Vergleichen von Datensätzen

Handout

Jannis Hutt

2. Februar 2020

Ausgangsbedingungen

Warum vergleichen?

Zwei mögliche Szenarien:

1. Aufbereitung von Datensätzen: vergleichen, welche Verarbeitungsverfahren am Ende der Datenaufbereitung zu welchen Ergebnissen führen.
2. Unterschiedliche Versionen eines Datensatzes vergleichen, um herauszufinden, wo diese verändert wurden.

Installation von cfout

Um einen Datensatz im Speicher (das *master dataset*) mit einem anderen (*using dataset*) zu vergleichen, wird das Programm `cfout` benötigt. Es kann mit über den Stata-Paketmanager heruntergeladen und installiert werden:

```
. ssc install cfout
```

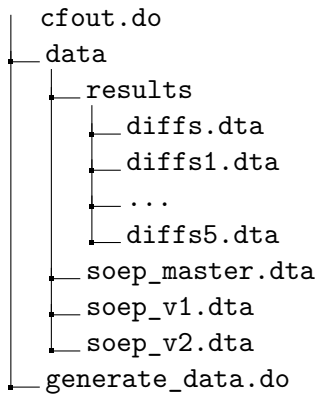
Voraussetzungen

Um Datensätze vergleichen zu können, müssen zwei Grundvoraussetzungen erfüllt sein:

- Eine Variable zur eindeutigen Identifikation des Datensatzes (in diesem Fall `persnr`)
- Die Variablen sowohl im *Master-* als auch *Using Dataset* müssen benannt sein.

Arbeitsumgebung

Auch wenn das verarbeitende Do-File `cfout.do` ein Verzeichnis höher liegt, ist das Arbeitsverzeichnis für alle Beispiele `data`. Zu Beginn jedes Do-Files wird es mit dem Befehl `cd ~/pfad/zu/data` festgelegt. Hier eine schematische Darstellung der zugrundeliegenden Ordnerstruktur.[1]



Verglichen werden sollen verschiedene Versionen des SOEP-Datensatzes von Wagner et al. (vielen im Seminar auch als `data1.dat` bekannt). Zu Demonstrationszwecken wurden zwei Variationen desselben generiert (Skript dazu siehe Hutt 2020). Die Datensätze liegen im Ordner `data` und haben folgende Namen:

- `soep_master.dta`
- `soep_v1.dta`
- `soep_v2.dta`

Im Arbeitsverzeichnis befindet sich außerdem der Ordner `results`. Darin werden später die Ergebnisse der Vergleiche abgelegt.

Vorgehen

Master Dataset in den Speicher laden

Zuerst muss das *Master Dataset* definiert werden. Es ist die Grundlage für den Vergleich:

```
. use data/soep_master.dta, clear
```

[1] Die Dateien `diffs*.dta` werden erst im Laufe der Verarbeitung generiert.

Einfacher Vergleich

Es sollen die Variablen `state-mar` in den Datensätzen `soep_master.dta` und `soep_v1.dta` verglichen werden. Wichtig dabei ist, dass es in beiden Datensätzen eine gleiche, für jede Beobachtung einmalige Indexvariable geben muss, anhand derer man den Vergleich anstellen kann. In diesem Datensatz ist dies durch `persnr` gegeben. Diese Variable referenziert die immergleiche Person, ganz egal, welche Version des Datensatzes aus welchem Jahr genutzt wird.

```
. cfout state-mar using ///
    soep_v1, id(persnr)

--------------------------------
Number of differences: 17599
Number of values compared: 21644
Percent differences: 81.311%
--------------------------------
```

➔ 17.599 von 21.644 Werten unterscheiden sich (81,311%).

Einfacher Vergleich bei fehlenden Variablen

Bei Datensätzen, die aus Speicherplatzgründen oder für einen bestimmten Forschungszweck modifiziert wurden, kann es durchaus passieren, dass einige Variablen fehlen. Um das zu illustrieren, soll das *Master Dataset* mit `soep_v2.dta` verglichen werden:

```
. cfout hhnr2009-xweights using ///
    soep_v2, id(persnr)

note: the following variables are not in the using data:  yedu eqpter
note: the following observations are only in the master data:

+---------+
|  persnr |
|---------|
|  409601 |
...
| 1310001 |
|---------|

...

 --------------------------------
Number of differences: 5319
Number of values compared: 333932
Percent differences: 1.593%
--------------------------------
```

➔ 5.319 von 333.932 verglichenen Werten unterscheiden sich (1,593%).

➔ Die Variablen `yedu` und `eqpter` fehlen im *Using Dataset*.

➔ Es wird eine Liste mit Beobachtungen ausgegeben, die im *Using Dataset* fehlen.

Unterschiede speichern

Wie die im vorigen Beispiel ausgegebene Tabelle mit fehlenden Beobachtungen bereits vermuten lässt, ist es von Vorteil, die im Vergleich gewonnenen Daten weiterverarbeiten zu können. Im Paket `cfout` wird dies durch die `saving`-Option ermöglicht: mit ihr kann man die festgestellten Unterschiede in einem neuen Datensatz speichern:

```
. cfout wor01-wor12 using soep_v1, ///
    id(persnr) saving(results/diffs, replace)

--------------------------------
Number of differences: 52522
Number of values compared: 64932
Percent differences: 80.888%
--------------------------------

. use results/diffs
. browse
```

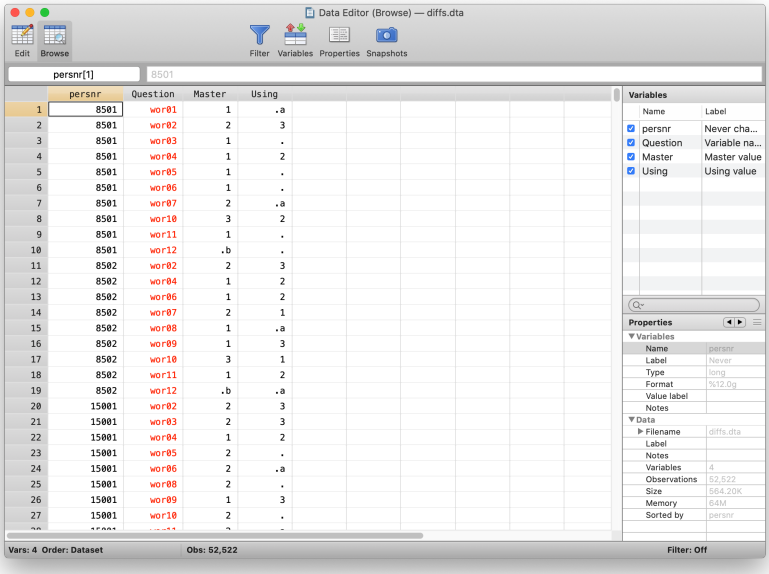


Abbildung 1: Generierte Werte in diffs.dta

- 52.522 von 64.932 verglichenen Werten unterscheiden sich (80,888%).
- Mit `use results/diffs` wird der neu erzeugte Datensatz mit den Unterschieden geöffnet.
- Dort ist angegeben, bei welcher Beobachtung (eindeutig zuzuordnen durch die Variable `persnr`) sich welche Variable (`Question`) wie unterscheidet (`Master` bzw. `Using`; vgl. Abbildung 1).

Unterschiede mit eigenen Variablennamen speichern

Um die Ergebnisse weiterverarbeiten zu können, bietet die *saving*-Option die Möglichkeit, die Variablen des generierten Datensatzes gleich umzubenennen: mit dem Folgenden Befehl werden aus `Question`, `Master` und `Using` gleich `varname`, `soep_master` und `soep_v2`.

```
. use soep_master, clear
. cfout hhnr2009-xweights using soep_v2, id(persnr) ///
    saving(results/diffs1, variable(varname) masterval(soep_master) ///
    usingval(soep_v2) replace)
```

- Die Variable `Question` heißt jetzt `varname`.
- Die Variable `Master` heißt jetzt `soep_master`.
- Die Variable `Using` heißt jetzt `soep_v2`.

Alle Variablen in einem neuen Datensatz speichern

Mit der Sub-Option `all` ist es möglich, alle Variablen des *Master Datasets* in den neuen Datensatz zu übertragen, auch wenn sie nicht Gegenstand des Vergleichs waren. Eine Dummy-Variable `diff` wird erzeugt und für jeden Wert eine neue Zeile angelegt (vgl. Abbildung 2). Die Anzahl an Unterschieden lässt sich nun auch ermitteln, in dem man beispielsweise zählt, wie häufig die Variable `diff` den Wert 1 hat.

```
. cfout hhnr2009-xweights using soep_v2, id(persnr) ///
    saving(results/diffs2, all replace)

--------------------------------
Number of differences: 5319
Number of values compared: 333932
Percent differences: 1.593%
--------------------------------

. use results/diffs2
. count if diff
5,319
```

- Jede Variable jeder einzelnen Beobachtung hat nun eine eigene Reihe.
- Die Variable `limit` gibt Auskunft darüber, ob es Veränderungen gab (das war exakt 5.319 mal der Fall).
- Der Datensatz ist nun einige hundert Zeilen länger.

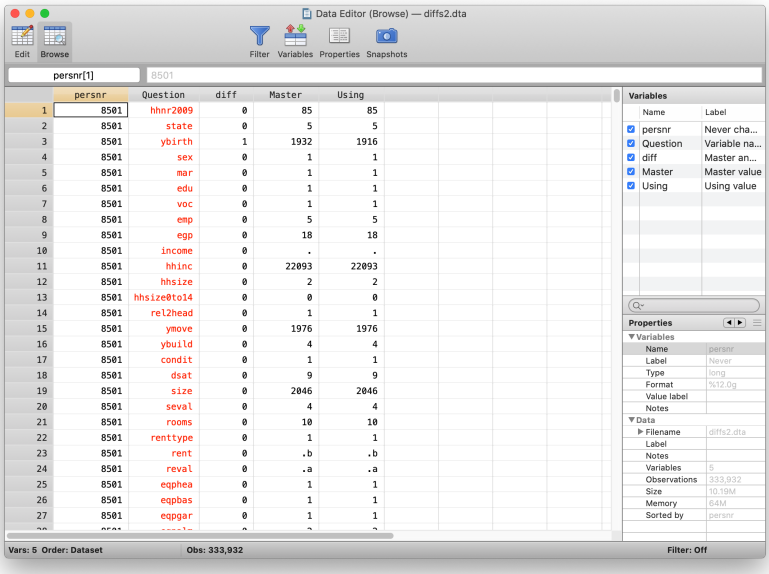


Abbildung 2: Vergleichs-Datensatz mit Dummy-Variable *diff*

Variablen aus dem *Master*- oder *Using Dataset* übernehmen

Die Sub-Optionen `keepmaster` und `keepusing` übernehmen Variablen, die nicht Gegenstand des Vergleichs waren, aus dem *Master*- bzw. *Using Dataset* in den neuen Datensatz. Das kann vor allem dann von Vorteil sein, wenn eine Variable in einem der Datasets fehlt.

```
. cfout hhnr2009-xweights using soep_v2, id(persnr) ///
    saving(results/diffs3, keepmaster(yedu) replace)

note: the following variables are not in the using data:  yedu eqpter
note: the following observations are only in the master data:

--------------------------------
Number of differences: 5319
Number of values compared: 333932
Percent differences: 1.593%
--------------------------------
note: not all observations were compared; there are observations only in the master data.
```

- Die Variablen `yedu` und `eqpter` sind nur im *Master Dataset* enthalten. Erste Variable wurde in den neuen Datensatz kopiert.
- `cfout` warnt, dass nicht alle Beobachtungen verglichen werden konnten, weil manche davon nicht im *Using Dataset* vorhanden sind.

Speicherart der verglichenen Variablen erfassen

Gerade wenn man mit großen Datensätzen arbeitet oder wenn die Struktur des Datensatzes weiter bearbeitet werden soll, kann es nützlich sein, den Datentyp, in dem die Werte einer Variable gespeichert sind, zu erfassen. Möglich ist das mit dem Parameter properties(type), der die Spalte type erzeugt (vgl. Abbildung 3):

```
. use soep_master, clear
. cfout hhnr2009-xweights using soep_v1, id(persnr) ///
    saving(results/diffs5, properties(type) replace)

. use results/diffs5
. browse
```

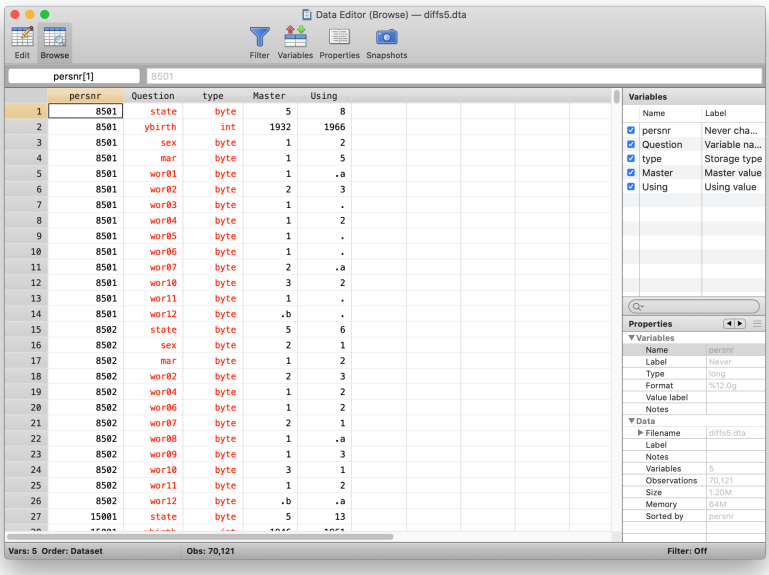


Abbildung 3: In der Spalte *type* ist der Datentyp der Variablen vermerkt.

- Die Variable `type` gibt nun den Datentyp der jeweiligen Variablen an.
- Um eine Dummy-Variable zur Kennzeichnung aller Zeichenketten (strings) zu erstellen, könnte man den Befehl `gen isString = strmatch(type, ßtr*")` nutzen.

Literatur- und Quellenverzeichnis

Jannis Hutt. Skript zum Generieren von Beispieldatensätzen mit zufälligen Werten auf Basis des SOEP-Datensatzes, 2020. URL: https://gist.github.com/hutt/388768dc813cf0192ff9f00258ab01b0.

Ryan Knight. cfout manual entry. URL: http://fmwww.bc.edu/RePEc/bocode/c/cfout.html.

Wolfgang Ludwig-Mayerhofer. Internet Guide to Stata, 2005–2020. URL: http://wlm.userweb.mwn.de/Stata/. Letzter Zugriff: 02.02.2020.

PovertyAction. cfout-Projektseite. URL: https://github.com/PovertyAction/cfout.

Gert. G. Wagner, Joachim R. Frick, Jürgen Schupp, Silke Anger, Jan Goebel, Markus M. Grabka, Elke Holst, Peter Krause, Martin Kroh, Elisabeth Liebau, Henning Lohmann, Christian Schmitt, und C. Katharina Spieß. Sozio-oekonomisches Panel (SOEP), Daten der Jahre 1984 - 2009, 2010. DOI: 10.5684/soep.v26.