

Глава 3

Краткое описание пакета Stata

Программа Stata StataCorp. (1999, 2001)— это универсальный пакет для решения статистических задач в самых разных прикладных областях: экономике, медицине, биологии, социологии. Впервые пакет вышел на рынок под этим названием в начале 80-х гг. В январе 1999 г. была выпущена шестая версия, в декабре 2000 г. — седьмая. Основными достоинствами Stata являются:

- большой спектр реализованных статистических методов (хотя и есть методы, не реализованные практически никак, например, дискриминантный анализ, кластерный анализ, обобщенный метод моментов, ряд других);
- возможности гибкой пакетной обработки данных (т. е. программирования всей последовательности команд, начиная от загрузки данных в память и вплоть до всех деталей анализа). Возможности интерактивного режима работы полностью идентичны возможностям пакетной обработки;
- относительная простота написания собственных программных модулей, и, вместе с тем, весьма серьезный спектр средств программирования;
- мощная поддержка как со стороны производителя, так и со стороны других пользователей Stata (через интернетовский список рассылки); огромный архив пользовательских программ в открытом доступе;
- возможность максимизации функций правдоподобия, задаваемых пользователем;

- наличие совместимых по функциональным возможностям и форматам данных реализаций для большинства популярных платформ (Windows, Macintosh, UNIX).

По поводу графических средств мнения пользователей разнятся: с одной стороны, они вполне достаточны для текущего графического анализа данных и подготовки научных публикаций (все рисунки в этой книге выполнены в Stata и импортированы в L^AT_EX), с другой, несравнимы с графическими возможностями специализированных пакетов типа Harvard Graphics или презентационных программ типа PowerPoint.

Ниже будет приведена сводка наиболее важных команд пакета. Эта сводка вряд ли сможет заменить изучение этих (и, естественно, других) команд по руководствам пользователя или хотя бы по встроенной подсказке Stata (например, не все детали синтаксиса и не все опции могут быть упомянуты в данном кратком введении). Скорее, она поможет найти, какими командами и для чего следует воспользоваться; более полное и точное описание этих команд можно найти во встроенной помощи Stata и в руководствах. Многие команды будут упомянуты лишь на уровне названия (что, впрочем, достаточно для поиска по встроенной подсказке Stata). *Читателю настоятельно советуется овладеть и пользоваться встроенной помощью Stata по командам и деталям внутреннего устройства пакета.*

3.1 Договориться: обозначения

Мы будем пользоваться следующими обозначениями, выдержанными в стиле руководств Stata. Так, `command` — команда, которую можно набирать целиком, а можно сократить до первых трех букв (например, `regress` можно написать как `reg`, а можно как `regress`). [в квадратных скобках] будут указаны необязательные фрагменты команды — необязательные опции, списки переменных и т. п. *Курсивом* мы будем обозначать то, что пользователь подставляет по своему разумению — названия переменных, численные значения параметров программ и т. п. Через вертикальную черту будут перечисляться возможные варианты: [вариант 1|вариант 2]. Таким образом, запись `describe [переменные | using имя файла]` может разворачиваться в следующие варианты:

```
d
describe
```

```
describe x1 x2 x3
d using source
desc using source.dta
```

Эта команда выдает краткое описание файла данных в памяти Stata или на диске.

Ссылки на руководства также оформляются в стиле Stata: [R] **команда** означает, что эту команду можно найти в четырехтомном справочнике команд (Reference); [U] **3 A brief description of Stata** — это ссылка на Руководство пользователя, а именно на главу 3 в книге User's Guide (для Stata 6) — описание Stata в руководстве пользователя (то, что можно почитать о Stata вместо этого параграфа); [G] **twoway** — описание двумерных графиков в руководстве по графике.

3.2 Открыть: установка и запуск Stata

Обычно Stata устанавливается в каталог `c:/stata`, если при установке не было явно указано иное. Исполняемый файл называется `wstata.exe` (Stata for Windows).

Команда `verinst` проверяет корректность установки пакета.

Сам этот исполняемый модуль выполняет сравнительно небольшое число (около 200) базовых процедур. Подавляющее большинство собственно статистических задач выполняется внешними программами с расширением `.ado`, находящимися в каталоге `c:/stata/ado` и его подкаталогах. Эти `ado`-файлы с некоторой степенью условности можно разделить на *базовые* (около 900), отлаженные разработчиком и входящие в комплект поставки Stata, (хотя и в них иногда находят ошибки, и тогда Stata делает официальные обновления `ado`-файлов); *официально распространяемые*, входящие в состав официальных дополнений к Stata — Stata Technical Bulletin, сокращенно STB, которые рассылаются подписчикам и распространяются бесплатно через Internet; и, наконец, *пользовательские*.

При запуске Stata устанавливает ряд внутренних параметров, таких, как объем выделяемой памяти, и некоторые другие (о них можно узнать в [R] **limits** или в подсказке `help limits`). Практически наверняка вам придется менять следующие установки:

```
set memory объем памяти[k|m]
```

Объем памяти, выделяемой операционной системой для Stata. Чтобы отвести 10 мегабайт, надо напечатать: `set memory 10m`. Можно выделить память при запуске па-

раметром командной строки: `wstata /k 10240`. Если количество переменных ограничено 2047, то количество наблюдений может быть ограничено только возможностью выделения памяти операционной системой. При выделении количества памяти, приближающейся к физическому объему ОЗУ (или тем более превышающего этот объем), Stata начинает пользоваться виртуальной памятью (постоянно перезагружаемой с жесткого диска), и работа может замедляться в сотни раз.

`set matsize число`

Максимальный размер матрицы, которую Stata сможет обработать. По умолчанию устанавливается 10. Максимальный размер — 800. Этот параметр влияет на размерность статистических моделей, которые Stata будет в состоянии оценить.

Stata может быть запущена в *пакетном* режиме, в котором она обрабатывает заданную в качестве входного параметра программу¹, а по завершении выполнения этой программы — передает управление операционной системе (или, попросту говоря, самоликвидируется). Такой вариант запуска задается (в Windows) как `wstata /b do имя файла с программой`.

Выход из Stata осуществляется командой `exit`. Если при этом данные не были сохранены, Stata об этом напомнит.

См. также: [U] 5 Starting and stopping Stata, [U] 6 Troubleshooting starting and stopping Stata

3.3 Придти, увидеть, посчитать: интерфейс Stata

Внешний вид Stata (рис. 3.3) несколько отличен от того, что обычно можно увидеть в других статистических пакетах. Внешний аскетизм интерфейса унаследован от идеологии командной строки UNIX, и пользователю Windows требуется некоторое привыкание.

Stata использует в работе несколько окон: окно ввода команд (Stata Command), окно вывода результатов (Stata Results), окно истории, или предыдущих команд (Review), окно переменных (Variables), окно поиска и помощи (Help), графический экран (Graph), окно файла-протокола, или log-файла (Log; в 7-й версии его функцию выполняет окно Viewer). Можно также вызвать окна просмотра данных (Stata Browser) или

¹О программах см. ниже параграф 3.13.

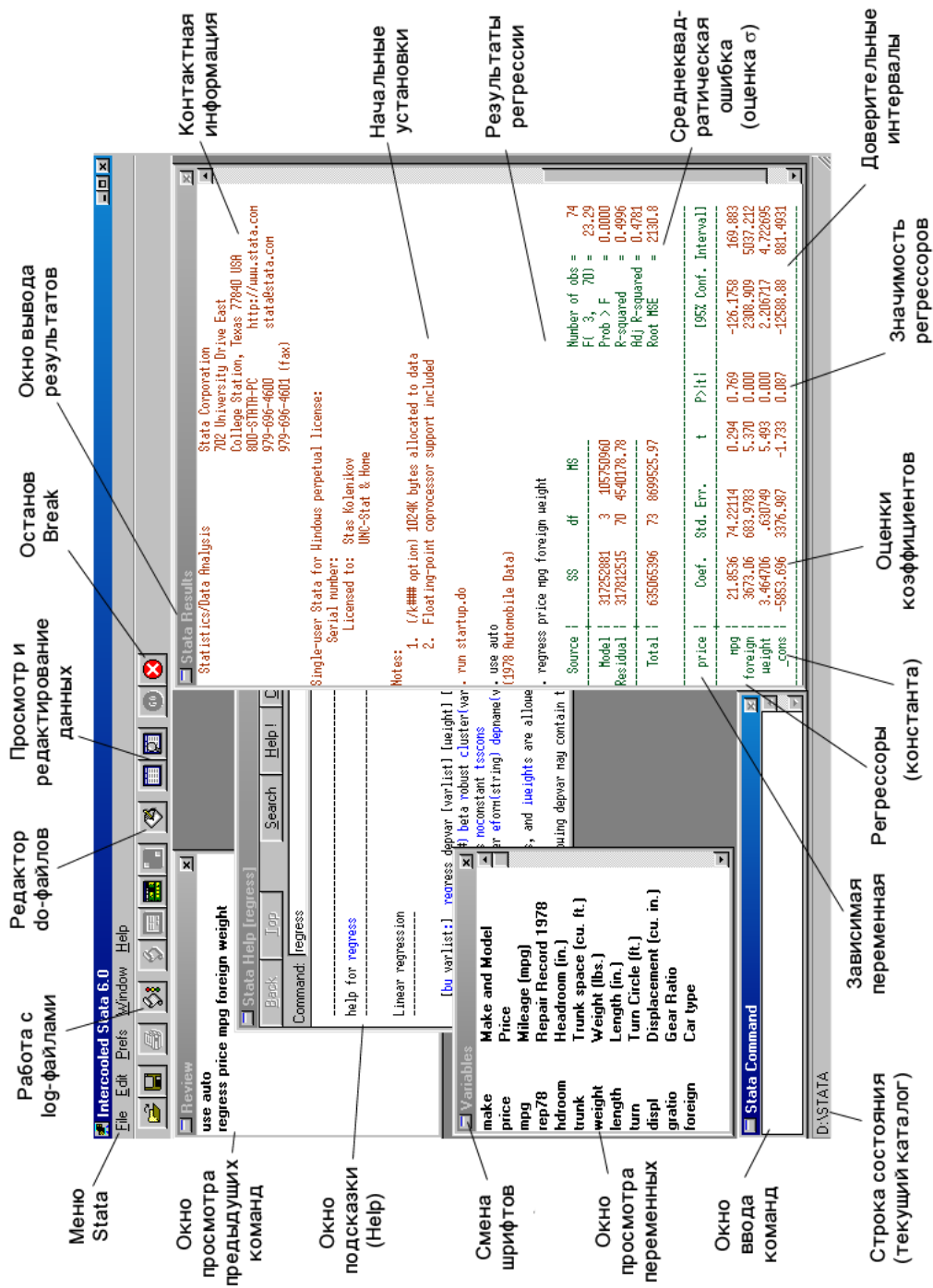


Рис. 3.1: Интерфейс Stata.

редактирования данных (Stata Editor), а также редактор программ (Stata Do-file Editor). Переключаться между окнами можно, тыкаясь мышкой в любое место на нужном окне, либо через меню **W**indows.

При вводе команд в окне Stata Command можно пользоваться стандартными средствами редактирования в Windows (выделения, стирания, вставки в буфер и из буфера). Можно вызывать предыдущие команды, нажимая PgUp и PgDn, и редактировать их (что очень полезно, если при вводе команды были допущены мелкие опечатки, или если надо добавить какие-то опции или условия к предыдущей команде). Можно менять кое-какие установки Stata в меню **P**refs, например, сохранить текущие установки окон (размеры, положение, шрифты).

См. также: [GSW] , т.е. руководство Getting Started for Windows.

3.4 Обобщить: как выглядят команды Stata

Команды Stata, как правило, имеют следующий вид:

команда [*список переменных*] [*if условие*] [*in диапазон*] [*using имя файла*] [[*веса*]], [*опции*]

Список переменных может состоять из одной переменной (например, если нужно получить сводные статистики или построить гистограмму), из двух (расчет корреляций или построение диаграммы рассеяния) и более (регрессии, графики со многими переменными). Условия *if* и *in* выделяют те наблюдения, для которых необходимо провести анализ (см. ниже параграф 3.6). Если команда предполагает работу с файлами (чтение, объединение и т.п.), то имя файла, с которым необходимо провести указанные действия, передается в конструкции **using**. Если разным наблюдениям необходимо придать разные веса, то для этого используется конструкция типа [**weight=выражение**] (см. **help weights**; квадратные скобки обязательны). Наконец, дополнительные модификаторы и параметры, влияющие на выполнение команд Stata или вывод результатов, а также все, что не поместилось в упомянутые рамки синтаксиса, записываются в опции.

Есть несколько исключений из вышеупомянутого синтаксиса, в т.ч. команды, выполняющие повторные действия — см. ниже параграф 3.11.

См. также: [U] **14 Language syntax**

3.5 Узнать: помощь

В Windows-версии Stata для поиска нужной информации проще всего воспользоваться меню **Help**, в котором имеются подменю **Search** (поиск по ключевым словам, например, *Durbin Watson statistic*) и **Stata Command** (файл помощи по конкретной команде Stata). Впрочем, практически все то же самое можно сделать с клавиатуры командами **search**, **help** и **whelp**. Содержимое встроенной подсказки полностью дублируется в открытом доступе на сайте Stata: <http://www.stata.com/info/capabilities/>.

Встроенная помощь Stata устроена гипертекстовым образом: если подвести мышку к фрагменту текста, выделенному зеленым цветом, то курсор превратится в ладошку, а если нажать при этом на левую кнопку мыши², то будет выведен соответствующий фрагмент подсказки Stata. Если зеленым цветом помечена ссылка в Internet, то Stata запустит внешний браузер (MS Internet Explorer, Netscape Navigator). В Stata 7 эти действия можно выполнять и с результатами, выводимыми в окно Results.

Полный список стандартных команд, входящих в состав начальной установки Stata, можно найти в меню **Help/Contents** (или по команде **help contents**). Эти команды сгруппированы по тематическим разделам: общее представление о пакете, синтаксис команд, работа с данными, графика, статистические средства, матричные команды, программирование, особенности работы в среде Windows.

Все файлы помощи представляют собой специальным образом отформатированные текстовые файлы с расширением **.hlp**³.

В Stata имеются собственные обучающие средства — мини-уроки (являющиеся, с технической точки зрения, специальным видом программ), доступ к которым обеспечивается командой **tutorial**. Они дают краткое введение в пакет, в графические и табличные средства Stata, знакомят с данными, поставляемыми вместе со Stata, и способами перевода текстовых файлов в формат Stata, а также освещают ряд основных статистических команд.

См. также: [U] **8 Stata's on-line help and search facilities**, [U] **9 Stata's on-line tutorials and sample datasets**.

²Для левшей эта кнопка, возможно, будет правой — имеется в виду та кнопка, на которой лежит указательный палец.

³ В ОС Windows также имеется формат гипертекстовой помощи, несовместимый с форматом Stata, поэтому кликание на файлах помощи Stata из Проводника (Explorer) Windows ни к чему не приведет.

3.6 Ограничить: условные модификаторы

Многие команды Stata позволяют ограничить свое действие на определенные наблюдения. Делается это с помощью *условных модификаторов* [`if условие`] [`in диапазон`]. Условие, задаемое под `if` — это логическое выражение, в котором могут использоваться операторы отношений `>` ("больше"), `<` ("меньше"), `>=` ("больше или равно"), `<=` ("меньше или равно"), `==` ("равно", двойной знак использован для того, чтобы не спутать с операцией присвоения), `!=` или `~ =` ("не равно"); логические операции `&` ("и"), `|` ("или"), `!` или `~` ("не"), указание на текущее наблюдение `_n` и на последнее `_N`, обычные операции и функции, а также скобки для указания приоритета. `in` указывает диапазон наблюдений вида *начало/конец*, где в качестве конца диапазона может быть использовано последнее наблюдение, обозначаемое латинской "эл" (1) или как `-1`.

Пропущенное значение (см. ниже описание команды `generate`, с. 79) тоже может фигурировать как аргумент логической команды. Так, `count if x!=.` выдаст количество наблюдений, для которых известно значение переменной `x`.

3.7 Загрузить, сохранить, объединить: работа с файлами

Естественно, для того, чтобы данные анализировать, их надо как минимум загрузить в память. Stata обладает достаточно гибкими средствами ввода данных из текстовых файлов (команды `infile`; `infix`; `insheet`; см. также `help dictionary` и [U] 24 **Commands to input data**), однако файлы других форматов (Excel, SAS, SPSS, Statistica и т.п.) необходимо предварительно сохранить в виде текста (с разделением данных запятыми, табуляциями, или в фиксированном формате), либо воспользоваться внешними средствами для конвертации данных. В комплект поставки Professional Stata входит чрезвычайно полезная Windows-утилита StatTransfer (<http://www.stattransfer.com>), позволяющая преобразовывать данные между двумя десятками различных форматов. Другая похожая по функциональным возможностям программа — DBMS/COPY.

Работу с уже имеющимися файлами данных формата Stata можно осуществлять из меню File, а можно и с клавиатуры.

```
use имя файла, [clear]
```


Загрузить в память указанный файл. Опция `use ... , clear` показывает, что при этом нужно уничтожить все данные, находящиеся в памяти. Если размер оперативной памяти (точнее, размер свободной памяти, остающейся после Windows и прочих приложений) не позволяет втиснуть в нее необходимый файл, то можно воспользоваться вариантом `use переменные using имя файла [if условие] [in диапазон]`, выбрав модификаторами только те переменные и/или только те наблюдения, которые нужны для работы. С помощью этого трюка можно проводить “черновой” анализ для задач большого объема, т.е. отработать последовательность команд на некоторой подвыборке, сохранить алгоритм работы с данными в виде do-файла (см. параграф 3.13), а потом оставить на выходные машину считать все то же самое по полной выборке.

`save имя файла , [replace old]`

Сохранить данные из памяти на диск под указанным именем. Опция `replace` указывает, что файл надо переписать, если он существует. Если нет — не беда, он будет создан. Опция `old` нужна для сохранения данных из-под Stata 6 в формате Stata 4-5 (т.е. для обмена данными с обладателями Stata более ранних версий). В Stata 7 опция `old` позволяет записать данные в формате Stata 6.

`merge список ключевых переменных using имя файла , [nokeep]`

Добавить данные из указанного файла к данным, находящимся в памяти. Необходимо для пополнения данных “вширь”, т.е. для добавления *переменных*. Необходимо, чтобы в обоих файлах (которые на жаргоне Stata называются master data и using data) имелись *ключевые переменные*, т.е. переменные, однозначно идентифицирующие наблюдения, а также чтобы файлы были отсортированы по этим переменным, см. [R] `sort` и ниже команду `sort`. Некоторые из этих ограничений преодолеваются командой `mmerge` Wessie (1999), которую необходимо устанавливать дополнительно (см. раздел 3.17). Опция `nokeep` указывает, что не надо добавлять наблюдения, которые встречаются только в using data.

`append using имя файла`

Добавить данные из указанного файла *в длину*, т.е. добавить новые наблюдения.

См. также: [U] **25 Commands for combining data**

3.8 Добавить, выбросить, переименовать: работа с данными

В Stata имеется несколько типов данных. Первый уровень разделения — это данные строковые и числовые. Числовые делятся в свою очередь на целые и действительные, а внутри каждого класса есть различия в точности представления; см. [U] **data types**, **help datatypes**.

generate [*тип*] *имя переменной* = *выражение* [**if** *условие*] [**in** *диапазон*]

Создать новую переменную, возможно, указанного типа, и присвоить ей значение выражения. Имя переменной в шестой версии Stata может быть длиной до восьми символов, а в седьмой — 32, включать в себя буквы (верхний и нижний регистр различаются), цифры или знак подчеркивания, и должно начинаться с буквы. В *выражение* могут входить числа, переменные, фигурировать арифметические операции, функции (математические, статистические, строковые и пр.), логические условия (которые вычисляются как 1 — истина и 0 — ложь), а также обозначение для пропущенного значения (missing value) — точка (.). Stata устроена так, что пропущенное значение равняется плюс бесконечности. Все операции с пропущенным значением будут давать пропущенное значение (кроме логических операций сравнения). Команда **generate nonmissx=x** создаст новую переменную *nonmiss* типа **byte** (т.е. наименьшего возможного размера), которая будет равна 1, если переменная *x* имеет конечное значение, и 0, если *x* не определена. Об условиях и диапазонах говорится ниже, в разделе 3.6. См. также [U] **14 Language syntax**, [U] **15 Data**, [16] **Functions and expressions**.

egen [*тип*] *имя переменной* = *egen-функция(выражение)* [**if** *условие*] [**in** *диапазон*], [**by**(*идентификатор группы*)]

Более мощная функция для создания новых переменных, позволяющая рассчитывать средние, медианы, минимумы, максимумы, суммы значений и т. п. по всей выборке или по группам, задаваемым переменными-идентификторами. Подробный список поддерживаемых функций и статистик имеется в [R] **egen** или **help egen**.

xi *специальные выражения*

xi: *команда Stata со специальными выражениями*

Позволяет создать набор бинарных (0/1) переменных из категориальной, или выполнить указанную команду, включив в список переменных создаваемый на ходу набор бинарных переменных. Одна из категорий берется как базовая, и для нее бинарная пе-

ременная не создается, т.е. корректно отрабатывается проблема статистической связи между получаемыми бинарными переменными. Специальные выражения имеют вид *i.категорийная переменная*.

recode

Изменяет значения переменной. Актуально для перекодировки значений категорийной переменной или для соединения нескольких категорий в одну.

replace *имя переменной=выражение* [**if** *условие*] [**in** *диапазон*]

Заменить значения уже существующей переменной.

rename *имя переменной новое имя*

Переименовать переменную.

drop if *условие* | **in** *диапазон*

Удалить наблюдения, удовлетворяющие указанным условиям.

drop *переменные*

Удалить указанные переменные.

list [*переменные*] [**if** *условие*] [**in** *диапазон*]

Вывести значения указанных переменных (если не указано ничего, то всех) для наблюдений, удовлетворяющих указанным условиям (если никаких условий не указано, то вывести все наблюдения).

edit [*переменные*] [**if** *условие*] [**in** *диапазон*]

Вручную редактировать указанные переменные для указанных наблюдений. Stata предоставляет для этой цели что-то вроде примитивных электронных таблиц. Использовать подобный режим для внесения изменений в данные не рекомендуется в целях обеспечения воспроизводимости результатов.

browse [*переменные*] [**if** *условие*] [**in** *диапазон*]

Просмотреть значения указанных переменных для указанных наблюдений. То же, что и **edit**, только изменять ничего нельзя.

aorder

Отсортировать переменные по алфавиту.

sort *переменные*

gsort +|-*переменная* ...

Отсортировать данные по указанным переменным.

compress [*переменные*]

Привести переменные (если не указано, какие, то все) к минимально возможному типу, снижая тем самым объем памяти, необходимый для их хранения.

reshape

Достаточно продвинутая команда, необходимая для изменения представления группированных данных — например, панельных. Она переводит данные из “длинного” (long) формата, в котором на каждый объект панели имеется несколько наблюдений, соответствующих разным моментам времени (много наблюдений, откладываемых “в длину”— мало переменных, откладываемых “в ширину”), в “широкий” формат (мало наблюдений — много переменных), в котором наблюдения соответствуют объектам, а данные записаны в виде переменных, названия которых заканчиваются на “дату”. Так, файл с переменными `income96`, `income97`, `income98` — это данные в широком формате, а файл с переменными `income`, `year`, где `year` принимает значения 96, 97, 98 — это данные в длинном формате. Панельные команды Stata, имеющие префикс `xt`, работают с данными в “длинном” формате.

describe [*переменные*] [*using имя файла*], [**short**]

Вывести описание данных и переменных: формат, метки и т. п. Эта команда показывает также количество наблюдений и переменных, изменялись ли данные с момента последнего сохранения, по каким переменным отсортированы наблюдения. Можно указать файл, находящийся на жестком диске.

label

Приписать метки к данным или переменным. `label variable имя переменной "текст"` создает метку переменной, которая выводится командой `describe` и видна в окне переменных. Можно также задать метку для файла данных (информация о файле данных хранится в сопровождающем его объекте `_dta`, и соответствующая команда будет иметь вид `label data`). Эта метка будет выводиться при исполнении `use` и `describe`. Можно также задать метки для отдельных значений дискретной переменной через `label define` и `label values`. Признаком хорошего стиля работы с данными является придание меток создаваемым переменным: после любой команды `generate` или `egen` должно идти `label variable`.

notes [`_dta` | *переменная*] : " *текст*"

Еще один вариант создания примечаний о файле или переменных. Если командой `label` всем данным в целом или отдельной переменной можно приписать только одну метку ограниченной длины, то `notes` позволяет приписать к каждой переменной или к `_dta` произвольное число меток произвольной длины. Примечания удобны для внесения комментариев типа: “Разобраться с этой переменной”; “Данные за 1994 г. сверены”; “Файл получен программой `households.do`” и т.п.

`lookfor` *текст*

Ищет указанный текст в названиях и метках переменных.

`clear`

Очистить память, выгрузив все данные, метки, программы, макросы.

3.9 Оценить: основные статистические средства

`summarize` *переменные* [`if` *условие*] [`in` *диапазон*], [`detail`]

Сводка описательных статистик, таких, как количество наблюдений, среднее, стандартное отклонение, максимум, минимум. Опция `detail` позволяет вывести также характерные квантили, несколько самых больших и самых маленьких значений и коэффициенты асимметрии и эксцесса. Прочие команды, описывающие данные в компактном виде — `lv`; `codebook` и `inspect`. Для дискретных переменных, принимающих небольшое число значений, будут полезны команды табуляции `tabulate` или `table` — см. ниже.

`correlate` *переменные* [`if` *условие*] [`in` *диапазон*], [`covariance`]

Выводит матрицу корреляций между переменными в указанном диапазоне. Опция `covariance` указывает, что надо вывести ковариационную матрицу. Матрицы вычисляются по тем наблюдениям, для которых имеются значения *всех* указанных переменных.

`pwcorr` *переменные* [`if` *условие*] [`in` *диапазон*], `sig` `obs`

Выводит матрицу *попарных* корреляций, т. е. корреляций, рассчитанных по наблюдениям, в которых значения соответствующих переменных попарно не пропущены. Опция `sig` выводит уровень значимости корреляции (в предположении совместной нормальности), а `obs` — количество наблюдений.

`tabulate` *переменные* и `table` *переменные*

Построение различных таблиц, содержащих агрегированную информацию по переменным. Поддерживаются метки переменных и отдельных значений. Введение в эти команды дается уроком `tutorial tables`. См. также [U] **28 Commands for dealing with categorical variables**

`regress` *зависимая переменная объясняющие переменные* [`if` *условие*] [`in` *диапазон*], `robust`
`noconst` `cluster`(*групповая переменная*)

Оценивание линейной регрессии зависимой переменной на объясняющие. Выводятся основные результаты оценивания: количество наблюдений, таблица дисперсионного анализа, статистики F, R^2, R^2_{adj} , а также таблица оценок коэффициентов, стандартных отклонений оценок, t -статистик и доверительных интервалов (см. с. 55 с примером регрессии). Опция **robust** задает оценку ковариационной матрицы оценок коэффициентов в форме Вайта (2.29), учитывающей гетероскедастичность. Опция **cluster** указывает, что ковариационная матрица должна учитывать группировку наблюдений (как в стратифицированных обследованиях). Опция **noconst** указывает, что в модель, оцениваемую Stata, не следует включать константу (как это делается по умолчанию). После команды **regress** можно получать прогнозные значения, остатки и строить диагностические переменные командой **predict** или проводить диагностику регрессии, не прогоняя регрессию заново. Введение в эту команду предоставляется уроком **tutorial regress**.

Команды оценивания статистических моделей в Stata имеют много общего. В частности, после всех таких команд можно отдавать команду **predict**, которая будет строить значения тех или иных выражений, связанных с результатами оценивания; получать матрицы самих оценок параметров (матрица-столбец $e(b)$) и их ковариационную матрицу ($e(V)$); строить тесты на линейные (**test**) и нелинейные (**testnl**, с использованием дельта-метода для получения ковариационной матрицы нелинейных функций оценок) комбинации параметров, и т.д. Можно вывести результаты оценивания, не показанные в основном блоке вывода, командой **estimates list**. Отдельные коэффициенты можно получать в виде **_b[имя переменной]**, а их стандартные ошибки — **_se[имя переменной]**. Специфика команд, оценивающих параметрические модели, описана в разделах **help est** и **help postest** встроенной подсказки Stata.

В пакете Stata имеется широчайший спектр статистических команд, важных для эконометрического анализа⁴:

- регрессия с инструментальными переменными **ivreg**, робастная регрессия **rreg**, одновременные уравнения **reg3**, нелинейный МНК **nl**;
- модели временных рядов (**help time**): модели авторегрессии со скользящим средним **arima**; автокорреляции **ac** и частные автокорреляции **pac**; модели с условной гетероскедастичностью **arch**; регрессия с ковариационной матрицей Ньюи-

⁴ Подробности см. **help название команды**.

Веста (2.31) `newey`; проверка гипотез о единичных корнях временного ряда `dfuller`; `pperron`;

- обобщенные линейные модели (`glm`);
- средства дисперсионного анализа (`anova`; `oneway`; `loneway`),
- средства факторного анализа и анализа главных компонент (`factor`);
- средства анализа таблиц сопряженности (более подробные опции команд `table`; `tabulate`; `epitab`);
- средства анализа панельных моделей (команды с префиксом `xt`, например, `xtreg`, `re` и `xtreg, fe` — регрессии со случайным и фиксированным панельным эффектом; `xtgls` — регрессии с коррелированными остатками; `xtlogit` и `xtprobit` — панельные регрессии с бинарной зависимой переменной. Подробности см. `help xt`, а также [U] **29.13 Panel-data models**);
- средства анализа данных типа длительностей, или времени жизни, или времени отказа (`survival time`; команды с префиксом `st`; см. `help st`, а также [U] **29.14 Survival-time (failure time) models**);
- средства анализа стратифицированных обследований (`survey`; команды с префиксом `svy`; см. `help svy`, а также [U] **30 Overview of survey estimation**);
- средства анализа моделей с бинарной зависимой переменной (`logit`; `logistic`; `lfit`; `probit`; `dprobit` — предельные эффекты в пробит-модели);
- тесты на равенство средних (`ttest`), дисперсий (`sdtest`) и медиан (ранговые и знаковые тесты `signrank`; `signtest`; `ranksum`; `kwallis`);
- ранговые корреляции (`spearman`; `ktau`);
- возможность максимизации функций правдоподобия, запрограммированных пользователем (`ml`);
- в Stata 7 — исчерпывающий набор средств кластерного анализа;
- и многое, многое другое.

Полная стандартная поставка пакета Stata насчитывает около 500 команд для конечного пользователя (плюс большое число внутренних или программистских модулей). Примерно столько же содержится в официальных дополнениях (STB), и еще около полутысячи команд (по состоянию на конец 2000 г.) находится в интернетовском архиве SSC-IDEAS (см. раздел 3.16).

3.10 Посчитать: функции

В пакете Stata реализовано довольно большое число различных функций: математических (логарифмы, тригонометрические функции, модуль, корень и т. п.); статистические (плотности и функции распределения; в седьмой версии они приведены в некоторый порядок по сравнению с предыдущими), генератор псевдослучайных чисел (с периодом $\approx 2^{126}$, 2^{32} различными значениями и с возможностью инициализации пользователем для воспроизводимости вычислительных экспериментов), строковые функции, функции для работ с датами, функции от матриц (определитель, обратная), константа π (`_pi`) и ряд других. Полный список можно получить через `help functions` или [U] **16.3 Functions**, [R] **functions**. О возможностях написания пользовательских функций см. раздел 3.17.

3.11 Повторить: циклы

Stata обладает довольно своеобразными средствами повтора тех или иных команд для разных групп наблюдений, разных переменных и прочих случаев. Обычно этих средств хватает для выполнения требуемых задач, но иногда приходится прибегать к более изощренным трюкам.

by идентификатор(ы) групп : команда Stata

Эта команда повторяет указанную команду Stata отдельно для каждого набора наблюдений с одинаковыми значениями групповых переменных. Иными словами, Stata разбивает все данные на отдельные группы согласно групповым переменным (идентификаторам), и выполняет указанную команду для каждой из групп по отдельности. При этом указатель последнего наблюдения `_N` указывает на последнее наблюдение в группе. Необходимо, чтобы данные были отсортированы по этим групповым перемен-

ным, в противном случае Stata выдаст сообщение об ошибке.

`for min списка список [: for min списка список ...]`: команда Stata с символом X [Y] [
\ команда Stata с символом X [Y] ...]

Поддерживаемые типы списков: список чисел (numlist), список переменных (varlist), произвольный список (anylist).

Числа от 1 до 10 можно задать следующими способами: 1(1)10, или 1 2 to 10, или 1/10.

В списке переменных можно использовать переменные, стоящие друг за другом, через тире. Можно использовать * как заменитель любого символа: `u*` означает все переменные, начинающиеся на "u".

Подробнее о списках: [U] 14 Language syntax, help numlist, help varlist.

Команда `for` осуществляет цикл в обычном алгоритмическом понимании этого слова. Она перебирает элементы списка и подставляет их вместо X в исполняемой команде (командах). Если задано больше одного `for` через двоеточие, то Stata выполнит указанные команды для всех сочетаний X из первого списка × Y из второго, и т.д.

Stata честно пытается информировать пользователя о каждом отдельном значении групповых переменных `by` или параметра `for`, для которого выполняется очередное действие, и если список составляет несколько сотен, то весь процесс может оказаться несколько утомителен, да и вывод на экран иногда является самым медленным элементом вычислительного конвейера Stata. Чтобы команды ничего не выводили на экран, перед `for` и `by` можно задать, как и перед любой из команд Stata, префикс quietly, например: `qui for var x1-x5: g lX=log(X) \ lab var lX "log of X"`

В седьмой версии пакета возможности циклов несколько расширены командами `forvalues` и `foreach`.

3.12 Запомнить: результаты работы

Естественно, результаты работы по статистическому анализу данных не должны погибать вместе с концом сеанса Stata. Можно копировать эти результаты непосредственно из окна результатов Stata и через буфер обмена переносить в прочие приложения, однако есть более естественный способ.

`log using имя файла, [append | replace]`

`log on | off | close`

Эта команда записывает все, что Stata выводит в окно результатов, в указанный файл (добавляя либо перезаписывая этот файл, в соответствии с опциями `append` либо `replace`, если такой файл существует). `log off` временно прекращает запись в файл, `log on` возобновляет запись в файл, `log close` прекращает запись и закрывает файл. Команды, связанные с log-файлом, продублированы на панели инструментов Stata кнопкой со светофором. Log-файлы лучше всего печатать непосредственно из Stata, поскольку Stata умеет автоматически приукрашивать текст (выделяя полужирным шрифтом команды, проставляя даты и т.п.).

В Stata 7 есть два вида log-файлов: командный (в который пишутся только команды, отдаваемые пользователем, что дает возможность быстро конвертировать результаты работы в программу) и полный (в который пишутся как команды, так и результаты их исполнения). Запись команд в командный log-файл задается конструкцией `cmdlog using имя файла`. Есть также недокументированные способы записи log-файлов в форматах HTML и `texman` — `log html имя файла` и `log texman имя файла`.

Есть еще один вариант сохранения статистических результатов исследований — прекрасная пользовательская команда `outreg` Gallup (2001), которая записывает результаты регрессий в отдельный текстовый файл в соответствии с принятыми в статистической и эконометрической литературе обозначениями: столбцы коэффициентов со стандартными ошибками в скобках, число наблюдений, статистика R^2 и прочие статистики. Этот модуль требует, впрочем, отдельной установки, см. [R] `stb`, `help stb`. Самую свежую версию можно найти на сайте архива SSC-IDEAS, см. параграф 3.16.

Наконец, список нескольких последних команд можно получить командой `#review [количество команд]` .

См. также: [U] **Printing and preserving output**.

3.13 Запустить: do-файлы

Произвольную последовательность команд Stata можно записать в отдельный файл — не более одной команды в строке — и выполнить всю последовательность одной командой. Традиционно файлы, в которых записаны подготовленные таким образом программы, носят расширение `.do`, а команда, выполняющая эти do-файлы, так и называется:

`do` имя файла аргументы , [nostop]

Stata прекращает исполнение `do`-файла, когда натывается на ошибку. Можно этого избежать, установив опцию `nostop`.

Если не требуется вывод на экран, то вместо `do` можно запустить программу командой `run`. Впрочем, в случае аварийного останова Stata все равно выдаст сообщение об ошибке, вполне справедливо полагая, что пользователь должен об этом знать.

В текст `do`-файла можно вставлять комментарии, оформляемые в стиле языка программирования C, т. е. `/*` открывает комментарий, а `*/` — закрывает. Кроме того, строка, начинающаяся со звездочки `*`, также считается комментарием и полностью игнорируется. Эта строка, тем не менее, является командой, в том смысле, что Stata выводит ее в окно вывода и в `log`-файл. Можно таким образом вводить комментарии и при интерактивной работе.

Когда возможностей `for` не хватает, можно попробовать написать отдельный `do`-файл для выполнения требуемых действий и передавать ему `X` (или каким-то образом преобразованное выражение с `X`) в качестве одного из аргументов.

Можно дать несколько советов по созданию `do`-файлов ⁵.

- Для того, чтобы гарантировать воспроизводимость всех результатов, необходимо оформлять все полезные действия, вплоть до изменения значения одной переменной в одном наблюдении, как строки `do`-файла. Автору этих строк неоднократно приходилось выяснять вместе с коллегами, почему у них получаются разные результаты при использовании вроде бы одних и тех же методов обработки и анализа и вроде бы одних и тех же файлов данных, и именно для исключения подобных ситуаций и разработаны эти советы.
- На каждый отдельный исследовательский проект надо заводить отдельный каталог, а исходные данные сохранять неизменными и соответствующими исходным статистическим первоисточникам (справочникам, известным базам данных и т.п.), вынося все необходимые поправки и изменения в `do`-файлы. После существенных изменений в данных (таких, как команды `reshape`, `merge` или создания большого количества новых переменных — не забывайте придавать им мет-

⁵С разрешения Stata Corp., по материалам Net Course 151 по программированию в пакете Stata.

ки!) стоит сохранить полученные (промежуточные) результаты. Название файла должно отражать его происхождение или содержание, либо же должно содержать признаки того, что данные вторичны (например, можно начинать названия несущественных файлов с тильды или подчеркивания). Более подробную информацию о происхождении файлов данных можно записывать в эти файлы командами `label data` и `notes`.

- Стоит каждый do-файл начинать "с нуля", а log-файл, отслеживающий происходящее, должен открываться в том же do-файле. Отслеживать результаты работы будет проще всего, если имя log-файла совпадает с именем do-файла (или, если do-файл должен использовать какие-то параметры, эти параметры также должны фигурировать в названии log-файла, благо Stata поддерживает длинные имена Windows).

Примерная "рыба" do-файла, который называется, скажем, `income98.do`, будет такова:

```
clear
version 6
set memory 10m
log using income98, replace
use income98
* еще какие-то действия
...
log close
exit
```

Stata Corporation предлагает превосходные Internet-курсы по программированию в пакете Stata. Автор этого пособия участвовал в таких курсах и считает, что они заметно помогли ему в освоении возможностей пакета.

См. также: [U] **19 Do-files**

3.14 Нарисовать: графика

Мир графических средств пакета Stata начинается командой `graph`, у которой имеется добрая сотня разнообразных опций на разнообразные случаи жизни. Наиболее часто используемые графики реализованы в виде отдельных команд.

`graph` *переменные*, [*опции*]

Команда `graph` одна, но вариантов воплощения у нее очень много. Краткий рассказ об этих возможностях дается уроком `tutorial graphics`.

Если команда `graph` содержит одну переменную, то эта команда интерпретируется как задание построить гистограмму. По умолчанию Stata разбивает диапазон изменения переменной на пять интервалов (bins), что, как правило, недостаточно информативно, поэтому имеет смысл увеличить число интервалов опцией `graph ... , bin(50)`. Можно наложить поверх гистограммы плотность нормального распределения с аналогичным средним и дисперсией для визуального контроля нормальности с помощью опции `graph ... , norm`. Еще несколько разновидностей графиков, описывающих одну переменную, даются опциями `graph ... , box` (график box-whisker, отражающий основные квантили распределения⁶) | `star` (роза ветров) | `bar` (столбцовая диаграмма) | `pie` (круговая диаграмма). Более подробную помощь можно найти по ключевым словам `grhist` и `graph`.

Диаграмма рассеяния выводится командой `graph`, но с двумя аргументами: `graph` “ось y” “ось x”. Из основных опций (перечисляемых через запятую в командной строке), которые имеет смысл указывать для диаграммы рассеяния, стоит упомянуть:

- `symbol` — символ, которым будет помечаться выводимое наблюдение; `symbol(.)` выведет маленькую точку, `symbol(o)` — маленький кружок, `symbol([переменная])` — значение указанной переменной; `symbol([_n])` — номер наблюдения.
- `connect` — соединение точек; `connect(.)` означает, что точки соединять *не* надо, `connect(1)` — что точки надо соединить тонкой линией; `connect(s)` — провести сплайн через соседние точки. Сплайн является одним из видов *непараметриче-*

⁶На таком графике ящик (box) ограничен верхним и нижним квартилями, средняя линия ящика проводится на уровне медианы, а усы (whiskers) — это удвоенные разности между медианой и квартилями.

ской регрессии (см. параграф 2.6.5). В седьмой версии пакета можно задать стили линий, указывая их в квадратных скобках после символа, задающего соединение точек: `connect(1[-])` — пунктирная линия, `connect(1[_])` — длинная пунктирная линия, `connect(1[.])` — короткая пунктирная линия. Эти стили можно сочетать — `connect(1[-.])` выдаст штрих-пунктирную линию.

- `sort` — перед соединением точек, задаваемой опцией `connect`, отсортировать наблюдения по переменной на оси x (во избежание заполнения экрана паутинообразной ломаной).
- `bands` — количество соседних точек, используемых для вычисления сплайна. Чем ниже число, задаваемое этой опцией, тем более гладкой будет кривая непараметрической регрессии.
- `density` — количество точек на графике сплайна. Чем больше это число, тем более гладким будет изображение сплайна. Гладкость самого сплайна регулируется опцией `bands`.
- `xlab` и `ylab` — числовые метки на осях.
- `xtick` и `ytick` — “зарубки” на осях.
- `xline` и `yline` — вертикальные и горизонтальные линии на графике.
- `xscale` и `yscale` — диапазон осей.
- `title` — заглавие графика. В данном контексте Stata не понимает русский текст.

Эти и другие опции описываются в подсказке `grtwoway`.

Если в команде `graph` указать более двух переменных, то Stata построит графики зависимости всех переменных от последней, т.е. список переменных интерпретируется как y_1, \dots, y_{n-1}, x . Матрица попарных диаграмм рассеяния выводится с помощью опции `graph, matrix`.

Графики Stata можно сохранять в собственном формате `.gph`, указывая после любой графической команды опцию `graph ... , saving(имя файла)`. Эти сохраненные графики можно потом просмотреть заново командой `graph using имя файла(ов)`. Stata позволяет сочетать на одном рисунке несколько графиков — см. подсказку по команде

`help grother`. Кроме того, через меню `File` можно сохранять графику и в виде, понятном Windows-приложениям (в виде растровой графики `.bmp` или векторной `.wmf`), или переносить в другие приложения через буфер Windows.

Для встраивания графики Stata 6 в \LaTeX надо приложить определенные старания. Общее направление действий — сохранить в формате PostScript (`.ps`) → конвертировать в Encapsulated PostScript (`.eps`) или PDF → экспортировать в \LaTeX средствами пакета `graphicx`. В UNIX-овых версиях Stata есть утилиты `gphpen` и `gphprint`, которые позволяют сохранить графику в виде PostScript-файла. Пользователям Windows, видимо, необходимо установить PostScript-принтер и получать PostScript-файлы, посылая графику на печать и отмечая опцию "Печатать в файл". В качестве PS-принтера можно использовать и Acrobat Distiller из пакета Adobe Acrobat (*не* Acrobat Reader). Возможно, впрочем, что полученный таким образом файл будет содержать специфические команды принтера, поэтому его предварительно необходимо пропустить через конвертер, который оставил бы только "ортодоксальные" ключевые слова PostScript. Если используется пакет, создающий из \LaTeX овского файла PDF-файл (например, `pdflatex` пакета `fpTeX`), то можно из Stata отправлять графические файлы на печать в Acrobat Distiller и получать PDF-файлы с рисунками, которые потом, возможно, надо будет обрезать в Adobe Acrobat.

Все эти процессы упрощены в Stata 7 и делаются одной командой `translate`, которая конвертирует графику в форматы PostScript и Encapsulated PostScript.

См. также: [G]

3.15 Уточнить: команды для удовлетворения любопытства

В данном разделе будет рассказано о командах, показывающих определенные параметры состояния Stata, и о случаях, когда бывает полезна представляемая ими информация.

query

Выводит установки текущих параметров (в т. ч. размер матрицы, см. выше `set matsize`, уровень значимости по умолчанию статистических тестов `level`, в %, имя текущего log-файла, и т. п.). Эти установки можно менять с помощью команды `set`,

примеры которой приводились в разделе 3.2.

about

Выводит основные параметры Stata и компьютера, на котором работает пакет: версия программы, дата создания exe-файла, общий и доступный объем памяти.

memory

Выводит информацию о том, сколько памяти отведено для Stata и как она используется. Рекомендуется иметь памяти по меньшей мере на 15–20 % больше, чем требуется для данных, поскольку очень многие команды создают временные переменные, временные матрицы или используют память иным образом.

adopath

Выводит информацию о том, в каких каталогах Stata ищет ado-файлы с новыми программами (см. с. 72 об ado-файлах). Необходимо для установки новых компонент Stata (например, STB-дополнений при их ручном скачивании из Internet, см. параграф 3.17), а также при написании собственных программ в виде ado-файлов.

which *название команды*

Выводит информацию о том, в каком файле и в каком каталоге найден ado-файл, выполняющий требуемую команду, а также информацию о версии команды. Может оказаться полезным, если программа дорабатывается автором и необходимо отслеживать наличие последних версий, а также при появлении сообщений об ошибках для обращений в службу технической поддержки Stata или к автору программы.

3.16 Законnectиться: Internet-возможности Stata

Адрес Stata в Интернете — <http://www.stata.com/>. На этом корпоративном сайте размещаются новости (выход обновлений и новых версий, дополнений к Stata — STB, встреч пользовательских групп, объявления об Интернет-курсах по программированию и использованию пакета). Еще один очень полезный адрес — <http://ideas.uqam.ca/>. Здесь располагается поисковая система архива RePEc (Research Papers in Economics), умение пользоваться которой само по себе полезно для всякого экономиста. Одной из составных частей RePEc является архив программ SSC-IDEAS (Statistical Software Components), написанных пользователями Stata. В этом архиве содержится несколько сотен различных программных модулей, что вполне сопоставимо с количеством команд в

минимальном варианте установки. Из прочих ресурсов стоит упомянуть лист поддержки — statalist@hsphsun2.harvard.edu⁷, на котором можно получить квалифицированную помощь как от других пользователей Stata, так и от самих разработчиков, вплоть до президента корпорации Уильяма Гулда (William Gould). По его словам, оперативная и персональная поддержка пользователей — это один из важнейших приоритетов компании. От себя добавлю — это еще и одно из самых больших ее достоинств, особенно по сравнению с огромными монстрами типа SAS.

Начиная с шестой версии, Stata обладает рядом полезных возможностей, реализуемых через всемирную сеть Интернет. Это — обновление пакета, а также доступ к пользовательским программам.

update

Позволяет загрузить официальные обновления Stata через Интернет. Запрос **update query** показывает, что нужно обновить (статистические компоненты, находящиеся в ado-файлах, или исполняемый файл `wstata.exe`). Затем можно обновить необходимые фрагменты с помощью **update ado**, **update executable** или **update all**.

net [from URL]

Установка программ Stata через Internet. Эта команда ищет по указываемым Интернет-адресам (URL) — или, по умолчанию, на вышеуказанном сайте Stata — описания пакетов, которые может установить пользователь, скачивает необходимые файлы и устанавливает их на вашем компьютере.

webseek *ключевые слова*

Осуществляет поиск в Internet команд Stata, соответствующих указанным ключевым словами. **webseek** обращается на сервер Stata, на котором содержится информация о программах STB и других архивах программ Stata, по которым и осуществляется рекурсивный поиск. В седьмой версии команда **webseek** заменена на **net search**.

Помимо этих команд, работающих через Internet, Stata может выполнять многие действия, связанные с файлами, используя URL файлов вместо их имен. Так, вполне осмысленная команда

```
use http://www.stata.com/users/vwiggins/auto.dta
```

загрузит ценный файл `auto.dta`, на тот случай, если вы случайно стерли оригинал,

⁷ Чтобы подписаться на этот лист, надо послать письмо на адрес majordomo@hsphsun2.harvard.edu с текстом `subscribe stataлист`.

поставляющийся вместе с пакетом. Можно получать через Интернет текстовые файлы с данными и конвертировать их в файлы Stata командами `infile`, `infix`, `insheet`, и т.п.

Для корректной работы через прокси-сервер необходимо установить его параметры в меню `Prefs/General Preferences/Internet Prefs`.

См. также: [U] **32 Using Internet to keep up to date.**

3.17 Надстроить: расширение возможностей Stata

Stata — динамичный и открытый пакет. От одного до трех раз в месяц Stata выпускает обновления на уровне ado-файлов, доступные по команде `update`, и примерно раз в квартал выходят обновления исполняемого файла. Однако основная динамика происходит на листе `statalist` и на архиве программ SSC-IDEAS, где за день может появиться с десятков новых команд (написанных пользователями Stata для решения своих исследовательских задач, либо в качестве ответа на вопросы, задаваемые на `statalist`).

Stata публикует официальные дополнения к пакету под названием `Stata Technical Bulletin` (или, сокращенно, `STB`). Они выходят раз в два месяца, и состоят из программ (распространяемых бесплатно через Интернет) и небольшой брошюры, похожей по формату на руководства пользователя Stata. Для того, чтобы установить у себя программы из этих бюллетеней, надо отдать команды

```
net
net cd stb
```

или обратиться к меню `Help/STB and User-written Programs` для доступа к ado- и hlp-файлам на сервере Stata.

В предыдущих версиях Stata всех этих возможностей работы через Интернет не было, поэтому для установки STB-дополнений или программ из архива SSC-IDEAS было необходимо скачивать их вручную с Интернета, а потом либо копировать в каталог, зарегистрированный в `adopath` (см. с. 93), либо устанавливать средствами Stata — командой `install`. Пользователи 6-й или 7-й версии, у которых нет постоянного или хотя бы модемного соединения с Интернетом, будут вынуждены ходить с дискетами к знакомым, у которых доступ есть, скачивать необходимые команды на дискету, а потом устанавливать их командой `install from a:.`

Есть еще один технический момент, связанный с представлением текстовых файлов в Windows и UNIX. В этих двух операционных системах концы строк представляются по-разному, причем UNIX понимает тексты Windows, но не наоборот. В архиве SSC-IDEAS находятся программы, написанные в обоих форматах. При копировании командой `net Stata` корректно обрабатывает концы строк, однако при описанном выше “ручном” копировании возможны проблемы у пользователей, работающих в Windows. Симптомом того, что у вас возникла проблема, связанная с концами строк, является неработоспособность свежеставленных файлов — Stata возвращает ошибку с кодом 199 (`unrecognized command: xyz not defined by xyz.ado` — команда не распознана; программа `xyz` не определена в файле `xyz.ado`); при этом Stata находит файл помощи на новую команду, но в нем все оказывается перепутано. Эту проблему можно решить, открыв оба файла (`.ado` и `.hlp`) в текстовом редакторе и сохранив их обратно — есть вероятность, что концы строк при этом будут расставлены заново.

На определенном уровне владения пакетом оказывается удобным писать по разным случаям свои собственные программы (ado-файлы). Их можно публиковать их в Интернете для всеобщего доступа. Например, страничка автора этой книги, посвященная Stata, размещается по адресу: <http://www.komkon.org/~tacik/stata>. На ней находятся программы, уроки (tutorials) и PDF-файл с этой книгой.

Частным случаем пользовательских программ являются функции для команды `egen`. Они позволяют в какой-то степени обойти невозможность написания функций пользователя, применимых наравне со встроенными. Файлы, в которых содержатся такие функции, имеют префикс `_g` и должны быть написаны в соответствии с определенными требованиями на обработку входных аргументов.

3.18 Научиться на опыте: сообщения об ошибках

В соответствии с общепринятыми программистскими соглашениями, каждая команда и программа должна уметь сообщать о результатах своей работы. Чаще всего это делается в виде целочисленного кода завершения программы. Нулевое значение этого кода свидетельствует об отсутствии каких-либо ошибок и проблем при выполнении задания;

ненулевое, как правило, обозначает те или иные ошибки. Помимо кода завершения, многие программы Stata сохраняют те или иные результаты своей работы, которые можно получить, в зависимости от выполненной команды, через `estimates list` (вспомним обсуждение команды `regress` на с. 83) или `results list`. См. `help estimates` и `help results`.

В окне вывода Stata текст подсвечивается одним из пяти цветов: белым, желтым, зеленым, голубым или красным. Белым цветом показываються команды, отданные пользователем, а также некоторые специфические сообщения; голубым — команды, записанные в do-файле, а также запрос на продолжение вывода `--more--` (пауза в процессе обработки данных; для получения следующей строки вывода надо нажать **Enter**, следующей страницы — клавишу "пробел", как в программе `more` ОС UNIX); зеленым — информационный (постоянный) текст; желтым — рассчитываемые числовые значения (переменный текст); красным — сообщения об ошибках. Сообщения об ошибках сопровождаются кодом ошибки, по которым можно найти более подробную информацию в [R] **error messages** или через меню `Help/Search/rc код ошибки`. Чаще всего ошибки вызваны неправильным синтаксисом вводимых пользователем команд (использованием одинарного = в условиях `if`, ссылкой на несуществующую переменную из-за опечатки в названии переменной, ссылкой на несуществующую команду при опечатке в названии команды, попыткой создать вновь уже существующую переменную, и т.п.). Иногда, впрочем, ситуации могут быть более серьезными и свидетельствовать о статистических или вычислительных проблемах — например, когда не достигается сходимость итерационных процессов или не хватает наблюдений для оценивания модели — или проблемах компьютерных — нехватке памяти (сообщение `no room to add more variables`, см. выше `set memory`).

В пакете Stata 7 функции голубого цвета несколько изменены: он означает ссылку на файл встроенной подсказки, на URL в Интернете или просто на команду Stata. Можно навести мышку на фрагмент, показанный голубым цветом, и по нажатию левой кнопки мыши Stata покажет необходимый файл помощи, запустит браузер или выполнит необходимую команду. В частности, коды ошибок показываються голубым цветом, и при кликании на коде ошибки показывается файл подсказки, поясняющий, почему возникла данная ошибка.

См. также: [U] **11 Error messages and return codes**

3.19 Разобраться: прочее

В этом разделе приведены сведения, которые пригодятся уже при достаточно серьезном уровне владения пакетом и достаточно серьезных запросах к сложности программ.

Матрицы

Пакет Stata не является матричным, как, например, GAUSS. В нем, однако, реализовано большинство популярных матричных задач и алгоритмов: основные алгебраические действия, обращение, разложение Холецкого, решение задачи на собственные значения, сингулярное разложение. Столбцы и строки матриц можно “называть по именам” (что вполне естественно, например, для ковариационных матриц, возникающих при оценивании параметров статистических моделей). Знакомство с матричными средствами Stata можно начать с `help matrix`.

См. также: [U] **17 Matrix expressions**

Макросы

Наиболее близким к понятию макросов Stata является, пожалуй, понятие локальной переменной в программировании. Макросы — это строки, имеющие содержанием другие строки (в т.ч. числовые значения, записанные в экспоненциальном формате). С их помощью в программах Stata можно устраивать циклы, получать передаваемые подпрограмме значения, и т.п. Макросы делятся на локальные, которые будут забыты по окончании того процесса, который их создал⁸, и глобальные, доступные всем программам Stata. Среди глобальных макросов есть ряд зарезервированных, описывающих состояние Stata (версия, дата, время, режим работы, пути для поиска ado-файлов и т.п.). Ссылки на глобальные макросы Stata начинаются со знака доллара (\$). Так, уровень значимости, по умолчанию используемый для построения доверительных интервалов, обозначается как `$$_level` и равен по умолчанию 95 (в процентах).

См. также: [U] **21.3 Macros**

⁸ Собственно, эти макросы и являются аналогами локальных переменных.

Русификация

К сожалению российских пользователей, пакет Stata не русифицирован в том смысле, что у него отсутствуют русские описания. Теоретически и технически, русификация выводимых результатов и встроенной подсказки возможна, но объем работы измеряется, как мне кажется, несколькими человеко-годами, так что всерьез на это рассчитывать пока что не приходится.

Тем не менее, Stata может оперировать нелатинскими символами в качестве строк. Русские буквы можно использовать в качестве содержимого строковых переменных, для примечаний, меток переменных и данных, однако нельзя использовать в названиях переменных. Чтобы эти буквы отображались, надо в соответствующем окне (в первую очередь, в окне результатов) установить русские шрифты. Для этого надо ткнуть мышкой в пиктограмму окна в левом верхнем углу нужного окна (см. рис. 3.3) и установить какой-нибудь из русских шрифтов.

3.20 С чего начать?

Самое трудное — начать работу с пакетом в первые минуты, и это верно для любого программного средства. Один из важнейших навыков, которым необходимо овладеть с самого начала — это умение пользоваться встроенной подсказкой (см. раздел 3.5, а также подсказки по ключевым словам `help`, `winhelp`).

Другой хороший вариант самообучения и начала эффективной работы — воспользоваться встроенными мини-уроками `tutorials`. Достаточно набрать `tutorial` в командной строке Stata — и дальше Stata сама расскажет, какие мини-уроки у нее есть и как их вызвать. Первый мини-урок вызывается командой `tutorial intro`, и именно с него мы начинали наши практические занятия с пакетом Stata. Цель этих мини-уроков — не решить какую бы то ни было статистическую задачу, а показать, как работают те или иные команды в практической работе, поэтому при просмотре этих уроков надо обращать внимание не на то, что *выводит* Stata, а того, что в нее **вводится**.

Для данного курса прикладной эконометрики автором этого пособия была написана обучающая программа, демонстрирующая основные средства диагностики регрессий. Эта программа доступна со страницы <http://www.komkon.org/~tacik/stata> или, пользуясь интернет-возможностями Stata (см. раздел 3.16), из самого пакета:

```
. net from http://www.komkon.org/~tacik/stata  
. net get aboutreg
```

Темпы обучения, безусловно, индивидуальны, однако обычно уже нескольких часов достаточно для того, чтобы начать самому вводить команды и понимать, что они означают. Для профессионального овладения пакетом нужны, наверно, недели и месяцы постоянной работы с разными задачами и разными данными, отлаживание собственных программ и попытки разобраться в чужих, участие в интернет-курсах по пакету, предлагаемых разработчиками Stata Corp., участие в листе рассылки. Никакая книжка не может заменить самостоятельного активного освоения!