User Guide Cell-Key-Methode

Inhalt

1. Überblick über die Cell-Key-Methode

1.1 Allgemeine Informationen

Die Cell-Key-Methode (CKM) wird für bestimmte Statistiken als neue automatisierte Geheimhaltungsmethode in den Forschungsdatenzentren des Bundes und der Länder eingeführt. Dank ihr soll der Aufwand in der Geheimhaltungsprüfung reduziert und somit eine zügigere Freigabe der Ergebnisse ermöglicht werden. Bei der Cell-Key-Methode handelt es sich um ein posttabulares datenveränderndes Geheimhaltungsverfahren. Das bedeutet, dass die Geheimhaltung durch eine Veränderung von Tabellenwerten erfolgt, indem diese mit einem deterministischen Fehlerterm überlagert werden. Dabei stellt das Verfahren sicher, dass die Überlagerung konsistent ist, also logisch identische Tabellenwerte über alle Tabellen hinweg identisch bleiben. So nehmen Randsummen einer Kreuztabelle, beispielsweise „Bundesland x Alter“, immer die gleichen Werte an, die auch in den entsprechenden Fallzahltabellen der beiden Merkmale ausgegeben werden. Das gilt unabhängig davon, ob der Wert als Innen- oder als Randfeld einer Tabelle auftritt.[1]

Hinweis:

Damit Ergebnisse freigegeben werden können, ist es zwingend erforderlich, die in diesem User-Guide beschriebenen Funktionen zu verwenden. Werden Ergebnisse nicht mit den CKM-Funktionen erzeugt, so ist eine Freigabe ausgeschlossen.

[1] Weitere Informationen zur Cell-Key-Methode sind in folgendem WISTA-Artikel zu finden:

„DIE CELL-KEY-METHODE IN DEN FORSCHUNGSDATENZENTREN DER STATISTISCHEN ÄMTER DES BUNDES UND DER LÄNDER“

https://www.destatis.de/DE/Methoden/WISTA-Wirtschaft-und-Statistik/2024/03/cell-key-methode-teil1-032024.pdf?__blob=publicationFile

https://www.destatis.de/DE/Methoden/WISTA-Wirtschaft-und-Statistik/2024/03/cell-key-methode-teil2-032024.pdf?__blob=publicationFile&v=7

2. Nutzung der Cell-Key-Methode in den Forschungsdatenzentren

Um eine fehlerfreie Verwendung der Cell-Key-Methode sicherzustellen, wurde eine Schnittstelle programmiert, die auf Grundlage verschiedener Eingangsparameter die Überlagerung automatisch vornimmt und die Ergebnisse abspeichert. Dabei wurde eine Reihe deskriptiver Auswertungsarten integriert, deren tabellarischer Output mit der Cell-Key-Methode erstellt werden kann.

2.1. Einrichtung des CKM-Tools an den Gastwissenschaftsarbeitsplätzen

Neben den üblichen Anforderungen an das Auswertungsskript, die in der allgemeinen Mustersyntax für GWAP-Nutzungen beschrieben werden, sind für die Cell-Key-Methode weitere Vorbereitungen erforderlich.

Der Pfad zum CKM-Tool muss korrekt gesetzt sein. Dies geschieht vor der Ankunft der Datennutzenden am Gastwissenschaftsarbeitsplatz durch die Mitarbeiterinnen und Mitarbeiter des FDZ. Außerdem sind die R-Pakete *curl*, *data.table*, *dplyr*, *httr2*, *jsonlite* und *tidyverse* erforderlich.

```r
# Pfadangaben fuer die notwendigen Pfade

pfad_ckm      <- "C:/Programme                                    # Pfad zu dem Ordner "CKM-3.1.2" angeben

library(data.table)
library(jsonlite)
library(dplyr)
library(httr2)
library(curl)
library(tidyverse)
```

Das Tool muss nun eingerichtet werden, um die Funktionalitäten verwenden zu können. Dies geschieht durch Aufrufen des Skriptes *start_ckm*. In diesem Schritt werden ebenfalls die CSV-Dateien erstellt, in denen die (überlagerten) Ergebnisse gespeichert werden. Daher ist es erforderlich, zuvor die Variable *outputname*, die den Namen der CSV-Datei angibt, sowie den Pfad, an dem die Outputdatei gespeichert werden soll, zu definieren.

```r
# Pfad und Name der CSV-Datei, die die Ergebnisse enthaelt

outputpfad    <- "[Outputpfad hier angeben]"
outputname    <- "Ergebnisse"

#CKM-Tool starten
source(paste(pfad_ckm, "/CKM-3.1.2/ckm_scripts/", "start_ckm.R", sep = ""))
```

Analog der Aufbau in Stata:

```
* Pfadangaben für die notwendigen Pfade

global pfad_ckm     C:/Programme

global outputpfad ${pfad_ckm}/Test                          // hier werden die (überlagerten) Kreuztabellen gespeichert
global outputname = "Ergebnisse_Test"


global logname  $outputname                                 // Name des log-files einfügen



* ------ CKM-Tool starten ------

cd "${pfad_ckm}/CKM-3.1.2/ckm_scripts/"

global rpath        "C:\Program Files\R\R-4.5.0\bin\R.exe"

do start_ckm

* -----------------------------
```

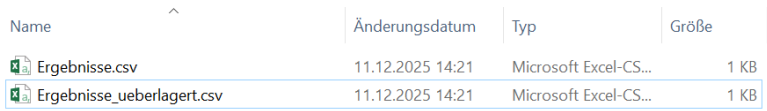
Die Variable *rpath* ist notwendig, da Stata einen Umweg über R gehen muss, um die CKM aufrufen zu können. Dies passiert jedoch im Hintergrund und man muss keine Erfahrung mit R besitzen, um die CKM mit Stata verwenden zu können.

Hinweis:

Durch Ausführen des Skriptes *start_ckm* werden automatisch bestehende CSV-Dateien, die denselben Namen haben, wie durch die Variabel *outputname* definiert, überschrieben! Dadurch wird sichergestellt, dass die Ergebnisdateien, bei einem finalen Start des Skriptes für die Erzeugung von Ergebnissen, nicht zusätzlich aus vorläufigen Ergebnissen von Testläufen bestehen.

Der Outputname kann auch im Laufe eines Auswertungsskriptes geändert werden, wodurch automatisch eine weitere Outputdatei angelegt wird. Somit ist die Erstellung mehrerer Ergebnisdateien in einem Skript möglich.

Nach Ausführen des Skriptes *start_ckm* werden unter dem Pfad, der durch die Variable *outputpfad* definiert ist, zwei (zunächst leere) CSV-Dateien mit dem unter der Variable *outputname* angegebenen Namen erstellt. In diesen werden jeweils die originalen bzw. überlagerten Ergebnisse gespeichert.



2.2. Auswertungen mit dem CKM-Tool

Im Folgenden werden die Funktionalitäten des CKM-Tools beschrieben. Grundsätzlich wird das Tool verwendet, um tabellarischen Output sowie Perzentile zu erzeugen. Da Fallzahlen ausschließlich überlagert veröffentlicht werden dürfen, wirkt sich die Einführung der Cell-Key-Methode auch auf multivariate Analysen aus. Weitere Informationen dazu befinden sich in Kapitel 2.2.8.

Um die Funktionalitäten aufzuzeigen, nehmen wir an, dass ein Datensatz mit dem Namen *daten* in die Umgebung geladen wurde. Dieser Datensatz besteht aus 1000 Einzeldaten und enthält verschiedene Merkmale wie Alter, Geschlecht und wohnhaftes Bundesland sowie die zu den Einzeldaten gehörigen *Record Keys*.

Hinweis:

Ein Datensatz kann weiterhin beliebig gefiltert und zugeschnitten werden. Eine Ausnahme bildet dabei die Spalte der Record Keys, die für die Cell-Key-Methode erforderlich ist und nicht aus dem Datensatz entfernt werden darf. Die CKM führt ansonsten nicht die Überlagerung aus und gibt eine Fehlermeldung zurück.

Anmerkung: Für die Zensusdaten existiert das Merkmal *AGS_12*, welches ebenfalls nicht entfernt werden darf, da mit diesem Merkmal geprüft wird, ob Fallzahlen ab Gemeindeebene überlagert werden müssen oder nicht.

2.2.1. Fallzahlen & Benennung von Tabellen

Um eine eindimensionale Fallzahltabelle zu erzeugen, muss die Funktion *ckm* mit dem gewünschten Datensatz sowie der Variable, deren Fallzahlen berechnet werden soll, als Input ausgeführt werden.

```
ckm(daten = daten, x = "ALTER")
```

Allgemein gilt: `ckm(daten = NAME_DES_DATENSATZES, x = NAME_DER_VARIABLE)`

In der CSV-Datei mit den Originalergebnissen erhalten wir in unserem Beispiel:

Tabelle ALTER						
	ALTER	18 bis unter 30	30 bis unter 50	50 bis unter 65	65 und aelter	Insgesamt
Insgesamt		182	590	155	73	1000

Analog erhalten wir eine Fallzahltabelle mit den überlagerten Ergebnissen in der entsprechenden CSV-Datei:

Tabelle ALTER ueberlagert						
	ALTER	18 bis unter 30	30 bis unter 50	50 bis unter 65	65 und aelter	Insgesamt
Insgesamt		182	593	157	73	999

Wenn nicht explizit angegeben, handelt es sich um R-Syntax. Jedes Beispiel wird mit entsprechendem Hinweis jedoch auch in Stata aufgezeigt.

In Stata wird beispielsweise eine überlagerte Fallzahltabelle mit dem Merkmal *Kinderanzahl* folgendermaßen erzeugt:

```
do ckm KINDERANZAHL KINDERANZAHL
```

Hinweis:

Auch in Stata können Tabellen mit einem Merkmal erstellt werden. Anders als in R muss jedoch das Merkmal, für das eine eindimensionale Tabelle erstellt werden soll, zwei Mal hintereinander aufgeschrieben werden.

Um eine zweidimensionale Fallzahltabelle zu erzeugen, muss die Funktion *ckm* mit dem gewünschten Datensatz sowie den beiden Variablen, deren Fallzahlen berechnet werden soll, als Input ausgeführt werden.

```
ckm(daten = daten, x = "EU_SEX", y = "ALTER")
```

In Stata:

```
do ckm EU_SEX ALTER
```

Tabelle EU_SEX x ALTER						
	ALTER	18 bis unter 30	30 bis unter 50	50 bis unter 65	65 und aelter	Insgesamt
EU_SEX						
m		86	290	80	28	484
w		96	300	75	45	516
Insgesamt		182	590	155	73	1000

Tabelle EU_SEX x ALTER ueberlagert						
	ALTER	18 bis unter 30	30 bis unter 50	50 bis unter 65	65 und aelter	Insgesamt
EU_SEX						
m		86	290	76	27	482
w		95	300	77	45	513
Insgesamt		182	593	157	73	999

Die bereits erwähnte Konsistenz von Ergebnissen wird hier deutlich, wenn man die Randwerte des Merkmals *Alter* der Tabellen mit den obigen Fallzahltabellen vergleicht. Die entsprechenden überlagerten Zellen sind identisch.

Standardmäßig werden die verwendeten Merkmale sowie die Auswertungsart als Tabellentitel erstellt. Es besteht die Möglichkeit, jeden CKM-Output einzeln zu benennen. Im folgenden Beispiel soll eine Fallzahltabelle der verschiedenen aggregierten Ausprägungen des Merkmals Alter erzeugt werden. Der Tabellenname soll dementsprechend “Aggregierte Fallzahlen des Merkmals Alter“ lauten.

```
ckm(daten = daten, x = "ALTER", tabellenname = "Aggregierte Fallzahlen des Merkmals Alter")
```

In Stata:

```
global Tabellenname "Fallzahlen des Merkmals Alter"
do ckm ALTER ALTER
```

Aggregierte Fallzahlen des Merkmals Alter						
	ALTER	18 bis unter 30	30 bis unter 50	50 bis unter 65	65 und aelter	Insgesamt
Insgesamt		182	590	155	73	1000

Aggregierte Fallzahlen des Merkmals Alter ueberlagert						
	ALTER	18 bis unter 30	30 bis unter 50	50 bis unter 65	65 und aelter	Insgesamt
Insgesamt		182	593	157	73	999

Somit ist es beispielsweise auch möglich, nummerierten Output automatisiert durch Iteration zu erzeugen.

2.2.2. Relative Häufigkeiten

Um relative Häufigkeiten von Merkmalskombinationen tabellarisch ausgeben zu lassen, muss der Input der Funktion *ckm* um den Eintrag *mode = "Relativ"* ergänzt werden:

```
ckm(daten = daten, x = "EU_SEX", y = "ALTER", mode = "Relativ")
```

In Stata:

```
do ckm EU_SEX ALTER Relativ
```

Die Zelle, die den Gesamtwert angibt (rechts unten in der Tabelle), ist dabei auch bei der überlagerten Tabelle immer auf den Wert 1 gesetzt.

Damit ergibt sich mit dem Beispieldatensatz folgender Output:

Tabelle EU_SEX x ALTER Relativ						
	ALTER	18 bis unter 30	30 bis unter 50	50 bis unter 65	65 und aelter	Insgesamt
EU_SEX						
m		0.086	0.29	0.08	0.028	0.484
w		0.096	0.3	0.075	0.045	0.516
Insgesamt		0.182	0.59	0.155	0.073	1

Tabelle EU_SEX x ALTER Relativ ueberlagert						
	ALTER	18 bis unter 30	30 bis unter 50	50 bis unter 65	65 und aelter	Insgesamt
EU_SEX						
m		0.08609	0.29029	0.07608	0.02703	0.48248
w		0.0951	0.3003	0.07708	0.04505	0.51351
Insgesamt		0.18218	0.59359	0.15716	0.07307	1

2.2.3. Mittelwerte

Mittelwerte werden auf Grundlage eines weiteren Merkmals berechnet, das ebenfalls in der Funktion *ckm* angegeben werden muss. Wichtig ist, dass dieses Merkmal numerisch ist, da ansonsten der Mittelwert nicht sinnvoll berechnet werden kann. Das Merkmal *ALTER* des Beispieldatensatzes in seiner aggregierten Form mit den Ausprägungen *18 bis unter 30, 30 bis unter 50, 50 bis unter 65* sowie *65 und aelter* kann dementsprechend nicht gemittelt werden, da es von der Programmiersprache als Merkmal mit *character*- bzw. *string*-Ausprägungen interpretiert wird.

```
> ckm(daten = daten, x = "EU_SEX", mode = "Mittelwert", z = "ALTER")
Fehler in auswertungsarten(mode = mode, daten = daten, x = x, y = y, z = z,  :
  Fuer diese Auswertungsart muessen die Werte des Merkmals ausschliesslich numerisch sein.
```

Ein weiteres Merkmal des Beispieldatensatzes ist *KINDERANZAHL*. Da dieses Merkmal numerisch ist, kann dessen Mittelwert, erneut als tabellarischer Output über die Merkmalsausprägungen von

EU_SEX, berechnet werden. Nachfolgend ist der entsprechende Output nicht wie zuvor aus der CSV-Outputdatei entnommen, sondern aus der R-Konsole:

```
> ckm(daten = daten, x = "EU_SEX", mode = "Mittelwert", z = "KINDERANZAHL")
[1] "CKM: Tabelle  EU_SEX  x  EU_SEX   Mittelwert   KINDERANZAHL"
[1] "Tabelle  EU_SEX  x     Mittelwert   KINDERANZAHL"
          EU_SEX m         w         Insgesamt
Insgesamt ""     "1.32882" "1.48033" "1.402"   ""
          EU_SEX m         w         Insgesamt
Insgesamt ""     "1.32689" "1.47412" "1.401"   ""
```

```
Die csv-Dateien wurden erfolgreich erstellt und in C:/Program Files/PythonENV/CKM-3.1.2/ckm_user/Tabellen_R/ gespeichert.
```

In Stata:

```
do ckm EU_SEX EU_SEX Mittelwert KINDERANZAHL
```

Es wird die durchschnittliche Kinderanzahl, unterschieden nach Geschlecht, ausgegeben. Eine Kreuztabelle mit zwei Merkmalen ist ebenfalls möglich. Eine Variable *y* mit dem gewünschten weiteren Merkmal muss dafür im Input für die Funktion *ckm* ergänzt werden.

2.2.4. Wertsummen & Relative Wertsummen

Die formalen Anforderungen an den Input für die Auswertungsarten *Wertsumme* und *Wertsumme_Relativ* sind analog zu den Anforderungen in Kapitel *2.2.3. Mittelwerte*.

```
ckm(daten = daten, x = "EU_SEX", mode = "Wertsumme", z = "KINDERANZAHL")
```

In Stata:

```
do ckm EU_SEX EU_SEX Wertsumme KINDERANZAHL
```

Tabelle EU_SEX Wertsumme KINDERANZAHL				
	EU_SEX	m	w	Insgesamt
Insgesamt		687	715	1402

Tabelle EU_SEX Wertsumme KINDERANZAHL ueberlagert				
	EU_SEX	m	w	Insgesamt
Insgesamt		682.01934	710.52588	1399.599

```
ckm(daten = daten, x = "EU_SEX", mode = "Wertsumme_Relativ", z = "KINDERANZAHL")
```

In Stata:

```
do ckm EU_SEX EU_SEX Wertsumme_Relativ KINDERANZAHL
```

Tabelle EU_SEX Wertsumme_Relativ KINDERANZAHL				
	EU_SEX	m	w	Insgesamt
Insgesamt		0.49001	0.50999	1

Tabelle EU_SEX Wertsumme_Relativ KINDERANZAHL ueberlage				
	EU_SEX	m	w	Insgesamt
Insgesamt		0.48712	0.5082	1

2.2.5. Quantile

Die Funktion *quantile* aus R wurde in die entsprechende CKM-Funktion eingebettet. Analog zu der R-Funktion wird als Input eine Liste von Daten sowie eine Liste mit den gewünschten Perzentilen benötigt. Die Werte des angegebenen Merkmals müssen erneut numerisch sein.

Dieses Mal sind die Werte des Merkmals nicht aggregiert, sondern geben das tatsächliche Alter der entsprechenden Person als numerischen Wert an. Um den Median zu berechnen, wird als Input für die Variable *percentile* eine Liste mit dem Wert *0.5* gesetzt.

```
ckm(daten = daten, x = "ALTER", mode = "Quantil", percentile = c(0.5))
```

In Stata funktioniert die *pctile* so, dass eine ganze Zahl *n* als Input gegeben wird, die das Intervall [0,1] in *n* gleichgroße Stücke aufteilt. Dementsprechend wird der Median mit der Zahl *n=2* berechnet:

```
do ckm ALTER ALTER Quantil 2
```

Perzentile ALTER			
freq	perzentile	publish	ALTER
1000	0.5	TRUE	38

Perzentile ALTER _ueberlagert			
freq	perzentile	publish	ALTER
999	0.5	TRUE	38

Wie man sieht, wird die überlagerte Fallzahl bei dem CKM-Output unter *freq* angegeben. Wenn der Wert in der Spalte *publish* auf TRUE gesetzt ist, darf die entsprechende Zeile veröffentlicht werden.

Man kann in R beliebige Zahlen zwischen 0 und 1 in die Liste der Perzentile ergänzen, um den Wert für das entsprechende Perzentil zu erhalten. Gemäß den Vorgaben für die Geheimhaltung, hängt die Freigabe sowohl von der Fallzahl als auch von dem gewünschten Perzentil ab. Die Werte 0 (Minimum) sowie 1 (Maximum) dürfen nie freigegeben werden. Dementsprechend ist im folgenden Beispiel der Wert für 1 in der Spalte *publish* auf FALSE gesetzt.

```
ckm(daten = daten, x = "ALTER", mode = "Quantil", percentile = c(0.1, 0.25, 0.5, 0.6, 0.99, 1))
```

Perzentile ALTER			
freq	perzentile	publish	ALTER
1000	0.1	TRUE	22
1000	0.25	TRUE	30
1000	0.5	TRUE	38
1000	0.6	TRUE	39
1000	0.99	TRUE	67
1000	1	FALSE	80

Perzentile ALTER _ueberlagert			
freq	perzentile	publish	ALTER
999	0.1	TRUE	22
999	0.25	TRUE	30
999	0.5	TRUE	38
999	0.6	TRUE	39
999	0.99	TRUE	67
999	1	FALSE	80

In Stata würde der Befehl für eine Aufteilung in die Perzentile 0.1, 0.2, 0.3, …, 0.9 folgendermaßen aussehen:

```
do ckm ALTER ALTER Quantil 10
```

2.2.6. Salden

Für Salden und Veränderungsraten werden zwei unterschiedliche Zeiträume miteinander verglichen. Für den Fall des Zensus bedeutet dies, dass Salden beziehungsweise Veränderungsraten zwischen dem Zensus 2011 und dem Zensus 2022 berechnet werden.

Für das Beispiel wird erneut der Datensatz *daten* mit 1000 Beobachtungen (stellvertretend für den Zensus 2022) sowie ein Datensatz *daten_vergl* (Zensus 2011) mit 1027 Beobachtungen verwendet.

Um ein besseres Verständnis für die Funktion *Saldo* zu erhalten, wird die Funktion *ckm* ohne Merkmals- sowie Auswertungsangabe auf die beiden Datensätze angewendet, um die originalen und überlagerten Fallzahlen zu erhalten:

```
> # Überlagerte Fallzahlen erzeugen
> ckm(daten = daten)
[1] "CKM: Fallzahl"
[1] 1000
[1] 999

> # Überlagerte Fallzahlen erzeugen
> ckm(daten = daten_vergl)
[1] "CKM: Fallzahl"
[1] 1027
[1] 1026
```

```
[1] "Saldo Geschlecht"
          EU_SEX m    w     Insgesamt
Insgesamt ""     "19" "-46" "-27"     ""
          EU_SEX m    w     Insgesamt
Insgesamt ""     "19" "-46" "-30"     ""
```

In Stata:

```
do ckm EU_SEX EU_SEX Saldo
```

2.2.7. Veränderungsraten & Veränderungsraten von Werten

Analog zu Salden werden bei Veränderungsraten zwei Datensätze betrachtet. Die Auswertungsart gibt die relative Veränderung der Fallzahlen ausgehend von dem Datensatz *daten_vergl* an. Mit den Fallzahlen aus dem vorherigen Kapitel zu Salden ergibt sich

$$\frac{1000 - 1027}{1027} = -\frac{27}{1027} \approx -0.02629,$$

beziehungsweise

$$\frac{999 - 1026}{1026} = -\frac{27}{1026} \approx -0.02632,$$

was genau dem folgenden Ergebnis entspricht:

```
ckm(daten = daten, daten_vergl = daten_vergl, x = "EU_SEX", mode = "Veraenderungsrate", tabellenname = "Veränderungsrate")
```

```
[1] "Veränderungsrate"
          EU_SEX m          w           Insgesamt
Insgesamt ""     "0.03918" "-0.08487" "-0.02629" ""
          EU_SEX m          w           Insgesamt
Insgesamt ""     "0.04132" "-0.07807" "-0.02632" ""
```

Unter Angabe eines weiteren numerischen Merkmals kann man mit der Auswertungsart *Veraenderungsrate_Werte* die Veränderungsrate der Wertsumme dieses Merkmals berechnen:

```
ckm(daten = daten, daten_vergl = daten_vergl, x = "EU_SEX", mode = "Veraenderungsrate_Werte", z = "ALTER", tabellenname = "Veränderungsrate Alter")
[1] "Veränderungsrate Alter"
          EU_SEX m          w           Insgesamt
Insgesamt ""     "0.04585" "-0.07959" "-0.02104" ""
          EU_SEX m          w           Insgesamt
Insgesamt ""     "0.04951" "-0.0727"  "-0.02131" ""
```

In Stata:

```
do ckm EU_SEX EU_SEX Veraenderungsrate

do ckm EU_SEX EU_SEX Veraenderungsrate_Werte ALTER
```

2.2.8. Multivariate Analysen

Es gilt, dass bei einer Geheimhaltung mit der Cell-Key-Methode die originale Beobachtungsanzahl (Fallzahl) niemals veröffentlicht werden darf. Bei multivariaten Analysen beziehungsweise Auswertungen, in denen die Fallzahl unmittelbar abgeleitet werden kann, muss daher die originale Fallzahl durch die überlagerte Fallzahl ersetzt werden. Im folgenden Beispiel wurde eine einfache Regressionsanalyse auf den Beispieldatensatz angewendet. Da die Anzahl der Freiheitsgrade hier im konkreten Beispiel durch **Freiheitsgrade = Fallzahl – 2** berechnet wird, kann die originale Fallzahl direkt zurückgerechnet werden.

```
> summary(lm(ALTER ~ KINDERANZAHL,
+     data = daten))

Call:
lm(formula = ALTER ~ KINDERANZAHL, data = daten)

Residuals:
    Min      1Q  Median      3Q     Max
-29.169  -8.207   0.312   7.831  39.275

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   36.1688     0.6121  59.085  < 2e-16 ***
KINDERANZAHL   1.5188     0.3601   4.218 2.69e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.97 on 998 degrees of freedom
Multiple R-squared:  0.01751,   Adjusted R-squared:  0.01653
F-statistic: 17.79 on 1 and 998 DF,  p-value: 2.689e-05

> # Überlagerte Fallzahlen erzeugen
> ckm(daten = daten)
[1] "CKM: Fallzahl"
[1] 1000
[1] 999
```

Es ist daher für die Freigabe der Ergebnisse **unbedingt notwendig** die Funktion *ckm* auf dem Datensatz, auf dem die multivariate Auswertung ausgeführt wurde, unmittelbar im Anschluss auszuführen. Dann können die Mitarbeitenden des FDZ den Identifikator der Fallzahl (hier die Anzahl der Freiheitsgrade) durch den überlagerten Wert ersetzen. In diesem Beispiel wäre der überlagerte Wert für die Freiheitsgrade 999 – 2 = 997.

2.3. Betrachten von überlagerten Ergebnissen

Wie bereits geschildert, werden die Ergebnisse automatisch in dem angegebenen Outputverzeichnis als CSV bzw. xlsx-Datei gespeichert. In R wird zudem nach jeder Ausführung der CKM die überlagerte Tabelle in der Konsole ausgegeben. Um eine vollständige Kopie des Outputfiles in Stata einzusehen, ist ein Framewechsel nötig. Durch Aufrufen der Funktion *ckm* in Stata wird der Frame *ckm* erstellt, der die überlagerte Outputdatei spiegelt. Ein Framewechsel erfolgt durch die Eingabe

frame change ckm

Vor erneuter Ausführung sollte durch die Eingabe

frame change default

wieder in den Frame des auszuwertenden Datensatzes gewechselt werden.

2.4. Einrichtung und Anwenden des CKM-Tools per KDFV

Beim Versenden der Datenstrukturfiles wird eine Syntax mitversendet, die für die Prüfung der formalen Anforderungen der CKM an das Auswertungsskript verwendet wird.

Die Funktion kann nicht für Überlagerungen oder Auswertungen verwendet werden, sondern prüft lediglich, ob eine tatsächliche Ausführung per KDFV im FDZ möglich ist, oder eine Anforderung nicht erfüllt ist. Beispielsweise etwa, falls die Variable für die Berechnung eines Mittelwertes nicht numerisch ist, oder die Spalte der Record Keys nicht gefunden werden kann. Außerdem wird die Syntax (Input der Funktion korrekt und wie in diesem User-Guide beschrieben) geprüft. Sollte eine Anforderung nicht erfüllt werden, so bricht die Funktion das Skript ab und gibt eine entsprechende Fehlermeldung zurück. Erst wenn das Skript fehlerfrei läuft, darf es für einen KDFV-Lauf auf den echten Daten an das FDZ versendet werden.

2.5. Protokollieren von Auswertungen mit der Cell-Key-Methode

Die Auswertungen müssen protokolliert werden. Dafür ist es notwendig, ein Log während der Ausführung des Auswertungsskriptes laufen zu lassen. Exemplarisch im Folgenden für R und Stata dargestellt:

In R:

```r
sink(paste(pfad_ckm, "/CKM-3.1.2/ckm_user/Tabellen_R/", outputname,".log", sep = ""))

# --------------------
# -- Auswertungen --
# --------------------

sink()
```

In Stata:

```
*   Aufzeichnung in Protokoll starten
capture log close
log using "$outputpfad/${logname}.log", replace

/* -----------
-- Auswertungen
------------- */

*   log-file schließen
log close
```