GESStabs

Introduction to tabulation

Gesellschaft für Software
in der Sozialforschung mbH

Waterloohain 6 - 8
22769 Hamburg
Tel.: + 49 40 - 853 753 - 0
Fax: + 49 40 - 853 753 - 33
www.gessgroup.de

GESStabs
Introduction to tabulation

Andreas Rechenberg
Seline Günther
Judith Krüger

Version: April 2018

GESS Gesellschaft für Software in der Sozialforschung mbH
Waterloohain 6-8
22769 Hamburg
Tel. 0049 - 40 - 853 753 0
Fax 0049 - 40 - 853 753 433
software@gessgroup.de
www.gessgroup.de

Table of Contents

List of Figures

1 Introduction to tabulation

In this tutorial, we will demonstrate the tabulation with GESStabs using a simple example project. For this purpose, we use a small survey on the topic of Internet usage, which was processed in the winter semester 2010/2011 at the University of Hamburg and filled with random numbers for this introduction.

We recommend writing the examples described here yourself. However, you do not have to type everything, please find the files in the archive EXAMPLE.ZIP.

1.1 First steps

First, please install GESStabs on your computer.

1. Execute the program provided by us and follow the installation menu. By default, GESStabs will be installed on your computer at c://GESS. We will provide a (test) license key. Its one-time validity check requires an internet connection.
2. You can use any text editor to work with the GESStabs scripts. We recommend the Notepad++ Editor: This is efficient in application and is supported by us with a syntax highlighting (please find .zip file on our homepage).
3. Reading, processing and writing SPSS files requires a SPSS input/ output interface. We provide these so-called DLL-files. Please save it in the c://GESS directory.
4. You can view created tables (among others) as a PostScript file and save it as a PDF document. A PostScript viewer and PDF Destiller is required. We recommend GhostView and GPL GhostScript, both of which you can download for free on the Internet.
5. At last, provide the directories where you can access the necessary work programs (text editor, PostScript viewer and PDF destiller) in Options.

In the following, we will introduce you to your first steps with GESStabs. After you have worked through this tutorial and the attached example file, you will know how to...

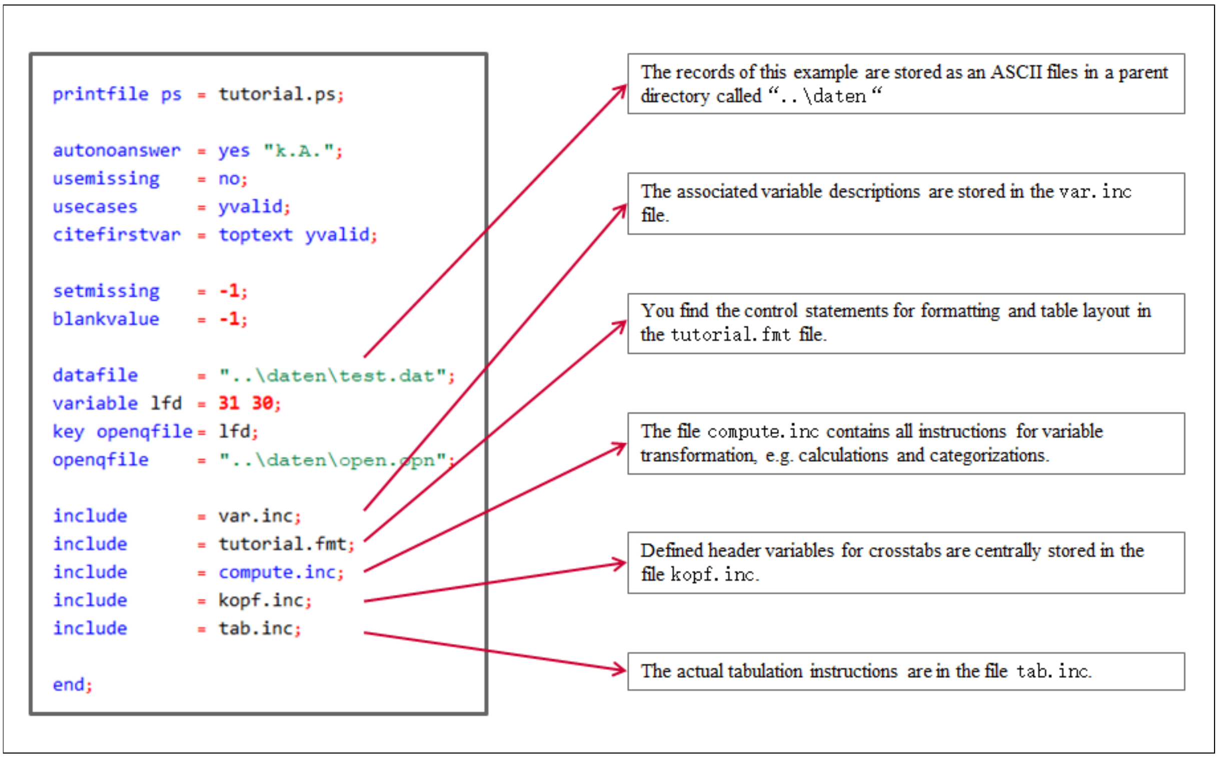
- design a working tabulation script,
- work with an ASCII, SPSS or CSV data record,
- compile cross tables (with multiple table headers),
- vary the appearance of the tables,
- generate top(2)- and bottom(2)-boxes,
- sort by frequency,
- recode,
- display mean values and mean values tables and
- vary the layout of tables.

Let´s start!

GESStabs is a script-oriented tabulation program which means: The software is regulated by a collection of commands in script form. This script is then processed by GESStabs and generates the specified tables.

It is a good programming style to split a script into multiple files. Among our users, the following style has proven to be efficient over the last 25 years:
A file with the extension *.tab serves as the main program. This file includes thematically defined files (subprograms) using the `INCLUDE`-statement.

This main program could look like this, for example:



```
printfile ps  = tutorial.ps;

autonoanswer  = yes "k.A.";
usemissing    = no;
usecases      = yvalid;
citefirstvar  = toptext yvalid;

setmissing    = -1;
blankvalue    = -1;

datafile      = "..\daten\test.dat";
variable lfd  = 31 30;
key openqfile = lfd;
openqfile     = "..\daten\open.opn";

include       = var.inc;
include       = tutorial.fmt;
include       = compute.inc;
include       = kopf.inc;
include       = tab.inc;

end;
```

The records of this example are stored as an ASCII files in a parent directory called “..\daten“

The associated variable descriptions are stored in the `var.inc` file.

You find the control statements for formatting and table layout in the `tutorial.fmt` file.

The file `compute.inc` contains all instructions for variable transformation, e.g. calculations and categorizations.

Defined header variables for crosstabs are centrally stored in the file `kopf.inc`.

The actual tabulation instructions are in the file `tab.inc`.

Abb. 1.1: main.tab

In this tutorial the main program is called `main.tab`. In the listed order, the following files are integrated:

- the data source (...\data\test.dat),
- the variable description of the read data record (`var.inc`),
- the layout file named `tutorial.fmt`,
- variable transformations in a file called `compute.inc`,
- the definition of the table headers[1] in `head.inc`
- and the file with the actual tabulation statements called `tab.inc`.

GESStabs processes these separated control statements from top to bottom, so that ultimately all individual statements are processed as one single stack of commands.

---

[1] In market and opinion research tables are often presented as follows: The groups, according to which something is considered, are located at the top of the table in horizontal direction, the considered answers to a question are displayed vertically, usually showing column percentages. In this instruction to GESStabs, we refer to these so-called break-up groups as the 'header' of the table.

In addition to the `INLCLUDE`-files, the `main.tab` includes the following commands:

```
PRINTFILE PS  = tutorial.ps;

AUTONOANSWER  = YES "k.A.";
USEMISSING    = NO;
USECASES      = YVALID;
CITEFIRSTVAR  = TOPTEXT YVALID;

SETMISSING    = -1;
BLANKVALUE    = -1;

DATAFILE      = "..\daten\test.dat";
VARIABLE lfd  = 31 30;
KEY OPENQFILE = lfd;
OPENQFILE     = "..\daten\open.opn";

END;
```

With a little practice, many of these control statements are self-explanatory. Let's go through the individual lines.

Each control instruction terminates with a semicolon.

`PRINTFILE PS = tutorial.ps;` names an output file. The keyword `PRINTFILE` is followed by the output format, in this case PostScript (`PS`), and the file name, here 'tutorial.ps'.

`AUTONOANSWER = YES;` sets missing replies as "k.A." by default for each question (filters will be considered)[2]. `USEMISSING` causes missing values not to be included in calculations (e.g. mean values). The `USECASES = YVALID;` statement determines that only valid cases of the side variable, i.e. the Y-variable, are taken into account in the tabulation. `CITEFIRSTVAR = TOPTEXT YVALID;` means: print the toptext (usually the question text) of the column variable on top of the table.

`SETMISSING` determines the internal missing value, here -1. `BLANKVALUE` specifies the value to be set for empty data cells. Here, this corresponds to the missing value, i. empty data fields are interpreted as missing.

The final point of the (active) script is set by the keyword `end;`. The following content is read, but no longer interpreted.

---

[2] `AUTONOANSWER = NO;` would be base-reducing.

2 Reading in data records

So, we explained the basic systematic of our GESStabs script. Now we turn to the next fundamental question: How to integrate a dataset?

Typically, these days, you work with an ASCII, SPSS, or CSV data file.

The only difference in handling the data formats is in the respective variable definition. While fixed variable descriptions can be stored in SPSS, the variables in ASCII and CSV records still have to be defined. SPSS and CSV records are structurally similar; in ASCII, variable mapping is controlled by column specifications.

Here is a brief introduction to the three data types.

2.1 ASCII

ASCII records are included in the script as follows[1]:

```
DATAFILE = "filename.dat";
```

In ASCII records, one line corresponds to one case. Variables are described by their column number.

2.1.1 Manual preperation of a var.inc

If there is no prepared `var.inc`[2], this must be written by yourself.

Basically, for a variable definition you only need the variable type, name and column specification.

Single answers

Here is an example of the definition of a single question:

```
SingleQ f0 = 30 1;
```

This defines the following:
The variable contains a maximum of one answer per interview, so it's a variable of type `SingleQ`. The variable name is f0, so that the variable can be called later and processed further. The digits after the equals sign indicate the column position in the record and the number of columns used. The variable f0 is therefore in the 30th column of the data record and has a length of one column.

Then you can assign a `TEXT` and a `TITLE` to the question.

```
SingleQ f2 = 497 1;
TEXT="Since when do you personally use the Internet?";
TITLE="Since...";
```

If no assignment to a particular question exists, text and title are assigned to the last defined question. For a direct assignment, the variable name can be written behind the instruction word[3].

The `LABELS` command identifies the contents of each variable's code.

---

[1] See **??** for the procedure with additional inclusion of an OpenQ data file.

[2] For surveys, which were carried out with GESS Q. or GESS Cati, these can be generated automatically, see next section

[3] You are safe with this procedure, for example if the order of the script content changes.

```
LABELS f2 =
1 "less than 1 year"
2 "1 - 3 years"
3 "3 - 5 years"
4 "more than 5 years"
;
```

Multiple answers, dichotomous

The description of variables with multiple answers is similar to the procedure for `SingleQ` variables.

```
MultiQ f9 = 392 22 2;
TEXT = "For what purposes do you use the Internet?";
TITLE = "Please select everything that applies.";
LABELS =
1 "Rechearch/ advisor"
2 "Videos/ music"
3 "E-mail"
4 "Online gaming"
5 "Shopping/ booking"
6 "Offices/ administration (university, tax)"
7 "Chats & forums"
8 "Online banking"
9 "Online media (magazines, blogs)"
10 "Social networks (Facebook, Parship, XING)"
11 "Other"
;
```

The definition of the column specification is a little different here: All in all, a `MultiQ` needs as many columns as one can give answers (varibale f9 has 11 codes, so it can get a maximum of 11 answers, hence here: 11 columns). Because the label codes 10 and 11 are two digits, each possible answer must provide space for two digits. Accordingly, at most 11x2 = 22 columns are needed. In the column specification above it is specified that the MultiQ f9 starts in column 392 of the data set, requires a total of 22 columns and has a maximum of two digits per answer.

2.1.2 Automatically generated var.inc

If a survey was conducted in GESS Q. or GESS Cati, an associated `var.inc` can be created with just a mouse click. This file contains all variables with their column definition and width. The wording of the variables corresponds to the texts used in the survey.

For our sample dataset we created an automatically generated `var.inc`.

For example, the variable definition of f1 looks like this:

```
SingleQ f1 = 478 1;
TEXT = "Where do you use the Internet at the moment?";
LABELS =
1 "At home"
2 "At work/ training place"
3 "At school/ university"
4 "In a public place (internet cafe, multimedia terminal, ...)"
5 "On the road (via WLAN hotspot, cellphone, ...)"
6 "Outside of the house (with friends, relatives)"
;
```

There was no variable title in the questionnaire. If this is desired in the course of the tabulation, it can be supplemented with the `TITLE` command. For example:

```
TITLE f1 = "Place of internet usage";
```

f3, a multiple-answer question, was described in the same way. With seven labels, one of which is double-digit, the variable needs a maximum of 7x2 = 14 columns per case. Text, title and labels are also defined here automatically.

```
MultiQ f3 = 479 14 2;
TEXT = "At which of the following places do you occasionally use the Internet
?";
TITLE = "Please select everything that applies to you!";
LABELS =
1 "At home"
2 "At work/ training place"
3 "At school/ university"
4 "In a public place (internet cafe, multimedia terminal, ...)"
5 "On the road (via WLAN hotspot, cellphone, ...)"
6 "Outside of the house (with friends, relatives)"
99 "None of that"
;
```

The description of a matrix question can not simply be taken 1 by 1 from the questionnaire script. Instead, a macro is created here and automatically integrated into the `var.inc`.

```
#MACRO #f10.1(&gItemVal &title &col)
SingleQ f10.&gItemVal.1 = &col 1;
TEXT = "How often do you use these services on the Internet?";
TITLE = "&title";
LABELS =
1 "very frequently"
2 "frequently"
3 "sometimes"
4 "rarely"
5 "very rarely"
;
#ENDMACRO

#f10.1(1 "Rechearch/ advisor" 520)
#f10.1(2 "Videos/ music" *)
```

```
#f10.1(3 "E-mail " *)
#f10.1(4 "Online gaming " *)
#f10.1(5 "Shopping/ booking " *)
#f10.1(6 "Videos/ music" *)
#f10.1(7 "Chats & forums" *)
#f10.1(8 "Online banking " *)
#f10.1(9 "Online media (magazines, blogs)" *)
#f10.1(10 "Social networks (Facebook, Parship, XING)" *)
```

The first section defines the question structure. Placeholders are set for the item values, title, and the start column. In the second section, the macro is called up for each question item, each is stored as a separate variable. Values for item value, title and start column are set. From the second macro call, an asterisk is set instead of the start column. This causes a simple counting up of the column indication: Item 1 lies on column 520, for item 2 is counted up with 521, item 3 lies in column 522, etc ..

2.2 SPSS

SPSS records are included in the script with the keyword `SPSSINFILE`:

```
SPSSINFILE = "filename.sav";
```

SPSS records also list one case per row. The program can store variable texts and labels. Ideally, you are working with a SPSS record that is nicely populated with texts and label texts, and you can start tabulating right away. Unfortunately, this is rarely the case.

Overview of an SPSS record

GESStabs provides several ways to get an overview of a given record.

The keyword `SYNTAX` can be used to write various variable information from the record into text files, namely the variable types, text, title and labels. To extract the properties from the dataset, the respective keyword is readjusted to the `SYNTAX` command.
It is a matter of taste whether you write all informations in a single text file:

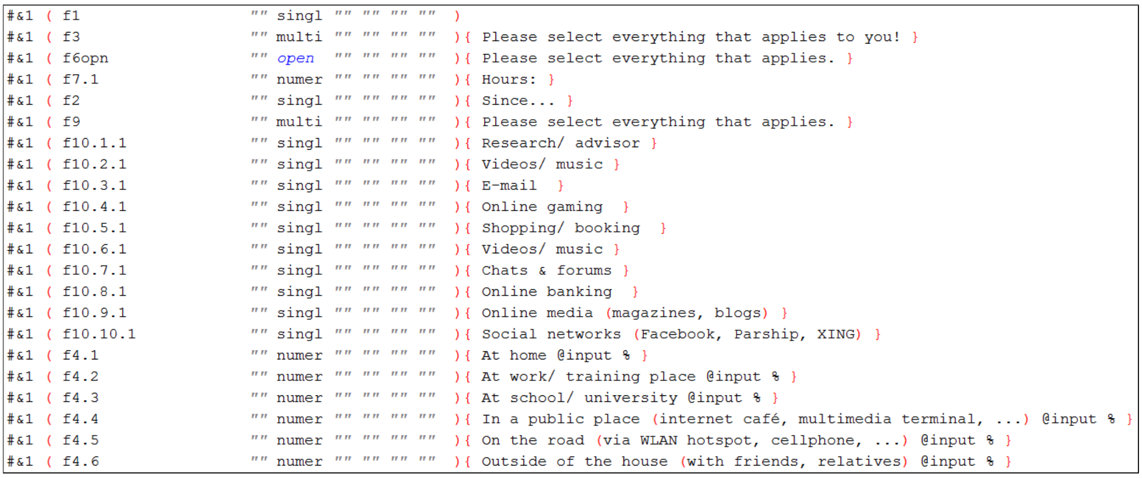
```
SYNTAX VARIABLES TITLE TEXT LABELS = everything.inc;
```

or thematically divided:

```
SYNTAX VARIABLES  = variablen.inc;
SYNTAX TITLE      = title.inc;
SYNTAX TEXT       = texte.inc;
SYNTAX LABELS     = labels.inc;
```

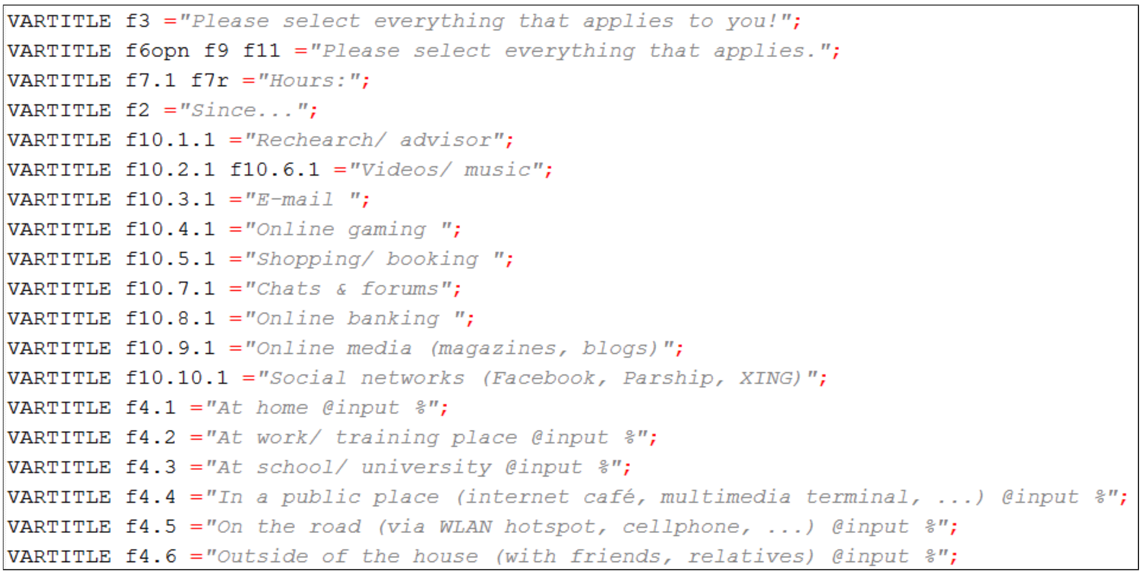
It should be mentioned here: SPSS does not differentiate between texts and titles, but instead performs simple variable labels. GESStabs handles them as variable titles by default.

The created files now contain the information about types, titles and Labels of the record´s variables:



```
#&1 ( f1                 "" singl "" "" "" ""  )
#&1 ( f3                 "" multi "" "" "" ""  ){ Please select everything that applies to you! }
#&1 ( f6opn              "" open  "" "" "" ""  ){ Please select everything that applies. }
#&1 ( f7.1               "" numer "" "" "" ""  ){ Hours: }
#&1 ( f2                 "" singl "" "" "" ""  ){ Since... }
#&1 ( f9                 "" multi "" "" "" ""  ){ Please select everything that applies. }
#&1 ( f10.1.1            "" singl "" "" "" ""  ){ Research/ advisor }
#&1 ( f10.2.1            "" singl "" "" "" ""  ){ Videos/ music }
#&1 ( f10.3.1            "" singl "" "" "" ""  ){ E-mail  }
#&1 ( f10.4.1            "" singl "" "" "" ""  ){ Online gaming  }
#&1 ( f10.5.1            "" singl "" "" "" ""  ){ Shopping/ booking  }
#&1 ( f10.6.1            "" singl "" "" "" ""  ){ Videos/ music }
#&1 ( f10.7.1            "" singl "" "" "" ""  ){ Chats & forums }
#&1 ( f10.8.1            "" singl "" "" "" ""  ){ Online banking  }
#&1 ( f10.9.1            "" singl "" "" "" ""  ){ Online media (magazines, blogs) }
#&1 ( f10.10.1           "" singl "" "" "" ""  ){ Social networks (Facebook, Parship, XING) }
#&1 ( f4.1               "" numer "" "" "" ""  ){ At home @input % }
#&1 ( f4.2               "" numer "" "" "" ""  ){ At work/ training place @input % }
#&1 ( f4.3               "" numer "" "" "" ""  ){ At school/ university @input % }
#&1 ( f4.4               "" numer "" "" "" ""  ){ In a public place (internet café, multimedia terminal, ...) @input % }
#&1 ( f4.5               "" numer "" "" "" ""  ){ On the road (via WLAN hotspot, cellphone, ...) @input % }
#&1 ( f4.6               "" numer "" "" "" ""  ){ Outside of the house (with friends, relatives) @input % }
```

Abb. 2.1: Variable names the SPSS record



```
VARTITLE f3 ="Please select everything that applies to you!";
VARTITLE f6opn f9 f11 ="Please select everything that applies.";
VARTITLE f7.1 f7r ="Hours:";
VARTITLE f2 ="Since...";
VARTITLE f10.1.1 ="Rechearch/ advisor";
VARTITLE f10.2.1 f10.6.1 ="Videos/ music";
VARTITLE f10.3.1 ="E-mail ";
VARTITLE f10.4.1 ="Online gaming ";
VARTITLE f10.5.1 ="Shopping/ booking ";
VARTITLE f10.7.1 ="Chats & forums";
VARTITLE f10.8.1 ="Online banking ";
VARTITLE f10.9.1 ="Online media (magazines, blogs)";
VARTITLE f10.10.1 ="Social networks (Facebook, Parship, XING)";
VARTITLE f4.1 ="At home @input %";
VARTITLE f4.2 ="At work/ training place @input %";
VARTITLE f4.3 ="At school/ university @input %";
VARTITLE f4.4 ="In a public place (internet café, multimedia terminal, ...) @input %";
VARTITLE f4.5 ="On the road (via WLAN hotspot, cellphone, ...) @input %";
VARTITLE f4.6 ="Outside of the house (with friends, relatives) @input %";
```

Abb. 2.2: Vartitles from SPSS record

Another way to get an overview of a given SPSS record is through the `CODEBOOK` statement.

This causes a simple frequency count over all

```
CODEBOOK;
```

or selected variables

```
CODEBOOK f1;
```

and looks like this:

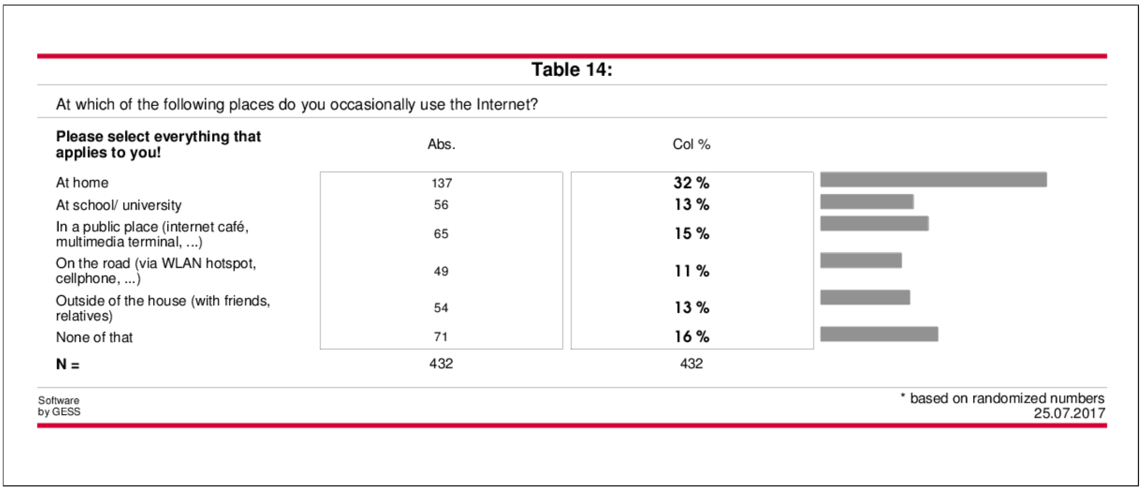


Table 14:

At which of the following places do you occasionally use the Internet?

Please select everything that applies to you!	Abs.	Col %
At home	137	**32 %**
At school/ university	56	**13 %**
In a public place (internet café, multimedia terminal, ...)	65	**15 %**
On the road (via WLAN hotspot, cellphone, ...)	49	**11 %**
Outside of the house (with friends, relatives)	54	**13 %**
None of that	71	**16 %**
N =	432	432

Software by GESS

* based on randomized numbers
25.07.2017

Abb. 2.3: Frequency counting

Tip: Codebooks are also great for checking your subsequently created data tables.

2.3 CSV

`CSVINFILE` includes CSV records into a GESStabs project:

```
CSVINFILE = "filename.csv";
```

Structurally, CSV records are similar to SPSS files: each row contains an interview case.

CSV data records do include raw data and variable names, but no further variable informations. You can define these later, please see Manual preperation of a var.inc.

To ensure the correct representation of open texts (and especially umlauts) in the table output, we recommend setting the encoding of CSV data files to UTF-8 by default:

```
ENCODING CSV = UTF8;
```

3 Simple cross table of existing variables

The simplest thing is to tabulate two existing variables in a crosstable. For this we define a table that contains one variable in the header and one in the page drilldown. We would like to look at the distribution of question f1 (place of internet use) by gender (variable s3), so we write the following statement in the `tab.inc`:

```
TABLE = s3 BY f1;
```

This is the simplest form of a GESStabs table statement. The keyword `TABLE` and an equals sign are followed by the header variable. `BY` is followed by the variable, which should be positioned in the pre column[1]. The latter is the actually interesting variable whose distribution is considered. As always, the instruction is completed by a semicolon.

For tabulation we call the GESStabs menu, the following minimalist program interface opens:

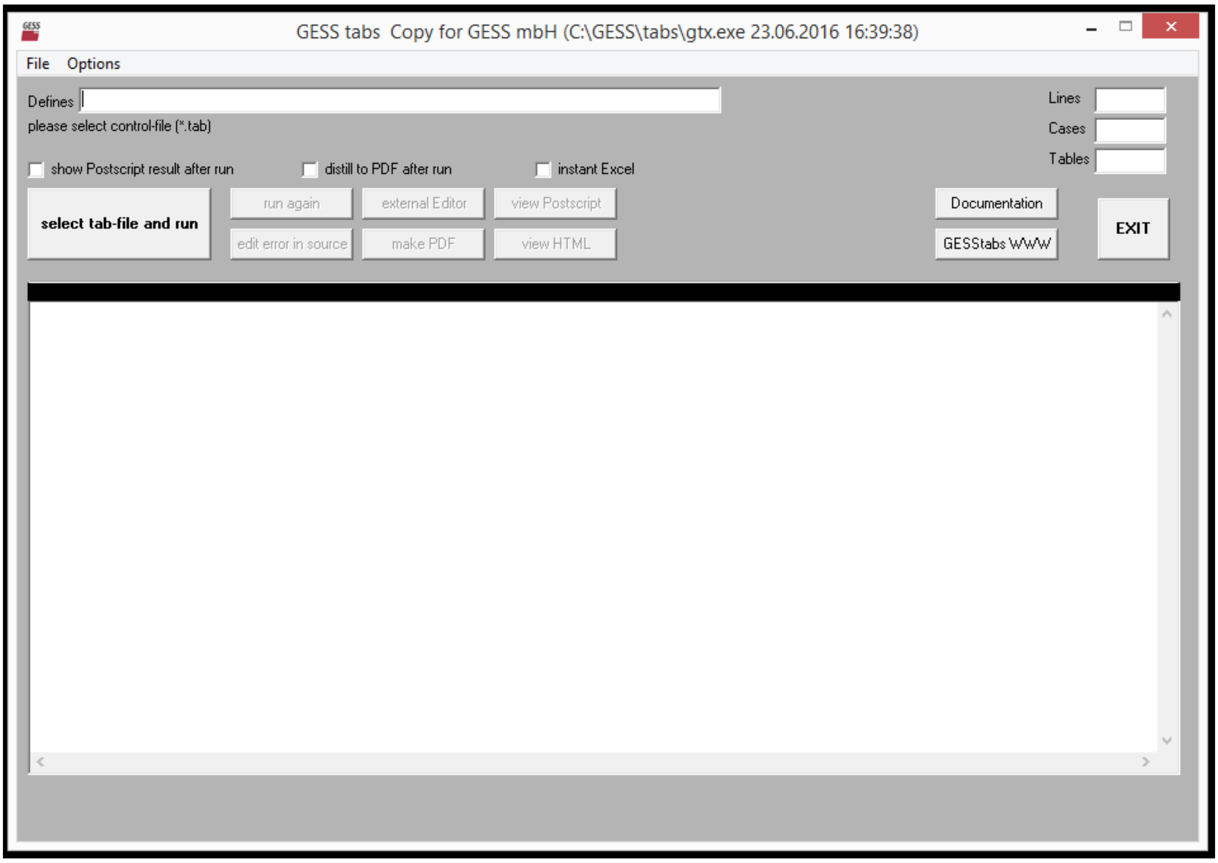
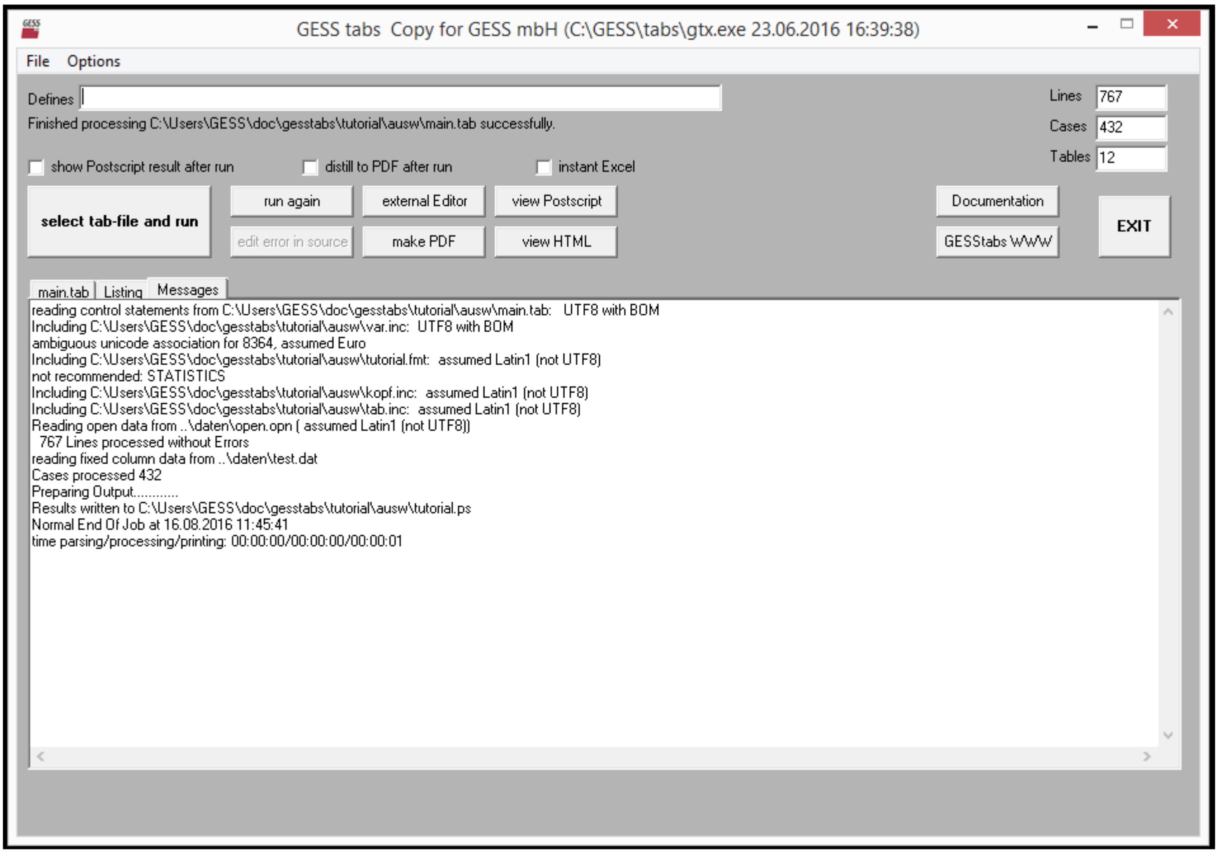


Abb. 3.1: The GESStabs surface

---

[1] The terms for this variable vary: Pre column, Outline, Y-variable ...

With the button select tab-file and run we call our main.tab. GESStabs executes the script and displays the following messages:



Defines

Finished processing C:\Users\GESS\doc\gesstabs\tutorial\ausw\main.tab successfully.

Lines 767
Cases 432
Tables 12

☐ show Postscript result after run ☐ distill to PDF after run ☐ instant Excel

```
reading control statements from C:\Users\GESS\doc\gesstabs\tutorial\ausw\main.tab:   UTF8 with BOM
Including C:\Users\GESS\doc\gesstabs\tutorial\ausw\var.inc:  UTF8 with BOM
ambiguous unicode association for 8364, assumed Euro
Including C:\Users\GESS\doc\gesstabs\tutorial\ausw\tutorial.fmt:  assumed Latin1 (not UTF8)
not recommended: STATISTICS
Including C:\Users\GESS\doc\gesstabs\tutorial\ausw\kopf.inc:  assumed Latin1 (not UTF8)
Including C:\Users\GESS\doc\gesstabs\tutorial\ausw\tab.inc:  assumed Latin1 (not UTF8)
Reading open data from ..\daten\open.opn ( assumed Latin1 (not UTF8))
  767 Lines processed without Errors
reading fixed column data from ..\daten\test.dat
Cases processed 432
Preparing Output...........
Results written to C:\Users\GESS\doc\gesstabs\tutorial\ausw\tutorial.ps
Normal End Of Job at 16.08.2016 11:45:41
time parsing/processing/printing: 00:00:00/00:00:00/00:00:01
```

Abb. 3.2: Creating tables

Our sample program has been processed successfully. The number of processed script lines, included data cases and created tables can be seen in the upper right corner.

Click on the button view PostScript to open the created table in your PostScript Viewer:

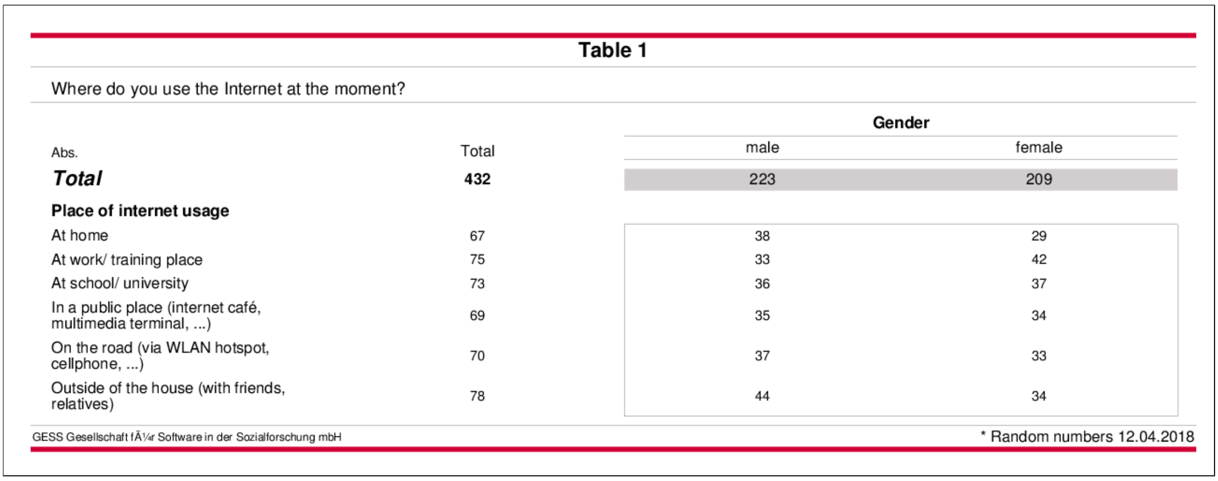


Table 1

Where do you use the Internet at the moment?

Abs.	Total	Gender: male	Gender: female
Total	**432**	223	209
Place of internet usage			
At home	67	38	29
At work/ training place	75	33	42
At school/ university	73	36	37
In a public place (internet café, multimedia terminal, ...)	69	35	34
On the road (via WLAN hotspot, cellphone, ...)	70	37	33
Outside of the house (with friends, relatives)	78	44	34

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 3.3: Simple cross table

The result is a simple crosstable. The text of the pre-column variable is displayed above the table, and the respective title is displayed above the labels of each variable. The table shows absolute values (see caption 'Abs.'), it contains a total column and total row.

Clicking on make PDF would create an additional PDF file, identical to the PostScript file. For this moment, we leave it with the PostScript file and leave it open. After restarting the GESStabs run (click on run again) it automatically refreshes, allowing you to quickly review changes and additions made.

4 Cross table with several header variables

A table with only two columns in the header looks a little bit stodgy. We therefore decide to adopt more variables to the header. This is quite simple: add the corresponding variable names to the `TABLE`-statement.

```
TABLE = s1 s2 s3 BY f1;
```

These variables (device usage, main residence & gender) are supposed to form our new header.

Because this header will often be used, it is functional to arrange a generalized name for it. For this purpose, a universal mechanism of the GESStabs scripting language is offered: `#EXPAND`. It is a simple text replacement.

We call our table head `#K1`[1] and list all variables that should be included:

```
#EXPAND #K1 s1 s2 s3
```

As a simple text replacement, the complete line following the keyword `#EXPAND` is subsequently transferred, including comment mark, semicolon, etc. Thus, there must not be a terminating semicolon here, as this would end the `TABLE` command after '#K1'.

The table request for the identical table is then shortened to:

```
TABLE = #K1 BY f1;
```

Now you can go ahead and tabulate the same question with other headers. Often you want to look at more crossed variables than would fit on a spreadsheet page. Therefore, we create a second table header that contains questions about income and education.

```
#EXPAND #K2 s8 s9
```

Effective scripting requires clarity. Therefore, we create the table headers in a separate file, the `head.inc` mentioned earlier. In our example, this file contains only these two lines:

```
#EXPAND #K1 s1 s3 s2
#EXPAND #K2 s8 s9
```

Since our example is still quite clear (because short), this division may seem too strict. In practice, however, one often works with numerous and complex table headers. Other programmers (or even yourself, half a year later) find it easier to cope with your script using these kinds of shared formalisms.

Then we tabulate our question f1 with both headers:

```
TABLE = #K1 BY f1;
TABLE = #K2 BY f1;
```

---

[1] Names of `EXPAND`s must always start with a cross of a lath.

As a result, we get the following two tables:

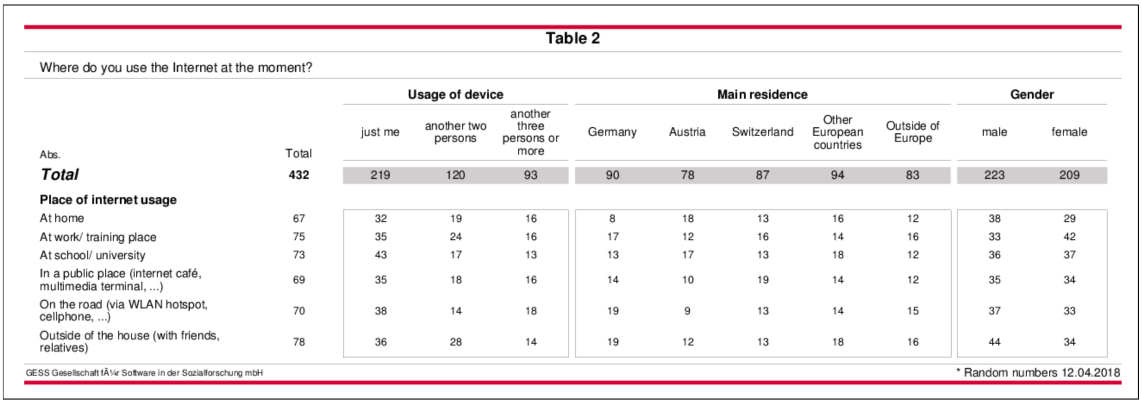


Table 2

Where do you use the Internet at the moment?

		Usage of device			Main residence					Gender	
Abs.	Total	just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Place of internet usage											
At home	67	32	19	16	8	18	13	16	12	38	29
At work/ training place	75	35	24	16	17	12	16	14	16	33	42
At school/ university	73	43	17	13	13	17	13	18	12	36	37
In a public place (internet café, multimedia terminal, ...)	69	35	18	16	14	10	19	14	12	35	34
On the road (via WLAN hotspot, cellphone, ...)	70	38	14	18	19	9	13	14	15	37	33
Outside of the house (with friends, relatives)	78	36	28	14	19	12	13	18	16	44	34

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 4.1: Table header with text replacement

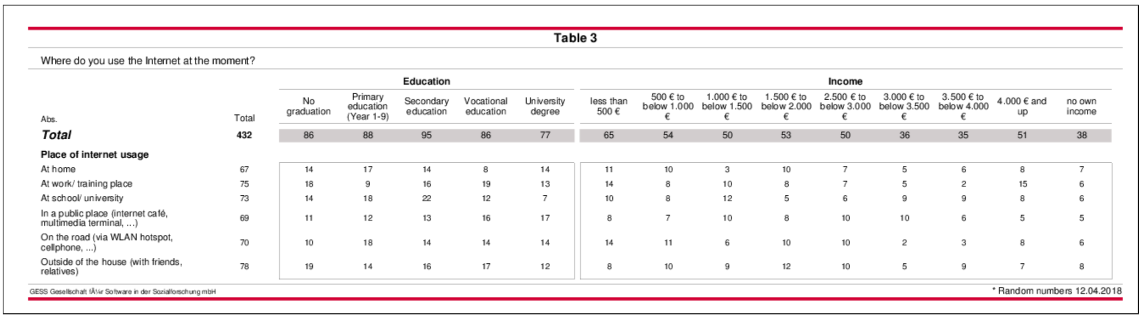


Table 3

Where do you use the Internet at the moment?

		Education					Income								
Abs.	Total	No graduation	Primary education (Year 1-9)	Secondary education	Vocational education	University degree	less than 500 €	500 € to below 1.000 €	1.000 € to below 1.500 €	1.500 € to below 2.000 €	2.500 € to below 3.000 €	3.000 € to below 3.500 €	3.500 € to below 4.000 €	4.000 € and up	no own income
Total	**432**	86	88	95	86	77	65	54	50	53	50	36	35	51	38
Place of internet usage															
At home	67	14	17	14	8	14	11	10	3	10	7	5	6	8	7
At work/ training place	75	18	9	16	19	13	14	8	10	8	7	5	2	15	6
At school/ university	73	14	18	22	12	7	10	8	12	5	6	9	9	8	6
In a public place (internet café, multimedia terminal, ...)	69	11	12	13	16	17	8	7	10	8	10	10	6	5	5
On the road (via WLAN hotspot, cellphone, ...)	70	10	18	14	14	14	14	11	6	10	10	2	3	8	6
Outside of the house (with friends, relatives)	78	19	14	16	17	12	8	10	9	12	10	5	9	7	8

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 4.2: Cross table with several headers

5 Editing tables

5.1 Groups

Now, the second header is kind of long and the table is therefore not very legible, which is mainly due to the numerous labels of variable s9. It makes sense to summarize the income classes[1]. For this we generate a new variable, more precisely a `GROUPS`. With a `GROUPS` labels can be summarized, new label texts and titles assigned and/ or variables filtered.

```
GROUPS s9new =
| "No income"                  :    10 in s9
| "1 € to below 1.000 €"       : [1:2] in s9
| "1.000 € to below 2.500 €"   : [3:5] in s9
| "2.500 € to below 3.500 €"   : [6:7] in s9
| "3.500 € and up"             : [8:9] in s9
;
TITLE s9new ="Income (new)";
```

The label definitions of a `GROUPS` always start with a |, followed by the label text. After the colon, there is defined which values of s9 should be included in the respective label[2]. Including the newly generated `GROUPS`, a new table header is generated:

```
#EXPAND #K3 s8 s9neu
```

The table and according statement look like this:

```
TABLE = #K3 BY f1;
```

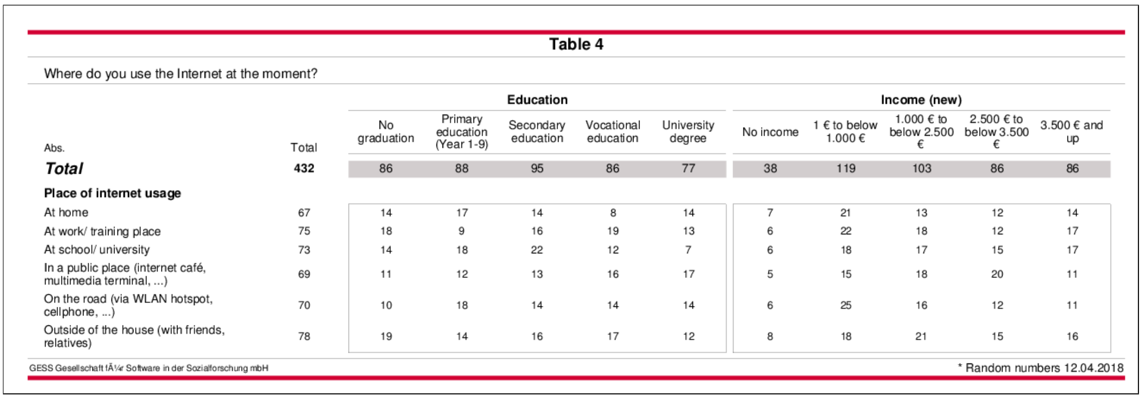


Table 4

Where do you use the Internet at the moment?

Abs.	Total	Education: No graduation	Education: Primary education (Year 1-9)	Education: Secondary education	Education: Vocational education	Education: University degree	Income (new): No income	Income (new): 1 € to below 1.000 €	Income (new): 1.000 € to below 2.500 €	Income (new): 2.500 € to below 3.500 €	Income (new): 3.500 € and up
Total	**432**	86	88	95	86	77	38	119	103	86	86
Place of internet usage											
At home	67	14	17	14	8	14	7	21	13	12	14
At work/ training place	75	18	9	16	19	13	6	22	18	12	17
At school/ university	73	14	18	22	12	7	6	18	17	15	17
In a public place (internet café, multimedia terminal, ...)	69	11	12	13	16	17	5	15	18	20	11
On the road (via WLAN hotspot, cellphone, ...)	70	10	18	14	14	14	6	25	16	12	11
Outside of the house (with friends, relatives)	78	19	14	16	17	12	8	18	21	15	16

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 5.1: Customizing the header variables

5.2 Overcodes

The `LABELS` statement makes it easy to add, sort, and edit labels of existing variables.

Let´s have a look at variable f11_1:

```
SingleQ f11_1 = 713 2;
TEXT = "And which topic do you research most frequently?";
```

---

[1] Following our script system, we perform variable manipulations in `compute.inc`.

[2] Make sure to define the labels neatly. If the conditions overlap, the values of one case may be included in several labels of the Groups.

```
LABELS =
1 "News"
2 "Family and kids"
3 "Business/ finances"
4 "Politics"
5 "Sports"
6 "Food and drink"
7 "Eroticism"
8 "Cinema and TV program"
9 "Weather"
10 "Beauty and fashion"
11 "Friends/ partners"
12 "Art/ culture/ music"
;
```

The topics here are rather jumbled and not ordered by overpoints. With GESStabs we can easily determine the order of the labels ourselves.

```
LABELS f11_1 =
1 "News"
4 "Politics"
9 "Weather"
3 "Business/ finances"

10 "Beauty and fashion"
6 "Food and drink"
8 "Cinema and TV program"
12 "Art/ culture/ music"
5 "Sports"

7 "Eroticism"
2 "Family and kids"
11 "Friends/ partners"
;
```

In the printed table, the topics would appear in that order. Now it is often desirable to formulate top topics and have them counted out. To do this, you just have to tell the labels the parent terms as `OVERCODE`.

```
LABELS f11_1 =
OVERCODE 1 3 4 9 "Public (net count)"
1 "News"
4 "Politics"
9 "Weather"
3 "Business/ finances"
Overcode 5 6 8 10 12 "\Lifestyle (net count)"
10 "Beauty and fashion"
6 "Food and drink"
8 "Cinema and TV program"
12 "Art/ culture/ music"
5 "Sports"
Overcode 2 7 11 "\Private (net count)"
7 "Eroticism"
```

```
2 "Family and kids"
11 "Friends/ partners"
;
```

If the keyword `SUM` was added to `OVERCODE`, a gross count would be made.

With #K1 in the header, the table of the new f11_1 looks like this:

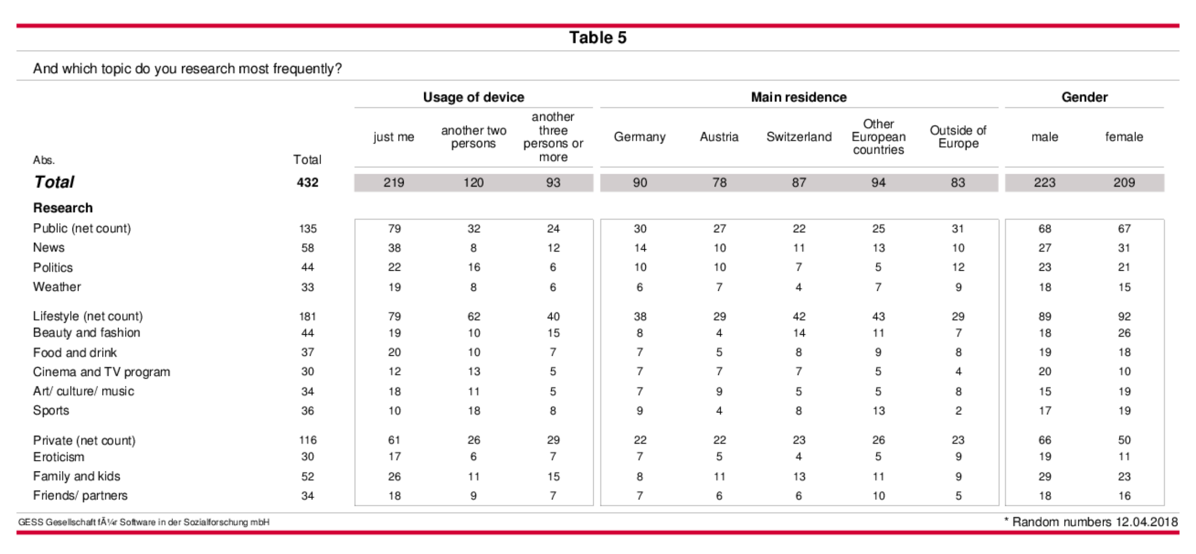


Table 5

And which topic do you research most frequently?

Abs.	Total	Usage of device			Main residence					Gender	
		just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Research											
Public (net count)	135	79	32	24	30	27	22	25	31	68	67
News	58	38	8	12	14	10	11	13	10	27	31
Politics	44	22	16	6	10	10	7	5	12	23	21
Weather	33	19	8	6	6	7	4	7	9	18	15
Lifestyle (net count)	181	79	62	40	38	29	42	43	29	89	92
Beauty and fashion	44	19	10	15	8	4	14	11	7	18	26
Food and drink	37	20	10	7	7	5	8	9	8	19	18
Cinema and TV program	30	12	13	5	7	7	7	5	4	20	10
Art/ culture/ music	34	18	11	5	7	9	5	5	8	15	19
Sports	36	10	18	8	9	4	8	13	2	17	19
Private (net count)	116	61	26	29	22	22	23	26	23	66	50
Eroticism	30	17	6	7	7	5	4	5	9	19	11
Family and kids	52	26	11	15	8	11	13	11	9	29	23
Friends/ partners	34	18	9	7	7	6	6	10	5	18	16

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 5.2: Overcodes

5.3 Top(2)- and Bottom(2)-Boxes

Often, scales should be displayed in addition to the frequency distribution as top- or bottom-boxes. Again, this can easily be done using `OVERCODEs`.

Here is an example of the formulation of overcodes in question 10.1.1:

```
SingleQ f10.1.1 = 520 1;
TEXT f10.1.1 = "How often do you use these services on the Internet?";
TITLE f10.1.1 = "Rechearch/ advisor";
LABELS f10.1.1 =
1 "very frequently"
2 "frequently"
3 "sometimes"
4 "rarely"
5 "very rarely"
OVERCODE 1 2 "Top-2-Box (1+2)"
OVERCODE 4 5 "Bottom-2-Box (4+5)"
;
```

... and its tabular representation:

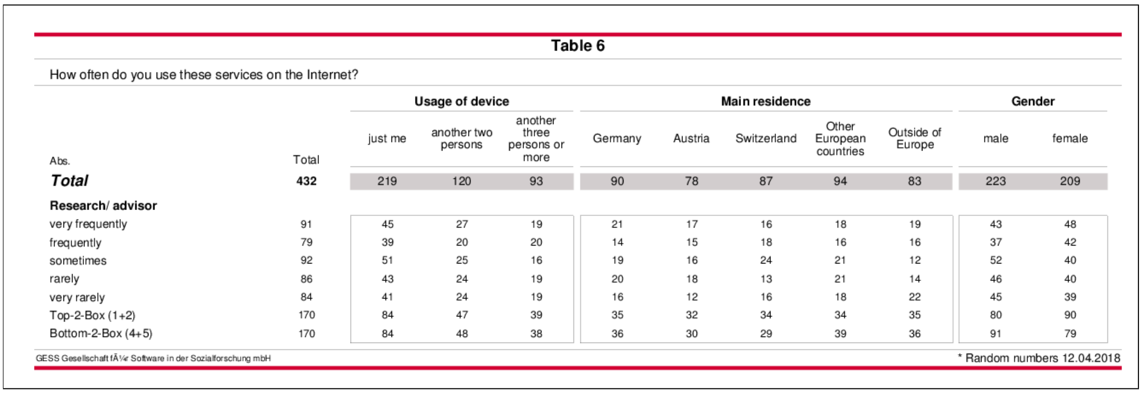


Table 6

How often do you use these services on the Internet?

Abs.	Total	Usage of device			Main residence					Gender	
		just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Research/ advisor											
very frequently	91	45	27	19	21	17	16	18	19	43	48
frequently	79	39	20	20	14	15	18	16	16	37	42
sometimes	92	51	25	16	19	16	24	21	12	52	40
rarely	86	43	24	19	20	18	13	21	14	46	40
very rarely	84	41	24	19	16	12	16	18	22	45	39
Top-2-Box (1+2)	170	84	47	39	35	32	34	34	35	80	90
Bottom-2-Box (4+5)	170	84	48	38	36	30	29	39	36	91	79

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 5.3: Table with overcodes

5.4 Vargroup

Datasets always contain item batteries whose top- and bottom-boxes should be summarized in one overview table. In GESStabs there is the variable type `VARGROUP`, which is well suited for this.

A `VARGROUP` is made up of several simple variables. A `VARGROUP` of n variables automatically has n labels that are calculated according to defined conditions.

This is easier in practice than in theory. Let´s take another look at a matrix variable from our sample data set. It consists of ten items that ask for the frequency of use of certain Internet offers (f10). This item battery is defined as follows:

```
#MACRO #f10.1(&gItemVal &title &col)
SingleQ f10.&gItemVal.1 = &col 1;
TEXT = "How often do you use these services on the Internet?";
TITLE = "&title";
LABELS =
1 "very frequently"
2 "frequently"
3 "sometimes"
4 "rarely"
5 "very rarely"
;
#ENDMACRO

#f10.1(1 "Rechearch/ advisor" 520)
#f10.1(2 "Videos/ music" *)
#f10.1(3 "E-mail " *)
#f10.1(4 "Online gaming " *)
#f10.1(5 "Shopping/ booking " *)
#f10.1(6 "Videos/ music" *)
#f10.1(7 "Chats & forums" *)
#f10.1(8 "Online banking " *)
#f10.1(9 "Online media (magazines, blogs)" *)
#f10.1(10 "Social networks (Facebook, Parship, XING)" *)
```

Now, we want to create an overview of the top-2- and bottom-2-boxes of all items. For this we include all the individual variables (these correspond to the individual items of the question complex f10) and then define the calculation of the top-2- and bottom-2-boxes:

```
VARGROUP top2 = (f10.1.1 f10.2.1 f10.3.1 f10.4.1 f10.5.1 f10.6.1 f10.7.1 f10
.8.1 f10.9.1 f10.10.1) eq 1 2;
VARTITLE top2 = "Top-2-Box \very frequently + frequently";

VARGROUP bot2 = (f10.1.1 f10.2.1 f10.3.1 f10.4.1 f10.5.1 f10.6.1 f10.7.1 f10
.8.1 f10.9.1 f10.10.1) eq 4 5;
VARTITLE bot2 = "Bottom-2-Box \very rarely + rarely";
```

A `VARGROUP` automatically translates the original `TITLE` of the variables as `LABELS` of the newly generated `VARGROUP`.

Let´s have a look at the tabulation:

```
TABLE = #K1 BY top2;
TABLE = #K1 BY bot2;
```

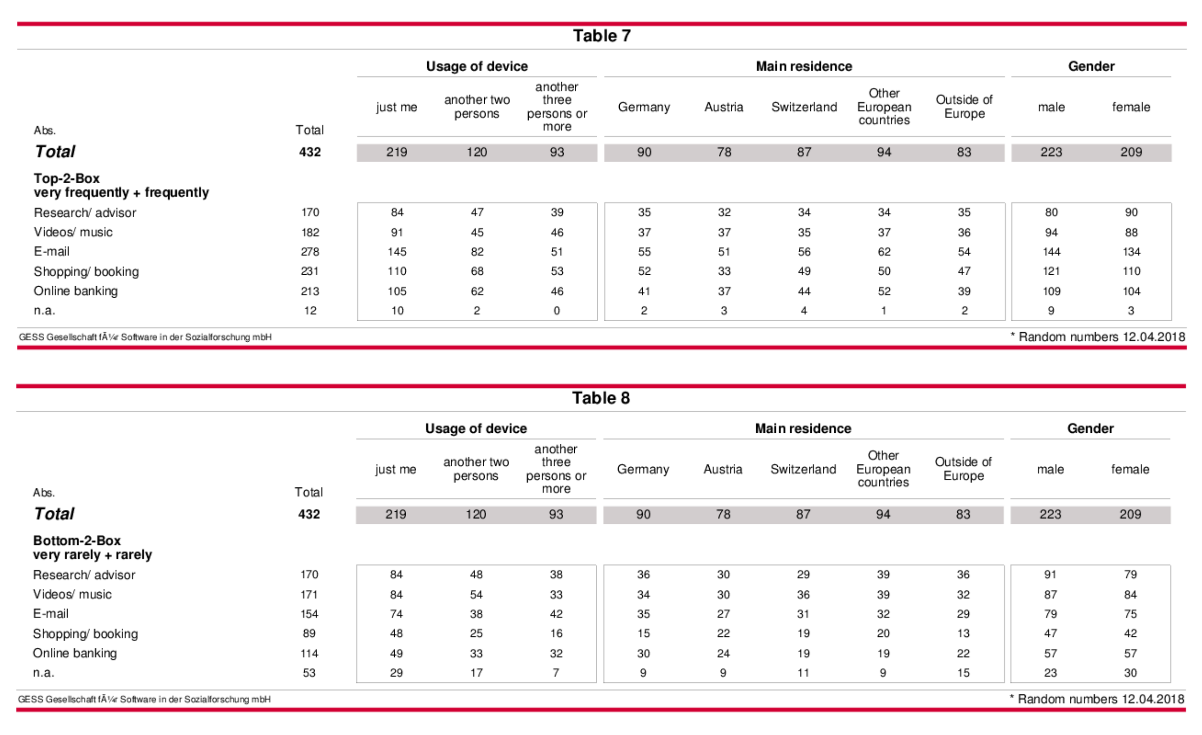


Table 7

		Usage of device			Main residence					Gender	
Abs.	Total	just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Top-2-Box very frequently + frequently											
Research/ advisor	170	84	47	39	35	32	34	34	35	80	90
Videos/ music	182	91	45	46	37	37	35	37	36	94	88
E-mail	278	145	82	51	55	51	56	62	54	144	134
Shopping/ booking	231	110	68	53	52	33	49	50	47	121	110
Online banking	213	105	62	46	41	37	44	52	39	109	104
n.a.	12	10	2	0	2	3	4	1	2	9	3

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Table 8

		Usage of device			Main residence					Gender	
Abs.	Total	just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Bottom-2-Box very rarely + rarely											
Research/ advisor	170	84	48	38	36	30	29	39	36	91	79
Videos/ music	171	84	54	33	34	30	36	39	32	87	84
E-mail	154	74	38	42	35	27	31	32	29	79	75
Shopping/ booking	89	48	25	16	15	22	19	20	13	47	42
Online banking	114	49	33	32	30	24	19	19	22	57	57
n.a.	53	29	17	7	9	9	11	9	15	23	30

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 5.4: Top-2-/ Bottom-2-Boxes

5.5 Mean values

As soon as a question has a scale, one usually wants to show a mean value as well. This can be done in GESStabs very simple, directly in the course of a tabulation.

As an example, let’s take the variable f10.1.1[3] again.

---

[3] See Top(2)- and Bottom(2)-Boxes.

We request a mean value in the `TABLE` statement by adding the keyword `MEAN` and the specification of the desired variable:

```
TABLE = #K1 BY f10.1.1 MEAN(f10.1.1);
```

Note: It is best to calculate means based only on basis of the valid answers and to exclude missing answers (often coded as -1 or 99) from the calculation. The latter would erroneously distort the mean. We have defined this in the first lines of our `main.tab`. All respondents who did not respond should automatically receive “k.A.“ and missing values should not be included in the calculation: `AUTONOANSWER = YES "k.A."; USEMISSING = no;`.

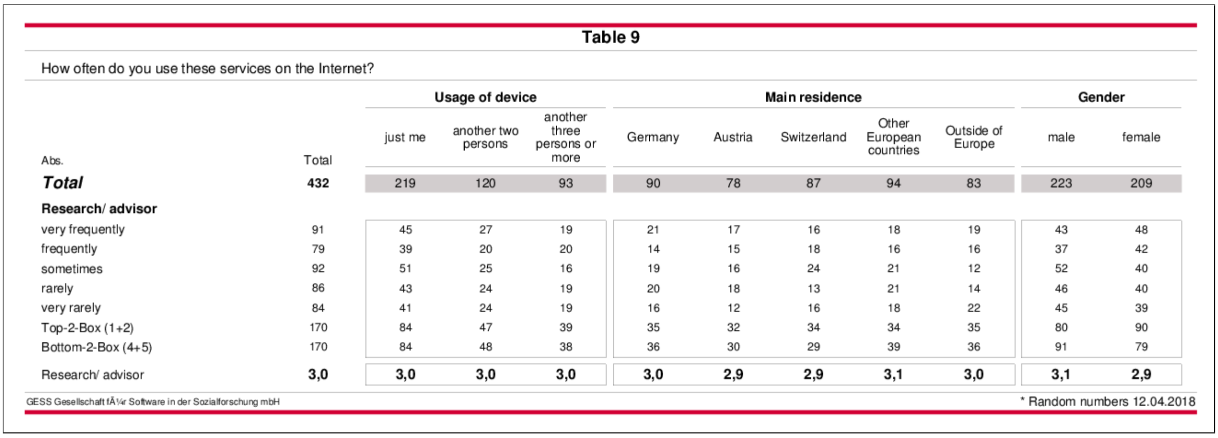


Table 9

How often do you use these services on the Internet?

Abs.	Total	Usage of device			Main residence					Gender	
		just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Research/ advisor											
very frequently	91	45	27	19	21	17	16	18	19	43	48
frequently	79	39	20	20	14	15	18	16	16	37	42
sometimes	92	51	25	16	19	16	24	21	12	52	40
rarely	86	43	24	19	20	18	13	21	14	46	40
very rarely	84	41	24	19	16	12	16	18	22	45	39
Top-2-Box (1+2)	170	84	47	39	35	32	34	34	35	80	90
Bottom-2-Box (4+5)	170	84	48	38	36	30	29	39	36	91	79
Research/ advisor	**3,0**	**3,0**	**3,0**	**3,0**	**3,0**	**2,9**	**2,9**	**3,1**	**3,0**	**3,1**	**2,9**

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 5.5: Mean values

Now the `TITLE` of the variable f10.1.1 is still displayed, describing the mean value line. It would be nice if there was simply “mean“, so we add the keyword `DESCRIPTION` to the tabulation statement:

```
TABLE = #K1 BY f10.1.1 MEAN :DESCRIPTION "MEAN"(f10.1.1);
```

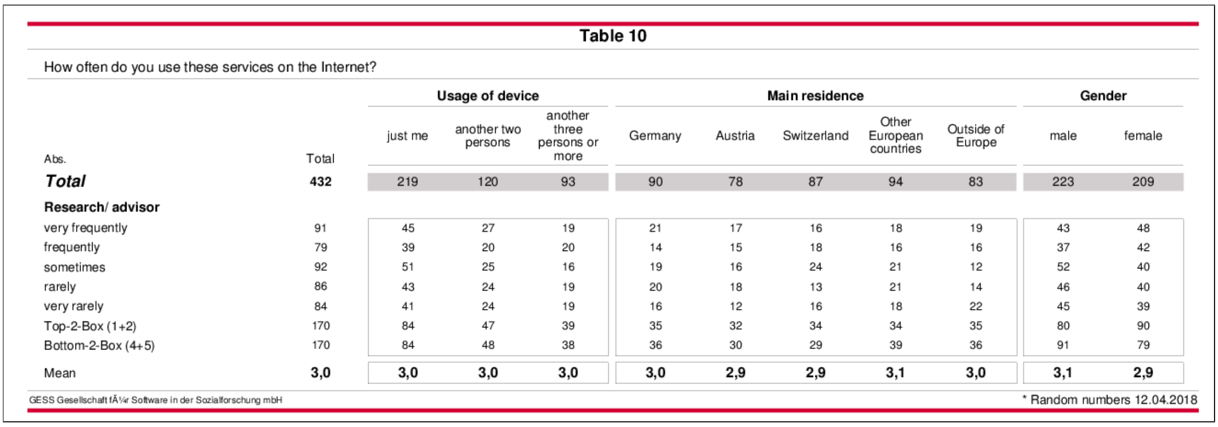


Table 10

How often do you use these services on the Internet?

Abs.	Total	Usage of device			Main residence					Gender	
		just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Research/ advisor											
very frequently	91	45	27	19	21	17	16	18	19	43	48
frequently	79	39	20	20	14	15	18	16	16	37	42
sometimes	92	51	25	16	19	16	24	21	12	52	40
rarely	86	43	24	19	20	18	13	21	14	46	40
very rarely	84	41	24	19	16	12	16	18	22	45	39
Top-2-Box (1+2)	170	84	47	39	35	32	34	34	35	80	90
Bottom-2-Box (4+5)	170	84	48	38	36	30	29	39	36	91	79
Mean	**3,0**	**3,0**	**3,0**	**3,0**	**3,0**	**2,9**	**2,9**	**3,1**	**3,0**	**3,1**	**2,9**

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 5.6: Mean values

Now, this could be continued for all following items of the question complex. As a result, we received ten table pages, each listing the scale distribution plus the mean value of each item. Not very handy, right?

In order to be able to compare the items at a glance, it makes sense to create a mean overview instead. That is quite easy:

```
TABLE = #K1 BY MEAN(f10.1.1) MEAN(f10.2.1) MEAN(f10.3.1) MEAN(f10.4.1) MEAN(f10
.5.1) MEAN(f10.6.1) MEAN(f10.7.1) MEAN(f10.8.1) MEAN(f10.9.1) MEAN(f10.10.1);
```

We removed the statement `:DESCRIPTION` and use the already assigned titles of the individual items. This is the result:

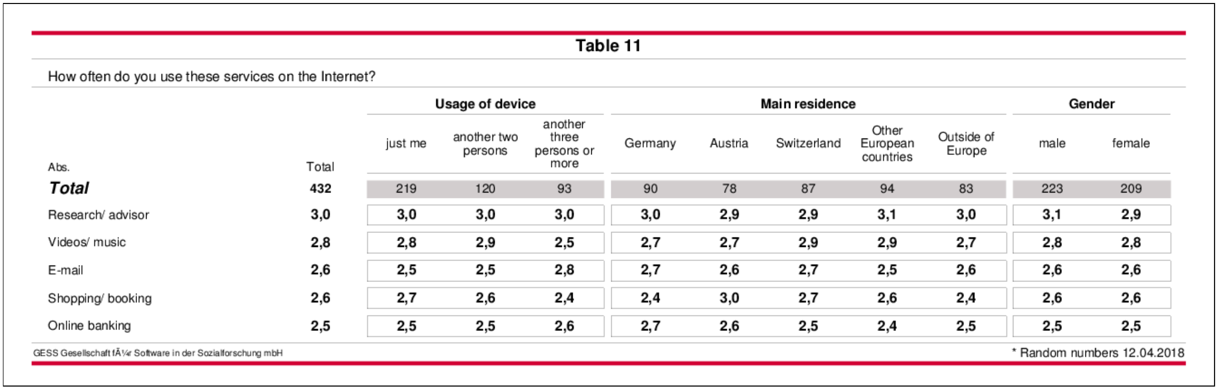


Table 11

How often do you use these services on the Internet?

		Usage of device			Main residence					Gender	
Abs.	Total	just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Research/ advisor	**3,0**	**3,0**	**3,0**	**3,0**	**3,0**	**2,9**	**2,9**	**3,1**	**3,0**	**3,1**	**2,9**
Videos/ music	**2,8**	**2,8**	**2,9**	**2,5**	**2,7**	**2,7**	**2,9**	**2,9**	**2,7**	**2,8**	**2,8**
E-mail	**2,6**	**2,5**	**2,5**	**2,8**	**2,7**	**2,6**	**2,7**	**2,5**	**2,6**	**2,6**	**2,6**
Shopping/ booking	**2,6**	**2,7**	**2,6**	**2,4**	**2,4**	**3,0**	**2,7**	**2,6**	**2,4**	**2,6**	**2,6**
Online banking	**2,5**	**2,5**	**2,5**	**2,6**	**2,7**	**2,6**	**2,5**	**2,4**	**2,5**	**2,5**	**2,5**

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 5.7: Mean value overview

5.6 Sort by frequency

Also for the sake of clarity, it is often appropriate to sort a tabulated variable according to the frequency of their entries. For example, we script this for question f9.
For a descending sorting of the absolute values (by total), the known tabulation statement only has to be supplemented with three keywords:

```
TABLE = #K1 BY f9 SORT ABSOLUTE DESCEND;
```

5.7 Recode

Frequently, surveys also contain numerical questions. In the tabulation, it usually makes no sense to list all the individual answers. Instead, the individual inputs of the respondents can be grouped into categories[4]. For this purpose GESStabs offers the command `RECODE`. A `RECODE` allows recoding individual variable values of the last defined variable or a list of explicitly named variables.

As an example we take the question f7.1. It is described in our automatically generated var.inc as followed:

```
#MACRO #f7(&p1, &title, &col, &width)
  SingleQ f7.&p1 = &col &width;
  TEXT = "How much time did you spend on the Internet yesterday?";
  TITLE = "&title";
#endmacro

#f7(1 "Hours:" 493 4)
```

---

[4] Of course, it should be noted that a categorization does not distort the original distribution of the data.

Only numbers between 0 and 24 (hours) have been allowed. Without a `RECODE`, every input made is mapped:

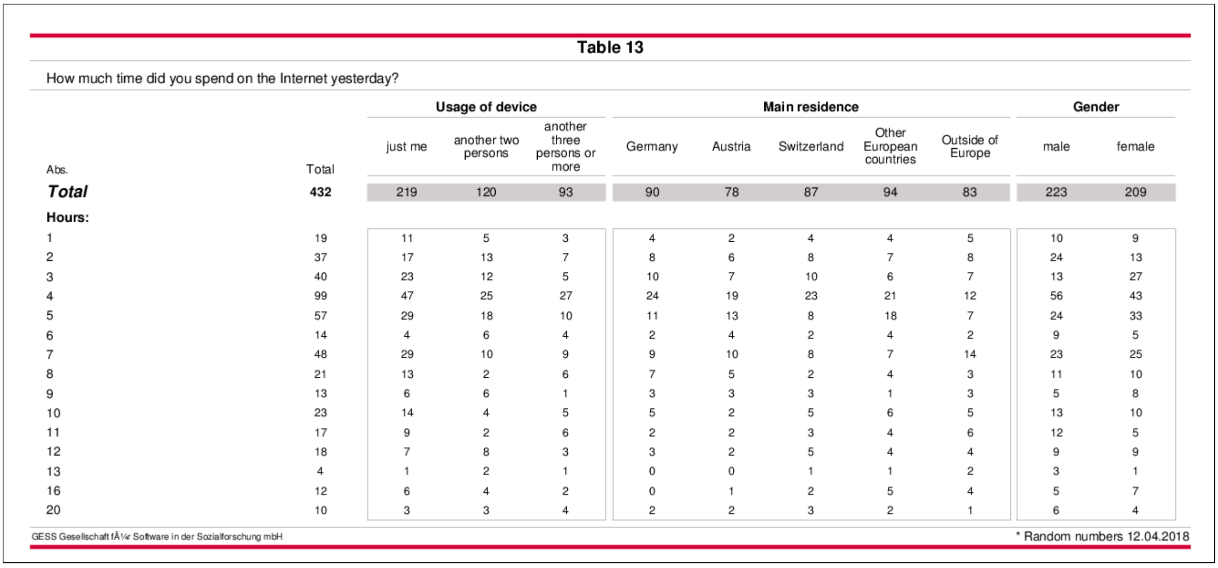


Table 13

How much time did you spend on the Internet yesterday?

		Usage of device			Main residence					Gender	
Abs.	Total	just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Hours:											
1	19	11	5	3	4	2	4	4	5	10	9
2	37	17	13	7	8	6	8	7	8	24	13
3	40	23	12	5	10	7	10	6	7	13	27
4	99	47	25	27	24	19	23	21	12	56	43
5	57	29	18	10	11	13	8	18	7	24	33
6	14	4	6	4	2	4	2	4	2	9	5
7	48	29	10	9	9	10	8	7	14	23	25
8	21	13	2	6	7	5	2	4	3	11	10
9	13	6	6	1	3	3	3	1	3	5	8
10	23	14	4	5	5	2	5	6	5	13	10
11	17	9	2	6	2	2	3	4	6	12	5
12	18	7	8	3	3	2	5	4	4	9	9
13	4	1	2	1	0	0	1	1	2	3	1
16	12	6	4	2	0	1	2	5	4	5	7
20	10	3	3	4	2	2	3	2	1	6	4

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 5.8: Before recoding

To group the individual values, we first create a new variable, more precisely an identical copy of the f7.1. Of course we could also edit the original variable directly, but then the original values were lost[5]. It is safer – and not more complex – to generate a new variable.

```
COMPUTE   f7r = f7.1;
COPYTEXT  f7r = f7.1;
COPYTITLE f7r = f7.1;
```

The command `COMPUTE` creates a new variable (f7r), the original values (of f7.1) are copied. `COPYTEXT` and `COPYTITLE` adapt the texts of the output variable and assign them to the new variable.

To create groups, assign the output values (0 to 24) to new labels:

```
RECODE f7r
0:5   = 1/
6:10  = 2/
11:15 = 3/
16:20 = 4/
21:24 = 5
;
```

Then we name the new labelcodes:

```
LABELS f7r =
1 "1 to 3 hours"
2 "4 to 6 hours"
3 "7 to 9 hours"
4 "10 to 15 hours"
5 "15 to 24 hours"
;
```

---

[5] And, as you will see, we´ll still need the original variable.

The distribution of the new variable looks like this:

```
TABLE = #K1 BY f7r MEAN(f7.1);
```

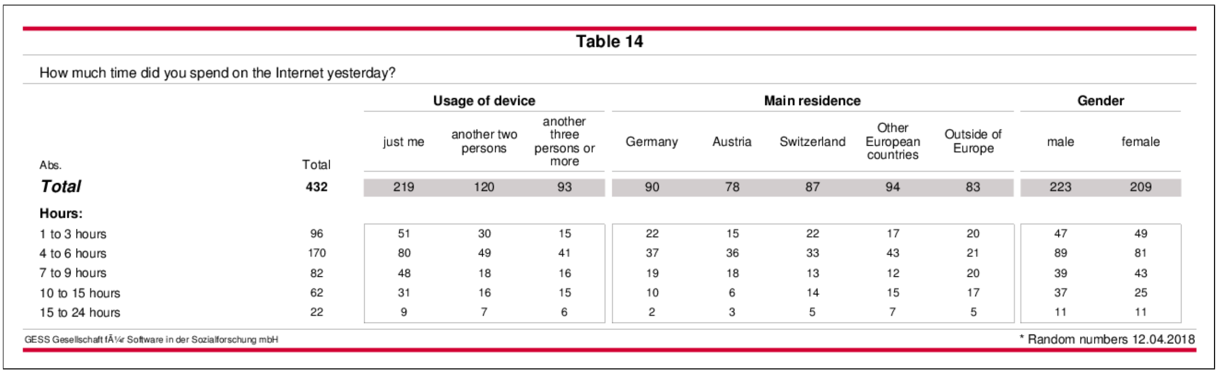


Table 14

How much time did you spend on the Internet yesterday?

Abs.	Total	Usage of device			Main residence					Gender	
		just me	another two persons	another three persons or more	Germany	Austria	Switzerland	Other European countries	Outside of Europe	male	female
Total	**432**	219	120	93	90	78	87	94	83	223	209
Hours:											
1 to 3 hours	96	51	30	15	22	15	22	17	20	47	49
4 to 6 hours	170	80	49	41	37	36	33	43	21	89	81
7 to 9 hours	82	48	18	16	19	18	13	12	20	39	43
10 to 15 hours	62	31	16	15	10	6	14	15	17	37	25
15 to 24 hours	22	9	7	6	2	3	5	7	5	11	11

GESS Gesellschaft fÃ¼r Software in der Sozialforschung mbH

* Random numbers 12.04.2018

Abb. 5.9: After recoding

A note on the `TABLE` statement: Of course, we calculate the mean value on basis of all individual entries from the original variable. A mean based on the grouped f7r would only use the label codes between 1 and 5 for the mean calculations – mathematically correct, wrong in content.

6 Table design

After we have devoted ourselves to the basics of the table content, we now take a look at an equally important part of the evaluation, namely the design of the table volume and the individual tables.

The font color and size and the color design as well as the insertion of logos and images, the positioning of the table on the sheet and setting individual distances – the entire table layout can be set individually with GESStabs syntax. At this point we can only go into the basics of the table design. If you would like to redesign the layout of your spreadsheet, we are happy to support you!

6.1 General formatting

Basically, we combine formatting instructions in a separate format file, here `tutorial.fmt`. All instructions are valid until redefined[1].

Some basic formatting instructions are the following:

```
NUMBERCHAR    = #;
SPLITCHAR     = ~;
SPLITCHARSTAY = -;
LINEFEEDCHAR  = \;
```

`NUMBERCHAR` is a numeric wildcard (see below). If this is used in page numbers for example, they will increment automatically.

`SPLITCHAR` and `SPLITCHARSTAY` specify delimiters: while the latter marks a persistent break, `SPLITCHAR` represents a conditional delimiter and marks the place where wrapped, if necessary[2].

`LINEFEEDCHAR` forces a line break.

Fundamental is `USEFONT`: This sets the font color and size. For a smooth GESStabs run we have to define this in general:

```
USEFONT = "Helvetica" SIZE 9;
```

In addition, `USEFONT` can be defined for individual table elements, e.g. `USEFONT ABSOLUTE = " Helvetica" SIZE 8;`, `USEFONT MEAN = "Helvetica-Bold" SIZE 10;` usw..

`FORMAT` determines the numeric formats of the different measures. If, for example, you want two decimals in mean values, you determine:

```
FORMAT MEAN = '#,##';
```

`ALIGN`, `DISTANCE` or `FRAMECOLOR` are other keywords that can be used to align box content, define distances or set frame colors. A complete formatting file is available in our sample project.

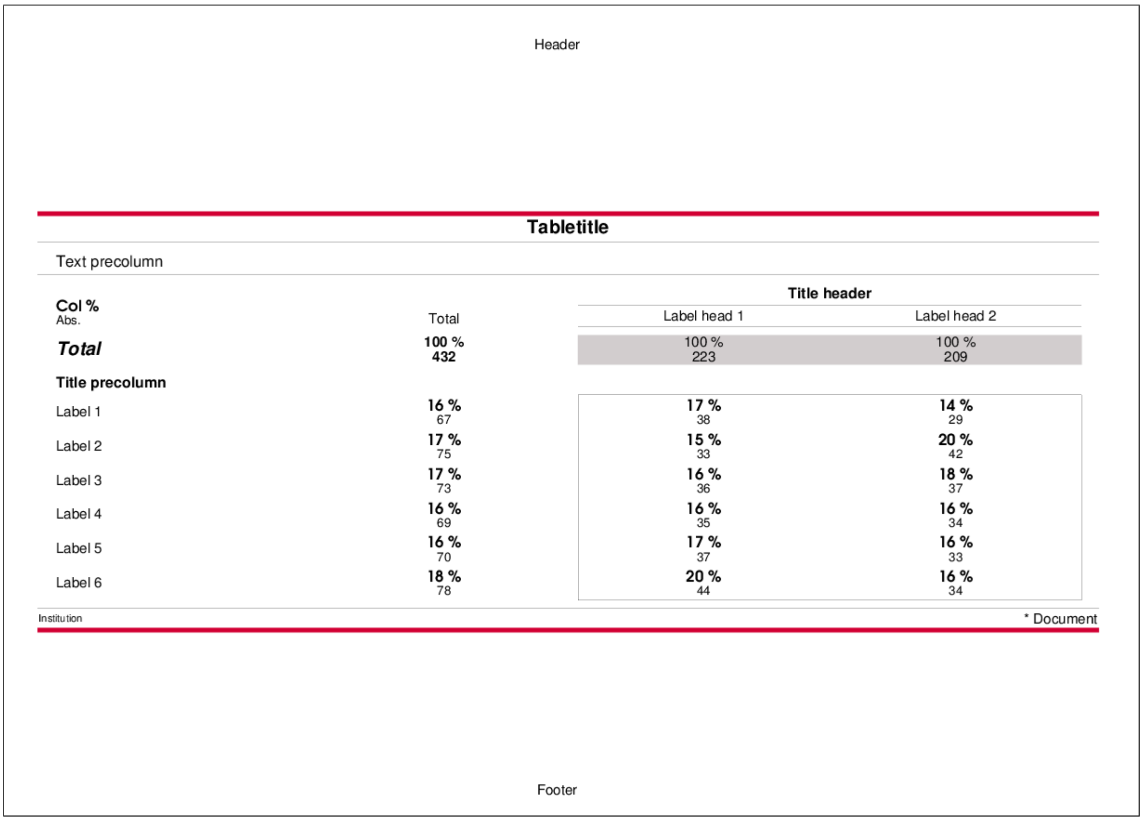
---

[1] At this point we would like to point out again the linear processing of the script by the software: Formatting commands apply to all subsequent script lines and until the formatting command is redefined. This then applies again for all subsequent lines etc ..

[2] Then, the character defined under `SPLITCHARSTAY` is used.

6.2 Table design

A table consists of different areas. Here is an overview of the structure of a typical GESStabs table:



Header

Tabletitle

Text precolumn

Col % Abs.	Total	**Title header** Label head 1	 Label head 2
Total	**100 %** **432**	100 % 223	100 % 209
Title precolumn			
Label 1	**16 %** 67	**17 %** 38	**14 %** 29
Label 2	**17 %** 75	**15 %** 33	**20 %** 42
Label 3	**17 %** 73	**16 %** 36	**18 %** 37
Label 4	**16 %** 69	**16 %** 35	**16 %** 34
Label 5	**16 %** 70	**17 %** 37	**16 %** 33
Label 6	**18 %** 78	**20 %** 44	**16 %** 34

Institution

* Document

Footer

Abb. 6.1: Overview table design

`HEADER` and `FOOTER` are usually specified for the entire table file, the latter usually contains the page number:

```
FOOTER = "-#-";
```

Also, `INSTITUTION` and `DOCUMENT` usually do not change within a table file. For example, the institution name and any other annotation can be noted here:

```
INSTITUTION = "GESS Gesellschaft für Software in der Sozialforschung mbH";
DOCUMENT    = "based on randomized numbers   \ "DATE;
```

The latter was supplemented with `DATE` – the current date of the table generation will be inserted.

The `TABLETITLE` usually contains the study name and/ or a thematic assignment. For example:

```
TABLETITLE = "Example project: GESStabs tutorial";
```

Below, there is the `TOPTEXT` of the column variable[3].

On the left there is a description of the header variable: if necessary, the `TITLE` is displayed here and the label codes or (if given) texts are listed. The header variable is listed in the horizontal direction according to the same scheme.

---

[3] Theoretically, something else can also be found here, see the corresponding setting in main.tab.

The presentation of the data within the table can be individually controlled.

The frame is made up of the so-called `FRAMEELEMENTS`. In our scheme there is a total row and total column. These elements are not automatically presented but are added with the following command:

```
FRAMEELEMENTS   = TOTALROW TOTALCOLUMN;
```

The content of the individual data cell, which metrics should be shown for the mapped data, is controlled by the command keyword `CELLELEMENTS`. Column percentages and absolutes are requested as follows:

```
CELLELEMENTS = COLUMNPERCENT ABSOLUTE;
```

As described in section 6.1, the formatting of each measure is determined by the keyword `FORMAT`, for example:

```
FORMAT COLUMNPERCENT = "`#,# \%"';
```

Finally, we can control how the cell content is described. This will then appear in the upper left of the table, below the question text. For example:

```
DESCRIPTION COLUMNPERCENT = "Col \%";
DESCRIPTION ABSOLUTE      = "Abs.";
```

Index