

# Package: esmprep (via r-universe)

October 26, 2024

**Type** Package

**Version** 0.2.0

**Date/Publication** 2019-05-28 18:03:31 CEST

**Title** Data Preparation During and After the Use of the Experience Sampling Methodology (ESM)

**Description** Support in preparing a raw ESM dataset for statistical analysis. Preparation includes the handling of errors (mostly due to technological reasons) and the generating of new variables that are necessary and/or helpful in meeting the conditions when statistically analyzing ESM data. The functions in 'esmprep' are meant to hierarchically lead from bottom, i.e. the raw (separated) ESM dataset(s), to top, i.e. a single ESM dataset ready for statistical analysis. This hierarchy evolved out of my personal experience in working with ESM data.

**Author** Marcel Miché <esmprep@gmail.com>

**Maintainer** Marcel Miché <esmprep@gmail.com>

**Depends** R (>= 3.2)

**Imports** lubridate (>= 1.6.0)

**License** GPL (>=2)

**URL** <https://github.com/mmiche/esmprep>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** yes

**Repository** <https://mmiche.r-universe.dev>

**RemoteUrl** <https://github.com/mmiche/esmprep>

**RemoteRef** HEAD

**RemoteSha** 8cd3330d9621ba6e69b2d9aa8df62d97eb988a95

## Contents

computeDuration . . . . .	3
computeTimeBetween . . . . .	4
computeTimeLag . . . . .	5
convertChars . . . . .	6
dateTimeFormats . . . . .	7
dateTimeFormats2 . . . . .	8
dayControl . . . . .	8
dayTest . . . . .	9
esAssign . . . . .	10
esAssigned . . . . .	13
esComplete . . . . .	14
esFinal . . . . .	16
esIdentical . . . . .	17
esItems . . . . .	18
esList . . . . .	19
esMerge . . . . .	20
esMerged1 . . . . .	21
esmprep . . . . .	22
esPlausible . . . . .	36
eveningControl . . . . .	38
eveningTest . . . . .	39
expectedDf . . . . .	40
expectedPromptIndex . . . . .	40
findChars . . . . .	42
genDateTime . . . . .	43
genKey . . . . .	45
intolerable . . . . .	46
isCompleteLs . . . . .	47
keyLsNew . . . . .	48
makeShift . . . . .	48
missingEndDateTime . . . . .	50
morningControl . . . . .	51
morningTest . . . . .	52
printRmInvalid . . . . .	53
printSuggestedShift . . . . .	54
randomMultSelection . . . . .	56
referenceDf . . . . .	57
referenceDfNew . . . . .	58
refPlausible . . . . .	58
relevantESVN . . . . .	60
RELEVANTINFO_ES . . . . .	61
relevantREFVN . . . . .	62
RELEVANTVN_ESext . . . . .	63
RELEVANTVN_REFext . . . . .	64
rmInvalid . . . . .	64
setES . . . . .	65



```
# Run function 27 of 29; see esmprep functions' hierarchy.
# -----
# randSelLs[["esRandSelIn"]] is the result of function 'randomMultSelection'.
durDf <- computeDuration(randSelLs[["esRandSelIn"]], RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
```

---

computeTimeBetween      *computeTimeBetween*

---

## Description

computeTimeBetween computes the duration between when an ESM questionnaire was started and when the subsequent one was finished, across all ESM questionnaires per person.

## Usage

```
computeTimeBetween(esDf, refDf, RELEVANTVN_ES = NULL,
  RELEVANTVN_REF = NULL)
```

## Arguments

esDf                    a data.frame. A single ESM dataset. It must contain the 2 columns that hold the date-time object for when an ESM questionnaire was started and finished, respectively.

refDf                    a data.frame. The reference dataset.

RELEVANTVN\_ES          a list. This list is generated by function [setES](#) and it is extended once either by function [genDateTime](#) or by function [splitDateTime](#).

RELEVANTVN\_REF        a list. This list is generated by function [setREF](#) and it is extended once either by function [genDateTime](#) or by function [splitDateTime](#).

## Value

esDf with the additional column TBESQ, i.e. the T\_ime B\_etween E\_xperience S\_ampling Q\_uestionnaires.

## See Also

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 28 of 29).

## Examples

```
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute computeTimeBetween. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# Use example list delivered with the package
RELEVANTVN_REF <- RELEVANTVN_REFext
intoleranceDf <- data.frame(prompt = c(2, 3, 4, 1, 1),
  expect = c(1, 1, 1, 2, 3))
```



## Examples

```
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute computeTimeLag. Start -----
# RELEVANTINFO_ES is delivered with the package
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
intoleranceDf <- data.frame(prompt = c(2, 3, 4, 1, 1),
expect = c(1, 1, 1, 2, 3))
# expectedDf is a raw ESM dataset, delivered with the package.
intolLs <- intolerable(expectedDf, intoleranceDf, RELEVANTINFO_ES)
randSelLs <- randomMultSelection(intolLs[["cleanedDf"]])
# Prerequisites in order to execute computeTimeLag. End -----
# -----
# Run function 26 of 29; see esmprep functions' hierarchy.
# -----
# randSelLs[["esRandSelIn"]] is the result of function 'randomMultSelection'.
lagDf <- computeTimeLag(randSelLs[["esRandSelIn"]], RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
```

---

convertChars

*convertChars*

---

## Description

convertChars converts specified single characters within specified ESM variables that contain text.

## Usage

```
convertChars(esDf, charCols, convDf, ignoreCase = FALSE)
```

## Arguments

esDf	a data.frame. A single ESM dataset. It must contain the 2 columns that hold the date-time object for when an ESM questionnaire was started and finished, respectively.
charCols	a vector of character strings. Each character string specifies a column name in the event sampling (raw) dataset which contains text in which specific characters shall be replaced by other characters, e.g. the dashed letter e by e.
convDf	a data.frame. The data frame must contain exactly 2 columns, the first column specifies the characters that shall be replaced, the second column specifies the characters that are going to replace the ones in the first column.
ignoreCase	logical. Enter TRUE if capitalization can be ignored in the course of replacement, else enter FALSE.

## Value

esDf Each column which contains text (as specified by the user!) now contains the text with the specific characters being converted (as specified by the user!).

**See Also**

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 16 of 29).

**Examples**

```

# -----
# Prerequisites in order to execute convertChars. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# isCompleteLs is a list of datasets, also delivered with the package
esMerged <- esMerge(isCompleteLs, RELEVANTVN_ES)
esMerged[,RELEVANTVN_ES[["ES_IMEI"]]] <- as.character(esMerged[,RELEVANTVN_ES[["ES_IMEI"]]])
findTextIdx <- findChars(esMerged)
# Prerequisites in order to execute convertChars. End -----
# -----
# Run function 16 of 29; see esmprep functions' hierarchy.
# -----
# From result of function 'findChars' select the indices of the items
# in the ESM dataset that contain text answers of the participants.
findTextIdx1 <- findTextIdx[c(1,2,9,10)]
# Use findTextIdx1 to generate the 3rd argument of function 'convertChars'.
textColumns <- names(findTextIdx1)
# Generate data.frame specifying the conversion of single characters.
convertCharsDf <- data.frame(c("ä", "ü"), c("ae", "ue"))
# Apply function. esMerged is the result of function 'esMerge'.
esMerged1 <- convertChars(esMerged, textColumns, convertCharsDf)
# # As default upper and lower case are NOT ignored! If you want them
# # ignored, additionally pass TRUE to the argument ignoreCase, like this:
# convertChars(esMerged, textColumns, convertCharsDf, ignoreCase=TRUE)
# -----

```

---

dateTimeFormats	<i>dateTimeFormats</i>
-----------------	------------------------

---

**Description**

dateTimeFormats displays the possible date-time options available (with examples).

**Usage**

```
dateTimeFormats()
```

**Value**

No return values. Instead a matrix is displayed with all the options for date-time objects, with examples.

**Examples**

```
# Run this function at any time you want to.
dateTimeFormats()
```

---

dateTimeFormats2	<i>dateTimeFormats2</i>
------------------	-------------------------

---

**Description**

dateTimeFormats displays the possible date-time options available (without examples).

**Usage**

```
dateTimeFormats2()
```

**Value**

No return values. Instead a vector is displayed with all the options for date-time objects.

**Examples**

```
# Run this function at any time you want to.
dateTimeFormats2()
```

---

dayControl	<i>Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the day by participants of the control group.</i>
------------	--

---

**Description**

Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the day by participants of the control group.

**Usage**

```
dayControl
```



**Format**

A data frame with 90 rows and 18 variables:

- V1. Arbitrary item, answer format numeric, ranging from 0 to 3.
- V1\_1. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V1\_2. Arbitrary item, answer format text.
- V2. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V2\_1. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3\_1. Arbitrary item, answer format text.
- V4. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V4\_1. Arbitrary item, answer format numeric, ranging from 0 to 4.
- V5. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V5\_1. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V7. Arbitrary item, answer format numeric, ranging from 0 to 1.
- survey\_name. Name of the ESM version.
- IMEI. IMEI number of the mobile device, used by the participant.
- start\_date. Date of when a single ESM questionnaire was started.
- start\_time. Time of when a single ESM questionnaire was started.
- end\_date. Date of when a single ESM questionnaire was ended.
- end\_time. Time of when a single ESM questionnaire was ended.

**Examples**

```
# Display the whole dataset in the console
dayControl
```

---

dayTest	<i>Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the day by participants of the test group.</i>
---------	---

---

**Description**

Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the day by participants of the test group.

**Usage**

```
data(dayTest)
```

**Format**

A data frame with 66 rows and 19 variables:

- V1. Arbitrary item, answer format numeric, ranging from 0 to 3.
- V1\_1. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V1\_2. Arbitrary item, answer format text.
- V2. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V2\_1. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3\_1. Arbitrary item, answer format text.
- V4. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V4\_1. Arbitrary item, answer format numeric, ranging from 0 to 4.
- V5. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V5\_1. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V7. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V8. Arbitrary item, answer format text.
- survey\_name. Name of the ESM version.
- IMEI. IMEI number of the mobile device, used by the participant.
- start\_date. Date of when a single ESM questionnaire was started.
- start\_time. Time of when a single ESM questionnaire was started.
- end\_date. Date of when a single ESM questionnaire was ended.
- end\_time. Time of when a single ESM questionnaire was ended.

**Examples**

```
# Display the whole dataset in the console
dayTest
```

---

esAssign

*esAssign*

---

**Description**

esAssign assigns ESM questionnaires to the (selected) persons who generated them.

**Usage**

```
esAssign(esDf, refDf, RELEVANTINFO_ES = NULL, RELEVANTVN_ES = NULL,
  RELEVANTVN_REF = NULL, singlePerson = NULL, prompted = NULL,
  promptTimeframe = 30, midnightPrompt = FALSE, dstDates = NULL)
```

## Arguments

esDf	a data.frame. A single ESM dataset. It must contain the 2 columns that hold the date-time object for when an ESM questionnaire was started and finished, respectively.
refDf	a data.frame. The reference dataset.
RELEVANTINFO_ES	a list. This list is generated by function <code>setES</code> .
RELEVANTVN_ES	a list. This list is generated by function <code>setES</code> and it is extended once either by function <code>genDateTime</code> or by function <code>splitDateTime</code> .
RELEVANTVN_REF	a list. This list is generated by function <code>setREF</code> and it is extended once either by function <code>genDateTime</code> or by function <code>splitDateTime</code> .
singlePerson	a character string. If you want to select a single person (must be contained in the reference dataset) enter its identification code (ID); else all participants in the reference dataset are selected (default).
prompted	logical. If there was no prompt per ESM day at all, enter FALSE; else ignore this argument, meaning that per default at least one prompt per ESM day is assumed.
promptTimeframe	an integer value. The default value is 30, i.e. within "genDateTime" <b>minutes</b> around each prompt a participant is expected to have answered the ESM questionnaire. This argument must be set to an integer value larger than 0, even if there was no prompt at all. If NA, NULL, 0 is passed to this argument, the function returns an error. If less than "genDateTime" minutes is passed to this argument, the function returns a warning message.
midnightPrompt	logical. Default FALSE, i.e. no participant was able to start a questionnaire around the midnight hour. However, if this was possible, set this argument to TRUE, in which case it takes the function <code>esAssign</code> a little longer to do its job, compared to <code>midnightPrompt = FALSE</code> .
dstDates	a vector of character strings. If a check shall be made concerning the daylight saving time (last weekend in March and October, respectively), enter the respective date(s) in the form yyyy-mm-dd as vector, e.g. <code>c("2007-10-28", "2008-03-30")</code> .

## Details

Data can only be assigned to those individuals who are contained in the reference dataset. `esAssign` is of utmost importance in preparing the ESM dataset because the results of any statistical analysis depends on the correct assignment of data to the persons who generated it. New columns in output list of function `esAssign` are

1. ID. Unique identification code of each participant.
2. CV\_ES. CV is short for count variable. It counts all the questionnaires that have been filled out by the participant during the ESM period. In incrementing order it starts at 1 and skips a number, whenever a questionnaire is missing.
3. CV\_ESDAY. This variable counts the single ESM days. In incrementing order it starts at 1. It only skips a number when all questionnaires of that day are missing.

4. CV\_ESWEEKDAY. This variable counts the weekday, with Monday represented by the value 1, ..., Sunday = 7.
5. PROMPT. Correspondance of the actual start time of the questionnaire to its prompt (in our exemplary dataset this ranges between 1 and 4).
6. PROMPTEND. Correspondance of the actual end time of the questionnaire to its prompt (in our exemplary dataset this ranges between 1 and 4).
7. LAG\_MINS. Time difference in minutes (rounded) between the scheduled time of the prompt and the actual start time of a questionnaire.
8. ES\_MULT. Dichotomous variable. The value 1 represents a questionnaire that has been filled out repeatedly at one specific prompt.
9. ES\_MULT2. Alternative representation of ES\_MULT. The very first questionnaire at a prompt is represented by the value 1, the second questionnaire (i.e. the first repeatedly filled out q.) is represented by the value 2, etc.
10. ST. Assigns the prompt/scheduled time (ST) to the actual start time of a questionnaire, by choosing the minimal time difference between all possible prompts (per participant) and the actual start time of the single ESM questionnaire.
11. STDATE. Variable is returned only if argument midnightPrompt is set to TRUE. Possible values and meaning: -1 = scheduled start date is prior to actual start date; 0 = scheduled start date and actual start date are equal to one another; 1 = scheduled date is subsequent to actual start date.
12. TFRAME. Dichotomous variable. The value 1 represents a questionnaire that is within the time frame, as specified by the user.
13. DST. Dichotomous variable. The value 1 represents a questionnaire's date to be equal or later than the daylight saving date, as specified by the user.
14. QWST. Dichotomous variable. The value 1 represents a questionnaire to be fully within the scheduled time, i.e. the time differences of both the actual start time and the actual end time are minimal relative to the same scheduled time.

Additionally, if the reference dataset contains any duplicates in the column representing the participant IDs, then esAssign stops and an error message is printed in the R console.

## Value

The user receives a list containing 4 datasets:

1. ES, i.e. ESM data assigned to selected participants with new columns added, see **Details**.
2. ESopt, i.e. the optimal ESM data sequence for all selected participants.
3. ESout, i.e. ESM data that couldn't be assigned.
4. ESrate, i.e. the average completion rates per participant, both per prompt and overall.

The effective ESM completion rates per selected individual and per prompt are also printed to the console. However, these are not the final completion rates, since some of the current questionnaires later might either be removed (see function [intolerable](#)) or be shifted to a neighboring prompt index (see functions [suggestShift](#) and [makeShift](#)).

**See Also**

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 17 of 29).

**Examples**

```
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esAssign. Start -----
# esMerged1 is the raw ESM dataset, also delivered with the package
# referenceDfNew is the modified reference dataset, delivered with
# the package.
# RELEVANTINFO_ES is delivered with the package
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# Use example list delivered with the package
RELEVANTVN_REF <- RELEVANTVN_REFext
# Prerequisites in order to execute esAssign. End -----
# -----
# Run function 17 of 29; see esmprep functions' hierarchy.
# -----
# Assign questionnaires contained in the raw ESM dataset to all participants listed
# in the reference dataset. esMerged1 is the result of function 'convertChars',
# referenceDfNew is the result of function 'genDateTime' or of function
# 'splitDateTime'.
# Run only the first 2 out of 8 participants (saves time). The warning message the
# user receives in this case (2 out of 8 participants) is correct.
esAssigned <- esAssign(esDf = esMerged1, refDf = referenceDfNew[1:2,], RELEVANTINFO_ES,
RELEVANTVN_ES, RELEVANTVN_REF)
# # Assign questionnaires contained in the raw ESM dataset to participant P001 listed
# # in the reference dataset.
# esAssigned <- esAssign(esDf = esMerged1, refDf = referenceDfNew, RELEVANTINFO_ES,
# RELEVANTVN_ES, RELEVANTVN_REF, singlePerson="P001")
# More options can be passed to 'esAssign', see parameter description. Note that when
# setting the argument midnightPrompt to TRUE, esAssign takes a bit longer to do its job.
# Output: List with 4 data.frames.
names(esAssigned)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
```

---

esAssigned

*Resulting dataset after function esAssign has been executed on the raw ESM dataset.*

---

**Description**

**Note:** The list of datasets esAssigned is the result of the function [esAssign](#), when the raw ESM dataset esMerged1 is one of the function arguments. esMerged1 is the result of function [convertChars](#).

**Usage**

esAssigned

**Format**

An object of class `list` of length 4.

---

esComplete

*esComplete*

---

**Description**

esComplete checks whether each ESM questionnaire is complete as specified by the user.

**Usage**

```
esComplete(dfList, lastItemList)
```

**Arguments**

`dfList` a list. Each element of the list must be a `data.frame`. Each `data.frame` is a separate raw ESM dataset/an ESM questionnaire version. If there is just one ESM version the list therefore contains one `data.frame`.

`lastItemList` a list. Each list element too must be a list. Each of these inner lists must contain exactly 4 elements:

1. First, a character string specifying the ESM questionnaire version
2. Second, a character string specifying the penultimate item of its respective ESM questionnaire version (i.e. the column name of the raw ESM dataset)
3. Third, a vector of at least one numeric value, specifying the condition upon which the last item is expected to contain a value, so that the questionnaire can be considered complete, and
4. Fourth, a character string specifying the last item of its respective ESM questionnaire version (i.e. the column name of the raw ESM dataset).

If there is no condition that determines which is the last item expected to contain a value, then the second and the third argument will have to be set to `NA`. See **Details** for more information.

**Details**

If due to some technical error an ESM questionnaire does not contain an end date and/or end time it might be a complete questionnaire nonetheless. Completion can be defined as the last item of the questionnaire containing valid data. In addition it is possible that the value in the penultimate item sets a condition upon which the questionnaire's completion is achieved either if the penultimate item contains a specific value (e.g. 0), which might signal that the questionnaire ends right there. However, if the penultimate item contains a value greater than 0, this might signal that the questionnaire's last item is expected to contain valid data.

### Value

dfList with additional column INCOMPLETE denoting an ESM questionnaire to be complete (= 0) or incomplete (= 1). If within the 2nd argument lastItemList at least one of the ESM questionnaire versions are passed more than once, then the additional columns will be named INCOMPLETE\_i, where i specifies the number of how often an ESM version has been passed (see **Examples**).

### See Also

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 13 of 29).

### Examples

```
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute esComplete. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)
plausibItems <- esItems(dfList=rmInvLs[["dfValid"]], RELEVANTVN_ES)
# Prerequisites in order to execute esComplete. End -----
# -----
# Run function 13 of 29; see esmprep functions' hierarchy.
# -----
# Generate second argument of function 'esComplete'. It's strongly recommended
# to read the explanation of this 2nd argument in the esmprep vignette, function
# 'esComplete'.
lastItemList <- list(
# If in survey version "morningTestGroup" variable "V6" contains the value 0,
# then variable "V6_1" is the last item expected to contain data, else "V6" is the last item
# expected to contain data.
list("morningTestGroup", "V6", 0, "V6_1"),
# In survey version "dayTestGroup" variable "V7" is the last item expected to contain data;
# unlike above, no conditions; NA as 2nd and 3rd element of the inner list are mandatory.
list("dayTestGroup", NA, NA, "V7"),
# Information of all further ESM versions are passed accordingly:
list("eveningTestGroup", "V9", 1, "V9_1"),
list("morningControlGroup", "V6", 0, "V6_1"),
list("dayControlGroup", NA, NA, "V7"),
# The last ESM version has 2 conditions, therefore it is passed 2 times:
# If V8_1 contains a value between 1 and 5, then V8_3 is the last item expected to
# contain data.
list("eveningControlGroup", "V8_1", 1:5, "V8_3"),
# If V8_1 contains the value 0, then V8_2 is the last item expected to contain data.
list("eveningControlGroup", "V8_1", 0, "V8_2"))
# Apply function 'esComplete'. rmInvLs[["dfValid"]] is one of the results of function
# rmInvalid.
isCompleteLs <- esComplete(rmInvLs[["dfValid"]], lastItemList)
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
```

---

esFinal	<i>esFinal</i>
---------	----------------

---

### Description

esFinal generates the final ESM dataset.

### Usage

```
esFinal(esDf, esOpt = NULL, complianceRate = NULL,
        RELEVANTINFO_ES = NULL, RELEVANTVN_ES = NULL, maxRows = NULL)
```

### Arguments

esDf	a data.frame. A single ESM dataset. It must contain the 2 columns that hold the date-time object for when an ESM questionnaire was started and finished, respectively.
esOpt	a data.frame. This data.frame is generated as part of a list returned by the function <a href="#">esAssign</a> . The name of this list's element is ESopt. Extract this element and pass it to the argument esOpt of this function.
complianceRate	integer. This value must be set to a value between 0 and 100. If you want to have the completion rates for all participants and you also want to make sure that all participants are kept in the final ESM dataset, simply set the argument complianceRate to 0.
RELEVANTINFO_ES	a list. This list is generated by function <a href="#">setES</a> .
RELEVANTVN_ES	a list. This list is generated by function <a href="#">setES</a> and it is extended once either by function <a href="#">genDateTime</a> or by function <a href="#">splitDateTime</a> .
maxRows	a numeric value. The number of data lines (per participant) in the final event sampling dataset; must be equal for all participants. If no number is entered the maximum number across all participants is used.

### Details

The empty rows will either denote ESM questionnaires that were missed by the participant or it will denote fillers, i.e. rows of empty data to fill up the number of rows to be equal across all participants. The number of maximum rows per participant either are computed by searching the actual maximum number of questionnaires started by the participant, or by what the user defines to be the maximum number of questionnaires (no less than 2).

### Value

The user receives a list containing 3 elements:

1. ESfinal, i.e. with empty rows of data added and with 2 additional columns MISSED and FILLER. MISSED refers to questionnaires that should have been answered by the participants but weren't. FILLER refers to empty rows of data due to all participants having to have equally many rows of data for multilevel modeling. See **Details** for more information..



2. ESrateFinal, i.e. the average completion rates per participant, both per prompt and overall.
3. ESfinalOut, i.e. ESM data that contains participants that won't be used for statistical analyses due to having less than a minimum number of answered ESM questionnaires. If no participant is deselected a character string is returned that affirms that no deselection took place.

Unlike the interim result returned by the function `esAssign` (whenever there was at least one daily prompt) the data.frame `ESrateFinal` shows the final completion rates per participant.

**See Also**

Exemplary code (fully executable) in the documentation of `esmprep` (function 29 of 29).

**Examples**

```
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esFinal. Start -----
# RELEVANTINFO_ES is delivered with the package
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# tbsqDf is a raw ESM dataset, delivered with the package.
# Prerequisites in order to execute esFinal. End -----
# -----
# Run function 29 of 29; see esmprep functions' hierarchy.
# -----
# tbsqDf is the result of function 'computeTimeBetween'.
esDfFin <- esFinal(tbsqDf, esOpt=esAssigned[["ESopt"]], complianceRate=50,
RELEVANTINFO_ES, RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
```

---

<code>esIdentical</code>	<i>esIdentical</i>
--------------------------	--------------------

---

**Description**

`esIdentical` checks whether there are fully identical lines of data in the ESM dataset.

**Usage**

```
esIdentical(esDf, RELEVANTVN_ES = NULL)
```

**Arguments**

- `esDf`                  a data.frame. A single ESM dataset. It must contain the 2 columns that hold the date-time object for when an ESM questionnaire was started and finished, respectively.
- `RELEVANTVN_ES`        a list. This list is generated by function `setES` and it is extended once either by function `genDateTime` or by function `splitDateTime`.

### Details

At least 2 ESM questionnaires must be 100 percent identical in order to be registered as duplicates.

### Value

esDf with the additional column IDENT denoting an ESM questionnaire to be unique (= 0) or to be duplicated (= 1).

### See Also

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 19 of 29).

### Examples

```
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute esIdentical. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# esAssigned is a list of datasets, delivered with the package. It is
# the result of the assignment of the ESM questionnaires to ALL 8
# participants in the reference dataset.
noEndDf <- missingEndTime(esAssigned[["ES"]], RELEVANTVN_ES)
# Prerequisites in order to execute esIdentical. End -----
# -----
# Run function 19 of 29; see esmprep functions' hierarchy.
# -----
# noEndDf is the result of function 'noEndTime'.
identDf <- esIdentical(noEndDf, RELEVANTVN_ES)
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
```

---

esItems

*esItems*

---

### Description

esItems registers the ESM questionnaire items as opposed to variables containing information about the questionnaire, e.g. the time stamps of when it was started and finished.

### Usage

```
esItems(dfList, RELEVANTVN_ES = NULL)
```

### Arguments

**dfList** a list. Each element of the list must be a `data.frame`. Each `data.frame` is a separate raw ESM dataset/an ESM questionnaire version. If there is just one ESM version the list therefore contains one `data.frame`.

**RELEVANTVN\_ES** a list. This list is generated by function `setES` and it is extended once either by function `genDateTime` or by function `splitDateTime`.

**Value**

A list with all the item names per ESM questionnaire version. Columns specifying dates, times, or IMEI numbers - as specified in function [setES](#) - are not used in function [esPlausible](#), which takes the result of `esItems` as 2nd argument.

**See Also**

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 11 of 29).

**Examples**

```
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esItems. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)
# Prerequisites in order to execute esItems. End -----
# -----
# Run function 11 of 29; see esmprep functions' hierarchy.
# -----
# Extract the item names of the raw ESM datasets. rmInvLs[["dfValid"]]
# is one of the results from function 'rmInvalid'
plausibItems <- esItems(dfList=rmInvLs[["dfValid"]], RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
```

---

esList

*esList*

---

**Description**

`esList` holds all separate raw ESM datasets in one list, which is an R-built-in data structure.

**Usage**

```
esList(dfList, RELEVANTVN_ES = NULL)
```

**Arguments**

`dfList` a list. Each element of the list must be a data.frame. Each data.frame is a separate raw ESM dataset/an ESM questionnaire version. If there is just one ESM version the list therefore contains one data.frame.

`RELEVANTVN_ES` a list. This list is generated by function [setES](#) and it is extended once either by function [genDateTime](#) or by function [splitDateTime](#).

**Details**

The separate raw ESM datasets are still separated in the list. Each list element is named according to the survey version as specified by the content of the respective column name in each ESM dataset.

**Value**

A named list of the ESM datasets, where each list name is equal to the name of the respective ESM questionnaire version.

**See Also**

Exemplary code (fully executable) in the documentation of `esmprep` (function 5 of 29).

**Examples**

```
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esList. Start -----
# Generate argument RELEVANTVN_ES
reLEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
  STARTDATE="start_date", STARTTIME="start_time",
  ENDDATE="end_date", ENDTIME="end_time")
imeiNumbers <- as.character(referenceDf$imei)
surveyNames <- c("morningTestGroup", "dayTestGroup", "eveningTestGroup",
  "morningControlGroup", "dayControlGroup", "eveningControlGroup")
RELEVANT_ES <- setES(4, imeiNumbers, surveyNames, reLEs)
RELEVANTVN_ES <- RELEVANT_ES[["RELEVANTVN_ES"]]
# Prerequisites in order to execute esList. End -----
# -----
# Run function 6 of 29; see esmprep functions' hierarchy.
# -----
# 6 exemplary raw ESM (sub-)datasets.
esLs <- esList(list(morningControl, dayControl, eveningControl,
  morningTest, dayTest, eveningTest), RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
```

---

*esMerge*

---

*esMerge*

---

**Description**

`esMerge` merges all the ESM versions into one single dataset.

**Usage**

```
esMerge(dfList, RELEVANTVN_ES = NULL)
```

**Arguments**

- |                            |  |
|----------------------------|--|
| <code>dfList</code>        | a list. Each element of the list must be a <code>data.frame</code> . Each <code>data.frame</code> is a separate raw ESM dataset/an ESM questionnaire version. If there is just one ESM version the list therefore contains one <code>data.frame</code> . |
| <code>RELEVANTVN_ES</code> | a list. This list is generated by function <code>setES</code> and it is extended once either by function <code>genDateTime</code> or by function <code>splitDateTime</code> .  |

**Value**

One single (merged) ESM dataset, where all datasets of the argument `dfList` are included.

**See Also**

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 14 of 29).

**Examples**

```
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute esMerge. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# isCompleteLs is a list of datasets, also delivered with the package
# Prerequisites in order to execute esMerge. End -----
# -----
# Run function 14 of 29; see esmprep functions' hierarchy.
# -----
# Merge all raw ESM datasets. isCompleteLs is the result
# of function 'esComplete'.
esMerged <- esMerge(isCompleteLs, RELEVANTVN_ES)
# If preferred convert the 15 digit IMEI number from scientific notation to text.
esMerged[,RELEVANTVN_ES[["ES_IMEI"]]] <- as.character(esMerged[,RELEVANTVN_ES[["ES_IMEI"]]])
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
```

---

esMerged1

*Resulting dataset after function `convertChars` has been executed on the raw ESM dataset.*

---

**Description**

**Note:** The dataset `esMerged1` is the result of the function [convertChars](#), when the raw ESM dataset `esMerged` is one of the function arguments. `esMerged` is the result of function [esMerge](#).

**Usage**

```
esMerged1
```

**Format**

An object of class `data.frame` with 228 rows and 32 columns.



```
# This is function 2 of 29 in the hierarchy of 'esmprep' functions.
# -----
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute setREF. Start -----
relRef <- relevantREFVN(ID="id", IMEI="imei", ST="st",
  STARTDATE="start_date", STARTTIME="start_time",
  ENDDATE="end_date", ENDTIME="end_time")
# Prerequisites in order to execute setREF. End -----
# -----
# Run function 2 of 29; see esmprep functions' hierarchy.
# -----
# 4 is the number of daily prompts.
# relRef is the result of function 'relevantREFVN'
# Relevant variables names of reference dataset.
(RELEVANTVN_REF <- setREF(4, relRef))
# With date-time objects instead of separate date and time
relRef <- relevantREFVN(ID="id", IMEI="imei", ST="st",
  START_DATETIME="start_dateTime", END_DATETIME="end_dateTime")
(RELEVANTVN_REF <- setREF(4, relRef))
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# FUNCTION NAME: relevantESVN
# -----
# This is function 3 of 29 in the hierarchy of 'esmprep' functions.
# -----
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# -----
# Run function 3 of 29; see esmprep functions' hierarchy.
# -----
# With date and time as separate arguments
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
  STARTDATE="start_date", STARTTIME="start_time",
  ENDDATE="end_date", ENDTIME="end_time")
# With date-time objects instead of separate date and time
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
  START_DATETIME="start_dateTime", END_DATETIME="end_dateTime")
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# FUNCTION NAME: setES
# -----
# This is function 4 of 29 in the hierarchy of 'esmprep' functions.
# -----
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute setES. Start -----
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
  STARTDATE="start_date", STARTTIME="start_time",
  ENDDATE="end_date", ENDTIME="end_time")
# Prerequisites in order to execute setES. End -----
# -----
# Run function 4 of 29; see esmprep functions' hierarchy.
# -----
# imeiNumbers is the vector containing all IMEI numbers used in
# the ESM study; use the respective entries in the referenceDf.
imeiNumbers <- as.character(referenceDf$imei)
# surveyNames is the vector containing all ESM version names.
```

```

surveyNames <- c(
# Test group
  "morningTestGroup", "dayTestGroup", "eveningTestGroup",
# Control group
  "morningControlGroup", "dayControlGroup", "eveningControlGroup")
# 4 is the number of daily prompts
# relEs is the result of function 'relevantESVN'
RELEVANT_ES <- setES(4, imeiNumbers, surveyNames, relEs)
# Extract relevant ESM general information
RELEVANTINFO_ES <- RELEVANT_ES[["RELEVANTINFO_ES"]]
# Extract list of relevant variables names of raw ESM datasets.
RELEVANTVN_ES <- RELEVANT_ES[["RELEVANTVN_ES"]]

# With date-time objects instead of separate date and time
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
START_DATETIME="start_dateTime", END_DATETIME="end_dateTime")
RELEVANT_ES <- setES(4, imeiNumbers, surveyNames, relEs)
# Extract relevant ESM general information
RELEVANTINFO_ES <- RELEVANT_ES[["RELEVANTINFO_ES"]]
# Extract list of relevant variables names of raw ESM datasets.
RELEVANTVN_ES <- RELEVANT_ES[["RELEVANTVN_ES"]]
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# FUNCTION NAME: esList
# -----
# This is function 5 of 29 in the hierarchy of 'esmprep' functions.
# -----
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute esList. Start -----
# Generate argument RELEVANTVN_ES
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
STARTDATE="start_date", STARTTIME="start_time",
ENDDATE="end_date", ENDTIME="end_time")
imeiNumbers <- as.character(referenceDf$imei)
surveyNames <- c("morningTestGroup", "dayTestGroup", "eveningTestGroup",
"morningControlGroup", "dayControlGroup", "eveningControlGroup")
RELEVANT_ES <- setES(4, imeiNumbers, surveyNames, relEs)
RELEVANTVN_ES <- RELEVANT_ES[["RELEVANTVN_ES"]]
# Prerequisites in order to execute esList. End -----
# -----
# Run function 6 of 29; see esmprep functions' hierarchy.
# -----
# 6 exemplary raw ESM (sub-)datasets.
esLs <- esList(list(morningControl, dayControl, eveningControl,
morningTest, dayTest, eveningTest), RELEVANTVN_ES)
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# FUNCTION NAME: genKey
# -----
# This is function 6 of 29 in the hierarchy of 'esmprep' functions.
# -----
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute genKey. Start -----
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
STARTDATE="start_date", STARTTIME="start_time",

```



```

ENDDATE="end_date", ENDTIME="end_time")
imeiNumbers <- as.character(referenceDf$imei)
surveyNames <- c("morningTestGroup", "dayTestGroup", "eveningTestGroup",
"morningControlGroup", "dayControlGroup", "eveningControlGroup")
RELEVANT_ES <- setES(4, imeiNumbers, surveyNames, relEs)
RELEVANTVN_ES <- RELEVANT_ES[["RELEVANTVN_ES"]]
esLs <- esList(list(morningControl, dayControl, eveningControl,
morningTest, dayTest, eveningTest), RELEVANTVN_ES)
# Prerequisites in order to execute genKey. End -----
# -----
# Run function 6 of 29; see esmprep functions' hierarchy.
# -----
# esLs is the result of function 'esList'.
keyLs <- genKey(esLs)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: genDateTime
# -----
# This is function 7 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute genDateTime. Start -----
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
STARTDATE="start_date", STARTTIME="start_time",
ENDDATE="end_date", ENDTIME="end_time")
imeiNumbers <- as.character(referenceDf$imei)
surveyNames <- c("morningTestGroup", "dayTestGroup", "eveningTestGroup",
"morningControlGroup", "dayControlGroup", "eveningControlGroup")
RELEVANT_ES <- setES(4, imeiNumbers, surveyNames, relEs)
RELEVANTINFO_ES <- RELEVANT_ES[["RELEVANTINFO_ES"]]
RELEVANTVN_ES <- RELEVANT_ES[["RELEVANTVN_ES"]]
esLs <- esList(list(morningControl, dayControl, eveningControl,
morningTest, dayTest, eveningTest), RELEVANTVN_ES)
keyLs <- genKey(esLs)
relRef <- relevantREFVN(ID="id", IMEI="imei", ST="st",
STARTDATE="start_date", STARTTIME="start_time",
ENDDATE="end_date", ENDTIME="end_time")
RELEVANTVN_REF <- setREF(4, relRef)
# Prerequisites in order to execute genDateTime. End -----
# -----
# Run function 7 of 29; see esmprep functions' hierarchy.
# -----
# Applying function to reference dataset (7a of 29)
referenceDfList <- genDateTime(referenceDf, "REF", RELEVANTINFO_ES, RELEVANTVN_ES,
RELEVANTVN_REF)

# Extract reference dataset from output
referenceDfNew <- referenceDfList[["refOrEsDf"]]

# Extract extended list of relevant variables names of reference dataset
RELEVANTVN_REF <- referenceDfList[["extendedVNList"]]

# Applying function to raw ESM dataset(s) (7b of 29)
# keyLs is the result of function 'genKey'.

```

```

keyList <- genDateTime(keyLs, "ES", RELEVANTINFO_ES, RELEVANTVN_ES,
RELEVANTVN_REF)

# Extract list of raw ESM datasets from output
keyLsNew <- keyList[["refOrEsDf"]]

# Extract extended list of relevant variables names of raw ESM datasets
RELEVANTVN_ES <- keyList[["extendedVNList"]]
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: refPlausible
# -----
# This is function 8 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute refPlausible. Start -----
# Use example list delivered with the package
RELEVANTVN_REF <- RELEVANTVN_REFext
# Prerequisites in order to execute refPlausible. End -----
# -----
# Run function 8 of 29; see esmprep functions' hierarchy.
# -----
# In an ESM study all participants answer questionnaires during a time period which
# usually is equal across all participants (e.g. seven days). This function enables the
# user to check whether in the reference dataset the ESM period is plausible. For
# instance, a negative ESM time period would clearly be implausible, e.g. the user
# setting the beginning of the ESM time period after the end of it (which is
# impossible unless a functioning time machine is involved :-).
referenceDfNew1 <- refPlausible(refDf=referenceDfNew, units="days", RELEVANTVN_REF)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: rmInvalid
# -----
# This is function 9 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute rmInvalid. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
# Prerequisites in order to execute rmInvalid. End -----
# -----
# Run function 9 of 29; see esmprep functions' hierarchy.
# -----
# keyLsNew is the result of function 'genDateTime' (or of function 'splitDateTime').
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)
# Result of function 'rmInvalid' is a list with 4 elements:
names(rmInvLs)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: printRmInvalid
# -----
# This is function 10 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute printRmInvalid. Start -----

```

```

# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)
# Prerequisites in order to execute printRmInvalid. End -----
# -----
# Run function 10 of 29; see esmprep functions' hierarchy.
# -----
# rmInvLs is the result of function 'rmInvalid'. Display its result
# in the console both tabulated and in detail.
key_rmLs <- printRmInvalid(rmInvLs, smr="both", RELEVANTVN_ES)
# Display the list containing the KEY values of all questionnaires
# that have been removed.
key_rmLs
# Since there have been warning messages in 4 of the 6 datasets,
# the first ESM item (name: V1) was automatically converted to class
# character, although it is numeric. So we'll re-convert V1's class.
# Check V1 prior to conversion
str(rmInvLs[["dfValid"]][[2]][1:2])
rmInvLs[["dfValid"]] <- sapply(rmInvLs[["dfValid"]], function(x) {
  x[,"V1"] <- as.numeric(x[,"V1"])
  return(x) } )
# Check V1 after conversion
str(rmInvLs[["dfValid"]][[2]][1:2])
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: esItems
# -----
# This is function 11 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esItems. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)
# Prerequisites in order to execute esItems. End -----
# -----
# Run function 11 of 29; see esmprep functions' hierarchy.
# -----
# Extract the item names of the raw ESM datasets. rmInvLs[["dfValid"]]
# is one of the results from function 'rmInvalid'
plausibItems <- esItems(dfList=rmInvLs[["dfValid"]], RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: esPlausible
# -----
# This is function 12 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esPlausible. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)

```

```

plausibItems <- esItems(dfList=rmInvLs[["dfValid"]], RELEVANTVN_ES)
# Prerequisites in order to execute esPlausible. End -----
# -----
# Run function 12 of 29; see esmprep functions' hierarchy.
# -----
# Help checking the plausibility of items in the raw ESM datasets. rmInvLs[["dfValid"]]
# is one of the results from function 'rmInvalid'.
# plausibItems is the result of function 'esItems'.
plausibLs <- esPlausible(dfList=rmInvLs[["dfValid"]], itemVecList=plausibItems)
# Display the results (4 data frames) to the console
# plausibNames gives an overview of the item names across all ESM versions
plausibLs[["plausibNames"]]
# plausibClass gives an overview of the variable types of all items
plausibLs[["plausibClass"]]
# plausibRowNa shows for each ESM version the number of lines in the raw
# ESM datasets and how much percent of the data is missing (NAs).
plausibLs[["plausibRowNa"]]
# plausibMinMax shows for each numeric variable the minimum and maximum.
plausibLs[["plausibMinMax"]]
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: esComplete
# -----
# This is function 13 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esComplete. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)
plausibItems <- esItems(dfList=rmInvLs[["dfValid"]], RELEVANTVN_ES)
# Prerequisites in order to execute esComplete. End -----
# -----
# Run function 13 of 29; see esmprep functions' hierarchy.
# -----
# Generate second argument of function 'esComplete'. It's strongly recommended
# to read the explanation of this 2nd argument in the esmprep vignette, function
# 'esComplete'.
lastItemList <- list(
# If in survey version "morningTestGroup" variable "V6" contains the value 0,
# then variable "V6_1" is the last item expected to contain data, else "V6" is the last item
# expected to contain data.
list("morningTestGroup", "V6", 0, "V6_1"),
# In survey version "dayTestGroup" variable "V7" is the last item expected to contain data;
# unlike above, no conditions; NA as 2nd and 3rd element of the inner list are mandatory.
list("dayTestGroup", NA, NA, "V7"),
# Information of all further ESM versions are passed accordingly:
list("eveningTestGroup", "V9", 1, "V9_1"),
list("morningControlGroup", "V6", 0, "V6_1"),
list("dayControlGroup", NA, NA, "V7"),
# The last ESM version has 2 conditions, therefore it is passed 2 times:
# If V8_1 contains a value between 1 and 5, then V8_3 is the last item expected to
# contain data.

```

```

list("eveningControlGroup", "V8_1", 1:5, "V8_3"),
# If V8_1 contains the value 0, then V8_2 is the last item expected to contain data.
list("eveningControlGroup", "V8_1", 0, "V8_2"))
# Apply function 'esComplete'. rmInvLs[["dfValid"]] is one of the results of function
# rmInvalid.
isCompleteLs <- esComplete(rmInvLs[["dfValid"]], lastItemList)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: esMerge
# -----
# This is function 14 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esMerge. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# isCompleteLs is a list of datasets, also delivered with the package
# Prerequisites in order to execute esMerge. End -----
# -----
# Run function 14 of 29; see esmprep functions' hierarchy.
# -----
# Merge all raw ESM datasets. isCompleteLs is the result
# of function 'esComplete'.
esMerged <- esMerge(isCompleteLs, RELEVANTVN_ES)
# If preferred convert the 15 digit IMEI number from scientific notation to text.
esMerged[,RELEVANTVN_ES[["ES_IMEI"]]] <- as.character(esMerged[,RELEVANTVN_ES[["ES_IMEI"]]])
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: findChars
# -----
# This is function 15 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute findChars. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# isCompleteLs is a list of datasets, also delivered with the package
esMerged <- esMerge(isCompleteLs, RELEVANTVN_ES)
esMerged[,RELEVANTVN_ES[["ES_IMEI"]]] <- as.character(esMerged[,RELEVANTVN_ES[["ES_IMEI"]]])
# Prerequisites in order to execute findChars. End -----
# -----
# Run function 15 of 29; see esmprep functions' hierarchy.
# -----
# esMerged is the result of function 'esMerge'
findTextIdx <- findChars(esMerged)
# Display structure of function output
str(findTextIdx)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: convertChars
# -----
# This is function 16 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute convertChars. Start -----
# Use example list delivered with the package

```

```

RELEVANTVN_ES <- RELEVANTVN_ESext
# isCompleteLs is a list of datasets, also delivered with the package
esMerged <- esMerge(isCompleteLs, RELEVANTVN_ES)
esMerged[,RELEVANTVN_ES[["ES_IMEI"]]] <- as.character(esMerged[,RELEVANTVN_ES[["ES_IMEI"]]])
findTextIdx <- findChars(esMerged)
# Prerequisites in order to execute convertChars. End -----
# -----
# Run function 16 of 29; see esmprep functions' hierarchy.
# -----
# From result of function 'findChars' select the indices of the items
# in the ESM dataset that contain text answers of the participants.
findTextIdx1 <- findTextIdx[c(1,2,9,10)]
# Use findTextIdx1 to generate the 3rd argument of function 'convertChars'.
textColumns <- names(findTextIdx1)
# Generate data.frame specifying the conversion of single characters.
convertCharsDf <- data.frame(c("ä", "ü"), c("ae", "ue"))
# Apply function. esMerged is the result of function 'esMerge'.
esMerged1 <- convertChars(esMerged, textColumns, convertCharsDf)
# # As default upper and lower case are NOT ignored! If you want them
# # ignored, additionally pass TRUE to the argument ignoreCase, like this:
# convertChars(esMerged, textColumns, convertCharsDf, ignoreCase=TRUE)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: esAssign
# -----
# This is function 17 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esAssign. Start -----
# esMerged1 is the raw ESM dataset, also delivered with the package
# referenceDfNew is the modified reference dataset, delivered with
# the package.
# RELEVANTINFO_ES is delivered with the package
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# Use example list delivered with the package
RELEVANTVN_REF <- RELEVANTVN_REFext
# Prerequisites in order to execute esAssign. End -----
# -----
# Run function 17 of 29; see esmprep functions' hierarchy.
# -----
# Assign questionnaires contained in the raw ESM dataset to all participants listed
# in the reference dataset. esMerged1 is the result of function 'convertChars',
# referenceDfNew is the result of function 'genDateTime' or of function
# 'splitDateTime'.
# Run only the first 2 out of 8 participants (saves time). The warning message the
# user receives in this case (2 out of 8 participants) is correct.
esAssigned <- esAssign(esDf = esMerged1, refDf = referenceDfNew[1:2,], RELEVANTINFO_ES,
RELEVANTVN_ES, RELEVANTVN_REF)
# # Assign questionnaires contained in the raw ESM dataset to participant P001 listed
# # in the reference dataset.
# esAssigned <- esAssign(esDf = esMerged1, refDf = referenceDfNew, RELEVANTINFO_ES,
# RELEVANTVN_ES, RELEVANTVN_REF, singlePerson="P001")
# More options can be passed to 'esAssign', see parameter description. Note that when

```



```

# participants in the reference dataset.
noEndDf <- missingEndTime(esAssigned[["ES"]], RELEVANTVN_ES)
identDf <- esIdentical(noEndDf, RELEVANTVN_ES)
# Prerequisites in order to execute suggestShift. End -----
# -----
# Run function 20 of 29; see esmprep functions' hierarchy.
# -----
# identDf is the result of function 'esIdentical'.
# 100 represents the number of minutes that at least must have passed
# between the scheduled start of an ESM questionnaire at its actual start
# in order for the questionnaire to be eligible for shifting (see function
# makeShift).
sugShift <- suggestShift(identDf, 100, RELEVANTINFO_ES, RELEVANTVN_ES)
# Display output element 'suggestShiftDf':
sugShift$suggestShiftDf
# Display output element 'printShiftDf':
sugShift$printShiftDf
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# FUNCTION NAME: printSuggestedShift
# -----
# This is function 21 of 29 in the hierarchy of 'esmprep' functions.
# -----
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute printSuggestedShift. Start -----
# Use example list delivered with the package
RELEVANTINFO_ES <- RELEVANTINFO_ES
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# esAssigned is a list of datasets, delivered with the package. It is
# the result of the assignment of the ESM questionnaires to ALL 8
# participants in the reference dataset.
noEndDf <- missingEndTime(esAssigned[["ES"]], RELEVANTVN_ES)
identDf <- esIdentical(noEndDf, RELEVANTVN_ES)
sugShift <- suggestShift(identDf, 100, RELEVANTINFO_ES, RELEVANTVN_ES)
# Prerequisites in order to execute printSuggestedShift. End -----
# -----
# Run function 21 of 29; see esmprep functions' hierarchy.
# -----
# Display the result of function 'suggestShift' in the console.
printSuggestedShift(sugShift, RELEVANTVN_ES)
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# FUNCTION NAME: makeShift
# -----
# This is function 22 of 29 in the hierarchy of 'esmprep' functions.
# -----
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute makeShift. Start -----
# RELEVANTINFO_ES is delivered with the package
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# Use example list delivered with the package
RELEVANTVN_REF <- RELEVANTVN_REFext
# esAssigned is a list of datasets, delivered with the package. It is

```



```

# the result of the assignment of the ESM questionnaires to ALL 8
# participants in the reference dataset.
noEndDf <- missingEndTime(esAssigned[["ES"]], RELEVANTVN_ES)
identDf <- esIdentical(noEndDf, RELEVANTVN_ES)
sugShift <- suggestShift(identDf, 100, RELEVANTINFO_ES, RELEVANTVN_ES)
# Prerequisites in order to execute makeShift. End -----
# -----
# Run function 22 of 29; see esmprep functions' hierarchy.
# -----
# sugShift is the result of function 'suggestShift'. referenceDfNew is the result
# of function 'genDateTime' or of function 'splitDateTime'.
# keyPromptDf is generated by using part of the output of function suggestShift,
# i.e. by selecting the columns NEW_PROMPT and SHIFTKEY from suggestShiftDf.
keyPromptDf <- sugShift$suggestShiftDf[,c("NEW_PROMPT", "SHIFTKEY")]
madeShift <- makeShift(sugShift, referenceDfNew, keyPromptDf, RELEVANTINFO_ES, RELEVANTVN_REF)
# Tip! Display the result of function 'makeShift' in the console
# in order to check whether the shifting was successful.
printSuggestedShift(madeShift, RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: expectedPromptIndex
# -----
# This is function 23 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute makeShift. Start -----
# Use example list delivered with the package
RELEVANTINFO_ES <- RELEVANTINFO_ES
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# Use example list delivered with the package
RELEVANTVN_REF <- RELEVANTVN_REFext
# esAssigned is a list of datasets, delivered with the package. It is
# the result of the assignment of the ESM questionnaires to ALL 8
# participants in the reference dataset.
noEndDf <- missingEndTime(esAssigned[["ES"]], RELEVANTVN_ES)
identDf <- esIdentical(noEndDf, RELEVANTVN_ES)
sugShift <- suggestShift(identDf, 100, RELEVANTINFO_ES, RELEVANTVN_ES)
keyPromptDf <- sugShift$suggestShiftDf[,c("NEW_PROMPT", "SHIFTKEY")]
madeShift <- makeShift(sugShift, referenceDfNew, keyPromptDf, RELEVANTINFO_ES, RELEVANTVN_REF)
# Prerequisites in order to execute makeShift. End -----
# -----
# Run function 23 of 29; see esmprep functions' hierarchy.
# -----
# Generate second argument of function 'expectedPromptIndex'. It's strongly
# recommended to read the explanation of this 2nd argument in the esmprep
# vignette, function 'expectedPromptIndex'.
expIdxList <- list(
# I - the user - expect in the ESM version morningTestGroup that
# prompt no.1 is always linked to the value 1.
list("morningTestGroup", 1, 1),
# I - the user - expect in the ESM version dayTestGroup that
# prompt no. 2 and no.3 are always linked to the value 2.
list("dayTestGroup", c(2, 3), 2),

```

```

# Information of all further ESM versions are passed accordingly:
list("eveningTestGroup", 4, 3),
list("morningControlGroup", 1, 1),
list("dayControlGroup", c(2,3), 2),
list("eveningControlGroup", 4, 3))
# madeShiftDf$esDf is part of the output of function 'makeShift', if at
# least one questionnaire was shifted to a neighboring prompt index.
expectedDf <- expectedPromptIndex(madeShift$esDf, expIdxList, RELEVANTINFO_ES,
RELEVANTVN_ES)
# If no questionnaire is suggested for shifting (see function suggestShift)
# use the result of function suggestShift as 1st argument, like this:
# expectedDf <- expectedPromptIndex(sugShift$esDf, expIdxList, RELEVANTINFO_ES,
RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: intolerable
# -----
# This is function 24 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute intolerable. Start -----
# RELEVANTINFO_ES is delivered with the package
# expectedDf is a raw ESM dataset, delivered with the package.
# Prerequisites in order to execute intolerable. End -----
# -----
# Run function 24 of 29; see esmprep functions' hierarchy.
# -----
# Generate second argument of function 'intolerable'
(intoleranceDf <- data.frame(
# Column 'prompt': Prompts that must NEVER be comined with expected categories.
prompt = c(2, 3, 4, 1, 1),
# Column 'expect': Expected categories that must NEVER be combined with the prompts.
expect = c(1, 1, 1, 2, 3)))
# Read: Prompts 2, 3, and 4 must never be combined with expected category 1.
# Read: Prompt 1 must never be combined with expected category 2.
# Read: Prompt 1 must never be combined with expected category 3.
# expectedDf is the result of function 'expectedPromptIndex'.
intolLs <- intolerable(expectedDf, intoleranceDf, RELEVANTINFO_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: randomMultSelection
# -----
# This is function 25 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute randomMultSelection. Start -----
# RELEVANTINFO_ES is delivered with the package
intoleranceDf <- data.frame(prompt = c(2, 3, 4, 1, 1),
expect = c(1, 1, 1, 2, 3))
# expectedDf is a raw ESM dataset, delivered with the package.
intolLs <- intolerable(expectedDf, intoleranceDf, RELEVANTINFO_ES)
# Prerequisites in order to execute randomMultSelection. End -----
# -----
# Run function 25 of 29; see esmprep functions' hierarchy.
# -----

```

```

# intolLs[["cleanedDf"]] is the result of function 'intolerable'.
randSelLs <- randomMultSelection(intolLs[["cleanedDf"]])
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: computeTimeLag
# -----
# This is function 26 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute computeTimeLag. Start -----
# RELEVANTINFO_ES is delivered with the package
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
intoleranceDf <- data.frame(prompt = c(2, 3, 4, 1, 1),
expect = c(1, 1, 1, 2, 3))
# expectedDf is a raw ESM dataset, delivered with the package.
intolLs <- intolerable(expectedDf, intoleranceDf, RELEVANTINFO_ES)
randSelLs <- randomMultSelection(intolLs[["cleanedDf"]])
# Prerequisites in order to execute computeTimeLag. End -----
# -----
# Run function 26 of 29; see esmprep functions' hierarchy.
# -----
# randSelLs[["esRandSelIn"]] is the result of function 'randomMultSelection'.
lagDf <- computeTimeLag(randSelLs[["esRandSelIn"]], RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: computeDuration
# -----
# This is function 27 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute computeDuration. Start -----
# RELEVANTINFO_ES is delivered with the package
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
intoleranceDf <- data.frame(prompt = c(2, 3, 4, 1, 1),
expect = c(1, 1, 1, 2, 3))
# expectedDf is a raw ESM dataset, delivered with the package.
intolLs <- intolerable(expectedDf, intoleranceDf, RELEVANTINFO_ES)
randSelLs <- randomMultSelection(intolLs[["cleanedDf"]])
# Prerequisites in order to execute computeDuration. End -----
# -----
# Run function 27 of 29; see esmprep functions' hierarchy.
# -----
# randSelLs[["esRandSelIn"]] is the result of function 'randomMultSelection'.
durDf <- computeDuration(randSelLs[["esRandSelIn"]], RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: computeTimeBetween
# -----
# This is function 28 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute computeTimeBetween. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext

```

```

# Use example list delivered with the package
RELEVANTVN_REF <- RELEVANTVN_REFext
intoleranceDf <- data.frame(prompt = c(2, 3, 4, 1, 1),
expect = c(1, 1, 1, 2, 3))
# expectedDf is a raw ESM dataset, delivered with the package.
intoLs <- intolerable(expectedDf, intoleranceDf, RELEVANTINFO_ES)
randSels <- randomMultSelection(intoLs[["cleanedDf"]])
# Prerequisites in order to execute computeTimeBetween. End -----
# -----
# Run function 28 of 29; see esmprep functions' hierarchy.
# -----
# randSels[["esRandSelIn"]] is the result of function 'randomMultSelection'.
tbsqDf <- computeTimeBetween(randSels[["esRandSelIn"]], referenceDfNew, RELEVANTVN_ES,
RELEVANTVN_REF)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# FUNCTION NAME: esFinal
# -----
# This is function 29 of 29 in the hierarchy of 'esmprep' functions.
# -----
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esFinal. Start -----
# RELEVANTINFO_ES is delivered with the package
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# tbsqDf is a raw ESM dataset, delivered with the package.
# Prerequisites in order to execute esFinal. End -----
# -----
# Run function 29 of 29; see esmprep functions' hierarchy.
# -----
# tbsqDf is the result of function 'computeTimeBetween'.
esDfFin <- esFinal(tbsqDf, esOpt=esAssigned[["ESopt"]], complianceRate=50,
RELEVANTINFO_ES, RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
## End(Not run)

```

---

esPlausible

*esPlausible*

---

## Description

esPlausible helps detecting implausibilities in the raw ESM datasets.

## Usage

```
esPlausible(dfList, itemVecList)
```

## Arguments

**dfList** a list. Each element of the list must be a data.frame. Each data.frame is a separate raw ESM dataset/an ESM questionnaire version. If there is just one ESM version the list therefore contains one data.frame.

`itemVecList` a list. Each list element must be a vector. Each vector element must contain all the variable names of the respective ESM questionnaire version. Use function `esItems` to generate `itemVecList`.

**Details**

`esPlausible` makes no sense if there is only one ESM questionnaire version. However, if there is more than one version it makes sense. The list which is returned by `esPlausible` contains 4 elements:

1. `PlausibNames` shows which item names occur in the different ESM questionnaire versions
2. `plausibClass` shows the class of each item in each of the different ESM questionnaire versions (R built-in class registration)
3. `plausibRowNa` shows the number of rows in each of the different ESM questionnaire versions and the percentage of existing data therein
4. `plausibMinMax` shows the minimum and the maximum value for all items containing numeric data.

**Value**

A list of dataframes, each containing separate information regarding the plausibility of the ESM datasets. See **Details** for more information.

**See Also**

Exemplary code (fully executable) in the documentation of `esmprep` (function 12 of 29).

**Examples**

```
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute esPlausible. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)
plausibItems <- esItems(dfList=rmInvLs[["dfValid"]], RELEVANTVN_ES)
# Prerequisites in order to execute esPlausible. End -----
# -----
# Run function 12 of 29; see esmprep functions' hierarchy.
# -----
# Help checking the plausibility of items in the raw ESM datasets. rmInvLs[["dfValid"]]
# is one of the results from function 'rmInvalid'.
# plausibItems is the result of function 'esItems'.
plausibLs <- esPlausible(dfList=rmInvLs[["dfValid"]], itemVecList=plausibItems)
# Display the results (4 data frames) to the console
# plausibNames gives an overview of the item names across all ESM versions
plausibLs[["plausibNames"]]
# plausibClass gives an overview of the variable types of all items
plausibLs[["plausibClass"]]
# plausibRowNa shows for each ESM version the number of lines in the raw
# ESM datasets and how much percent of the data is missing (NAs).
```



**Examples**

```
# Display the whole dataset in the console
eveningControl
```

---

eveningTest	<i>Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the evening by participants of the test group.</i>
-------------	---

---

**Description**

Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the evening by participants of the test group.

**Usage**

```
eveningTest
```

**Format**

A data frame with 14 rows and 24 variables:

- V1. Arbitrary item, answer format numeric, ranging from 0 to 3.
- V1\_1. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V1\_2. Arbitrary item, answer format text.
- V2. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V2\_1. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3\_1. Arbitrary item, answer format text.
- V4. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V4\_1. Arbitrary item, answer format numeric, ranging from 0 to 4.
- V5. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V5\_1. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V5\_2. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V7. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V8. Arbitrary item, answer format text.
- V8\_1. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V8\_3. Arbitrary item, answer format numeric, ranging from 1 to 4.
- V9. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V9\_1. Arbitrary item, answer format text.
- survey\_name. Name of the ESM version.

- IMEI. IMEI number of the mobile device, used by the participant.
- start\_date. Date of when a single ESM questionnaire was started.
- start\_time. Time of when a single ESM questionnaire was started.
- end\_date. Date of when a single ESM questionnaire was ended.
- end\_time. Time of when a single ESM questionnaire was ended.

### Examples

```
# Display the whole dataset in the console
eveningTest
```

---

expectedDf	<i>Resulting dataset after function <code>expectedPromptIndex</code> has been executed on the raw ESM dataset.</i>
------------	--

---

### Description

**Note:** The dataset `expectedDf` is the result of the function `expectedPromptIndex`, when the raw ESM dataset `madeShift$esDf` is one of the function arguments. `madeShift$esDf` is one of the results of function `makeShift`.

### Usage

```
expectedDf
```

### Format

An object of class `data.frame` with 225 rows and 54 columns.

---

<code>expectedPromptIndex</code>	<i><code>expectedPromptIndex</code></i>
----------------------------------	---

---

### Description

```
expectedPromptIndex
```

### Usage

```
expectedPromptIndex(esDf, expectedPromptIndexList,
  RELEVANTINFO_ES = NULL, RELEVANTVN_ES = NULL)
```



**Arguments**

- `esDf` a data.frame. A single ESM dataset. It must contain the 2 columns that hold the date-time object for when an ESM questionnaire was started and finished, respectively.
- `expectedPromptIndexList` a list. Each list element must also be a list. Each of these inner lists must contain exactly 3 elements: first a character string specifying the ESM questionnaire version; second a vector of at least one integer specifying the daily prompt(s) that the first argument's version correspond(s) to, and third an integer specifying the time of day that the first argument's version corresponds to.
- `RELEVANTINFO_ES` a list. This list is generated by function `setES`.
- `RELEVANTVN_ES` a list. This list is generated by function `setES` and it is extended once either by function `genDateTime` or by function `splitDateTime`.

**Details**

The return values directly correspond to the elements within the second argument: If there are prompt indices (as computed within the function 'esAssign') diverge from the expected prompt indices the variable `PROMPTFALSE`'s value is 1, else 0. The variable `EXPCATEGORY` corresponds to category, with which the user expects the respective prompt(s) to be combined with.

**Value**

`esDf` with the additional columns `PROMPTFALSE` and `EXPCATEGORY`. See **Details** for more information.

**See Also**

Exemplary code (fully executable) in the documentation of `esmprep` (function 23 of 29).

**Examples**

```
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute makeShift. Start -----
# Use example list delivered with the package
RELEVANTINFO_ES <- RELEVANTINFO_ES
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# Use example list delivered with the package
RELEVANTVN_REF <- RELEVANTVN_REFext
# esAssigned is a list of datasets, delivered with the package. It is
# the result of the assignment of the ESM questionnaires to ALL 8
# participants in the reference dataset.
noEndDf <- missingEndTime(esAssigned[["ES"]], RELEVANTVN_ES)
identDf <- esIdentical(noEndDf, RELEVANTVN_ES)
sugShift <- suggestShift(identDf, 100, RELEVANTINFO_ES, RELEVANTVN_ES)
keyPromptDf <- sugShift$suggestShiftDf[,c("NEW_PROMPT", "SHIFTKEY")]
madeShift <- makeShift(sugShift, referenceDfNew, keyPromptDf, RELEVANTINFO_ES, RELEVANTVN_REF)
# Prerequisites in order to execute makeShift. End -----
```

```

# -----
# Run function 23 of 29; see esmprep functions' hierarchy.
# -----
# Generate second argument of function 'expectedPromptIndex'. It's strongly
# recommended to read the explanation of this 2nd argument in the esmprep
# vignette, function 'expectedPromptIndex'.
expIdxList <- list(
  # I - the user - expect in the ESM version morningTestGroup that
  # prompt no.1 is always linked to the value 1.
  list("morningTestGroup", 1, 1),
  # I - the user - expect in the ESM version dayTestGroup that
  # prompt no. 2 and no.3 are always linked to the value 2.
  list("dayTestGroup", c(2, 3), 2),
  # Information of all further ESM versions are passed accordingly:
  list("eveningTestGroup", 4, 3),
  list("morningControlGroup", 1, 1),
  list("dayControlGroup", c(2,3), 2),
  list("eveningControlGroup", 4, 3))
# madeShiftDf$esDf is part of the output of function 'makeShift', if at
# least one questionnaire was shifted to a neighboring prompt index.
expectedDf <- expectedPromptIndex(madeShift$esDf, expIdxList, RELEVANTINFO_ES,
RELEVANTVN_ES)
# If no questionnaire is suggested for shifting (see function suggestShift)
# use the result of function suggestShift as 1st argument, like this:
# expectedDf <- expectedPromptIndex(sugShift$esDf, expIdxList, RELEVANTINFO_ES,
# RELEVANTVN_ES)
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0

```

---

*findChars**findChars*

---

### Description

`findChars` picks all variables from the (merged) ESM dataset that are of the class character.

### Usage

```
findChars(esDf)
```

### Arguments

<code>esDf</code>	a data.frame. A single ESM dataset. It must contain the 2 columns that hold the date-time object for when an ESM questionnaire was started and finished, respectively.
-------------------	--

### Details

`findChars` prints to the console the structure of all variables in `esDf` that contain character values (i.e. text). Among these variables (the index of which is returned by the function) the user can select those that are suitable to apply the function [`convertChars`](#) to.

**Value**

Indices (with corresponding variable names as attributes) of the columns of esDf, containing character strings. See **Details** for more information and see **Examples**.

**See Also**

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 15 of 29).

**Examples**

```
#
# Prerequisites in order to execute findChars. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# isCompleteLs is a list of datasets, also delivered with the package
esMerged <- esMerge(isCompleteLs, RELEVANTVN_ES)
esMerged[,RELEVANTVN_ES[["ES_IMEI"]]] <- as.character(esMerged[,RELEVANTVN_ES[["ES_IMEI"]]])
# Prerequisites in order to execute findChars. End -----
# -----
# Run function 15 of 29; see esmprep functions' hierarchy.
# -----
# esMerged is the result of function 'esMerge'
findTextIdx <- findChars(esMerged)
# Display structure of function output
str(findTextIdx)
#
```

genDateTime

*genDateTime***Description**

genDateTime generates a date-time object from the single components date and time.

**Usage**

```
genDateTime(refOrEsDf = NULL, refOrEs = NULL, RELEVANTINFO_ES = NULL,
            RELEVANTVN_ES = NULL, RELEVANTVN_REF = NULL, dateFormat = "ymd",
            timeFormat = "HMS")
```

**Arguments**

**refOrEsDf** a data.frame. Either the reference dataset or the event sampling raw dataset (already merged to a single dataset).

**refOrEs** a character string. Enter "REF" if the argument refOrEs is the reference dataset, enter "ES" if it is the event sampling dataset.

**RELEVANTINFO\_ES** a list. This list is generated by function [setES](#).







## Details

The user must specify exactly which kind of ESM questionnaires are intolerable relative to the expectation, which was specified in the function [expectedPromptIndex](#). For example, a questionnaire which was expected to be answered in the morning (specified by the expected category 1) must **never** be combined with the last prompt of the day (e.g. 4). That is, it doesn't make sense to have the answers to the morning questionnaire, which was filled out in the evening.

## Value

The user receives a list containing 2 datasets:

1. cleanedDf, i.e. the ESM dataset from which the intolerable lines of data were removed.
2. intoleranceDf, i.e. the intolerable lines of data that were removed.

## See Also

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 24 of 29).

## Examples

```
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute intolerable. Start -----
# RELEVANTINFO_ES is delivered with the package
# expectedDf is a raw ESM dataset, delivered with the package.
# Prerequisites in order to execute intolerable. End -----
# -----
# Run function 24 of 29; see esmprep functions' hierarchy.
# -----
# Generate second argument of function 'intolerable'
(intoleranceDf <- data.frame(
# Column 'prompt': Prompts that must NEVER be comined with expected categories.
prompt = c(2, 3, 4, 1, 1),
# Column 'expect': Expected categories that must NEVER be combined with the prompts.
expect = c(1, 1, 1, 2, 3)))
# Read: Prompts 2, 3, and 4 must never be combined with expected category 1.
# Read: Prompt 1 must never be combined with expected category 2.
# Read: Prompt 1 must never be combined with expected category 3.
# expectedDf is the result of function 'expectedPromptIndex'.
intolLs <- intolerable(expectedDf, intoleranceDf, RELEVANTINFO_ES)
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
```

---

isCompleteLs

*Resulting dataset after function esComplete has been executed on the list of ESM datasets.*

---

## Description

**Note:** The dataset isCompleteLs is the result of the function [esComplete](#), when the list of datasets rmInvLs[["dfValid"]] is one of the function arguments. rmInvLs[["dfValid"]] is one of the results of function [rmInvalid](#).

**Usage**

```
isCompleteLs
```

**Format**

An object of class `list` of length 7.

---

keyLsNew	<i>Resulting dataset after function <code>genDateTime</code> has been executed on the list of ESM datasets.</i>
----------	---

---

**Description**

**Note:** The dataset `keyLsNew` is the result of the function `genDateTime`, when the list of datasets `keyLs` is one of the function arguments.

**Usage**

```
keyLsNew
```

**Format**

An object of class `list` of length 6.

---

makeShift	<i>makeShift</i>
-----------	------------------

---

**Description**

`makeShift` can modify prompts which were assigned by the function `link{esAssign}`.

**Usage**

```
makeShift(esDfShift, refDf, keyPromptDf, RELEVANTINFO_ES = NULL,
          RELEVANTVN_REF = NULL)
```

**Arguments**

<code>esDfShift</code>	a list. Each element of the list must be a <code>data.frame</code> . This argument is generated by <code>suggestShift</code> if at least one ESM questionnaire is eligible for shifting to a neighboring prompt. See <b>Details</b> for more information.
<code>refDf</code>	a <code>data.frame</code> . The reference dataset.



- keyPromptDf a data.frame. The data.frame must have exactly 2 columns. The one column's name must be SHIFTKEY (a numeric value), specifying the exact ESM questionnaire (via the use of the variable KEY). The other column's name must be NEW\_PROMPT, specifying the new prompt index, i.e. the index after the shifting process will be over. See **Details** for more information.
- RELEVANTINFO\_ES a list. This list is generated by function [setES](#).
- RELEVANTVN\_REF a list. This list is generated by function [setREF](#) and it is extended once either by function [genDateTime](#) or by function [splitDateTime](#).

### Details

After applying makeShift the result can be checked by using the function [printSuggestedShift](#) once again. If at least one shifting of the prompt index is suggested, use the additional data.frame, which is then supplied by [suggestShift](#), called 'suggestShiftDf'. Use its 2 columns SHIFTKEY and NEW\_PROMPT to generate the argument 'keyPromptDf' in function makeShift.

### Value

a list with 3 data.frames, if at least one prompt was shifted, i.e.:

1. the first data.frame (called 'esDf') is the raw ESM dataset in its current state (with new column SHIFTED),
2. the second data.frame (called 'suggestShiftDf') includes all relevant information to act according to the decision as to which questionnaires shall be shifted. See **Examples** to get a clear idea of how to use 'suggestShiftDf',
3. the third data.frame (called 'printShiftDf') contains the relevant information to print all questionnaires registered by [suggestShift](#) to the console, before as well as after having made the shifting. This printing to the console is done by applying the function [printSuggestedShift](#).

**Note.** Each questionnaire that got shifted to a neighboring prompt (which prior to the shift has not been assigned), will have its status changed, i.e. values get adapted in variables CV\_ES, ES\_MULT, PROMPT, and PROMPTEND.

### See Also

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 22 of 29).

### Examples

```
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute makeShift. Start -----
# RELEVANTINFO_ES is delivered with the package
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_EStxt
# Use example list delivered with the package
RELEVANTVN_REF <- RELEVANTVN_REFext
# esAssigned is a list of datasets, delivered with the package. It is
# the result of the assignment of the ESM questionnaires to ALL 8
# participants in the reference dataset.
```

```
noEndDf <- missingEndTime(esAssigned[["ES"]], RELEVANTVN_ES)
identDf <- esIdentical(noEndDf, RELEVANTVN_ES)
sugShift <- suggestShift(identDf, 100, RELEVANTINFO_ES, RELEVANTVN_ES)
# Prerequisites in order to execute makeShift. End -----
# -----
# Run function 22 of 29; see esmprep functions' hierarchy.
# -----
# sugShift is the result of function 'suggestShift'. referenceDfNew is the result
# of function 'genDateTime' or of function 'splitDateTime'.
# keyPromptDf is generated by using part of the output of function suggestShift,
# i.e. by selecting the columns NEW_PROMPT and SHIFTKEY from suggestShiftDf.
keyPromptDf <- sugShift$suggestShiftDf[,c("NEW_PROMPT", "SHIFTKEY")]
madeShift <- makeShift(sugShift, referenceDfNew, keyPromptDf, RELEVANTINFO_ES, RELEVANTVN_REF)
# Tip! Display the result of function 'makeShift' in the console
# in order to check whether the shifting was successful.
printSuggestedShift(madeShift, RELEVANTVN_ES)
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
```

---

missingEndTime	<i>missingEndTime</i>
----------------	-----------------------

---

## Description

`missingEndTime` registers all ESM questionnaires with a missing time stamp of when they were finished.

## Usage

```
missingEndTime(esDf, RELEVANTVN_ES = NULL)
```

## Arguments

<code>esDf</code>	a data.frame. A single ESM dataset. It must contain the 2 columns that hold the date-time object for when an ESM questionnaire was started and finished, respectively.
<code>RELEVANTVN_ES</code>	a list. This list is generated by function <code>setES</code> and it is extended once either by function <code>genDateTime</code> or by function <code>splitDateTime</code> .

## Details

Both new columns are dummy variables: 1 = true = end date/end time is missing, 0 = false = end date/end time is present.

## Value

`esDf` with the additional columns `NOENDDATE` and `NOENDTIME`. See **Details** for more information

**See Also**

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 18 of 29).

**Examples**

```
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute missingEndTime. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# esAssigned is a list of datasets, delivered with the package. It is
# the result of the assignment of the ESM questionnaires to ALL 8
# participants in the reference dataset.
# Prerequisites in order to execute missingEndTime. End -----
# -----
# Run function 18 of 29; see esmprep functions' hierarchy.
# -----
# esAssigned[["ES"]] is one of the results of function 'esAssign'.
noEndDf <- missingEndTime(esAssigned[["ES"]], RELEVANTVN_ES)
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
```

---

morningControl	<i>Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the morning by participants of the control group.</i>
----------------	--

---

**Description**

Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the morning by participants of the control group.

**Usage**

```
morningControl
```

**Format**

A data frame with 14 rows and 20 variables:

- V1. Arbitrary item, answer format numeric, ranging from 0 to 3.
- V1\_1. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V1\_2. Arbitrary item, answer format text.
- V2. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V2\_1. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3\_1. Arbitrary item, answer format text.
- V4. Arbitrary item, answer format numeric, ranging from 0 to 1.

- V4\_1. Arbitrary item, answer format numeric, ranging from 0 to 4.
- V5. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V5\_1. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V6. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V6\_1. Arbitrary item, answer format numeric, ranging from 1 to 4.
- V7. Arbitrary item, answer format numeric, ranging from 0 to 1.
- survey\_name. Name of the ESM version.
- IMEI. IMEI number of the mobile device, used by the participant.
- start\_date. Date of when a single ESM questionnaire was started.
- start\_time. Time of when a single ESM questionnaire was started.
- end\_date. Date of when a single ESM questionnaire was ended.
- end\_time. Time of when a single ESM questionnaire was ended.

### Examples

```
# Display the whole dataset in the console
morningControl
```

---

morningTest	<i>Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the morning by participants of the test group.</i>
-------------	---

---

### Description

Raw ESM dataset simulating a series of ESM questionnaires that were scheduled to be filled out during the morning by participants of the test group.

### Usage

```
morningTest
```

### Format

A data frame with 20 rows and 20 variables:

- V1. Arbitrary item, answer format numeric, ranging from 0 to 3.
- V1\_1. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V1\_2. Arbitrary item, answer format text.
- V2. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V2\_1. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3. Arbitrary item, answer format numeric, ranging from 0 to 100.
- V3\_1. Arbitrary item, answer format text.

- V4. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V4\_1. Arbitrary item, answer format numeric, ranging from 0 to 4.
- V5. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V5\_1. Arbitrary item, answer format numeric, ranging from 0 to 6.
- V6. Arbitrary item, answer format numeric, ranging from 0 to 1.
- V6\_1. Arbitrary item, answer format numeric, ranging from 1 to 4.
- V7. Arbitrary item, answer format numeric, ranging from 0 to 1.
- survey\_name. Name of the ESM version.
- IMEI. IMEI number of the mobile device, used by the participant.
- start\_date. Date of when a single ESM questionnaire was started.
- start\_time. Time of when a single ESM questionnaire was started.
- end\_date. Date of when a single ESM questionnaire was ended.
- end\_time. Time of when a single ESM questionnaire was ended.

### Examples

```
# Display the whole dataset in the console
morningTest
```

---

```
printRmInvalid      printRmInvalid
```

---

### Description

printRmInvalid prints to the console the lines that were registered to be invalid in the raw ESM dataset.

### Usage

```
printRmInvalid(rmInvalidList, RELEVANTVN_ES = NULL, smr = "tabulate")
```

### Arguments

**rmInvalidList** a list. Pass the result obtained from function `rmInvalid` as argument.

**RELEVANTVN\_ES** a list. This list is generated by function `setES` and it is extended once either by function `genDateTime` or by function `splitDateTime`.

**smr** a character string. Specify the level of detail you want to be displayed in the console. There are 3 options. The default option is 'tabulate', which displays a table, indicating the number of removed questionnaires for each ESM version. The next option is 'detail', which displays the entire removed questionnaires. The last option is 'both', which displays both the table and the detailed information.

**Value**

In addition to the information the user wants to be displayed (see argument *smr*), the function returns a list containing the **KEY** value (see function [genKey](#)) of all the questionnaires that have been removed from each of the raw ESM datasets.

**See Also**

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 10 of 29).

**Examples**

```
# =====
# Prerequisites in order to execute printRmInvalid. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)
# Prerequisites in order to execute printRmInvalid. End -----
# -----
# Run function 10 of 29; see esmprep functions' hierarchy.
# -----
# rmInvLs is the result of function 'rmInvalid'. Display its result
# in the console both tabulated and in detail.
key_rmLs <- printRmInvalid(rmInvLs, smr="both", RELEVANTVN_ES)
# Display the list containing the KEY values of all questionnaires
# that have been removed.
key_rmLs
# Since there have been warning messages in 4 of the 6 datasets,
# the first ESM item (name: V1) was automatically converted to class
# character, although it is numeric. So we'll re-convert V1's class.
# Check V1 prior to conversion
str(rmInvLs[["dfValid"]][[2]][1:2])
rmInvLs[["dfValid"]] <- sapply(rmInvLs[["dfValid"]], function(x) {
  x[, "V1"] <- as.numeric(x[, "V1"])
  return(x) })
# Check V1 after conversion
str(rmInvLs[["dfValid"]][[2]][1:2])
# =====
```

---

`printSuggestedShift`    *printSuggestedShift*

---

**Description**

`printSuggestedShift` prints to the console which ESM prompts are suggested to be modified.

**Usage**

```
printSuggestedShift(esDfShift, RELEVANTVN_ES = NULL)
```

**Arguments**

esDfShift	a list. Each element of the list must be a data.frame. This argument is generated by <code>suggestShift</code> if at least one ESM questionnaire is eligible for shifting to a neighboring prompt. See <b>Details</b> for more information.
RELEVANTVN_ES	a list. This list is generated by function <code>setES</code> and it is extended once either by function <code>genDateTime</code> or by function <code>splitDateTime</code> .

**Details**

The output to the console shall give the user the necessary information to decide whether lines of data might be shifted and where they shall be shifted to (by altering the values of the variable PROMPT).

**Value**

No return value. See **Details** for more information.

**See Also**

Exemplary code (fully executable) in the documentation of `esmprep` (function 21 of 29).

**Examples**

```
# =====
# Prerequisites in order to execute printSuggestedShift. Start -----
# Use example list delivered with the package
RELEVANTINFO_ES <- RELEVANTINFO_ES
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# esAssigned is a list of datasets, delivered with the package. It is
# the result of the assignment of the ESM questionnaires to ALL 8
# participants in the reference dataset.
noEndDf <- missingEndDateTime(esAssigned[["ES"]], RELEVANTVN_ES)
identDf <- esIdentical(noEndDf, RELEVANTVN_ES)
sugShift <- suggestShift(identDf, 100, RELEVANTINFO_ES, RELEVANTVN_ES)
# Prerequisites in order to execute printSuggestedShift. End -----
# -----
# Run function 21 of 29; see esmprep functions' hierarchy.
# -----
# Display the result of function 'suggestShift' in the console.
printSuggestedShift(sugShift, RELEVANTVN_ES)
# =====
```







---

referenceDfNew	<i>Dataset 'referenceDf' in modified form</i>
----------------	---

---

### Description

**Note:** The dataset `referenceDfNew` is the result of the function `genDateTime`, when the dataset `referenceDf` is one of the function arguments.

### Usage

```
referenceDfNew
```

### Format

An object of class `data.frame` with 8 rows and 12 columns.

### Examples

```
# Convert the 15 digit IMEI number from scientific notation to text.
referenceDfNew$imei <- as.character(referenceDfNew$imei)
# Display the whole dataset in the console
referenceDfNew
```

---

refPlausible	<i>refPlausible</i>
--------------	---------------------

---

### Description

`refPlausible` enables the user to quickly check the reference dataset concerning the expected ESM time period for each participant.

### Usage

```
refPlausible(refDf = NULL, units = "days", RELEVANTVN_REF)
```

### Arguments

<code>refDf</code>	a <code>data.frame</code> . The reference dataset.
<code>units</code>	a character. This character must be exactly one of the following options: <code>auto</code> , <code>secs</code> , <code>mins</code> , <code>hours</code> , <code>days</code> , <code>weeks</code> . For more information see the R base function <code>difftime</code> .
<code>RELEVANTVN_REF</code>	a list. This list is generated by function <code>setREF</code> and it is extended once either by function <code>genDateTime</code> or by function <code>splitDateTime</code> .



---

relevantESVN	<i>relevantESVN</i>
--------------	---------------------

---

### Description

relevantESVN creates a list of the relevant variable names (VN) of (each of) the experience sampling (ES) dataset(s).

### Usage

```
relevantESVN(svyName = NULL, IMEI = NULL, STARTDATE = NULL,
             STARTTIME = NULL, ENDDATE = NULL, ENDTIME = NULL,
             START_DATETIME = NULL, END_DATETIME = NULL)
```

### Arguments

svyName	a vector of character string(s). The column name in (each of) the ESM dataset(s) that specifies the particular survey version. This is the only optional argument, i.e. if there exists no such column in the raw ESM dataset(s), keep the default value of NULL for this argument. If this default value is kept, the function <code>esList</code> will generate such a column and will fill it with 'ESMVERSION_ESMPREP' in each line of raw ESM data at the current state of the dataset(s). <b>NOTE:</b> It is not recommended to let <code>esList</code> generate this column, though. For reasons of clarity it is recommended that the user generates such a column for each raw ESM dataset(s) beforehand. The column name of (all) the raw ESM dataset(s) must be identical, however, the content of the new columns must uniquely specify their respective ESM dataset, e.g. "morningControlGroup" for the morning ESM version of the control group.
IMEI	a character string. Column name that specifies the variable in the respective dataset holding the the IMEI number.
STARTDATE	a character string that specifies the column name in the respective dataset holding the date of when the participant started ... see <b>Details</b> .
STARTTIME	a character string that specifies the column name in the respective dataset holding the time of when the participant started ... see <b>Details</b> .
ENDDATE	a character string that specifies the column name in the respective dataset holding the date of when the participant ended ... see <b>Details</b> .
ENDTIME	a character string that specifies the column name in the respective dataset holding the time of when the participant ended ... see <b>Details</b> .
START_DATETIME	a character string. If STARTDATE and STARTTIME are not in two separate columns but are combined as a date-time object in a single column pass that column's name and ignore the arguments STARTDATE and STARTTIME.
END_DATETIME	a character string. If ENDDATE and ENDTIME are not in two separate columns but are combined as a date-time object in a single column pass that column's name and ignore the arguments ENDDATE and ENDTIME.

## Details

The relevant variable names (i.e. column names) must refer to the raw ESM dataset(s). The date and time of both start and end refers to the single ESM questionnaires, i.e. the date and time they actually were started on the phone and the date and time they were finished.

## Value

A list of the relevant variable/column names in the ESM dataset(s). To be used as the last argument in the function `setES`. The last element of the list is a logical value indicating whether the date and time are either separated (TRUE) or already a date-time object (FALSE). It is meaningful only for the subsequent function `setES`.

## See Also

Exemplary code (fully executable) in the documentation of `esmprep` (function 3 of 29).

## Examples

```
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# -----
# Run function 3 of 29; see esmprep functions' hierarchy.
# -----
# With date and time as separate arguments
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
STARTDATE="start_date", STARTTIME="start_time",
ENDDATE="end_date", ENDTIME="end_time")
# With date-time objects instead of separate date and time
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
START_DATETIME="start_dateTime", END_DATETIME="end_dateTime")
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
```

---

RELEVANTINFO\_ES

*Resulting list after function setES has been executed.*


---

## Description

**Note:** The list RELEVANTINFO\_ES is the result of the function `setES`.

## Usage

```
RELEVANTINFO_ES
```

## Format

An object of class `list` of length 3.

---

relevantREFVN	<i>relevantREFVN</i>
---------------	----------------------

---

### Description

relevantREFVN creates a list of the relevant variable names (VN) of the reference (REF) dataset.

### Usage

```
relevantREFVN(ID = NULL, IMEI = NULL, ST = NULL, STARTDATE = NULL,
  STARTTIME = NULL, ENDDATE = NULL, ENDTIME = NULL,
  START_DATETIME = NULL, END_DATETIME = NULL)
```

### Arguments

ID	a character string that specifies the column name which holds the unique identification code for the participant.
IMEI	a character string. Column name that specifies the variable in the respective dataset holding the the IMEI number.
ST	a character string. It must be the first component of the column name in the reference dataset that specifies the prompts on the mobile device, e.g. ST for scheduled time. The column names must all be equal except for the last character, which must specify the respective number of the prompt, e.g. ST3 for the third prompt of the day.
STARTDATE	a character string that specifies the column name in the respective dataset holding the date of when the participant started ... see <b>Details</b> .
STARTTIME	a character string that specifies the column name in the respective dataset holding the time of when the participant started ... see <b>Details</b> .
ENDDATE	a character string that specifies the column name in the respective dataset holding the date of when the participant ended ... see <b>Details</b> .
ENDTIME	a character string that specifies the column name in the respective dataset holding the time of when the participant ended ... see <b>Details</b> .
START_DATETIME	a character string. If STARTDATE and STARTTIME are not in two separate columns but are combined as a date-time object in a single column pass that column's name and ignore the arguments STARTDATE and STARTTIME.
END_DATETIME	a character string. If ENDDATE and ENDTIME are not in two separate columns but are combined as a date-time object in a single column pass that column's name and ignore the arguments ENDDATE and ENDTIME.

### Details

The relevant variable names (i.e. column names) must refer to the reference dataset. The date and time of both start and end refers to the ESM period for each participant, starting with the date and time he/she fills out the very first ESM questionnaire on his/her own and ending with the date and time he/she fills out the very last ESM questionnaire.

### Value

A list of the relevant variable/column names in the reference dataset. To be used as the last argument in the function `setREF`. The last element of the list is a logical value indicating whether the date and time are either separated (TRUE) or a single date-time object (FALSE). It is meaningful only for the subsequent function `setREF`.

### See Also

Exemplary code (fully executable) in the documentation of `esmprep` (function 1 of 29).

### Examples

```
# =====
# Run function 1 of 29; see esmprep functions' hierarchy.
# =====
# With date and time as separate arguments
relRef <- relevantREFVN(ID="id", IMEI="imei", ST="st",
STARTDATE="start_date", STARTTIME="start_time",
ENDDATE="end_date", ENDTIME="end_time")
# With date-time objects instead of separate date and time
relRef <- relevantREFVN(ID="id", IMEI="imei", ST="st",
START_DATETIME="start_dateTime", END_DATETIME="end_dateTime")
# =====
```

---

RELEVANTVN_ESext	<i>Resulting list after function genDateTime has been executed on the list of ESM datasets.</i>
------------------	---

---

### Description

**Note:** The extended (see ending 'ext') list RELEVANTVN\_ESext is the result of the function `genDateTime`, when the list of datasets `keyLs` is one of the function arguments. The list RELEVANTVN\_ES (before being extended) is generated by the function `setES`.

### Usage

```
RELEVANTVN_ESext
```

### Format

An object of class `list` of length 8.

---

RELEVANTVN_REFext	<i>Resulting list after function <code>genDateTime</code> has been executed on the reference dataset.</i>
-------------------	---

---

### Description

**Note:** The extended (see ending 'ext') list RELEVANTVN\_REFext is the result of the function `genDateTime`, when the reference dataset `referenceDf` is one of the function arguments. The list RELEVANTVN\_REF (before being extended) is generated by the function `setREF`.

### Usage

```
RELEVANTVN_REFext
```

### Format

An object of class `list` of length 9.

---

rmInvalid	<i>rmInvalid</i>
-----------	------------------

---

### Description

`removeInvalid` removes the invalid ESM questionnaires as specified by the user.

### Usage

```
rmInvalid(dfList, RELEVANTVN_ES = NULL)
```

### Arguments

<code>dfList</code>	a list. Each element of the list must be a <code>data.frame</code> . Each <code>data.frame</code> is a separate raw ESM dataset/an ESM questionnaire version. If there is just one ESM version the list therefore contains one <code>data.frame</code> .
<code>RELEVANTVN_ES</code>	a list. This list is generated by function <code>setES</code> and it is extended once either by function <code>genDateTime</code> or by function <code>splitDateTime</code> .

### Details

A data line is assumed to be invalid if both the start date and the start time are missing (NA = not available).



**Value**

The user receives a list containing 4 elements:

1. `dfValid`, i.e. the raw ESM dataset(s), after removing all invalid lines of data.
2. `listInvalid`, i.e. the raw ESM dataset(s), containing only the removed lines of data.
3. `rmInvalidFinished`, i.e. a logical value, specifying whether at least one line of data was removed due to being invalid.
4. `noLinesRemovedAtAll`, i.e. a logical vector, specifying in which of the ESM raw dataset(s) there were invalid lines of data.

See **Details** for more information.

**See Also**

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 9 of 29).

**Examples**

```
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
# Prerequisites in order to execute rmInvalid. Start -----
# Use example list delivered with the package
RELEVANTVN_ES <- RELEVANTVN_ESext
# keyLsNew is a list of datasets, also delivered with the package
# Prerequisites in order to execute rmInvalid. End -----
# -----
# Run function 9 of 29; see esmprep functions' hierarchy.
# -----
# keyLsNew is the result of function 'genDateTime' (or of function 'splitDateTime').
rmInvLs <- rmInvalid(keyLsNew, RELEVANTVN_ES)
# Result of function 'rmInvalid' is a list with 4 elements:
names(rmInvLs)
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
```

---

setES

*setES*


---

**Description**

`setES` sets the relevant variable names concerning the ESM dataset(s).

**Usage**

```
setES(MAXPROMPT = NULL, IMEI_NUMBERS = NULL, SVYNAMES = NULL,
      ESVN = NULL)
```

## Arguments

MAXPROMPT	a numeric value. The number of prompts per day.
IMEI_NUMBERS	a vector of character strings. Each element of the vector specifies an IMEI number of at least one of the phone that have been used in the ESM study.
SVYNAMES	a vector of character strings. Each element of the vector specifies one ESM version, which must exist as a separate column in the raw ESM dataset(s). If in function <a href="#">relevantESVN</a> the default value NULL has been kept for the argument svyName, also keep the default value NULL for this argument! Keeping the default value NULL is not recommended, though. See <b>NOTE</b> in the description of the argument 'svyName' of function <a href="#">relevantESVN</a> .
ESVN	a list. Each element of the list must specify one of the relevant column names of the raw ESM dataset(s); i.e. the ESM survey version, the IMEI number, the start date, the start time, the end date, and the end time. Use function <a href="#">relevantESVN</a> to generate ESVN.

## Details

The one list is named RELEVANTINFO\_ES. It contains 3 elements.

1. MAXPROMPT: the number of daily prompts on the mobile device
2. IMEI\_NUMBERS: all the IMEI numbers that are used in the study
3. SVYNAMES: the names of all ESM questionnaire versions used in the study.

The other list is named RELEVANTVN\_ES, it contains either 4 or 6 elements, depending on whether the start date and the start time already exist as a date-time object (same for end date and end time). In the latter case the list elements' names are:

1. ES\_SVY\_NAME: the column name in the ESM dataset(s) holding the name of the ESM questionnaire version
2. ES\_IMEI: the column name in the ESM dataset(s) holding the IMEI number
3. ES\_START\_DATE: the date of when an ESM questionnaire was started
4. ES\_START\_TIME: the time of when an ESM questionnaire was started
5. ES\_END\_DATE: the date of when an ESM questionnaire was finished
6. ES\_END\_TIME: the time of when an ESM questionnaire was finished

If the start date and start time (same for end date and end time) are combined to a date-time object, the 3rd list element will be ES\_STARTDATETIME and the 4th element will be ES\_ENDDATETIME. The last element will always be ES\_DATETIMES\_SEP: TRUE if date and time are separated, FALSE if they are a single date-time object.

## Value

2 separate lists. Each element of the lists is named according to the variable's content. See **Details** for more information.

## See Also

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 4 of 29).

**Examples**

```

# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute setES. Start -----
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
STARTDATE="start_date", STARTTIME="start_time",
ENDDATE="end_date", ENDTIME="end_time")
# Prerequisites in order to execute setES. End -----
# -----
# Run function 4 of 29; see esmprep functions' hierarchy.
# -----
# imeiNumbers is the vector containing all IMEI numbers used in
# the ESM study; use the respective entries in the referenceDf.
imeiNumbers <- as.character(referenceDf$imei)
# surveyNames is the vector containing all ESM version names.
surveyNames <- c(
# Test group
  "morningTestGroup", "dayTestGroup", "eveningTestGroup",
# Control group
  "morningControlGroup", "dayControlGroup", "eveningControlGroup")
# 4 is the number of daily prompts
# relEs is the result of function 'relevantESVN'
RELEVANT_ES <- setES(4, imeiNumbers, surveyNames, relEs)
# Extract relevant ESM general information
RELEVANTINFO_ES <- RELEVANT_ES[["RELEVANTINFO_ES"]]
# Extract list of relevant variables names of raw ESM datasets.
RELEVANTVN_ES <- RELEVANT_ES[["RELEVANTVN_ES"]]

# With date-time objects instead of separate date and time
relEs <- relevantESVN(svyName="survey_name", IMEI="IMEI",
START_DATETIME="start_dateTime", END_DATETIME="end_dateTime")
RELEVANT_ES <- setES(4, imeiNumbers, surveyNames, relEs)
# Extract relevant ESM general information
RELEVANTINFO_ES <- RELEVANT_ES[["RELEVANTINFO_ES"]]
# Extract list of relevant variables names of raw ESM datasets.
RELEVANTVN_ES <- RELEVANT_ES[["RELEVANTVN_ES"]]
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o

```

---

setREF

*setREF*


---

**Description**

setREF sets the relevant variable names concerning the reference dataset.

**Usage**

```
setREF(MAXPROMPT = NULL, REFVN = NULL)
```

## Arguments

MAXPROMPT	a numeric value. The number of prompts per day.
REFVN	a list. Each element of the list must specify one of the relevant column names of the reference dataset; i.e. the identification number (e.g. ID), the IMEI number, the character substring that all columns have in common which specify the prompt time, the start date, the start time, the end date, and the end time. Use function <a href="#">relevantREFVN</a> to generate REFVN.

## Details

The list is named "RELEVANTVN\_REF". It contains either 5 or 7 elements, depending on whether the start date and the start time already exist as a date-time object (same for end date and end time). In the latter case the list elements' names are:

1. REF\_ID: the column name in the reference dataset holding the name of the unique participant identification code.
2. REF\_IMEI: the column name in the reference dataset holding the IMEI number.
3. REF\_ST: the column name in the reference dataset holding the scheduled times (st)/prompts, except for the numeric end of the column name.
4. REF\_START\_DATETIME: the date-time object of when the very first ESM questionnaire was scheduled/prompted.
5. REF\_END\_DATETIME: the date-time object of when the very last ESM questionnaire was scheduled/prompted.

If the start date and start time (same for end date and end time) are separated, the list elements will be

1. REF\_ID: the column name in the reference dataset holding the name of the unique participant identification code.
2. REF\_IMEI: the column name in the reference dataset holding the IMEI number.
3. REF\_ST: the column name in the reference dataset holding the scheduled times (st)/prompts, except for the numeric end of the column name.
4. REF\_START\_DATE: the date of when the very first ESM questionnaire was scheduled/prompted.
5. REF\_START\_TIME: the time of when the very first ESM questionnaire was scheduled/prompted.
6. REF\_END\_DATE: the date of when the very last ESM questionnaire was scheduled/prompted.
7. REF\_END\_TIME: the time of when the very last ESM questionnaire was scheduled/prompted.

The last element of the output list will always be "REF\_DATETIMES\_SEP": TRUE if date and time are separated, FALSE if they are a single date-time object.

## Value

A list. Each element of the list is named according to the variable's content. See **Details** for more information.

## See Also

Exemplary code (fully executable) in the documentation of [esmprep](#) (function 2 of 29).

**Examples**

```

# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o
# Prerequisites in order to execute setREF. Start -----
relRef <- relevantREFVN(ID="id", IMEI="imei", ST="st",
STARTDATE="start_date", STARTTIME="start_time",
ENDDATE="end_date", ENDTIME="end_time")
# Prerequisites in order to execute setREF. End -----
# -----
# Run function 2 of 29; see esmprep functions' hierarchy.
# -----
# 4 is the number of daily prompts.
# relRef is the result of function 'relevantREFVN'
# Relevant variables names of reference dataset.
(RELEVANTVN_REF <- setREF(4, relRef))
# With date-time objects instead of separate date and time
relRef <- relevantREFVN(ID="id", IMEI="imei", ST="st",
START_DATETIME="start_dateTime", END_DATETIME="end_dateTime")
(RELEVANTVN_REF <- setREF(4, relRef))
# o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o=o

```

splitDateTime

*splitDateTime***Description**

splitDateTime splits a date-time object into its components date and time.

**Usage**

```

splitDateTime(refOrEsDf = NULL, refOrEs = NULL,
RELEVANTINFO_ES = NULL, RELEVANTVN_ES = NULL,
RELEVANTVN_REF = NULL, dateTimeFormat = "ymd_HMS")

```

**Arguments**

refOrEsDf	a data.frame. Either the reference dataset or the event sampling raw dataset (already merged to a single dataset).
refOrEs	a character string. Enter "REF" if the argument refOrEs is the reference dataset, enter "ES" if it is the event sampling dataset.
RELEVANTINFO_ES	a list. This list is generated by function <a href="#">setES</a> .
RELEVANTVN_ES	a list. This list is generated by function <a href="#">setES</a> and it is extended once either by function <a href="#">genDateTime</a> or by function <a href="#">splitDateTime</a> .
RELEVANTVN_REF	a list. This list is generated by function <a href="#">setREF</a> and it is extended once either by function <a href="#">genDateTime</a> or by function <a href="#">splitDateTime</a> .
dateTimeFormat	a character string. Choose the current date-time format, "ymd_HMS" (default), "mdy_HMS", or "dmy_HMS".









```
# Display output element 'printShiftDf':
sugShift$printShiftDf
# 0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0=0
```

---

tbsqDf

*Resulting dataset after function computeTimeBetween has been executed on the raw ESM dataset.*

---

### Description

**Note:** The dataset tbsqDf is the result of the function [computeTimeBetween](#), when the raw ESM dataset durDf is one of the function arguments. durDf is the result of function [computeDuration](#).

### Usage

```
tbsqDf
```

### Format

An object of class `data.frame` with 186 rows and 59 columns.

# Index

## \* datasets

- dayControl, 8
  - dayTest, 9
  - esAssigned, 13
  - esMerged1, 21
  - eveningControl, 38
  - eveningTest, 39
  - expectedDf, 40
  - isCompleteLs, 47
  - keyLsNew, 48
  - morningControl, 51
  - morningTest, 52
  - referenceDf, 57
  - referenceDfNew, 58
  - RELEVANTINFO\_ES, 61
  - RELEVANTVN\_ESext, 63
  - RELEVANTVN\_REFext, 64
  - tbsqDf, 73
- computeDuration, 3, 73
- computeTimeBetween, 4, 73
- computeTimeLag, 5
- convertChars, 6, 13, 21, 42
- dateTimeFormats, 7
- dateTimeFormats2, 8
- dayControl, 8
- dayTest, 9
- diffTime, 58
- esAssign, 10, 13, 16, 17
- esAssigned, 13
- esComplete, 14, 47
- esFinal, 16
- esIdentical, 17
- esItems, 18, 37
- esList, 19, 60
- esMerge, 20, 21
- esMerged1, 21
- esmprep, 3–5, 7, 13, 15, 17–21, 22, 37, 41, 43, 44, 46, 47, 49, 51, 54–56, 59, 61, 63, 65, 66, 68, 70, 72
- esmprep-package (esmprep), 22
- esPlausible, 19, 36
- eveningControl, 38
- eveningTest, 39
- expectedDf, 40
- expectedPromptIndex, 40, 40, 46, 47
- findChars, 42
- genDateTime, 3–5, 11, 16–20, 41, 43, 44, 48–50, 53, 55, 58, 63, 64, 69–71
- genKey, 45, 54
- intolerable, 12, 46
- isCompleteLs, 47
- keyLsNew, 48
- makeShift, 12, 40, 48, 72
- missingEndTime, 50
- morningControl, 51
- morningTest, 52
- printRmInvalid, 53
- printSuggestedShift, 49, 54, 72
- randomMultSelection, 56
- referenceDf, 57, 58, 64
- referenceDfNew, 58
- refPlausible, 58
- relevantESVN, 60, 66
- RELEVANTINFO\_ES, 61
- relevantREFVN, 62, 68
- RELEVANTVN\_ESext, 63
- RELEVANTVN\_REFext, 64
- rmInvalid, 47, 64
- setES, 3–5, 11, 16–20, 41, 43, 44, 46, 49, 50, 53, 55, 61, 63, 64, 65, 69, 71

setREF, [4](#), [11](#), [44](#), [49](#), [58](#), [63](#), [64](#), [67](#), [69](#)  
splitDateTime, [3–5](#), [11](#), [16–20](#), [41](#), [44](#), [49](#),  
[50](#), [53](#), [55](#), [58](#), [64](#), [69](#), [69](#), [71](#)  
suggestShift, [12](#), [48](#), [49](#), [55](#), [71](#)  
tbsqDf, [73](#)