Package 'expstudy'

December 8, 2022

Title Experience Study Tools for Analytics and Communications

Version 1.0.3

Description Provides a data class of 'tbl_es' to help aid in the formation and analyses of recurrent or novel experience studies. A 'tbl_es' has attributes which identify the key variables used for calculating metrics under an actuarial perspective. Common metrics (such as actual-to-expected analysis) can be quickly generated in aggregate or according to different qualitative factors. If multiple factors are of interest, grouped metrics can be automatically computed for each factor individually as well as for all possible combinations. All resulting output can then be formatted for presentations or left unformatted for subsequent analyses. Ultimately, this package aims to reduce time spent completing repetitive code therefore increasing time for analysis and insight.

License MIT + file LICENSE

URL https://github.com/cb12991/expstudy,
https://cb12991.github.io/expstudy/

BugReports https://github.com/cb12991/expstudy/issues

Imports assertthat (>= 0.2.1), checkmate, data.table (>= 1.14.2),
dplyr (>= 1.0.7), dtplyr (>= 1.2.0), glue (>= 1.6.0), lifecycle
(>= 1.0.1), magrittr (>= 2.0.1), methods, pillar (>= 1.6.4),
purrr (>= 0.3.4), rlang (>= 0.4.11), stringr (>= 1.4.0), tibble
(>= 3.1.6), tidyr (>= 1.1.4), tidyselect (>= 1.1.1), utils

Suggests covr (>= 3.5.1), roxygen2 (>= 7.1.2), testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.2

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author Cody Buehler [aut, cre]

R topics documented:

---

    add_credibility   *Add Credibility Factors*

Description

Add credibility factors for an expstudy's expecteds metric variable. The credibility calculation uses a classical credibility approach also known as limited fluctuation partial credibility. Under this approach, the credibility factor is calculated so that actuals are within $k\backslash$ of expecteds with probability p.

Credibility range parameter k and probability level p are set using the function arguments .cred_k and .cred_p, respectively.

Usage

```
add_credibility(
  expstudy,
  .cred_k = 0.05,
  .cred_p = 0.95,
  .cred_nms = "CREDIBILITY"
)
```

Arguments

expstudy	an expstudy
.cred_k	number within range (0, 1); range parameter of credibility equation
.cred_p	number within range (0, 1); probability parameter of credibility equation
.cred_nms	character vector of column names for the added credibility column. If more than one credibility column will be created, you can distinguish them here.

Value

An `expstudy` with added credibility factors.

Examples

```
es <- expstudy(
  data = mortexp,
  actuals = ACTUAL_DEATHS,
  expecteds = EXPECTED_DEATHS,
  exposures =  EXPOSURE,
  variances = VARIANCE_DEATHS
)

 es %>%
   aggregate(ATTAINED_AGE) %>%
   add_credibility

 es %>%
   aggregate(
     UNDERWRITING_CLASS,
     GENDER,
     SMOKING_STATUS
   ) %>%
   add_credibility
```

---

add_metrics   *Add Metrics*

Description

Add commonly used metrics to an [expstudy()](). This typically would be after grouping by variables of interest, but doesn't have to be. The following metrics are available by default and use the expstudy's metric variables (`actuals`, `expecteds`, and`exposures`).

- `'act2expec'`: actuals to expecteds (i.e., A/E ratios)
- `'act2expos'`: actuals to exposures (i.e., average actually observed)
- `'expec2expos'`: expecteds to exposures (i.e., average expected)

You can also create custom metrics to add by providing (optionally named) metric formulae.

Usage

```
add_metrics(
  expstudy,
  ...,
  .metrics = c("act2expec", "act2expos", "expec2expos"),
  .metric_nms = list(act2expec = "ACTUAL_TO_EXPECTED", act2expos = "ACTUAL_TO_EXPOSED",
    expec2expos = "EXPECTED_TO_EXPOSED")
)
```

Arguments

`expstudy`	an `expstudy()`
`...`	additional metrics to add. Can be name-value pairs or simply the metric's formula.
`.metrics`	default methods to add; if omitted, all metrics listed will be added
`.metric_nms`	pairlist of .metric names; can be character vector for `expstudy` object with multiple columns per `.metrics` argument

Value

An `expstudy` object with added metrics. See `expstudy()` for additional detail on `expstudy` objects.

Examples

```
es <- expstudy(
  data = mortexp,
  actuals = ACTUAL_DEATHS,
  expecteds = EXPECTED_DEATHS,
  exposures =  EXPOSURE,
  variances = VARIANCE_DEATHS
)

# If no arguments are provided, all default metrics will be generated. This
# mostly makes sense for already aggregated expstudy objects, but can be
# used with unaggregated objects as well.
 es %>%
   aggregate(ATTAINED_AGE) %>%
   add_metrics

 es %>%
   aggregate(
     UNDERWRITING_CLASS,
     GENDER,
     SMOKING_STATUS
   ) %>%
   add_metrics
```

---

add_proportions	*Add Proportions*

Description

Add proportions of `expstudy` metric variables. Other variables can be provided via the <dynamic-dots> argument. Proportions will sum to 100% per group if `.base_grp_nms` are provided. If `.base_grp_nms` are omitted, proportions will sum to 100% in total.

Usage

```
add_proportions(expstudy, ..., .base_grp_nms = character(0), .min_ungrpd = 0L)
```

Arguments

expstudy	an expstudy()
...	other columns other than the expstudy metric variables (actuals, expecteds, and exposures) to generate proportions of
.base_grp_nms	character vector of column names to use as the base of added proportions
.min_ungrpd	minimum number of non-grouping columns required before using .base_grp_nms as proportion base

Value

An expstudy with added proportions.

Examples

```
es <- expstudy(
  data = mortexp,
  actuals = ACTUAL_DEATHS,
  expecteds = EXPECTED_DEATHS,
  exposures =  EXPOSURE,
  variances = VARIANCE_DEATHS
)

# If no arguments are provided, proportions will be generated for expstudy
# metric variables only. This mostly makes sense for already aggregated
# expstudy objects, but can be used with unaggregated objects as well.
es %>%
  aggregate(
    ATTAINED_AGE
  ) %>%
  add_proportions

# For grouped proportions, use `.base_grp_nms` to identify which variable
# to use as the base for proportions.
es %>%
  aggregate(
    GENDER,
    SMOKING_STATUS
  ) %>%
  add_proportions(
    .base_grp_nms = 'GENDER'
  )

# `.min_ungrpd` is useful only when generating multiple combinations of
# results with compile_results(); this prevents proportions all equaling
# 100% when all grouping columns are used.
```

---

aggregate	*Aggregate an* expstudy

Description

Often an expstudy needs to be aggregated according to different variables for analysis. This function provides easy aggregation of the expstudy metric variables exposures, expected, and actuals.

Grouping can be performed by passing variables as unnamed arguments. If no groups are specified, the grand total of the expstudy metric variables will be returned.

By default, only the metric variables are summed and returned. Other variables can also be totaled by using the .oth_sum_vars to specify.

Usage

```
aggregate(expstudy, ..., .oth_sum_vars = NULL)
```

Arguments

expstudy	an expstudy()
...	variables to use as groups
.oth_sum_vars	additional variable to total. Multiple variables can be provided using c().

Value

an aggregated expstudy.

Examples

```
es <- expstudy(
  data = mortexp,
  actuals = ACTUAL_DEATHS,
  expecteds = EXPECTED_DEATHS,
  exposures =  EXPOSURE,
  variances = VARIANCE_DEATHS
)

es %>%
  aggregate(ATTAINED_AGE)

es %>%
  aggregate(
    GENDER,
    UNDERWRITING_CLASS,
    SMOKING_STATUS
  )
```

compile_results	*Compile resulting analyses*

Description

For a streamlined approach, this function combines multiple other expstudy-package functions to generate multiple analyses simultaneously. All grouping combinations will be generated from variables passed as <dynamic-dots>, then analytics will be created for each.

Resulting output can have the typical metrics via add_metrics(), or proportions via add_proportions(). Default parameters are used for each function, however, you can pass additional parameters in a list to the corresponding function element via the output_args argument.

Furthermore, you can choose to have the results return unformatted (for further calculations or analysis) or formatted (for presentation purposes).

This function was meant to provide quick results for routine analysis. Any additional in-depth analyses should make use of individual expstudy functions instead.

Usage

```
compile_results(
  expstudy,
  ...,
  output = c("metrics", "proportions"),
  output_args = list(metrics = NULL, proportions = NULL),
  output_format = c("unformatted", "formatted")
)
```

Arguments

expstudy	an expstudy()
...	variables to generate grouped analyses. All combinations of variables provided will be generated and used as groups in results.
output	type of result to output (i.e., with metrics or proportions). If omitted, both will be generated and returned in a list.
output_args	additional arguments to pass to add_metrics() and add_proportions(). If omitted, default parameters will be used for both functions.
output_format	resulting output format. If omitted, both will be generated and returned as a list.

Value

a (potentially nested) list with resulting analysis according to arguments passed to the function.

See Also

aggregate() add_metrics() add_credibility() add_proportions() format_metrics()

Examples

```
es <- expstudy(
  data = mortexp,
  actuals = ACTUAL_DEATHS,
  expecteds = EXPECTED_DEATHS,
  exposures =  EXPOSURE,
  variances = VARIANCE_DEATHS
)

# Quickly generate multiple analyses by simply declaring variables of
# interest within an experience study.
es %>%
  compile_results(
    GENDER,
    SMOKING_STATUS,
    UNDERWRITING_CLASS
  )
```

---

expstudy   *Create an expstudy object*

Description

expstudy() creates a subclass, `tbl_es`, of a `dtplyr::lazy_dt()`, that stores attributes relating to
an experience study.These attributes provide other package functions arguments which reduce time
needed to review an assumption.

Usage

```
expstudy(data, actuals, expecteds, exposures, variances = NULL, keys = NULL)
```

Arguments

data	the dataset of an experience study. Can be any kind of organized data (e.g., `base::data.frame()`, `tibble::tibble()`, etc.) but must be able to be converted to a `data.table::data.table()`.
actuals	the observed variable (or variables) within the experience study (e.g., actual lapse counts for a lapse study).
expecteds	the expected variable (or variables) within the experience study (e.g., actual deaths for a mortality study).
exposures	the exposure variable (or variables) within the experience study. This will be the time the records spans in years (or within (0, 1) for records covering less than a year).
variances	the variance variable (or variables) within the experience study. This may not be available for all studies, thus can be omitted. Variance is used for credibility calculations.

keys   variables that uniquely describe every record within the experience study. This typically is policy number and a measure of time, such as annual or monthly duration. Can be omitted, but providing enables quicker binary searches within the `dtplyr::lazy_dt()`. For more information, refer to the data.table vignette: `vignette('datatable-keys-fast-subset', package = 'data.table')`.

Value

an `expstudy`

Examples

```
expstudy(
  data = mortexp,
  actuals = ACTUAL_DEATHS,
  expecteds = EXPECTED_DEATHS,
  exposures =  EXPOSURE,
  variances = VARIANCE_DEATHS
)
```

---

format_metrics   *Format* expstudy *metrics*

Description

Easily format metrics and metric variables for better readability.

Usage

```
format_metrics(expstudy)
```

Arguments

expstudy   an `expstudy` object

Value

the same `expstudy` with formatted metrics

Examples

```
es <- expstudy(
  data = mortexp,
  actuals = ACTUAL_DEATHS,
  expecteds = EXPECTED_DEATHS,
  exposures =  EXPOSURE,
  variances = VARIANCE_DEATHS
)
```

```
# Unformatted result:
es %>%
  aggregate(
    GENDER,
    UNDERWRITING_CLASS
  ) %>%
  add_proportions %>%
  add_metrics %>%
  add_credibility

# Formatted result:
es %>%
  aggregate(
    GENDER,
    UNDERWRITING_CLASS
  ) %>%
  add_proportions %>%
  add_metrics %>%
  add_credibility %>%
  format_metrics
```

---

is_expstudy   *Test if the object is an expstudy*

Description

This function returns TRUE for expstudies or subclasses thereof, and FALSE for all other objects.

Usage

```
is_expstudy(x)
```

Arguments

x   An object

Value

TRUE if the object inherits from the tbl_es class.

---

mortexp *Sample mortality experience*

Description

A dataset containing an example of a mortality experience study for 1000 fictional whole life insurance policyholders.

Usage

```
mortexp
```

Format

A [tibble](#) with over 175,000 rows and 18 variables:

AS_OF_DATE This indicates which point in time a record encompasses.

POLICY_HOLDER An index used to distinguish policyholders. In this example the policyholder is also the (only) insured.

GENDER, SMOKING_STATUS, UNDERWRITING_CLASS, INSURED_DOB, ISSUE_DATE, ISSUE_AGE Various characteristics of insured at time of issue.

ATTAINED_AGE The age of the insured at the record's AS_OF_DATE

DURATION_MONTH, DURATION_YEAR An index describing how long a policy has been in-force at the AS_OF_DATE. For example, when a policy is first issued (i.e., $t = 0$), it is in duration year one, duration month one (i.e.., from $t = 0$ months to $t = 1$ months).

POLICY_STATUS The current status of the policy, either in-force, surrendered, or death. The value will be listed for each policy record even though a decrement only occurs at the end of the policy's duration (for policies which are no longer in-force).

TERMINATION_DATE If terminated, the effective date of termination. An NA value will be listed for policies that are still in-force.

EXPOSURE A measure that reflects how many persons or contracts were exposed to the possibility or risk of the event under study, and for how long. This is calculated by dividing the number of days exposed by the number of days in the calendar year.

ACTUAL_DEATHS The number of actual deaths reported. This will only be 0 or 1 for any given record and it's main purpose is for aggregation.

EXPECTED_MORTALITY_RT An expected mortality rate for an insured. The rate is calculated according to De Moivre's Law (also known as uniform distribution of deaths, or UDD) with $\omega = 120$.

EXPECTED_DEATHS The number of deaths expected for a given record. This is calculated by multiplying exposure by the expected mortality rate, and it provides insight on the expected mortality rate's performance upon aggregation.

VARIANCE_DEATHS The variance of the number of expected deaths, used primarily for assessing the credibility of a subsample's analyses.

Source

All policy record detail is randomly generated. See [https://www.soa.org/globalassets/assets/](https://www.soa.org/globalassets/assets/Files/Research/2016-10-experience-study-calculations.pdf) [Files/Research/2016-10-experience-study-calculations.pdf](https://www.soa.org/globalassets/assets/Files/Research/2016-10-experience-study-calculations.pdf) for additional information regarding experience study calculations.

---

tbl_es-class   tbl_es *class*

Description

The tbl_es class is a subclass of [dtplyr::lazy_dt()](#) created in order to store attributes for experience studies. The colloquial term 'expstudy' refers to a [lazy_dt()](#) that has the tbl_es subclass.

Properties of tbl_es

- Attributes that store variable names used for key metric calculations:
  - actuals: the observed variable (or variables) within the experience study (e.g., actual lapse counts for a lapse study).
  - expecteds: the expected variable (or variables) within the experience study (e.g., expected number of deaths for a mortality study).
  - exposures: the exposure variable (or variables) within the experience study. This will be the time the records spans in years (or within (0, 1) for records covering less than a year).
  - variances: optional, the variance variable (or variables) within the experience study. This may not be available for all studies, thus can be omitted. Variance is used for the credibility metric.
- An additional (optional) keys attribute that can uniquely describe every record within the experience study. This typically is policy number and a measure of time, such as annual or monthly duration. Can be omitted, but providing enables quicker binary searches within the [dtplyr::lazy_dt()](#). For more information, refer to the [data.table](#) vignette: vignette('datatable-keys-fast-subs package = 'data.table').

Behavior of tbl_es

A tbl_es behaves the same as a [dtplyr::lazy_dt()](#) only differing by retaining the attributes listed above. This saves time in routine, repetitive coding, leading to increased time for analysis.

See Also

[dtplyr::lazy_dt](#)

Index