Package ‘PsyToolkit’

March 7, 2026

Version 1.1.4

Title Analysis Tools for 'PsyToolkit'

Description Analyses and reports questionnaire and experiment data exported from 'PsyToolkit'. The package reads downloaded study folders, parses questionnaire structure, optionally merges demographic exports from CloudResearch or Prolific, and produces summary overviews of responses and completion times. It also provides helper functions to extract and aggregate experiment measures and survey variables, and to export results to spreadsheet files for further analysis and archiving. See Stoet (2017) <doi:10.1177/0098628316677643> for the 'PsyToolkit' platform.

Encoding UTF-8

Imports openxlsx2

NeedsCompilation no

Author Gijsbert Stoet [cre, aut] (ORCID: <`https://orcid.org/0000-0002-7557-483X`>)

Maintainer Gijsbert Stoet <`g.stoet@essex.ac.uk`>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2026-03-07 17:10:02 UTC

Contents

`PsyToolkit-package` *PsyToolkit: Analysis Tools for PsyToolkit Data*

Description

Functions to read, summarise, and export PsyToolkit questionnaire and experiment data.

Details

Typical workflow

1. Download PsyToolkit data from your PsyToolkit account (a zip file).
2. Create your R working directory, and within that, a folder `data`.
3. Extract the downloaded PsyToolkit zip file into your new folder `data`.
4. Read the data from the folder with `psytkReadData`.
5. Summarise the data with `psytkReport`.
6. Process data with other functions, such as `psytkExtractFromExperiment`.
7. Export data with `psytkExport`.

Main functions

- `psytkReadData`: Read PsyToolkit data into an R object.
- `psytkReport`: Print a summary report.
- `psytkExport`: Export data to an Excel workbook.
- `psytkTable`: Print formatted tables.
- `psytkExtractFromExperiment`: Extract data for detailed processing.
- `psytkExtractTail`: Extract last line from experiment files.

Other functions

- `psytkParseSurvey`
- `psytkReadTable`
- `psytkRemoveData`
- `psytkQ`
- `psytkExtractCount`
- `psytkExtractNum`

- `psytkExtractPerc`
- `psytkExtractTail`
- `psytkExtractFromExperiment`

Example

A small example object is included for documentation examples: `data(psytkDemo)`. This is not required for normal usage.

`psytkDemo` *Example PsyToolkit Data Object*

Description

A small example object illustrating the structure returned by `psytkReadData()`.

Format

An object used in examples.

Examples

```
data(psytkDemo)
str(psytkDemo, max.level = 1)
```

`psytkExport` *psytkExport*

Description

Exports most of the loaded PsyToolkit data to Excel file.

Usage

```
psytkExport(surveydata , file = "exported_data.xlsx" , labels = FALSE ,
orderByType = FALSE )
```

Arguments

`surveydata`	The PsyToolkit survey data object
`file`	If no filename provided, default exported_data.xlsx will be used
`labels`	Not yet implemented
`orderByType`	Order output by question type instead of order of survey

Value

No returned value.

Examples

```
## Real workflow (requires your own PsyToolkit data files in myfolder)
## Not run:
psytkDemo = psytkReadData( "myfolder" )

## End(Not run)
## runnable example for CRAN checks:
data( psytkDemo ) # or for runnable example

tmp_out_file = tempfile(fileext=".xlsx")

psytkExport( psytkDemo , tmp_out_file )

unlink( tmp_out_file)
```

`psytkExtractCount` *psytkExtractCount*

Description

Find out how for each participant how many cases of condition exist

Usage

```
psytkExtractCount(data , names , conditions)
```

Arguments

`data`	data object
`names`	names of conditions
`conditions`	The actual conditions as logical vectors

Value

A dataframe with for each condition a vector with data for each participant/file.

Examples

```
## Not run:
d = psytkReadData("data")

stroopCorrect = d$expData.stroopForR[,"V7"] == 1 ## correct key pressed
stroopWrong   = d$expData.stroopForR[,"V7"] == 2 ## wrong key pressed
stroopError   = d$expData.stroopForR[,"V7"] != 1 ## wrong key pressed
```

```
                                                ## or no key at all (timeout)
stroopSlow    = d$expData.stroopForR[,"V7"] == 3 ## timeout
con           = d$expData.stroopForR[,"V4"] == 1
inc           = d$expData.stroopForR[,"V4"] == 0
realdata      = d$expData.stroopForR[,"V1"] == "real"

conditionCongruent   = con
conditionIncongruent = inc

basicSelection = stroopCorrect & realdata

subsel = realdata

stroopTrials = psytkExtractCount(
                   data = d$expData.stroopForR ,
                   names = c("congruent","incongruent","all"),
                   conditions = cbind( conditionCongruent & subsel ,
                   conditionIncongruent & subsel  , realdata & subsel ))

## End(Not run)
```

`psytkExtractFromExperiment`

psytkExtractFromExperiment

Description

Extract the data of one participant from one experiment in the PsyToolkit data object

Usage

```
psytkExtractFromExperiment(dataset , expname , id = 1 , removeID = TRUE)
```

Arguments

`dataset`	The name of the PsyToolkit survey data
`expname`	The name of the experiment
`id`	The id, can either be numerical or UUID
`removeID`	Returned data does not contain participant IDs

Value

A data frame with the participant's data.

Examples

```
## Not run:
## imagine your data contains an experiment named "Stroop"
## imagine the Stroop data have as 5th column the reaction time
## this is a way to calculate the mean RT of each participant

d = psytkReadData("data")
meanRTs = numeric( d$n )
for ( i in d$IDcount ){
  tmpdata = psytkExtractFromExperiment( d , "Stroop" , id = i )
  meanRTs = mean( tmpdata[,5] )
}

## End(Not run)
```

`psytkExtractNum` *psytkExtractNum*

Description

Apply function to data each participants' experiment data.

Usage

```
psytkExtractNum(data , dependent , names , conditions , f = mean)
```

Arguments

`data`	The experiment data frame
`dependent`	The column in the experiment data of interest
`names`	The names of the conditions
`conditions`	For each name of a condition a vector of logical selecting data
`f`	A function such as mean (default), median, min, max

Value

Returns a dataframe with one column for each condition.

Examples

```
## Not run:
d = psytkReadData("data")
stroopCorrect = d$expData.stroopForR[,"V7"] == 1 ## correct key pressed
stroopWrong   = d$expData.stroopForR[,"V7"] == 2 ## wrong key pressed
stroopError   = d$expData.stroopForR[,"V7"] != 1 ## wrong key pressed or
                                                 ## no key at all (timeout)
stroopSlow    = d$expData.stroopForR[,"V7"] == 3 ## timeout
con           = d$expData.stroopForR[,"V4"] == 1
```

```
inc           = d$expData.stroopForR[,"V4"] == 0
realdata      = d$expData.stroopForR[,"V1"] == "real"

conditionCongruent   = con
conditionIncongruent = inc

basicSelection = stroopCorrect & realdata

## the psytkExtractNum function is part of the PsyToolkit library. It
## extracts from the experiment data for each participant one row. So
## you end with a row of RT for the congruent and incongruent
## condition; you can later use this for the t test.

stroopRT = psytkExtractNum(
                data = d$expData.stroopForR ,
                dependent = 10 ,
                names = c("congruent","incongruent"),
                conditions = cbind( conditionCongruent   & basicSelection ,
                                    conditionIncongruent & basicSelection ))

## End(Not run)
```

`psytkExtractPerc` *psytkExtractPerc*

Description

For each participant, calculate a percentage based on conditions.

Usage

```
psytkExtractPerc(data , dependent , names , conditions , f = mean)
```

Arguments

`data`	The experiment data frame
`dependent`	The column in the experiment data of interest
`names`	The names of the conditions
`conditions`	For each name of a condition a vector of logical selecting data
`f`	A function such as mean (default), median, min, max

Value

A vector the length of survey IDs in your dataset.

Examples

```
## Not run:
d = psytkReadData("data")
stroopCorrect = d$expData.stroopForR[,"V7"] == 1 ## correct key pressed
stroopWrong   = d$expData.stroopForR[,"V7"] == 2 ## wrong key pressed
stroopError   = d$expData.stroopForR[,"V7"] != 1 ## wrong key pressed
                                                 ## or no key at all (timeout)
stroopSlow    = d$expData.stroopForR[,"V7"] == 3 ## timeout
con           = d$expData.stroopForR[,"V4"] == 1
inc           = d$expData.stroopForR[,"V4"] == 0
realdata      = d$expData.stroopForR[,"V1"] == "real"

conditionCongruent   = con
conditionIncongruent = inc

basicSelection = stroopCorrect & realdata

## the psytkExtractNum function is part of the PsyToolkit library. It
## extracts from the experiment data for each participant one row. So
## you end with a row of RT for the congruent and incongruent
## condition; you can later use this for the t test.

stroopRT = psytkExtractNum( data = d$expData.stroopForR ,
              dependent = 10 ,
              names = c("congruent","incongruent"),
              conditions = cbind( conditionCongruent   & basicSelection ,
              conditionIncongruent & basicSelection ) )

## the psytkExtractPerc function is part of the PsyToolkit library.

## It extracts from the experiment data for each participant one
## row. It calculates the Percentage of Errors (PE) for each
## participant. This is useful for two reasons: 1) You can use it to
## calculate the Stroop effect in error rates. 2) You can use it to
## find out if some participants performed extremely poorly (for
## example at chance level)

stroopPE = psytkExtractPerc(
              data = d$expData.stroopForR ,
              dependent = !stroopCorrect ,
              names = c("congruent","incongruent"),
              conditions = cbind( conditionCongruent & realdata ,
              conditionIncongruent & realdata ) )

## End(Not run)
```

`psytkExtractTail` *psytkExtractTail*

Description

Extracts a value from each participants experiment data

Usage

```
psytkExtractTail(data , dependent)
```

Arguments

`data`	The experiment data (not the survey object)
`dependent`	The column of the table you are interested in.

Value

A vector of values as long as there are files/participants/sessions.

Examples

```
## Not run:
## assume there is an experiment Stroop and the 5th column of this file
## is the reaction time

d = psytkReadData("data")
myExpData = d$expData.Stroop
speed = psytkExtractTail( myExpData , 5 )

## End(Not run)
```

`psytkParseSurvey` *psytkParseSurvey*

Description

Parses the survey.txt file in your data file. Rarely needed on its own. Is used by psytkReadData

Usage

```
psytkParseSurvey(datadir = NA , surveyfilename = NA)
```

Arguments

`datadir`	The folder where you extracted your PsyToolkit data zip file
`surveyfilename`	The name of the survey if not survey.txt

Value

Returns a survey objects giving data about questions and items in it.

Examples

```
## Not run:
mySurvey = psytkParseSurvey( "data" )

## End(Not run)
```

`psytkQ` *psytkQ*

Description

Overview of questionnaire structure

Usage

```
psytkQ( data )
```

Arguments

`data`	The survey data object

Value

Structure of the survey

Examples

```
## Not run:
d = psytkReadData ( "myfolder" )
psytkQ(d)

## End(Not run)
```

`psytkReadData` *psytkReadData*

Description

Load PsyToolkit raw data.

Usage

```
psytkReadData(surveyOrDir , psytkAllowRepeatExp = TRUE , verbose = TRUE)
```

Arguments

`surveyOrDir`	The name of the folder where you unzipped your PsyToolkit downloaded data file
`psytkAllowRepeatExp`	If more than one version of the same experiment available, the last will be used
`verbose`	With TRUE you get detailed information about each datafile.

Details

This is the start of any workflow for working with PsyToolkit data.

Value

Returns on object holding all the data of the survey, including experiment data if available.

Examples

```
## Not run:
d = psytkReadData("myfolder")

## End(Not run)
```

`psytkReadTable` *psytkReadTable*

Description

Read any textfile containing data. Idea for textfiles with unequal row lengths.

Usage

```
psytkReadTable(filename)
```

Arguments

`filename`	The name of the text file to be rad

Details

The benefit of this specific function over read.table() is that it can read datafiles with really different numbers of items per row easily. That can be helpful for reading in experiment data.
This function is rarely needed for end users.

Value

A data frame

Examples

```
## Not run:
z = psytkReadTable( "mydata.txt" )

## End(Not run)
```

`psytkRemoveData` *psytkRemoveData*

Description

Allows you to remove participant datafiles from a PsyToolkit survey data object.

Usage

```
psytkRemoveData(psytkSurveyData , selection)
```

Arguments

`psytkSurveyData`	The object holding the PsyToolkit survey data
`selection`	A logical vector as long as the number of participants/files used

Value

Returns a new object without the removed data. This includes removal from the participant data

Examples

```
## Not run:
## example of removing participant with ID number lower than 10 from all data

dataset1 = psytkReadData("data")
selection = rep( T , dataset$n )
selection[ d$IDcount < 10 ] = F
dataset2 = psytkRemoveData( d , selection )

## End(Not run)
```

`psytkReport` *Create a Summary Report for PsyToolkit Data object*

Description

Creates a textual summary report for PsyToolkit questionnaire data.

Usage

```
psytkReport(psytkSurveyData, textbox = FALSE, question = TRUE , print =
TRUE )
```

Arguments

`psytkSurveyData`	A PsyToolkit survey data object (typically returned by `psytkReadData(foldername)`).
`textbox`	Logical; whether to include summaries for textbox items. Optional, because these can sometimes be long.
`question`	Logical; whether to include question text from q: lines when available.
`print`	Logical; whether or not the report is shown in terminal.

Value

Returns a report object, which can be shown with function print.

Examples

```
## Real workflow (requires your own PsyToolkit data files in myfolder)
## Not run:
d = psytkReadData( "myfolder" )

## End(Not run)
## runnable example for CRAN checks:
data(psytkDemo) # or for runnable example

myReport = psytkReport( psytkDemo ) # this will show the report in the terminal
print( myReport ) # you can print the report later as well
```

`psytkTable` *psytkTable*

Description

Present R data frame or R matrix in a nice table for R terminal output

Usage

```
psytkTable(x, header = NULL , align = "L" , padding=2  , wrap = 20)
```

Arguments

`x`	A data frame or matrix with some data.
`header`	A header above the content of the table.
`align`	L, C, or R for left, center, or right alignment for all or vector of each column
`padding`	Spacing between columns, default is 2
`wrap`	Wrap cells if they have more than 20 characters

Value

This function simply outputs to terminal screen.

Examples

```
## Not run:
x = data.frame( items=1:3,names=c("John","Claudia","Mel") )
psytkTable( x , header=TRUE )

## End(Not run)
```

Index