на примере. Возьмем данные о ежемесячной безработице с сайта Sophist.hse.ru, а также квартальный ВВП с сайта Росстат.

```
gdp <- read.csv("gdp.csv") %>% tail(-2)
unemp <- read.csv("unemp.csv")
```

Для начала, разделим имеющиеся данные на тренировочную и тестовую выборки, чтобы иметь возможность оценить качество полученных прогнозов. В тестовую выборку войдут данные по ВВП за 2014 и 2015 года (то есть 8 квартальных наблюдений). А также обрежем данные по безработице, чтобы они соответствовали тренировочной выборке по ВВП. У нас должно быть ровно в 3 раза больше наблюдений по безработице, чем по ВВП, так как мы наблюдаем ее в 3 раза чаще.

```
gdp_full <- gdp[,2]
unemp_full <- unemp[, 2]
tr_gdp <- gdp_full %>% head(-8) %>% tail(-1)
tr_unemp <- unemp_full %>%
          head(which(unemp[, 1] == "dec 2013")) %>%
          tail(-1)
trend <- 1:length(tr_gdp)
```

Для того, чтобы оценить модель, необходимо привести все данные к матричному виду. Так, если предполагаем, что Y зависит от трех своих

предыдущих значений, а также от 6 лагов высокочастотного ряда, модель в матричной форме принимает следующий вид:

$$\begin{bmatrix} Y_3 \\ Y_4 \\ \vdots \\ Y_{T_y} \end{bmatrix} = \gamma + \begin{bmatrix} Y_2 & Y_1 \\ Y_3 & Y_2 \\ \vdots & \vdots \\ Y_{T_y-1} & Y_{T_y-2} \end{bmatrix} \alpha + \begin{bmatrix} X_6 & \dots & X_1 \\ X_9 & \dots & X_4 \\ \vdots & & \vdots \\ X_{T_x} & \dots & X_{T_x-5} \end{bmatrix} \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_6 \end{bmatrix} + \begin{bmatrix} \varepsilon_3 \\ \varepsilon_4 \\ \vdots \\ \varepsilon_{T_y} \end{bmatrix} \quad (9)$$

Для того,чтобы перевести имеющийся вектор данных в матричный вид, можно воспользоваться функцией `mls(x, k,m)`, которая из вектора x делает матрицу размерности $\frac{dimx}{m} \times dimk$. Ее можно использовать для перевода высокочастотной переменной x в $dimk$ низкочастотных переменных, по $dimx/m$ наблюдений в каждой. Рассмотрим простой пример, в котором вектор из 12 высокочастотных наблюдений переведем в 3 низкосатотные переменные.

```
x <- 1:12
mls(x, 1:3, 3)
##      X.1/m X.2/m X.3/m
## [1,]    NA    NA    NA
## [2,]     5     4     3
## [3,]     8     7     6
## [4,]    11    10     9
```

Мы получили 3 низкочастотные переменные, каждая из которых состоит из 4 наблюдений. k показывает, какие именно лаги исходной переменной мы хотим включить в матрицу. В примере мы начинаем с первого лага, поэтому самое последнее наблюдение не было включено.

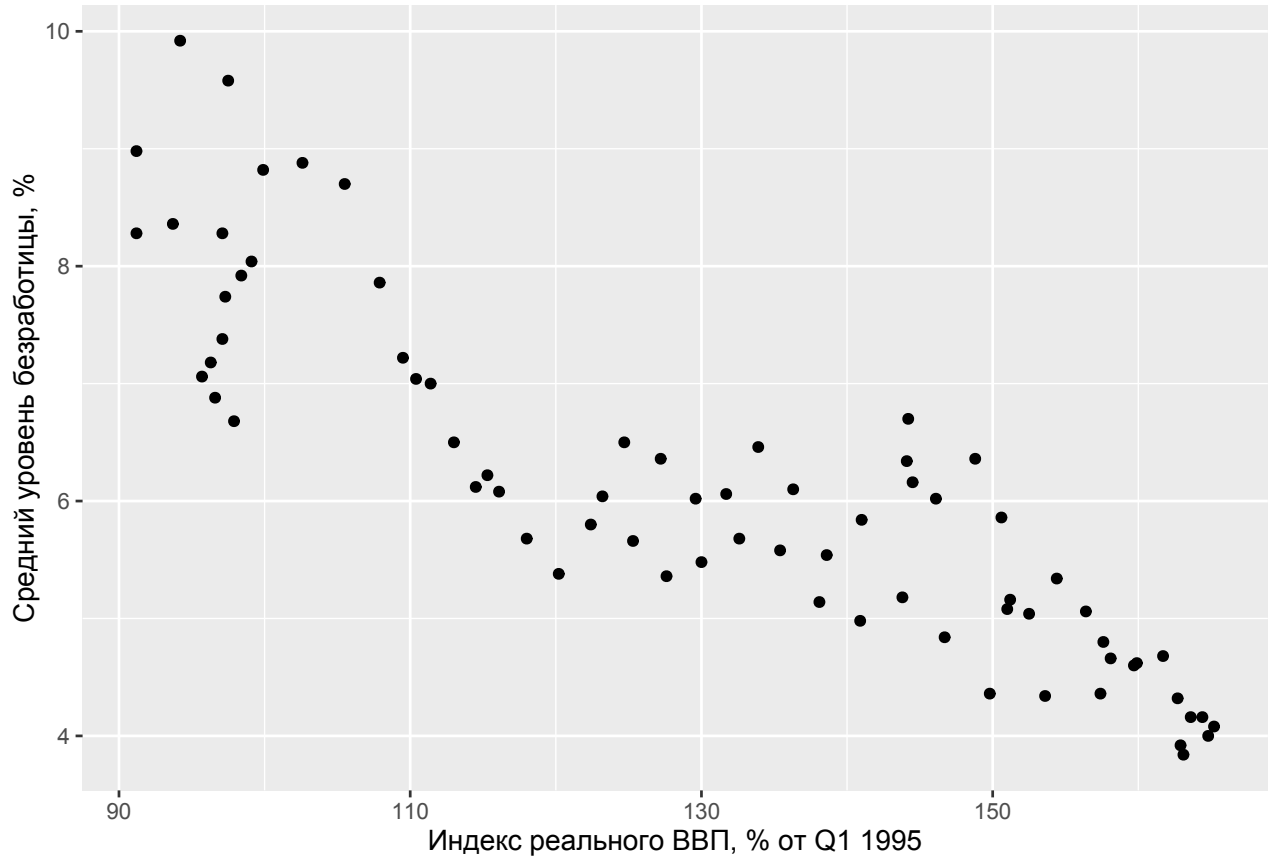
В нашем случае, если мы хотим, чтобы в модель входило 6 лагов вектора безработицы, включая самое последнее наблюдение, и два предыдущих значения ВВП, то надо построить следующие матрицы:

```
unemp_matr <- mls(tr_unemp, 0:5, 3)
gdp_matr <- mls(tr_gdp, 1:2,1)
```

Прежде чем строить регрессию, проверим наличие связи между квартальным ВВП и средней безработицей за квартал, изобразив их на графике.

```
qplot(tr_gdp, rowMeans(mls(tr_unemp, 0:4, 3)))+
  labs(list(
   title = "Рисунок 3. Зависимость между ВВП и безработицей",
    x = "Индекс реального ВВП, % от Q1 1995",
    y = "Средний уровень безработицы, %"))
```

Рисунок 3. Зависимость между ВВП и безработицей



Мы можем видеть на графике, что квартальный ВВП отрицательно зависит от среднего уровня безработицы за этот же период.

3.2. Оценка моделей: выбор ограничений и числа лагов

Мы уже определии, что существует отрицательная зависимость между ВВП и бехработицей, но для прогнозирования этого не достаточно. Для того, чтобы построить модель с разночастотными данными, удобно использовать функцию `midas_r()`. Начнем с модели без ограничений, включив в регрессию 4 и 8 лагов низкочастотного и высокочастотного рядов соответственно. Совершенно аналогичный результат мы получим, если будем использовать обычный МНК (`lm`), так как никаких ограничений на коэффициенты мы не накладываем.

```
model <- midas_r(tr_gdp ~ trend + mls(tr_gdp,1:4 ,1) +
                   mls(tr_unemp, 0:7,3), start = NULL)
tab1 <- xtable(summary(model)$coefficients,
       caption="Коэффициенты в неограниченной модели")
align(tab1) <- "XXXXX"
print(tab1,tabular.environment = "tabularx",
      width = "\\textwidth")
```

В данном случае мы можем легко интерпретировать коэффициенты, а также понять, как именно выглядит оцененная зависимость. Так, наблюдается значимая полодительная связь между текущим и предыдущим значением ВВП. А вот коэффициенты перед лагами безработицы оказались значимы только для удаленных ее значений (начиная с 5 лага) и не все

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	39.15	15.59	2.51	0.02
trend	0.26	0.13	2.00	0.05
tr_gdp1	0.97	0.24	3.96	0.00
tr_gdp2	-0.13	0.17	-0.74	0.47
tr_gdp3	-0.15	0.06	-2.42	0.02
tr_gdp4	0.01	0.07	0.15	0.88
tr_unemp1	-2.22	1.81	-1.22	0.23
tr_unemp2	1.14	2.98	0.38	0.70
tr_unemp3	-0.28	1.88	-0.15	0.88
tr_unemp4	-1.44	1.88	-0.77	0.45
tr_unemp5	5.54	2.38	2.33	0.02
tr_unemp6	-4.04	1.31	-3.09	0.00
tr_unemp7	-5.40	1.88	-2.87	0.01
tr_unemp8	5.19	1.75	2.97	0.00

Таблица 1: Коэффициенты в неограниченной модели

имеют отрицательный знак.

$$\begin{aligned}\widehat{GDP_t} = 39.148 &+ 0.256t + 0.967GDP_{t-1} - 0.126GDP_{t-2} - 0.152GDP_{t-3} + \\ &+ 0.01GDP_{t-4} - 2.221unemp_{3t} + 1.144unemp_{3t-1} - \\ &+ 0.279unemp_{3t-2} - 1.44unemp_{3t-3} + 5.545unemp_{3t-4} - \\ &- 4.042unemp_{3t-5} - 05.403unemp_{3t-6} + 5.191unemp_{3t-7}\end{aligned} \tag{10}$$

Мы получили достаточно громоздкую формулу с большим количество коэффициентов. Однако, можно значительно сократить их число, введя функцию, которая будет ограничивать коэффициенты перед высокочастотной переменной. Мы по-прежнему будем использовать 8 лагов безработицы, однако, используя команду `nealmon`, ограничим их функцией экспоненциального лага (7) и, таким образом, сократим число оцениваемых коэффициентов до двух.

```
model_r1_n <- midas_r(tr_gdp ~ trend +
                        mls(tr_gdp,1:4 ,1) +
                        mls(tr_unemp, 0:5,3, nealmon),
                        start = list(tr_unemp = rep(0,2)))
tab2 <- xtable(summary(model_r1_n)$coefficients,
       caption="Коэффициенты в ограниченной модели")
align(tab2) <- "XXXXX"
print(tab2,tabular.environment = "tabularx",
      width = "\\textwidth")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	38.59	19.60	1.97	0.05
trend	0.27	0.16	1.74	0.09
tr_gdp1	0.98	0.27	3.61	0.00
tr_gdp2	-0.16	0.16	-1.00	0.32
tr_gdp3	-0.10	0.06	-1.85	0.07
tr_gdp4	-0.03	0.08	-0.33	0.74
tr_unemp1	-1.28	0.67	-1.91	0.06
tr_unemp2	-1.09	3.56	-0.31	0.76

Таблица 2: Коэффициенты в ограниченной модели

Конечно, теперь гораздо сложнее интерпретировать значения полученных коэффициентов, так как мы видим просто параметры функции, которая и задает веса лагам высокочастотного ряда. Аналогично можно получить модель, используя в качестве ограничений Бета функцию (8). В пакете `midasr` такое ограничение получило название `nbeta`.

```
model_r1_b <- midas_r(tr_gdp ~ trend +
                        mls(tr_gdp,1:4 ,1) +
                        mls(tr_unemp, 0:5,3, nbeta),
                        start = list(tr_unemp = c(0,1,0)))
```

Заметим также, что ограничение можно накладывать не только на высокочастотную переменную, но и на лаги низкочастотной. Так, мы включили в модель 4 предыдущих значения ВВП и, введя ограничение, можем оценивать 2 коэффициента вместо 4.

```
model_r2_n <- midas_r(tr_gdp ~ trend +
                      mls(tr_gdp,1:4 ,1, nealmon) +
                      mls(tr_unemp, 0:5,3, nealmon),
                      start = list(tr_unemp = rep(0,2),
                                   tr_gdp = rep(0,2)))
model_r2_b <- midas_r(tr_gdp ~ trend +
                      mls(tr_gdp,1:4 ,1, nbeta) +
                      mls(tr_unemp, 0:5,3, nbeta),
                      start = list(tr_unemp = rep(0,3),
                                   tr_gdp = rep(0,3)))
```

Итак, мы уже оценили 4 модели и это далеко не предел. Для того, чтобы получить качественный прогноз, нам надо ответить на два вопроса относительно нашей модели:

1. Какие функции ограничении лучше всего подходят для оценки параметров модели

2. Каково оптимальное число лагов для каждой переменной

Чтобы ответить на первый вопрос, можно провести тесты на адекватность ограничений. В них проверяется нулевая гипотеза о том, что значения функции ограничений для каждого лага совпадают с истинными коэффициентами перед ними.

$$H_0 : f_\theta = \beta$$

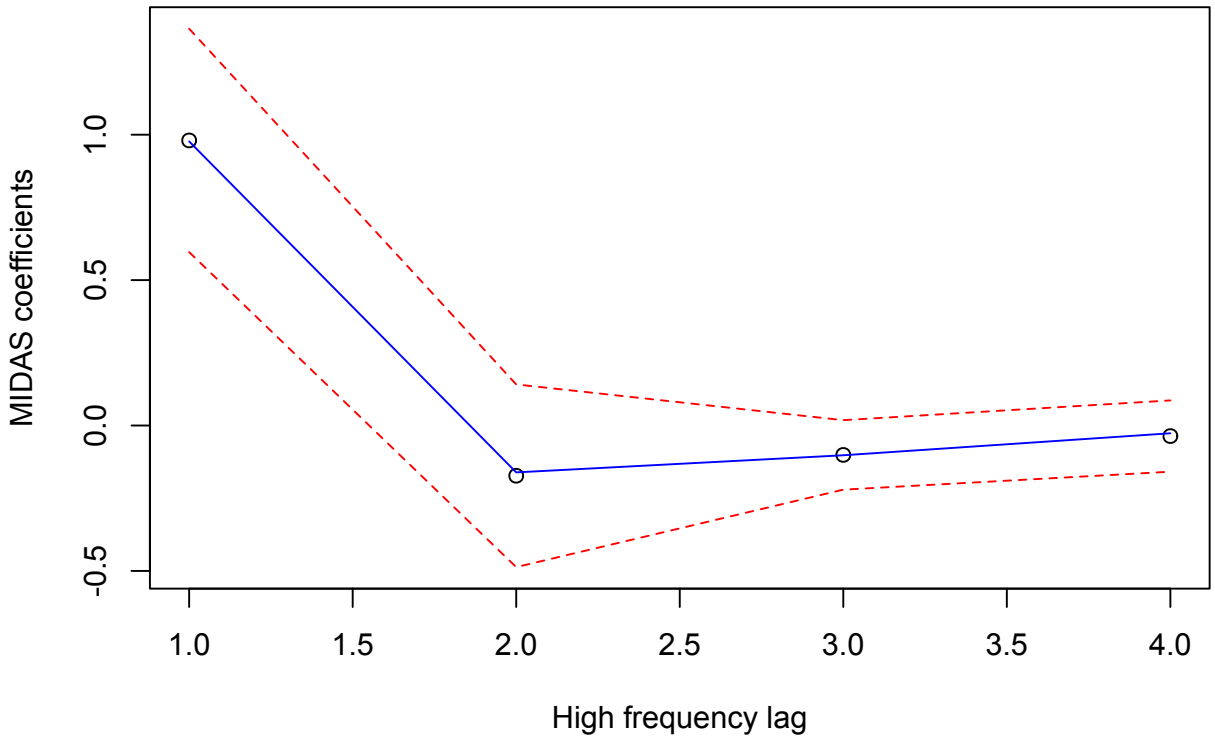
В R тест на адекватность ограничений легко реализуется с помощью команд hah_test() и hahr_test() (во втором случае используются устойчивые к гетероскедастичности и автокорреляции стандартные ошибки). Для любой модели мы получаем значение тестовой статистики, количество степеней свободы и главное — P-Value.

```
hAhr_test(model_r1_b)
##
##  hAh restriction test (robust version)
##
## data:
## hAhr = 7.2767, df = 3, p-value = 0.06358
hAhr_test(model_r1_n)
##
##  hAh restriction test (robust version)
##
## data:
## hAhr = 1.103, df = 4, p-value = 0.8938
```

Тест показал, что ограничения вида nealmon наиболее близки к истинными коэффициентам. Мы можем увидеть это, изобразив на графике коэффициенты ограниченной и неограниченной моделей.

```
plot_midas_coef(model_r1_n,
title = "Рисунок 4. Коэффициенты с ограничениями и без")
```

Рисунок 4. Коэффициенты с ограничениями и без



На Рисунке 4 видно, что оцененная функция ограничений (синяя линия) достаточно похожа на коэффициенты в неограниченной модели (черные точки) и не выходит за их 95-% доверительный интервал, что свидедельствует о том, что функция экспоненциальный лаг Алмон достаточно хорошо подходит для лагов безработицы.

Допустим, мы определились с тем, что оптимальным ограничением для коэффициентов нашей модели является функция `nealmon`. Теперь надо понять, какое количество лагов надо включить в модель. Для этого в пакете есть функция `hf_lags_table()`, в которой можно указать номер первого (минимального) лага —параметр `from`, а также количество лагов, из которого стоит выбирать — `to`.

```
nlag <- hf_lags_table(tr_gdp ~ trend +
                        mls(tr_gdp,1:2 ,1) +
```

```
                        fmls(tr_unemp, 0, 3, nealmon),
                start = list(tr_unemp = rep(0,3)),
                from = list(tr_unemp = 0),
                to = list(tr_unemp = c(3,11)))
```

Теперь мы можем посмотреть, какая модель лучше, сравнив их по Байесовскому информационному критерию (BIC) или критерию Акаике (AIC), как между собой, так и с неограниченными моделями. Напомним, что чем меньше значение информационного критерия, тем лучше выбранная модель. А считаются они по следующим формулам:

$$BIC = -2ln(L) + kln(n)$$

$$AIC = -2ln(L) + 2k$$

L - функция правдоподобия, k - количество оцениваемых коэффициентов в модели.

```
tab3 <- xtable(nlag$table[1],
                caption="Список оцененных моделей")
tab4 <- xtable(nlag$table[2:6],
                caption="Свойства оцененных моделей")
digits(tab4) <- matrix(rep(4,54), nrow = 9, ncol = 6)
```

```
align(tab3) <- "cX"
print(tab3,tabular.environment = "tabularx",
      width = "\\textwidth")
```

	model
1	tr_gdp ~ trend + mls(tr_gdp, 1:2, 1) + mls(tr_unemp, 0:3, 3, nealmon)
2	tr_gdp ~ trend + mls(tr_gdp, 1:2, 1) + mls(tr_unemp, 0:4, 3, nealmon)
3	tr_gdp ~ trend + mls(tr_gdp, 1:2, 1) + mls(tr_unemp, 0:5, 3, nealmon)
4	tr_gdp ~ trend + mls(tr_gdp, 1:2, 1) + mls(tr_unemp, 0:6, 3, nealmon)
5	tr_gdp ~ trend + mls(tr_gdp, 1:2, 1) + mls(tr_unemp, 0:7, 3, nealmon)
6	tr_gdp ~ trend + mls(tr_gdp, 1:2, 1) + mls(tr_unemp, 0:8, 3, nealmon)
7	tr_gdp ~ trend + mls(tr_gdp, 1:2, 1) + mls(tr_unemp, 0:9, 3, nealmon)
8	tr_gdp ~ trend + mls(tr_gdp, 1:2, 1) + mls(tr_unemp, 0:10, 3, nealmon)
9	tr_gdp ~ trend + mls(tr_gdp, 1:2, 1) + mls(tr_unemp, 0:11, 3, nealmon)

Таблица 3: Список оцененных моделей

```
print(tab4, scalebox = 0.9)
```

	AIC.restricted	BIC.restricted	AIC.unrestricted	BIC.unrestricted	hAh_test.p.value
1	345.7131	363.7011	347.6457	367.8822	0.8073
2	345.7039	363.6918	349.6292	372.1142	0.9680
3	345.7049	363.6929	351.4616	376.1951	0.9762
4	345.7041	363.6921	353.1342	380.1161	0.9752
5	345.7037	363.6917	351.3038	380.5343	0.5841
6	345.7039	363.6918	352.6981	384.1770	0.6462
7	345.7040	363.6920	353.9324	387.6598	0.6828
8	345.7040	363.6919	354.5920	390.5680	0.6605
9	345.7040	363.6919	353.3263	391.5508	0.4722

Таблица 4: Свойства оцененных моделей

Полученные результаты показывают, что самая лучшая модель (по обоим информационным критериям) — ограниченная модель с 8 лагами (0:7). Причем ограниченные модели во всех случаях лучше неограниченных. Помимо значений информационного критерия, мы можем видеть P-Value для теста на адекватность ограничений. Можно заметить, что оно убывает с увеличением количества лагов, но тем не менее для 5-ой модели (лучшей по информационному критерию) гипотеза не отвергается.

Но если мы не хотим отдельно выбирать функцию ограничений и количество лагов, то можно предоставить этот выбор R. Для начала

надо создать таблицу, в которой будут все варианты ограничений, лагов и необходимые начальные условия.Сделать это можно при помощи функции `expand_weights_lags()`.

```
unemp_set <- expand_weights_lags(
                          weights = c("nealmon", "nbeta"),
                           from = 0, to = c(3, 11), m = 1,
                          start = list(nealmon = rep(0,3),
                                       nbeta = rep(0,4)))
```

Далее, используя функцию `midas_r_ic_table()`, мы оцениваем все варианты моделей, которых в нашем случае $10 \times 2 = 20$, и получаем таблицу со всеми формулами, значениями информационных критериев и P-Value для проверки гипотезы об адекватности ограничений. Стоит заметить, что здесь уже не требуется указывать число лагов и тип ограничений внутри `mls()`, так как любая информация будет заменена данными из таблицы `unemp_set`, полученной выше.

```
models_ic <- midas_r_ic_table(tr_gdp ~ trend +
                                    mls(tr_gdp, 1:2, m = 1) +
                                     mls(tr_unemp, 0, m = 3),
                          table = list(tr_unemp = unemp_set))
```

Так как выбирать вручную из 20 моделей не интересно, более того, моделей может быть гораздо больше, можно воспользоваться функцией `modsel()`, которая выбирает модель по нужному критерию и показывает всю информацию о ней. Параметр `print` отвечает за то, нужно ли выводить всю информацию о выбранной модели на экран сразу.

```
choice <- modsel(models_ic, IC = "AIC", print = FALSE,
       type = c('restricted', 'unrestricted'))
```

```
choice_info <- summary(choice)
```

Теперь мы можем посмотреть, какая именная модель была признана наиболее оптимальной по критерию Акаике.

```
choice_info$formula
```

```
## tr_gdp ~ trend + mls(tr_gdp, 1:2, m = 1) + mls(tr_unemp, 0:7,
##     m = 3, nealmon)
## <environment: 0x000000000dde66d0>
```

```
tab5 <- xtable(choice_info$coefficients,
               caption="Коэффициенты выбранной модели")
align(tab5) <- "XXXXX"
print(tab5,tabular.environment = "tabularx",
      width = "\\textwidth")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	36.02	15.22	2.37	0.02
trend	0.24	0.13	1.88	0.07
tr_gdp1	1.03	0.28	3.67	0.00
tr_gdp2	-0.31	0.16	-1.96	0.05
tr_unemp1	-1.27	0.53	-2.41	0.02
tr_unemp2	7.62	2047.26	0.00	1.00
tr_unemp3	-2.78	681.45	-0.00	1.00

Таблица 5: Коэффициенты выбранной модели

Мы получили тот же самый результат — лучшей моделью для зависимости ВВП и безработицы по критерию Акаике является ограниченная модель (функция ограничений `nealmon`) с 8 лагами безработицы.

3.3. Прогноз и сравнение моделей

Итак, сравнив различные варианты спецификаций модели, мы получили оптимальную модель для зависимости ВВП и безработицы:

$$GDP_t = 36.02 + 0.24t + 1.03GDP_{t-1} - 0.31GDP_{t-2} + \sum_{j=0}^{7} \beta_j unemp_{3t-j} \quad (11)$$

Значит, если мы хотим получить прогноз на 1 период вперед, то есть $\widehat{GDP}_{T+1|T}$, нам необходимо знать значение ВВП двух предыдущих кварталов, а также следующие значения безработицы: $unemp_{3T+3}, unemp_{3T+2}, \ldots, unemp$

Функция `forecast()` предполагает, что модель, для которой мы считаем прогноз, была оценена по данным известным вплоть до момента T, а все, что известно после этого момента, находится в параметре `newdata`. Далее, на основании всех доступных данных строится прогноз.

Возьмем новые данные по безработице и спрогнозирует следующее значение ВВП.

```
unemp_new <- unemp_full %>%
             tail(-which(unemp[,1] == "dec 2013")) %>%
             head(3)
model_opt <- midas_r(tr_gdp ~ trend +
                       mls(tr_gdp, 1:2, m = 1) +
                      mls(tr_unemp, 0:7, m = 3, nealmon),
                      start = list(tr_unemp = rep(0,2)))
forecast(model_opt,
         newdata = list(trend = length(trend)+1,
                        tr_unemp = unemp_new))
##   Point Forecast
## 1        167.192
```

Может быть такое, что для построения прогноза не требуются новые данные. Это происходит в том случае, если часть высокочастотных лагов не участвовала в оценке коэффициентов. То есть достаточно сдвинуть лаги безработицы в модели (11) на 1 низкочастотный или 3 высокочастотных периода, и мы может получить прогноз на 1 квартал, указав в `newdata` только следующее значение тренда.

$$GDP_t = \gamma_1 + \gamma_2 t + \alpha_1 GDP_{t-1} + \alpha_2 GDP_{t-2} + \sum_{j=3+0}^{3+7} \beta_j unemp_{3t-j} \qquad (12)$$

```
model_opt1 <- midas_r(tr_gdp ~ trend +
                        mls(tr_gdp, 1:2, m = 1) +
                      mls(tr_unemp, 3+0:7, m = 3, nealmon),
                        start = list(tr_unemp = rep(0,2)))
forecast(model_opt1,
         newdata = list(trend = length(trend)+1,
                        tr_unemp = rep(NA, 3)))
##   Point Forecast
## 1       167.0727
```

Пакет `midasr` предоставляет также возможность получить прогнозы сразу по нескольким моделям, а затем сравнить их по прогнозной силе на тестовой выборке. Для таких целей можно использовать функцию `select_and_forecast()`. В общем случае она осуществляет выбор лучшей модели для каждого горизонта прогноза среди определенного количества моделей с различными ограничениями и лагами. Однако, ее можно использовать и для того, чтобы просто получить качество прогноза для

конкретной модели.

В целом, процесс выбора лучшей модели для каждого горизонта прогноза происходит аналогично тому, как это было описано выше. Однако, нам необходимо специфицировать набор моделей для каждого горизонта `h = 0, 1, 2,...`. Главное отличие появляется в лагах высокочастотных рядов. Минимальный порядок лага, включенного в модель с горизонтом 1, должен быть равен m, а с горизонтом 2 — 2m, и так далее. За минимальное значение лагов отвечает опция `from`, которая представляет собой список векторов для каждой высокочастотной переменной. Помимо этого необходимо указать разброс лагов. За это отвечает параметр `to`, который представляет собой матрицу с минимальным и максимальным номером последнего включаемого в модель лага для каждого горизонта h.

Например, если в нашей модели с ВВП и безработицей мы хотим получить прогноз на 1, 2 и 3 периода вперед. Тогда минимальные лаги, включенные в модель должны иметь номера `(3,6,9)`. Далее, допустим, что мы предполагаем включение от 6 до 12 лагов, тогда мы должны указать следующие верхние границы: `(8,14), (11,17), (14, 20)`.

Помимо этого надо указать какая часть выборки является трениpововной, а какая — тестовой. За это отвечают параметры `insample` И `outsample`.

Также данная функция предоставляет возможность получать не только прогнозы отдельно по каждой модели, но и взвешанный прогноз по всем моделям для каждого значения горизонта прогноза. В статье Ghysels (2013) была описано 4 основных способа взвешивания прогнозов, каждый из которых представлен в функции `select_and_forecast`. Взвешанный прогноз получется следующим образом:

$$Y^F_{T+h|T} = \sum_{i=1}^{n} \omega_{i,T} \cdot \widehat{Y}_{i,T+h|T} \tag{13}$$

Веса прогнозов для (13) могут определяться следующим образом:

- EW — equally weighted

$$\omega_{i,t} = \frac{1}{n}$$

- BICW — взвешивание на основании критерия BIC.

$$\omega_{i,t} = \frac{exp(-BIC_i)}{\sum_j exp(-BIC_j)}$$

- MSFE (DMSFE) - взвешивание на основании суммы квадратов ошибок проноза.

$$\omega_{i,t} = \frac{m_{i,t}^{-1}}{\sum_j m_{j,t}^{-1}}$$

$$m_{i,t} = \sum_{i=T_0}^{T} \delta^{T-i}(Y_{s+h} - \widehat{Y}_{i,s+h|s})^2$$

$$\delta = \begin{cases} 1, MSFE \\ 0.9, DNSFE \end{cases}$$

Исследования показывают, что использование взвешанных прогнозов улучшает их качество. Однако, нет единого мнения на счет того, какой именно способ взвешивания является оптимальным.

```
trend_full <- 1:82
forecasts <- select_and_forecast(
  gdp_full ~ trend_full + mls(gdp_full, 1:2, 1) +
                          mls(unemp_full, 0, 3),
```

```
  from = list(unemp_full = c(3, 6, 9)),
 to= list(unemp_full = rbind(c(8,14), c(11,17), c(14,20))),
  insample = 1:73, outsample = 74:82,
  weights = list(unemp_full = c("nealmon", "nbeta")),
  wstart = list(nealmon = rep(0, 2), nbeta = rep(0, 3)),
  IC = "AIC",
  measures = c("MSE", "MAPE", "MASE"),
  fweights = c("EW", "BICW", "MSFE", "DMSFE"),
  ftype = 'recursive')
```

В `forecasts` сейчас хранится информация по лучшим моделям для каждого горизонта прогноза и по каждому ограничению, то есть по 6 моделям. Мы можем посмотреть на качество прогноза по каждой из них.

```
tab6 <- xtable(forecasts$accuracy$individual[1],
       caption="Список выбранных моделей")
align(tab6) <- "cX"
print(tab6,tabular.environment = "tabularx",
      width = "\\textwidth")
```

	Model
1	gdp_full ~ trend_full + mls(gdp_full, 1:2, 1) + mls(unemp_full, 3:8, 3, nealmon)
2	gdp_full ~ trend_full + mls(gdp_full, 1:2, 1) + mls(unemp_full, 3:14, 3, nbeta)
3	gdp_full ~ trend_full + mls(gdp_full, 1:2, 1) + mls(unemp_full, 6:15, 3, nealmon)
4	gdp_full ~ trend_full + mls(gdp_full, 1:2, 1) + mls(unemp_full, 6:15, 3, nbeta)
5	gdp_full ~ trend_full + mls(gdp_full, 1:2, 1) + mls(unemp_full, 9:15, 3, nealmon)
6	gdp_full ~ trend_full + mls(gdp_full, 1:2, 1) + mls(unemp_full, 9:15, 3, nbeta)

Таблица 6: Список выбранных моделей

```
tab7 <- xtable(forecasts$accuracy$individual[, 2:4],
       caption="Качество прогноза в выбранных моделях")
align(tab7) <- "cXXX"
print(tab7,tabular.environment = "tabularx",
      width = "\\textwidth")
```

	MSE.out.of.sample	MAPE.out.of.sample	MASE.out.of.sample
1	8.07	1.66	3.09
2	7.88	1.65	3.07
3	7.78	1.63	3.04
4	8.40	1.70	3.16
5	7.78	1.63	3.04
6	7.84	1.64	3.05

Таблица 7: Качество прогноза в выбранных моделях

Можно увидеть, что разброс значений не велик. То есть модели не сильно отличаются по качеству прогноза. Тем не менее, минимальные ошибки по всем трем показателям у моделей 3 и 5.

Помимио этого, можно сравнить результаты отдельных моделей с результатами взвешанного прогноза. Так как у нас было 6 моделей с тремя различными горизонтами, каждый взвешанный прогноз будет состоять из двух моделей (с одинаковым горизонтом). В параметрах функции `select_and_forecast` мы указали все 4 способа взвешивания, значит таблица бцдет состоять из характеристик двенадцати различных прогнозов.

```
tab8 <- xtable(forecasts$accuracy$average,
       caption="Качество взвешанных прогнозов")
align(tab8) <- "cXXXXX"
print(tab8,tabular.environment = "tabularx",
      width = "\\textwidth")
```

	Horizon	Scheme	MSE	MAPE	MASE
1	1	EW	7.97	1.66	3.08
2	1	BICW	8.06	1.66	3.09
3	1	MSFE	7.97	1.66	3.08
4	1	DMSFE	7.97	1.66	3.08
5	2	EW	8.08	1.66	3.10
6	2	BICW	7.81	1.64	3.04
7	2	MSFE	8.07	1.66	3.10
8	2	DMSFE	8.07	1.66	3.10
9	3	EW	7.81	1.64	3.05
10	3	BICW	7.78	1.63	3.04
11	3	MSFE	7.81	1.64	3.05
12	3	DMSFE	7.81	1.64	3.05

Таблица 8: Качество взвешанных прогнозов

Если же у нас нет необхожимости сравнивать и выбирать из большого числа моделей, можно получить похожий результат, воспользовавшишь функцией `average_forecast()`. С помощью нее можно получить прогнозы по одной или нескольким моделям, взвешанные прогнозы по всем выбранным моделям, а также их качество.

```
newmod1 <- midas_r(gdp_full ~ trend_full  +
                           mls(gdp_full, 1:2, m = 1) +
                       mls(unemp_full, 3:8, m = 3, nealmon),
                        start = list(unemp_full = rep(0,2)))
newmod2 <- midas_r(gdp_full ~ trend_full  +
                           mls(gdp_full, 1:2, m = 1) +
                       mls(unemp_full, 3:15, m = 3, nbeta),
                       start = list(unemp_full = rep(0,3)))
```

```
avf <- average_forecast(
                   list(newmod1, newmod2),
                   data = list(trend_full = trend_full,
```

```
                                    gdp_full = gdp_full,
                                    unemp_full = unemp_full),
                        insample = 1:73, outsample = 74:82,
                        type = "recursive",
                        measures = c("MSE", "MAPE", "MASE"),
                    fweights = c("EW", "BICW", "MSFE", "DMSFE"),
                        show_progress = FALSE)
```

```
tab9 <- xtable(avf$accuracy$individual[, 2:4],
       caption="Качество прогноза всех моделей")
align(tab9) <- "cXXX"
print(tab9,tabular.environment = "tabularx",
      width = "\\textwidth")
```

	MSE.out.of.sample	MAPE.out.of.sample	MASE.out.of.sample
1	8.07	1.66	3.09
2	7.93	1.66	3.08

Таблица 9: Качество прогноза всех моделей

В данном случае нельзя однозначно сказать, какая из двух моделей лучше прогнозирует. Как мы видим, средняя квадратичная ошибка (MSE) и средняя абсолютная масштабированная ошибка (MASE) показывают противоположные результаты. В то же время средняя абсолютная процентная ошибка (MAPE) у обоих прогнозов одинаковая.

Если же не выбирать из двух, а использовать взвешанный прогноз по обеим моделям, мы получим следующие результаты:

```
tab10 <- xtable(avf$accuracy$average,
       caption="Качество взвешанного прогноза")
align(tab10) <- "cXXXX"
```

```
print(tab10, tabular.environment = "tabularx",
        width = "\\textwidth")
```

	Scheme	MSE	MAPE	MASE
1	EW	8.00	1.66	3.09
2	BICW	7.93	1.66	3.08
3	MSFE	8.00	1.66	3.09
4	DMSFE	8.00	1.66	3.09

Таблица 10: Качество взвешанного прогноза

4. Прогнозирование Российских макроэкономических данных

4.1. Данные

Для применения описанных выше моделей были выбраны следующие квартальные переменные:

- **gdp** — индекс реального ВВП, % (1 квартал 1995 года — 100%)
- **inv** — индекс рельных инвестиций в основной капитал, % (1 квартал 1993 года — 100%)

Следующие высокочастотные переменные, с ежемесячными наблюдениями, были выбраны объясняющими.

- **unemp** — беработица, %
- **cpi** — индекс потребительский цен, в % к предыдущему году
- **bci** — business confidence index, рассчитывается ОЭСР на основании текущих состояний компаний и прогнозов относительно ближайшего будущего, 100 соответствует среднему по всем странам значению
- **oil** — цена нефти марки Brent, $

В модели включалось 3 лага низкочастотно переменной, и от 6 до 9 лагов высокочастотных.В тренировочную выборку по квартальным переменным бли включены данные вплоть до 4 квартала 2014 года, в тестовую - первые 3 квартала 2015 года. Что касается высокочастотных данных, их тестовая выборка имеет три различные конфигурации: данные до февраля 2015, до марта 2015 и до апреля 2015. Это связано с различной часто-

той публикации данных в реальной жизни. Квартальные данные последнего периода года становятся доступны только в апреле следующего года, к этому моменту месячные показатели доступны вплоть до февраля. Следущие квартальные данные (за 1 квартал текущего года) будут опубликованы только в июле. Тем не менее в мае и июне доступные месячные показатели будут обновляться. Отсюда можно сделать два вывода:

- Есть необходимость пронозировать не только будущие значение квартальных данных, но и показатель предыдущего и текущего квартала, так как пукликация происходит с задержкой

- Состав данных, которыми располагает исследователь зависит от того, в каком месяце квартала он находится

4.2. Методы прогнозирования

Для сравнения были выбраны прогнозы по 8 различным моделям. Для каждого класса моделей была выбрана спецификация с минимальным MSE (mean squred error) для тренировочной выборки. Далее - посчитано значение MSE для тестовой выборки. В разделе результаты для каждой зависимой переменной представлены результаты 5 лучших моделей в зависимости от того, в каком месяце квартала производился прогноз. Рассмотрим подробнее выбранные модели.

Для прогнозирования ВВП и инвестиций было использовано 4 различных спецификации **MIDAS**:

- **N**(Nealmon) — ограничения вида экспоненциальный лаг Алмон (7) для всех высокочастотных переменных, модель с минимальным MSE на тренировочной выборке

- **NW**(Nealmon weighted) — средневзвешанный (EW) прогноз по всем моделям Nealmon
- **B**(Beta) — ограничения вида Бета (8) для всех высокочастотных переменных, модель с минимальным MSE на тренировочной выборке
- **BW**(Beta weighted) — средневзвешанный (EW) прогноз по всем моделям с ограничением Бета функции

Для сравнения, были также оценены 4 более простые модели:

- **SW**(Step Weightning) — неограниченная модель (2), модель с минимальным MSE на тренировочной выборке
- **TA**(Time Averaging) — агрегирование данных с помощью среднего арифметического (3), модель с минимальным MSE на тренировочной выборке
- **ARIMA** — авторегресионная модель со скользящим средним
- **AR** — авторегрегресионная модель

В двух последних случаях результат не зависит от номера месяца, так как высокочастотне данные не включены в модель, то есть доступная информация не изменяется внутри одного квартала.

4.3. Результаты

В таблицах ниже представлены результаты, полученные при прогнозировании квартальных инвестиций и ВВП с использованием их лагов, а также лагов таких высокочастотных переменных как уровень цен, индекс деловой уверенности, цена нефти и уровень безработицы. В таблице 11 содержатся пять моделей с минимальной среднеквадратичной ошибкой при прогнозировании инвестиций для каждого из трех месяцев квартала.

	Месяц 1	Месяц 2	Месяц 3
MSE	56.856 NW	22.121 BW	54.805 N
	65.573 N	22.854 B	70.483 NW
	129.746 AR	64.767 NW	129.746 AR
	202.566 TA	66.38 N	204.556 TA
	509.851 BW	129.746 AR	218.849 B

Таблица 11: Качество прогноза, инвестиции

Мы можем видеть, что во всех трех случаях MIDAS модели показали наилучий результат. Из 12 выбанных для сравнения моделей, в таблицу вошли 10. Причем почти всегда взвешанный прогноз по всем моделям оказался лучше прогноза одной, выбранной по тестовой выборке, модели. Помимо моделей MIDAS в список лучших вошла простая авторегрессия и модель со средневзвешанными по времени высокочастотными данными.

Второй переменной, для которой был построен прогноз является ВВП. Результаты мы можем наблюдать в таблице 12.

	Месяц 1	Месяц 2	Месяц 3
MSE	2.883 AR	2.883 AR	2.883 AR
	4.972 NW	4.778 N	3.587 N
	6.4418 N	4.806 NW	3.971 NW
	21.589 $ARIMA$	21.5895 $ARIMA$	20.734 BW
	21.635 BW	22.121 BW	21.589 $ARIMA$

Таблица 12: Качество прогноза, ВВП

В этом случае высокие результаты показали авторегрессионные модели: AR и ARIMA вошли в число лучших для каждого месяца. Тем не менее, MIDAS модели показали достаточно высокие результаты, так как в каждом из месяцев 3 из 4 вошли в списоклучших по среднеквадратичной ошибке.

Все расчеты можно найти в репозитории на GitHub.

5. Выводы

Модели MIDAS предоставляют широкие возможности для построения моделей с данными разной частоты. В данной работы был описан принцип работы таких моделей, который заключается в том, что на коэффициенты перед лагами высокочастотных переменных накладывается определенное функциональное ограничение. Гибкость данных функций способствует тому, что качество прогнозов не страдает, в то время как количество параметров, которые необходимо оценить, снижается

Для применения моделей к реальным данным крайне удобно использовать пакет `midasr`. В работе были описаны его основные функции, с помощью которых можно оценивать модели, выбирать виды ограничений и количество лагов. Также он обладает широким набором инструментов для прогнозирования моделей с данными разной частоты и оценки качества полученных прогнозов.

Для сравнения моделей MIDAS с такими моделями как ARIMA, AR, Time Averaging и Step Weightning, были оценены модели для инвестиций и ВВП в зависимости от безработицы, цен на нефть, курса доллара и индекса деловой уверенности. MIDAS модели показали высокие результаты, однако, нельзя однозначно утверждать, что они являются лучшим инструментом в данной сфере.

6. Приложения

Приложение 1а. Построение графика функции экпоненциальный лаг Алмон

```
lags_exp <- seq(1,252, 1)
len_exp <- length(lags_exp)
exp_func <- function(k, theta_1, theta_2){
  f = exp(theta_1*k+theta_2*(k^2))
  return(f)
}
restriction_exp <- function(k, theta_1, theta_2){
  inter = 0
  for (i in 1:len_exp){
   inter = inter + exp_func(lags_exp[i], theta_1, theta_2)
  }
  return(exp_func(k, theta_1, theta_2)/inter)
}
weights_exp <- data.frame(lags_exp = lags_exp,
                          weight_1 = rep(NA, len_exp),
                          weight_2 = rep(NA, len_exp),
                          weight_3 = rep(NA, len_exp),
                          weight_4 = rep(NA, len_exp))
for (i in 1:len_exp){
  weights_exp[i,2] <-
    restriction_exp(lags_exp[i], 0.0007,-0.0004)
  weights_exp[i,3] <-
    restriction_exp(lags_exp[i], 0.006,-0.0008)
```

```
  weights_exp[i,4] <-
    restriction_exp(lags_exp[i], 0.03,-0.0007)
  weights_exp[i,5] <-
    restriction_exp(lags_exp[i], 0.0001,-0.0001)
}
weight_long_exp <- melt(weights_exp,
                        value.name = "weight",
                        measure.vars = c("weight_1",
                                         "weight_2",
                                         "weight_3",
                                         "weight_4"))
qplot(data = weight_long_exp, x = lags_exp,
      y = weight,color = variable, geom = "line") +
  theme(legend.position = c(0.86, 0.74),
        legend.key.size = unit(1.5,"cm"),
        text = element_text(size = 14))+
  labs(list(title = "Рисунок 1. Экспоненциальный лаг",
            x = "Лаги", y = "Веса наблюдений "))+
  scale_colour_discrete(name="",
                        breaks=c("weight_1", "weight_2",
                                 "weight_3","weight_4"),
                        labels=c("0.0007,-0.0004",
                                 "0.006,-0.0008",
                                 "0.03,-0.0007",
                                 "0.0001,-0.0001"))
```

Приложение 1b. Построение графика Бета функции

```
lags <- seq(1,84, 1)
len <- length(lags)
beta_func <- function(k, theta_1, theta_2){
  a = k/len
 f = (a^(theta_1-1)*(1-a)^(theta_2-1))/beta(theta_1, theta_2)
  return(f)
}
restriction <- function(k, theta_1, theta_2){
  inter = 0
  for (i in 1:len){
    inter = inter + beta_func(lags[i], theta_1, theta_2)
  }
  return(beta_func(k, theta_1, theta_2)/inter)
}
weights <- data.frame(lags = lags, weight_1 = rep(NA, len),
                      weight_2 = rep(NA, len),
                      weight_3 = rep(NA, len),
                      weight_4 = rep(NA, len))
for (i in 1:len){
  weights[i,2] <- restriction(lags[i], 1,30)
  weights[i,3] <- restriction(lags[i], 4,9)
  weights[i,4] <- restriction(lags[i], 1.5,3)
  weights[i,5] <- restriction(lags[i], 2,18)
}
weight_long <- melt(weights, value.name = "weight",
                    measure.vars = c("weight_1",
```

```
                                        "weight_2",
                                        "weight_3",
                                        "weight_4"))
qplot(data = weight_long, x = lags,
      y = weight,color = variable, geom = "line") +
  theme(legend.position = c(0.91, 0.74),
        legend.key.size = unit(1.5,"cm"),
        text = element_text(size = 14))+
  labs(list(title = "Рисунок 2. Бета-функция",
            x = "Лаги", y = "Веса наблюдений "))+
  scale_colour_discrete(name="",
                        breaks=c("weight_1", "weight_2",
                                 "weight_3","weight_4"),
                        labels=c("1; 30","4; 9",
                                 "1.5; 3", "2; 18"))+
  ylim(0,0.15)
```

Список литературы

[1] Armesto M. T., Engemann K. M., Owyang M. T. Forecasting with mixed frequencies //Review. – 2010. – Т. 92.

[2] Clements M. P., Galvão A. B. Forecasting US output growth using leading indicators: An appraisal using MIDAS models //Journal of Applied Econometrics. – 2009. – Т. 24. – №. 7. – С. 1187-1206.

[3] Ghysels E., Santa-Clara P., Valkanov R. The MIDAS touch: Mixed data sampling regression models //Finance. – 2004.

[4] Ghysels E., Santa-Clara P., Valkanov R. There is a risk-return trade-off after all //Journal of Financial Economics. – 2005. – Т. 76. – №. 3. – С. 509-548.

[5] Ghysels G., Valkanov R. Linear time series processes with mixed data sampling and MIDAS regression models //Social science research network. – 2006.

[6] Ghysels E., Sinko A., Valkanov R. MIDAS regressions: Further results and new directions //Econometric Reviews. – 2007. – Т. 26. – №. 1. – С. 53-90.

[7] Ghysels E. Matlab Toolbox for Mixed Sampling Frequency Data Analysis using MIDAS Regression Models //Unpublished Manuscript. – 2013.

[8] Kuzin V., Marcellino M., Schumacher C. MIDAS vs. mixed-frequency VAR: Nowcasting GDP in the euro area //International Journal of Forecasting. – 2011. – Т. 27. – №. 2. – С. 529-542.