АНАЛИЗ ПОСЛЕДОВАТЕЛЬНОСТЕЙ В R: TraMineR

БИРЮКОВА СВЕТЛАНА
Н.С. ЦЕНТРА АНАЛИЗА ДОХОДОВ И УРОВНЯ ЖИЗНИ НИУ ВШЭ

10 ОКТЯБРЯ 2014

СЕМИНАР ПРОВОДИТСЯ В РАМКАХ РАБОТЫ НАУЧНО-УЧЕБНОЙ ГРУППЫ «ИЗУЧЕНИЕ РОЖДАЕМОСТИ, ФОРМИРОВАНИЯ И РАЗВИТИЯ СЕМЕЙ НА ДАННЫХ ВЫБОРОЧНЫХ ОБСЛЕДОВАНИЙ»

ПЛАН СЕМИНАРА

Часть 1

- Вступление 1: R
- Вступление 2: TraMineR

- TraMineR — Дескриптивный анализ

Часть 2

- TraMineR — Регрессионный анализ

- Где поучиться R или почитать про TraMineR?
- Вместо авторских прав

ВСТУПЛЕНИЕ 1: R

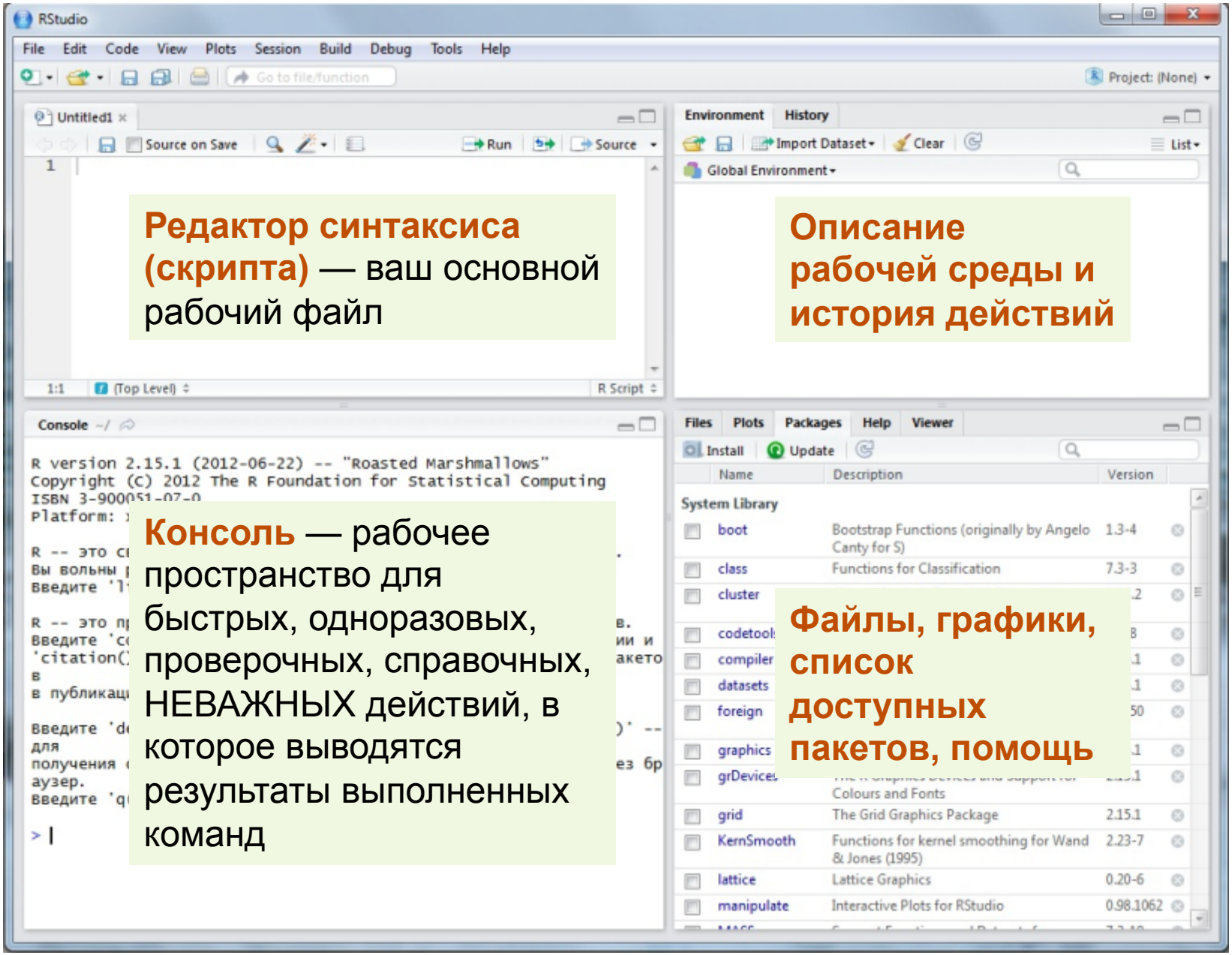
ЧТО ТАКОЕ R?

- R — это программная среда для анализа, обработки и визуализации данных с открытым исходным кодом
- В основе лежит язык программирования S
- Широко используется в области эпидемиологии, медицины, биоинформатики, социологии

В ЧЕМ ОСНОВНЫЕ ПРЕИМУЩЕСТВА R?

- Бесплатное распространение
- Огромная команда разработчиков, система общественного контроля и верификации кода ⇒ очень быстрые темпы развития
- Много возможностей получить помощь при работе с пакетом
  - http://www.r-project.org/, http://cran.r-project.org/ (Manuals, Books, FAQs, Mailing lists…)
  - StackExchange, StackOverflow, CrossValidated Вопросы и ответы по тегам R и TraMineR

НАЧАЛО РАБОТЫ: РАБОЧАЯ СРЕДА R R-STUDIO



ПЕРВЫЕ ШАГИ

- Как быстро переключаться между окном скрипта, консолью и рабочей средой? Попробуйте сочетания клавиш Ctrl+1, Ctrl+2, Ctrl+4, Ctrl+5.
- Как вообще здесь что-то делать? Ничего не понятно. Где кнопки для анализа, графиков? Кнопок нет, все, что вы хотите сделать, нужно писать в консоли, а лучше – сразу сохранять в окне со скриптом. Например, поставьте курсор в окно консоли и напечатайте в нем

```
help()
```

- Чтобы экономить свое время и писать сразу в скрипт, не копируя в консоль или наоборот, используйте сочетание клавиш Ctrl +Enter. Например, перейдите в окно скрипта, наберите там `print("hello!")` и нажмите Ctrl+Enter.
- Как и в любой другой программе, в ситнаксисе вы можете оставлять для себя комментарии. В R комментарии помечаются символом # в начале строки.

ПЕРВЫЕ ШАГИ

- Основные две категории, с которыми вам нужно будет работать, — это объекты и функции.
  - Объектом может быть набор данных (матрица), одно наблюдение (строка матрицы), один параметр (столбец матрицы), один параметр одного объекта (конкретная характеристика одного наблюдения), одно число, текст, вектор значений…

    Объекты создаются при помощи оператора <-

    ```
    v <- c(1, 2, 4, 8)
    ```

  - Функции могут быть встроенными в пакет или заданными вами.

- Регистр в именах объектов и команд в R имеет значение. Ggs07 b ggs07 программа будет считать разными объектами.
- Что такое пакеты, зачем они нужны и как их подключать? Пакеты – специально разработанные для решения специфических задач наборы функций. Подключать их можно через меню Packages в правом нижнем окне (если они там есть) или командой вида:

```
install.packages("TraMineR", dependencies = TRUE)
```

ЧТО НАМ ПОНАДОБИТСЯ СЕГОДНЯ?

Vectors

```
x[n]              nth element
x[-n]             all but the nth element
x[1:n]            first n elements
x[-(1:n)]         elements from n+1 to the end
x[c(1,4,2)]       specific elements
x["name"]         element named "name"
x[x > 3]          all elements greater than 3
x[x > 3 & x < 5]  all elements between 3 and 5
x[x %in% c("a","and","the")] elements in the given set
```

Matrices

```
x[i,j]            element at row i, column j
x[i,]             row i
x[,j]             column j
x[,c(1,3)]        columns 1 and 3
x["name",]        row named "name"
```

Data frames (same as matrix plus the following)

```
x[["name"]]       column named "name"
x$name            idem
```

ВСТУПЛЕНИЕ 2: TRAMINER

УСТАНОВКА ПАКЕТА

- Установка через меню:

  Packages ➔Install Packages ➔Install from Repository (CRAN)

- Установка через консоль:

```
install.packages("TraMineR", dependencies = TRUE)
install.packages("TraMineRextras", dependencies = TRUE)
install.packages("WeightedCluster", dependencies = TRUE)
```

ЧТО ТАКОЕ TRAMINER?

TraMineR = Trajectory Miner in R

Это надстройка (пакет), разработанный специально для анализа последовательностей событий в R в Женевском университете под руководством Жильбера Ришара.

Официальный сайт пакета со всей информацией по нему: http://mephisto.unige.ch/traminer/

Полный перечень функций, доступных в пакете можно увидеть, вызвав команду

```
?TraMineR
help(TraMineR)
```

По сути у TraMineR есть целый набор аналогов в других ПО, но этот пока что является самым полным, самым развивающимся и самым бесплатным.

Аналоги: SADI & SQ (в Stata), CHESA, TDA

ЧТО МОЖЕТ TRAMINER?

- При работе с последовательностью состояний:
  - Инструментарий для описания (вычисления) и графического изображения основных параметров отдельных последовательностей
  - Инструментарий для вычисления попарных различий между последовательностями (расстояний), которые дают возможность затем перейти к использованию традиционных математических методов
    - Выделению кластеров последовательностей
    - Анализу отличий (ANOVA и регрессионные графы)
    - Выделению репрезентативных последовательностей (траекторий)
- При работе с последовательностью событий:
  - Инструментарий для поиска наиболее частых подпоследовательностей событий и дискриминирующих переходов

ЧТО МОЖЕТ TRAMINER?

- Графически изображать последовательности
  распределение состояний, распределение последовательностей, простое изображение последовательностей…
- Вычислять характеристики последовательностей
  длина, время пребывания в каждом из состояний, энтропия, сложность последовательности (complexity), разнообразие переходов (turbulence)….
- Вычислять характеристики момента времени (вертикальной последовательности)
  распределение состояний, модальное состояние, вертикальная энтропия…
- Вычислять агрегированные характеристики
  скорость переходов, среднюю длительность пребывания в каждом состоянии, частоту повторения последовательностей…

- Вычислять попарные различия между последовательностями с использованием разных методов формализации «расстояния»
- Оценивать различия и их факторы с использованием регрессионных методов
- …

TRAMINER: ДЕСКРИПТИВНЫЙ АНАЛИЗ

МАССИВ ДАННЫХ

- Работаем с встроенным в R массивом mvad.
- Данные ирландского обследования 2002 года. Респонденты – 16-летние школьники, за поведением в области образования и трудоустройства которых наблюдают с июля 1993 по июнь 1999 года.
- Все они достигли 16 лет в одно время, здесь календарная шкала (абсолютная) совпадает для них с возрастной (относительной).
- Загружаем данные:

```
data(mvad)
```

В окне с описанием рабочей среды появились сведения о нем.
В нем 712 наблюдений и 86 переменных по каждому из них.
72 – ежемесячный статус, еще 14 – дополнительные переменные.

```
View(mvad)
```

- Статусы:

EM Employment
FE Further education
HE Higher education
JL Joblessness
SC School
TR Training

ФОРМАТЫ ДАННЫХ ДЛЯ SA

- Длинный формат записи (STS, XX)

  1 0 0 0 1 0 1993 0 546.4 SC SC HE HE HE HE HE EM EM ….

- Короткий формат записи (SPS, XT)

  1 0 0 0 1 0 1993 0 546.4 2/SC 5/HE 2/EM ….

- Person-period Format

- …

- R умеет работать со всеми форматами. По умолчанию он ждет формат STS.

ШАГ 1: ЗАДАЕМ АЛФАВИТ

- Почти все функции пакета TraMineR в качестве одного из аргументов требуют т.н. **state sequence object**.
- Что это такое? Это непосредственно набор последовательностей + все его атрибуты.
- К обязательным атрибутам относятся алфавит и набор лейблов к нему. Факультативно могут быть заданы веса, цвета для графики и другие ее параметры, порядок обращения с пропущенными значениями… и т.д. В R используется т.н. *ленивая оценка аргументов функций*
- Этот объект задается функцией **seqdef** и полный перечень ее аргументов можно увидеть, вызвав функцию ?seqdef

- Сначала посмотрим, что вообще есть в файле, какой там используется алфавит? Для этого есть функция **seqstatl**

```
(mvad.alph <- seqstatl(mvad[, 17:86]))
```

*Здесь из рассмотрения исключены июль и август

ШАГ 2:
ЗАДАЕМ НАБОР ПОСЛЕДОВАТЕЛЬНОСТЕЙ

- Теперь зададим набор длинных и коротких лейблов и определим набор последовательностей:

```
mvad.lab <- c("employment", "further education", "higher education", "joblessness", "school", "training")
mvad.shortlab <- c("EM", "FE", "HE", "JL", "SC", "TR")
mvad.seq <- seqdef(mvad[, 17:86], alphabet = mvad.alph, labels = mvad.lab, states = mvad.shortlab, weights = mvad$weight, xtstep = 6)
```

ШАГ 3: НАБОР ПРОСТЫХ РИСУНКОВ

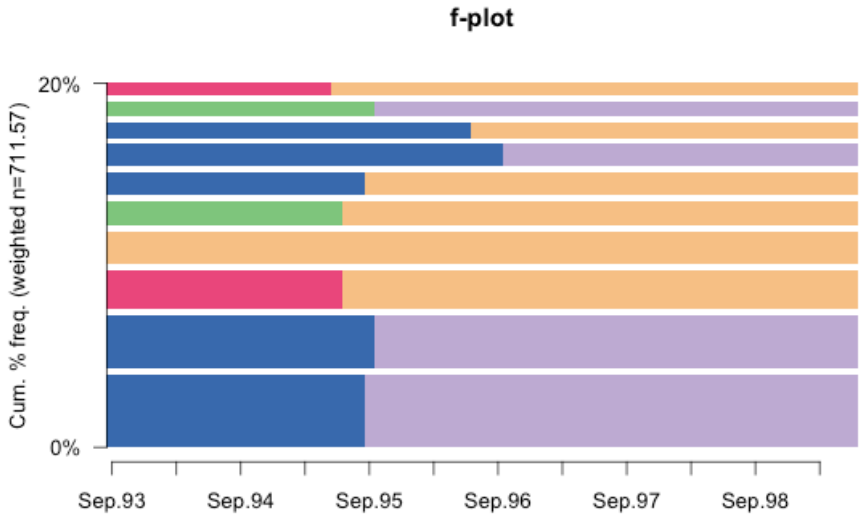
- Можно вывести на рисунок 10 первых последовательностей **seqiplot**

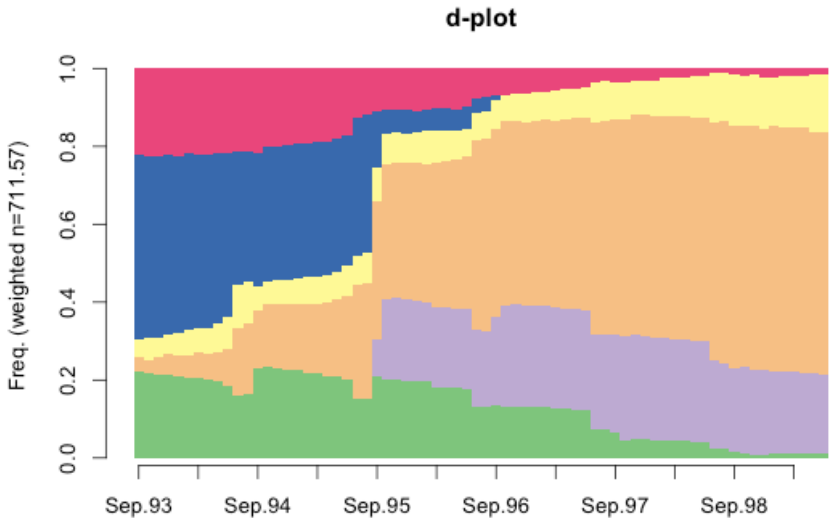
```
seqiplot(mvad.seq, withlegend = T, title = "Index plot (10 first sequences)", border = NA, legend.prop=0.21)
```

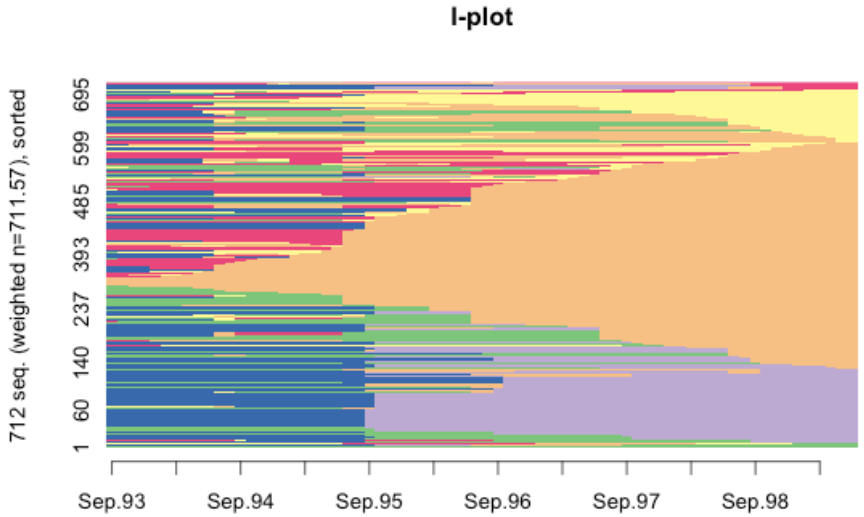
- Вывести 10 самых популярных последовательностей **seqfplot**
- Посмотреть, как индивиды распределяются по состояниям в каждый момент времени – что меняется с течением времени в структуре состояний? **Seqdplot**
- Посмотреть типы поведения («на глазок» оценить наличие кластеров)

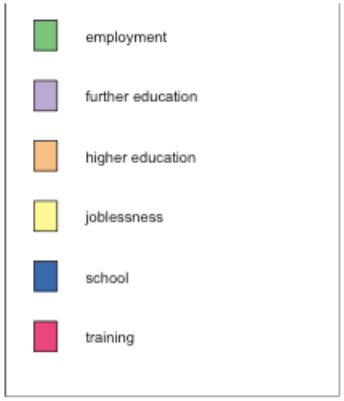
```
par(mfrow = c(2, 2))
seqfplot(mvad.seq, border = NA, withlegend = FALSE, title = "f-plot")
seqdplot(mvad.seq, border = NA, withlegend = FALSE, title = "d-plot")
seqIplot(mvad.seq, border = NA, withlegend = FALSE, title = "I-plot", sortv = "from.end")
seqlegend(mvad.seq, position = "bottomright", fontsize = 0.7)
```

ШАГ 3: НАБОР ПРОСТЫХ РИСУНКОВ







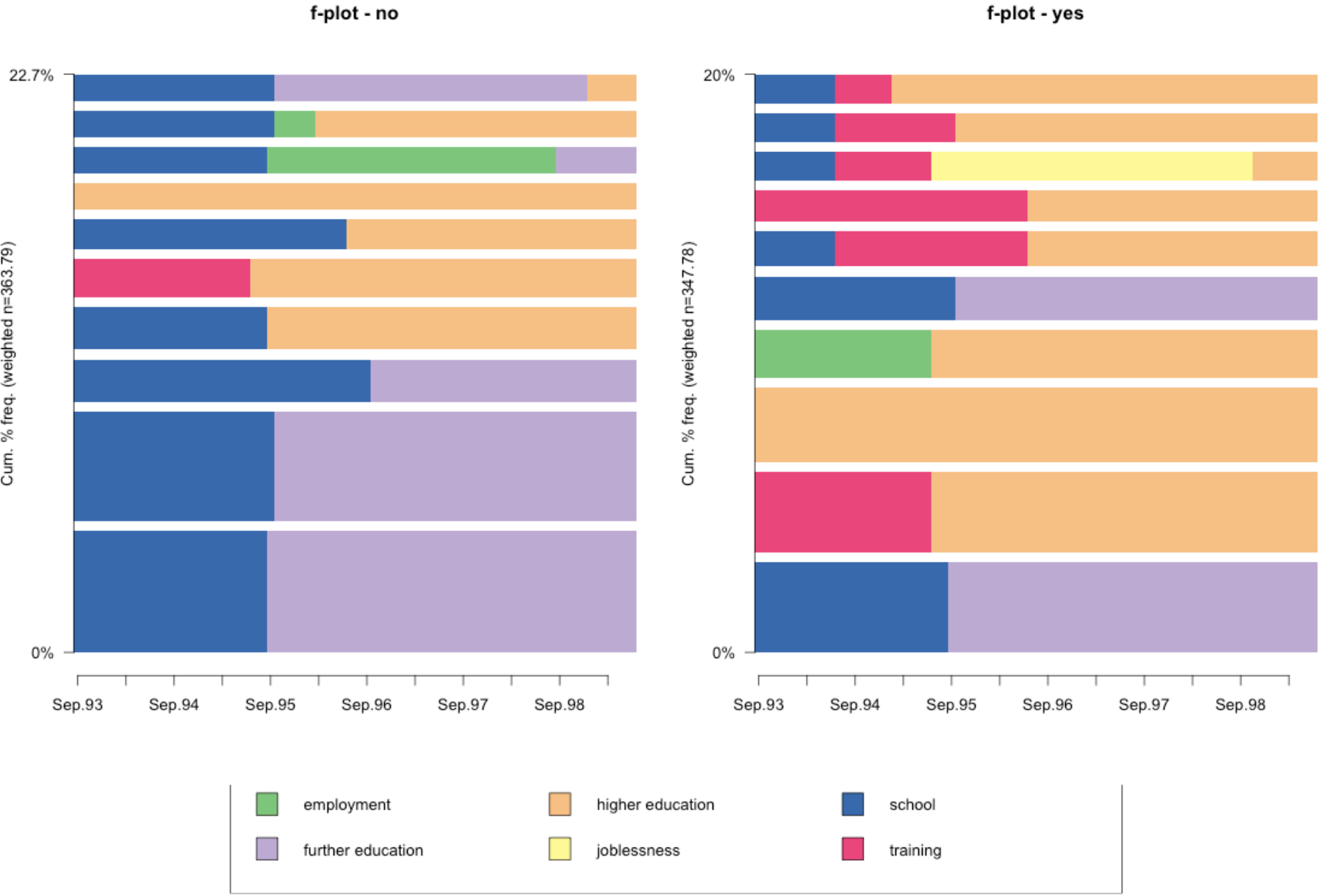


ШАГ 4: РИСУНКИ + ГРУППИРОВКА

- Всеми теми же способами можно визуализировать данные с разбивкой по группам
- Если вам повезет (групп будет немного и они будут действительно дифференцирующими), то различия будут видны невооруженным взглядом и их можно будет интерпретировать
- Например, можно сделать группировку по полу:

```
seqIplot(mvad.seq, group = mvad$male, sortv = "from.start", title = "Sex")

seqfplot(mvad.seq, group = mvad$male, border = NA, withlegend = FALSE, title = "f-plot")
```

ШАГ 4: РИСУНКИ + ГРУППИРОВКА



ШАГ 5:
МАТРИЦА ВЕРОЯТНОСТЕЙ ПЕРЕХОДОВ

- TraMineR позволяет вам рассчитать матрицу вероятностей переходов из одного состояния в другое на основе имеющихся данных при помощи функции **seqtrate**
- Размеры матрицы n*n, где n – число состояний
- Одним из аргументов функции может быть наличие зависимости от времени

```
round(trate <- seqtrate(mvad.seq), 3)
```

```
##          [-> EM] [-> FE] [-> HE] [-> JL] [-> SC] [-> TR]
## [EM ->]   0.986   0.002   0.003   0.007   0.000   0.002
## [FE ->]   0.027   0.950   0.007   0.011   0.001   0.003
## [HE ->]   0.010   0.000   0.988   0.001   0.000   0.001
## [JL ->]   0.037   0.012   0.002   0.938   0.001   0.010
## [SC ->]   0.012   0.008   0.019   0.007   0.950   0.004
## [TR ->]   0.037   0.004   0.000   0.015   0.001   0.944
```

- Матрица вероятностей переходов используется для определения «стоимости перехода»

ШАГ 6:
СРЕДНЕЕ ВРЕМЯ ПРЕБЫВАНИЯ В СОСТОЯНИИ

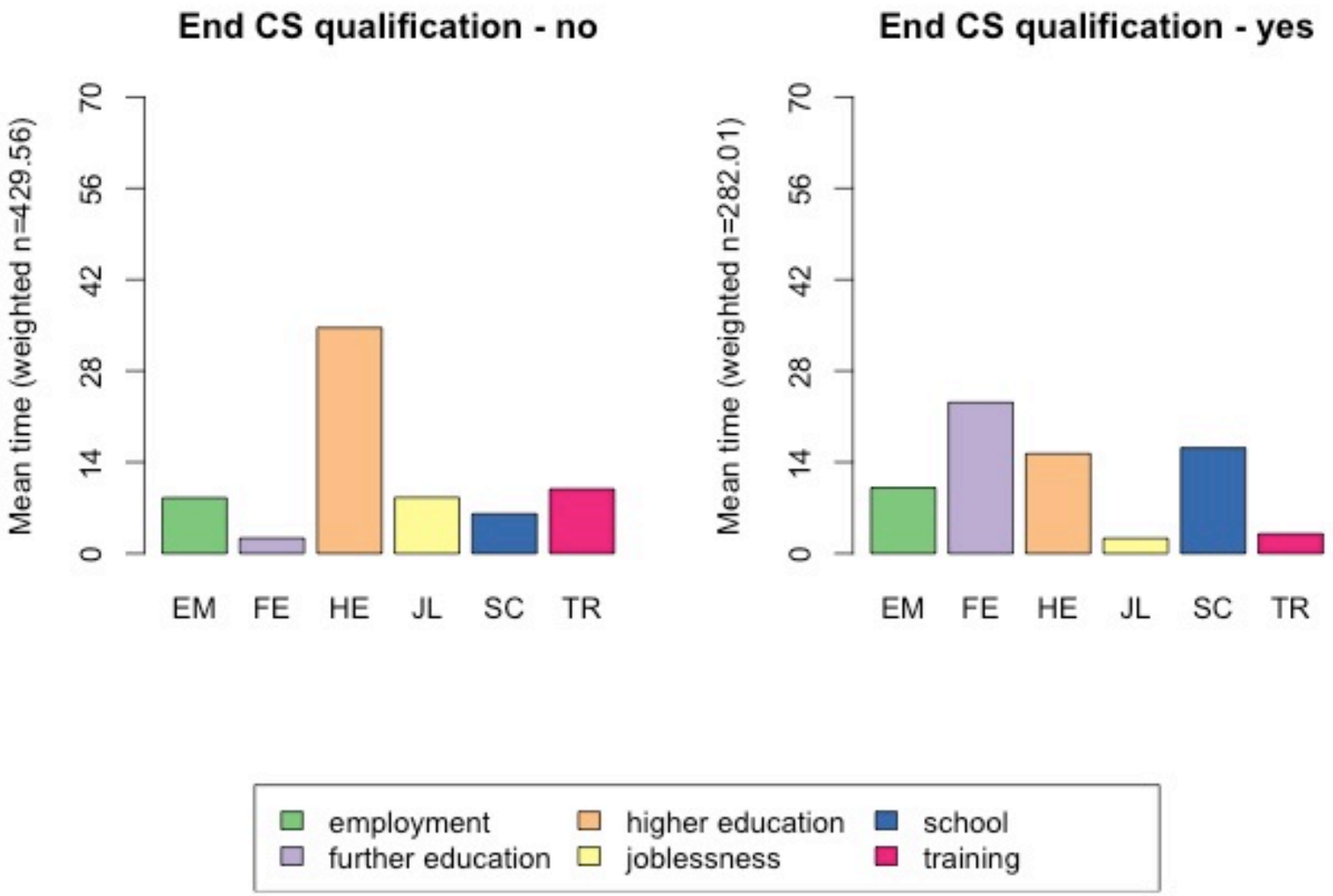
- Мы можем вычислить среднее время пребывания в каждом состоянии при помощи функции **seqmeant** и нарисовать диаграмму с ними функцией **seqmtplot**

```
seqmtplot(mvad.seq, title="Mean time in each state", legend.prop=0.2)
```

- В ней, как и в других функция, можно добавить группировку и посмотреть различия

```
seqmtplot(mvad.seq, group=mvad$gcse5eq, title="End CS qualification",
legend.prop=0.2)
```

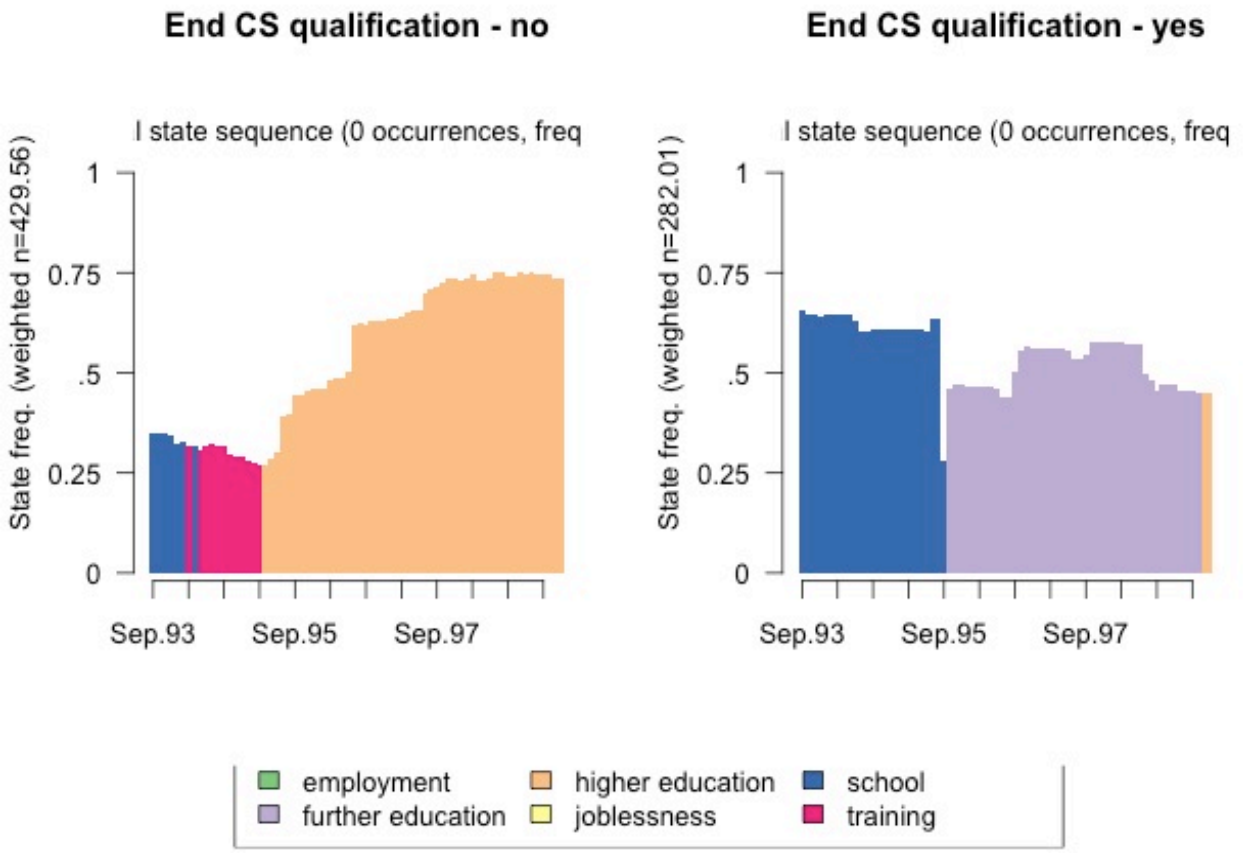
ШАГ 6:
СРЕДНЕЕ ВРЕМЯ ПРЕБЫВАНИЯ В СОСТОЯНИИ



ШАГ 7: ПОИСК МОДАЛЬНЫХ СОСТОЯНИЙ

- Для поиска модальных состояний в каждый момент времени есть функция seqmodst, а для их отображения — seqmsplot

```
seqmsplot(mvad.seq, group = mvad$gcse5eq, title = "End CS qualification",
border = NA)
```



ЭНТРОПИЯ

- Мера непредсказуемости, неоднородности, неопределенности

$$H(X) = -\sum_{x \in \mathcal{A}} p(x) \log p(x)$$

- Чем разнообразнее набор состояний, тем выше энтропия
- Чем равномернее распределение состояний, тем выше энтропия
- Энтропия максимальна тогда, когда все состояния (исходы) равновероятны
- Энтропия может быть индивидуальной (горизонтальной) или кроссекторальной (вертикальной)
- Для обеспечения сопоставимости переходят к относительной энтропии
- Индекс энтропии (относительная энтропия) принимает значения от 0 до 1

ШАГ 8: КРОССЕКТОРАЛЬНАЯ ЭНТРОПИЯ

- R вычисляет распределение состояний в каждый момент времени и считает индекс энтропии
- Вы можете вывести это в форме таблицы

```
seqstatd(mvad.seq[, 6:15])
```

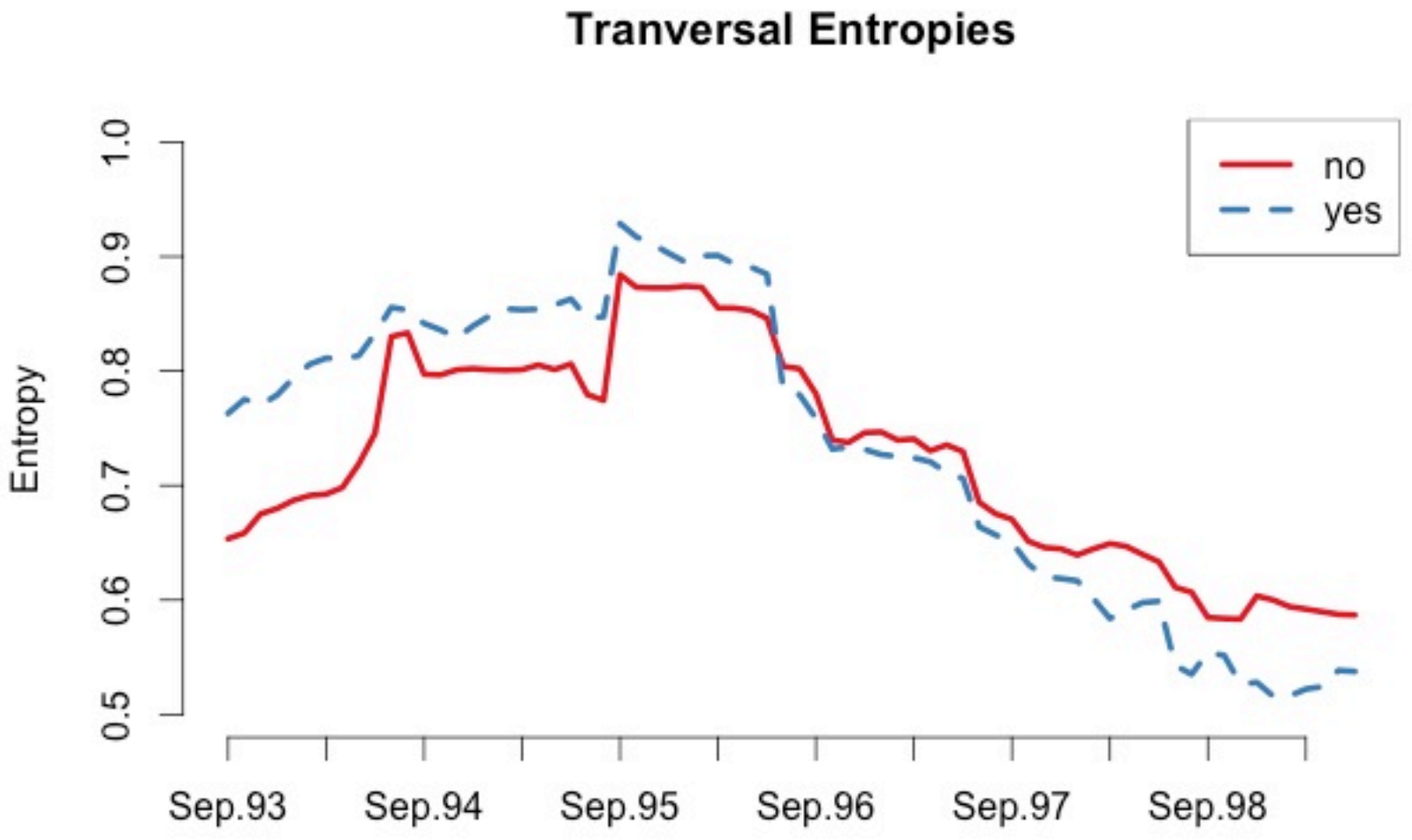
- Здесь тоже можно использовать группировки или фильтры и сравнивать энтропию в группах

```
seqstatd(mvad.seq[mvad$gcse5eq == "no", 6:15])
```

- Можно вывести динамику энтропии на диаграмму (требуется подключение пакета TraMineRextras)

```
seqplot.tentrop(mvad.seq, group=mvad$male)
```

ШАГ 8: КРОССЕКТОРАЛЬНАЯ ЭНТРОПИЯ



ШАГ 9:
ОЦЕНКА СЛОЖНОСТИ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

- Горизонтальная энтропия (longitudinal entropy)

  ABCABCABC = AAABBBCCC

- Турбулентность (Turbulence)
  - Учитывает число подпоследовательностей и длительность пребывания в состоянии

- Индекс сложности (Index of complexity)
  - Учитывает общее число совершенных переходов и горизонтальную энтропию

- Работают по-разному, у каждого свои недостатки, неодинаково характеризуют последовательности

```
mvad.ient <- seqient(mvad.seq)
mvad.cplx <- seqici(mvad.seq)
mvad.turb <- seqST(mvad.seq)
ctab <- data.frame(mvad.ient, mvad.cplx, mvad.turb)
plot(ctab)
```

TRAMINER:
ПОПАРНЫЕ РАЗЛИЧИЯ И РЕГРЕССИОННЫЙ АНАЛИЗ

СЛОЖНОСТЬ ПОСЛЕДОВАТЕЛЬНОСТЕЙ:
ПЕРВАЯ ВОЗМОЖНОСТЬ РЕГРЕССИОННОГО АНАЛИЗА

Complexity выражается числом, и мы можем посмотреть, как ее значение связано с характеристиками индивидов

```
lm.ici <- lm(mvad.cplx ~ male + funemp + gcse5eq, data = mvad)
summary(lm.ici)
```

```
Call:
lm(formula = mvad.cplx ~ male + funemp + gcse5eq, data = mvad)

Residuals:
      Min        1Q    Median        3Q       Max
-0.118956 -0.037583 -0.000176  0.032291  0.224911

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.108996   0.003891  28.009  < 2e-16 ***
maleyes     -0.013152   0.004328  -3.039  0.00246 **
funempyes    0.007165   0.005783   1.239  0.21572
gcse5eqyes   0.009961   0.004526   2.201  0.02806 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0567 on 708 degrees of freedom
Multiple R-squared:  0.02501,   Adjusted R-squared:  0.02088
F-statistic: 6.053 on 3 and 708 DF,  p-value: 0.0004519
```

ОЦЕНКА РАЗЛИЧИЙ МЕЖДУ ПОСЛЕДОВАТЕЛЬНОСТЯМИ

- Разница в индексе сложности плохо характеризует различия в последовательностях
- Основная головная боль — как формализовать различия в последовательностях? Как выразить разницу численно?
- Сейчас развиваются два принципиально разных подхода:
  - Оценка, подсчет общих частей
    - Longest common prefix
    - Longest common suffix
    - Longest common subsequence
    - Simple Hamming (подсчет числа позиций, на которых символы отличаются)
  - Оценка стоимости перехода (правок)
    - **Optimal matching**
    - Hamming
    - Dynamic Hamming
- Все, что связано с оценкой стоимости требует построения матрицы стоимости переходов (cost matrix)

МАТРИЦА ПЕРЕХОДОВ И ОЦЕНКА РАССТОЯНИЙ: ПОСТРОЕНИЕ

- Матрица переходов должна удовлетворять нескольким условиям:
  - Расстояние от объекта до самого себя равно нулю, до другого – больше нуля
  - Симметричность
  - Неравенство треугольника
- Фактически она задает пространство, отсюда и условия
- Ее можно задавать вручную, можно попросить вычислить R (опасно, нужно за ним проверять!)
- Чем больше статусов, тем сложнее определить матрицу переходов – не с точки зрения ваших временных затрат, а с точки зрения того, существует ли она вообще
- В нашей с вами области условия, накладываемые на матрицу переходов, очень часто будут нас не устраивать

МАТРИЦА ПЕРЕХОДОВ И ОЦЕНКА РАССТОЯНИЙ: ПОСТРОЕНИЕ

```
subm.custom <- matrix(
 c(0,1,1,2,1,1,
   1,0,1,2,1,2,
   1,1,0,3,1,2,
   2,2,3,0,3,1,
   1,1,1,3,0,2,
   1,2,2,1,2,0),
 nrow = 6, ncol = 6, byrow = TRUE,
 dimnames = list(mvad.shortlab, mvad.shortlab))
mvad.dist <- seqdist(mvad.seq, method="OM", indel=4, sm=subm.custom)
dim(mvad.dist)
```

- И тут сразу становится ясно, что при анализе последовательностей нам не только не обязательно набирать большие массивы, но и категорически не рекомендуется…

МАТРИЦА ПЕРЕХОДОВ И ОЦЕНКА РАССТОЯНИЙ: ЧТО У НАС ПОЛУЧИЛОСЬ?

```
print(mvad.seq[1:4, ], format = "SPS")
```

```
##     Sequence
## [1] (EM,4)-(TR,2)-(EM,64)
## [2] (FE,36)-(HE,34)
## [3] (TR,24)-(FE,34)-(EM,10)-(JL,2)
## [4] (TR,47)-(EM,14)-(JL,9)
```

```
mvad.dist[1:4, 1:6]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0   72   60   63   72   33
## [2,]   72    0   86  135   11  104
## [3,]   60   86    0   71   97   49
## [4,]   63  135   71    0  135   32
```

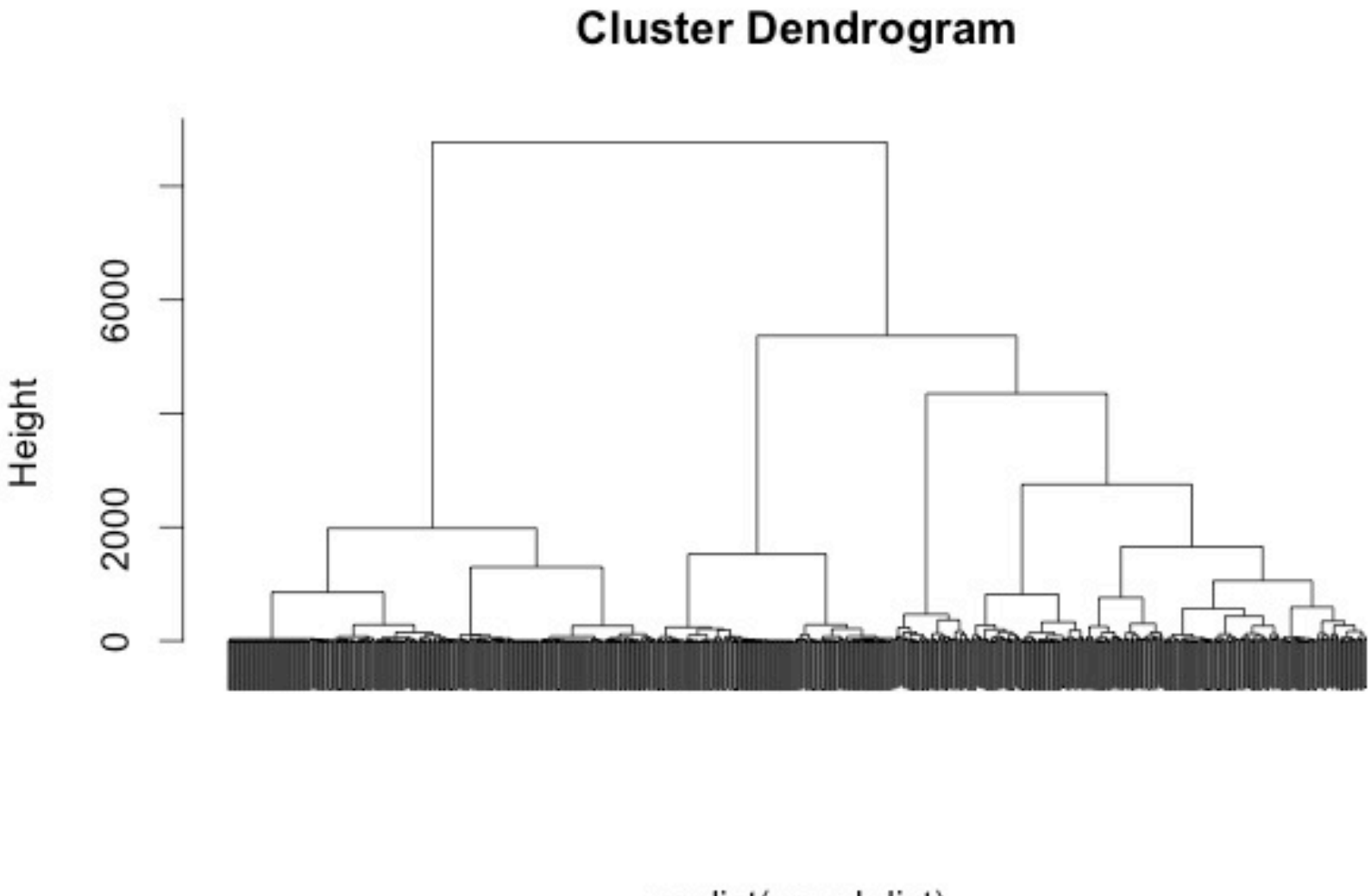
КЛАСТЕРИЗАЦИЯ

- После того, как мы получили матрицу расстояний, мы можем выделять кластеры любым из методов, к которому мы привыкли.
- В R доступны методы иерархической кластеризации и прямой кластеризации (PAM и другие)
- Сделаем иерархическую и сразу нарисуем ее:

```
mvad.clusterward <- hclust(as.dist(mvad.dist), method =
"ward", members = mvad$weight)
plot(mvad.clusterward, labels = FALSE)
```

- Прямая кластеризация делается быстрее, требует меньше ресурсов, но в ней вы задаете число кластеров самостоятельно

ИЕРАРХИЧЕСКАЯ КЛАСТЕРИЗАЦИЯ



КЛАСТЕРИЗАЦИЯ

- Плюсом является то, что потом по выделенным кластерам можно смотреть любые распределения
- Сделаем прямую кластеризацию с k=4 методом PAM:

```
set.seed(4)
pam.mvad <- wcKMedoids(mvad.dist, k = 4, weight = mvad $weight)
mvad.cl4 <- pam.mvad$clustering
xtabs(~mvad.cl4)
```

- И теперь мы можем посмотреть, например, простое распределение состояний по ним

```
seqdplot(mvad.seq, group = group.p(mvad.cl4), border = NA)
```

- А после этого решить, что за кластеры получились…

АНАЛИЗ РАЗЛИЧИЙ: ANOVA И ГРАФЫ

- У нас нет никакого среднего показателя, есть только попарные расстояния. Поэтому мы не можем классическим образом оценить дисперсию, чтобы затем оценивать различия (проверять значимость, выделять факторы…)
- Что делать?
- Перейти к оценке дисперсии через расстояния между последовательностями = к псевдо-вариации.
- Тогда дисперсия вычисляется так:

$$SS = \sum_{i=1}^{n}(y_i - \bar{y})^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(y_i - y_j)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i+1}^{n} d_{ij}$$

- А в R ее значение можно посмотреть вот так:

```
dissvar(mvad.dist)
```

ANOVA

- После того, как мы научились оценивать вариацию, мы можем проводить оценку групповых различий методом ANOVA

```
da <- dissassoc(mvad.dist, group = mvad$gcse5eq, R = 1000)
print(da)
```

ГРАФЫ (ДЕРЕВО ПЕРЕХОДОВ)

- Зачем их строить, что мы на самом деле делаем? Мы ищем наиболее значимые предикторы различий в последовательностях и их взаимодействия. Делается это при помощи регрессионного анализа.
- Мы постепенно разделяем все наблюдения на группы таким образом, чтобы они были максимально гомогенными (оцениваем различия в последовательностях).
- На каждом шагу мы выбираем тот предиктор, который позволяет разделить группы с максимальным R квадрат.
- Значимость оценивается на основе F-статистики.
- Процедура заканчивается, когда не остается возможности разделить на значимо отличающиеся группы.

ГРАФЫ (ДЕРЕВО ПЕРЕХОДОВ)

- Построение графа:

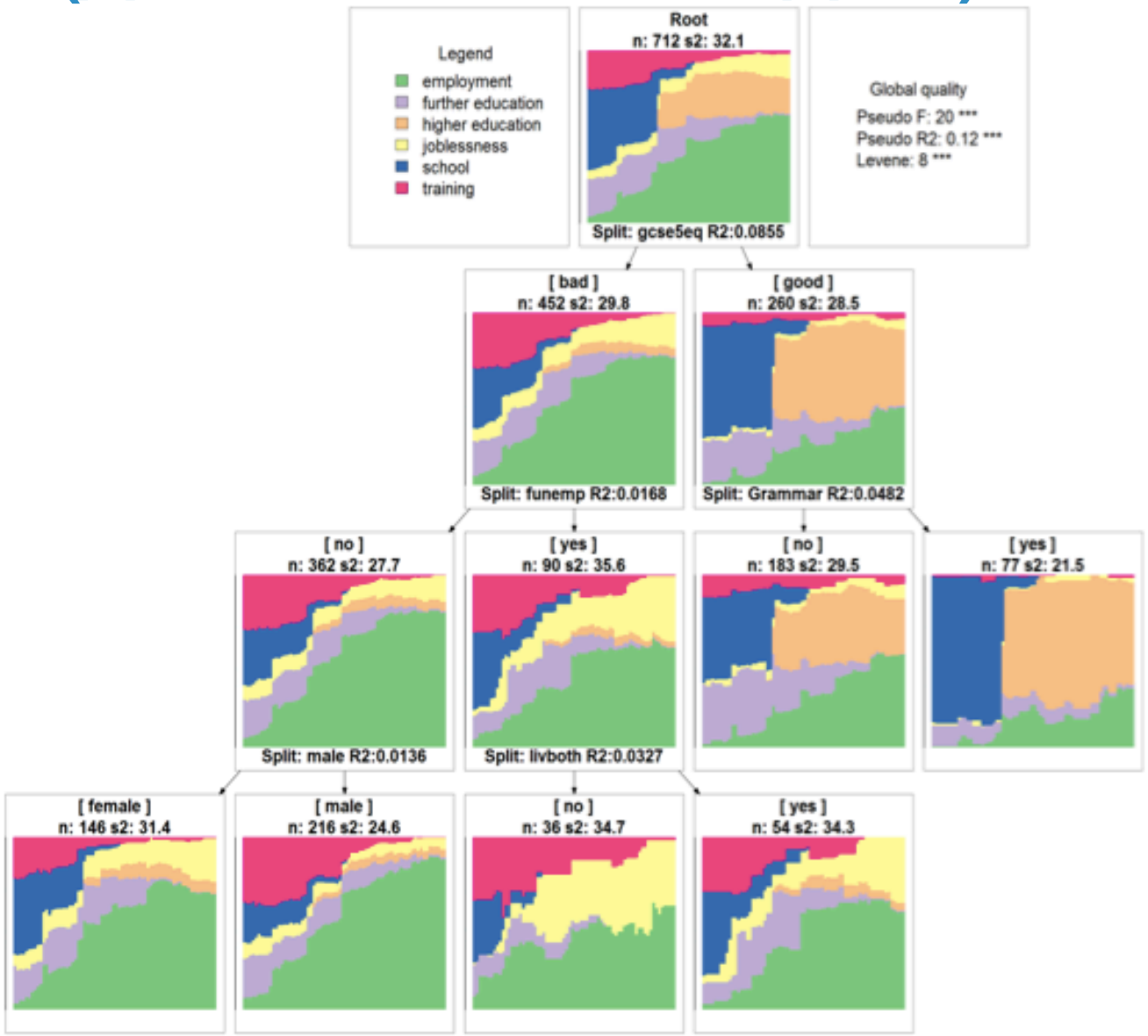
```
dt <- seqtree(mvad.seq ~ male + Grammar + funemp +
gcse5eq + fmpr + livboth, weighted = FALSE, data = mvad,
diss = mvad.dist, R = 5000)

print(dt, gap = 3)
```

- Визуализация:

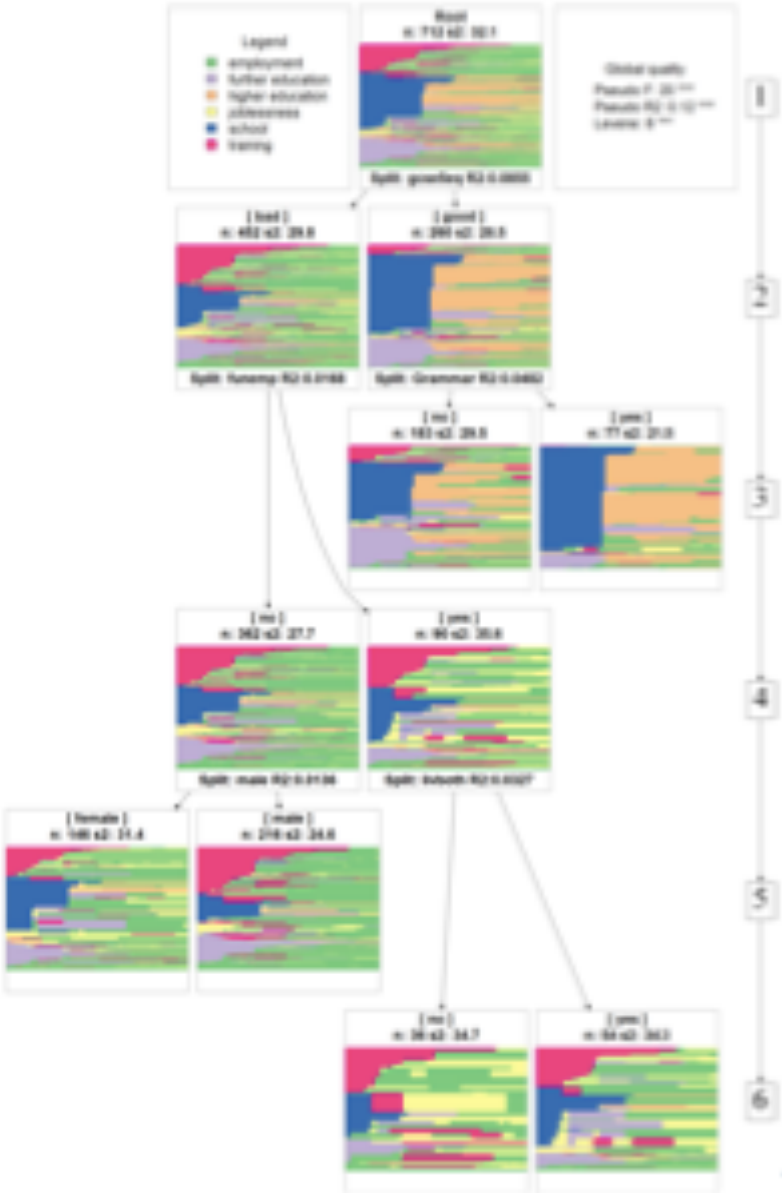
```
seqtreedisplay(dt, filename = "fg_mvadseqtree.png", type =
"d", border = NA)
```

Сейчас не заработает! Для работы этой команды на компьютер нужно установить Graphviz
http://www.graphviz.org/

ГРАФЫ (ДЕРЕВО ПЕРЕХОДОВ)



ГРАФЫ (ДЕРЕВО ПЕРЕХОДОВ)



R И TRAMINER:
ЧТО МОЖНО ПОЧИТАТЬ / ПОСМОТРЕТЬ?

ОБУЧАЮЩИЕ КУРСЫ И САМОУЧИТЕЛИ

TraMineR разработан специально для анализа последовательностей событий в демографии и социологии.

Но в R реализованы и другие возможности, в частности, все виды статистического и регрессионного анализа. А кроме того, есть возможность легко интегрировать в работу простейшие логические блоки и писать небольшие программы.

В последнее время появляется все больше обучающих курсов по R, в том числе бесплатных. Некоторые из них:

- Специализация Data Science на coursera.org (R Programming)
- Explore Statistics with R на edx.org
- Набор простых упражнений с подробными инструкциями (бесплатные модули) на datacamp.com/

По пакету TraMineR есть очень подробная инструкция с упражнениями

- User's guide of TraMineR http://mephisto.unige.ch/pub/TraMineR/doc/TraMineR-Users-Guide.pdf

R И TRAMINER:
ВМЕСТО АВТОРСКИХ ПРАВ

ЦИТИРОВАНИЕ

- Упоминайте в тексте Вашей статьи, препринта, презентации, на основе какого программного обеспечения Вы работали

Для общего цитирования R:

R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Для общего цитирования TraMineR:

Gabadinho, A., Ritschard, G., Muller, N. S., Studer, M. (2011). Analyzing and Visualizing State Sequences in R with TraMineR. Journal of Statistical Software, 40(4), 1-37. URL http://www.jstatsoft.org/v40/i04/.

- Актуальную информацию и сведения о публикациях по отдельным методам анализа последовательностей всегда можно получить, набрав в консоли R

```
citation()
citation("TraMineR")
```

BONUS: ИМПОРТИРОВАНИЕ ДАННЫХ

- Для импорта файлов из других статистических программ необходимо установить пакет **foreign**
- После этого вы можете воспользоваться командой **read.spss**

```
ArrayName <- read.spss("Path to your file..../File name.sav", to.data.frame = TRUE)
```

- HELP по ней вызывается командой

```
?read.spss
```

- ***Импорт файлов из Stata осуществляется аналогичным образом с использованием команды **read.dta()**