

Package ‘eesim’

July 22, 2025

Type Package

Title Simulate and Evaluate Time Series for Environmental Epidemiology

Version 0.1.0

Date 2017-06-02

Description Provides functions to create simulated time series of environmental exposures (e.g., temperature, air pollution) and health outcomes for use in power analysis and simulation studies in environmental epidemiology. This package also provides functions to evaluate the results of simulation studies based on these simulated time series. This work was supported by a grant from the National Institute of Environmental Health Sciences (R00ES022631) and a fellowship from the Colorado State University Programs for Research and Scholarly Excellence.

License GPL (>= 2)

LazyData TRUE

URL <http://github.com/sakoehler7/eesim>

BugReports <http://github.com/sakoehler7/eesim/issues>

Imports dplyr (>= 0.5.0), lubridate (>= 1.5.6), purrr (>= 0.2.2),
splines, viridis (>= 0.4.0)

RoxygenNote 6.0.1

Suggests dlnm (>= 2.3.2), ggplot2 (>= 2.2.1), gridExtra (>= 2.2.1),
knitr (>= 1.15.1), rmarkdown (>= 1.5.0), tidyr (>= 0.6.2)

VignetteBuilder knitr

NeedsCompilation no

Author Sarah Koehler [aut],
Brooke Anderson [aut, cre]

Maintainer Brooke Anderson <brooke.anderson@colostate.edu>

Repository CRAN

Date/Publication 2017-06-03 17:55:52 UTC

Contents

beta_bias	2
beta_var	3
binary_exposure	4
bin_t	5
calc_t	6
calendar_plot	7
check_sims	8
continuous_exposure	9
coverage_beta	10
coverage_plot	11
create_baseline	12
create_lambda	13
create_sims	14
custom_baseline	17
custom_exposure	18
eesim	19
fit_mods	22
format_out	23
mean_beta	24
power_beta	25
power_calc	25
sim_baseline	28
sim_exposure	29
sim_outcome	31
spline_mod	32
std_exposure	33
Index	35

beta_bias	<i>Percent Bias of Estimated Coefficient</i>
-----------	--

Description

This function returns the relative bias of the mean of the estimated coefficients.

Usage

```
beta_bias(df, true_rr)
```

Arguments

df	A data frame of replicated simulations which must include a column titled "Estimate" with the effect estimate from the fitted model.
true_rr	The true relative risk used to simulate the data.

Details

This function estimates the percent bias in the estimated log relative risk (b) as:

$$100 * \frac{\beta - \hat{\beta}}{\beta}$$

where $\hat{\beta}$ is the mean of the estimated log relative risk values from all simulations and β is the true log relative risk used to simulate the data.

Value

A data frame with a single value: the percent bias of the mean of the estimated coefficients over n_reps simulations.

Examples

```
sims <- create_sims(n_reps = 10, n = 600, central = 100,
                  sd = 10, exposure_type = "continuous",
                  exposure_trend = "cos1",
                  exposure_amp = 0.6,
                  average_outcome = 20,
                  outcome_trend = "no trend",
                  rr = 1.01)
fits <- fit_mods(data = sims, custom_model = spline_mod,
                custom_model_args = list(df_year = 1))
beta_bias(fits, true_rr = 1.02)
```

beta_var

Standard Deviation of Estimated Coefficients

Description

Measures the variance of the point estimates of the estimated log relative risk (\hat{beta}) over the n_rep simulations and the mean of the variances of each $\hat{\beta}$.

Usage

```
beta_var(df)
```

Arguments

df A data frame of replicated simulations which must include columns titled "Estimate" and "Std.Error".

Value

A data frame of the variance across all values of beta hat and the mean variance of the beta hats

Examples

```
sims <- create_sims(n_reps = 10, n = 600, central = 100,
                  sd = 10, exposure_type = "continuous",
                  exposure_trend = "cos1",
                  exposure_amp = 0.6,
                  average_outcome = 20,
                  outcome_trend = "no trend",
                  rr = 1.01)
fits <- fit_mods(data = sims, custom_model = spline_mod,
                 custom_model_args = list(df_year = 1))
beta_var(fits)
```

binary_exposure	<i>Simulate binary exposure data</i>
-----------------	--------------------------------------

Description

Simulates a time series of binary exposure values with or without seasonal trends.

Usage

```
binary_exposure(n, p, trend = "no trend", slope, amp = 0.05,
               start.date = "2000-01-01", cust_expdraw = NULL,
               cust_expdraw_args = list(), custom_func = NULL, ...)
```

Arguments

n	A non-negative integer specifying the number of days to simulate.
p	A numeric value between 0 and 1 giving the mean probability of exposure across study days.
trend	A character string that gives the trend function to use. Options are: <ul style="list-style-type: none"> • "no trend": No trend, either seasonal or long-term (default). • "cos1": A seasonal trend only. • "cos2": A seasonal trend with variable amplitude across years. • "cos3": A seasonal trend with steadily decreasing amplitude over time. • "linear": A linear long-term trend with no seasonal trend. • "monthly": Uses a user-specified probability of exposure for each month.
slope	A numeric value specifying the slope of the trend, to be used with trend = "linear" or trend = "cos1linear".
amp	A numeric value specifying the amplitude of the seasonal trend. Must be between -.5 and .5.
start.date	A date of the format "yyyy-mm-dd" from which to begin simulating daily exposures

cust_expdraw	An R object name specifying a user-created function which determines the distribution of random noise off of the trend line. This function must have inputs "n" and "prob" and output a vector of simulated exposure values.
cust_expdraw_args	A list of arguments other than n required by the cust_expdraw function.
custom_func	An R object specifying a customized function from which to create a trend variable. Must accept arguments n and p.
...	Optional arguments to a custom trend function

Value

A data frame with columns for the dates and daily exposure values for n days.

Examples

```
binary_exposure(n = 5, p = 0.1, trend = "cos1", amp = .02,
                start.date = "2001-02-01")
binary_exposure(n=10, p=.1, cust_expdraw=rnbinom,
                cust_expdraw_args=list(size=10))
```

bin_t

Create a binary exposure trend vector

Description

Creates a trend vector for binary exposure data, centered at a probability p.

Usage

```
bin_t(n, p, trend = "no trend", slope = 1, amp = 0.01,
      start.date = "2000-01-01", custom_func = NULL, ...)
```

Arguments

n	A non-negative integer specifying the number of days to simulate.
p	A numeric value between 0 and 1 giving the mean probability of exposure across study days.
trend	A character string that gives the trend function to use. Options are: <ul style="list-style-type: none"> • "no trend": No trend, either seasonal or long-term (default). • "cos1": A seasonal trend only. • "cos2": A seasonal trend with variable amplitude across years. • "cos3": A seasonal trend with steadily decreasing amplitude over time. • "linear": A linear long-term trend with no seasonal trend. • "monthly": Uses a user-specified probability of exposure for each month.

slope	A numeric value specifying the slope of the trend, to be used with trend = "linear" or trend = "cos1linear".
amp	A numeric value specifying the amplitude of the seasonal trend. Must be between -.5 and .5.
start.date	A date of the format "yyyy-mm-dd" from which to begin simulating values.
custom_func	An R object specifying a customized function from which to create a trend variable. Must accept arguments n and p.
...	Optional arguments to a custom trend function

Value

A numeric vector of daily expected probability of exposure, to be used to generate binary exposure data with seasonal trends.

Examples

```
bin_t(n = 5, p = .3, trend = "cos1", amp = .3)
```

calc_t	<i>Create a continuous exposure trend vector</i>
--------	--

Description

Creates a trend vector for a continuous exposure.

Usage

```
calc_t(n, trend = "no trend", slope = 1, amp = 0.6, custom_func = NULL,
      ...)
```

Arguments

n	A non-negative integer specifying the number of days to simulate.
trend	A character string that specifies the desired trend function. Options are: <ul style="list-style-type: none"> "no trend": No trend, either seasonal or long-term (default). "cos1": A seasonal trend only. "cos2": A seasonal trend with variable amplitude across years. "cos3": A seasonal trend with steadily decreasing amplitude over time. "linear": A linear long-term trend with no seasonal trend. "curvilinear": A curved long-term trend with no seasonal trend. "cos1linear": A seasonal trend plus a linear long-term trend. See the package vignette for examples of the shapes of these trends.
slope	A numeric value specifying the slope of the trend, to be used with trend = "linear" or trend = "cos1linear".

amp	A numeric value specifying the amplitude of the seasonal trend. Must be between -1 and 1.
custom_func	An R object specifying a customized function from which to create a trend variable. Must accept the arguments n and mean.
...	Optional arguments to a custom trend function

Value

A numeric vector of simulated exposure values for each study day, to be used to generate data with seasonal trends.

Examples

```
calc_t(5, "cos3", amp = .5)
```

calendar_plot	<i>Create calendar plot</i>
---------------	-----------------------------

Description

Creates a calendar plot of a time series of continuous or discrete data. The time series data frame input to this function must have only two columns, one for the date and one with the values to plot.

Usage

```
calendar_plot(df, type = "continuous", labels = NULL,
              legend_name = "Exposure")
```

Arguments

df	Data frame with one column named date for date with entries in the format "yyyy-mm-dd" and one column for the daily values of the variable to plot.
type	Character string specifying whether the exposure is continuous or discrete
labels	Vector of character strings naming the levels of a discrete variable to be used in the figure legend.
legend_name	Character string specifying the title to be used in the figure legend.

Details

The output of this function is a ggplot object, so you can customize this output object as with any ggplot object.

Examples

```
testdat <- sim_exposure(n = 1000, central = 0.1,
  exposure_type = "binary")
testdat$X[c(89,101,367,500,502,598,678,700,895)] <- 3
calendar_plot(testdat, type = "discrete", labels = c("no", "yes", "maybe"))
```

 check_sims

Assess model performance

Description

Calculates several measures of model performance, based on results of fitting a model to all simulated datasets.

Usage

```
check_sims(df, true_rr)
```

Arguments

df	A data frame of replicated simulations which must include a column titled "Estimate" with the effect estimate from the fitted model.
true_rr	The true relative risk used to simulate the data.

Value

A dataframe with one row with model assessment across all simulations. Includes values for:

- `beta_hat`: Mean of the estimated log relative risk across all simulations.
- `rr_hat`: Mean value of the estimated relative risk across all simulations.
- `var_across_betas`: Variance of the estimated log relative risk across all simulations
- `mean_beta_var`: The mean of the estimated variances of the estimated log relative risks across all simulations.
- `percent_bias`: The relative bias of the estimated log relative risks compared to the true log relative risk.
- `coverage`: Percent of simulations for which the estimated 95% confidence interval for log relative risk includes the true log relative risk.
- `power`: Percent of simulations for which the null hypothesis that the log relative risk equals zero is rejected based on a p-value of 0.05.

See Also

The following functions are used to calculate these measurements: [beta_bias](#), [beta_var](#), [coverage_beta](#), [mean_beta](#), [power_beta](#)

Examples

```
sims <- create_sims(n_reps = 100, n = 1000, central = 100,
                  sd = 10, exposure_type = "continuous",
                  exposure_trend = "cos1",
                  exposure_amp = 0.6,
                  average_outcome = 20,
                  outcome_trend = "no trend",
                  rr = 1.02)
fits <- fit_mods(data = sims, custom_model = spline_mod,
                custom_model_args = list(df_year = 1))
check_sims(df = fits, true_rr = 1.02)
```

continuous_exposure *Simulate continuous exposure data*

Description

Simulates a time series of continuous exposure values with or without a seasonal and / or long-term trend.

Usage

```
continuous_exposure(n, mu, sd = 1, trend = "no trend", slope, amp = 0.6,
                   cust_expdraw = NULL, cust_expdraw_args = list(),
                   start.date = "2000-01-01", ...)
```

Arguments

n	A non-negative integer specifying the number of days to simulate.
mu	A numeric value giving the mean exposure across all study days.
sd	A numeric value giving the standard deviation of the exposure values from the exposure trend line.
trend	A character string that specifies the desired trend function. Options are: <ul style="list-style-type: none"> • "no trend": No trend, either seasonal or long-term (default). • "cos1": A seasonal trend only. • "cos2": A seasonal trend with variable amplitude across years. • "cos3": A seasonal trend with steadily decreasing amplitude over time. • "linear": A linear long-term trend with no seasonal trend. • "curvilinear": A curved long-term trend with no seasonal trend. • "cos1linear": A seasonal trend plus a linear long-term trend. See the package vignette for examples of the shapes of these trends.
slope	A numeric value specifying the slope of the trend, to be used with trend = "linear" or trend = "cos1linear".

amp	A numeric value specifying the amplitude of the seasonal trend. Must be between -1 and 1.
cust_expdraw	A character string specifying a user-created function which determines the distribution of random noise off of the trend line. This function must have inputs "n" and "mean" and output a vector of simulated exposure values.
cust_expdraw_args	A list of arguments other than "n" and "mean" required by the cust_expdraw function.
start.date	A date of the format "yyyy-mm-dd" from which to begin simulating daily exposures
...	Optional arguments to a custom trend function

Value

A data frame with the dates and simulated daily exposure values from n days.

Examples

```
continuous_exposure(n = 5, mu = 100, sd = 10, trend = "cos1")
continuous_exposure(n=10, mu=3, trend="linear", slope = 2,
                    cust_expdraw=rnorm, cust_expdraw_args = list(sd=.5))
```

coverage_beta	<i>Empirical coverage of confidence intervals</i>
---------------	---

Description

Calculates the percent of simulations in which the estimated 95% confidence interval for the log relative risk includes the true value of the log relative risk.

Usage

```
coverage_beta(df, true_rr)
```

Arguments

df	A data frame of replicated simulations which must include columns titled lower_ci and upper_ci.
true_rr	The true relative risk used to simulate the data.

Value

A data frame with the percent of confidence intervals for the estimated log relative risk over n_reps simulations which include the true log relative risk.

Examples

```
sims <- create_sims(n_reps = 10, n = 600, central = 100,
                  sd = 10, exposure_type = "continuous",
                  exposure_trend = "cos1", exposure_slope=1,
                  exposure_amp = 0.6,
                  average_outcome = 20,
                  outcome_trend = "no trend",
                  rr = 1.01)
fits <- fit_mods(data = sims, custom_model = spline_mod,
                custom_model_args = list(df_year = 1))
coverage_beta(df=fits, true_rr = 1.02)
```

coverage_plot

*Plot coverage of empirical confidence intervals***Description**

Plots the relative risk point estimates and their confidence intervals for model fit results for each simulation, compared to the true relative risk. This gives a visualization of the coverage of the specified method for the relative risk. The confidence intervals which do not contain the true relative risk appear in red. The input to this function should be either the output of `fit_mods` or the second element of the output of `eesim`.

Usage

```
coverage_plot(summarystats, true_param)
```

Arguments

`summarystats` A list or data frame of summary statistics from many repetitions of a simulation. Must include columns titled `Estimate`, `lower_ci`, and `upper_ci`. This could be the second object from the output of `eesim`, specified by using the format `eesim_output[[2]]`.

`true_param` The true value of the relative risk used to simulate the data.

Value

A plot displaying the coverage for the true value of the parameter by the confidence intervals resulting from each repetition of the simulation.

Examples

```
ex_sim <- eesim(n_reps = 100, n = 1000, central = 100, sd = 10,
              exposure_type = "continuous", average_outcome = 20, rr = 1.02,
              custom_model = spline_mod, custom_model_args = list(df_year = 1))
coverage_plot(ex_sim[[2]], true_param = 1.02)
```

create_baseline *Create a series of baseline outcomes*

Description

Creates a time series of baseline outcome values. This function allows the user to input a custom function if desired to specify outcome trend.

Usage

```
create_baseline(n, average_baseline = NULL, trend = "no trend", slope = 1,
  amp = 0.6, cust_base_func = NULL, ...)
```

Arguments

n	A numeric value specifying the number of days for which to simulate data
average_baseline	A non-negative numeric value specifying the average outcome value over all simulated days.
trend	A character string that specifies the desired trend function. Options are: <ul style="list-style-type: none"> • "no trend": No trend, either seasonal or long-term (default). • "cos1": A seasonal trend only. • "cos2": A seasonal trend with variable amplitude across years. • "cos3": A seasonal trend with steadily decreasing amplitude over time. • "linear": A linear long-term trend with no seasonal trend. • "curvilinear": A curved long-term trend with no seasonal trend. • "cos1linear": A seasonal trend plus a linear long-term trend. See the package vignette for examples of the shapes of these trends.
slope	A numeric value specifying the slope of the trend, to be used with trend = "linear" or trend = "cos1linear".
amp	A numeric value specifying the amplitude of the seasonal trend. Must be between -1 and 1.
cust_base_func	A R object name specifying a user-made custom function for baseline trend.
...	Optional arguments to a custom baseline function

Value

A numeric vector of baseline outcome values

Examples

```
create_baseline(n = 5, average_baseline = 22, trend = "linear")
```

create_lambda	<i>Create a series of mean outcome values</i>
---------------	---

Description

Creates a vector of expected daily outcome count by relating exposure to baseline outcome values with the function:

$$\log(\lambda_t) = \log(B_t) + \log(RR) * X_t$$

where λ_t is the expected outcome count on day t , B is the expected base outcome count on day t (incorporating long-term and seasonal trends, but not the influence of the exposure), RR is the relative risk of the outcome for a one-unit increase in exposure, and X_t is the simulated exposure on day t . The user may input a custom function to relate exposure, relative risk, and baseline.

Usage

```
create_lambda(baseline, exposure, rr, cust_lambda_func = NULL, ...)
```

Arguments

baseline	A non-negative numeric vector of baseline outcome values, typically the output of create_baseline .
exposure	A numeric vector of exposure values, typically the output of sim_exposure .
rr	A non-negative numeric value specifying the relative risk (i.e., the relative risk per unit increase in the exposure).
cust_lambda_func	An R object name specifying a user-made custom function for relating baseline, relative risk, and exposure
...	Optional arguments for a custom lambda function

Value

A numeric vector of mean outcome values for each day in the simulation.

Examples

```
base <- create_baseline(n = 10, average_baseline = 22, trend = "linear",
                       slope = .4)
exp <- sim_exposure(n = 5, central = 100, sd = 10, amp = .6,
                   exposure_type = "continuous")
create_lambda(baseline = base, exposure = exp$x, rr = 1.01)
```

 create_sims

Create simulated data for many repetitions

Description

Creates a collection of synthetic datasets that follow a set of user-specified conditions (e.g., exposure mean and variance, average daily outcome count, long-term and seasonal trends in exposure and outcome, association between exposure and outcome). These synthetic datasets can be used to investigate performance of a specific model or to estimate power or required sample size for a hypothetical study.

Usage

```
create_sims(n_reps, n, rr, central, average_outcome, sd = NULL, exposure_type,
  exposure_trend, exposure_slope = 1, exposure_amp = NULL,
  outcome_trend = NULL, outcome_slope = 1, outcome_amp = NULL,
  start.date = "2000-01-01", cust_exp_func = NULL, cust_exp_args = NULL,
  cust_expdraw = NULL, cust_expdraw_args = NULL, cust_base_func = NULL,
  cust_lambda_func = NULL, cust_base_args = NULL, cust_lambda_args = NULL,
  cust_outdraw = NULL, cust_outdraw_args = NULL)
```

Arguments

n_reps	An integer specifying the number of datasets to simulate (e.g., n_reps = 1000 would simulate one thousand time series datasets with the specified characteristics, which can be used for a power analysis or to investigate the performance of a proposed model).
n	An integer specifying the number of days to simulate (e.g., n = 365 would simulate a dataset with a year's worth of data).
rr	A non-negative numeric value specifying the relative risk (i.e., the relative risk per unit increase in the exposure).
central	A numeric value specifying the mean probability of exposure (for binary data) or the mean exposure value (for continuous data).
average_outcome	A non-negative numeric value specifying the average daily outcome count.
sd	A non-negative numeric value giving the standard deviation of the exposure values from the exposure trend line (not the total standard deviation of the exposure values).
exposure_type	A character string specifying the type of exposure. Choices are "binary" or "continuous".
exposure_trend	A character string specifying a seasonal and / or long-term trend for expected mean exposure. See the vignette for eesim for examples of each option. The shapes are based on those used in Bateson and Schwartz (1999). For trends with a seasonal component, the amplitude of the seasonal trend can be customized using the exposure_amp argument. For trends with a long-term pattern, the

slope of the long-term trend can be set using the `exposure_slope` argument. If using the "monthly" option for a binary exposure, you must input a numeric vector of length 12 for the central argument that gives the probability of exposure for each month, starting in January and ending in December. Options for continuous exposure are:

- "no trend": No trend, either seasonal or long-term (default).
- "cos1": A seasonal trend only.
- "cos2": A seasonal trend with variable amplitude across years.
- "cos3": A seasonal trend with steadily decreasing amplitude over time.
- "linear": A linear long-term trend with no seasonal trend.
- "curvilinear": A curved long-term trend with no seasonal trend.
- "cos1linear": A seasonal trend plus a linear long-term trend.

Options for binary exposure are:

- "no trend": No trend, either seasonal or long-term (default).
- "cos1": A seasonal trend only.
- "cos2": A seasonal trend with variable amplitude across years.
- "cos3": A seasonal trend with steadily decreasing amplitude over time.
- "linear": A linear long-term trend with no seasonal trend.
- "monthly": Uses a user-specified probability of exposure for each month.

<code>exposure_slope</code>	A numeric value specifying the linear slope of the exposure, to be used with <code>exposure_trend = "linear"</code> or <code>exposure_trend = "cos1linear"</code> . The default value is 1. Positive values will generate data with an increasing expected value over the years while negative values will generate data with decreasing expected value over the years.
<code>exposure_amp</code>	A numeric value specifying the amplitude of the exposure trend. Must be between -1 and 1 for continuous exposure or between -0.5 and 0.5 for binary exposure. Positive values will simulate a pattern with higher values at the time of the year of the start of the dataset (typically January) and lowest values six months following that (typically July). Negative values can be used to simulate a trend with lower values at the time of year of the start of the dataset and higher values in the opposite season.
<code>outcome_trend</code>	A character string specifying the seasonal trend in health outcomes. Options are the same as for continuous exposure data.
<code>outcome_slope</code>	A numeric value specifying the linear slope of the outcome trend, to be used with <code>outcome_trend = "linear"</code> or <code>outcome_trend = "cos1linear"</code> . The default value is 1. Positive values will generate data with an increasing expected value over the years while negative values will generate data with decreasing expected value over the years.
<code>outcome_amp</code>	A numeric value specifying the amplitude of the outcome trend. Must be between -1 and 1.
<code>start.date</code>	A date of the format "yyyy-mm-dd" from which to begin simulating daily exposures
<code>cust_exp_func</code>	An R object name specifying the name of a custom trend function to generate exposure data

cust_exp_args	A list of arguments and their values for the user-specified custom exposure function.
cust_expdraw	An R object specifying a user-created function which determines the distribution of random noise off of the trend line. This function must have inputs n and prob for a binary exposure function and inputs n and mean for a continuous exposure function. The custom function must output a vector of simulated exposure values.
cust_expdraw_args	A list of arguments other than n required by the cust_expdraw function.
cust_base_func	A R object name specifying a user-made custom function for baseline trend.
cust_lambda_func	An R object name specifying a user-made custom function for relating baseline, relative risk, and exposure
cust_base_args	A list of arguments and their values used in the user-specified custom baseline function
cust_lambda_args	A list of arguments and their values used in the user-specified custom lambda function
cust_outdraw	An R object name specifying a user-created function to randomize the outcome values off of the baseline for outcome values. This function must take inputs n and lambda and output a vector of outcome values.
cust_outdraw_args	A list of arguments besides n passed to the user-created custom outcome draw function.

Value

A list object of length n_rep, in which each list element is one of the synthetic datasets simulated under the input conditions. Each synthetic dataset includes columns for date (date), daily exposure (x), and daily outcome count (outcome).

References

Bateson TF, Schwartz J. 1999. Control for seasonal variation and time trend in case-crossover studies of acute effects of environmental exposures. *Epidemiology* 10(4):539-544.

Examples

```
create_sims(n_reps=3, n=100, central = 100, sd = 10,
           exposure_type="continuous", exposure_trend = "cos1",
           exposure_amp = .6, average_outcome = 22,
           outcome_trend = "no trend", outcome_amp = .6, rr = 1.01)
```

custom_baseline	<i>Pull smoothed Chicago NMMAPS health outcome data</i>
-----------------	---

Description

Example of a custom baseline function that can be passed to `eesim` or `power_calc`. By default, this function pulls smoothed data from the `chicagoNMMAPS` data set in the `dlnm` package. The user may also input a different data set from which to pull data. The function uses a smoothed function of this observed data as the underlying baseline outcome trend in simulating data.

Usage

```
custom_baseline(n, df = dlnm::chicagoNMMAPS, outcome_type = "cvd",
  start.date = "2000-01-01")
```

Arguments

<code>n</code>	A numeric value specifying the number of days for which to obtain an exposure value.
<code>df</code>	Data frame from which to pull exposure values.
<code>outcome_type</code>	A character string specifying the desired health outcome metric. Options are: <ul style="list-style-type: none">• "death"• "cvd"• "resp" (Note: These are the column names for outcome counts in the observed data.)
<code>start.date</code>	A date of the format "yyyy-mm-dd" from which to begin pulling exposure values. Dates in the Chicago NMMAPS data set are from 1987-01-01 to 2000-12-31.

Value

A data frame with one column for date and one column for baseline outcome values.

Examples

```
custom_baseline(n = 5)
custom_baseline(n = 5, outcome_type = "death")
```

custom_exposure	<i>Pull exposure series from data set</i>
-----------------	---

Description

Example of a custom exposure function that can be passed to `eesim` or `power_calc`. By default, this function pulls exposure data from the Chicago NMMAPS data set in the `dlnm` package. The user may specify a different data set from which to pull exposure values.

Usage

```
custom_exposure(n, df = dlnm::chicagoNMMAPS, metric = "temp",
  start.date = NULL)
```

Arguments

<code>n</code>	A numeric value specifying the number of days for which to obtain an exposure value.
<code>df</code>	Data frame from which to pull exposure values.
<code>metric</code>	A character string specifying the desired exposure metric. Options are: <ul style="list-style-type: none"> • "temp" • "dptp" • "rhum" • "pm10" • "o3" (Note: These are the column names for exposure measurements in the observed data.)
<code>start.date</code>	A date of the format "yyyy-mm-dd" from which to begin pulling exposure values. Dates in the Chicago NMMAPS data set are from 1987-01-01 to 2000-12-31.

Value

A numeric vector of length `n` giving exposure values.

Examples

```
custom_exposure(n = 5, metric = "temp", start.date = "2000-01-01")
```

eesim

*Simulate data, fit models, and assess models***Description**

Generates synthetic time series datasets relevant for environmental epidemiology studies and tests performance of a model on that simulated data. Datasets can be generated with seasonal and long-term trends in either exposure or outcome. Binary or continuous outcomes can be simulated or incorporated from observed datasets. The function includes extensive options for customizing each step of the simulation process; see the `eesim` vignette for more details and examples.

Usage

```
eesim(n_reps, n, rr, exposure_type, custom_model, central = NULL, sd = NULL,
      exposure_trend = "no trend", exposure_slope = NULL, exposure_amp = NULL,
      average_outcome = NULL, outcome_trend = "no trend",
      outcome_slope = NULL, outcome_amp = NULL, start.date = "2000-01-01",
      cust_exp_func = NULL, cust_exp_args = NULL, cust_expdraw = NULL,
      cust_expdraw_args = NULL, cust_base_func = NULL,
      cust_lambda_func = NULL, cust_base_args = NULL, cust_lambda_args = NULL,
      cust_outdraw = NULL, cust_outdraw_args = NULL, custom_model_args = NULL)
```

Arguments

<code>n_reps</code>	An integer specifying the number of datasets to simulate (e.g., <code>n_reps = 1000</code> would simulate one thousand time series datasets with the specified characteristics, which can be used for a power analysis or to investigate the performance of a proposed model).
<code>n</code>	An integer specifying the number of days to simulate (e.g., <code>n = 365</code> would simulate a dataset with a year's worth of data).
<code>rr</code>	A non-negative numeric value specifying the relative risk (i.e., the relative risk per unit increase in the exposure).
<code>exposure_type</code>	A character string specifying the type of exposure. Choices are "binary" or "continuous".
<code>custom_model</code>	The object name of an R function that defines the code that will be used to fit the model. This object name should not be in quotations. See Details for more.
<code>central</code>	A numeric value specifying the mean probability of exposure (for binary data) or the mean exposure value (for continuous data).
<code>sd</code>	A non-negative numeric value giving the standard deviation of the exposure values from the exposure trend line (not the total standard deviation of the exposure values).
<code>exposure_trend</code>	A character string specifying a seasonal and / or long-term trend for expected mean exposure. See the vignette for <code>eesim</code> for examples of each option. The shapes are based on those used in Bateson and Schwartz (1999). For trends with a seasonal component, the amplitude of the seasonal trend can be customized

using the `exposure_amp` argument. For trends with a long-term pattern, the slope of the long-term trend can be set using the `exposure_slope` argument. If using the "monthly" option for a binary exposure, you must input a numeric vector of length 12 for the central argument that gives the probability of exposure for each month, starting in January and ending in December. Options for continuous exposure are:

- "no trend": No trend, either seasonal or long-term (default).
- "cos1": A seasonal trend only.
- "cos2": A seasonal trend with variable amplitude across years.
- "cos3": A seasonal trend with steadily decreasing amplitude over time.
- "linear": A linear long-term trend with no seasonal trend.
- "curvilinear": A curved long-term trend with no seasonal trend.
- "cos1linear": A seasonal trend plus a linear long-term trend.

Options for binary exposure are:

- "no trend": No trend, either seasonal or long-term (default).
- "cos1": A seasonal trend only.
- "cos2": A seasonal trend with variable amplitude across years.
- "cos3": A seasonal trend with steadily decreasing amplitude over time.
- "linear": A linear long-term trend with no seasonal trend.
- "monthly": Uses a user-specified probability of exposure for each month.

<code>exposure_slope</code>	A numeric value specifying the linear slope of the exposure, to be used with <code>exposure_trend = "linear"</code> or <code>exposure_trend = "cos1linear"</code> . The default value is 1. Positive values will generate data with an increasing expected value over the years while negative values will generate data with decreasing expected value over the years.
<code>exposure_amp</code>	A numeric value specifying the amplitude of the exposure trend. Must be between -1 and 1 for continuous exposure or between -0.5 and 0.5 for binary exposure. Positive values will simulate a pattern with higher values at the time of the year of the start of the dataset (typically January) and lowest values six months following that (typically July). Negative values can be used to simulate a trend with lower values at the time of year of the start of the dataset and higher values in the opposite season.
<code>average_outcome</code>	A non-negative numeric value specifying the average daily outcome count.
<code>outcome_trend</code>	A character string specifying the seasonal trend in health outcomes. Options are the same as for continuous exposure data.
<code>outcome_slope</code>	A numeric value specifying the linear slope of the outcome trend, to be used with <code>outcome_trend = "linear"</code> or <code>outcome_trend = "cos1linear"</code> . The default value is 1. Positive values will generate data with an increasing expected value over the years while negative values will generate data with decreasing expected value over the years.
<code>outcome_amp</code>	A numeric value specifying the amplitude of the outcome trend. Must be between -1 and 1.
<code>start.date</code>	A date of the format "yyyy-mm-dd" from which to begin simulating daily exposures

<code>cust_exp_func</code>	An R object name specifying the name of a custom trend function to generate exposure data
<code>cust_exp_args</code>	A list of arguments and their values for the user-specified custom exposure function.
<code>cust_expdraw</code>	An R object specifying a user-created function which determines the distribution of random noise off of the trend line. This function must have inputs <code>n</code> and <code>prob</code> for a binary exposure function and inputs <code>n</code> and <code>mean</code> for a continuous exposure function. The custom function must output a vector of simulated exposure values.
<code>cust_expdraw_args</code>	A list of arguments other than <code>n</code> required by the <code>cust_expdraw</code> function.
<code>cust_base_func</code>	A R object name specifying a user-made custom function for baseline trend.
<code>cust_lambda_func</code>	An R object name specifying a user-made custom function for relating baseline, relative risk, and exposure
<code>cust_base_args</code>	A list of arguments and their values used in the user-specified custom baseline function
<code>cust_lambda_args</code>	A list of arguments and their values used in the user-specified custom lambda function
<code>cust_outdraw</code>	An R object name specifying a user-created function to randomize the outcome values off of the baseline for outcome values. This function must take inputs <code>n</code> and <code>lambda</code> and output a vector of outcome values.
<code>cust_outdraw_args</code>	A list of arguments besides <code>n</code> passed to the user-created custom outcome draw function.
<code>custom_model_args</code>	A list of arguments and their values for a custom model. These arguments are passed through to the function specified with <code>custom_model</code> .

Value

A list object with three elements:

- `simulated_datasets`: A list of length `n_reps`, in which each element is a data frame with one of the simulated time series datasets, created according to the specifications set by the user.
- `indiv_performance`: A dataframe with one row per simulated dataset (i.e., total number of rows equal to `n_reps`). Each row gives the results of fitting the specified model to one of the simulated datasets. See [fit_mods](#) for more on this output.
- `overall_performance`: A one-row dataframe with overall performance summaries from fitting the specified model to the synthetic datasets. See [check_sims](#) for more on this output.

References

Bateson TF, Schwartz J. 1999. Control for seasonal variation and time trend in case-crossover studies of acute effects of environmental exposures. *Epidemiology* 10(4):539-544.

Examples

```
# Run a simulation for a continuous exposure (mean = 100, standard
# deviation after long-term and seasonal trends = 10) that increases
# risk of a count outcome by 0.1% per unit increase, where the average
# daily outcome is 22 per day. The exposure outcome has a seasonal trend,
# with higher values in the winter, while the outcome has no seasonal
# or long-term trends beyond those introduced through effects from the
# exposure. The simulated data are fit with a model defined by the `spline_mod`
# function (also in the `eesim` package), with its `df_year` argument set to 7.

sims <- eesim(n_reps = 3, n = 5 * 365, central = 100, sd = 10,
             exposure_type = "continuous", exposure_trend = "cos3",
             exposure_amp = .6, average_outcome = 22, rr = 1.001,
             custom_model = spline_mod, custom_model_args = list(df_year = 7))
names(sims)
sims[[2]]
sims[[3]]
```

fit_mods

Fit a model to simulated datasets

Description

Fits a specified model to each of the simulated datasets and returns a dataframe summarizing results from fitting the model to each dataset, including the estimated effect and the estimated standard error for that estimated effect. The model is specified through a user-created R function, which must take specific input and return output in a specific format. For more details, see the parameter definitions, the Details section, and the vignette for the `eesim` package.

Usage

```
fit_mods(data, custom_model = NULL, custom_model_args = list())
```

Arguments

<code>data</code>	A list of simulated data sets. Each simulated dataset must include a column called <code>x</code> with daily exposure values and a column called <code>outcome</code> with daily outcome values. Typically, this will be the outcome from <code>create_sims</code> .
<code>custom_model</code>	The object name of an R function that defines the code that will be used to fit the model. This object name should not be in quotations. See Details for more.
<code>custom_model_args</code>	A list of arguments and their values for a custom model. These arguments are passed through to the function specified with <code>custom_model</code> .

Details

The function specified by the `custom_model` argument should be a user-created function that inputs a data frame with columns named "x" for exposure values and "outcome" for outcome values. The function must output a data frame with columns called Estimate, Std. Error, t value, $\Pr(>|t|)$, 2.5%, and 97.5%. Note that these columns are the output from `summary` and `confint` for models fit using a `glm` call. You may use the function `format_out` from `eesim` within your function to produce output with these columns if this model is fit using `glm` or something similar. For more details and examples, see the vignette for `eesim`.

Value

A data frame in which each row gives the results from the model-fitting function run on one of the simulated datasets input to the function as the data object. The returned data frame has one row per simulated dataset and the following columns:

- `Estimate`: The estimated β (log relative risk) as estimated by the model specified with `custom_model`.
- `Std.Error`: The standard error for the estimated β .
- `t.value`: The test statistic for a test of the null hypothesis $\beta = 0$.
- `p.value`: The p-value for a test of the null hypothesis $\beta = 0$.
- `lower_ci`: The lower value in the 95% confidence interval estimated for β .
- `upper_ci`: The upper value in the 95% confidence interval estimated for β .

Examples

```
# Create a set of simulated datasets and then fit the model defined in `spline_mod` to
# all datasets, using the argument `df_year = 7` in the call to `spline_mod`. The `spline_mod`
# function is included in the `eesim` package and can be investigated by calling the function
# name without parentheses (i.e., `spline_mod`).
sims <- create_sims(n_reps = 10, n = 5 * 365, central = 100, sd = 10,
  exposure_type = "continuous", exposure_trend = "cos1",
  exposure_amp = .6, average_outcome = 22,
  outcome_trend = "no trend", outcome_amp = .6, rr = 1.01)
fit_mods(data = sims, custom_model = spline_mod, custom_model_args = list(df_year = 7))
```

format_out

Format output for custom model to use in eesim

Description

Formats the output within a modeling function to be used in a call to `eesim` when the model is fit using `glm` or something similar.

Usage

```
format_out(mod)
```

Arguments

mod A model object from lm, glm, etc.

Value

Output with the correct values and column names needed for a modeling function to pass to `eesim`.

Examples

```
dat <- data.frame(x=rnorm(1000, 0, 1), outcome = rnorm(1000, 5, 1))
lin_mod <- lm(outcome~x, data=dat)
format_out(lin_mod)
```

mean_beta	<i>Average Estimated Coefficient</i>
-----------	--------------------------------------

Description

This function gives the mean value of the estimated log relative risks ($\hat{\beta}$ s) and the mean of the estimated relative risk values over the n simulations.

Usage

```
mean_beta(df)
```

Arguments

df A data frame of replicated simulations which must include a column titled "Estimate" with the effect estimate from the fitted model.

Value

A data frame with the mean estimated log relative risk and mean estimated relative risk. The mean estimated risk is based on first calculating the mean log relative risk and then exponentiating this mean value.

Examples

```
sims <- create_sims(n_reps=10, n=50, central = 100, sd = 10,
  exposure_type="continuous", exposure_trend = "cos1",
  exposure_amp = .6, average_outcome = 22,
  outcome_trend = "no trend", outcome_amp = .6, rr = 1.01)
fits <- fit_mods(data = sims, custom_model = spline_mod,
  custom_model_args = list(df_year = 1))
mean_beta(df=fits)
```

power_beta	<i>Estimate power</i>
------------	-----------------------

Description

Calculates the estimated power of a hypothesis test that the log relative risk equals 0 at a 5% significance level across all simulated data.

Usage

```
power_beta(df)
```

Arguments

df	A data frame of replicated simulations which must include columns titled lower_ci and upper_ci.
----	---

Value

A data frame with one row with the estimated power of the analysis at the 5% significance level.

Examples

```
sims <- create_sims(n_reps = 10, n = 600, central = 100,
                  sd = 10, exposure_type = "continuous",
                  exposure_trend = "cos1",
                  exposure_amp = 0.6,
                  average_outcome = 20,
                  outcome_trend = "no trend",
                  rr = 1.01)
fits <- fit_mods(data = sims, custom_model = spline_mod,
                 custom_model_args = list(df_year = 1))
power_beta(fits)
```

power_calc	<i>Power Calculations</i>
------------	---------------------------

Description

Calculates the expected power of an environmental epidemiology time series analysis based on simulated datasets. This function uses the simulation provided by `eesim` to simulate multiple environmental epidemiology datasets under different scenarios (e.g., total days in study, size of association between exposure and outcome, or baseline average daily count of the outcome in the study) and estimates the power of a specified analysis to detect the hypothesized association.

Usage

```
power_calc(varying, values, n_reps, custom_model, central, exposure_type,
           n = NULL, sd = NULL, exposure_trend = "no trend", exposure_amp = NULL,
           average_outcome = NULL, outcome_trend = "no trend", outcome_amp = NULL,
           rr = NULL, start.date = "2000-01-01", cust_exp_func = NULL,
           cust_exp_args = NULL, cust_base_func = NULL, cust_lambda_func = NULL,
           cust_base_args = NULL, cust_lambda_args = NULL,
           custom_model_args = NULL, plot = FALSE)
```

Arguments

varying	A character string specifying the parameter to be varied. Choices are 'n' (which varies the number of days in each dataset of simulated data), 'rr' (which varies the relative rate per unit increase in exposure that is used to simulate the data), or 'average_outcome' (which varies the average value of the outcomes in each dataset). For whichever of these three values is not set to vary in this argument, the user must specify a constant value to this function through the n, rr, or average_outcome arguments.
values	A numeric vector with the values you would like to test for the varying parameters. For example, values = c(1.05, 1.10, 1.15) would produce power estimates for the four specified values of relative risk if the user has specified varying = 'rr'.
n_reps	An integer specifying the number of datasets to simulate (e.g., n_reps = 1000 would simulate one thousand time series datasets with the specified characteristics, which can be used for a power analysis or to investigate the performance of a proposed model).
custom_model	The object name of an R function that defines the code that will be used to fit the model. This object name should not be in quotations. See Details for more.
central	A numeric value specifying the mean probability of exposure (for binary data) or the mean exposure value (for continuous data).
exposure_type	A character string specifying the type of exposure. Choices are "binary" or "continuous".
n	An integer specifying the number of days to simulate (e.g., n = 365 would simulate a dataset with a year's worth of data).
sd	A non-negative numeric value giving the standard deviation of the exposure values from the exposure trend line (not the total standard deviation of the exposure values).
exposure_trend	A character string specifying a seasonal and / or long-term trend for expected mean exposure. See the vignette for eesim for examples of each option. The shapes are based on those used in Bateson and Schwartz (1999). For trends with a seasonal component, the amplitude of the seasonal trend can be customized using the exposure_amp argument. For trends with a long-term pattern, the slope of the long-term trend can be set using the exposure_slope argument. If using the "monthly" option for a binary exposure, you must input a numeric vector of length 12 for the central argument that gives the probability of exposure for each month, starting in January and ending in December. Options for continuous exposure are:

- "no trend": No trend, either seasonal or long-term (default).
- "cos1": A seasonal trend only.
- "cos2": A seasonal trend with variable amplitude across years.
- "cos3": A seasonal trend with steadily decreasing amplitude over time.
- "linear": A linear long-term trend with no seasonal trend.
- "curvilinear": A curved long-term trend with no seasonal trend.
- "cos1linear": A seasonal trend plus a linear long-term trend.

Options for binary exposure are:

- "no trend": No trend, either seasonal or long-term (default).
- "cos1": A seasonal trend only.
- "cos2": A seasonal trend with variable amplitude across years.
- "cos3": A seasonal trend with steadily decreasing amplitude over time.
- "linear": A linear long-term trend with no seasonal trend.
- "monthly": Uses a user-specified probability of exposure for each month.

exposure_amp	A numeric value specifying the amplitude of the exposure trend. Must be between -1 and 1 for continuous exposure or between -0.5 and 0.5 for binary exposure. Positive values will simulate a pattern with higher values at the time of the year of the start of the dataset (typically January) and lowest values six months following that (typically July). Negative values can be used to simulate a trend with lower values at the time of year of the start of the dataset and higher values in the opposite season.
average_outcome	A non-negative numeric value specifying the average daily outcome count.
outcome_trend	A character string specifying the seasonal trend in health outcomes. Options are the same as for continuous exposure data.
outcome_amp	A numeric value specifying the amplitude of the outcome trend. Must be between -1 and 1.
rr	A non-negative numeric value specifying the relative risk (i.e., the relative risk per unit increase in the exposure).
start.date	A date of the format "yyyy-mm-dd" from which to begin simulating daily exposures
cust_exp_func	An R object name specifying the name of a custom trend function to generate exposure data
cust_exp_args	A list of arguments and their values for the user-specified custom exposure function.
cust_base_func	A R object name specifying a user-made custom function for baseline trend.
cust_lambda_func	An R object name specifying a user-made custom function for relating baseline, relative risk, and exposure
cust_base_args	A list of arguments and their values used in the user-specified custom baseline function
cust_lambda_args	A list of arguments and their values used in the user-specified custom lambda function

`custom_model_args` A list of arguments and their values for a custom model. These arguments are passed through to the function specified with `custom_model`.

`plot` "TRUE" or "FALSE" for whether to produce a plot

Value

Data frame with the values of the varying parameter and the estimated power for each. If the `plot` argument is set to TRUE, it also returns a power curve plot as a side effect. Because these estimates are based on simulations, there will be some random variation in estimates of power. Estimates will be more stable if a higher value is used for `n_reps`, although this will increase the time it takes the function to run.

Examples

```
# Calculate power for studies that vary in the total length of the study period
# (between one and twenty-one years of data) for the association between a continuous
# exposure with a seasonal trend (mean = 100, sd from seasonal baseline = 10) and a count
# outcome (e.g., daily number of deaths, mean daily value across the study period of 22).
# The alternative hypothesis is that there is a relative rate of the outcome of 1.001 for
# every one-unit increase in exposure. The null hypothesis is that there is no association
# between the exposure and the outcome. The model used to test for an association is a
# case-crossover model
## Not run:
pow <- power_calc(varying = "n", values = floor(365.25 * seq(1, 21, by = 5)), n_reps = 20,
  central = 100, sd = 10, rr = 1.001, exposure_type = "continuous",
  exposure_trend = "cos1", exposure_amp = .6, average_outcome = 22,
  outcome_trend = "no trend", outcome_amp = .6,
  custom_model = spline_mod, plot = TRUE)

## End(Not run)
```

sim_baseline

Expected baseline health outcomes

Description

Generates expected baseline health outcome counts based on average outcome and desired seasonal and / or long-term trends.

Usage

```
sim_baseline(n, lambda, trend = "no trend", slope = 1, amp = 0.6,
  start.date = "2000-01-01")
```

Arguments

n	A numeric value specifying the number of days for which to simulate data
lambda	A numeric value specifying the mean for the expected outcomes
trend	A character string that specifies the desired trend function. Options are: <ul style="list-style-type: none"> • "no trend": No trend, either seasonal or long-term (default). • "cos1": A seasonal trend only. • "cos2": A seasonal trend with variable amplitude across years. • "cos3": A seasonal trend with steadily decreasing amplitude over time. • "linear": A linear long-term trend with no seasonal trend. • "curvilinear": A curved long-term trend with no seasonal trend. • "cos1linear": A seasonal trend plus a linear long-term trend. <p>See the package vignette for examples of the shapes of these trends.</p>
slope	A numeric value specifying the slope of the trend, to be used with trend = "linear" or trend = "cos1linear".
amp	A numeric value specifying the amplitude of the seasonal trend. Must be between -1 and 1.
start.date	A date in the format "yyyy-mm-dd" specifying the first day for which to simulate data

Value

A data frame with the date and expected baseline outcome count for each day of simulated data.

Examples

```
sim_baseline(n = 5, lambda = 100, trend = "cos1")
```

sim_exposure	<i>Simulate time series of exposure values</i>
--------------	--

Description

Simulates a time series of binary or continuous exposure values with or without seasonal trends. It also allows the user to use a custom function for the shape of the exposure trend.

Usage

```
sim_exposure(n, central = NULL, sd = NULL, trend = "no trend",
  slope = 1, amp = 0.6, exposure_type = NULL, start.date = "2001-01-01",
  cust_exp_func = NULL, cust_exp_args = NULL, cust_expdraw = NULL,
  cust_expdraw_args = list())
```

Arguments

n	A non-negative integer specifying the number of days to simulate.
central	A numeric value specifying the mean probability of exposure (for binary data) or the mean exposure value (for continuous data).
sd	A numeric value giving the standard deviation of the exposure values from the exposure trend line.
trend	A character string that specifies the desired trend function. Options are: <ul style="list-style-type: none"> • "no trend": No trend, either seasonal or long-term (default). • "cos1": A seasonal trend only. • "cos2": A seasonal trend with variable amplitude across years. • "cos3": A seasonal trend with steadily decreasing amplitude over time. • "linear": A linear long-term trend with no seasonal trend. • "curvilinear": A curved long-term trend with no seasonal trend. • "cos1linear": A seasonal trend plus a linear long-term trend. <p>See the package vignette for examples of the shapes of these trends.</p>
slope	A numeric value specifying the slope of the trend, to be used with trend = "linear" or trend = "cos1linear".
amp	A numeric value specifying the amplitude of the seasonal trend. Must be between -1 and 1.
exposure_type	A character string specifying the type of exposure. Choices are "binary" or "continuous".
start.date	A date of the format "yyyy-mm-dd" from which to begin simulating daily exposures
cust_exp_func	An R object specifying the function to use to generate custom exposure values.
cust_exp_args	A list of arguments used in the user-specified custom function.
cust_expdraw	An R object specifying a user-created function which determines the distribution of random noise off of the trend line. This function must have inputs n and prob for a binary exposure function and inputs n and mean for a continuous exposure function. The custom function must output a vector of simulated exposure values.
cust_expdraw_args	A list of arguments other than n required by the cust_expdraw function.

Value

A data frame with two columns: date (date) and simulated exposure values (x).

Examples

```
sim_exposure(n = 5, central = 0.25, exposure_type = "binary")
sim_exposure(n = 5, central = 100, sd = 10, exposure_type = "continuous")
library(ggplot2)
x_cont <- sim_exposure(n = 1000, central = 100, sd = 10,
                      exposure_type = "continuous",
```

```

      trend = "cos1linear", amp = 0.6)
ggplot(x_cont, aes(x = date, y = x)) + geom_point()
x_cust <- sim_exposure(n = 1000, cust_exp_func = "custom_exposure",
                      cust_exp_args = list(metric = "temp"))
ggplot(x_cust, aes(x = date, y = x)) + geom_point()

```

 sim_outcome

Simulate outcome

Description

Simulates daily outcome counts for each study day based on user specifications for average outcome count, any underlying trends in expected outcome counts, and the association between exposure and outcome. This function starts from a vector of expected outcome count on each study day and simulates through a draw from a Poisson distribution based on this expected daily value. If desired, a user can also use a custom function to customize this stage of the simulation; see the vignette for `eesim` for more details and examples.

Usage

```

sim_outcome(exposure, average_outcome = NULL, trend = "no trend",
            slope = 1, amp = 0.6, rr = 1.01, start.date = "2000-01-01",
            cust_base_func = NULL, cust_lambda_func = NULL, cust_base_args = list(),
            cust_lambda_args = list(), cust_outdraw = NULL,
            cust_outdraw_args = list())

```

Arguments

<code>exposure</code>	A numeric vector of exposure values, typically the output of <code>sim_exposure</code> .
<code>average_outcome</code>	A non-negative numeric value specifying the average daily outcome count.
<code>trend</code>	A character string that specifies the desired trend function. Options are: <ul style="list-style-type: none"> "no trend": No trend, either seasonal or long-term (default). "cos1": A seasonal trend only. "cos2": A seasonal trend with variable amplitude across years. "cos3": A seasonal trend with steadily decreasing amplitude over time. "linear": A linear long-term trend with no seasonal trend. "curvilinear": A curved long-term trend with no seasonal trend. "cos1linear": A seasonal trend plus a linear long-term trend. See the package vignette for examples of the shapes of these trends.
<code>slope</code>	A numeric value specifying the slope of the trend, to be used with <code>trend = "linear"</code> or <code>trend = "cos1linear"</code> .
<code>amp</code>	A numeric value specifying the amplitude of the seasonal trend. Must be between -1 and 1.

<code>rr</code>	A non-negative numeric value specifying the relative risk (i.e., the relative risk per unit increase in the exposure).
<code>start.date</code>	A date of the format "yyyy-mm-dd" from which to begin simulating values
<code>cust_base_func</code>	A R object name specifying a user-made custom function for baseline trend.
<code>cust_lambda_func</code>	An R object name specifying a user-made custom function for relating baseline, relative risk, and exposure
<code>cust_base_args</code>	A list of arguments and their values used in the user-specified custom baseline function
<code>cust_lambda_args</code>	A list of arguments and their values used in the user-specified custom lambda function
<code>cust_outdraw</code>	An R object name specifying a user-created function to randomize the outcome values off of the baseline for outcome values. This function must take inputs <code>n</code> and <code>lambda</code> and output a vector of outcome values.
<code>cust_outdraw_args</code>	A list of arguments besides <code>n</code> passed to the user-created custom outcome draw function.

Value

A dataframe with a simulated time series, with columns for date, `x` (exposure), and outcome.

Examples

```
exp <- sim_exposure(n = 5, central = 100, sd = 10, amp = .6,
                  exposure_type = "continuous")
sim_outcome(exposure = exp, average_outcome = 22, trend = "linear")
```

spline_mod

Fit a generalized linear model

Description

Fits a model to estimate the log relative risk between an exposure and outcome by using a natural cubic spline to control for long-term and seasonal trends and assuming a quasi-Poisson distribution for the outcome. This function provides an example of a function that can be input to [eesim](#) or [power_calc](#).

Usage

```
spline_mod(df, df_year = 7)
```


Arguments

df	A data frame with columns for daily exposure and outcome called "x" and "outcome", respectively
df_year	A numeric value specifying the degrees of freedom per year to use in the spline to control for long-term and seasonal trends in the outcome.

Value

A numeric vector of length six with summary measures from fitting this model to simulated data.

Examples

```
exp <- sim_exposure(n = 500, central = 100, sd = 10, trend = "cos1",
                  amp = .6, exposure_type = "continuous")
out <- sim_outcome(exposure = exp, average_outcome = 22, rr = 1.01)
spline_mod(df = out)
```

std_exposure	<i>Simulate exposure data using default methods</i>
--------------	---

Description

Simulates binary or continuous exposure data with or without seasonal trends using default functions.

Usage

```
std_exposure(n, central, sd = NULL, trend = "no trend",
            exposure_type = "binary", slope, amp, start.date = "2000-01-01", ...)
```

Arguments

n	A non-negative integer specifying the number of days to simulate.
central	A numeric value specifying the mean probability of exposure (for binary data) or the mean exposure value (for continuous data).
sd	A numeric value giving the standard deviation of the exposure values from the exposure trend line.
trend	A character string that specifies the desired trend function. Options are: <ul style="list-style-type: none"> • "no trend": No trend, either seasonal or long-term (default). • "cos1": A seasonal trend only. • "cos2": A seasonal trend with variable amplitude across years. • "cos3": A seasonal trend with steadily decreasing amplitude over time. • "linear": A linear long-term trend with no seasonal trend. • "curvilinear": A curved long-term trend with no seasonal trend.

- "coslinear": A seasonal trend plus a linear long-term trend.

See the package vignette for examples of the shapes of these trends.

exposure_type	A character string specifying the type of exposure. Choices are "binary" or "continuous".
slope	A numeric value specifying the slope of the trend, to be used with trend = "linear" or trend = "coslinear".
amp	A numeric value specifying the amplitude of the seasonal trend. Must be between -1 and 1.
start.date	A date of the format "yyyy-mm-dd" from which to begin simulating daily exposures
...	Optional arguments to a custom trend function

Value

A data frame with two columns: date (date) and simulated exposure values (x).

Examples

```
std_exposure(n = 5, central = .1, trend = "cos1", amp = .02)
std_exposure(n = 5, central = 50, sd = 5, trend = "cos3", amp = .6,
             exposure_type = "continuous", start.date = "2001-04-01")
std_exposure(n=50, central=.1, amp=.05,cust_expdraw=rnbinom,
             cust_expdraw_args=list(size=10))
```

Index

beta_bias, 2, 8
beta_var, 3, 8
bin_t, 5
binary_exposure, 4

calc_t, 6
calendar_plot, 7
check_sims, 8, 21
continuous_exposure, 9
coverage_beta, 8, 10
coverage_plot, 11
create_baseline, 12, 13
create_lambda, 13
create_sims, 14, 22
custom_baseline, 17
custom_exposure, 18

eesim, 11, 17, 18, 19, 23, 24, 32

fit_mods, 11, 21, 22
format_out, 23

mean_beta, 8, 24

power_beta, 8, 25
power_calc, 17, 18, 25, 32

sim_baseline, 28
sim_exposure, 13, 29, 31
sim_outcome, 31
spline_mod, 32
std_exposure, 33