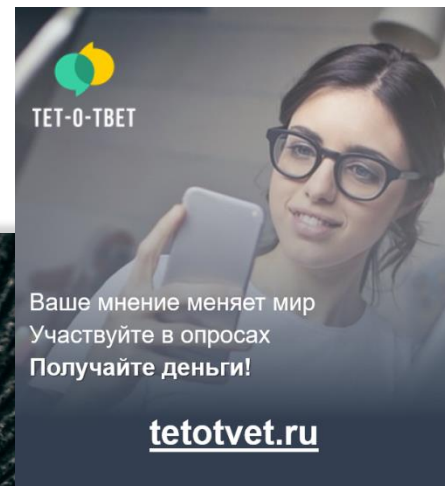


ТЕМА 13

РАБОТА С КОМАНДНЫМ СИНТАКСИСОМ. ОСНОВЫ.



Содержание

1. Синтаксис SPSS — Основы

- 1.1. Начало работы
- 1.2. Наборы переменных

2. Создание, изменение значений переменных

- 2.1. Условие if, range, any
- 2.2. Операторы, выражения

3. Создание и вычисление переменных

- 3.1. Работа с числовыми переменными
- 3.2. Работа с текстовыми переменными

4. Преобразования переменных

- 4.1. Формат переменных
- 4.2. Работа с пустыми ячейками
- 4.3. Перекодировка числовых переменных
- 4.4. Обработка переменных в цикле с помощью команды Do repeat
- 4.5. Удаление переменных
- 4.6. Переименование переменных

5. Подсчёт значений

- 5.1. Сумма, среднее, минимум, максимум значений переменных
- 5.2. Вычисление количества ответов, переменных
- 5.3. Работа с несколькими действиями с помощью команд Do if, Else, Else if

6. Работа с разными типами файлов (sav, excel)

- 6.1. Сохранение массива в формате «sav» (SPSS файл)
- 6.2. Сохранение массива в формате excel
- 6.3. Сохранение массива в формате csv

7. Работа с метками значений и переменных

- 7.1. Работа с командами Value labels, Variable labels
- 7.2. Работа с командами Apply Dictionary, Display Dictionary

8. Отбор и вывод результатов

- 8.1. Отбор и фильтрация данных
- 8.2. Частотные распределения, простые пересечения, отчёт по данным

1. СИНТАКСИС SPSS ОСНОВЫ



1. Синтаксис SPSS — Основы

Командный синтаксис (syntax) SPSS представляет собой набор текстовых функций для пакета SPSS IBM Statistics. Является своего рода языком программирования. По сути, это обычный текстовый файл с расширением «.sps». Создается либо с помощью пакета SPSS, либо как обычный текстовый файл с расширением «*.sps».

Имеет ряд преимуществ:

- работа с повторяющимися операциями на обновлённом либо новом массиве данных
- работа с учётом операций, требующих изменений
- создание и применение пользовательских функций
- возможность использования циклов для объёмных массивов данных
- более широкие возможности для работы со сложными операциями, касающимися трансформации данных
- наличие легко читаемого журнала с комментариями

1. Синтаксис SPSS — Основы

1.1. НАЧАЛО РАБОТЫ

Перед началом работы необходимо открыть программу SPSS. Далее в англоязычной версии пакета необходимо выбрать в верхней строке интерфейса вкладку **File — New — Syntax**. Кроме того, для удобства желательно настроить автозапуск журнала sps. Для этого в графическом редакторе необходимо выбрать **Edit — Options — General — open syntax window at startup**.

При выполнении синтаксиса, содержащего ошибки, программа напишет в окне и журнале вывода причину ошибки и, в зависимости от типа ошибки, остановится либо продолжит работать. Также программа может выдавать так называемые **«Warnings»**, своего рода предупреждения, которые не стоит игнорировать, а необходимо взять за привычку всякий раз их просматривать.

В файле командного синтаксиса (*.sps) рекомендуется комментировать описание выполняемых процедур на каждом этапе преобразования данных. Так, написав любой текст и поставив перед ним знак *, SPSS не будет воспринимать его как команду, но действует такая команда до ближайшей точки:

**Текст комментария.*

1. Синтаксис SPSS — Основы

1.2. НАБОРЫ ПЕРЕМЕННЫХ

Q1
Q2
HOBBY
Q8_1
Q8_2
Q8_3
Q8_4
Q8_5

Для переменных, идущих подряд в файле данных, в некоторых командах возможно применение наборов — списка переменных, перечисленных через пробел, запятую, ключевое слово «to» в качестве разделителя. Также возможно перечисление нескольких наборов и добавление к идущему диапазону других переменных.

Так, набор переменных от Q8_1 до Q8_5 можно задать разными способами.

Например:

- recode Q8_1 Q8_2 Q8_3 Q8_4 Q8_5 (1=2).
- recode Q8_1, Q8_2, Q8_3, Q8_4, Q8_5 (1=2).
- recode Q8_1 to Q8_5 (1=2).



2. СОЗДАНИЕ, ИЗМЕНЕНИЕ ЗНАЧЕНИЙ ПЕРЕМЕННЫХ

2. Создание, изменение значений переменных

2.1. УСЛОВИЕ IF, RANGE, ANY

Сгенерируем данные для работы с базой. Для этого в журнал syntax spss вставим следующий текст (скопировав отдельно каждый блок):

```
DATA LIST /
ID 1
AGE 3-4
GENDER 6
INCOME 8-9
EDUCAT 11
Q1 13
Q2 15
HOBBY 17-26 (A)
Q8_1 28
Q8_2 30
Q8_3 32
Q8_4 34
Q8_5 36.

BEGIN DATA
1 23 1 85 3 1 1 football 5 8 9 1 0
2 18 2 24 2 5 2 basketball 2 3 8 9 1
3 60 2 53 4 4 8 volleyball 2 2 9 8 6
4 38 2 156 4 3 1 4 1 1 2 6
5 43 1 120 5 4 Football 5 1 3 1 4
END DATA.
```


2. Создание, изменение значений переменных

Рассмотрим наиболее часто используемые команды пакета.

Сразу следует отметить, что **все команды SPSS, а также названия переменных не чувствительны к регистру**, что будет заметно во всех нижеперечисленных примерах.

КОМАНДА IF

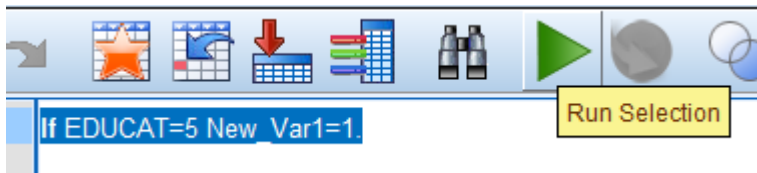
Общий пример: if (условие) переменная = действие.

Например, нам нужно отобрать людей с высшим образованием. Для этого сделаем дополнительную переменную Nv1 при условии, что в переменной EDUCAT стоит число 5.

Синтаксически данное условие можно прописать следующим образом*:

If EDUCAT=5 Nv1=1.

*Запуск любого условия с syntax осуществляется путем выделения строки с условием и нажатием кнопки **run** в графическом интерфейсе:



2. Создание, изменение значений переменных

Либо путём выделения строки с условием и нажатием горячих клавиш **Ctrl+R**.

Если необходимо запустить весь журнал с прописанными в нём условиями, можно нажать горячие клавиши **Ctrl+A** (выделение всего журнала), затем **Ctrl+R**.

Если необходимо запустить *часть* строк синтаксиса либо в режиме работы отладки, можно воспользоваться кнопкой **Toggle Breakpoint**, предварительно поставив курсор на нужную строку кода. Когда программа дойдёт до строки с красной меткой, она остановится.

При наличии вышеперечисленного условия в базе сразу же будет создана переменная `Nv1` с кодом 1. По умолчанию все новые переменные создаются в формате `f8.2` (ширина 8 знаков с двумя знаками после запятой).

Зачастую в условии применяются стандартные логические операторы **and**, **or**, **not(~)** и их комбинации.

2. Создание, изменение значений переменных

Например, необходимо сделать дополнительную переменную $Nv2=1$ при условии $Q2 = 2$ or $Q2=8$.

Условие можно прописать несколькими способами.

1. С помощью логического оператора **or**

If $(Q2=2$ or $Q2=8)$ $Nv2=1$.

2. С помощью функции **any**

If $any(Q2,2,8)$ $Nv2=1$.

Второй способ подразумевает наличие перечисленных кодов в вопросе $Q2$. Такой способ короче в написании и более удобен, если необходимых кодов в условии несколько.

Кроме того, функцию **any** удобно использовать при проверке отрицательных условий.

Например, то же самое условие, но наоборот:

If $\sim any(Q2,2,8)$ $Nv3=1$.

If not $any(Q2,2,8)$ $Nv3=1$.

В качестве отрицания используется знак тильда « \sim » .

Если необходимо сделать дополнительную переменную $Nv3=1$ при условии, $Q2=$ от 2 до 8, включительно достаточно неудобно писать длинное условие *If $(Q2=2$ or $Q2=3$ or or $Q2=4 \dots u$ т.д)* $Nv3=1$. Здесь следует воспользоваться функцией **range**. Например,

If $range(Q2,2,8)$ $Nv4=1$.

2. Создание, изменение значений переменных

2.2. ОПЕРАТОРЫ, ВЫРАЖЕНИЯ

Зачастую в условиях if используются операторы и /или:

And (&)

If (GENDER=1 And EDUCAT=5) Nv5=1.

If (GENDER=1 & EDUCAT=5) Nv5=1.

Or (|)

If (Q2=2 Or Q2=8) Nv6=1.

If (Q2=2 | Q2=8) Nv6=1.

2. Создание, изменение значений переменных

Как в условии, так и в действии могут быть использованы следующие арифметические операторы:

Оператор	Значение	Пример
+	Сложение	if $Q1+3=Q2$ $Q3=1$.
-	Вычитание	if $AGE>35$ $Q4=Q2-Q1$.
*	Умножение	if $(Q1*Q2)<8$ $Q5=1$.
/	Деление	if $Q5=1$ $Q6=Q2/3$.
**	Возведение в степень	if $Q1=3$ $Q7=Q4**2$.

2. Создание, изменение значений переменных


Задействуются следующие операторы сравнения:

Оператор	Эквивалент	Значение	Пример
=	EQ	Равно	if Q3=1 Q8=1.
>	GT	Больше	if GENDER>22 Q9=1.
>=	GE	Больше или равно	if Q2>=Q1 Q10=1.
<	LT	Меньше	if Q1>Q2 GROUP=1.
<=	LE	Меньше или равно	if Q2<=8 GROUP=2.
<>	NE ~= <>	Не равно	if Q1<>Q2 AND Q2<>8 GROUP=3.

ВАЖНО! Арифметические операторы, <>, < и **Not** не срабатывают на пустых значениях (System Missing), т.е. условие **if Nv2<>1 Nv7=1.** не работает, потому как в переменной Nv2 есть пропущенные значения.

В данном случае следует писать: **if Nv2<>1 or sys(Nv2) Nv7=1.**

3. СОЗДАНИЕ И ВЫЧИСЛЕНИЕ ПЕРЕМЕННЫХ



3. Создание и вычисление переменных

3.1. РАБОТА С ЧИСЛОВЫМИ ПЕРЕМЕННЫМИ

КОМАНДА COMPUTE

Создать либо вычислить переменную можно с помощью команды **compute**:

```
Compute Q5=$sysmis.
```

В данном случае в нашей базе уже есть переменная Q5, поэтому она будет полностью очищена.

Однако, если мы напишем следующие выражения, то в первом случае переменная примет значение =1, а во втором случае будет создана пустая переменная.

```
Compute Q5_D=1.
```

```
Compute Q5_D=$sysmis.
```



Совет

По [ссылке](#) вы можете ознакомиться с различными примерами количественных исследований НАФИ, для всех из них использовались команды синтаксиса

3. Создание и вычисление переменных

КОМАНДА NUMERIC

Создать пустую числовую переменную можно с помощью команды **numeric**.

Данная команда подразумевает создание именно новой переменной:

```
Numeric newVar1.
```

Если создаваемая переменная в базе присутствует, программа выдаст ошибку.

Аналогично всем новым переменным создается переменная с форматом f8.2, однако с помощью данной команды есть возможность задать нужный формат, указав при этом дополнительный параметр:

```
Numeric newVar2 (f2.0).
```

Кроме того, данная команда позволяет создать сразу несколько переменных, например:

```
Numeric TestVar_1 to TestVar_5 (f2.0).
```

3. Создание и вычисление переменных

КОМАНДА VECTOR

`vector VecVar_(10).`

Команда создаст 10 новых переменных `VtVar1` — `VtVar10`.

Удобна для создания переменных с индексами по порядку:

`vector VecVar2_(10,f4.0).`

3. Создание и вычисление переменных

3.2. РАБОТА С ТЕКСТОВЫМИ ПЕРЕМЕННЫМИ

Новую текстовую переменную можно создать с помощью следующей команды:

```
string TxtVar(A20).
```

«A20» обозначает длину текстовой переменной в размере 20 символов.

Специфика работы с текстовыми переменными заключается в том, что текстовые условия подразумевают *наличие кавычек*.

Допустим в переменной HOBBY необходимо выделить тех, кто увлекается баскетболом.

Для этого необходимо прописать следующее условие:

```
If (HOBBY='basketball') DopVar1=1.
```

```
exe.
```

Однако, следует обратить внимание, что данное условие подразумевает полное соответствие искомого текста, то есть все нижеприведенные условия на самом деле разные:

```
If (HOBBY='football') DopVar2 =1.
```

```
If (HOBBY='Football') DopVar3 =1.
```

```
If (HOBBY=' Football') DopVar4 =1.
```

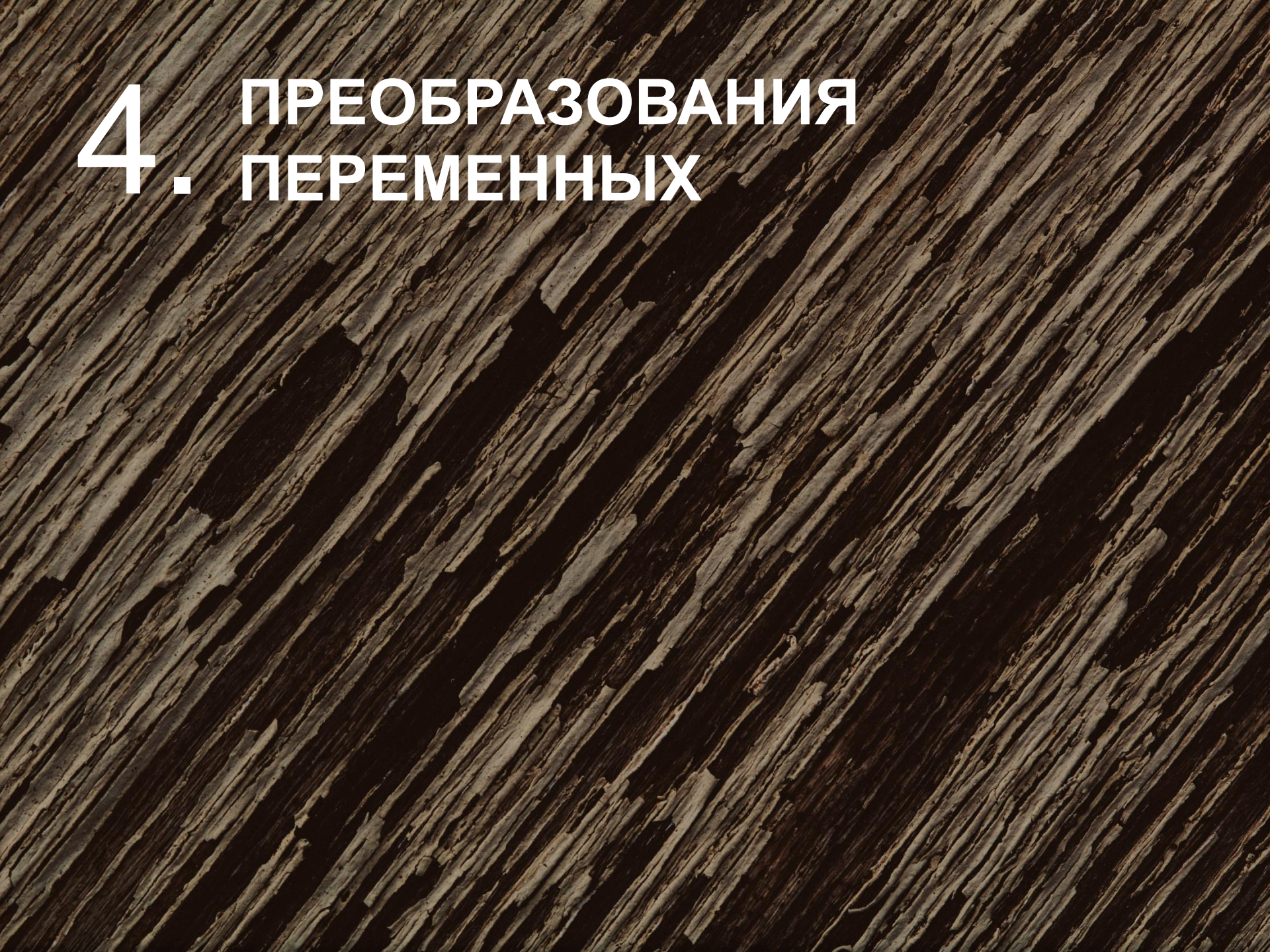
```
If (HOBBY='Foot ball') DopVar5 =1.
```

```
exe.
```

Данная команда позволяет создать ряд текстовых переменных разного формата, например:

```
string TxtVar1 to TxtVar5 (A5) / NTxtVar1 to NTxtVar5 (A15).
```

4. ПРЕОБРАЗОВАНИЯ 4. ПЕРЕМЕННЫХ



4. Преобразования переменных

4.1. ФОРМАТ ПЕРЕМЕННЫХ

КОМАНДА FORMATS

Изменить формат *числовой* переменной можно командой **Formats:**

Formats EDUCAT (f4.0).

4 – ширина (кол-во знаков *до запятой*)

0 – кол-во знаков *после запятой*

4. Преобразования переменных

КОМАНДА ALTER TYPE

С помощью команды **alter type** можно изменить не только длину, но и формат переменной.

Например, необходимо сделать текстовую переменную INCOME с длиной в 20 символов:

```
alter type INCOME (a20).
```

Необходимо быть особенно внимательным при изменении текстового формата на числовой. Если мы изменим формат текстовой переменной HOBBY на числовой, данные исчезнут:

```
alter type HOBBY (f20.0).
```

С помощью данной команды удобно менять размер текстовой/числовой переменной:

```
Alter type HOBBY (A50).
```

```
Alter type EDUCAT (f5.0).
```

Можно комбинировать несколько форматов в одной строке:

```
Alter type Q3 to Q5(F2) q6(A10).
```

4. Преобразования переменных

4.2. РАБОТА С ПУСТЫМИ ЯЧЕЙКАМИ

Если в числовой переменной Q2 пусто, то присвоить Q2mis=1.

```
if (sys(q2)) Q2mis=1.
```

Если в числовой переменной Q2 НЕ пусто, то присвоить Q2mis=2.

```
if (~sys(q2)) Q2mis=2.
```

Если в текстовой переменной HOBBY пусто, то присвоить HB_mis=1.

```
if (HOBBY="") HB_mis=1.
```

Если в текстовой переменной HOBBY НЕ пусто, то присвоить HB_mis=2.

```
if (HOBBY<> "") HB_mis=2.
```

Очистить числовую переменную q2 по условию (q1=1) можно следующим образом:

```
if q1=1 q2=$sysmis.
```

4. Преобразования переменных

4.3. ПЕРЕКОДИРОВКА ЧИСЛОВЫХ ПЕРЕМЕННЫХ

КОМАНДА RECODE

Позволяет перекодировать значения переменной. Данная команда перекодирует как текущую переменную, так и с занесением результатов в новую переменную.

Рассмотрим все варианты на примерах.

Допустим, нам необходимо перекодировать код 8 в код 3 в переменной Q2:

Recode Q2 (8=3).

По сути, мы заменили один код на другой, но бывают случаи, когда необходимо сделать группировки либо дополнительные переменные:

Пример 1. Коды 1 → 3, 2 → 4, 3 → 7, все остальные (не указанные коды) перекодируются в пусто:

```
recode Q1 (1=3)(2=4)(3=7)(else=sys).
```

Пример 2. Коды 1 → 8, 2 → 4, 3 → 6, все остальные (не указанные коды) перекодируются в 0:

```
recode Q1 (1=8)(2=4)(3=6)(else=0).
```


4. Преобразования переменных

Пример 3. Сделать группировку — От 18 до 24 →1, от 25 до 34 →2, от 35 до 44 →3, от 45 до 54 →4, 55 и выше →5 записать в новую переменную age_group:

```
RECODE AGE (18 thru 24=1) (25 thru 34=2) (35 thru 44=3) (45 thru 54=4) (55 thru higher=5) into AGE_GROUP.
```

Пример 4. Коды 1 → 3, 2 →4, 3 → 7, все остальные (не указанные коды) останутся в первоначальном виде:

```
recode Q1 (1=3)(2=4)(3=7)(else=copy).
```

Пример 5. Коды от 30 и ниже →99:

```
recode AGE (lo thru 30=99).
```

Пример 6. Все ответы → пусто:

```
recode Q1 (else=sys).
```

Пример 7. Возможен вариант взаимозаменяемой перекодировки:

```
recode Q1 (5=4) (4=5).
```

Пример 8. Зачистка всех ответов в Q1:

```
recode Q1 (else=sys).
```

Пример 9. Перекодировка по условию:

```
do if (GENDER=1).
```

```
    recode q2 (1=5)(2=4)(4=2)(5=1).
```

```
end if.
```

4. Преобразования переменных

4.4. ОБРАБОТКА ПЕРЕМЕННЫХ В ЦИКЛЕ С ПОМОЩЬЮ КОМАНДЫ DO REPEAT

Удобно при использовании преобразований с одной либо несколькими переменными.

Параметрами являются как набор переменных, так и набор цифр, однако при использовании нескольких наборов, количество элементов в каждом наборе должно быть одинаковым.

Пример 1. Если в интервале a есть значение (>0), то скопировать его в соответствующую переменную интервала b:

```
do repeat a=Q8_1 to Q8_5 / b=N5_1 to N5_5.
```

```
    if a>0 b=a.
```

```
end repeat.
```

```
exe.
```

При отсутствии переменных N5_1 to N5_5 в массиве, программа создаст их автоматически.

4. Преобразования переменных

Пример 2. Перенос кодов из одной переменной (q2) в несколько (перенос в Q8) с соответствующим кодом:

```
do repeat a=q8_1 to q8_10 / b=1 to 10.
```

```
    if q2=b a=b.
```

```
end repeat.
```

```
exe.
```

Пример 3. Очистка нескольких переменных по условию:

```
do repeat a=q8_1 to q8_5,q2.
```

```
    if q1=2 a=$sysmis.
```

```
end repeat.
```

```
exe.
```

4. Преобразования переменных

4.5. УДАЛЕНИЕ ПЕРЕМЕННЫХ

Переменную можно удалить с помощью команды синтаксиса: **DELETE VARIABLES:**
DELETE VARIABLES AGE_GROUP.

Удалять можно сразу несколько переменных с разделителем «пробел» либо «запятая»:

```
DELETE VARIABLES AGE,AGE_GROUP,GENDER.
```

```
DELETE VARIABLES Q8_1,Q8_2,Q8_3.
```

4.6. ПЕРЕИМЕНОВАНИЕ ПЕРЕМЕННЫХ

Переименование переменных осуществляется с помощью команды **RENAME VARIABLES**, например:

```
RENAME VARIABLES
```

```
(Q1= Q1N).
```

5

ПОДСЧЕТ
ЗНАЧЕНИЙ



5. Подсчет значений

5.1. СУММА, СРЕДНЕЕ, МИНИМУМ, МАКСИМУМ ЗНАЧЕНИЙ ПЕРЕМЕННЫХ

Рассчитать данные показатели с помощью синтаксиса можно на примерах ниже:

Сумма значений переменных q8_1-q8_5:

compute SumVar = sum(q8_1 to q8_5).

Среднее арифметическое значений переменных q8_1-q8_5:

compute MeanVar = mean(q8_1 to q8_5).

Минимальное значение переменных q8_1-q8_5:

compute MinVar = min(q8_1 to q8_5).

Максимальное значение переменных q8_1-q8_5:

compute MaxVar = max(q8_1 to q8_5).

Q8_1	Q8_2	Q8_3	Q8_4	Q8_5	SumVar	MeanVar	MinVar	MaxVar
5	8	9	1	0	23,00	4,60	,00	9,00
2	3	8	9	1	23,00	4,60	1,00	9,00
2	2	9	8	6	27,00	5,40	2,00	9,00
4	1	1	2	6	14,00	2,80	1,00	6,00
5	1	3	1	4	14,00	2,80	1,00	5,00

Важно помнить, что пустые значения в подсчетах не учитываются.

5. Подсчет значений

5.2. ВЫЧИСЛЕНИЕ КОЛИЧЕСТВА ОТВЕТОВ, ПЕРЕМЕННЫХ

КОМАНДА COUNT

Осуществляет подсчет количества ответов и кодов.

Пример 1. Подсчет количества ответов с кодом 1:

Count CQ8_1= Q8_1 to Q8_5 (1).

Пример 2. Подсчет количества ответов со значениями 2 или 4:

Count CQ8_2= Q8_1 to Q8_5 (2,4).

Пример 3. Количество кодов от 3 до 6:

Count CQ8_3= Q8_1 to Q8_5 (3 thru 6).

Пример 4. Количество кодов от 4 и выше:

Count CQ8_4= Q8_1 to Q8_5 (4 thru hi).

Пример 5. Подсчет всех имеющихся кодов в переменных:

Count CQ8_5= Q8_1 to Q8_5 (lo thru hi).

5. Подсчет значений

5.3. РАБОТА С НЕСКОЛЬКИМИ ДЕЙСТВИЯМИ С ПОМОЩЬЮ КОМАНД DO IF, ELSE, ELSE IF

Если EDUCAT=1, то q9=8 и q10=12:

```
do if EDUCAT=1.  
  compute q9=8.  
  compute q10=12.  
end if.
```

Если EDUCAT=2, то условие1,
иначе условие2:

```
do if EDUCAT=2.  
  compute q9=8.  
  compute q10=12.  
else.  
  compute q9=6.
```

```
  compute q10=11.  
end if.
```

РАБОТА С ИСПОЛЬЗОВАНИЕМ ELSE IF:

```
do if EDUCAT=1.  
  compute q13=1.  
else if EDUCAT=2.  
  compute q20=6.  
else if EDUCAT=3.  
  compute q20=7.  
else.  
  compute q20=99.  
end if.
```




6

РАБОТА С РАЗНЫМИ ТИПАМИ ФАЙЛОВ (SAV, EXCEL)

6. Работа с разными типами файлов (sav, excel)

6.1. СОХРАНЕНИЕ МАССИВА В ФОРМАТЕ «SAV» (SPSS ФАЙЛ)

Для открытия файла *.sav файл необходимо использовать команду **get file:**
get file='C:\AllBases\Base.sav'.

Сохранение файла осуществляется командой **save outfile:**
save outfile='C:\AllBases\BaseUpd.sav'.

Таким образом сохранится весь файл SPSS. Однако, бывают случаи, когда необходимо сохранить только часть данных либо сохранить массив без определенных переменных.

В таком случае удобно использовать ключи **/KEEP** и **/DROP**.

Пример 1. Массив будет сохранен *только* с переменными ID Q1 Q3 HOBBY:
save outfile='C:\AllBases\BaseK.sav'
/KEEP ID Q1 Q3 HOBBY.

Пример 2. Массив будет сохранен, *исключая* переменные ID Q1 Q3 HOBBY:
save outfile='C:\AllBases\BaseD.sav'
/DROP ID Q1 Q3 HOBBY.

6. Работа с разными типами файлов (sav, excel)

6.2. СОХРАНЕНИЕ МАССИВА В ФОРМАТЕ EXCEL

Сохранить массив в excel файл можно с помощью данной команды:

```
SAVE TRANSLATE OUTFILE='C:\AllBases\BaseValues.xlsx'
```

```
/TYPE=XLS
```

```
/VERSION=12
```

```
/MAP
```

```
/REPLACE
```

```
/FIELDNAMES
```

```
/CELLS=values.
```

Параметр /Cells принимает значения **values** (выгруженный массив excel будет содержать только коды значений вместо меток значений) или **labels** (выгруженный массив excel будет содержать только метки значений вместо кодов значений). Также в данном синтаксисе можно использовать ключи **/KEEP** и **/DROP**.

6. Работа с разными типами файлов (sav, excel)

6.3. СОХРАНЕНИЕ МАССИВА В ФОРМАТЕ CSV

```
SAVE TRANSLATE OUTFILE='Base.csv'
```

```
/type=csv /replace /textoptions delimiter="," decimal=dot format=variable /fieldnames.
```

Здесь также возможно использование параметра «**/cells=labels**».



7

РАБОТА С МЕТКАМИ ЗНАЧЕНИЙ И ПЕРЕМЕННЫХ

7. Работа с метками значений и переменных

7.1. РАБОТА С КОМАНДАМИ VALUE LABELS, VARIABLE LABELS

Допустим нам необходимо посмотреть частотное распределение переменной AGE_GROUP:

FREQUENCIES AGE_GROUP.

AGE_GROUP				
	Frequency	Percent	Valid Percent	Cumulative Percent
Valid 1,00	2	40,0	40,0	40,0
3,00	2	40,0	40,0	80,0
5,00	1	20,0	20,0	100,0
Total	5	100,0	100,0	

Как видно из журнала вывода, в переменной нет меток значений, поэтому работать с такой базой неудобно. В spss есть возможность «навешивания» меток как на значения, так и на переменные.

Общий синтаксис меток значений:

Value labels Q1

1 'ValueText1'

2 'ValueText2'.

7. Работа с метками значений и переменных

Value labels AGE_GROUP

1 "18 - 24"

2 "25 - 34"

3 "35 - 44"

4 "45 - 54"

5 "55 +".

AGE_GROUP					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1 18 - 24	2	40,0	40,0	40,0
	3 35 - 44	2	40,0	40,0	80,0
	5 55 +	1	20,0	20,0	100,0
	Total	5	100,0	100,0	

Команда `add val lab` добавляет новые метки значений:

Add value labels AGE_GROUP

99 "Отказ от ответа".

Для целого набора переменных:

Value labels Q8_1 to Q8_5

1 "ValueText1"

2 "ValueText2".

Общий синтаксис меток переменных:

Variable labels INCOME

'Доход (тыс.руб) '.

При работе с командой можно использовать как одинарные, так и двойные кавычки.

7. Работа с метками значений и переменных

7.2. РАБОТА С КОМАНДАМИ APPLY DICTIONARY, DISPLAY DICTIONARY

Скопировать метки можно с помощью команды **Apply Dictionary**.

Apply Dictionary from * /source variables=q8_1 /target variables=q2.
exe.

Вывести метки значений и метки переменных в журнал output можно с помощью команды **Display Dictionary**:

Display Dictionary.

Возможно вывести метки значений в ряде переменных:

DISPLAY DICTIONARY /VARIABLES=Q8_1 TO Q8_5.

8

ОТБОР И ВЫВОД РЕЗУЛЬТАТОВ



8. Отбор и вывод результатов

8.1. ОТБОР И ФИЛЬТРАЦИЯ ДАННЫХ

КОМАНДА FILTER

Допустим, нам нужно посмотреть распределение по фильтру EDUCAT=5. Для этого воспользуемся командой **FILTER**.

```
compute flt=(EDUCAT=5).
```

```
filter by flt.
```

```
FREQUENCIES Q8_5.
```

```
filter off.
```

Q8_5

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	4	100,0	100,0	100,0

После применения данной команды была создана дополнительная переменная flt. Если фильтрующее условие выполняется (compute flt=(EDUCAT=5).), то создается переменная flt=1, в остальных случаях flt=0. Соответственно, частотное распределение выводится по условию flt=1. Фильтр убирается командой **filter off**.

8. Отбор и вывод результатов

КОМАНДА SELECT IF

Общий синтаксис команды: **select if** {условие}.

В отличие от команды `filter`, выделяет, обрезает из базы тех, кто подходит по необходимому условию. Варианты условий аналогичны возможностям команды `if`.
Например:

```
Select if EDUCAT=4.
```

```
FREQUENCIES Q8_5.
```

!ВАЖНО помнить, что данные, не соответствующие условию, безвозвратно обрезаются, вернуть их можно либо заново открыв файл, либо применяя команду **temp** (если используется только одна команда).

8. Отбор и вывод результатов

КОМАНДА TEMP

Представляет собой временное преобразование данных.

Следующая за temp команда будет применена временно и время ее действия – следующая за преобразованием команда до точки.

Команду удобно использовать для временного преобразования данных, нужного для вывода частотных распределений:

Temp.

Recode q1(4=5).

FREQUENCIES Q8_5.

Команда recode будет применена только для последующей команды FREQUENCIES, после значения вернутся на первоначальные.

!ВАЖНО помнить, что команда не может быть использована с SORT CASES, MATCH FILES, ADD FILES.

8. Отбор и вывод результатов

8.2. ЧАСТОТНЫЕ РАСПРЕДЕЛЕНИЯ, ПРОСТЫЕ ПЕРЕСЕЧЕНИЯ, ОТЧЁТ ПО ДАННЫМ

Вывести результаты значений можно при помощи следующих команд.
Для этого воспользуемся массивом «e_sport.sav».

КОМАНДА FREQUENCIES

Простое частотное распределение по переменной:
FREQUENCIES Q1.

Q1 Q1. Как часто Вы в среднем играете в компьютерные игры?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1 Каждый день	22	22,0	22,0	22,0
	2 4-5 раз в неделю	15	15,0	15,0	37,0
	3 2-3 раза в неделю	33	33,0	33,0	70,0
	4 Раз в неделю и реже	30	30,0	30,0	100,0
	Total	100	100,0	100,0	

8. Отбор и вывод результатов

Частотное распределение, отсортированное по частоте появления:
FREQUENCIES Q1 /format=DFREQ.

Q1 Q1. Как часто Вы в среднем играете в компьютерные игры?					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3 2-3 раза в неделю	33	33,0	33,0	33,0
	4 Раз в неделю и реже	30	30,0	30,0	63,0
	1 Каждый день	22	22,0	22,0	85,0
	2 4-5 раз в неделю	15	15,0	15,0	100,0
	Total	100	100,0	100,0	

8. Отбор и вывод результатов

КОМАНДА CROSSTABS

Простое пересечение в абсолютных значениях:

CROSSTABS Q1 by Q3.

Q1 Q1. Как часто Вы в среднем играете в компьютерные игры? * Q3 Q3. Знаете ли Вы или слышите сейчас впервые понятие «киберспорт»? Crosstabulation

Count

		Q3 Q3. Знаете ли Вы или слышите сейчас впервые понятие «киберспорт»?			Total
		1 Знаю хорошо	2 Что-то слышал(а)	3 Слышу впервые	
Q1 Q1. Как часто Вы в среднем играете в компьютерные игры?	1 Каждый день	8	10	4	22
	2 4-5 раз в неделю	8	4	3	15
	3 2-3 раза в неделю	11	14	8	33
	4 Раз в неделю и реже	12	14	4	30
Total		39	42	19	100

8. Отбор и вывод результатов

Простое пересечение в процентах по столбцам:

CROSSTABS Q3 by Q4

/cells=column.

Q3 Q3. Знаете ли Вы или слышите сейчас впервые понятие «киберспорт»? ^ Q4 Q4. На Ваш взгляд, является ли киберспорт видом спорта или нет? Crosstabulation

% within Q4 Q4. На Ваш взгляд, является ли киберспорт видом спорта или нет?

		Q4 Q4. На Ваш взгляд, является ли киберспорт видом спорта или нет?			Total
		1 Скорее да	2 Скорее нет	99 Затрудняюсь ответить	
Q3 Q3. Знаете ли Вы или слышите сейчас впервые понятие «киберспорт»?	1 Знаю хорошо	68,2%	19,6%		39,0%
	2 Что-то слышал(а)	27,3%	56,5%	40,0%	42,0%
	3 Слышу впервые	4,5%	23,9%	60,0%	19,0%
Total		100,0%	100,0%	100,0%	100,0%

8. Отбор и вывод результатов

КОМАНДА TABLES

Таблица в абсолютных значениях с базой:

TABLES

/FTOTAL = FTOT 'Base'

/TABLE = Q3 + FTOT BY Q4.

		Q4. На Ваш взгляд, является ли киберспорт видом спорта или нет?		
		1 Скорее да	2 Скорее нет	99 Затрудняюсь ответить
Q3. Знаете ли Вы или слышите сейчас впервые понятие «киберспорт»?	1 Знаю хорошо	30	9	
	2 Что-то слышал(а)	12	26	4
	3 Слышу впервые	2	11	6
Base		44	46	10

8. Отбор и вывод результатов

КОМАНДА STABLES

Для вывода таблицы вопроса с множественным выбором удобно использовать команду `ctables` (custom tables). Для этого необходимо создать «виртуальную» переменную с помощью команды «MRSETS», указав минимальное количество параметров – название сета, формулировку вопроса и все переменные набора.

```
MRSETS /MCGROUP NAME=$Q2 LABEL= "Q2. На каких устройствах Вы играете?"  
VARIABLES= Q2_1 to Q2_99.
```

Для построения такой таблицы переменные должны быть номинальными. Изменить шкалу можно командой `VARIABLE LEVEL:`
`VARIABLE LEVEL Q2_1 to Q2_99 (nominal).`

Простейший синтаксис данной команды:
`STABLES`

```
/TABLE $Q2 [COUNT F40.0].
```

Данная команда вывода таблиц удобна множеством настраиваемых команд и является наиболее часто используемой.

		Count	
\$Q2 Q2. На каких устройствах Вы играете?	1 На компьютере	75	
	2 Приставка Плейстейшн (Playstation)	9	
	3 Приставка Икс Бокс (X-Box)	4	
	4 Приставка Nintendo (Nintendo)	1	
	5 На смартфоне или планшете	43	
	98 Другое устройство	3	
	99 Отказ от ответа / Затрудняюсь ответить	1	

8. Отбор и вывод результатов

КОМАНДА SUMMARIZE

Отчет по перечисленным переменным. Выводятся метки вместо значений:

SUMMARIZE Q1,Q4 /format=list.

Report

	Case Number	Q1 Q1. Как часто Вы в среднем играете в компьютерные игры?	Q4 Q4. На Ваш взгляд, является ли киберспорт видом спорта или нет?
1	1	4 Раз в неделю и реже	1 Скорее да
2	2	3 2-3 раза в неделю	1 Скорее да
3	3	4 Раз в неделю и реже	2 Скорее нет
4	4	4 Раз в неделю и реже	1 Скорее да
5	5	4 Раз в неделю и реже	2 Скорее нет
6	6	4 Раз в неделю и реже	1 Скорее да
7	7	3 2-3 раза в неделю	99 Затрудняюсь ответить
8	8	4 Раз в неделю и	2 Скорее нет

Приложение 1

Таблица 1. Применение команды «exe»

EXE НУЖЕН	EXE НЕ НУЖЕН	
COMPUTE	ALTER TYPE	VECTOR
COUNT	APPLY DICTIONARY	WEIGHT
IF, DO IF	CROSSTABS	RENAME VARIABLES
DO REPEAT-END REPEAT	GET FILE	SAVE OUTFILE
LOOP-END LOOP	DATASET ...	SCRIPT
RECODE	DISPLAY (DICTIONARY)	SET
SELECT IF	FILTER	STRING
WRITE	FORMATS	SUMMARIZE
ADD FILES	FREQUENCIES (FRE)	TEMPORARY (TEMP)
MATCH FILES	INSERT (INCLUDE) FILE	VALUE LABELS
UPDATE FILE	NUMERIC	ADD VALUE LABELS
	VARIABLE LABELS	VARSTOCASES
	VARIABLE LEVEL	

Источники по теме 13

ЛИТЕРАТУРА:

1. Levesque R. SPSS Programming And Data Management: A Guide For SPSS And SAS Users Paperback, 2004.
2. Collier J. Using SPSS syntax: a beginner's guide, 2009.

РУКОВОДСТВА ПОЛЬЗОВАТЕЛЯ ИВМ:

1. Manual IBM SPSS Statistics: HELP-Command Syntax Reference-syntaxreference.pdf
2. <https://developer.ibm.com/predictiveanalytics/wp-content/uploads/sites/48/2015/04/Programming-and-Data-Management-for-IBM-SPSS-Statistics-23.pdf>
3. https://www.ibm.com/support/knowledgecenter/en/SSLVMB_23.0.0/statistics_reference_project_ddita-gentopic2.html



Для свободного использования в образовательных целях
Copyright 2018 © Академия НАФИ. Москва
Все права защищены
www.nafi.ru