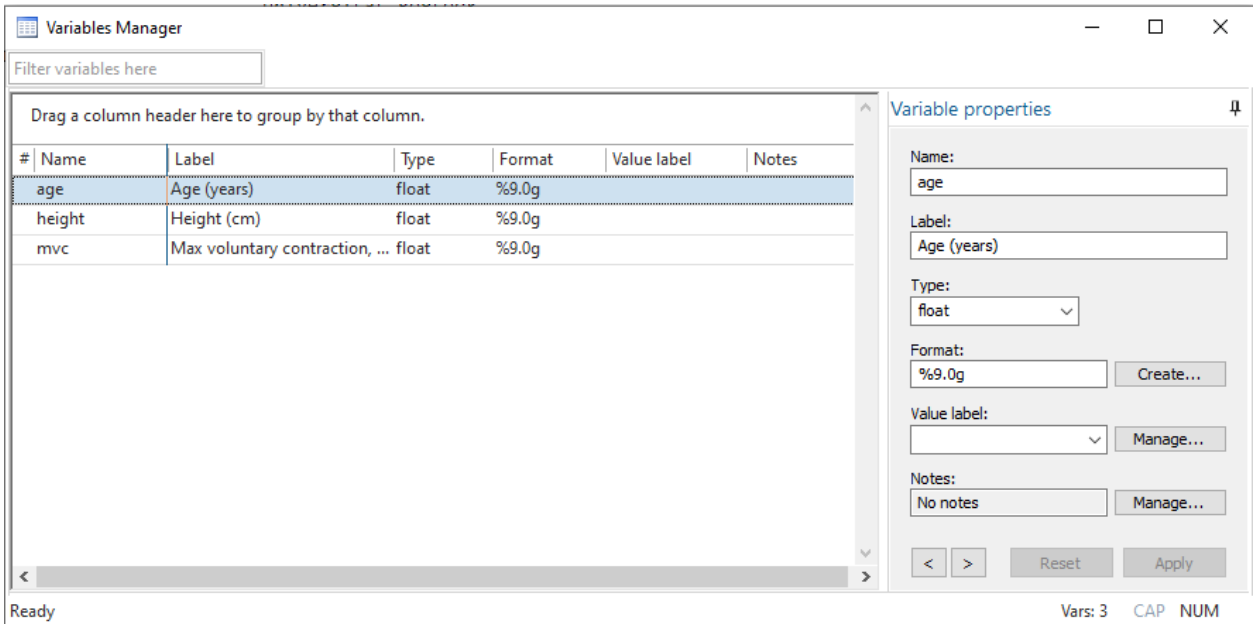
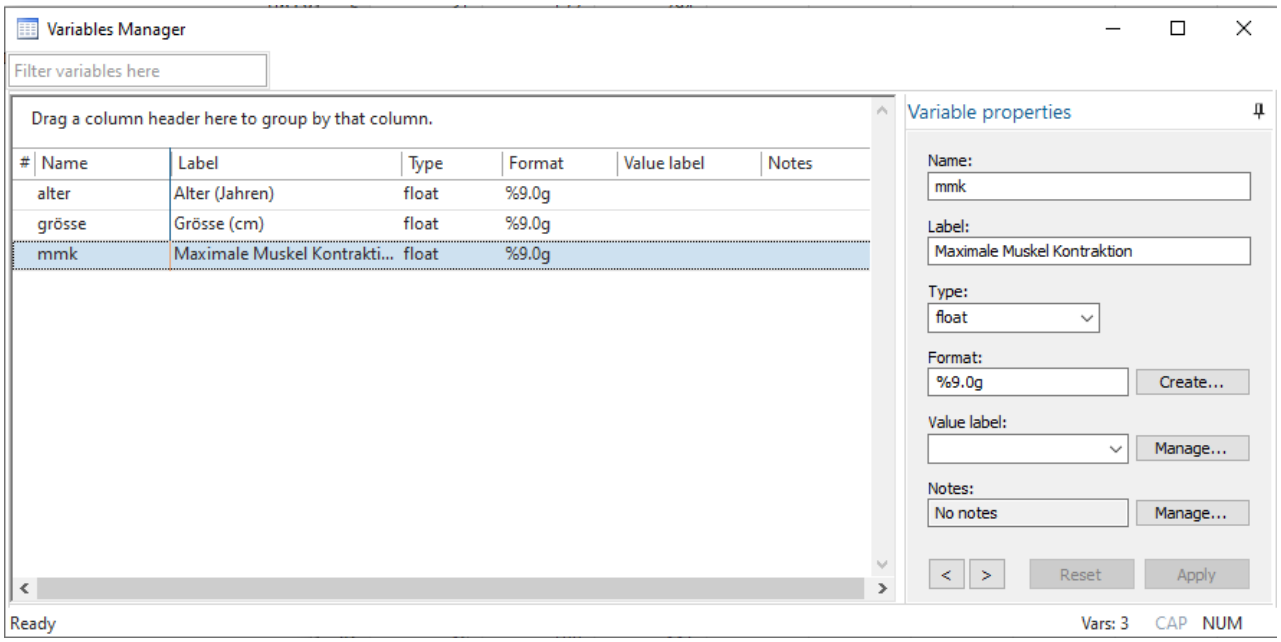
3. Datenmanagement

Hat man die Rohdaten im Programm, müssen sie für die weitere Analyse aufbereitet werden. Hierzu gehören z.B. die Identifikation fehlender Werte, Vergabe neuer Variablennamen, Klassifizierungen, Berechnungen von Scores usw.

Zur Veränderung von Variablennamen, Label, Typ oder Format kann man den *Variables Manager* nutzen. Mit ***use D:\Stata\muscle.dta*** z.B. lesen wir die Daten ein und starten den *Variables Manager.* Achten Sie auf die korrekte Pfadeingabe.



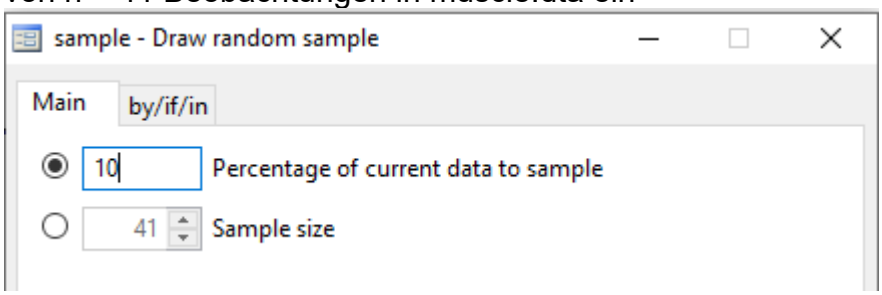
Jede Zeile im *Variables Manager* entspricht einer Variablen (Var). Es sollen einige Änderungen der Variablennamen vorgenommen werden. Dafür klicken wir in der rechten Spalte *Variable properties* in das Namensfeld und ändern ***age*** --> ***alter***. Ebenso ***height*** --> ***grösse*** und ***mvc*** --> ***mmk***. Auch die Labels werden entsprechend angepasst. Alle Var sind numerisch. Sie werden standardmäßig als Typ float im Format %9.0g gespeichert. Nach der Änderung erfolgt Speichern unter neuem Namen (z.B. muscle_2.dta).



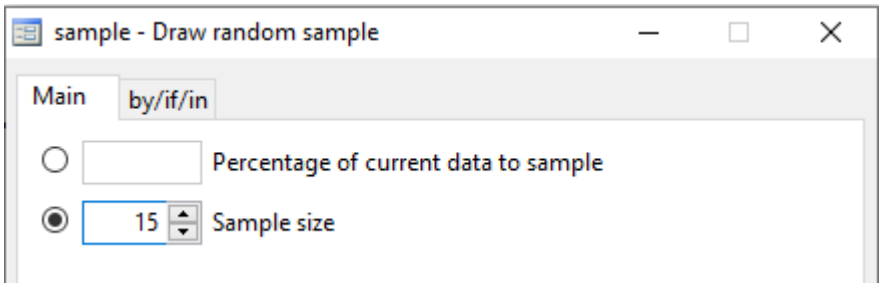
Die Änderung der Variablennamen kann auch über die Kommandozeile erfolgen, z.B.
rename age alter und die Änderung des Labels mit
label variable alter "Alter (Jahren)"
Die Einfügung einer neuen Variable: ***grösse_m = Grösse in Meter*** erfolgt mit dem Kommando ***generate grösse_m = grösse / 100*** oder über das Menü *Data >> Create or change date >> Create new variable*

Möchte man aus den Daten muscle.dta die ersten 10 Datensätze auflisten, so lautet das Kommando:
list in 1/10 oder man verwendet das Menü: *Data >> Describe data >> List data >> by/if/in >> Use a range of observations*

Möchte man eine Zufallsstichprobe von 10% der Daten oder eine vom Umfang n = 15, dann erhält man diese mit ***sample 10*** oder ***sample 15, count*** Nur diese zufällige Auswahl bleibt erhalten, daher ist das Speichern unter neuem Namen wichtig, da sonst die Originaldatei überschrieben wird.
Analog über das Menü *Statistics >> Resampling >> Draw random sample*
erhält man folgende Dialogbox. In der ersten Zeile trägt man den gewünschten Anteil 10% von n = 41 Beobachtungen in muscle.dta ein



oder man möchte 15 Datensätze zufällig auswählen.



Möchte man die Werte einer Var aufsteigend sortieren, dann z.B.
sort age oder in absteigender Reihenfolge
gsort -age wichtig ist hier das Minuszeichen vor dem Variablennamen
Das gleiche Ergebnis folgt über das Menü mit *Data >> Sort*

Einschränkungen mit **in** und **if** (in and if qualifiers):
Viele Kommandos können mit ***in*** und ***if*** auf bestimmte Untergruppen beschränkt werden.

Zur Entfernung fehlender Werte (missing values) in einer Variablen (var), nimmt man das Kommando ***drop if var == .*** Beachten Sie das doppelte Gleichheitszeichen. Es wird der gesamte Fall entfernt, wenn ein (oder mehrere) Merkmal(e) fehlt.

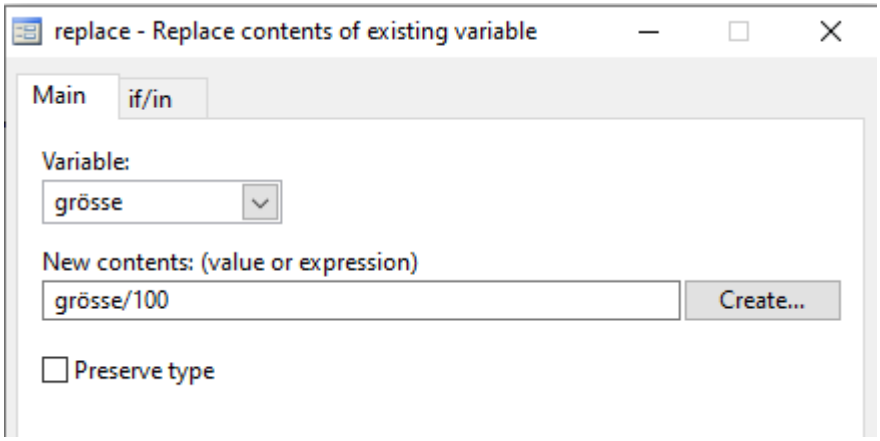
Die Entfernung von Daten (Beobachtungen) gelingt mit ***drop in*** # / # . Möchte man z.B. die 3. Zeile des Datensatzes entfernen, dann mit ***drop in*** 3 . Soll der 3. bis 6. Datensatz ge-

löscht werden, dann mit ***drop in*** 3/6 .

Für solche und andere Operationen sind folgende Operatoren wichtig:

==	ist gleich	|	+	Addition
!=	ist nicht gleich (auch ~=)	|	-	Subtraktion
>	ist größer als	|	*	Multiplikation
<	ist kleiner als	|	/	Division
>=	ist größer oder gleich als	|	^	Potenzieren
<=	ist kleiner oder gleich	|		

Möchte man keine neue Var erstellen und stattdessen eine alte Var verändern, so nimmt man ***replace***, z.B. zum Ersetzen der Var ***grösse*** in ***grösse/100***
replace grösse = grösse / 100 **Vorsicht beim Speichern !**
Beim Speichern unter dem gleichen Dateinamen wird die Var überschrieben.
Übersicht behält man im Menü: *Data >> Create or change data >> Change contents of variable* mit der Dialogbox und den entsprechenden Einträgen.



Viele statistische Operationen lassen sich nicht mit Textvariablen (Strings) durchführen. Daher möchte man diese Var mit einem Zahlen-Label versehen/ergänzen. Wir laden die Datei geschl_string.dta und fügen zur Text-Var ***gesch*** die Var ***geschnr*** hinzu, die zusätzlich zum Geschlecht eine Zahl als Label enthält.
encode gesch, generate(geschnr)
Die Var ***geschnr*** wird jetzt im Editor in blauer Schrift angezeigt. So erkennt man, daß eine Zahl (Zahlen-Label) zugeordnet ist. Standardmäßig werden in Stata folgende Farben im Editor angezeigt:

ROT reiner Text (String)
BLAU String mit Zahlen-Label
SCHWARZ reiner Zahlenwert

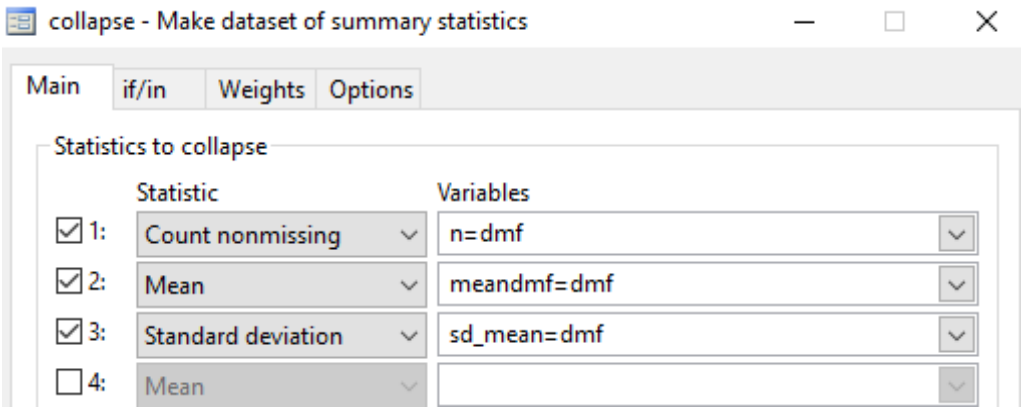
	gesch	geschnr
1	w	w
2	m	m
3	m	m
4	w	w

Über das Menü:
Data >> Create or change data >> Other variable-transformation commands >> Encode value labels from string variable...
erhält man eine Dialogbox für die entsprechenden Einträge.

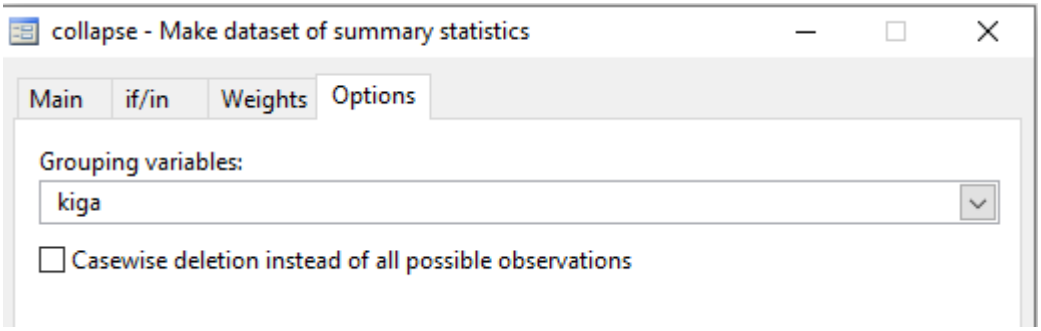
Sind Einzeldaten von verschiedenen Untergruppen in einem Datensatz zusammengefasst, z.B. Untersuchungen von Kindern aus verschiedenen Kindergärten, so kann man mit dem Kommando ***collapse*** einem neuen Datensatz erstellen, in dem Statistiken für jeden Kindergarten zusammengefasst (aggregiert) sind. Als Beispiel laden wir die Datei kiga_5.dta. In jeder Zeile des Editors stehen die Ergebnisse eines Kindes. Mit dem Kommando ***collapse (count) n=dmf (mean) meandmf=dmf (sd) sd_dmf=dmf , by(kiga)*** erhält man eine neue Datei, bei der jetzt in jeder Zeile die aggregierten Daten für die 5 Kindergärten stehen: Anzahl der Kinder pro Kindergarten, Mittelwert des dmf, Standardabweichung des dmf.

kiga	n	meandmf	sd_dmf
1	52	1.7115385	3.1269639
2	19	2.7368421	3.1595025
3	61	1.5409836	2.6049553
4	26	2.0769231	3.825421
5	43	2.2790698	4.2779923

Gleiches Ergebnis erhält man über das Menü:
Data >> Create or change data >> Other variable-transformation commands >> Make dataset of means, medians, etc. Es erscheint folgende Dialogbox, in die man Einträge vornimmt:



Weitere aggregierte Var sind möglich. Unter "Options" wird noch die Gruppierungsvariable eingetragen, hier "kiga".



Labels sind kurze Texte zur näheren

- Beschreibung eines Datensatzes ----Dataset labels
- Beschreibung von codierten Variablen ---- Variable labels
- Beschreibung von numerischen Daten mit Worten ---- Value labels

Als Beispiel laden wir die Datei geschl_karies.dta. Mit ***describe*** erhält man Informationen über die Datei:

```
Contains data from D:\Stata\geschl_karies.dta
  obs:             79
 vars:              2                          1 Aug 2021 16:30
 size:            158
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
geschl          byte    %8.0g
karies          byte    %8.0g
-------------------------------------------------------------------------------
```

Mit ***label data*** *"Karies in Abhängigkeit vom Geschlecht"* erhält die Datei einen Namen.

```
Contains data from D:\Stata\geschl_karies.dta
  obs:             79                          Karies in Abhängigkeit vom Geschlecht
 vars:              2                          1 Aug 2021 16:30
 size:            158
```

Den häufig codierten Namen von Variablen möchte man einen Klarnamen hinzufügen.
Mit ***label variable geschl "Geschlecht"*** und
label variable karies "Zahnkaries" kann dies erreicht werden und man erhält mit ***describe***

```
Contains data from D:\Stata\geschl_karies.dta
  obs:             79                          Karies in Abhängigkeit vom Geschlecht
 vars:              2                          1 Aug 2021 16:30
 size:            158
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
geschl          byte    %8.0g                 Geschlecht
karies          byte    %8.0g                 Zahnkaries
-------------------------------------------------------------------------------
```

Mit ***label define sexlabel 1 "W" 2 "M"*** und direkt danach
label values geschl sexlabel
werden die Value labels (Wertelabels) für die Var ***geschl*** vergeben (1=W , 2=M) und das Label erscheint blau. Analog lassen sich die Werte der Var ***karies*** labeln.

geschl	karies
1	0
2	0
2	0
1	0

geschl	karies
W	0
M	0
M	0
W	0

geschl	karies
W	nein
M	nein
M	nein
W	nein

label define karlabel 0 "nein" 1 "ja"
label values karies karlabel

ohne Label

geschl	karies 0	1	Total
1	14	27	41
2	12	26	38
Total	26	53	79

mit Label

Geschlecht	Zahnkaries nein	ja	Total
W	14	27	41
M	12	26	38
Total	26	53	79

Mit der Label-Vergabe erhält man besser verständliche Darstellungen der Ergebnisse (hier z.B. einer Vierfeldertafel).

Beim **Zusammenfügen von Datendateien** gibt es zwei Möglichkeiten:

1. Zusammenfügen von Fällen mit dem Kommando ***append***
Hier werden Fälle aus zwei Dateien mit (idealerweise) gleichen Variablen zusammengefügt. Beispiel: Patientendaten mit den Variablen ***age*** und ***height*** aus zwei verschiedenen Kliniken (***klin***=1 und ***klin***=2). Aus Klinik 1 seien die Daten muscle1.dta und aus Klinik 2 die Daten muscle2.dta.
append using "D:\Stata\muscle1.dta" "D:\Stata\muscle2.dta"
führt beide Dateien zusammen. Die Variable ***klin*** identifiziert, aus welcher Klinik die Daten stammen. Klinik 2 hatte noch ein zusätzliches Merkmal ***mvc*** dokumentiert, das bei den Fällen aus Klinik 1 als fehlende Werte angegeben wird.
Das gleiche Ergebnis erhält man nach Laden von muscle1.dta ins Programm mit
use "D:\Stata\muscle1.dta" und anschließendem Kommando
append using "D:\Stata\muscle2.dta"
mit ***replace id = _n*** erhält man wieder eine fortlaufende Identifikationsnummer.
Mit ***save "D:\Stata\muscle3.dta"*** werden die zusammengefügten Daten gespeichert.

2. Zusammenfügen von Variablen mit dem Kommando ***merge***
Hier werden zu einer Datei (master) neue Variable hinzugefügt. Beispiel: Patientendaten ***age*** und ***height*** aus einer Klinik werden nachträgliche Befunde ***mvc*** aus einer zweiten Datei (using)hinzugefügt.
use "D:\Stata\muscle4.dta" mit anschließendem Kommando
merge 1:1 id using "D:\Stata\muscle5.dta" liefert folgendes Ergebnis:

```
Result                      # of obs.
-----------------------------------------
not matched                 2
  from master               1  (_merge==1)
  from using                1  (_merge==2)

matched                     3  (_merge==3)
```

Da die ***id*** jeden Fall eindeutig identifiziert, spricht man von 1:1 merge. Drei Fälle konnten eindeutig zusammengefügt werden (matched). id = 4 fehlte in der Masterdatei (using only) und id = 5 fehlte in der Using-Datei (master only). In den zwei Fällen war das Zusammenführen nicht möglich. Speichern des Resultates mit ***save "D:\Stata\muscle6.dta"*** .

id	age	height	mvc	_merge
1	24	166	540	matched (3)
2	27	175	417	matched (3)
3	32	179	368	matched (3)
5	34	175	.	master only (1)
4	.	.	216	using only (2)

STATA - Kommandos für Datenmanagement

use

rename

generate

list in # / ##

sample #

sample # , count

sort

gsort

drop if

drop in

replace

encode var, generate(varnr)

collapse ..., by(var)

label data

label variable

label define label values

append

merge