

## **Занятие 4. Формирование лонгитюдных массивов за все раунды. Лонгитюдный семейный файл.**

Общая рекомендация ко всем выполняемым вами заданиям:

- 1) Сохраняйте исходные файлы под новым именем, чтобы работать с ними.
- 2) **СРАЗУ открывайте и сохраняйте файл аутпута.** Первая команда в нем должна быть

вида:

**\*Фамилия – номер семинара – номер задания**

3) Сохраняйте проделанную вами работу в виде кода, используя «сохранение» правильных команд в STATA (или функцию “paste” SPSS). В этом случае вы сможете дома повторить все сделанное вами в классе. Кроме того, рекомендуется прикладывать программу к вашим исследованиям.

4) В качестве отчета за семинар нужно предъявить созданные файлы данных, файл аутпута, и файл с кодом.

### **Исходные файлы.**

*Данные по домохозяйства за 6, 7 и 8 раунды, разделы общий, E и F*

r06hall41\_ef.dta

r07hall41\_ef.dta

r08hall41\_ef.dta

Эти файлы представляют собой части семейных файлов за 1995, 1996 и 1998 годы (5, 7 и 8 волна), оставлены только сервисные переменные и переменные из разделов E (расходы) и F (доходы), для учебных целей.

### **17. «Склеивание» семейных файлов из разных раундов «по вертикали» (т.е. в файлах должен быть одинаковый список переменных с одинаковыми именами, данные разных лет приклеиваются друг под другом).**

«Склеивание» файлов таким образом нужно для того, чтобы, во-первых, иметь возможность во всех волнах производить одинаковые изменения (перекодирование переменных, расчет новых переменных – например, душевого дохода и т.д.), и, во-вторых, оценивать регрессионные модели на всех совокупности раундов (вводя дамми для номера раунда) или панельные регрессии.

Для сравнения файлов и переменных из разных раундов можно использовать файлы

**sravn\_sem\_5\_24.xls** - сравнение семейных файлов

**sravn\_ind\_5\_24.xls** - сравнение индивидуальных файлов

Для того, чтобы таким образом соединить файлы, нужно чтобы во всех файлах одинаковые переменные имели одинаковые названия во всех раундах, так как в файлах за отдельный год все имена начинаются с одной буквы. Нужно также проверить, нет ли различий в написании имени одной и той же переменной в разные годы (сейчас все такие случаи уже исправлены).

Примерно одинаковая структура у раундов 5,6,7,8 – их можно соединить в один файл наиболее безболезненно.

Работа будет состоять из нескольких этапов:

1) Для каждого нужного раунда:

1.1) переименовать переменные в файле, убрав первую букву-идентификатор раунда, а также переименовав те переменные, которые в последующих раундах имели другие имена;

1.2) создать переменную-идентификатор номера раунда и единый «семейный» идентификатор для удобства склеивания индивидуальных и семейных данных за один

и тот же год;

1.3) склеить файлы

1.4) если необходимо, преобразовать переменные, которые были изменены в новом раунде.

17.1. Используем директорию на диске

**C:\RLMS\_work\seminar\_2\data**

Это позволит вам использовать готовые коды для переименований файлов 6-7-8 волн. Распакуйте архив данными.

Откройте программу STATA

Начните с открытия файла аутпута, назвав его своей фамилией.

**log using " C:\RLMS\_work\seminar\_4\data\seminar\_4\_Roshchina.smcl"**

Первая команда должна быть такая (тем самым вы подписываете ваш аутпут)

**\*Фамилия - номер семинара**

Если у вас 14я STATA, набираем команду;

**set more off**

Желательно также делать комментарии с номером задания (начинающиеся со \*), так как этот файл – ваш главный отчет по работе за семинар.

17.2. Переименование файла 7й волны.

\*Откроем файл семейных данных за 1996 год (7я волна).

**use "C:\RLMS\_work\seminar\_4\data\r07hall41\_ef.dta", lear**

Сохраним этот файл под новым именем

**save "C:\RLMS\_work\seminar\_4\data\hh07\_ef.dta"**

Все переменные пятого раунда, кроме идентификаторов, начинаются с буквы "а", шестого раунда – "b", седьмого раунда - "с", восьмого раунда – "d" и т.д. Эту букву в именах переменных нужно удалить, иначе склеить данные не получится..

**Необходимо НЕ ПЕРЕИМЕНОВЫВАТЬ переменные - идентификаторы семьи** (за все раунды), т.е. переменные aid\_h, bid\_h, id\_h, did\_h.

Будем использовать команду «переименовать группу переменных»: **rename (old1 old2 ...) (new1 new2 ...) [, options1]**

Переименовывать нужно переменные, начинающиеся с буквы, обозначающей номер раунда (раунда 5 - "а", раунда 6 – "b", раунда 8 – "d" и т.д.). Список всех переменных можно сформировать, перенося их в командную строку.

Необходимо проверить, чтобы в список не попали переменные, которые переименовывать не нужно - (ВСЕ идентификаторы за все раунды, **psu, ssu, stsus** и т.д.). Это список поместить в первые скобки в команде.

Затем скопировать этот список переменных во вспомогательный файл (это м.б. word и т.д.) и в этом вспомогательном файле заменить 'с' (пробел я буква ) на 2 пробела.

В этом же вспомогательном файле в получившихся переменных **\_origsm, \_hhwgt, \_nfm** убрать знак нижнего подчеркивания, т.е. получатся переменные **origsm, hhwgt, nfm**.

Важно следить, чтобы не пропустить ни одну переменную; а также чтобы в старом и новом списке переменных их было одинаковое количество и имена соответствовали друг другу.

Должна получиться команда:

```
rename (credid_h c_origsm c_hhwgt ca3 ca3_3 ca4_1 ca4_2 ca5_1 ca5_2 ca8 c_nfm ce2 ce3 ce4
ce5 ce6 ce7_1a ce7_1b ce7_2a ce7_2b ce7_3a ce7_3b ce7_4a ce7_4b ce7_5a ce7_5b ce7_6a
ce7_6b ce7_7a ce7_7b ce7_8a ce7_8b ce8_1a ce8_1b ce8_2a ce8_2b ce8_3a ce8_3b ce9_1a
ce9_1b ce9_2a ce9_2b ce9_3a ce9_3b ce9_4a ce9_4b ce9_4_1a ce9_4_1b ce9_5a ce9_5b ce9_6a
ce9_6b ce9_7a ce9_7b ce10 ce11 ce12 ce12_1 ce13_1a ce13_1b ce13_2a ce13_2b ce13_3a ce13_3b
ce13_31a ce13_31b ce13_4a ce13_4b ce13_5a ce13_5b ce13_6a ce13_6b ce13_7a ce13_7b
ce13_8a ce13_8b ce14 ce15 ce16 ce17 ce18 ce19_1a ce19_1b ce19_2a ce19_2b ce19_3a ce19_3b
ce19_4a ce19_4b ce19_5a ce19_5b cf1 cf2 cf3 cf4 cf5 cf6 cf7 cf8 cf9_1a cf9_1b cf9_2a cf9_2b
cf9_3a cf9_3b cf9_4a cf9_4b cf9_5a cf9_5b cf9_6a cf9_6b cf9_7a cf9_7b cf9_8a cf9_8b cf9_9a
cf9_9b cf9_10a cf9_10b cf10 cf11 cf12_1a cf12_1b cf12_2a cf12_2b cf12_3a cf12_3b cf12_4a
cf12_4b cf12_5a cf12_5b cf12_6a cf12_6b cf12_7a cf12_7b cf12_8a cf12_8b cf12_9a cf12_9b
cf12_10a cf12_10b cf13_1a cf13_1b cf13_2a cf13_2b cf14 cf15) (redid_h origsm hhwgt a3 a3_3
a4_1 a4_2 a5_1 a5_2 a8 nfm e2 e3 e4 e5 e6 e7_1a e7_1b e7_2a e7_2b e7_3a e7_3b
e7_4a e7_4b e7_5a e7_5b e7_6a e7_6b e7_7a e7_7b e7_8a e7_8b e8_1a e8_1b e8_2a e8_2b
e8_3a e8_3b e9_1a e9_1b e9_2a e9_2b e9_3a e9_3b e9_4a e9_4b e9_4_1a e9_4_1b e9_5a
e9_5b e9_6a e9_6b e9_7a e9_7b e10 e11 e12 e12_1 e13_1a e13_1b e13_2a e13_2b e13_3a
e13_3b e13_31a e13_31b e13_4a e13_4b e13_5a e13_5b e13_6a e13_6b e13_7a e13_7b
e13_8a e13_8b e14 e15 e16 e17 e18 e19_1a e19_1b e19_2a e19_2b e19_3a e19_3b e19_4a
e19_4b e19_5a e19_5b f1 f2 f3 f4 f5 f6 f7 f8 f9_1a f9_1b f9_2a f9_2b f9_3a f9_3b f9_4a
f9_4b f9_5a f9_5b f9_6a f9_6b f9_7a f9_7b f9_8a f9_8b f9_9a f9_9b f9_10a f9_10b f10
f11 f12_1a f12_1b f12_2a f12_2b f12_3a f12_3b f12_4a f12_4b f12_5a f12_5b f12_6a
f12_6b f12_7a f12_7b f12_8a f12_8b f12_9a f12_9b f12_10a f12_10b f13_1a f13_1b f13_2a
f13_2b f14 f15)
```

Если вы пропустили какие-то переменные в этом списке, их можно переименовать по одной, чтобы потом можно было склеить файлы за разные волны.

Создадим вначале нашего файла две новые переменные, с такими же именами, как в лонгитюдных файлах с сайта. Первая – идентификатор волны (номер раунда).

```
generate id_w = 7, before(redid_h)
```

Вторая переменная – идентификатор семьи, которая может использоваться только для склейки внутри каждого раунда (но ни в коем случае для переменных с лагом). Эта переменная для любого раунда равна идентификатору данной (старшей) волны, то есть для седьмой волны - **cid\_h**, для шестой - **bid\_h**, для восьмой - **did\_h** и так далее.

```
generate id_h = cid_h, before(redid_h)
label variable id_w "номер волны"
label variable id_h "идентификатор домохозяйства в данной волне"
```

Посмотрите на листе данных, или сделайте описательные характеристики этих новых переменных.

```
sum id_w id_h
```

Сохраните файл.

```
save "C:\RLMS_work\seminar_4\data\hh07_ef.dta", replace
```

17.3. Переименование переменных 8й волны. Откроем файл семейных данных за 1998 год.

```
use "C:\RLMS_work\seminar_4\data\r08hall41_ef.dta", lear
```

Сохраним этот файл под новым именем

```
save "C:\RLMS_work\seminar_4\data\hh08_ef.dta"
```

Переименуем (код на следующей странице)

```

rename (dredid_h d_origsm d_hhwgt da3 da3_3 da4_1 da4_2 da5_1 da5_2 da8 d_nfm de2 de3
de4 de5 de6_1 de6_2 de7_1a de7_1b de7_2a de7_2b de7_3a de7_3b de7_4a de7_4b de7_5a
de7_5b de7_6a de7_6b de7_7a de7_7b de7_8a de7_8b de8_1a de8_1b de8_2a de8_2b de8_3a
de8_3b de9_1a de9_1b de9_2a de9_2b de9_3a de9_3b de9_4a de9_4b de9_4_1a de9_4_1b
de9_5a de9_5b de9_6a de9_6b de9_7a de9_7b de10 de11 de12 de12_1 de13_1a de13_1b de13_2a
de13_2b de13_21a de13_21b de13_3a de13_3b de13_31a de13_31b de13_4a de13_4b de13_5a
de13_5b de13_6a de13_6b de13_7a de13_7b de13_8a de13_8b de14 de15 de16 de17 de18
de19_1a de19_1b de19_2a de19_2b de19_3a de19_3b de19_4a de19_4b de19_5a de19_5b de20
de21_1a de21_1b de21_1c de21_2a de21_2b de21_2c de21_3a de21_3b de21_3c de21_4a
de21_4b de21_4c de21_5a de21_5b de21_5c de21_6a de21_6b de21_6c de21_7a de21_7b
de21_7c de21_8a de21_8b de21_8c de21_9a de21_9b de21_9c de21_10a de21_10b de21_10c
de21_11a de21_11b de21_11c de21_12a de21_12b de21_12c de21_13a de21_13b de21_13c
de21_14a de21_14c de22_1a de22_1b de22_2a de22_2b de22_3a de22_3b de22_4a de22_4b
de22_5a de22_5b de22_6a de22_6b de22_7a de22_7b de22_8a de22_8b de23 de24 df3 df4 df5
df6 df7 df8 df9_1a df9_1b df9_2a df9_2b df9_3a df9_3b df9_4a df9_4b df9_5a df9_5b df9_6a
df9_6b df9_7a df9_7b df9_8a df9_8b df9_9a df9_9b df9_10a df9_10b df10 df11 df12_1a df12_1b
df12_2a df12_2b df12_3a df12_3b df12_4a df12_4b df12_5a df12_5b df12_6aa df12_6ab
df12_6ba df12_6bb df12_7a df12_7b df12_8a df12_8b df12_9a df12_9b df1210aa df1210ab
df1210ba df1210bb df13_1a df13_1b df13_2a df13_2b df13_3a df13_3b df14 df14_1 df14_2
df14_3 df15) (redid_h origsm hhwgt a3 a3_3 a4_1 a4_2 a5_1 a5_2 a8 nfm e2 e3 e4 e5
e6_1 e6_2 e7_1a e7_1b e7_2a e7_2b e7_3a e7_3b e7_4a e7_4b e7_5a e7_5b e7_6a e7_6b
e7_7a e7_7b e7_8a e7_8b e8_1a e8_1b e8_2a e8_2b e8_3a e8_3b e9_1a e9_1b e9_2a e9_2b
e9_3a e9_3b e9_4a e9_4b e9_4_1a e9_4_1b e9_5a e9_5b e9_6a e9_6b e9_7a e9_7b e10
e11 e12 e12_1 e13_1a e13_1b e13_2a e13_2b e13_21a e13_21b e13_3a e13_3b e13_31a
e13_31b e13_4a e13_4b e13_5a e13_5b e13_6a e13_6b e13_7a e13_7b e13_8a e13_8b e14
e15 e16 e17 e18 e19_1a e19_1b e19_2a e19_2b e19_3a e19_3b e19_4a e19_4b e19_5a
e19_5b e20 e21_1a e21_1b e21_1c e21_2a e21_2b e21_2c e21_3a e21_3b e21_3c e21_4a
e21_4b e21_4c e21_5a e21_5b e21_5c e21_6a e21_6b e21_6c e21_7a e21_7b e21_7c e21_8a
e21_8b e21_8c e21_9a e21_9b e21_9c e21_10a e21_10b e21_10c e21_11a e21_11b e21_11c
e21_12a e21_12b e21_12c e21_13a e21_13b e21_13c e21_14a e21_14c e22_1a e22_1b e22_2a
e22_2b e22_3a e22_3b e22_4a e22_4b e22_5a e22_5b e22_6a e22_6b e22_7a e22_7b e22_8a
e22_8b e23 e24 f3 f4 f5 f6 f7 f8 f9_1a f9_1b f9_2a f9_2b f9_3a f9_3b f9_4a f9_4b f9_5a
f9_5b f9_6a f9_6b f9_7a f9_7b f9_8a f9_8b f9_9a f9_9b f9_10a f9_10b f10 f11 f12_1a
f12_1b f12_2a f12_2b f12_3a f12_3b f12_4a f12_4b f12_5a f12_5b f12_6aa f12_6ab f12_6ba
f12_6bb f12_7a f12_7b f12_8a f12_8b f12_9a f12_9b f1210aa f1210ab f1210ba f1210bb
f13_1a f13_1b f13_2a f13_2b f13_3a f13_3b f14 f14_1 f14_2 f14_3 f15)

```

Создадим вначале нашего файла две новые переменные, с такими же именами, как в лонгитюдных файлах с сайта. Первая – идентификатор волны (номер раунда).

```
generate id_w = 8, before(redid_h)
```

Вторая переменная – идентификатор семьи, которая может использоваться только для склейки внутри каждого раунда (но ни в коем случае для переменных с лагом). Эта переменная для любого раунда равна идентификатору данной (старшей) волны, то есть для седьмой волны - **cid\_h**, для шестой - **bid\_h**, для восьмой - **did\_h** и так далее.

```
generate id_h = did_h, before(redid_h)
label variable id_w "номер волны"
label variable id_h "идентификатор домохозяйства в данной волне"
```

Посмотрите на листе данных, или сделайте описательные характеристики этих новых переменных.

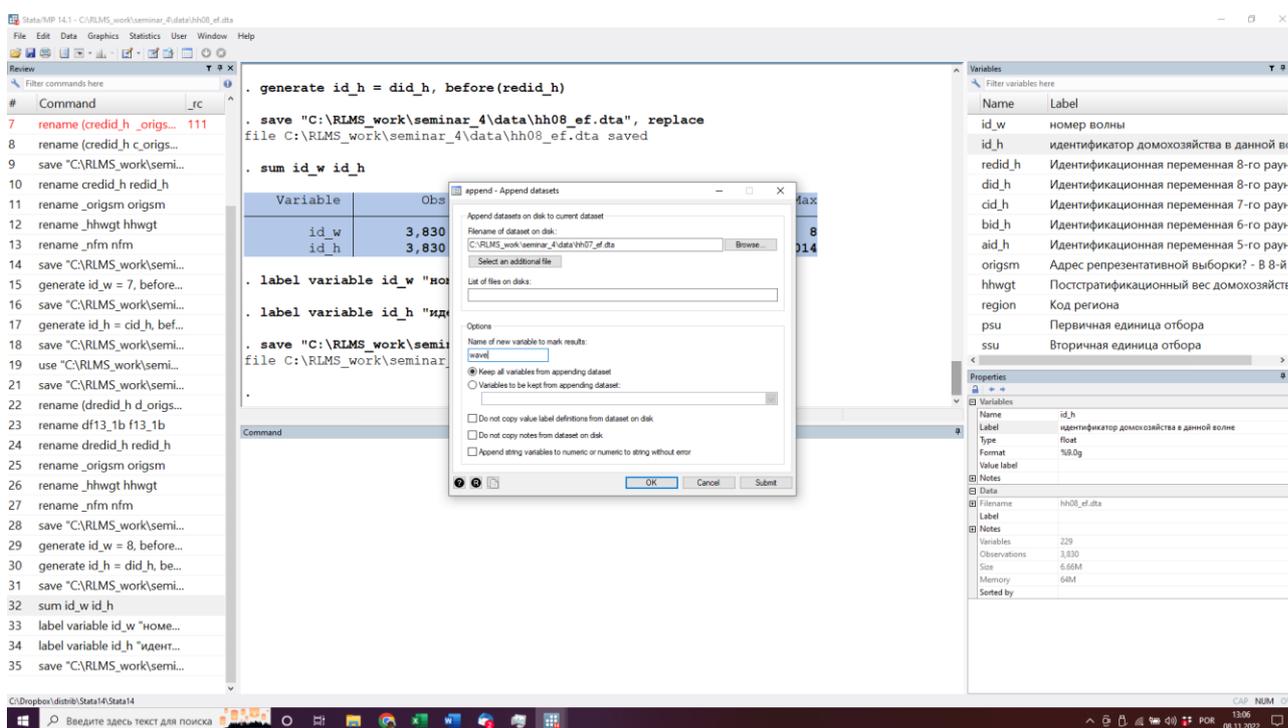
**sum id\_w id\_h**

Variable	Obs	Mean	Std. Dev.	Min	Max
id_w	3,830	8	0	8	8
id_h	3,830	830715.6	422624.6	10101	1600014

Сохраните файл.

**save "C:\RLMS\_work\seminar\_4\data\hh08\_ef.dta", replace**

17.4. \*Теперь приклеим данные седьмой волны к данным восьмой волны. Создадим на всякий случай переменную-маркер **wave**.



**append using "C:\RLMS\_work\seminar\_4\data\hh07\_ef.dta", generate(wave)**

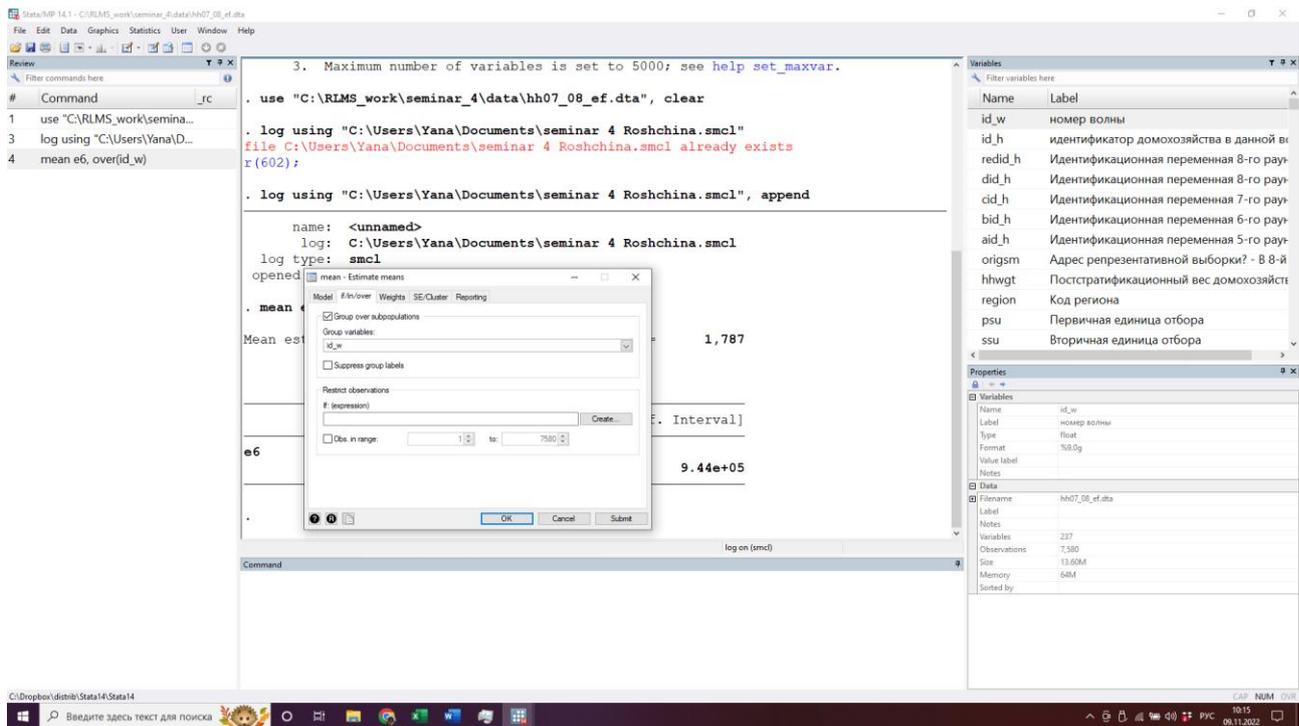
После переменной **wave** в списке переменных появились те переменные, которых не было в файле восьмой волны; проверим также результат нашей склейки, посмотрев распределение переменной **id\_w**.

**tab id\_w**

номер волны	Freq.	Percent	Cum.
7	3,750	49.47	49.47
8	3,830	50.53	100.00
Total	7,580	100.00	

Мы видим, что у нас в самом деле теперь в файле данные двух волн.





Используем теперь такой код, позволяющий суммировать «число» с «миссингом». ОБЯЗАТЕЛЬНО использовать условие, так как нам нужно суммировать только для кейсов из восьмой волны. Так как команда **egen** создает новую переменную, а наша **e6** уже есть, создадим вспомогательную переменную **e6\_sum**, а потом изменим нужную нам **e6**, а вспомогательную удалим.  
И снова посмотрим средние по годам:

```
egen e6_sum= rowtotal(e6_1 e6_2) if id_w == 8, missing
replace e6 = e6_sum if id_w ==8
drop e6_sum
mean e6, over(id_w)
```

```
Mean estimation                               Number of obs   =           3,503

       7: id_w = 7
       8: id_w = 8
```

	Over	Mean	Std. Err.	[95% Conf. Interval]	
e6	7	8.81e+05	32296.347	8.17e+05	9.44e+05
	8	3.16e+07	1.14e+06	2.93e+07	3.38e+07

Теперь у нас есть расходы  $d \setminus x$  на одежду за 7й и 8й раунды.

17.6. Вместо переменных **f12\_6a** и **f12\_6b** (наличие и сумма процентов и дивидендов) в 8 раунде появилось 4 переменных - отдельно про наличие и сумму процентов **f12\_6aa** **f12\_6ab**, и отдельно про наличие и сумму дивидендов **f12\_6ba** **f12\_6bb**.

Перекодируем миссинги и посмотрим описательные характеристики по раундам

```
recode f12_6a f12_6b f12_6aa f12_6ab f12_6ba f12_6bb (99999997 / 99999999= .)
```

**by id\_w, sort : summarize f12\_6a f12\_6b f12\_6aa f12\_6ab f12\_6ba f12\_6bb**

-> id\_w = 7

Variable	Obs	Mean	Std. Dev.	Min	Max
f12_6a	3,730	1.983378	.1278674	1	2
f12_6b	58	814434.5	1670228	2500	9000000
f12_6aa	0				
f12_6ab	0				
f12_6ba	0				
f12_6bb	0				

-> id\_w = 8

Variable	Obs	Mean	Std. Dev.	Min	Max
f12_6a	0				
f12_6b	0				
f12_6aa	3,823	1.995292	.0684646	1	2
f12_6ab	18	793.2222	738.0646	68	2400
f12_6ba	3,823	1.994507	.0739211	1	2
f12_6bb	18	538.6111	493.4574	5	1700

**replace f12\_6a=f12\_6aa if id\_w == 8**

**replace f12\_6a=1 if (id\_w == 8 & f12\_6ba == 1)**

**egen f12\_6bsum= rowtotal(f12\_6ab f12\_6bb) if id\_w == 8, missing**

**replace f12\_6b = f12\_6bsum if id\_w ==8**

**drop f12\_6bsum**

\*Посмотрим новые распределения

**by id\_w, sort : summarize f12\_6a f12\_6b f12\_6aa f12\_6ab f12\_6ba f12\_6bb**

-> id\_w = 7

Variable	Obs	Mean	Std. Dev.	Min	Max
f12_6a	3,730	1.983378	.1278674	1	2
f12_6b	58	814434.5	1670228	2500	9000000
f12_6aa	0				
f12_6ab	0				
f12_6ba	0				
f12_6bb	0				

-> id\_w = 8

Variable	Obs	Mean	Std. Dev.	Min	Max
f12_6a	3,823	1.99006	.099215	1	2
f12_6b	35	684.9429	631.399	5	2400
f12_6aa	3,823	1.995292	.0684646	1	2
f12_6ab	18	793.2222	738.0646	68	2400
f12_6ba	3,823	1.994507	.0739211	1	2
f12_6bb	18	538.6111	493.4574	5	1700

У нас появились значения для переменных **f12\_6a f12\_6b** за 8ю волну.

### 17.7. Самостоятельное задание (в классе или дома).

1) Сделать преобразования для переменных f12\_10a f12\_10b f1210aa f1210ab f1210ba f1210bb аналогично тому, что мы сделали для переменных **f12\_6a f12\_6b f12\_6aa f12\_6ab f12\_6ba f12\_6bb**. На основе команды **by id\_w, sort : summarize** посмотреть их распределение ДО и ПОСЛЕ ваших преобразований.

Сохранить файл.

2) Аналогично тому, как мы приклеили данные 7го раунда к 8му, сделать преобразования в файле 6го раунда (r06hall41\_ef.dta) и также к 8му раунда приклеить данные 6го раунда (но не наоборот). Посмотреть распределение переменной **id\_w**.

3) Сохраните ваш код. Закройте и сохраните файл аутпута вместе с выполненным заданием (отчет за первую половину семинара).

### **18. Работа с лонгитюдным файлом домохозяйства. Душевой доход. Квintильная группа. Мобильность домохозяйств по доходам.**

18.1. Откройте новый лог-файл (аутпута) и введите команду-комментарий с вашей фамилией

```
log using "C:\RLMS_work\seminar_4\data\seminar4_part2_Roshchina.smcl"
```

\* семинар 4 часть 2 Рощина

Откройте рабочий файл за 5-16 волны (часть переменных удалена).

```
use "C:\RLMS_work\seminar_4\data\hh_5_16.dta", clear
```

Сохраните под новым именем

```
save "C:\RLMS_work\seminar_4\data\hh_5_16_S4.dta"
```

18.2. Для расчета эквивалентного дешевого дохода нам надо знать количество взрослых и детей в семье.

\*Посчитаем количество взрослых 18 лет и старше.

\* Создадим переменную «год»

```
recode id_w (5=1994) (6=1995) (7=1996) (8=1998) (9=2000) (10=2001)(11=2002) (12=2003) (13=2004) (14=2005) (15=2006) (16=2007) (17=2008)(18=2009) (19=2010) (20=2011) (21=2012) (22=2013) (23=2014) (24=2015), into (YEAR)
```

\*В файле домохозяйства у каждого члена семьи есть информация о годе рождения. Сделаем миссинги в этих переменных.

```
recode b1_5 b2_5 b3_5 b4_5 b5_5 b6_5 b7_5 b8_5 b9_5 b10_5 b11_5 b12_5 b13_5 b14_5 b15_5 b16_5 b17_5 b18_5 b19_5 b20_5 b21_5 b22_5 b23_5 b24_5 (99999996 / 99999999 = .)
```

\*Рассчитаем возраст каждого члена семьи, вычтя год рождения из года опроса

```
gen age_1 = YEAR - b1_5
```

```
gen age_2 = YEAR - b2_5
```

```
gen age_3 = YEAR - b3_5
```

```
gen age_4 = YEAR - b4_5
```

```
gen age_5 = YEAR - b5_5
```

```
gen age_6 = YEAR - b6_5
```

gen age\_7 = YEAR - b7\_5  
 gen age\_8 = YEAR - b8\_5  
 gen age\_9 = YEAR - b9\_5  
 gen age\_10 = YEAR - b10\_5  
 gen age\_11 = YEAR - b11\_5  
 gen age\_12 = YEAR - b12\_5  
 gen age\_13 = YEAR - b13\_5  
 gen age\_14 = YEAR - b14\_5  
 gen age\_15 = YEAR - b15\_5  
 gen age\_16 = YEAR - b16\_5  
 gen age\_17 = YEAR - b17\_5  
 gen age\_18 = YEAR - b18\_5  
 gen age\_19 = YEAR - b19\_5  
 gen age\_20 = YEAR - b20\_5  
 gen age\_21 = YEAR - b21\_5  
 gen age\_22 = YEAR - b22\_5  
 gen age\_23 = YEAR - b23\_5  
 gen age\_24 = YEAR - b24\_5

sum age\_1 age\_2 age\_3 age\_4 age\_4 age\_5 age\_7 age\_6 age\_8 age\_9 age\_10 age\_11 age\_12  
 age\_13 age\_14 age\_15 age\_16 age\_17 age\_18 age\_19 age\_20 age\_21 age\_22 age\_23 age\_24

Variable	Obs	Mean	Std. Dev.	Min	Max
age_1	44,355	51.44611	16.10507	14	100
age_2	36,289	45.43619	16.83301	0	103
age_3	26,369	24.57257	17.72097	0	102
age_4	18,432	22.05621	17.13934	-1	97
age_4	18,432	22.05621	17.13934	-1	97
age_5	10,256	23.3947	20.11985	0	97
age_7	3,357	21.45696	20.56041	0	95
age_6	5,964	21.91247	19.95321	0	95
age_8	1,993	21.4285	21.04327	0	88
age_9	1,140	20.61053	20.10567	0	88
age_10	711	18.30942	17.738	0	95
age_11	371	20.58221	18.64492	0	84
age_12	262	15.69466	15.79388	0	75
age_13	154	16.09091	17.73972	0	69
age_14	94	26.87234	21.2508	0	72
age_15	66	37.54545	27.366	0	86
age_16	35	17.42857	15.56823	0	74
age_17	16	20.5	16.87207	0	53
age_18	11	18.54545	12.63617	1	43
age_19	15	8	8.823669	1	30
age_20	3	32	27.71281	0	48
age_21	2	33.5	.7071068	33	34
age_22	2	11.5	.7071068	11	12
age_23	1	27	.	27	27
age_24	1	0	.	0	0

Есть 2 случая, где возраст получился меньше 0: это происходит потому, что иногда опрос проводился после Нового года, и ребенок мог родиться в январе.

\*Перекодируем такие значения в 0 (ребенок младше года).

```

recode age_1 age_2 age_3 age_4 age_4 age_5 age_7 age_6 age_8 age_9 age_10 age_11 age_12
age_13 age_14 age_15 age_16 age_17 age_18 age_19 age_20 age_21 age_22 age_23 age_24 (-1 =
0)

```

\*Посчитаем количество взрослых от 18 лет и старше («от 18 до 200»)

```

egen num_adult = anycount( age_1 age_2 age_3 age_4 age_5 age_6 age_7 age_8 age_9 age_10
age_11 age_12 age_13 age_14 age_15 age_16 age_17 age_18 age_19 age_20 age_21 age_22
age_23 age_24 ), values(18 / 200)

```

```
sum num_adult
```

Variable	Obs	Mean	Std. Dev.	Min	Max
num_adult	53,511	2.165798	.9849649	0	12

Есть 42 семьи, где нет людей 18 лет и старше. Перекодируем значение 0 в 1.

```
recode num_adult (0 =1)
```

Рассчитаем количество детей до 18 лет

```

gen num_children17 = nfm - num_adult
label variable num_children17 "Количество детей 0-17 лет"
label variable num_adult "Количество взрослых 18+ (или = 1 если все моложе 18)"

```

18.2. Рассчитаем душевой доход и дефлированный душевой доход (к уровню 16й волны), как мы делали ранее. Перед этим удалим миссинги из переменной суммарного дохода домохозяйства.

```

recode f14 (99999997 / 99999999 = .)
gen INCOME_PC_N = f14 / nfm if nfm>0
label variable INCOME_PC_N "Душевой доход номинальный"

```

```

gen deflat_16 = 1 if id_w == 16
replace deflat_16 = 1.119 if id_w == 15
replace deflat_16 = 1.119 * 1.09 if id_w == 14
replace deflat_16 = 1.119 * 1.09*1.09 if id_w == 13
replace deflat_16 = 1.119 * 1.09*1.09*1.117 if id_w == 12
replace deflat_16 = 1.119 * 1.09*1.09*1.117*1.12 if id_w == 11
replace deflat_16 = 1.119 * 1.09*1.09*1.117*1.12*1.151 if id_w == 10
replace deflat_16 = 1.119 * 1.09*1.09*1.117*1.12*1.151*1.186 if id_w == 9
replace deflat_16 = 1.119 * 1.09*1.09*1.117*1.12*1.151*1.186*1.202*1.365 if id_w == 8
replace deflat_16 = 1.119 * 1.09*1.09*1.117*1.12*1.151*1.186*1.202*1.365*1.844*1.11/1000 if
id_w == 7
replace deflat_16 = 1.119 * 1.09 * 1.109 * 1.117 * 1.12 * 1.151 * 1.186 * 1.202 * 1.365 * 1.844 *
1.11 * 1.218 / 1000 if id_w == 6
replace deflat_16 = 1.119 * 1.09 * 1.109 * 1.117 * 1.12 * 1.151 * 1.186 * 1.202 * 1.365 * 1.844 *
1.11 * 1.218 * 2.3/1000 if id_w == 5
label variable deflat_16 "дефлятор приведения цен к 16 волне по среднерос. ИПЦ
(умножать на него)"

```

```

gen INCOME_PC = INCOME_PC_N* deflat_16
label variable INCOME_PC "Душевой доход дефлированный"

```

18.3. Рассчитаем эквивалентный душевой доход по формуле:

душ.доход = сум.доход / (1 + 0,7 \* (кол.взрослых - 1) + 0,5 \* кол.детей)

```
gen INCOME_PC_eq = f14 * deflat_16 / (1+ 0.7*( num_adult - 1)+ 0.5* num_children17)
label variable INCOME_PC_eq "Душевой доход дефлированный с учетом шкалы эквивалентности (f14)"
```

\*Рассчитаем средний реальный душевой доход обычный и с учетом шкалы эквивалентности по годам с весом по домохозяйствам (для репрезентативности)

```
tabstat INCOME_PC INCOME_PC_eq [aw = hhwgt], stat (mean) by (id_w )
```

```
Summary statistics: mean
by categories of: id_w (Волна (год проведения исследования))
```

id_w	INCOME~C	INCOME~q
1994 год	4084.108	5133.083
1995 год	3307.925	4126.171
1996 год	3321.976	4131.968
1998 год	2811.077	3749.988
2000 год	2899.35	3602.422
2001 год	3681.464	4579.564
2002 год	4001.801	4954.802
2003 год	4414.116	5442.209
2004 год	4800.708	5923.609
2005 год	5700.155	7024.438
2006 год	6635.459	8162.958
2007 год	6828.915	8367.97
Total	4404.68	5470.944

Сравните результаты.

18.4. Для расчета квинтильных групп (20% групп по доходу) установим дополнительные функции к команде.

```
ssc install egenmore
```

\*При необходимости для нее можно распечатать help

```
*ssc type egenmore.sthlp
```

Нам нужна функция

```
xtile(varname) [ , percentiles(numlist) nquantiles(#) weights(varname) altdef
by(byvarlist) ] categorizes varname by specific percentiles. The function
works like xtile. By default varname is dichotomized at the median.
percentiles() requests percentiles corresponding to numlist: for example,
p(25(25)75) is used to create a variable according to quartiles.
Alternatively you also may have specified n(4): to create a variable
according to quartiles. weights() requests weighted calculation of
percentiles. altdef uses an alternative formula for calculating percentiles.
See xtile. by() requests calculation by groups. You may also use the by:
construct. (Stata 8.2 required.)
```

```
. egen mpg4 = xtile(mpg) , by(foreign) p(25(25)75)
. egen mpg10 = xtile(mpg) , by(foreign) nq(10)
```

Она приписывает каждому кейсу число от 1 до 5 (номер группы) согласно уровню душевого дохода, так, чтобы получилось 5 групп по 20% (можно делать и 4, и 10 групп, тогда получатся

квартили и децили). Так как в домохозяйстве может быть больше одного человека, а доходные группы формируются по отношению к людям, а не домохозяйствам в целом, нужно сделать взвешивание по количеству человек в семье **weights(nfm)**. Такие группы нужно сформировать внутри каждого года, поэтому используем **by( YEAR )**.

```
egen NI_income =xtile( INCOME_PC_N ), n(5) by( YEAR ) weights(nfm)
label variable NI_income "'Квintильные группы по душевому доходу f14 (по индивидам) по годам"
```

\* отсортируем

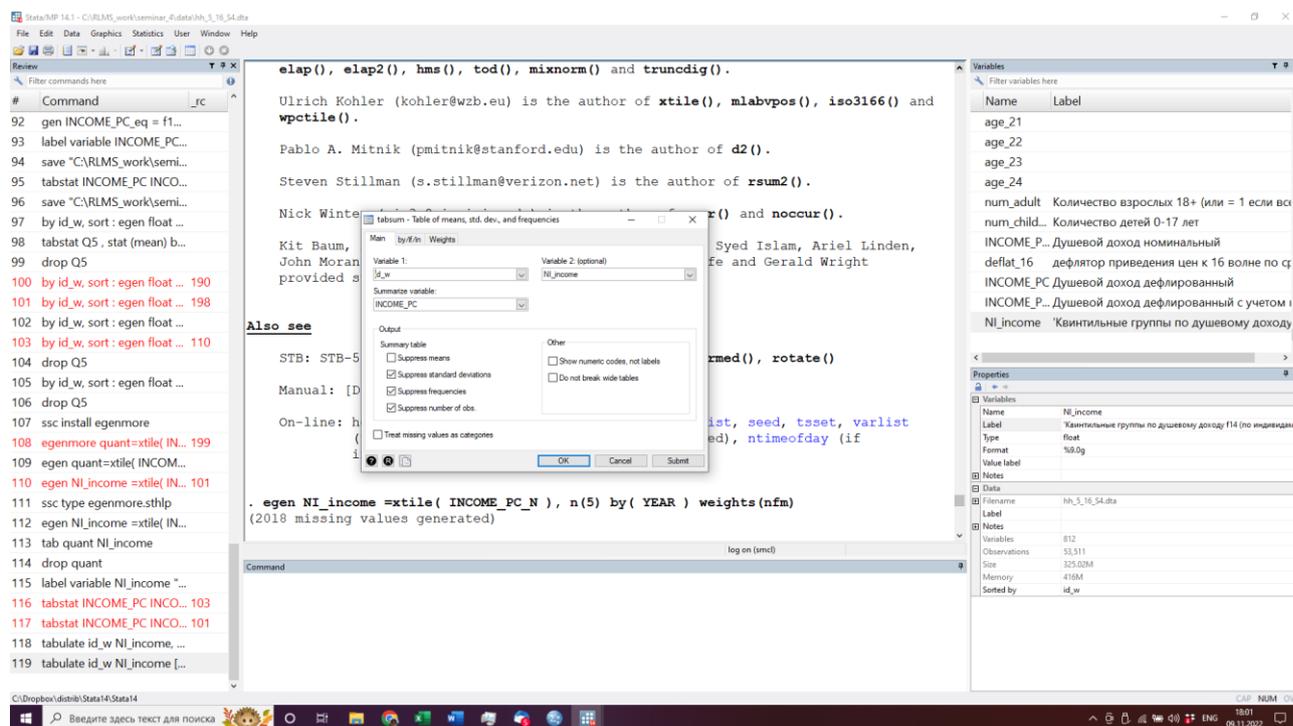
```
sort id_w INCOME_PC_N
```

И перейдите в таблицу данных, чтобы посмотреть результаты этой функции.

18.5. Посмотрим различия душевого дохода по годам и по квинтильным группам, с взвешиванием для репрезентативности.

Используем удобную команду из меню:

Summaries, tests and tables > Other tables > Table of mean, std.dev, and frequencies



```
tabulate id_w NI_income [aweight = hhwt], summarize(INCOME_PC) nostandard nofreq noobs
```

Means of Душевой доход дефлированный

Волна	1	2	3	4	5	Total
(год						
прове						
дения						
иссле						
дован						
ия)						

'Квintильные группы по душевому доходу f14 (по индивидам) по годам

1994 год		739.1	1730.8083	2548.567	3832.8787	10569.588		4084.1085
1995 год		470.27477	1403.1235	2112.1698	3149.1893	8631.4116		3307.9252
1996 год		173.82385	1181.6129	2118.9134	3315.1707	9208.4727		3321.9764
1998 год		202.99185	844.68869	1394.0688	2062.032	9074.2515		2811.0766
2000 год		495.34942	1315.5169	1974.4854	2927.3055	7472.7457		2899.3503
2001 год		770.76263	1764.2599	2610.5332	3849.3143	9489.3622		3681.4638
2002 год		875.5647	1960.3416	2889.9122	4244.7195	9807.2626		4001.8006
2003 год		996.69083	2248.602	3236.0658	4841.474	10729.643		4414.1163
2004 год		928.15901	2321.4614	3429.9056	5129.3132	11895.624		4800.7078
2005 год		1525.337	2990.8023	4231.4598	6122.2378	13347.891		5700.1549
2006 год		1756.9246	3342.8858	4797.5348	6905.0162	16075.976		6635.4591
2007 год		2136.8499	3734.8413	5290.1679	7551.7669	16041.691		6828.9153
-----+-----								
Total		949.38085	2136.4003	3032.238	4473.6904	11068.846		4404.68

### 18.6. Мобильность семей по доходам.

Рассмотрим позиционную мобильность, то есть как изменялось положение семьи в квинтильной группе в каждой волне. Для этого нам нужно сделать специальный файл «номер квинтильной группы семьи в прошлом году».

\*Сделаем переменную «идентификатор семьи в прошлом году». Не забываем указать тип новой переменной «двойной точности»!!!

```
gen double id_h_1 = kid_h if id_w ==16
replace id_h_1 = jid_h if id_w ==15
replace id_h_1 = iid_h if id_w ==14
replace id_h_1 = hid_h if id_w ==13
replace id_h_1 = gid_h if id_w ==12
replace id_h_1 = fid_h if id_w ==11
replace id_h_1 = eid_h if id_w ==10
replace id_h_1 = did_h if id_w ==9
replace id_h_1 = cid_h if id_w ==8
replace id_h_1 = bid_h if id_w ==7
replace id_h_1 = aid_h if id_w ==6
label variable id_h_1 "идентификатор семьи для склейки с прошлым годом"
```

Сохраните файл!!!

```
save "C:\RLMS_work\seminar_4\data\hh_5_16_S4.dta", replace
```

\*Сохраним файл под новым названием (это будут данные за прошлый год) и оставим в нем только три переменные

```
save "C:\RLMS_work\seminar_4\data\hh_5_16_S4_T1.dta"
keep id_w id_h NI_income
save "C:\RLMS_work\seminar_4\data\hh_5_16_S4_T1.dta"
```

\*Сделаем в этом новом маленьком файле преобразования. К номеру волны прибавим 1 (чтобы данные приклеились к будущему году), и переименуем переменные идентификатора и квинтильной группы.

```
replace id_w = id_w + 1
rename (id_h NI_income) (id_h_1 NI_income_t1)
label variable id_w "T-1 Волна (год проведения исследования)"
label variable id_h_1 "T-1 ИДЕНТИФИКАЦИОННАЯ ПЕРЕМЕННАЯ"
label variable NI_income_t1 "T-1 Квинтильные группы по душевому доходу f14 (по индивидам) по годам"
```

```
save "C:\RLMS_work\seminar_4\data\hh_5_16_S4_T1.dta"
```

\*Откроем снова наш рабочий семейный файл 5-16 волн

```
use "C:\RLMS_work\seminar_4\data\hh_5_16_S4.dta", clear
```

\*И приклеим к нему данные из маленького файла

```
merge m:1 id_w id_h_1 using "C:\RLMS_work\seminar_4\data\hh_5_16_S4_T1.dta"
```

```
(label id_w already defined)
```

Result	# of obs.	
not matched	22,480	
from master	10,608	( _merge==1)
from using	11,872	( _merge==2)
matched	42,903	( _merge==3)

Из более чем 53 тысяч домохозяйств есть данные о квинтильной группе прошлого года для 42903. У нас нет таких данных для 1994 года, а также для всех новых домохозяйств, или тех, которые не участвовали в прошлом году. Удалим «лишние» кейсы, приклеившиеся из маленького файла, и контрольную переменную.

```
drop if _merge ==2
drop _merge
```

\*Теперь мы можем построить матрицу мобильности по квинтильным группам.

```
by id_w, sort : tabulate NI_income_t1 NI_income, nofreq row
```

```
-----
-> id_w = 1994 год
no observations
-----
```

```
-----
-> id_w = 1995 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----+-----
    1 |    46.64    24.95    13.05     9.40     5.95 | 100.00
    2 |    17.91    38.47    23.20    11.89     8.52 | 100.00
    3 |    11.94    21.40    38.57    18.20     9.90 | 100.00
    4 |     6.48     9.72    22.99    36.11    24.69 | 100.00
    5 |     4.33     6.99    13.31    24.46    50.92 | 100.00
-----+-----+-----+-----+-----+-----
 Total |    16.41    20.52    22.94    20.27    19.85 | 100.00
-----
-> id_w = 1996 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----
    1 |    44.03    25.93    15.30     9.70     5.04 | 100.00
    2 |    21.71    25.99    31.38    12.52     8.40 | 100.00
    3 |    18.02    17.60    32.26    21.93    10.20 | 100.00
    4 |     8.99    10.29    20.92    34.97    24.84 | 100.00
    5 |     6.14     8.29    10.95    22.72    51.91 | 100.00
-----+-----+-----+-----+-----+-----
 Total |    19.17    17.50    22.76    20.63    19.95 | 100.00
-----
```

```

-> id_w = 1998 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----+-----
    1 |    36.48    25.04    20.56    12.60     5.31 | 100.00
    2 |    25.04    26.27    23.47    16.81     8.41 | 100.00
    3 |    13.30    17.40    33.04    24.56    11.70 | 100.00
    4 |     7.47    11.20    22.08    36.53    22.73 | 100.00
    5 |     5.42     8.81    11.19    24.07    50.51 | 100.00
-----+-----+-----+-----+-----+-----
  Total |    17.36    17.66    22.39    23.07    19.52 | 100.00
-> id_w = 2000 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----
    1 |    40.30    25.04    17.91     8.29     8.46 | 100.00
    2 |    24.71    32.10    21.85    11.76     9.58 | 100.00
    3 |     8.27    27.07    38.40    16.00    10.27 | 100.00
    4 |     5.99    12.11    31.29    28.98    21.63 | 100.00
    5 |     5.39     7.19    12.75    24.18    50.49 | 100.00
-----+-----+-----+-----+-----+-----
  Total |    16.05    20.58    25.31    18.24    19.82 | 100.00
-> id_w = 2001 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----
    1 |    54.90    21.15    10.66     6.12     7.17 | 100.00
    2 |    23.36    41.68    17.76    11.47     5.73 | 100.00
    3 |     7.32    20.61    44.48    20.05     7.55 | 100.00
    4 |     5.53    11.67    23.20    40.71    18.89 | 100.00
    5 |     4.02     8.18    13.39    27.23    47.17 | 100.00
-----+-----+-----+-----+-----+-----
  Total |    17.41    20.95    23.56    21.24    16.84 | 100.00
-> id_w = 2002 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----
    1 |    52.83    23.36    11.31     7.74     4.76 | 100.00
    2 |    19.90    41.56    20.28    13.22     5.04 | 100.00
    3 |     7.02    19.48    42.92    19.59    10.99 | 100.00
    4 |     5.23     9.99    22.07    36.00    26.71 | 100.00
    5 |     5.64     7.08    11.13    20.81    55.35 | 100.00
-----+-----+-----+-----+-----+-----
  Total |    16.89    20.35    22.63    20.09    20.04 | 100.00
-> id_w = 2003 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----
    1 |    55.43    20.26    10.71     8.10     5.50 | 100.00
    2 |    20.22    44.77    17.81    10.71     6.50 | 100.00
    3 |     7.60    22.58    42.95    17.29     9.58 | 100.00
    4 |     5.26    11.86    21.15    40.22    21.52 | 100.00
    5 |     4.49     3.72    10.77    23.72    57.31 | 100.00
-----+-----+-----+-----+-----+-----
  Total |    17.33    20.93    21.57    20.26    19.91 | 100.00
-> id_w = 2004 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----
    1 |    47.88    28.32    12.86     6.98     3.97 | 100.00
    2 |    16.04    47.54    21.66     9.84     4.92 | 100.00
    3 |     8.36    16.15    44.79    21.31     9.39 | 100.00
    4 |     7.38     8.10    20.31    39.06    25.15 | 100.00
    5 |     6.50     6.24     8.79    21.66    56.82 | 100.00
-----+-----+-----+-----+-----+-----
  Total |    16.51    21.38    22.29    20.00    19.83 | 100.00
-> id_w = 2005 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----
    1 |    54.92    19.85    12.92     5.54     6.77 | 100.00
    2 |    23.99    43.11    19.81     9.15     3.94 | 100.00
    3 |     7.45    25.36    38.15    19.58     9.45 | 100.00
    4 |     4.67     8.97    22.48    39.68    24.20 | 100.00
    5 |     3.41     5.61    11.59    23.54    55.85 | 100.00
-----+-----+-----+-----+-----+-----
  Total |    17.23    20.96    21.65    19.95    20.22 | 100.00
-> id_w = 2006 год
  годам |          1          2          3          4          5 | Total
-----+-----+-----+-----+-----+-----

```

1	60.97	21.03	9.08	5.60	3.33	100.00
2	18.52	48.15	18.89	9.63	4.81	100.00
3	7.81	23.30	39.41	19.83	9.67	100.00
4	4.06	10.35	23.07	40.24	22.28	100.00
5	2.60	5.75	9.17	25.17	57.32	100.00
Total	17.66	22.22	20.52	20.31	19.30	100.00
-> id_w = 2007 год						
годом	1	2	3	4	5	Total
1	62.47	19.34	10.18	4.96	3.05	100.00
2	22.98	44.05	20.97	8.37	3.63	100.00
3	7.63	25.95	38.60	19.74	8.07	100.00
4	4.91	7.53	21.58	42.69	23.29	100.00
5	3.03	4.67	8.05	22.17	62.08	100.00
Total	19.38	21.07	20.33	19.58	19.65	100.00