Package ‘radsafer’

May 27, 2019

Type Package

Title Radiation Safety

Version 2.0.1

Author Mark Hogue <mark.hogue.chp@gmail.com>

Maintainer Mark Hogue <mark.hogue.chp@gmail.com>

Description Provides functions for radiation safety, also known as ``radiation protection'' and ``radiological control''. The science of radiation protection is called ``health physics'' and its engineering functions are called ``radiological engineering''. Functions in this package cover many of the computations needed by radiation safety professionals. Examples include: obtaining updated calibration and source check values for radiation monitors to account for radioactive decay in a reference source, simulating instrument readings to better understand measurement uncertainty, correcting instrument readings for geometry and ambient atmospheric conditions. Many of these functions are described in Johnson and Kirby (2011, ISBN-13: 978-1609134198). Utilities are also included for developing inputs and processing outputs with radiation transport codes, such as MCNP, a general-purpose Monte Carlo N-Particle code that can be used for neutron, photon, electron, or coupled neutron/photon/electron transport (Werner et. al. (2018) <doi:10.2172/1419730>).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests testthat, tidyverse, scatterplot3d, beepr

Imports ggplot2, readr, stats, graphics, RadData, stringr, magrittr, dplyr, rlang

Depends R (>= 3.3)

URL https://github.com/markhogue/radsafer

BugReports https://github.com/markhogue/radsafer/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2019-05-27 20:50:03 UTC

R topics documented:

`air_dens_cf` *Correct for air density - useful for vented ion chambers*

Description

Obtain a correction factor for ion chamber temperature and pressure vs reference calibration values.

Usage

```
air_dens_cf(T.actual, P.actual, T.ref = 20, P.ref = 760)
```

Arguments

`T.actual`	The actual air temperature, in Celsius
`P.actual`	The actual air pressure, in mm Hg
`T.ref`	The reference air temperature - default is 20C
`P.ref`	The reference air pressure - default is 760 mm Hg

Value

The ratio of actual to reference air density.

See Also

Other rad measurements: `disk_to_disk_solid_angle`, `neutron_geom_cf`, `scaler_sim`, `tau_estimate`

Examples

```
air_dens_cf(T.actual = 20, P.actual = 760, T.ref = 20, P.ref = 760)
air_dens_cf(30, 750)
```

`bin_screen_phot` *Search for radioisotopes that dominate a specified energy bin*

Description

Search for photon emission spectra from representative radionuclides. Identify the bin of interest using E_min and E_max with optional min_prob. If minimum probability for photons in the minimized ranges is not identified, no screening will occur in the minimized ranges because the default is 100 (greater than the maximum probability of any photon).

Usage

```
bin_screen_phot(E_min = 0, E_max = 10, min_prob = 0,
  min_half_life_seconds = NULL, max_half_life_seconds = NULL,
  no_E_min = 0, no_E_max = 10, no_min_prob = 100, no_E_min2 = 0,
  no_E_max2 = 10, no_min_prob2 = 100)
```

Arguments

`E_min`	minimum energy in MeV, default = 0
`E_max`	maximum energy in MeV, default = 10
`min_prob`	minimum probability of selected bin with default = 0.
`min_half_life_seconds`	minimum half-life in seconds. Use multiplier as needed, e.g. 3 * 3600 for 3 hours. Default = NULL,
`max_half_life_seconds`	maximum half-life. See min_half_life_seconds.

no_E_min, no_E_min2
: minimum energy in bins to minimize in MeV, default = 0

no_E_max, no_E_max2
: maximum energy in bins to minimize in MeV, default = 10

no_min_prob, no_min_prob2
: minimum probability to minimize with default = 100.

Value

search results in order of half-life. Recommend assigning results to a viewable object, such as 'search_results'

See Also

[RN_plt()]

Other radionuclides: RN_Spec_Act, RN_index_screen, RN_info, RN_plt, search_alpha_by_E, search_beta_by_E, search_phot_by_E

Examples

```
spec_0.1_0.3 <- bin_screen_phot(E_min = 0.1, E_max = 0.3,
min_prob = 0.4, min_half_life_seconds = 30 * 24 * 3600,
max_half_life_seconds = 3.153e7, no_E_min = 0.015,
no_E_max = 0.0999, no_min_prob = 0.05, no_E_min2 = 0.301, no_E_max2 = 10, no_min_prob2 = 0.01)
```

---

disk_to_disk_solid_angle

Calculate fractional solid angle for disk to disk

Description

Returns fractional solid angle for a geometry frequently encountered in health physics analysis of air samples or disk smears. This is useful in correcting configurations that do not exactly match calibration (by ratioing the respective fractional solid angles). While units of steridian are used for solid angle, this function only uses a fraction of the total field of view.

Usage

```
disk_to_disk_solid_angle(r.source, gap, r.detector, plot.opt = "n",
  runs = 10000, off_center = 0, beep = "off")
```

Arguments

r.source	source radius (all units must be consistent)
gap	distance between source and detector
r.detector	detector radius
plot.opt	plot options - "2d", "3d" or "n".
runs	Number of particles to simulate. Running more particles improves accuracy. Default = 1e4.
off_center	measure of eccentricity between the center of the source and the center of the disk. This is applied to the x-dimension of the source.
beep	Set to "on" if desired. Default is "off". Alerts to end of run if runs is set to a high number.

Value

Fractional solid angle and plot of simulation.

References

https://karthikkaranth.me/blog/generating-random-points-in-a-sphere/ https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance

See Also

Other rad measurements: air_dens_cf, neutron_geom_cf, scaler_sim, tau_estimate

Examples

```
disk_to_disk_solid_angle(r.source = 15, gap = 20, r.detector = 10, plot.opt = "n", runs = 1e3)
```

---

dk_activity *Time for a radionuclide to decay to a target activity.*

Description

Calculate time for a radionuclide to decay to a target activity.

Usage

```
dk_activity(A0, half_life, target)
```

Arguments

A0	The original activity, or related parameter.
half_life,	Half-life. Units are arbitrary, but must match time past.
target	The target activity.

Value

time, in same units as half-life, to decay to target activity.

See Also

Other decay corrections: dk_cf, dk_pct_to_num_half_life, dk_reverse, dk_time

Examples

```
# How long does it take for original activity of 10000 Bq to decay to 2500 Bq
# if half-life is 5 minutes?
# (All time units are consistent, so answer will be in minutes)
dk_activity(A0 = 10000, half_life = 5, target = 2500)
```

---

dk_cf — *Correction factor for source decay.*

Description

Compute correction value for decay of a single-isotope source.

Usage

```
dk_cf(half_life, time_unit, date1, date2 = Sys.Date())
```

Arguments

half_life	The half-life numeric value
time_unit,	acceptable values are years, days, hours, and minutes. These may be shortened to y, d, h and m. Must be entered in quotes.
date1	Source reference date. If units are hours or shorter, include time. Format is "YYYY-mm-dd" for longer half-lives, or "YYYY-mm-dd-HH:MM".
date2	Date of interest. Format is same as date1. Default is today's date, obtained from system.

Value

The decay correction factor from the reference date to the date of interest.

See Also

Other decay corrections: dk_activity, dk_pct_to_num_half_life, dk_reverse, dk_time

Examples

```
dk_cf(half_life = 5.27, time_unit = "y", date1 = "2010-12-01", date2 = "2018-12-01")
#
# example defaulting to today's date:
dk_cf(half_life = 28.79, time_unit = "y", date1 = "2001-01-01")
```

dk_pct_to_num_half_life *Number of half-lives past*

Description

Given a percentage reduction in activity, calculate how many half-lives have passed.

Usage

```
dk_pct_to_num_half_life(pct_lost)
```

Arguments

pct_lost	Percentage of activity lost since reference time.

Value

Number of half-lives passed.

See Also

Other decay corrections: dk_activity, dk_cf, dk_reverse, dk_time

Examples

```
dk_pct_to_num_half_life(pct_lost = 93.75)
```

dk_reverse *Calculate amount of radioactivity given interval.*

Description

Calculate the activity at an earlier time, given the time past, the half-life, and the activity at time, t. The result will provide activity in the same units as provided for present activity. Time past and half-life must be in consistent units.

Usage

```
dk_reverse(A1, half_life, t)
```

Arguments

A1	The target activity or related parameter, such as dose rate.
half_life	Half-life. Units are arbitrary, but must match time past.
t	Time past since activity of interest. Units are arbitrary, but must match half-life.

Value

The original activity or related parameter.

See Also

Other decay corrections: dk_activity, dk_cf, dk_pct_to_num_half_life, dk_time

Examples

```
# A Sr-90 Radioisotope thermoelectric generator is discovered and measured.
# The activity is estimated to be around 400 TBq. Original RTG's of this
# type contained 1480 TBq when built 50 years earlier. We're wondering if
# much has leaked. So, we compute the original from what we have.
dk_reverse(A1 = 400, half_life = 28.79, t = 50)
```

---

dk_time — *Time to decay to target radioactivity.*

Description

Calculate time for a radionuclide to decay to a target activity.

Usage

```
dk_time(half_life, A0, A1)
```

Arguments

half_life,	Half-life. Units are arbitrary, but must match time past.
A0	The original activity, or related parameter.
A1	The target activity.

Value

Time, in same units as half-life, to decay to target activity.

See Also

Other decay corrections: dk_activity, dk_cf, dk_pct_to_num_half_life, dk_reverse

Examples

```
# A carbonaceous artifact has a C-14 measurement of 1 dpm per g pure carbon.
# The reference activity is 14 dpm per g pure carbon. How old is our sample?
dk_time(half_life = 5730, A0 = 14, A1 = 1)
```

half_life_2pt — *Calculate half-life based on two data points*

Description

Estimate half-life from two data points. Half-life units are consistent with time units of input. @family rad measurements

Usage

```
half_life_2pt(time1, time2, N1, N2)
```

Arguments

time1	First time: Must be numeric with no formatting.
time2	Second time: Must be numeric with no formatting.
N1	First measurement - can be count rate, dose rate, etc.
N2	Second measurement in units consistent with first measurement.

Value

The calculated half-life in units of time input.

Examples

```
# Between the first two data points in a series of counts
half_life_2pt(time1 = 0, time2 = 1, N1 = 45, N2 = 30)
#
# Between the second and third in the series (same intervals)
half_life_2pt(time1 = 1, time2 = 2, N1 = 30, N2 = 21)
#
# Use on a series
count_times <- 1:5
acts <- 10000 * 2^(-count_times/10) #activities
acts <- rpois(5, acts) #activities with counting variability applied
half_life_2pt(time1 = count_times[1:4], time2 = count_times[2:5],
N1 = acts[1:4], N2 = acts[2:5])
```

mcnp_cone_angle *MCNP Cone Opening Parameter*

Description

MCNP cone surface requires a term, t^2, which is the tangent of the cone angle, in radians, squared. This function takes an input in degrees and provides the parameter needed by MCNP.

Usage

```
mcnp_cone_angle(d)
```

Arguments

d	The cone angle in degrees.

Value

The ratio of actual to reference air density.

See Also

Other mcnp tools: mcnp_est_nps, mcnp_matrix_rotations, mcnp_plot_out_spec, mcnp_scan2spec, mcnp_si_hist, mcnp_si_sp_RD, mcnp_sp_hist

Examples

```
mcnp_cone_angle(45)
```

mcnp_est_nps *Copy and paste MCNP tally fluctuation charts*

Description

Provides quick estimate of number of particles histories, (nps) to obtain target MCNP 'error'. Paste may include up to three tallies side by side in the default MCNP order. For example, the headers of a three tally report includes column names: nps, mean, error, vov, slope, fom, mean, error, vov, slope, fom, mean, error, vov, slope, fom. The structure of the tfc has been the same for versions 4 through 6, including MCNPX.

Usage

```
mcnp_est_nps(err_target)
```

Arguments

err_target	The target Monte Carlo uncertainty

Value

estimate of number of particle histories needed

See Also

Other mcnp tools: `mcnp_cone_angle`, `mcnp_matrix_rotations`, `mcnp_plot_out_spec`, `mcnp_scan2spec`, `mcnp_si_hist`, `mcnp_si_sp_RD`, `mcnp_sp_hist`

Examples

```
# Since this function requires the user
# to copy and paste input, this example
# is set up to provide data for this purpose.
# To run the example, copy and paste the following
# into an input file and delete the hash tags to run.
# Enter '1' for number of tallies.
# mcnp_est_nps(0.01)
#      32768000   4.5039E+00 0.2263 0.0969  0.0 5.0E-02
#      65536000   3.9877E+00 0.1561 0.0553  0.0 5.1E-02
#      98304000   3.4661E+00 0.1329 0.0413  0.0 4.7E-02
#     131072000   3.5087E+00 0.1132 0.0305  0.0 5.0E-02
#     163840000   3.5568E+00 0.0995 0.0228  0.0 5.2E-02
#     196608000   3.8508E+00 0.0875 0.0164  0.0 5.5E-02
#     229376000   3.8564E+00 0.0810 0.0135  0.0 5.5E-02
#     262144000   3.9299E+00 0.0760 0.0118  0.0 5.5E-02
#     294912000   4.0549E+00 0.0716 0.0100  0.0 5.6E-02
#     327680000   4.0665E+00 0.0686 0.0090  0.0 5.4E-02
#     360448000   4.1841E+00 0.0641 0.0079  0.0 5.7E-02
```

`mcnp_matrix_rotations` *Rotation matrices for transformations in MCNP*

Description

Create 3 x 3 rotation matrix in cosines of the angles between the main and auxiliary coordinate systems in the form: xx' yx' zx' xy' yy' zy' xz' yz' zz'

Usage

```
mcnp_matrix_rotations(rot.axis, angle_degrees)
```

Arguments

`rot.axis`	axis of rotation
`angle_degrees`	degree of rotation

Value

rotational matrix for copy and paste to MCNP input

See Also

Other mcnp tools: mcnp_cone_angle, mcnp_est_nps, mcnp_plot_out_spec, mcnp_scan2spec, mcnp_si_hist, mcnp_si_sp_RD, mcnp_sp_hist

Examples

```
mcnp_matrix_rotations('x', 30)
mcnp_matrix_rotations('y', 7)
mcnp_matrix_rotations('z', 15)
# For combined rotations, use matrix multiplication (%*%)
# rotate 45 degrees on x-axis and 45 degrees on y-axis
mcnp_matrix_rotations('x', 45) %*% mcnp_matrix_rotations('y', 45)
```

---

mcnp_plot_out_spec *Convert histogram data to average points and plot as spectrum.*

Description

Model results from MCNP and perhaps other sources typically provide binned tally results with columns representing maximum energy in MeV, a column with the mean tally result titled 'mean' and an uncertainty column titled 'R'.

Usage

```
mcnp_plot_out_spec(spec.df, title)
```

Arguments

spec.df	A data frame with no header. Maximum energy in MeV should be in the first column, binned results in the second column, uncertainty in the third column.
title	Title for chart

See Also

scan2spec.df to copy and paste output spectrum.

Other mcnp tools: mcnp_cone_angle, mcnp_est_nps, mcnp_matrix_rotations, mcnp_scan2spec, mcnp_si_hist, mcnp_si_sp_RD, mcnp_sp_hist

Examples

```
mcnp_plot_out_spec(photons_cs137_hist, 'example Cs-137 well irradiator')
```

mcnp_scan2spec *Copy and paste MCNP output spectral data for use with* mcnp_plot_out_spec()

Description

Provides quick copy-and-paste conversion to data frame. Conversion is based on the two or three columns with input energy or output results from MCNP. Energy is expected to be in bins with maximum energy in MeV in the first column, the input probability or output mean result for the bin in the second column, and relative Monte Carlo uncertainty in the third column.

Usage

```
mcnp_scan2spec()
```

Value

spectrum file with maximum energy and MCNP bin value

See Also

Other mcnp tools: mcnp_cone_angle, mcnp_est_nps, mcnp_matrix_rotations, mcnp_plot_out_spec, mcnp_si_hist, mcnp_si_sp_RD, mcnp_sp_hist

Examples

```
# Since this function requires the user
# to copy and paste input, this example
# is set up to provide data for this purpose.
# To run the example, copy and paste the following
# into an input file and delete the hash tags to run.
# my_hist_data <- scan()
# 0.1000000 3.133122e-05 0.3348260
# 0.4222222 6.731257e-05 0.2017546
# 0.7444444 5.249198e-05 0.4524577
# 1.0666667 2.046046e-04 0.4201954
# 1.3888889 1.525125e-03 0.8049388
# 1.7111111 2.922743e-05 0.7985399
# 2.0333333 5.162954e-03 0.1974694
# 2.3555556 2.048186e-05 0.5011170
# 2.6777778 1.468040e-04 0.7248116
# 3.0000000 1.037092e-04 0.7659850
```

mcnp_si_hist *energy distribution histogram entries*

Description

Make MCNP histogram energy bins cards for source definition if inputs happen to be available in histogram format.

Usage

```
mcnp_si_hist(emin, emax)
```

Arguments

emin	A vector of energies forming the lower bound of the bin. (emin values other than the first value also provide a bin emax.)
emax	A single energy with the upper bound of the highest bin.

Value

A matrix of values for source input to copy and paste into an MCNP input. (The # should be changed to the appropriate distribution number. The NA's in the last row should be discarded.)

See Also

[mcnp_sp_hist()]

Other mcnp tools: mcnp_cone_angle, mcnp_est_nps, mcnp_matrix_rotations, mcnp_plot_out_spec, mcnp_scan2spec, mcnp_si_sp_RD, mcnp_sp_hist

Examples

```
mcnp_si_hist(1:10 / 10, 1.2)
```

mcnp_si_sp_RD *Produce MCNP source terms from ICRP 107 data except beta*

Description

Obtain emission data from the RadData package and write to a file for use with the radiation transport code, MCNP.

Usage

```
mcnp_si_sp_RD(desired_RN, rad_type = NULL, photon = FALSE, cut = 0,
  erg.dist = 1)
```

Arguments

`desired_RN`	Radionuclide in form Ba-137m
`rad_type`	Radiation type, leave NULL if selecting photons or select from: 'X' for X-Ray 'G' for Gamma 'AE' for Auger Electron 'IE' for Internal Conversion Electron 'A' for Alpha 'AR' for Alpha Recoil 'B-' for Beta Negative 'AQ' for Annihilation Quanta 'B+' for Beta Positive 'PG' for Prompt Gamma 'DG' for Delayed Gamma 'DB' for Delayed Beta 'FF' for Fission Fragment 'N' for Neutron
`photon`	'Y' to select all rad_types that are photons
`cut`	minimum energy defaults to 0
`erg.dist`	energy distribution number for MCNP input

Value

a data frame can be saved to memory if desired (i.e. by my_file <- mcnp_si_sp_RD(...)). For use with MCNP, a text file, 'si.sp.dat' is written to working directory. If file already exists, it is appended. The file contains all emission energies in the si 'card' and the Line indicator, L is included, e.g. si1 L 0.01 (showing a first energy of 0.01 MeV). This is followed by the emission probability of each si entry. An additional text entry is made summing up the probabilities. NA's may be included and require user deletion. They facilitate writing the output in a convenient format.

See Also

[si_hist()] and [sp_hist()] if radioactive emission data is available in histogram form and needs formatting for MCNP input. [RN_screen_plot] may be used for a simple plot of output.

Other mcnp tools: `mcnp_cone_angle`, `mcnp_est_nps`, `mcnp_matrix_rotations`, `mcnp_plot_out_spec`, `mcnp_scan2spec`, `mcnp_si_hist`, `mcnp_sp_hist`

Examples

```
mcnp_si_sp_RD('Co-60', photon = TRUE, cut = 0.01, erg.dist = 13)
mcnp_si_sp_RD('Sr-90', rad_type = 'B-', cut = 0.01, erg.dist = 15)
mcnp_si_sp_RD('Am-241', rad_type = 'A', cut = 0.01, erg.dist = 23)
```

`mcnp_sp_hist` *energy distribution histogram entries*

Description

Make MCNP histogram probabilities for energy bins.

Usage

```
mcnp_sp_hist(bin_prob)
```

Arguments

`bin_prob`	A vector of the bin probabilities.

Value

A matrix of values for source probabilities to copy and paste into an MCNP input. (The # should be changed to the appropriate distribution number. The NA's in the last row should be discarded.)

See Also

[mcnp_si_hist()]

Other mcnp tools: `mcnp_cone_angle`, `mcnp_est_nps`, `mcnp_matrix_rotations`, `mcnp_plot_out_spec`, `mcnp_scan2spec`, `mcnp_si_hist`, `mcnp_si_sp_RD`

Examples

```
mcnp_sp_hist(rep(1 / 11, 11))
```

`neutron_geom_cf` *Solid Angle Correction for Neutron Detectors with Point Source*

Description

Correction factors are needed when an Neutron Rem Detector (NRD) aka "Remball" is used in close proximity to a points source. This formula is per ISO ISO 8529-2-2000 section 6.2. Note, however, that the ISO formula predicts the response. The formula used here takes the inverse to correct for the over-response.

Usage

```
neutron_geom_cf(l, r.d, del = 0.5)
```

Arguments

`l`	The distance from the center of the detector to the center of the source. Units of l and r.d must be consistent.
`r.d`	The detector radius. Value for typical NRD is 11 cm. An example is also provided with a Rem 500 detector with a radius of 4.5 cm.
`del`	The neutron effectiveness factor, default per ISO.

Value

The correction factor for solid angle.

See Also

Other rad measurements: `air_dens_cf`, `disk_to_disk_solid_angle`, `scaler_sim`, `tau_estimate`

Examples

```
neutron_geom_cf(l = 11.1, r.d = 11)
neutron_geom_cf(30, 11)
neutron_geom_cf(5, 4.5)
```

`photons_cs137_hist` *File Description:*

Description

This data file was generated in MCNP from a model of Gamma Well Irradiator with no attenuator in place. MCNP will include in the output a histogram of tally results when there is an E Tally Energy card. Results in the output up to MCNP version 6 have no headers, but the columns are:

Usage

```
photons_cs137_hist
```

Format

A `data.frame`

E_max Maximum Energy in MeV

bin_tally Tally result for this bin

R Monte Carlo uncertainty for this bin

`rate_meter_sim` *Ratemeter Simulation*

Description

Plot simulated ratemeter readings once per second for 600 seconds. The meter starts with a reading of zero and builds up based on the time constant. Resolution uncertainty is established to express the uncertainty from reading an analog scale, including the instability of its readings. Many standard references identify the precision or resolution uncertainty of analog readings as half of the smallest increment. This should be considered the single coverage uncertainty for a very stable reading. When a reading is not very stable, evaluation of the reading fluctuation is evaluated in terms of numbers of scale increments covered by meter indication over a reasonable evaluation period.

Usage

```
rate_meter_sim(cpm_equilibrium, meter_scale_increments, trials = 600,
  tau = 9.5, log_opt = "")
```

Arguments

`cpm_equilibrium`
: The expected count rate.

`meter_scale_increments`
: The meter scale increments.

`trials`
: Number of seconds to run simulation. Default = 600.

`tau`
: equal to the Resistance * Capacitance of the counting circuit. Units = seconds. Default set to 9.5, which provides 90 seconds. If the user does not know the time constant, but has an estimate of equilibrium in some time, use tau.estimate.

`log_opt`
: If logarithmic scale is needed, set to "y". If set to anything but blank (default), scale will be logarithmic.

Value

Plot of simulated meter reading every second..

Examples

```
rate_meter_sim(cpm_equilibrium = 270, meter_scale_increments = seq(100, 1000, 20))
rate_meter_sim(cpm_equilibrium = 2.7e5, meter_scale_increments = seq(2e5, 1e6, 2e4))
rate_meter_sim(450, seq(20, 1000, 20), trials = 1200, tau = 24.8534)
```

RN_index_screen — *Screen radionuclide data to find matches to decay mode, half-life, and total emission energy*

Description

Provides a set of radionuclides matching screening criteria. This is a limited screening based on average energy per transformation. Consider [search_phot_by_E], [search_alpha_by_E], and [search_beta_by_E] for spectroscopic measurement matching. For best results, assign results to a named object, then view the object.

Usage

```
RN_index_screen(dk_mode = NULL, min_half_life_seconds = NULL,
  max_half_life_seconds = NULL, min_E_alpha = NULL,
  min_E_electron = NULL, min_E_photon = NULL)
```

Arguments

`dk_mode`
: default = NULL #' select from: 'A' for Alpha 'B-' for Beta Negative 'B+' for Beta Positive 'EC' for Electron Capture 'IT' for Isomeric Transition 'SF' for Spontaneous Fission

`min_half_life_seconds`
: default = NULL. If half-life is known in units other than seconds, enter with conversion factor, e.g. for 15 minutes, enter min_half_life_seconds = 15 * 60.

max_half_life_seconds
: default = NULL. If half-life is known in units other than seconds, enter with conversion factor, e.g. for 30 minutes, enter max_half_life_seconds = 30 * 60.

min_E_alpha
: default = NULL. This will be used to screen the index for average alpha energy per decay, including all decay branches.

min_E_electron
: default = NULL. This will be used to screen the index for average electron energy per decay, including all decay branches.

min_E_photon
: default = NULL. This will be used to screen the index for average photon energy per decay, including all decay branches.

Value

data frame of radionuclide data from the RadData package index data (RadData::ICRP_07.NDX), matching search criteria.

See Also

Other radionuclides: `RN_Spec_Act`, `RN_info`, `RN_plt`, `bin_screen_phot`, `search_alpha_by_E`, `search_beta_by_E`, `search_phot_by_E`

Examples

```
RNs_selected <- RN_index_screen(dk_mode = "SF")
RNs_selected <- RN_index_screen(dk_mode = "IT", max_half_life_seconds = 433 * 3.15e7)
```

RN_info — *Quick table of Radionuclide Data from the RadData package*

Description

Access a quick summary of radionuclide data. This is for convenience only and does not replace a more comprehensive view as is available in the Radiological Toolbox <doi:10.2172/1201298>

Usage

```
RN_info(RN_select)
```

Arguments

RN_select
: identify the radionuclide of interest in the format "Es-254m"

Value

a table including half-life, decay modes, decay progeny, and branch fractions

See Also

Other radionuclides: RN_Spec_Act, RN_index_screen, RN_plt, bin_screen_phot, search_alpha_by_E, search_beta_by_E, search_phot_by_E

Examples

```
RN_info("Es-254m")
RN_info("Cf-252")
RN_info("Cs-137")
RN_info("Am-241")
```

RN_plt *Quick plot to radionuclide emission data screens.*

Description

Plots with ggplot2 with geom_point and log y-scale. Useful for a small set of radionclides. The point of this is to make it easy to get the plot by entering only the data frame name.

Usage

```
RN_plt(df)
```

Arguments

df	data frame of results including RN (radionuclide), energy in E_MeV and probability (prob) of photon.

Value

plot of spectrum

See Also

Other radionuclides: RN_Spec_Act, RN_index_screen, RN_info, bin_screen_phot, search_alpha_by_E, search_beta_by_E, search_phot_by_E

Examples

```
RN_plt(spec_0.1_0.3)
```

RN_Spec_Act *Specific Activity*

Description

Provides specific activity of a radionuclide in Bq/g.

Usage

```
RN_Spec_Act(RN_select)
```

Arguments

RN_select	identify the radionuclide of interest in the format "Es-254m"

Value

specific activity in Bq / g

See Also

Other radionuclides: RN_index_screen, RN_info, RN_plt, bin_screen_phot, search_alpha_by_E, search_beta_by_E, search_phot_by_E

Examples

```
RN_Spec_Act("Ac-230")
RN_Spec_Act("At-219")
RN_Spec_Act("Es-251")
RN_Spec_Act("Pd-96")
RN_Spec_Act("Te-117")
RN_Spec_Act("Ba-137m")
```

scaler_sim *Count Room Scaler Simulation*

Description

Returns a plotted distribution of results for a scaler model based on the Poisson distribution. Inputs and outputs in counts per minute.

Usage

```
scaler_sim(true_bkg, true_samp, ct_time, trials)
```

Arguments

true_bkg	True background count rate in counts per minute.
true_samp	True sample count rate in counts per minute.
ct_time	How many iterations of counting are performed.
trials	How many times to run the model.

Value

A histogram of all trial results including limits for +/- 1 standard deviation.

See Also

Other rad measurements: air_dens_cf, disk_to_disk_solid_angle, neutron_geom_cf, tau_estimate

Examples

```
scaler_sim(true_bkg = 50, true_samp = 10, ct_time = 1, trials = 1e5)
scaler_sim(true_bkg = 50, true_samp = 30, ct_time = 1, trials = 1e5)
```

---

scan2spec.df — *Copy and paste MCNP output spectral data for use with 'mcnp_plot_out_spec.R'*

Description

Provides quick copy-and-paste conversion to data frame. Conversion is based on the three columns with results from MCNP output in energy bins with maximum energy in MeV in the first column, the mean result for the bin in the second column, and relative Monte Carlo uncertainty in the third column.

Usage

```
scan2spec.df()
```

Value

spectrum file with maximum energy and MCNP bin value

Examples

```
# Since this function requires the user
# to copy and paste input, this example
# is set up to provide data for this purpose.
# To run the example, copy and paste the following
# into an input file and delete the hash tags to run.
# my_hist_data <- my_scan_fun()
# 0.1000000 3.133122e-05 0.3348260
```

```
# 0.4222222 6.731257e-05 0.2017546
# 0.7444444 5.249198e-05 0.4524577
# 1.0666667 2.046046e-04 0.4201954
# 1.3888889 1.525125e-03 0.8049388
# 1.7111111 2.922743e-05 0.7985399
# 2.0333333 5.162954e-03 0.1974694
# 2.3555556 2.048186e-05 0.5011170
# 2.6777778 1.468040e-04 0.7248116
# 3.0000000 1.037092e-04 0.7659850
```

---

search_alpha_by_E *Search for alpha*

Description

Search for alpha emission based on energy, half-life and minimum probability.

Usage

```
search_alpha_by_E(E_min, E_max, min_half_life_seconds = NULL,
  max_half_life_seconds = NULL, min_prob = 0)
```

Arguments

E_min	minimum energy in MeV, default = 0
E_max	maximum energy in MeV, default = 10
min_half_life_seconds	minimum half-life in seconds. Use multiplier as needed, e.g. 3 * 3600 for 3 hours. Default = NULL,
max_half_life_seconds	maximum half-life. See min_half_life_seconds.
min_prob	minimum probability with default = 0.

Value

search results in order of half-life. Recommend assigning results to a viewable object, such as 'search_results'

See Also

[RN_plt()]

Other radionuclides: RN_Spec_Act, RN_index_screen, RN_info, RN_plt, bin_screen_phot, search_beta_by_E, search_phot_by_E

Examples

```
# between 7 and 8 MeV
search_results <- search_alpha_by_E(7, 8)

#1-4 MeV; half-life between 1 and 4 hours
search_results <- search_alpha_by_E(1, 4, 1 * 3600, 4 * 3600)

# between 7 and 10 MeV with at least 1e-3 probability
search_results <- search_alpha_by_E(7, 10, min_prob = 1e-3)
```

`search_beta_by_E` *Search for beta*

Description

Search for beta emission based on maximum energy and half-life.

Usage

```
search_beta_by_E(E_max, min_half_life_seconds = NULL,
  max_half_life_seconds = NULL)
```

Arguments

`E_max` maximum energy in MeV, default = 10

`min_half_life_seconds`
: minimum half-life in seconds. Use multiplier as needed, e.g. 3 * 3600 for 3 hours. Default = NULL,

`max_half_life_seconds`
: maximum half-life. See min_half_life_seconds.

Value

search results in order of half-life. Recommend assigning results to a viewable object, such as 'search_results'

See Also

[RN_plt()]

Other radionuclides: `RN_Spec_Act`, `RN_index_screen`, `RN_info`, `RN_plt`, `bin_screen_phot`, `search_alpha_by_E`, `search_phot_by_E`

Examples

```
# Max beta at least 2 MeV
search_results <- search_beta_by_E(2)
# Max beta at least 2 MeV and half-life between 1 s and 1 h
search_results <- search_beta_by_E(2, 1, 3600)

# Max beta at least 1 MeV and half-life between 1 d and 2 d
search_results <- search_beta_by_E(1, 3600 * 24, 2 * 3600 * 24)
```

`search_phot_by_E` *Search for photon*

Description

Search for photon emission based on energy, half-life and minimum probability.

Usage

```
search_phot_by_E(E_min = 0, E_max = 10, min_half_life_seconds = NULL,
  max_half_life_seconds = NULL, min_prob = 0)
```

Arguments

`E_min` minimum energy in MeV, default = 0

`E_max` maximum energy in MeV, default = 10

`min_half_life_seconds` minimum half-life in seconds. Use multiplier as needed, e.g. 3 * 3600 for 3 hours. Default = NULL,

`max_half_life_seconds` maximum half-life. See min_half_life_seconds.

`min_prob` minimum probability with default = 0.

Value

search results in order of half-life. Recommend assigning results to a viewable object, such as 'search_results'

See Also

[RN_plt()]

Other radionuclides: `RN_Spec_Act`, `RN_index_screen`, `RN_info`, `RN_plt`, `bin_screen_phot`, `search_alpha_by_E`, `search_beta_by_E`

Examples

```
# between 1 and 1.2 MeV, between 6 and 6.2 hours half-life,
#... probability at least 1e-4
search_results <- search_phot_by_E(1, 1.2, 6 * 3600, 6.2 * 3600, 1e-4)

# between 0.1 and 0.15 MeV, between 1 and 3 million years half-life
search_results <- search_phot_by_E(0.1, 0.15, 1e6 * 3.153e7, 3e6 * 3.153e7)
```

---

spec_0.1_0.3 *Output from example in [RN_bin_screen_phot]*

Description

Available for follow-up example in [RN_bin_screen_plot]

Usage

```
spec_0.1_0.3
```

Format

A `data.frame`

RN Radionuclide name

code_AN radiation type alpha-numeric

E_MeV photon energy in mega-electronvolts

prob probability of this photon per transformation

code_num radiation type number

is_photon TRUE if photon

---

stay_time *Stay time for radiation work.*

Description

Calculate stay time for radiation work.

Usage

```
stay_time(dose_rate, dose_allowed, margin = 20)
```

Arguments

`dose_rate`	Dose rate per hour for the work - units consistent with dose allowance, e.g. mRem/h, microSv/h.
`dose_allowed`	Dose that can not be exceeded for this job.
`margin`	Percent margin to protect limit, default = 20 percent.

Value

Time in minutes allowed for the work.

Examples

```
stay_time(dose_rate = 100, dose_allowed = 50, margin = 20)
```

`tau_estimate` *Estimate tau parameter for [ratemeter_sim]*

Description

If the time constant is not known, but the vendor specifies that the ratemeter will reach some percentage of equilibrium in some number of seconds, use this function to estimate tau.

Usage

```
tau_estimate(pct_eq, t_eq)
```

Arguments

`pct_eq`	Percent equilibrium
`t_eq`	Time, in seconds, to the given percent equilibrium is achieved.

Value

tau, the time constant, in seconds.

See Also

Other rad measurements: `air_dens_cf`, `disk_to_disk_solid_angle`, `neutron_geom_cf`, `scaler_sim`

Examples

```
tau_estimate(pct_eq = 90, t_eq = 22)
```

Index