Package: kobocruncher (via r-universe)

September 27, 2024

Title A Metapackage For Survey Data Crunching

Version 0.2.6

Description An organized workflow generating 'Rmd' files from an extended 'xlsform' questionnaire structure to facilitate survey data crunching.

License GPL-3

URL https://Edouard-Legoupil.github.io/kobocruncher/

BugReports https://github.com/Edouard-Legoupil/kobocruncher/issues

Imports config, cowplot, data.table, datawizard, dplyr, forcats, fs, ggplot2, ggwordcloud, glue, golem, here, httr, jsonlite, likert, magrittr, openxlsx, purrr, readxl, riddle, rmarkdown, scales, shiny, shinydashboard, shinyWidgets, showtext, SnowballC, stats, stringr, sysfonts, systemfonts, tibble, tidyr, tidyselect, tidyverse, tm, unhcrdown, unhcrshiny, utils, XlsFormUtil

Suggests knitr, testthat

VignetteBuilder knitr

Remotes edouard-legoupil/riddle, edouard-legoupil/unhcrshiny, unhcr-americas/XlsFormUtil, unhcr-dataviz/unhcrdown, unhcr-dataviz/unhcrthemes

Config/fusen/version 0.5.2

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3.9000

Repository https://humanitarian-user-group.r-universe.dev

RemoteUrl https://github.com/Edouard-Legoupil/kobocruncher

RemoteRef HEAD

RemoteSha 1150e11ac7b8ff1f450c92b4d189dfccfe3edb4e

Contents

kobo_anonymise . . . 2
kobo_cruncher . . . 3
kobo_data . . . 4
kobo_dico . . . 5
kobo_frame . . . 5
kobo_indicator . . . 6
kobo_likert . . . 9
kobo_prepare_form . . . 9
kobo_ridl . . . 11
label_choiceset . . . 12
label_varhint . . . 13
label_varname . . . 14
plot_correlation . . . 14
plot_header . . . 15
plot_integer . . . 16
plot_integer_cross . . . 17
plot_likert . . . 18
plot_select_multiple . . . 19
plot_select_multiple_cross . . . 20
plot_select_one . . . 21
plot_select_one_cross . . . 22
plot_text . . . 23
run_app . . . 24
template_1_exploration . . . 25

Index **27**

`kobo_anonymise` *Assess Statistical disclosure risk based on an intrusions scenario*

Description

When personal data is being collected, performing basic de-identification (i.e. removal of direct identifiers) and assessing risk of re-identification (i.e. using indirect identifiers to re-identify individuals) is a key sep to perform in order to be able to share the data with multiple analyst.

The initial step consist in defining potential intrusion scenario. This suppose to document the anonymise cell for each variable

Type	
Direct_identifier	Can be directly used to identify an individual. E.g. Name,
Quasi_identifier	Can be used to identify individuals when it is joined with other information. E.
Sensitive_information	& Community identifiable information Might not identify an individual but could put an individual

Direct identifiers will be automatically removed from the data. The function will perform the measurement of various statistical disclosure risk measurement for the selected quasi_identifier and sensitive_information.

Usage

```
kobo_anonymise(datalist, dico)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
dico	An object of the "kobodico" class format as defined in kobocruncher

Examples

```
# dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
# datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )
#
# kobo_anonymise(datalist = datalist,
#                    dico = dico,
#                 indicatoradd = indicatoradd   )
```

---

kobo_cruncher *Crunch all variables according to the analysis plan*

Description

Crunch all variables according to the analysis plan

Usage

```
kobo_cruncher(
  datalist = datalist,
  datasource = NULL,
  dico = dico,
  n = 5,
  n_by = 5
)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
datasource	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsfor
dico	path to the xlsform file used to collect the data
n	if not NULL, lumps all levels except for the n most frequent (or least frequent if n < 0) - cf forcats::fct_lump_n()
n_by	if not NULL, lumps all levels for the cross tabulation variable except for the n_by most frequent (or least frequent if n < 0) - cf forcats::fct_lump_n()

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

kobo_cruncher(datalist = datalist,
              dico = dico,
              datasource = "a great survey!")
```

---

kobo_data *Data loading*

Description

Data loading

Usage

```
kobo_data(datapath)
```

Arguments

datapath	path to the file with the data format as extracted from kobo with dot as group separator and xml header

Value

A "datalist" S3 class object (list) formatted to the specifications of "kobocruncher".

Examples

```
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )
# MainFrame
datalist[["main"]]
# Second Frame - based on presence of repeat within the form, aka nested or
# hierarchical data structure, etc...
datalist[["members"]]
```

kobo_dico *Prepare Analysis plan*

Description

Prepare Analysis plan

Usage

```
kobo_dico(xlsformpath)
```

Arguments

xlsformpath	path to the (extended) xlsform file used to collect the data

Value

A "kobodico" S3 class object (list) formatted to the specifications of "kobocruncher".

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
# Survey
questions <- as.data.frame(dico[["variables"]])
knitr::kable(utils::head(questions, 10))
# Choices
responses <- as.data.frame(dico[["modalities"]])
knitr::kable(utils::head(responses, 10))
# Settings
metadata <- as.data.frame(dico[["settings"]])
knitr::kable(utils::head(metadata, 10))
# Report ToC
toc <- as.data.frame(dico[["plan"]])
knitr::kable(utils::head(toc, 10))
# Indicator
indicator <- as.data.frame(dico[["indicator"]])
knitr::kable(utils::head(indicator, 10))
```

kobo_frame *get the correct frame for one selected variable - important when having variables within a repeat*

Description

get the correct frame for one selected variable - important when having variables within a repeat

Usage

```
kobo_frame(datalist, dico, var)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
dico	An object of the "kobodico" class format as defined in kobocruncher
var	variable

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

data <- kobo_frame(datalist = datalist,
                   dico = dico,
                   var = "members.sex"   )
knitr::kable(utils::head(data,5))
```

---

kobo_indicator *Apply existing documented indicators and add new ones*

Description

The function goes through steps: 1 - load the indicators, 2 - append the one from inidcatoradd if any, 3 - apply the indicator, i.e. do the calculation, 4 - re-save all the working indicator definition within the extended xlsform 5 - bind the new indicators in the dictionary in order to use the kobo_frame() function for further plotting 6 - rebuild the plan if indicators are allocated to chapter, subchapter

Usage

```
kobo_indicator(
  datalist,
  dico,
  xlsformpath,
  xlsformpathout,
  indicatoradd = NULL,
  showcode = FALSE
)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
dico	An object of the "kobodico" class format as defined in kobocruncher
xlsformpath	path to the (extended) xlsform file used to collect the data

xlsformpathout	path to save the xlsform file with newly added indicators
indicatoradd	a list containing all key information to add a calculated indicator within the analysis plan
showcode	display the code

Value

expanded object that includes both the expanded dico and datalist

Examples

```
xlsformpath <- system.file("sample_xlsform.xlsx", package = "kobocruncher")
xlsformpathout <- paste0(tempdir(),"/", "sample_xlsform_withindic.xlsx")

dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

## Check if we add no indicator
expanded  <- kobo_indicator(datalist = datalist,
                            dico = dico,
                            indicatoradd = NULL ,
                            xlsformpath = xlsformpath,
                            xlsformpathout = xlsformpathout)


## Example 1: Simple dummy filter
indicatoradd <- c(  name =  "inColombia",
                    type = "select_one",
                    label = "Is from Colombia",
                    repeatvar = "main",
  calculation = "dplyr::if_else(datalist[[\"main\"]]$profile.country ==
   \"COL\", \"yes\",\"no\")")

expanded  <- kobo_indicator(datalist = datalist,
                    dico = dico,
                indicatoradd = indicatoradd ,
                xlsformpath = xlsformpath,
                xlsformpathout = xlsformpathout)
## Replace existing
dico <- expanded[["dico"]]
datalist <- expanded[["datalist"]]

## Check my new indicator
table(datalist[[1]]$inColombia, useNA = "ifany")



## Example 2: calculation on nested elements and build an indicator list
indicatoradd2 <- c(  name =  "hasfemalemembers",
              type = "select_one",
              label = "HH has female members ",
              repeatvar = "main",
```

```
              calculation = "datalist[[\"members\"]] |>
                             dplyr::select( members.sex, parent_index) |>
                             tidyr::gather(  parent_index, members.sex) |>
                             dplyr::count(parent_index, members.sex) |>
                             tidyr::spread(members.sex, n, fill = 0)  |>
                            dplyr::select( female)")

indicatorall <- list(indicatoradd, indicatoradd2  )

expanded  <- kobo_indicator(datalist = datalist,
                    dico = dico,
                indicatoradd = indicatorall ,
                xlsformpath = xlsformpath,
                xlsformpathout = xlsformpathout)
## Replace existing
dico <- expanded[["dico"]]
datalist <- expanded[["datalist"]]


## Check my new indicator
table(datalist[[1]]$hasfemalemembers, useNA = "ifany")

#   Example of calculations:
#
#   1. Create a filters on specific criteria
#   'dplyr::if_else(datalist[["main"]]$variable =="criteria", "yes","no")'
#
#
#   2. Ratio between 2 numeric variable
#   'datalist[["main"]]$varnum1 / datalist[["main"]]$varnum2'
#
#
#   3. Calculation on date - month between data and now calculated in months
#   'lubridate::interval( datalist[["main"]]$datetocheck,
#                         lubridate::today()) %/%  months(1)'
#
#   4. Discretization of numeric variable according to quintile
#   'Hmisc::cut2(datalist[["main"]]$varnum, g =5)'
#
#   5. Discretization of numeric variable according to fixed break -
#   for instance case size from integer to categoric
#   'cut(datalist[["main"]]$casesize, breaks = c(0, 1, 2, 3,5,30),
#   labels = c("Case.size.1", "Case.size.2", "Case.size.3",
#   "Case.size.4.5", "Case.size.6.or.more" ), include.lowest=TRUE)'
#
#   6. Aggregate variable from nested frame (aka within repeat) to parent table
#   'datalist[["members"]] |>
#       dplyr::select( members.sex, parent_index) |>
#       tidyr::gather(  parent_index, members.sex) |>
#       dplyr::count(parent_index, members.sex) |>
#       tidyr::spread(members.sex, n, fill = 0)  |>
#       dplyr::select( female)'
```

kobo_likert *Crunch all likert variables according to the analysis plan*

Description

Crunch all likert variables according to the analysis plan

Usage

```
kobo_likert(datalist = datalist, datasource = NULL, dico = dico)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
datasource	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
dico	path to the xlsform file used to colllect the data

Examples

```
dicolikert <- kobo_dico( xlsformpath = system.file("form_likert.xlsx", package = "kobocruncher") )
datalistlikert <- kobo_data(datapath = system.file("data_likert.xlsx", package = "kobocruncher") )

kobo_likert(datalist = datalistlikert,
             dico = dicolikert,
             datasource = "a great survey!")
```

kobo_prepare_form *Prepare XLSform and review Analysis*

Description

Prepare XLSform by adding instructions for the analysis plan and checking that structure and settings are correct. This function open the xlsform - extend if required including the excel formatting, display the analysis plan summary and resave the file at the end.

Once those elements are set up, they will be automatically considered during the automatic crunching phase. An additional worksheet is also created to document the information required for registration on UNHCR CKAN instance http://ridl.unhcr.org

1. Configuration of how questions are grouped together in the report:
    - chapter: by default the crunching report is presented according to the group. Once set, this will replace the original grouping. Only variable defined within a chapter will be displayed in the crunching report. By default chapter will follow the questions sequence - if chapters start with a number that number will overrule the sequence

- subchapter: provides a second level of details below the chapter if sub-chapters start with a number that number will overrule the sequence

2. Configuration for data manipulation:
   - clean: define what variable shall be re-categorized during cleaning - a local copy of all levels will be locally saved in order to do the mapping in excel. When the mapping is available, it will be automatically applied to the data. Can be useful to reduce the number of categories.
   - anonymise: define what variables to consider to statistical disclosure risk measurement and subsequent data treatment
3. Configuration of specific charts, visualization and analysis:
   - disaggregation: define variable to use for visual cross tabulation - functions with "_cross"
   - correlate: define the variable to use to explore statistical association - works under certain restrictions (i.e. between 2 categorical variables only): kobo_correlate
   - cluster: define variable to generate an unsupervised classification (i.e. hierarchical clustering based on multiple correspondance analysis) kobo_cluster
   - predict: define variable to use to generate predictive model, ie.e the target variable and the predictors. kobo_predict
   - score: define the different dimensions of a score - and used the score set up for the choice test different aggregation approaches
   - mappoint, mappoly: define the variable to use to generate maps - kobo_map

In case if those fields do not yet exist, the function will create dummy column for each one. Also, coloring all rows that have type equal to "begin group", "end group", "begin repeat" or "end repeat" for better legibility

Usage

```
kobo_prepare_form(xlsformpath, xlsformpathout, label_language = "", ridl = "")
```

Arguments

`xlsformpath`	The full path and filename of the xlsform to be accessed (has to be xlsx file)
`xlsformpathout`	The full path and filename of the xlsform to be accessed (has to be xlsx file)
`label_language`	Optional if the form used multiple languages, indicate the language to use to prepare the analysis plan - check first in your original file. This is strictly based on what is inside your form for instance `english (en)` , `Español (es)`, or `spanish (es)`. Noe that language encoding description should follow https://xlsform.org/en/#multiple-language-support Do not includ the `::` that comes after the label and that separates it from the language suffix
`ridl`	If available, it will pre-fill the RIDL info through what was already recorded there

Examples

```
kobo_prepare_form(xlsformpath = system.file("form.xlsx", package = "kobocruncher"),
                  xlsformpathout = NULL,
                  label_language = "")
```

kobo_ridl *Archive all crunching files in RIDL*

Description

RIDL is UNHCR instance of a CKAN server and is accessible for UNHCR staff at https://ridl.unhcr.org . It is designed to keep track and document dataset within an organisation.

Usage

```
kobo_ridl(
  ridl,
  datafolder,
  form,
  namethisfile,
  visibility = "public",
  stage = "explo_initial"
)
```

Arguments

ridl	ridl container where the resources should be added
datafolder	folder where the data used by the notebook are stored
form	names of the file with the analysis plan
namethisfile	all files are archived based on the name of notebook you created. The function automatically get the name of the notebook where it is run from, using basename(rstudioapi::getSourceEditorContext()$path)
visibility	can be "public" per default or set to private for obscure reasons..
stage	allow to document your analysis stage • exploration_initial if your crunching is the very initial basic cleaning, relabeling of your data • exploration_advance if your crunching • interpretation_prez • dissemination_story

Details

You conveniently archive there your generated report and save the work you did on a notebook: As you have been working on the data, you want to keep track of it and save your work in a place where it can be useful for other people and avaialble for peer review and quality assessment.

The function saves within the the RIDL container you used to get the data from the following resources:

- the generated report

- the analysis plan, aka the extended xlsform used to record relabeling, clean, indicator creation, question grouping, exploration settings
- the source notebook

The function behavior is the following -

1. Get metadata from the RIDL dataset
2. check if the resources to be uploaded is already shared based on the name
3. if already there update, if not create

The function relies on # install.packages("pak") # pak::pkg_install("edouard-legoupil/riddle")

Value

nothing all analysis files are added as a resources

Examples

```
### Example used for each template
## Time to archive your work once done!!
# namethisfile = basename(rstudioapi::getSourceEditorContext()$path )
# if( params$publish == "yes"){
#   kobo_ridl(ridl = params$ridl,
#             datafolder = params$datafolder,
#             form = params$form,
#             namethisfile =  namethisfile ,
#             visibility =  params$visibility,
#             stage = params$stage) }
```

---

label_choiceset *Get all the choices labels options for a specific variable if available*

Description

This labeling function ia function factory - https://adv-r.hadley.nz/function-factories.html The output of this function is actually a function

Usage

```
label_choiceset(dico, x)
```

Arguments

dico	An object of the "kobodico" class format as defined in kobocruncher
x	variable

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

data <- kobo_frame(datalist = datalist,
                   dico = dico,
                   var = "profile.country"   )

label_choiceset(dico = dico,
                x="profile.country")(data$profile.country)

## Test when there's no dictionnary
data$profile.occupation
label_choiceset(dico = dico,
                x="profile.occupation")(data$profile.occupation)


label_choiceset(dico = dico,
                x="profile.occupation")(data$profile.occupation)
```

`label_varhint` *Get Interpretation hint for a specific variable*

Description

Get Interpretation hint for a specific variable

Usage

```
label_varhint(dico, x)
```

Arguments

`dico`	An object of the "kobodico" class format as defined in kobocruncher
`x`	variable

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )

label_varhint(dico = dico,
              x ="profile.country")
```

label_varname *Get the label for a specific variable*

Description

Get the label for a specific variable

Usage

```
label_varname(dico, x)
```

Arguments

dico	An object of the "kobodico" class format as defined in kobocruncher
x	character with the variable name

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )

label_varname(dico = dico,
              x ="profile.country")
```

plot_correlation *Plotting Correlation*

Description

Perform chisquare test and display results if significant

Usage

```
plot_correlation(
  datalist = datalist,
  dico = dico,
  var,
  by_var,
  datasource = NULL,
  showcode = FALSE
)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
dico	An object of the "kobodico" class format as defined in kobocruncher
var	name of the variable to display
by_var	variable to use for cross tabulation
datasource	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
showcode	display the code

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

plot_correlation(datalist = datalist,
              dico = dico,
              var = "profile.occupation",
              by_var = "profile.country",
              datasource = NULL)
```

---

plot_header *Output Header*

Description

Function to add headings within the crunching report - Headings are by defaults the groups defined in the xlsform - but can be replaced within the analysis plan by chapter and subchapter

Usage

```
plot_header(dico = dico, var)
```

Arguments

dico	path to the xlsform file used to colllect the data
var	name of the variable to display

Value

text formatted as markdown

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )

plot_header( dico = dico,
             var = "profile.profile")

# class(plot_header( dico = dico,
#               var = "profile.profile"))
#
dput(plot_header( dico = dico,
             var = "profile.profile"))
#
message(plot_header( dico = dico,
              var = "profile.profile"))

cat(plot_header( dico = dico,
              var = "profile.profile"))

print(plot_header( dico = dico,
             var = "profile.profile"),
      useSource = FALSE)
```

`plot_integer` *Plotting numeric variable*

Description

Plotting numeric variable

Usage

```
plot_integer(
  datalist = datalist,
  dico = dico,
  var,
  datasource = NULL,
  showcode = FALSE
)
```

Arguments

`datalist`	An object of the "datalist" class as defined in kobocruncher
`dico`	An object of the "kobodico" class format as defined in kobocruncher
`var`	name of the variable to display
`datasource`	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
`showcode`	display the code

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

plot_integer(datalist = datalist,
             dico = dico,
             var = "members.age",
             showcode = TRUE)
```

---

plot_integer_cross *Plotting numeric variable*

Description

Plotting numeric variable

Usage

```
plot_integer_cross(
  datalist = datalist,
  dico = dico,
  var,
  by_var,
  datasource = NULL,
  showcode = FALSE
)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
dico	An object of the "kobodico" class format as defined in kobocruncher
var	name of the variable to display
by_var	variable to use for cross tabulation
datasource	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
showcode	display the code

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

plot_integer_cross(datalist = datalist,
             dico = dico,
             var = "members.age",
             by_var = "members.sex",
             showcode = TRUE)
```

plot_likert *Plotting Likert*

Description

Detect if we have more than 3 questions with the same response options within the same questions group and represent the result using a standard likert plot - build from https://github.com/jbryer/likert

Usage

```
plot_likert(
  datalist = datalist,
  dico = dico,
  scopei,
  list_namei,
  repeatvari,
  datasource = NULL,
  showcode = FALSE
)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
dico	An object of the "kobodico" class format as defined in kobocruncher
scopei	group in which the likert frame are
list_namei	name of the likert option list
repeatvari	name of the frame within the dataset where to look for the data
datasource	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
showcode	display the code

Examples

```
dicolikert <- kobo_dico( xlsformpath = system.file("form_likert.xlsx", package = "kobocruncher") )
datalistlikert <- kobo_data(datapath = system.file("data_likert.xlsx", package = "kobocruncher") )

plot_likert(datalist = datalistlikert,
            dico = dicolikert,
            datasource = NULL,
            scopei =  "group_ei8jz33",
            repeatvari =   "main",
            ## getting the list_name and corresponding label
            list_namei = "yk0td68"
          )
```

plot_select_multiple *Plotting Select multiple variable*

Description

Note that if the column order is set in the xlsform choice part, the variable will be de factor considered as ordinal and the default ordering will not be done based on frequency

Usage

```
plot_select_multiple(
  datalist = datalist,
  dico = dico,
  var,
  datasource = NULL,
  n = NULL,
  showcode = FALSE
)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
dico	An object of the "kobodico" class format as defined in kobocruncher
var	name of the variable to display
datasource	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
n	if not NULL, lumps all levels except for the n most frequent (or least frequent if n < 0) - cf forcats::fct_lump_n()
showcode	display the code

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )


plot_select_multiple(datalist = datalist,
              dico = dico,
              var = "profile.reason",
              datasource = NULL,
              showcode = TRUE
            )

## Displaying the usage of the lumping option..
plot_select_multiple(datalist = datalist,
              dico = dico,
              var = "profile.reason",
```

```
               n = 5,
              datasource = NULL,
               showcode = TRUE
             )

# plot_select_multiple(datalist = datalist,
#               dico = dico,
#               var = "profile.reason1",
#               showcode = TRUE
#             )
```

`plot_select_multiple_cross` *Plotting Select multiple variable with cross tabulation on a second categorical variable*

Description

Note that if the column order is set in the xlsform choice part, the variable will be de factor considered as ordinal and the default ordering will not be done based on frequency

Usage

```
plot_select_multiple_cross(
  datalist = datalist,
  dico = dico,
  var,
  by_var,
  datasource = NULL,
  n = NULL,
  n_by = NULL,
  showcode = FALSE
)
```

Arguments

`datalist`	An object of the "datalist" class as defined in kobocruncher
`dico`	An object of the "kobodico" class format as defined in kobocruncher
`var`	name of the variable to display
`by_var`	variable to use for cross tabulation
`datasource`	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
`n`	if not NULL, lumps all levels except for the n most frequent (or least frequent if n < 0) - cf forcats::fct_lump_n()
`n_by`	if not NULL, lumps all levels for the cross tabulation variable except for the n_by most frequent (or least frequent if n < 0) - cf forcats::fct_lump_n()
`showcode`	display the code

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )
plot_select_multiple_cross(datalist = datalist,
              dico = dico,
              var = "profile.reason",
              by_var = "location",
              showcode = TRUE)

## test lumping
plot_select_multiple_cross(datalist = datalist,
              dico = dico,
              var = "profile.reason",
              by_var = "location",
              n = 4,
              showcode = TRUE)
```

---

plot_select_one *Plotting Select one variable*

Description

Note that if the column order is set in the xlsform choice part, the variable will be de factor considered as ordinal and the default ordering will not be done based on frequency

Usage

```
plot_select_one(
  datalist,
  dico,
  var,
  datasource = NULL,
  n = NULL,
  showcode = FALSE
)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
dico	An object of the "kobodico" class format as defined in kobocruncher
var	name of the variable to display
datasource	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
n	if not NULL, lumps all levels except for the n most frequent (or least frequent if n < 0) - cf forcats::fct_lump_n()
showcode	display the code

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

plot_select_one(datalist = datalist,
              dico = dico,
              var = "profile.country",
              showcode = TRUE)
## Exmaple with lumping
plot_select_one(datalist = datalist,
              dico = dico,
              var = "profile.country",
              n = 1,
              showcode = TRUE)

# plot_select_one(datalist = datalist,
#               dico = dico,
#               var = "profile.countryerror",
#               showcode = TRUE)
```

`plot_select_one_cross` *Plotting Select one variable with cross tabulation on a second categorical variable*

Description

Note that if the column order is set in the xlsform choice part, the variable will be de factor considered as ordinal and the default ordering will not be done based on frequency

Usage

```
plot_select_one_cross(
  datalist = datalist,
  dico = dico,
  var,
  by_var,
  datasource = NULL,
  n = NULL,
  n_by = NULL,
  showcode = FALSE
)
```

Arguments

`datalist`	An object of the "datalist" class as defined in kobocruncher
`dico`	An object of the "kobodico" class format as defined in kobocruncher
`var`	name of the variable to display

by_var	variable to use for cross tabulation
datasource	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
n	if not NULL, lumps all levels except for the n most frequent (or least frequent if n < 0) - cf forcats::fct_lump_n()
n_by	if not NULL, lumps all levels for the cross tabulation variable except for the n_by most frequent (or least frequent if n < 0) - cf forcats::fct_lump_n()
showcode	display the code

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

plot_select_one_cross(datalist = datalist,
              dico = dico,
              var = "profile.country",
              by_var = "profile.occupation",
              showcode = TRUE
              )
## test if variable are not in the same frame...
plot_select_one_cross(datalist = datalist,
              dico = dico,
              var = "profile.country",
              by_var = "members.sex",
              n = 5,
              n_by = 5,
              showcode = TRUE
              )
```

---

plot_text *Plotting Open Text variables*

Description

Plotting Open Text variables

Usage

```
plot_text(
  datalist = datalist,
  dico = dico,
  var,
  datasource = NULL,
  showcode = FALSE
)
```

Arguments

datalist	An object of the "datalist" class as defined in kobocruncher
dico	An object of the "kobodico" class format as defined in kobocruncher
var	name of the variable to display
datasource	name of the data source to display, if set to NULL - then pulls the form_title within the settings of the xlsform
showcode	display the code

Examples

```
dico <- kobo_dico( xlsformpath = system.file("sample_xlsform.xlsx", package = "kobocruncher") )
datalist <- kobo_data(datapath = system.file("data.xlsx", package = "kobocruncher") )

plot_text(datalist = datalist,
              dico = dico,
              var = "profile.occupation",
              showcode = TRUE)
```

---

run_app *Run the Shiny Application*

Description

Run the Shiny Application

Usage

```
run_app(
  onStart = NULL,
  options = list(),
  enableBookmarking = NULL,
  uiPattern = "/",
  ...
)
```

Arguments

onStart	A function that will be called before the app is actually run. This is only needed for shinyAppObj, since in the shinyAppDir case, a global.R file can be used for this purpose.
options	Named options that should be passed to the runApp call (these can be any of the following: "port", "launch.browser", "host", "quiet", "display.mode" and "test.mode"). You can also specify width and height parameters which provide a hint to the embedding environment about the ideal height/width for the app.

enableBookmarking
: Can be one of "url", "server", or "disable". The default value, NULL, will respect the setting from any previous calls to enableBookmarking(). See enableBookmarking() for more information on bookmarking your app.

uiPattern
: A regular expression that will be applied to each GET request to determine whether the ui should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful.

...
: arguments to pass to golem_opts. See ?golem::get_golem_options for more details.

Value

a shiny app

Examples

```
# run_app()
```

---

template_1_exploration

Initial Template for Automatic Data Exploration

Description

This template is designed for initial data crunching - The first RMD template gives an output in HTML for easy navigation - the left menu provides smooth transition. It includes a function to automatically run throughout all the survey content. During this stage, data cleaning and new variable creation can be performed through iterations This report also includes each plot syntax so that they can be easily pasted for the second report

Usage

```
template_1_exploration(
  datafolder = "data-raw",
  ridl = "ridlproject",
  data = "data.xlsx",
  form = "form.xlsx",
  datasource = "Study name reference",
  publish = "no",
  republish = "no",
  visibility = "public",
  stage = "exploration_initial",
  language = "",
  folder = "Report"
)
```

Arguments

datafolder	"data-raw" This is the default folder where to put you data in
ridl	"ridlproject" Name of the ridl project where you data is documented and archived
data	"data.xlsx" Name of the data file
form	"form.xlsx" Name of the xlsform -
datasource	"Study name reference" ## String used in caption for all your charts
publish	"no" Put to "yes" in order to add your report, source and analysis plan as ressource within the same ridl c
republish	"no"
visibility	"public"
stage	"exploration_initial" You may change this to exploration_advanced if you configuring many
language	"" Check what you have in your xlsform - ::english (en) -or ::french (fr) or ::spanish (es)
folder	folder within your project where to put the generated report. Folder will be created if it does not exist

Value

nothing the file for the report is generated

Examples

```
# template_1_exploration(datafolder= "data-raw",
#                                    ridl =  "ridlproject",
#                                     data = "data.xlsx" ,
#                                     form =  "form.xlsx",
#                                     datasource = "Study name reference",
#                                     publish =  "no",
#                                     republish = "no",
#                                     visibility = "public",
#                                     stage = "exploration_initial",
#                                    language = "",
#                                    folder = "Report")
```

Index

enableBookmarking(), *25*

kobo_anonymise, 2
kobo_cruncher, 3
kobo_data, 4
kobo_dico, 5
kobo_frame, 5
kobo_indicator, 6
kobo_likert, 9
kobo_prepare_form, 9
kobo_ridl, 11

label_choiceset, 12
label_varhint, 13
label_varname, 14

plot_correlation, 14
plot_header, 15
plot_integer, 16
plot_integer_cross, 17
plot_likert, 18
plot_select_multiple, 19
plot_select_multiple_cross, 20
plot_select_one, 21
plot_select_one_cross, 22
plot_text, 23

run_app, 24

template_1_exploration, 25