

# Package: presenter (via r-universe)

September 12, 2024

**Type** Package

**Title** Present Data with Style

**Version** 0.1.2

**Maintainer** Harrison Tietze <Harrison4192@gmail.com>

**Description** Consists of custom wrapper functions using packages 'openxlsx', 'flextable', and 'officer' to create highly formatted MS office friendly output of your data frames. These viewer friendly outputs are intended to match expectations of professional looking presentations in business and consulting scenarios. The functions are opinionated in the sense that they expect the input data frame to have certain properties in order to take advantage of the automated formatting.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** flextable, magrittr, dplyr, rlang, stringr, purrr, officer, randomcolor, tidyr, openxlsx, lubridate, janitor, tibble, stringi, berryFunctions, rvg, tidyselect, formattable, framecleaner

**RoxygenNote** 7.2.3

**Suggests** knitr, rmarkdown, ggplot2, badger

**VignetteBuilder** knitr

**URL** <https://github.com/Harrison4192/presenter>

**BugReports** <https://github.com/Harrison4192/presenter/issues>

**Repository** <https://harrison4192.r-universe.dev>

**RemoteUrl** <https://github.com/harrison4192/presenter>

**RemoteRef** HEAD

**RemoteSha** 0f8fdd8f5b475d226c23891f1571bc67bc36660c

## Contents

finish_excel_wb . . . . .	2
format_number . . . . .	3
get_piped_name . . . . .	4
is_percentage . . . . .	5
make_excel . . . . .	6
make_excel_wb . . . . .	6
make_flextable . . . . .	7
make_pivot_table . . . . .	8
make_powerpoint . . . . .	9
make_simple_excel . . . . .	9
pivot_summary . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

finish_excel_wb	<i>finish excel workbook</i>
-----------------	------------------------------

---

### Description

finish excel workbook

### Usage

finish\_excel\_wb(wb, wb\_name)

### Arguments

wb	wb
wb_name	wb name

### Value

an excel file

---

format_number	<i>format number</i>
---------------	----------------------

---

## Description

Number formatters to apply to a column in a dataframe. Helpful for displaying tibbles in console or in conjunction with [make\\_flextable](#). Based off the [formattable](#) package.

## Usage

```
format_number(tbl, ..., digits = 0)
```

```
format_percent(tbl, ..., digits = 0)
```

```
format_currency(tbl, ..., symbol = "yen", digits = 0)
```

## Arguments

tbl	dataframe
...	tidyselect.
digits	integer. trailing digits
symbol	chr. currency symbol

## Details

- `format_number` formats a number accounting style by inserting commas. default selection is integer columns
- `format_percent` formats a number as a percentage. default selection is numeric columns in between -1 and 1.
- `format_currency` formats a monetary value with the currency symbol. default currency symbol is yen.

## Value

dataframe  
dataframe

## Examples

```
tibble::tibble(  
  y = seq(1000L, 10000L, by = 1000L),  
  z = c(-.59, -.23, -.11, 0, .1, .21, .3, .4, .6, .9),  
  w = c(.1, 1.4, .23, -.10, 0, -2.3, .2,.3,.4,.5)) -> tbl1  
  
tbl1
```

```

# automatically formats the integer column
tbl1 %>%
  format_number()

# automatically formats to yen
tbl1 %>%
  format_currency(y)

# automatically detects columns between -1 and 1 to convert to percentages
tbl1 %>%
  format_percent()

# select specific columns to convert.
tbl1 %>%
  format_percent(z, w)

```

---

get_piped_name	<i>get_piped_name</i>
----------------	-----------------------

---

## Description

this function captures the name of an object piped into a function, and returns as a string. Powers the automatic naming found in presenter.

## Usage

```
get_piped_name(object, default_name = "Table")
```

## Arguments

object	an object
default_name	string Attempts to return this string if an error occurs.

## Value

string

## Examples

```

#necessary to specify this option when using get_piped_name in knitr
options(rlang_trace_top_env = rlang::current_env())

```

```

### works if the object is piped or given as an argument
iris %>%
  get_piped_name()

get_piped_name(iris)

```

```
### can even extract name from multistep pipes
iris %>%
  dplyr::select(1:3) %>%
  get_piped_name()

### can be placed inside other functions to capture the name and save it

find_name <- function(x){
  get_piped_name() -> new_name

  new_name
}

iris %>%
  dplyr::select(1:3) %>%
  find_name()
```

---

is\_percentage

*is\_percentage*

---

## Description

is\_percentage

## Usage

```
is_percentage(x)
```

## Arguments

x                    numeric vector

## Value

logical

## Examples

```
c(.1, 0, .5) %>%
  is_percentage
```

make\_excel

*Create Excel*

---

**Description**

Create Excel

**Usage**

```
make_excel(  
  df,  
  last_id_col = NULL,  
  header_word = NULL,  
  widths = 13,  
  random_color_seed = 1  
)
```

**Arguments**

df	data frame
last_id_col	index of last id col
header_word	character vector of header words
widths	col widths
random_color_seed	seed for random color scheme

**Value**an excel file

---

make\_excel\_wb

*create excel wb*

---

**Description**

create excel wb

**Usage**

```
make_excel_wb(  
  wb = NULL,  
  object,  
  last_id_col = NULL,  
  header_word = NULL,  
  widths = 13,  
  random_color_seed = 1  
)
```

**Arguments**

wb	wb
object	object
last_id_col	index of last id col
header_word	character vector of header words
widths	col widths
random_color_seed	seed for random color scheme

**Value**

excel wb object

---

make_flextable	<i>Make Flextable</i>
----------------	-----------------------

---

**Description**

Turns a data frame into a flextable

**Usage**

```
make_flextable(
  df,
  header_words = NULL,
  last_id_col = NULL,
  merge_col_indices = NULL,
  dbl_digits = 2,
  theme = c("zebra_blue", "zebra_gold", "tron", "vader", "vanilla", "booktabs",
            "alafoli")
)
```

**Arguments**

df	data frame
header_words	header words. Takes a character vector of header words. will be automatically generate via a heuristic if left NULL. can be completely disabled by the string "disable"
last_id_col	last id col
merge_col_indices	merge specific column indices
dbl_digits	integer. how many trailing digits should be displayed on dbls
theme	string to choose a preselected theme

**Value**

a flextable

---

make_pivot_table	<i>Make pivot table</i>
------------------	-------------------------

---

### Description

If col2 is not supplied, will make a frequency table for 1 column.

### Usage

```
make_pivot_table(  
  tbl,  
  col1,  
  col2 = NULL,  
  show_totals = TRUE,  
  show_percentages = c("none", "all", "row", "col"),  
  show_chi_test = FALSE,  
  theme = c("zebra_blue", "zebra_gold", "tron", "vader", "vanilla", "booktabs",  
            "alafoli"),  
  tbl_nm = NULL,  
  arrange_desc = TRUE  
)
```

### Arguments

tbl	a data frame to pivot
col1	unquoted col 1
col2	unquoted col 2
show_totals	logical; show row and col totals
show_percentages	string; denominator to use when calculating percentages
show_chi_test	logical; show results of chi squared test in footnote
theme	string to choose a predefined theme
tbl_nm	string to name table. If not given, automatically defaults to table name.
arrange_desc	param for single col pivot table. if True arranges table by decreasing n size

### Value

a flextable



---

make_powerpoint	<i>Send Table to Powerpoint</i>
-----------------	---------------------------------

---

**Description**

A table can be piped in to this function to be sent to a new ppt slide. Can be called with no arguments, then a new ppt is created named after the table. If `output_file` is specified, table is sent to a new slide on an existing powerpoint.

**Usage**

```
make_powerpoint(  
  tbl,  
  output_file = NULL,  
  layout = "Two Content",  
  master = "Office Theme",  
  show = TRUE  
)
```

**Arguments**

<code>tbl</code>	a data frame, flextable, or a list thereof
<code>output_file</code>	path to existing ppt
<code>layout</code>	master layout
<code>master</code>	master theme
<code>show</code>	logical to open the ppt

**Value**

none

---

make_simple_excel	<i>Make a simple excel</i>
-------------------	----------------------------

---

**Description**

Export a file to excel with minimal formatting and minimal effort.

**Usage**

```
make_simple_excel(object, show = TRUE)
```

**Arguments**

object	a data frame or list thereof
show	logical. open excel upon completion?

**Value**

an .xlsx file

---

<code>pivot_summary</code>	<i>Pivot Summary</i>
----------------------------	----------------------

---

**Description**

Wrapper around base r 't()' that returns a tibble. Transposes a data frame, intended for use on the output of a `dplyr::summarize` operation

**Usage**

```
pivot_summary(sumr, ...)
```

**Arguments**

sumr	A tibble
...	an optional tidyselect specification of grouping columns to pivot

**Details**

for an ungrouped summarize, ... argument can be left empty for a grouped summarize, use column names or tidyselect to pivot the group names. Multiple groups will be concatenated before pivoting.

**Value**

a tibble

# Index

finish\_excel\_wb, 2  
format\_currency (format\_number), 3  
format\_number, 3  
format\_percent (format\_number), 3  
  
get\_piped\_name, 4  
  
is\_percentage, 5  
  
make\_excel, 6  
make\_excel\_wb, 6  
make\_flextable, 3, 7  
make\_pivot\_table, 8  
make\_powerpoint, 9  
make\_simple\_excel, 9  
  
pivot\_summary, 10