Package 'SHIELD'

September 9, 2024

Title What the Package Does (One Line, Title Case)

Version 0.0.0.9000

Description What the package does (one paragraph).

License `use_mit_license()`, `use_gpl3_license()` or friends to pick a license

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

R topics documented:

do.get.age.contact.proportions.for.model

do.get.age.contact.proportions.for.model

Description

returns a list of age contact proportions for designated group

Usage

```
do.get.age.contact.proportions.for.model(
  specification.metadata,
  location,
  age.mixing.sd.mult,
  age.model,
  age.counts,
  availability
)
```

Arguments

specification.metadata	specification.metadata
location	location
age.mixing.sd.mult	multiplier of the standard deviation of the age mixing model (diff_ages_partners ~ Normal(mu,sd)) used for calibration by calling do.get.age.contact.proportions.for.model
age.model	specific age.model used to inform new partnership probabilities

Value

OUTPUT_DESCRIPTION

functions.sexual.contact.model

1.functions.sexual.contact.model

Description

sexual contacts are charactrized via 4 components: 1)transmission probability, 2)age mixing, 3)sex mixing, and 4)race mixing

Usage

```
functions.sexual.contact.model()
```

References

'inst/docs/sexual_contacts.docx'

get.base.initial.female.population

get.base.initial.female.population

Description

Generates the size of the 'female' population for the given years by calling `get.base.initial.population.for.sex` for sex-specific population data.

Usage

```
get.base.initial.female.population(
  location,
  specification.metadata,
  years = DEFAULT.POPULATION.YEARS
)
```

Arguments

`location`	The location for which the population data is being retrieved.
`specification.metadata`	Metadata for specification.
`years`	Vector of years for which to retrieve population data. Default is `DEFAULT.POPULATION.YEARS`.

Value

A 2D matrix showing the number of persons broken down by race (columns) within each age group (rows).

Examples

```
get.base.initial.female.population("C.12580", specification.metadata)
where: specification.metadata=get.specification.metadata("shield", "C.12580")
```

get.base.initial.male.population

get.base.initial.male.population

Description

Generates the size of the 'male' population for the given years by calling `get.base.initial.population.for.sex` for sex-specific population data.

Usage

```
get.base.initial.male.population(
  location,
  specification.metadata,
  years = DEFAULT.POPULATION.YEARS
)
```

Arguments

location	The location for which the population data is being retrieved.
specification.metadata	Metadata for specification.
years	Vector of years for which to retrieve population data. Default is DEFAULT.POPULATION.YEARS.

Value

A 2D matrix showing the number of persons broken down by race (columns) within each age group (rows).

---

get.base.initial.population.for.sex *get.base.initial.population.for.sex*

Description

Generates the size of the population for the given years based on sex.

Usage

```
get.base.initial.population.for.sex(
  location,
  specification.metadata,
  sex,
  years = DEFAULT.POPULATION.YEARS
)
```

Arguments

location	The location for which the population data is being retrieved.
specification.metadata	Metadata for specification.
sex	The sex of the designated population ('male' or 'female').
years	Vector of years for which to retrieve population data. Default is DEFAULT.POPULATION.YEARS.

Value

A vector of population for the given years.

`get.best.guess.msm.proportions`
get.best.guess.msm.proportions

Description

generates proportion of male who are msm by race

Usage

```
get.best.guess.msm.proportions(
  location,
  specification.metadata,
  years = 2013,
  ages = specification.metadata$dim.names$age,
  keep.age = T,
  keep.race = T,
  return.proportions = T
)
```

Arguments

`location`	location
`specification.metadata`	specification.metadata
`years`	years #Todd: which years are these
`ages`	agegroups read from specification.metadata$dim.names$age #Todd: isn't this redundant?
`keep.age`	keep age #Todd: what does this mean?
`keep.race`	keep race #Todd: what does this mean?
`return.proportions`	return proportions or frequency

Value

a 2D matrix showing the proportion of MSM by age (rows) and race (columns)

`get.best.guess.msm.proportions.by.race`
get.best.guess.msm.proportions.by.race

Description

assumes that within each county, relative risks of being MSM are as in MSM.PROPORTIONS and total risk of being MSM is as per read.msm.proportions

Usage

```
get.best.guess.msm.proportions.by.race(
  location,
  specification.metadata,
  min.age = 0,
  years = DEFAULT.POPULATION.YEARS,
 msm.proportions.by.race = c(black = 1 - 0.806, hispanic = 1 - 0.854, white = 1 - 0.848,
   other = 1 - 0.802),
  return.proportions = T,
  keep.ages = F
)
```

Arguments

location	location
specification.metadata	specification.metadata
min.age	min.age #Todd: what does this mean?
years	years #Todd: which years are these
msm.proportions.by.race	msm.proportions.by.race #Todd: what does this mean?
return.proportions	return proportions or frequency
keep.ages	keep.ages #Todd: what does this mean?

Value

a 2D matrix showing the proportion of MSM by age (rows) and race (columns) #Todd: true?

---

get.female.sexual.age.contact.proportions
1.1.get.female.sexual.age.contact.proportions

Description

returns a list of age contact proportions for females

Usage

```
get.female.sexual.age.contact.proportions(
  age.mixing.sd.mult,
  single.year.female.age.counts,
  single.year.age.sexual.availability,
  specification.metadata
)
```

Arguments

`age.mixing.sd.mult`
: multiplier of the standard deviation of the age mixing model (diff_ages_partners ~ Normal(mu,sd)) used for calibration by calling `do.get.age.contact.proportions.for.model`

`single.year.female.age.counts`
: number of individuals within each nominal age year for each age group

`single.year.age.sexual.availability`
: proportion of individuals within each nominal age year available engaged in sexual activity

`specification.metadata`
: specification.metadata

Value

OUTPUT_DESCRIPTION

References

'`inst/docs/sexual_contacts.docx`'

`get.female.single.year.age.counts` *get.female.single.year.age.counts*

Description

return counts of female in a single year

Usage

```
get.female.single.year.age.counts(
  location,
  population.years = DEFAULT.POPULATION.YEARS
)
```

Arguments

`location` location

`population.years`
: PARAM_DESCRIPTION, Default: DEFAULT.POPULATION.YEARS #Todd?

Value

OUTPUT_DESCRIPTION

get.geographically.aggregated.race.oes

get.geographically.aggregated.race.oes

Description

within.county.race.oesto,from is how much more likely someone it is for a person of race to to have a partner of race from, relative to race from's population prevalence

Usage

```
get.geographically.aggregated.race.oes(
  location,
  specification.metadata,
  within.county.race.oes,
  years = DEFAULT.POPULATION.YEARS,
  as.functional.form = T
)
```

Arguments

location	location
specification.metadata	specification.metadata
within.county.race.oes	observed to expected ration for mixing by age within each county
years	PARAM_DESCRIPTION, Default: DEFAULT.POPULATION.YEARS #Todd???
as.functional.form	PARAM_DESCRIPTION, Default: T #Todd???

Value

todd??

get.heterosexual.male.sexual.age.contact.proportions

get.heterosexual.male.sexual.age.contact.proportions

Description

returns a list of age contact proportions for het male

Usage

```
get.heterosexual.male.sexual.age.contact.proportions(
  age.mixing.sd.mult,
  single.year.heterosexual.male.age.counts,
  single.year.age.sexual.availability,
  specification.metadata
)
```

Arguments

`age.mixing.sd.mult`
: multiplier of the standard deviation of the age mixing model (diff_ages_partners ~ Normal(mu,sd)) used for calibration by calling `do.get.age.contact.proportions.for.model`

`single.year.age.sexual.availability`
: proportion of individuals within each nominal age year available engaged in sexual activity

`specification.metadata`
: specification.metadata

Value

OUTPUT_DESCRIPTION

`get.heterosexual.male.single.year.age.counts` *get.heterosexual.male.single.year.age.counts*

Description

To determine the proportion of the population that falls into specific age buckets

Usage

```
get.heterosexual.male.single.year.age.counts(
  location,
  specification.metadata,
  population.years = DEFAULT.POPULATION.YEARS
)
```

Arguments

`location` location

`specification.metadata`
: specification.metadata

`population.years`
: PARAM_DESCRIPTION, Default: DEFAULT.POPULATION.YEARS #Todd???

Value

OUTPUT_DESCRIPTION

`get.male.single.year.age.counts` *get.male.single.year.age.counts*

Description

return counts of male in a single year

Usage

```
get.male.single.year.age.counts(
  location,
  population.years = DEFAULT.POPULATION.YEARS
)
```

Arguments

`location`	location
`population.years`	PARAM_DESCRIPTION, Default: DEFAULT.POPULATION.YEARS #Todd?

Value

OUTPUT_DESCRIPTION

`get.msm.sexual.age.contact.proportions` *get.msm.sexual.age.contact.proportions*

Description

returns a list of age contact proportions for msm

Usage

```
get.msm.sexual.age.contact.proportions(
  age.mixing.sd.mult,
  single.year.msm.age.counts,
  single.year.age.sexual.availability,
  specification.metadata
)
```

Arguments

`age.mixing.sd.mult`	multiplier of the standard deviation of the age mixing model (diff_ages_partners ~ Normal(mu,sd)) used for calibration by calling `do.get.age.contact.proportions.for.model`
`single.year.age.sexual.availability`	proportion of individuals within each nominal age year available engaged in sexual activity
`specification.metadata`	specification.metadata

Value

OUTPUT_DESCRIPTION

`get.msm.single.year.age.counts` *get.msm.single.year.age.counts*

Description

return counts of msm in a single year

Usage

```
get.msm.single.year.age.counts(
  location,
  specification.metadata,
  population.years = DEFAULT.POPULATION.YEARS
)
```

Arguments

`location`	location
`population.years`	PARAM_DESCRIPTION, Default: DEFAULT.POPULATION.YEARS #Todd?

Value

OUTPUT_DESCRIPTION

`get.proportion.msm.of.male.by.race.functional.form` *get.proportion.msm.of.male.by.race.functional.form*

Description

Generates proportion of male who are msm by race

Usage

```
get.proportion.msm.of.male.by.race.functional.form(
  location,
  specification.metadata
)
```

Arguments

`location`	location
`specification.metadata`	specification.metadata

Value

??? #Todd??

`get.sexual.availability` *1-get.sexual.availability*

Description

Determines the proportion of people in each age bucket that are sexually available

Usage

```
get.sexual.availability()
```

Details

The model reflects an increase in sexual activity starting from age 13, reaching 100% at ages 20 to 64, and gradually tapering off until age 85, the final age group.

Value

1D vector with proportion of people in each age bucket that are sexually available

---

`oes.to.proportions` *oes.to.proportions*

Description

trasforming oe values to proportions of mixing with other groups (sum to 1)

Usage

```
oes.to.proportions(oes, population)
```

Arguments

`oes`	oes betwen groups
`population`	popualtion count in each group

Value

todd??

`sexual.oes.to.contact.proportions` *sexual.oes.to.contact.proportions*

Description

trasforming oe values to proportions of mixing with other races (sum to 1) calling code `oes.to.proportions`

Usage

```
sexual.oes.to.contact.proportions(race.sexual.oes, race.population.counts)
```

Arguments

`race.sexual.oes`	oes
`race.population.counts`	number of people by race

Value

todd??