

Automatisierte Experiment-Auswertung mit `generate_excel_eval.R`

`generate_excel_eval` ist ein in R geschriebenes Script, das Datenfiles von Unipark standardisiert zusammenfasst und die Auswertung in einer Excel-Datei speichert. R muss auf dem Rechner installiert sein.

Zu jeder Daten-Datei muss ein spezifisches Konfigurationsskript (nachfolgend `config file` genannt) existieren, welches alle notwendigen Informationen zum Aufbau des Datenfiles beinhaltet.

0. Ausführung des Skripts

Ausführung in der Windows Eingabeaufforderung oder im UNIX-Terminal mit `config file`

Im Terminal (UNIX, MacOS) muss `generate_excel_eval.R` zunächst ausführbar gemacht werden, danach wird das Skript mit angehängtem `config file` aufgerufen:

```
chmod +x generate_excel_eval.R
```

```
./generate_excel_eval.R my_directory/config_file.R
```

In der Windows Eingabeaufforderung lautet der Befehl zum Starten des Skripts

```
Rscript generate_excel_eval.R my_directory/config_file.R
```

Beachte: Windows Pfade werden in R mit einem „\“ getrennt, nicht wie üblich mit einem „/“

Ausführung ohne direkte Übergabe des `config files`

Wird `generate_excel_eval` ohne Parameter gestartet, öffnet sich ein Dialogfenster zur Auswahl des `config files`. Dies erfordert allerdings die Installation des Pakets *radiant.data*

1. Anpassung des `config files`: Globale Variablen

Im `config file` müssen zunächst einige globale Einstellungen und die zugehörigen Variablen gesetzt werden:

<code>name_data_file:</code>	Zeichenkette
<code>name_excel_file:</code>	Zeichenkette
<code>gr_ID_name:</code>	Zeichenkette
<code>gr_IDs:</code>	numerischer Vektor
<code>excel_col_names:</code>	Liste von Zeichenketten

`name_data_file` beinhaltet Pfad und Name der Datendatei. Aktuell werden Datendateien im .sav-Format und im .dta-Format unterstützt. Beispiel:

```
name_data_file <- "directory/my_data.sav"
```

`name_excel_file` beinhaltet den Namen der zu erzeugenden Excel-Datei. Die Excel-Datei wird in demjenigen Verzeichnis gespeichert, in welchem das Datenfile liegt. Beispiel:

```
name_excel_file <- "my_excel_table.xlsx"
```

`gr_ID_name` ist der Name der Spalte, die die IDs der Experimentgruppen enthält. Beispiel:

```
gr_ID_name <- "expgroup_id"
```

gr_IDs ist der Vektor aller Gruppen-IDs. Besteht das Experiment aus N Kohorten, hat *gr_IDs* die Länge N. Der Vektor wird mit dem R-Befehl *c(...)* angelegt, für collection. Mit einem Doppelpunkt können Bereiche angegeben werden. Beispiele:

```
gr_IDs <- c(1, 2, 3, 4, 11, 12, 13, 14)
gr_IDs <- c(c(1:4), c(11:14)) #is the same as above
```

excel_col_names gibt alle Namen der Gruppen als Liste an. Diese werden in die Titelzeile der Excel-Tabelle geschrieben. Die Liste muss aus N Zeichenketten bestehen, wenn N die Zahl der Kohorten ist. Beispiel:

```
excel_col_names <- list("Treatment 1", "Treatment 2", "Treatment 3")
```

2. Anpassung des config files: Zusätzliche Variablen definieren mit *append_vars*

2.1. Gruppenspezifische Variablen zusammenführen mit *merge_exp_columns*

Syntax: *merge_exp_columns(varname, col_name_list)*

varname: Zeichenkette
col_name_list: geschachtelte Liste von Zeichenketten

varname gibt den Spaltennamen der neuen Variable an, in der die einzelnen Gruppenvariablen zusammengeführt werden. Leer- und Sonderzeichen sind nicht erlaubt.

col_name_list gibt die Struktur an, wie gruppenspezifische Spalten bzw. Variablen zusammengeführt werden. Die Liste muss so viele Einträge haben wie es Gruppen gibt; die Zuordnung wird durch *gr_IDs* festgelegt. Einzelne Gruppenvariablen können mit dem Symbol *NULL* ausgeschlossen werden. Besteht eine Experimentgruppe aus mehreren Untergruppen, können diese in einer geschachtelten Liste adressiert werden.

Beispiel:

Angenommen es gibt vier Experimentgruppen, in der jeweils eine Partei zu wählen ist. Die Variable *v_54* beinhaltet die Wahlentscheidung für die erste Gruppe, *v_78* für die zweite Gruppe. Gruppe 3 soll ignoriert werden und in Gruppe 4 erfolgte die Wahl in zwei disjunkten Variablen *v_76* und *v_84*. Dann können alle Abstimmungen in einer Variablen *parteiwahl* zusammengeführt werden:

```
merge_exp_columns("parteiwahl", list("v_54", "v_78", NULL,
                                     list("v_76", "v_84")))
```

2.2. Subgruppen-Variablen erstellen mit *add_subgroup*

Syntax: *add_subgroup(varname = logical expression, varname2 = expr2, ...)*

varname: Zeichenkette
logical expression: Term, der Wahrheitswerte TRUE / FALSE ausgibt
...: weitere Parameterpaare, um mehrere Subgruppen anzulegen

varname gibt den Spaltennamen der neu anzulegenden Variable an, die den Datensatz in zwei disjunkte Subgruppen trennt. Leer- und Sonderzeichen sind nicht erlaubt.

logical_expression bestimmt die Wahrheitswerte der Subgruppenvariable. Es können mehrere Terme durch logische Operatoren verbunden werden. Hierbei bezeichnet “|” das logische Oder, “&” das logische Und. Gleichheit wird mit “==” überprüft.

Beispiel:

Angenommen in einem Experiment werden auf einer Skala Zustimmungswerte für alle großen Parteien abgefragt. Variable *v_101* speichert die Zustimmung zur SPD, *v_102* die Zustimmung zu den Grünen. Der Datensatz soll dann auf den Subgruppen untersucht werden, wo (a) die SPD-Identifikation sehr hoch ist, (b) wo die Grünen-Identifikation sehr hoch ist, und (c) wo Teilnehmer:innen eher links eingestellt sind:

```
add_subgroup(SPD_identifikation = (v_101 >= 7),  
             Gruene_identifikation = (v_102 >= 7),  
             ist_halblinks = (v_101 >= 7) & (v_102 >= 7))
```

3. Anpassung des config files: Ausgewählte Variablen in die Excel-Tabelle schreiben mit *add_data_to_main_overview*

3.1. Variablenweises Hinzufügen mit *add_to_excel*

Syntax: *add_to_excel(subgroup_var, varname, col_name_from_src, FUNC)*

<i>subgroup_var</i> :	NULL, Zeichenkette oder Liste von Zeichenketten
<i>varname</i> :	Zeichenkette
<i>col_name_from_src</i> :	Zeichenkette
<i>FUNC</i> :	Funktion

subgroup_var gibt an, ob der Datensatz entlang einer Subgruppe ausgewertet werden soll. Wird *NULL* angegeben, erfolgt keine Subgruppen-Auswertung. Wird nur eine Variable als Zeichenkette angegeben, erfolgt die Auswertung entlang dieser Variable. Werden mehrere Variablen als Liste angegeben, wird *add_to_excel* iterativ für jede einzelne Subgruppe der Liste ausgeführt.

varname beinhaltet den lesbaren Namen der auszuwertenden Variable und wird für die linke Spalte in der Excel-Tabelle benutzt

col_name_from_src beinhaltet die auszuwertende Variable in Form des Spaltennamens

FUNC gibt eine R-Funktion an, mit der *col_name_from_src* gruppenspezifisch ausgewertet werden soll (beispielsweise *min*, *max*, *mean*, *table*). Wird *mean* angegeben, wird automatisch eine ANOVA mit gerechnet, für *table* ein χ^2 -Test

Beispiel:

Bezogen auf die obigen Beispiele kann die gruppenspezifische Parteiwahl (Variable *parteiwahl*) ohne und mit Subgruppen-Unterteilung ausgewertet werden:

```
add_to_excel(list(NULL, "ist_halblinks"),  
             "gewählte Partei",  
             "parteiwahl",  
             table)
```