https://nataliaciria.com/mcmodule/
Available on CRAN!

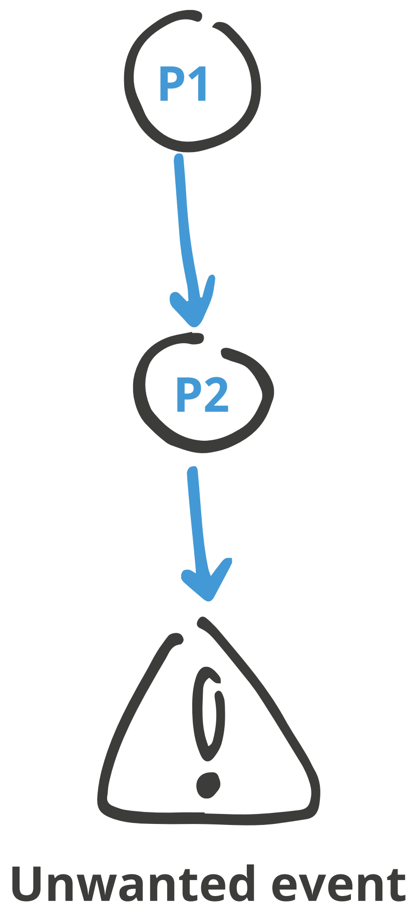
mcmodule::

An R Package for Multi-Pathway Monte Carlo Risk Assessment

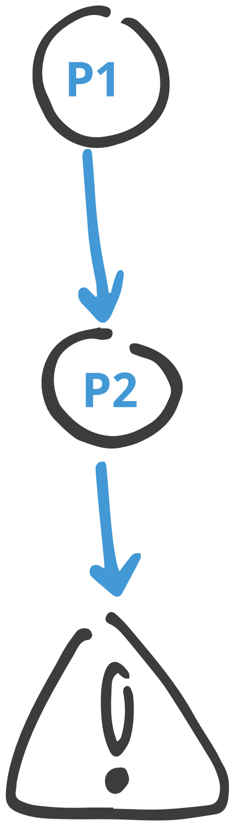
R!sk conference 2026 – 19th February

Natalia Ciria
Veterinary Epidemiology PhD Student
Universitat Autònoma de Barcelona (UAB)
natalia.ciria@uab.cat

Quantifying risk through probability steps is **neat**



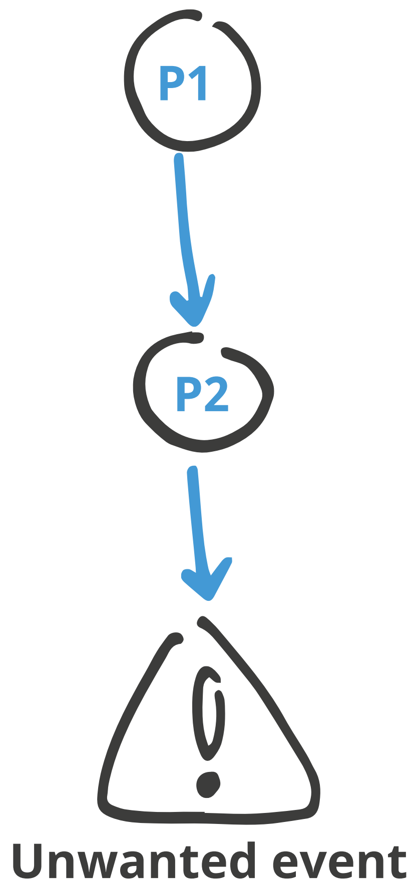
With mc2d
Quantifying risk through **stochastic** probability steps is **neat**



Unwanted event

```r
library(mc2d)
# Probability an animal is infected
inf_animal <- mcstoc(runif, min = 0.002, max = 0.08)


# Probability an infected animal is tested
test_animal <- mcstoc(rpert, min = 0.4, mode = 0.5, max = 0,6)
test_inf_animal <- inf_animal * test_animal


# Probability an animal infected animal is not detected
test_se <- mcstoc(runif, min = 0.8, max = 0.95)
not_detected <- test_inf_animal * (1 - test_se)
```

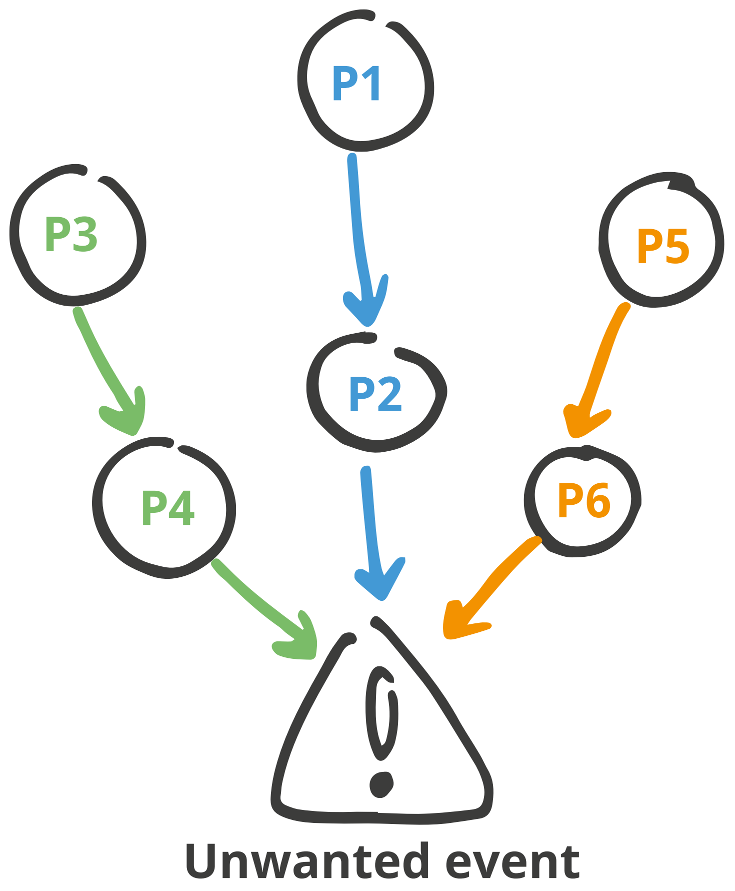
With `mc2d`
Quantifying risk through **stochastic** probability steps is **neat**



Thanks to `mc2d`

Quantifying risk through stochastic probability steps is **neat**

But real-life risk pathways are **messy**

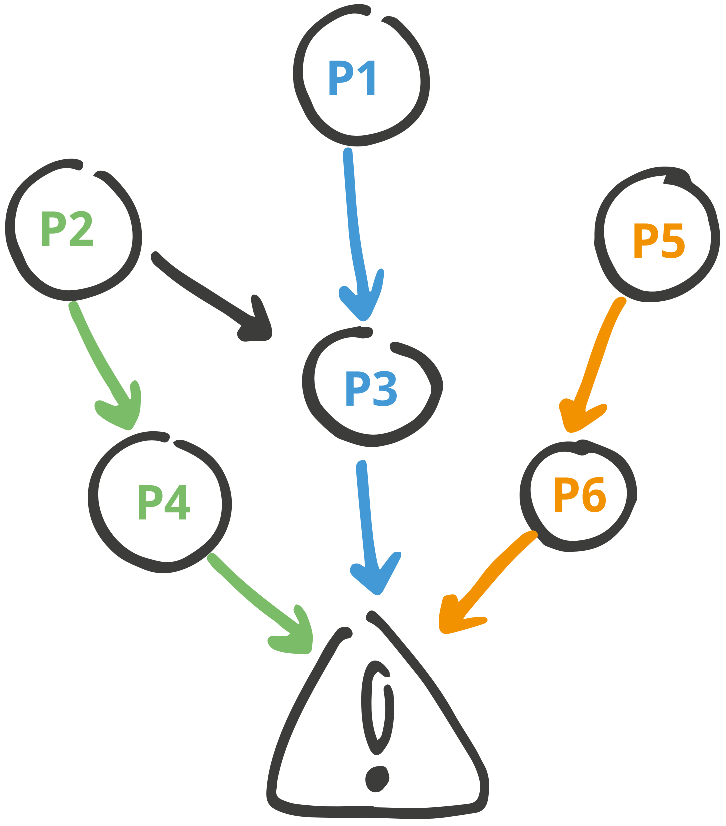


Thanks to `mc2d`

Quantifying risk through stochastic probability steps is **neat**

But real-life risk pathways are **messy**

interconnected



Unwanted event

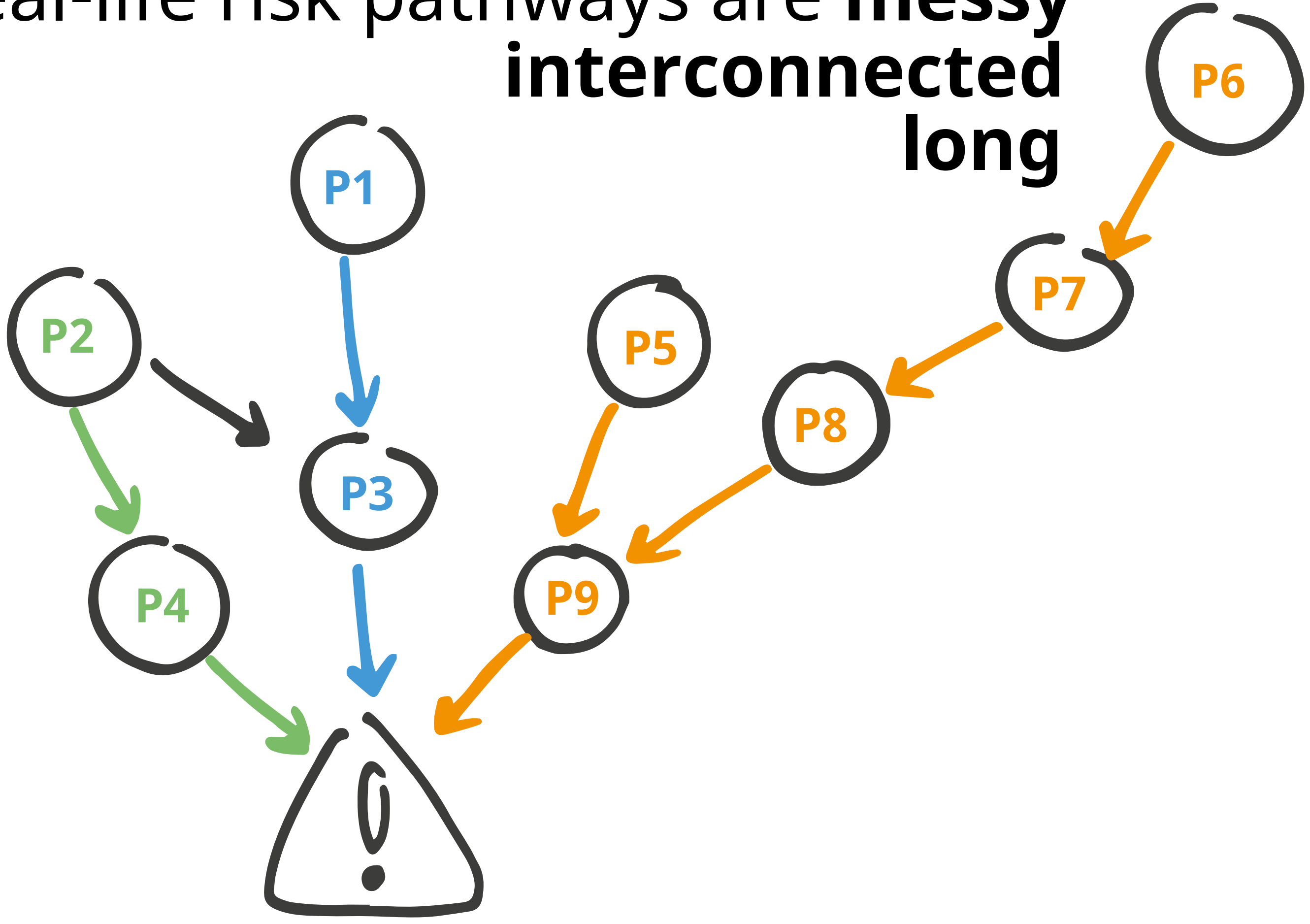
Thanks to `mc2d`

Quantifying risk through stochastic probability steps is **neat**

But real-life risk pathways are **messy**

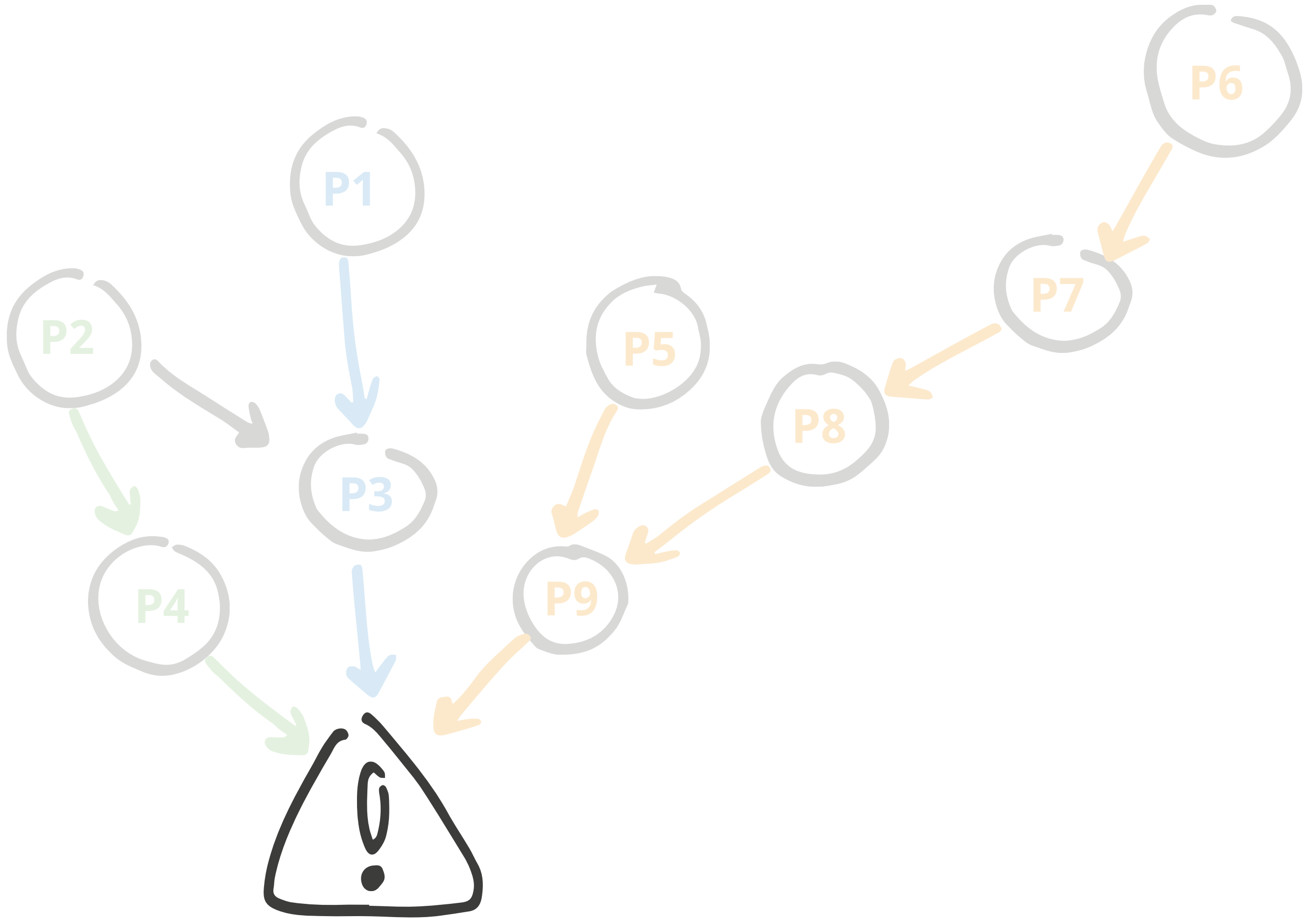
interconnected

long



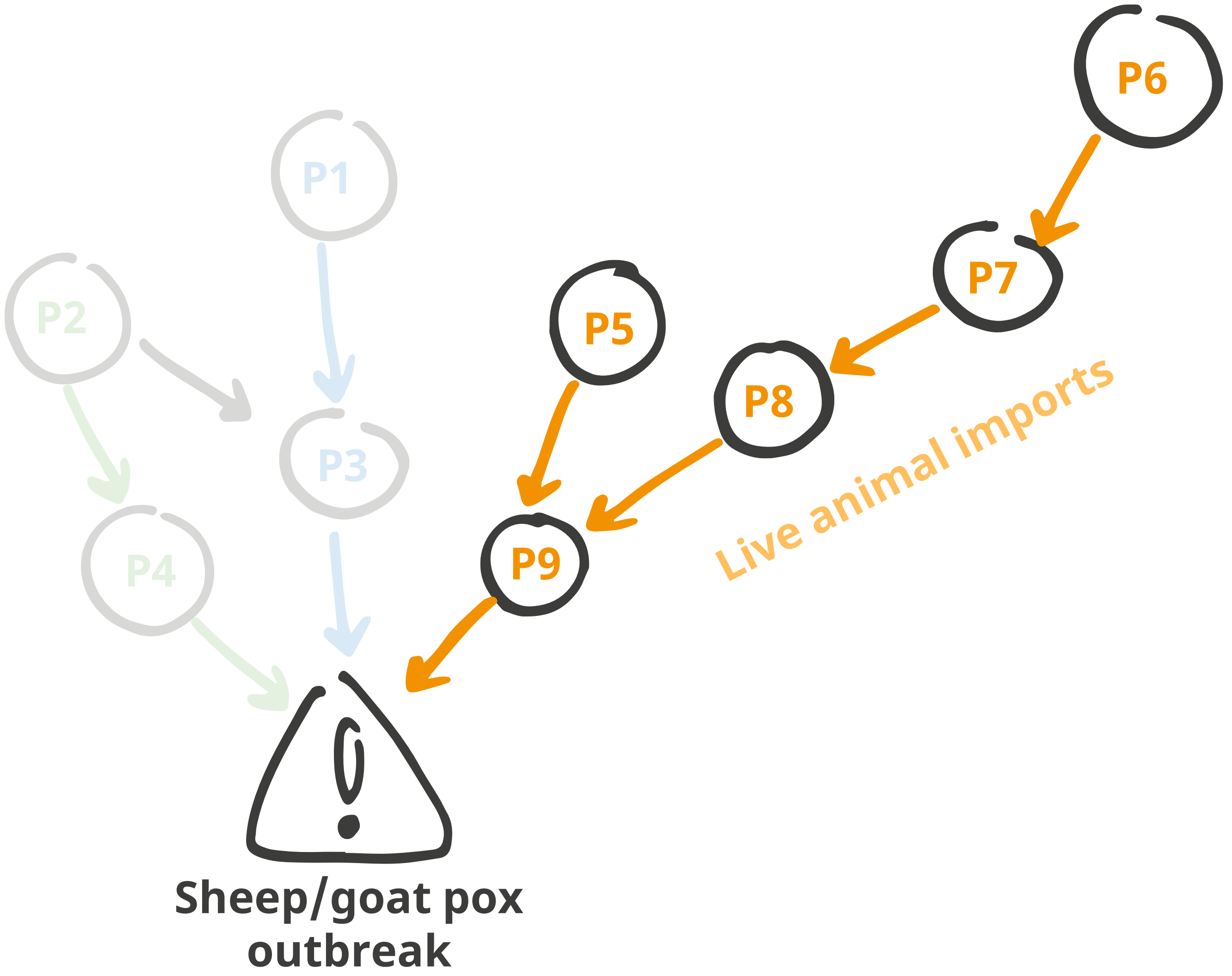
Unwanted event

For example

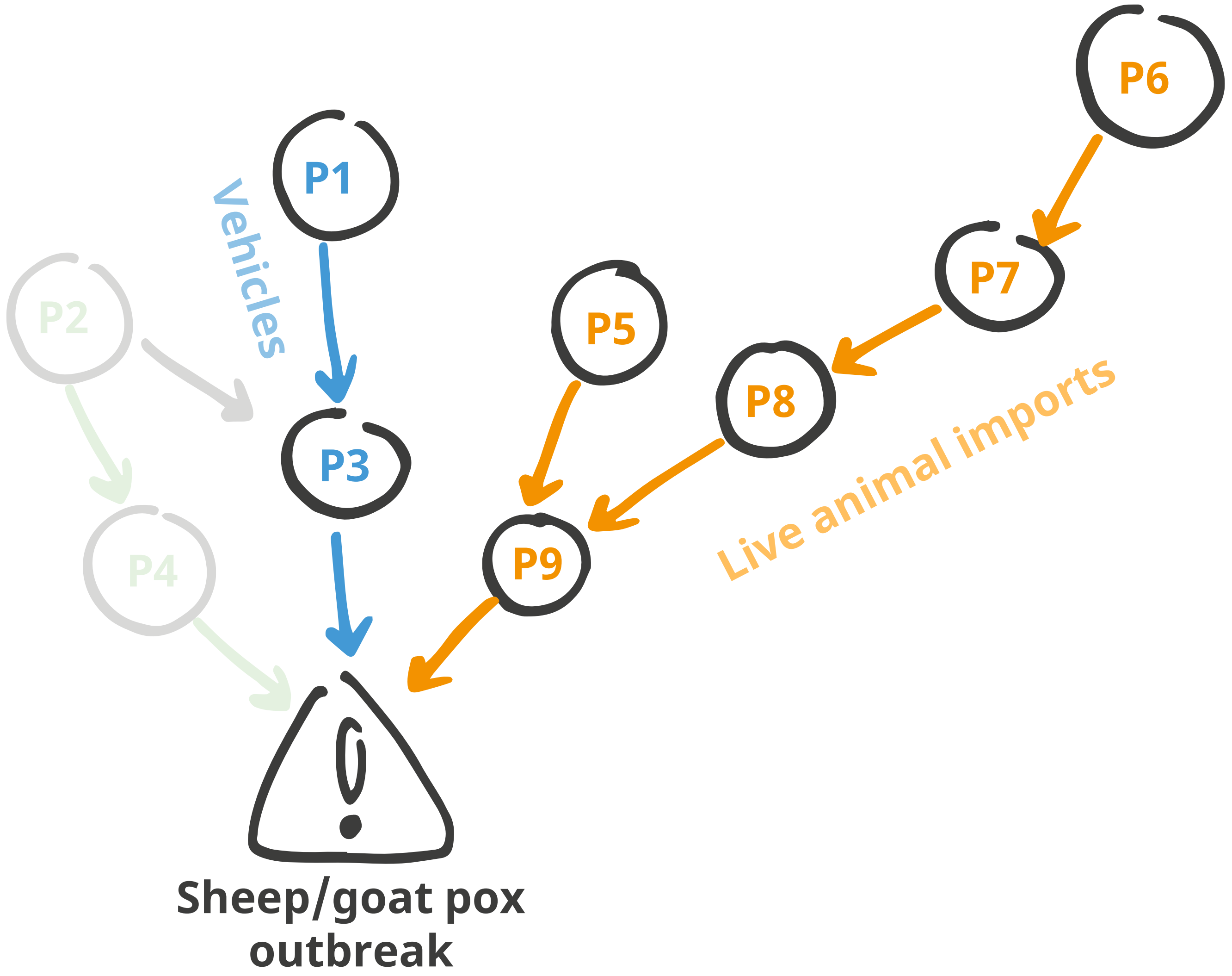


Sheep/goat pox outbreak

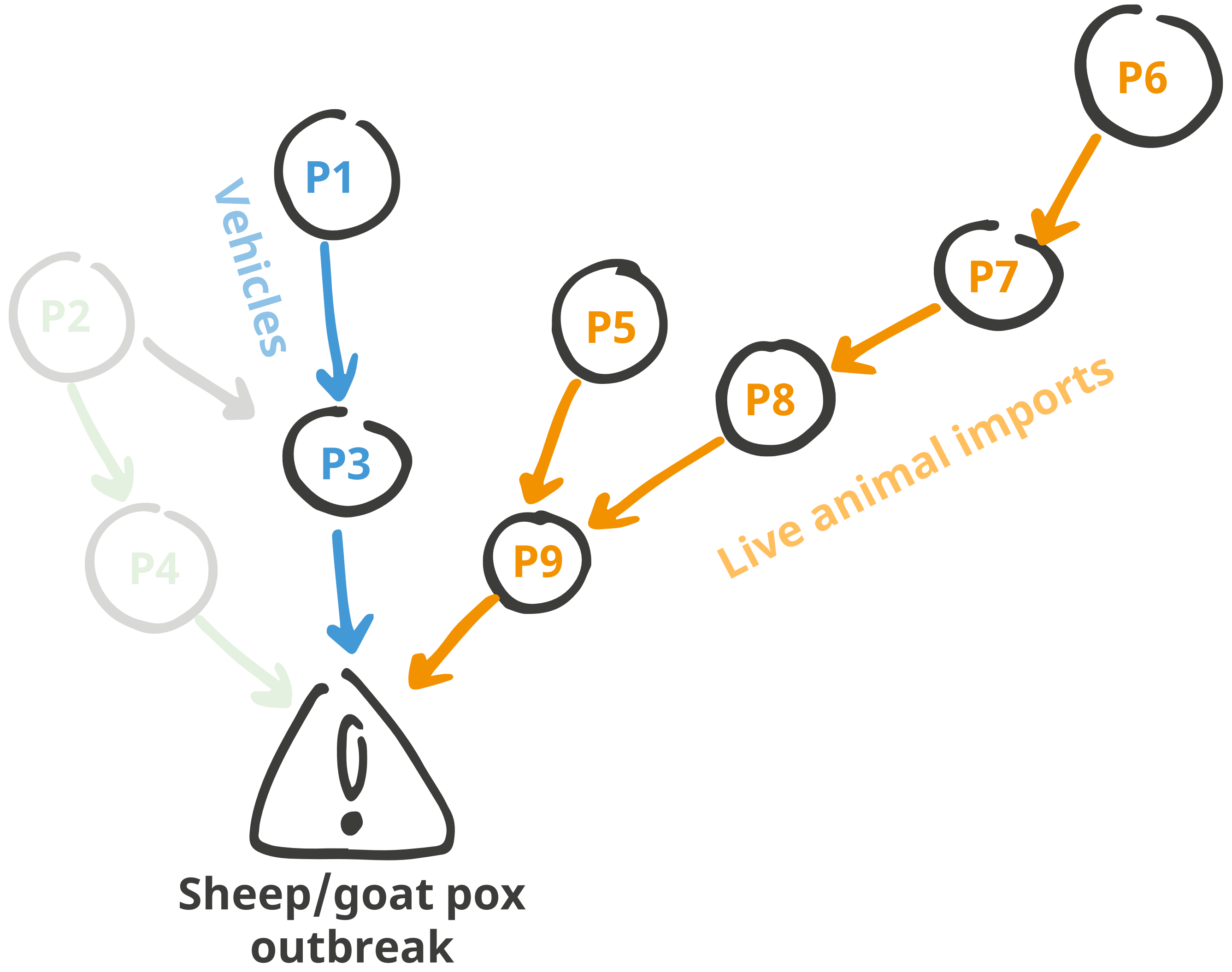
For example


P6
P1
P7
P5
P2
P8
P3
Live animal imports
P9
P4
Sheep/goat pox outbreak

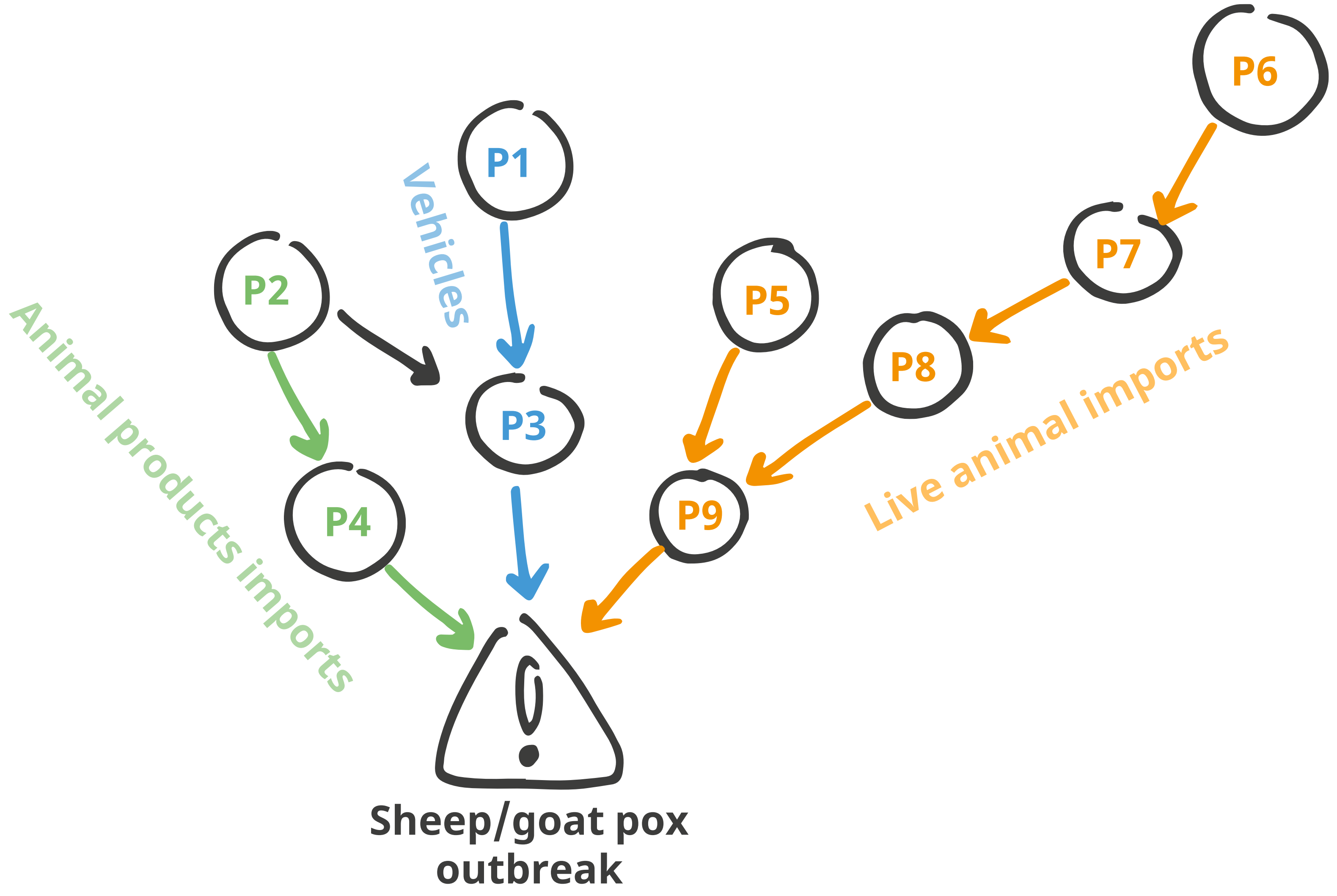
For example


P1
P2
P3
P4
P5
P6
P7
P8
P9
Vehicles
Live animal imports
Sheep/goat pox outbreak

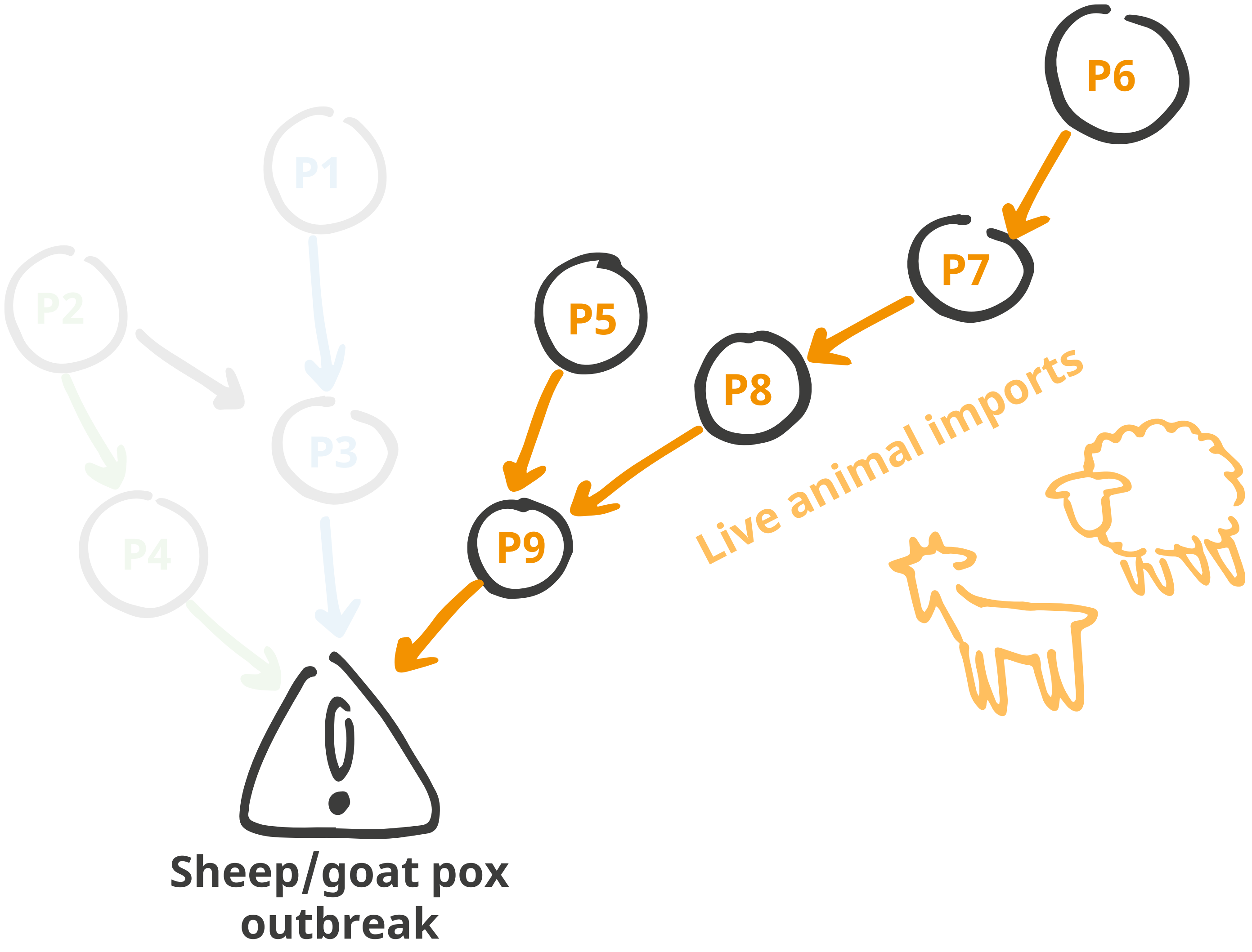
For example


P1
Vehicles
P2
P3
P4
P5
P6
P7
P8
P9
Live animal imports
Sheep/goat pox outbreak

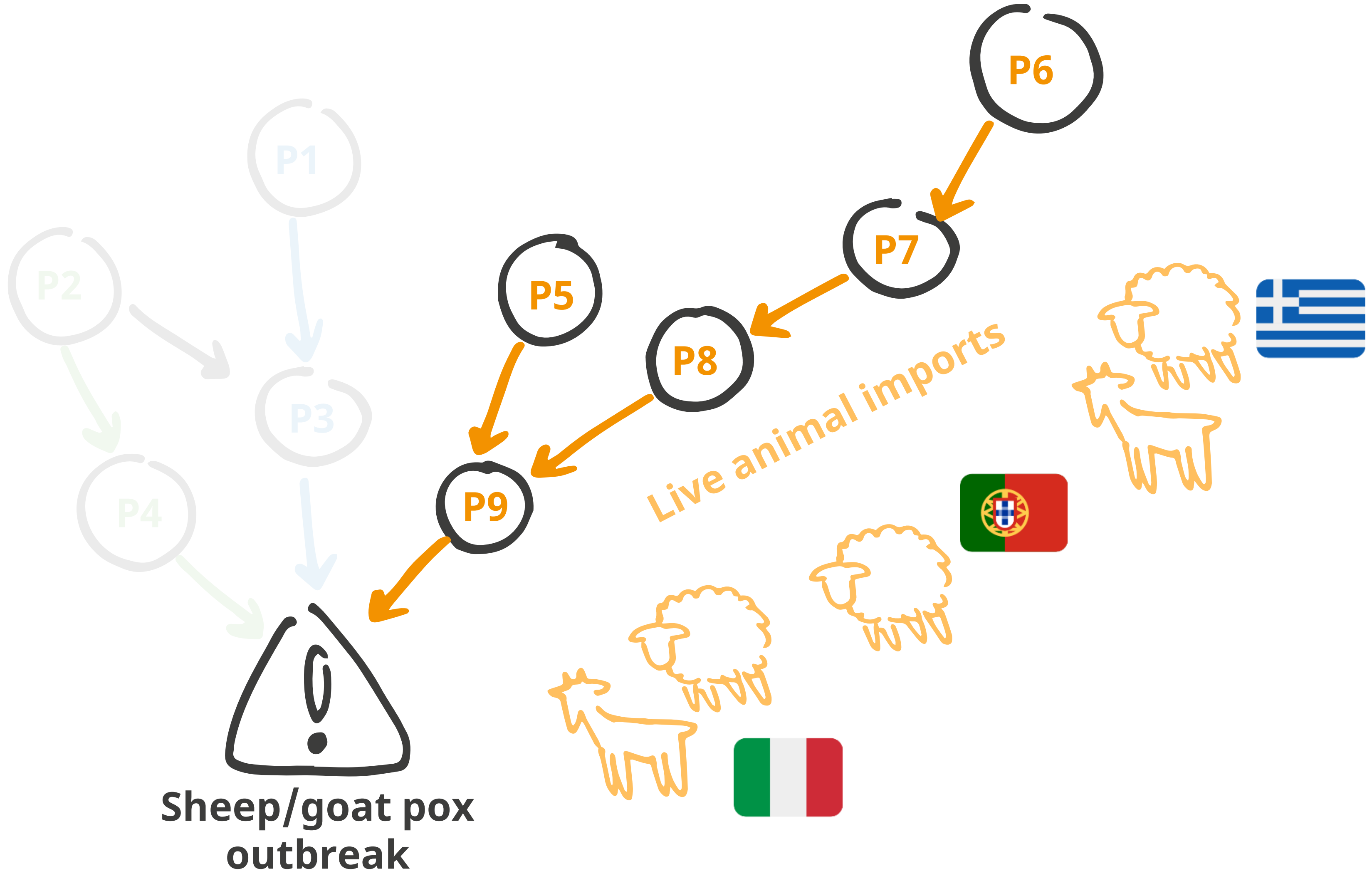
For example


P1
P2
P3
P4
P5
P6
P7
P8
P9
Vehicles
Animal products imports
Live animal imports
Sheep/goat pox outbreak

For example

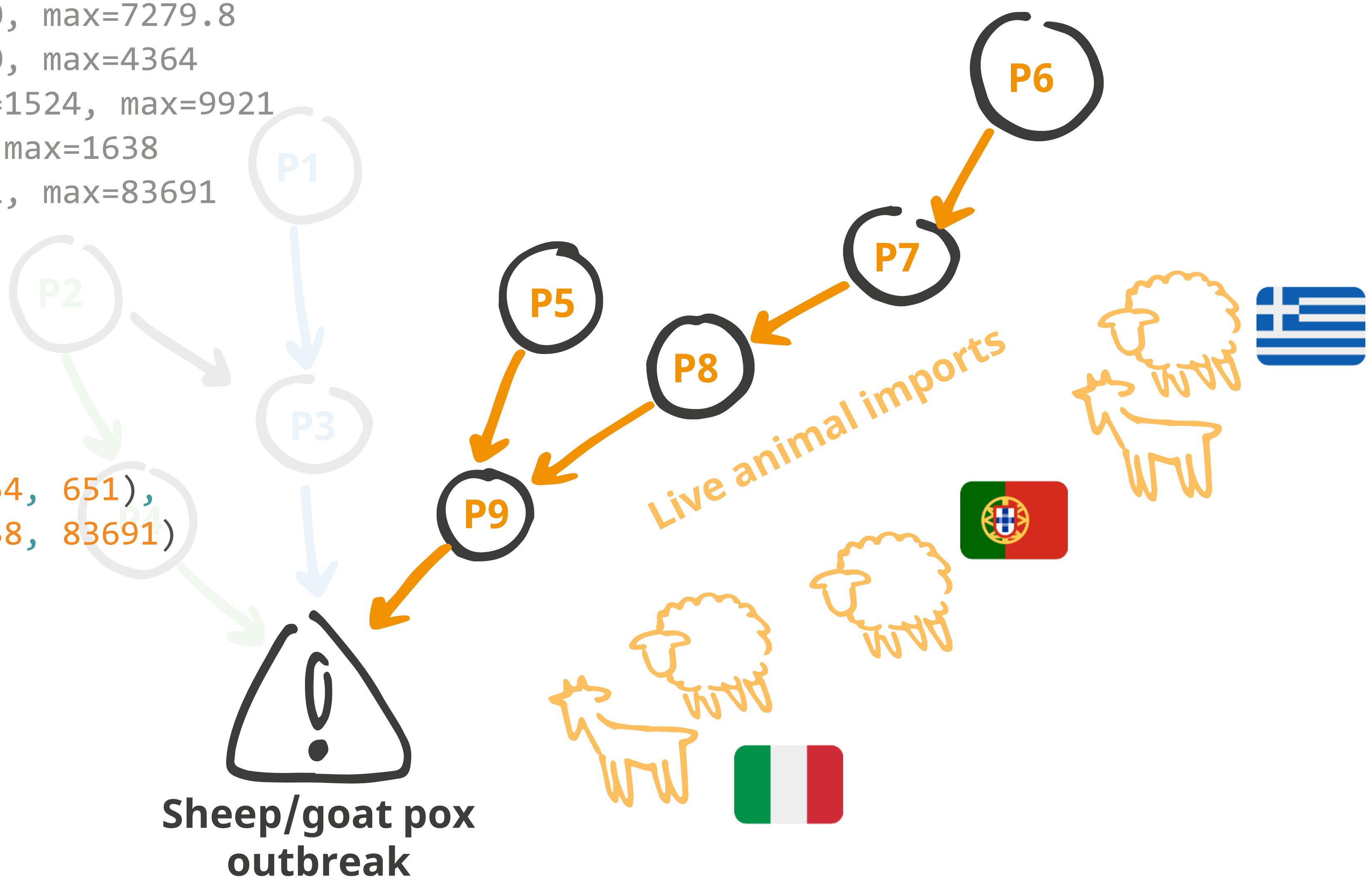

P1
P2
P3
P4
P5
P6
P7
P8
P9
Live animal imports
Sheep/goat pox outbreak

For example

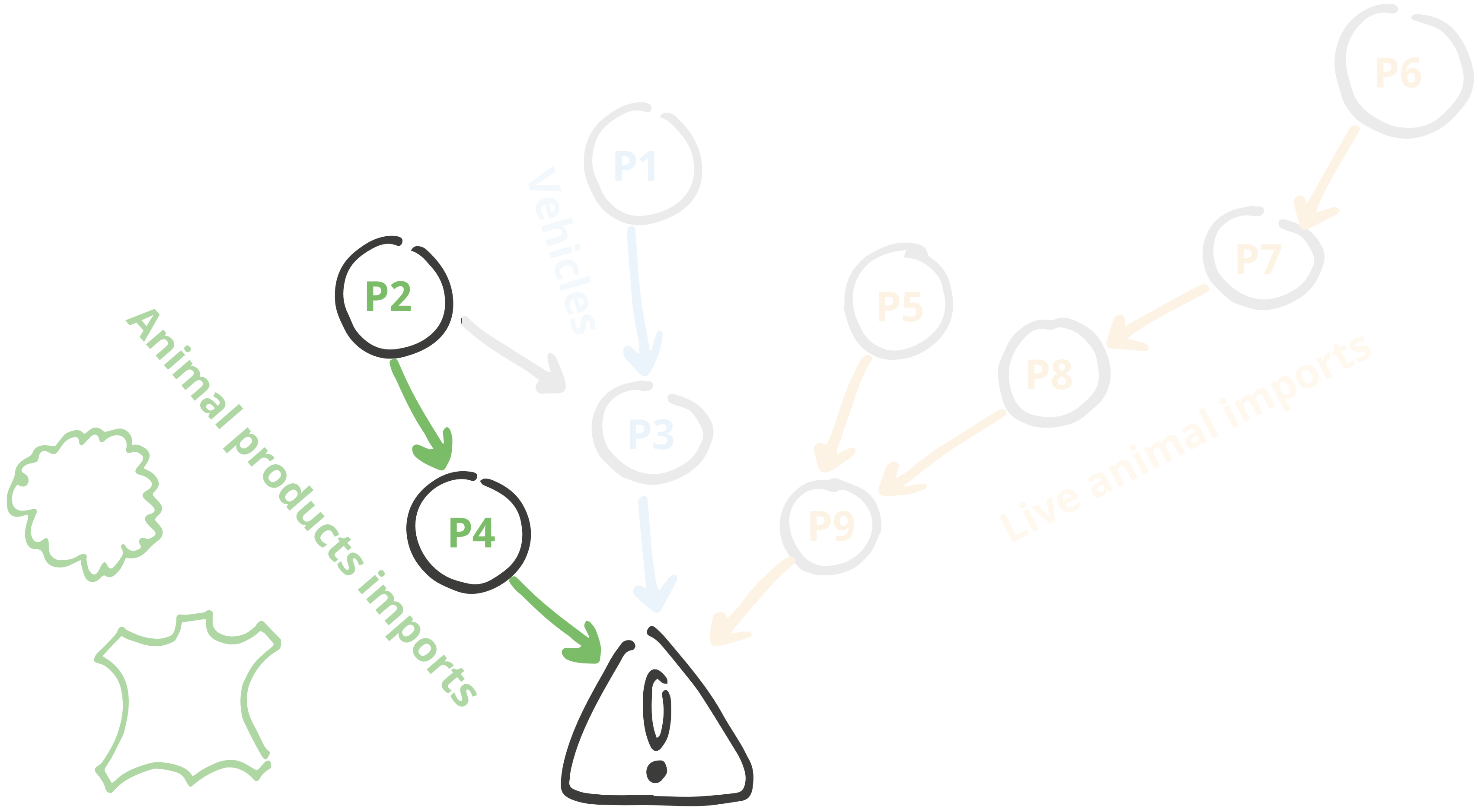

P1
P2
P3
P4
P5
P6
P7
P8
P9
Live animal imports
Sheep/goat pox outbreak

For example

```
# GRC - sheep: min=1, mode=1250, max=7279.8
# GRC - goat: min=74, mode=1650, max=4364
# ITA - sheep: min=10.38, mode=1524, max=9921
# ITA - goat: min=6, mode=154, max=1638
# PRT - sheep: min=49, mode=651, max=83691


animal_import_qty <- mcstoc(
  rpert,
  nvariates = 5,
  min = c(1, 74, 10.38, 6, 49),
  mode = c(1250, 1650, 1524, 154, 651),
  max = c(7279, 4364, 9921, 1638, 83691)
)
```

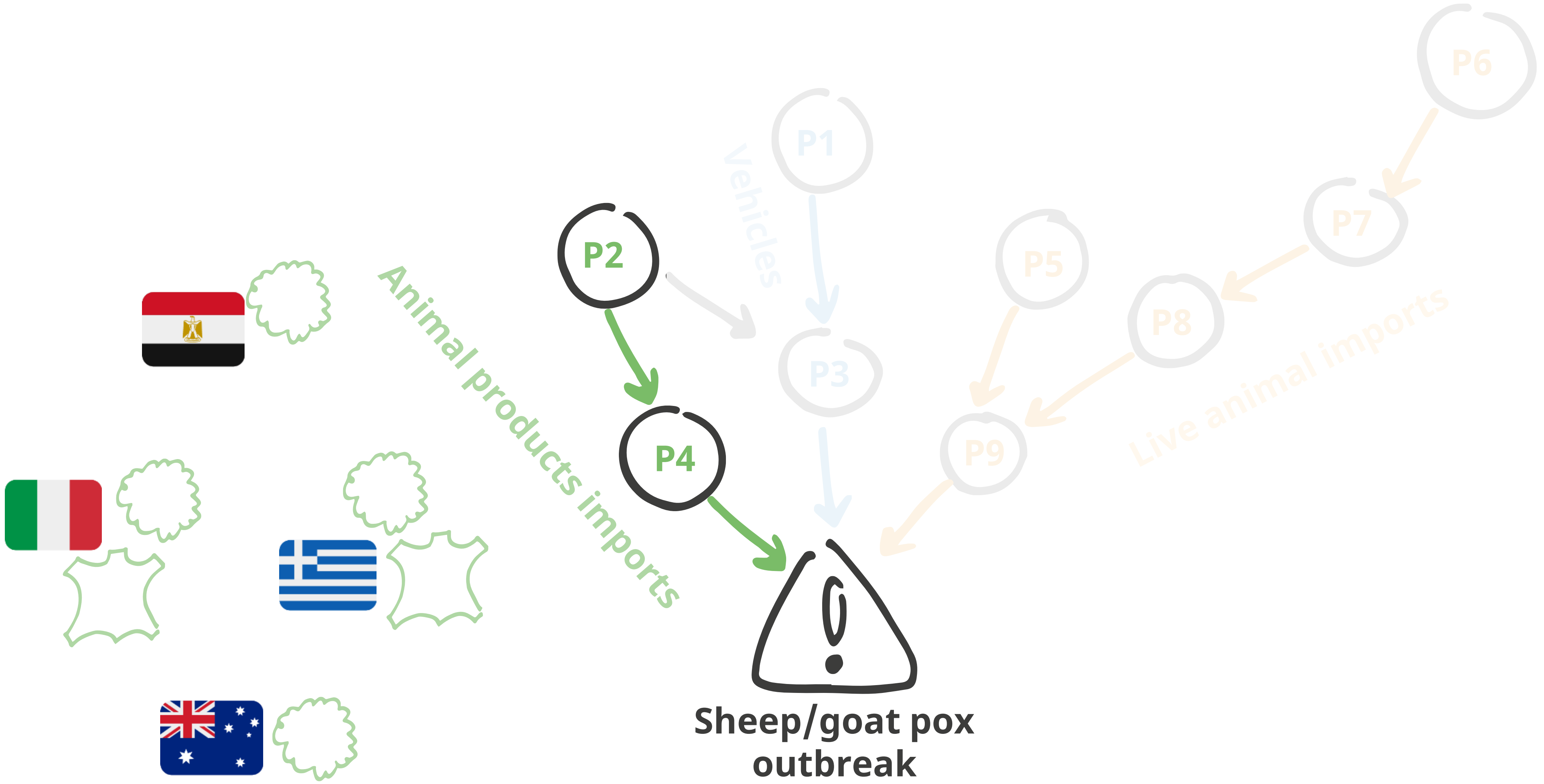


For example



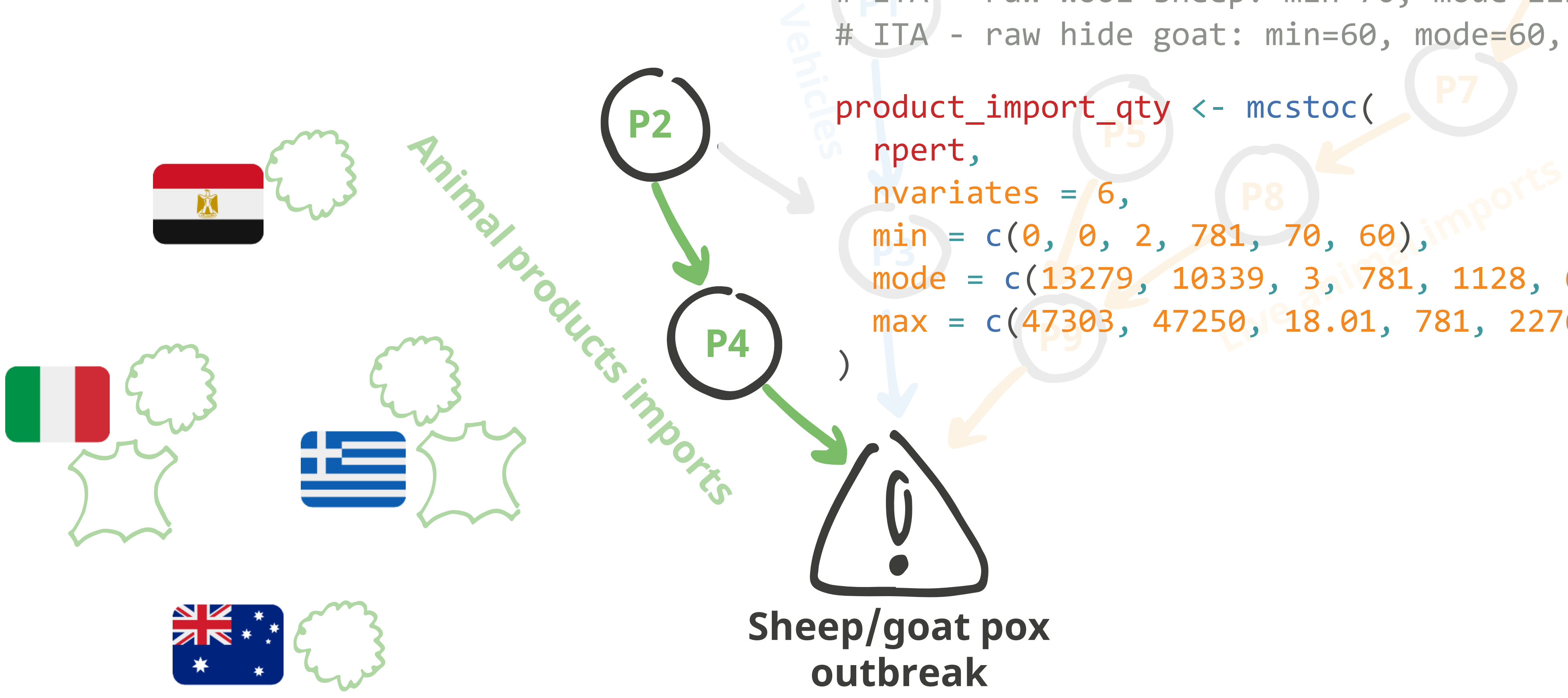
Sheep/goat pox outbreak

For example



P1
P2
P3
P4
P5
P6
P7
P8
P9
Vehicles
Animal products imports
Live animal imports
Sheep/goat pox outbreak

For example

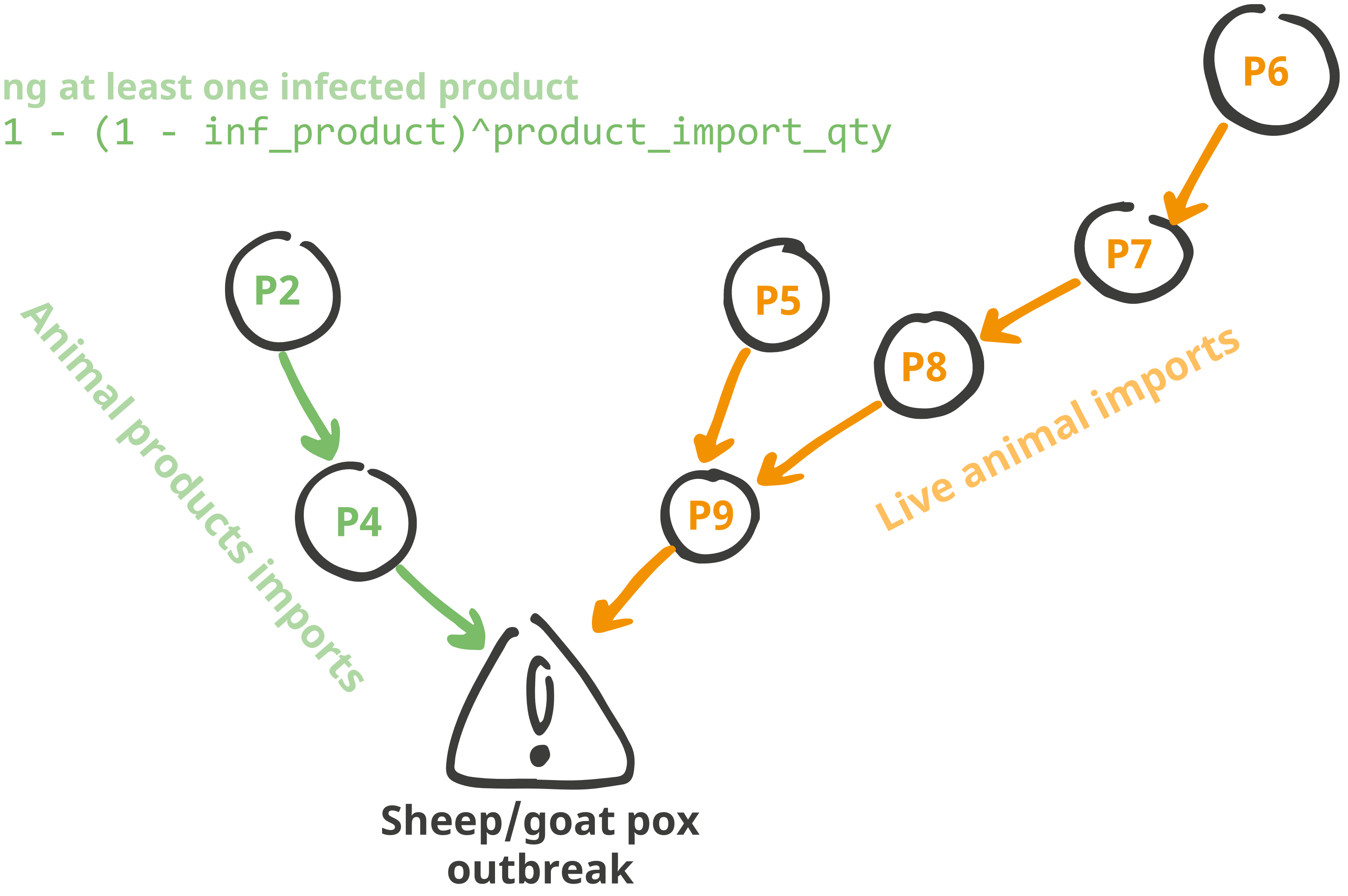


```
# AUS - raw wool sheep: min=0.5, mode=13279, max=47303
# BGR - raw wool sheep: min=0, mode=10339.59, max=47250
# GRC - raw wool sheep: min=1.46, mode=3, max=18.01
# GRC - raw hide goat: min=781.8, mode=781.8, max=781.8
# ITA - raw wool sheep: min=70, mode=1128, max=22768
# ITA - raw hide goat: min=60, mode=60, max=60

product_import_qty <- mcstoc(
  rpert,
  nvariates = 6,
  min = c(0, 0, 2, 781, 70, 60),
  mode = c(13279, 10339, 3, 781, 1128, 60),
  max = c(47303, 47250, 18.01, 781, 22768, 60)
)
```

For example

Probability of introducing at least one infected product

```
inf_product_all <- 1 - (1 - inf_product)^product_import_qty
```



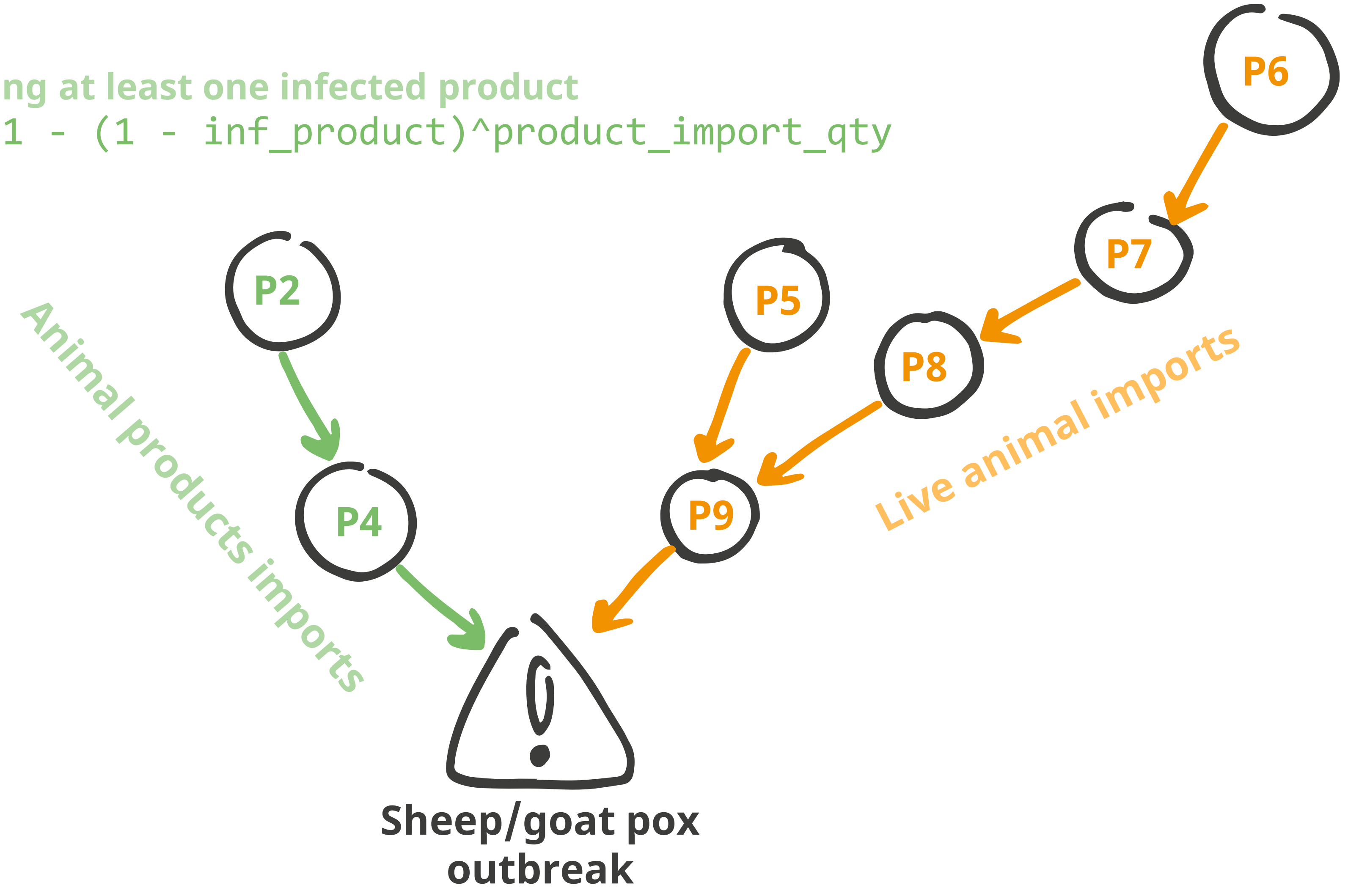
For example

Probability of introducing at least one infected animal

```
inf_animal_all <- 1 - (1 - inf_animal)^animal_import_qty
```

Probability of introducing at least one infected product

```
inf_product_all <- 1 - (1 - inf_product)^product_import_qty
```



For example

Probability of introducing at least one infected product

```
> inf_product_all
  node    mode  nsv nsu nva variate     min  mean median max Nas type outm
1    x numeric 1001   1   6       1 0.00441 0.708  0.996   1   0    V each
2    x numeric 1001   1   6       2 0.00483 0.705  0.996   1   0    V each
3    x numeric 1001   1   6       3 0.00337 0.705  0.995   1   0    V each
4    x numeric 1001   1   6       4 0.00568 0.706  0.997   1   0    V each
5    x numeric 1001   1   6       5 0.00445 0.704  0.996   1   0    V each
6    x numeric 1001   1   6       6 0.00337 0.706  0.997   1   0    V each
```

Probability of introducing at least one infected animal

```
> inf_animal_all
  node    mode  nsv nsu nva variate   min  mean median max Nas type outm
1    x numeric 1001   1   5       1 0.336 0.996      1   1   0    V each
2    x numeric 1001   1   5       2 0.505 0.997      1   1   0    V each
3    x numeric 1001   1   5       3 0.401 0.995      1   1   0    V each
4    x numeric 1001   1   5       4 0.483 0.996      1   1   0    V each
5    x numeric 1001   1   5       5 0.454 0.996      1   1   0    V each
```

Probability of introducing at least one infected import

```
> inf_all <- 1 - (1 - inf_animal_all) * (1 - inf_product_all)
```

For example

Probability of introducing at least one infected product

```
> inf_product_all
  node    mode  nsv nsu nva variate     min  mean median max Nas type outm
1    x numeric 1001   1   6       1 0.00441 0.708  0.996   1   0    V each
2    x numeric 1001   1   6       2 0.00483 0.705  0.996   1   0    V each
3    x numeric 1001   1   6       3 0.00337 0.705  0.995   1   0    V each
4    x numeric 1001   1   6       4 0.00568 0.706  0.997   1   0    V each
5    x numeric 1001   1   6       5 0.00445 0.704  0.996   1   0    V each
6    x numeric 1001   1   6       6 0.00337 0.706  0.997   1   0    V each
```

Probability of introducing at least one infected animal

```
> inf_animal_all
  node    mode  nsv nsu nva variate   min  mean median max Nas type outm
1    x numeric 1001   1   5       1 0.336 0.996      1   1   0    V each
2    x numeric 1001   1   5       2 0.505 0.997      1   1   0    V each
3    x numeric 1001   1   5       3 0.401 0.995      1   1   0    V each
4    x numeric 1001   1   5       4 0.483 0.996      1   1   0    V each
5    x numeric 1001   1   5       5 0.454 0.996      1   1   0    V each
```

Probability of introducing at least one infected import

```
> inf_all <- 1 - (1 - inf_animal_all) * (1 - inf_product_all)
Error in `Ops.mcnode()`:
! Incompatible mcnode dimensions
```

For example

Probability of introducing at least one infected product

```
> inf_product_all
  node     mode  nsv nsu nva variate     min  mean median max Nas type outm
1    x numeric 1001   1   6       1 0.00441 0.708  0.996   1   0    V each
2    x numeric 1001   1   6       2 0.00483 0.705  0.996   1   0    V each
3    x numeric 1001   1   6       3 0.00337 0.705  0.995   1   0    V each
4    x numeric 1001   1   6       4 0.00568 0.706  0.997   1   0    V each
5    x numeric 1001   1   6       5 0.00445 0.704  0.996   1   0    V each
6    x numeric 1001   1   6       6 0.00337 0.706  0.997   1   0    V each
```

Probability of introducing at least one infected animal

```
> inf_animal_all
  node     mode  nsv nsu nva variate   min  mean median max Nas type outm
1    x numeric 1001   1   5       1 0.336 0.996      1   1   0    V each
2    x numeric 1001   1   5       2 0.505 0.997      1   1   0    V each
3    x numeric 1001   1   5       3 0.401 0.995      1   1   0    V each
4    x numeric 1001   1   5       4 0.483 0.996      1   1   0    V each
5    x numeric 1001   1   5       5 0.454 0.996      1   1   0    V each
```

$$
\underset{\textbf{node\_x}}{\begin{array}{c|ccccc}
 & 1 & 2 & 3 & \dots & u \\ \hline
1 & 0.01 & 0.02 & 0.01 & \dots & 0.02 \\
2 & 0.02 & 0.02 & 0.01 & \dots & 0.01 \\
3 & 0.39 & 0.37 & 0.41 & \dots & 0.4 \\
4 & 0.42 & 0.38 & 0.4 & \dots & 0.37 \\
\dots & 0.25 & 0.56 & 0.32 & \dots & 0.69 \\
i & 0.61 & 0.42 & 0.28 & \dots & \boxed{0.33}
\end{array}}
\cdot
\underset{\textbf{node\_y}}{\begin{array}{c|ccccc}
 & 1 & 2 & 3 & \dots & u \\ \hline
1 & 0.005 & 0.001 & 0.002 & \dots & 0.003 \\
2 & 0.001 & 0.002 & 0.004 & \dots & 0.002 \\
3 & 0.55 & 0.53 & 0.58 & \dots & 0.54 \\
4 & 0.17 & 0.2 & 0.18 & \dots & 0.15 \\
\dots & 0.02 & 0.03 & 0.02 & \dots & 0.02 \\
i & 0.03 & 0.02 & 0.02 & \dots & \boxed{0.02}
\end{array}}
=
\underset{\textbf{node\_xy}}{\begin{array}{c|ccccc}
 & 1 & 2 & 3 & \dots & u \\ \hline
1 & 5E\text{-}05 & 2E\text{-}05 & 2E\text{-}05 & \dots & 6E\text{-}05 \\
2 & 2E\text{-}05 & 4E\text{-}05 & 4E\text{-}05 & \dots & 2E\text{-}05 \\
3 & 0.21 & 0.2 & 0.24 & \dots & 0.22 \\
4 & 0.07 & 0.08 & 0.07 & \dots & 0.06 \\
\dots & 0.01 & 0.02 & 0.01 & \dots & 0.01 \\
i & 0.02 & 0.01 & 0.01 & \dots & \boxed{0.01}
\end{array}}
$$

$$node\_x_{iu} \cdot node\_y_{iu} = node\_xy_{iu}$$

For example

Probability of introducing at least one infected product

```
> inf_product_all
 node     mode  nsv nsu nva variate     min  mean median max Nas type outm
1    x numeric 1001   1   6       1 0.00441 0.708  0.996   1   0    V each
2    x numeric 1001   1   6       2 0.00483 0.705  0.996   1   0    V each
3    x numeric 1001   1   6       3 0.00337 0.705  0.995   1   0    V each
4    x numeric 1001   1   6       4 0.00568 0.706  0.997   1   0    V each
5    x numeric 1001   1   6       5 0.00445 0.704  0.996   1   0    V each
6    x numeric 1001   1   6       6 0.00337 0.706  0.997   1   0    V each
```

Probability of introducing at least one infected animal

```
> inf_animal_all
 node     mode  nsv nsu nva variate   min  mean median max Nas type outm
1    x numeric 1001   1   5       1 0.336 0.996      1   1   0    V each
2    x numeric 1001   1   5       2 0.505 0.997      1   1   0    V each
3    x numeric 1001   1   5       3 0.401 0.995      1   1   0    V each
4    x numeric 1001   1   5       4 0.483 0.996      1   1   0    V each
5    x numeric 1001   1   5       5 0.454 0.996      1   1   0    V each
```

node_x

	1	2	3	…	u
1	0.01	0.02	0.01	…	0.02
2	0.02	0.02	0.01	…	0.01
3	0.39	0.37	0.41	…	0.4
4	0.42	0.38	0.4	…	0.37
…	0.25	0.56	0.32	…	0.69
i	0.61	0.42	0.28	…	0.33

·

node_y

	1	2	3	…	u
1	0.005	0.001	0.002	…	0.003
2	0.001	0.002	0.004	…	0.002
3	0.55	0.53	0.58	…	0.54
4	0.17	0.2	0.18	…	0.15
…	0.02	0.03	0.02	…	0.02
i	0.03	0.02	0.02	…	0.02

=

node_xy

	1	2	3	…	u
1	5E-05	2E-05	2E-05	…	6E-05
2	2E-05	4E-05	4E-05	…	2E-05
3	0.21	0.2	0.24	…	0.22
4	0.07	0.08	0.07	…	0.06
…	0.01	0.02	0.01	…	0.01
i	0.02	0.01	0.01	…	0.01

$$node\_x_{iu} \cdot node\_y_{iu} = node\_xy_{iu}$$

For example

Probability of introducing at least one infected product

```
  country_code species       commodity    min   mean median max Nas type outm
1          AUS   sheep raw_wool_sheep 0.00441 0.708  0.996   1   0    V each
2          BGR   sheep raw_wool_sheep 0.00483 0.705  0.996   1   0    V each
3          GRC   sheep raw_wool_sheep 0.00337 0.705  0.995   1   0    V each
4          GRC    goat   raw_hide_goat 0.00568 0.706  0.997   1   0    V each
5          ITA   sheep raw_wool_sheep 0.00445 0.704  0.996   1   0    V each
6          ITA    goat   raw_hide_goat 0.00337 0.706  0.997   1   0    V each
```

Probability of introducing at least one infected animal

```
  country_code species  commodity   min  mean median max Nas type outm
1          GRC   sheep live_sheep 0.336 0.996      1   1   0    V each
2          GRC    goat  live_goat 0.505 0.997      1   1   0    V each
3          ITA   sheep live_sheep 0.401 0.995      1   1   0    V each
4          ITA    goat  live_goat 0.483 0.996      1   1   0    V each
5          PRT   sheep live_sheep 0.454 0.996      1   1   0    V each
```

$$
\begin{array}{c}\textbf{node\_x}\\ \begin{array}{c|ccccc} & 1 & 2 & 3 & \dots & u \\ \hline 1 & 0.01 & 0.02 & 0.01 & \dots & 0.02 \\ 2 & 0.02 & 0.02 & 0.01 & \dots & 0.01 \\ 3 & 0.39 & 0.37 & 0.41 & \dots & 0.4 \\ 4 & 0.42 & 0.38 & 0.4 & \dots & 0.37 \\ \dots & 0.25 & 0.56 & 0.32 & \dots & 0.69 \\ i & 0.61 & 0.42 & 0.28 & \dots & \boxed{0.33} \end{array}\end{array}
\cdot
\begin{array}{c}\textbf{node\_y}\\ \begin{array}{c|ccccc} & 1 & 2 & 3 & \dots & u \\ \hline 1 & 0.005 & 0.001 & 0.002 & \dots & 0.003 \\ 2 & 0.001 & 0.002 & 0.004 & \dots & 0.002 \\ 3 & 0.55 & 0.53 & 0.58 & \dots & 0.54 \\ 4 & 0.17 & 0.2 & 0.18 & \dots & 0.15 \\ \dots & 0.02 & 0.03 & 0.02 & \dots & 0.02 \\ i & 0.03 & 0.02 & 0.02 & \dots & \boxed{0.02} \end{array}\end{array}
=
\begin{array}{c}\textbf{node\_xy}\\ \begin{array}{c|ccccc} & 1 & 2 & 3 & \dots & u \\ \hline 1 & 5E-05 & 2E-05 & 2E-05 & \dots & 6E-05 \\ 2 & 2E-05 & 4E-05 & 4E-05 & \dots & 2E-05 \\ 3 & 0.21 & 0.2 & 0.24 & \dots & 0.22 \\ 4 & 0.07 & 0.08 & 0.07 & \dots & 0.06 \\ \dots & 0.01 & 0.02 & 0.01 & \dots & 0.01 \\ i & 0.02 & 0.01 & 0.01 & \dots & \boxed{0.01} \end{array}\end{array}
$$

$$node\_x_{iu} \cdot node\_y_{iu} = node\_xy_{iu}$$

For example

Probability of introducing at least one infected product

```
> inf_product_all
  node     mode  nsv nsu nva variate     min  mean median max Nas type outm
1    x numeric 1001   1   6       1 0.00441 0.708  0.996   1   0    V each
2    x numeric 1001   1   6       2 0.00483 0.705  0.996   1   0    V each
3    x numeric 1001   1   6       3 0.00337 0.705  0.995   1   0    V each
4    x numeric 1001   1   6       4 0.00568 0.706  0.997   1   0    V each
5    x numeric 1001   1   6       5 0.00445 0.704  0.996   1   0    V each
6    x numeric 1001   1   6       6 0.00337 0.706  0.997   1   0    V each
```

Probability of introducing at least one infected animal

```
> inf_animal_all
  node     mode  nsv nsu nva variate   min  mean median max Nas type outm
1    x numeric 1001   1   5       1 0.336 0.996      1   1   0    V each
2    x numeric 1001   1   5       2 0.505 0.997      1   1   0    V each
3    x numeric 1001   1   5       3 0.401 0.995      1   1   0    V each
4    x numeric 1001   1   5       4 0.483 0.996      1   1   0    V each
5    x numeric 1001   1   5       5 0.454 0.996      1   1   0    V each
```

$$
\underset{\textbf{node\_x}}{\begin{array}{c|ccccc}
 & 1 & 2 & 3 & \dots & u \\ \hline
1 & 0.01 & 0.02 & 0.01 & \dots & 0.02 \\
2 & 0.02 & 0.02 & 0.01 & \dots & 0.01 \\
3 & 0.39 & 0.37 & 0.41 & \dots & 0.4 \\
4 & 0.42 & 0.38 & 0.4 & \dots & 0.37 \\
\dots & 0.25 & 0.56 & 0.32 & \dots & 0.69 \\
i & 0.61 & 0.42 & 0.28 & \dots & \boxed{0.33}
\end{array}}
\cdot
\underset{\textbf{node\_y}}{\begin{array}{c|ccccc}
 & 1 & 2 & 3 & \dots & u \\ \hline
1 & 0.005 & 0.001 & 0.002 & \dots & 0.003 \\
2 & 0.001 & 0.002 & 0.004 & \dots & 0.002 \\
3 & 0.55 & 0.53 & 0.58 & \dots & 0.54 \\
4 & 0.17 & 0.2 & 0.18 & \dots & 0.15 \\
\dots & 0.02 & 0.03 & 0.02 & \dots & 0.02 \\
i & 0.03 & 0.02 & 0.02 & \dots & \boxed{0.02}
\end{array}}
=
\underset{\textbf{node\_xy}}{\begin{array}{c|ccccc}
 & 1 & 2 & 3 & \dots & u \\ \hline
1 & 5E\text{-}05 & 2E\text{-}05 & 2E\text{-}05 & \dots & 6E\text{-}05 \\
2 & 2E\text{-}05 & 4E\text{-}05 & 4E\text{-}05 & \dots & 2E\text{-}05 \\
3 & 0.21 & 0.2 & 0.24 & \dots & 0.22 \\
4 & 0.07 & 0.08 & 0.07 & \dots & 0.06 \\
\dots & 0.01 & 0.02 & 0.01 & \dots & 0.01 \\
i & 0.02 & 0.01 & 0.01 & \dots & \boxed{0.01}
\end{array}}
$$

$$node\_x_{iu} \cdot node\_y_{iu} = node\_xy_{iu}$$

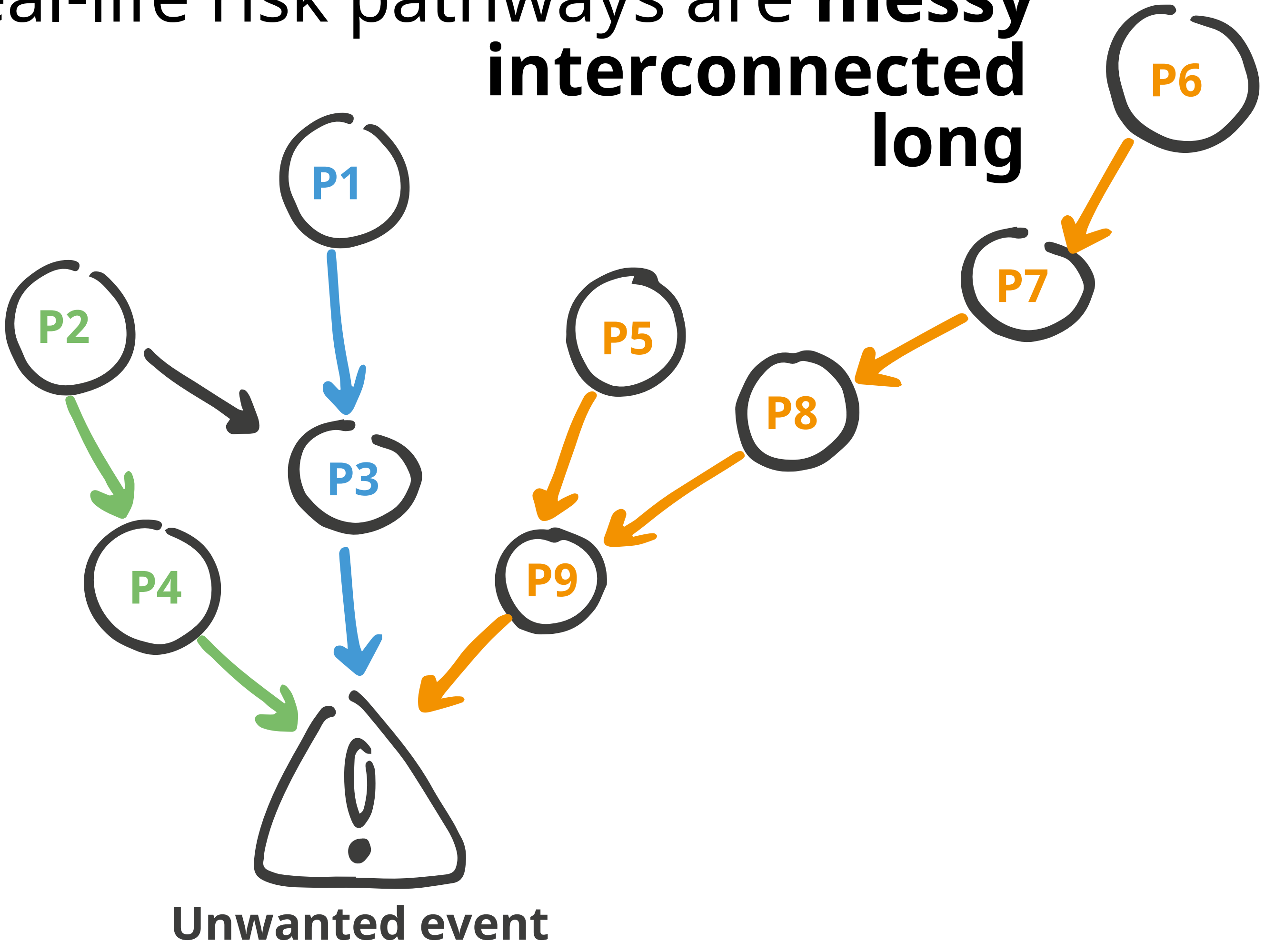
With `mc2d`

Quantifying risk through stochastic probability steps is **neat**

But real-life risk pathways are **messy**

interconnected

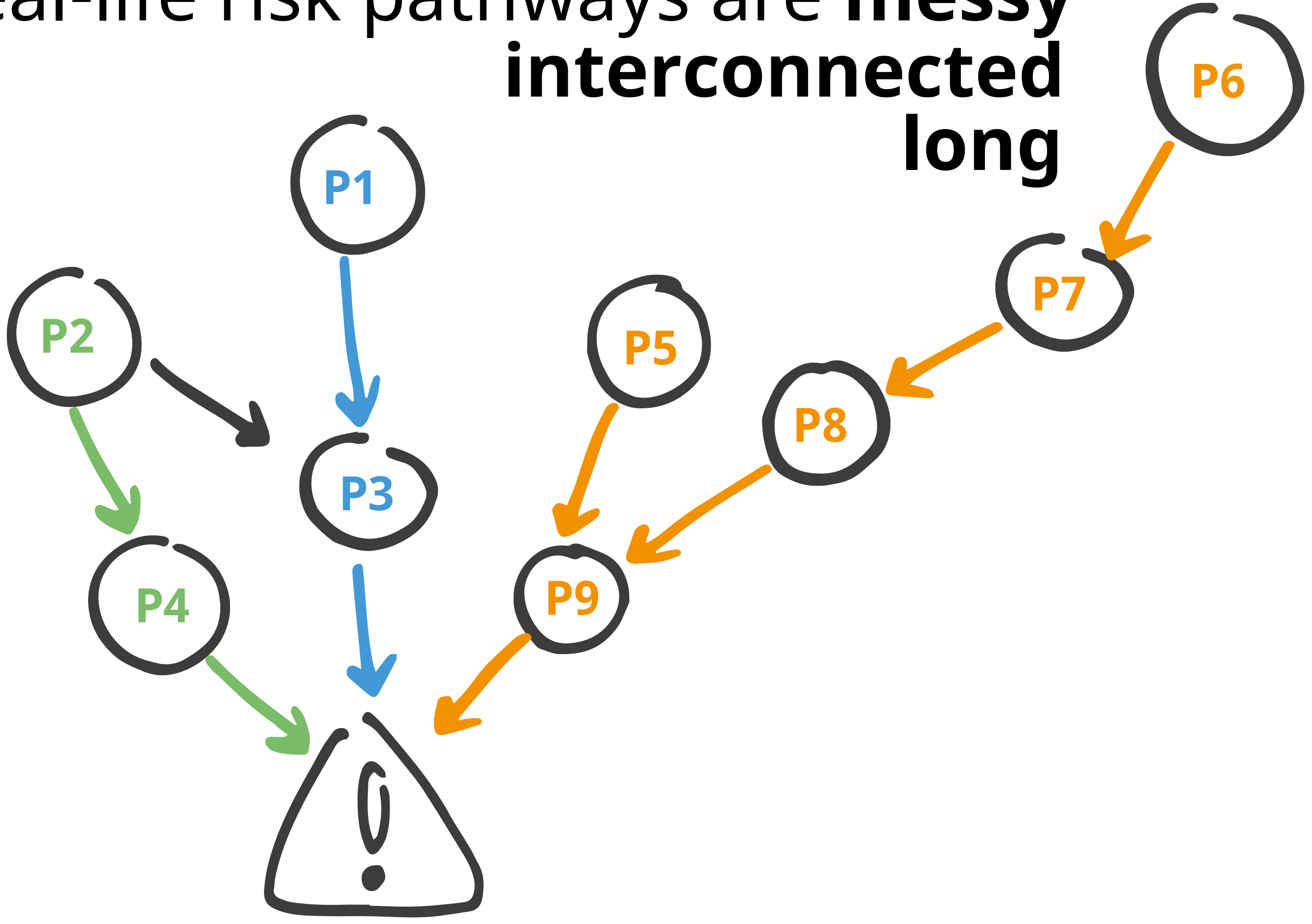
long



With `mc2d`
Quantifying risk through stochastic probability steps is **neat**

But real-life risk pathways are **messy**
interconnected
long

Multi-variate
(many rows)



Unwanted event

With `mc2d`

Quantifying risk through stochastic probability steps is **neat**

But real-life risk pathways are **messy**

interconnected

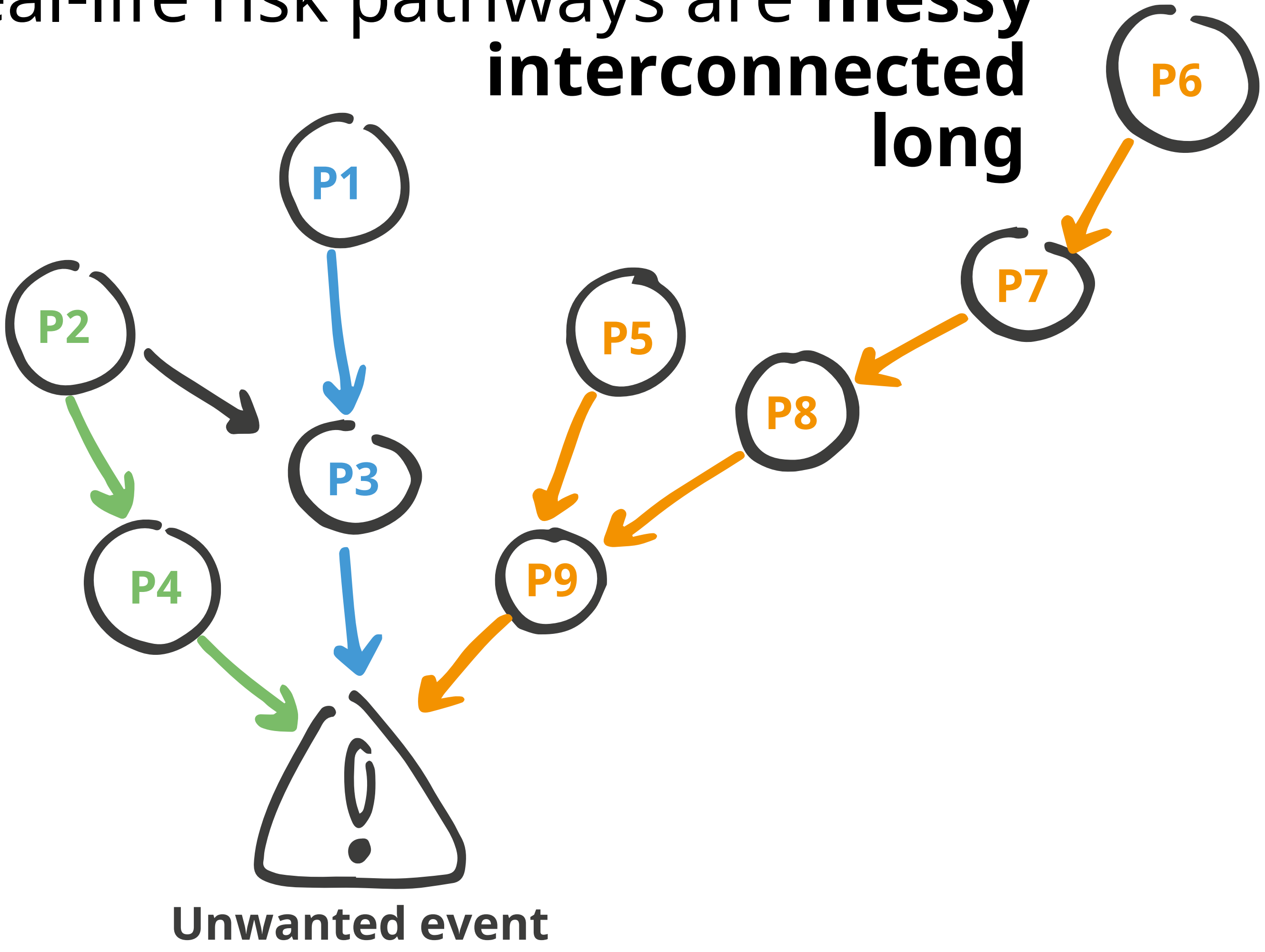
long

Multi-variate

(many rows)

Not-aligned pathway variates

(different rows)



With `mc2d`

Quantifying risk through stochastic probability steps is **neat**

But real-life risk pathways are **messy**

interconnected

long

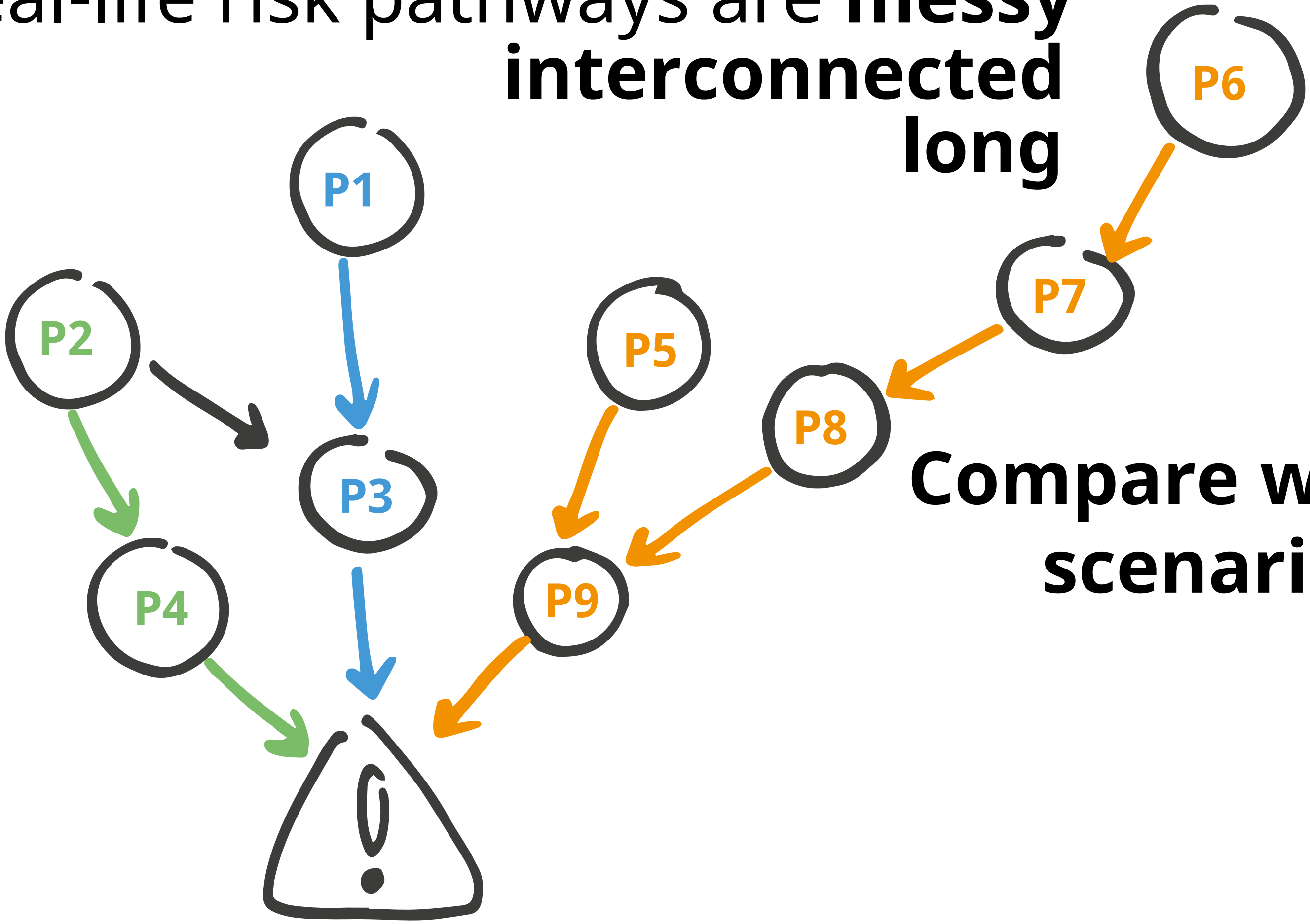
Multi-variate

(many rows)

Not-aligned pathway variates

(different rows)

Compare what-if scenarios



Unwanted event

With `mc2d`

Quantifying risk through stochastic probability steps is **neat**

But real-life risk pathways are **messy**

interconnected

long

Multi-variate

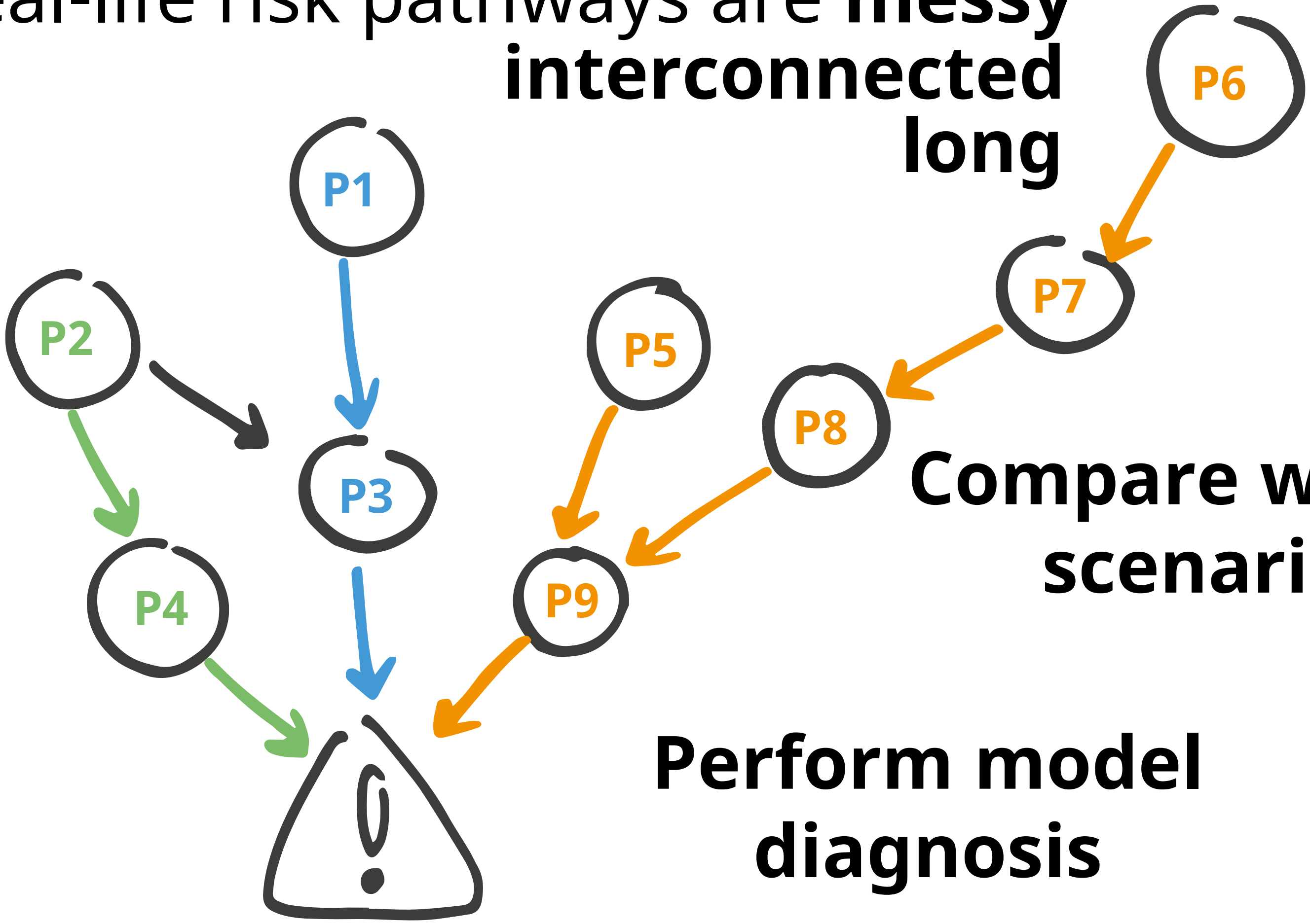
(many rows)

Not-aligned pathway variates

(different rows)

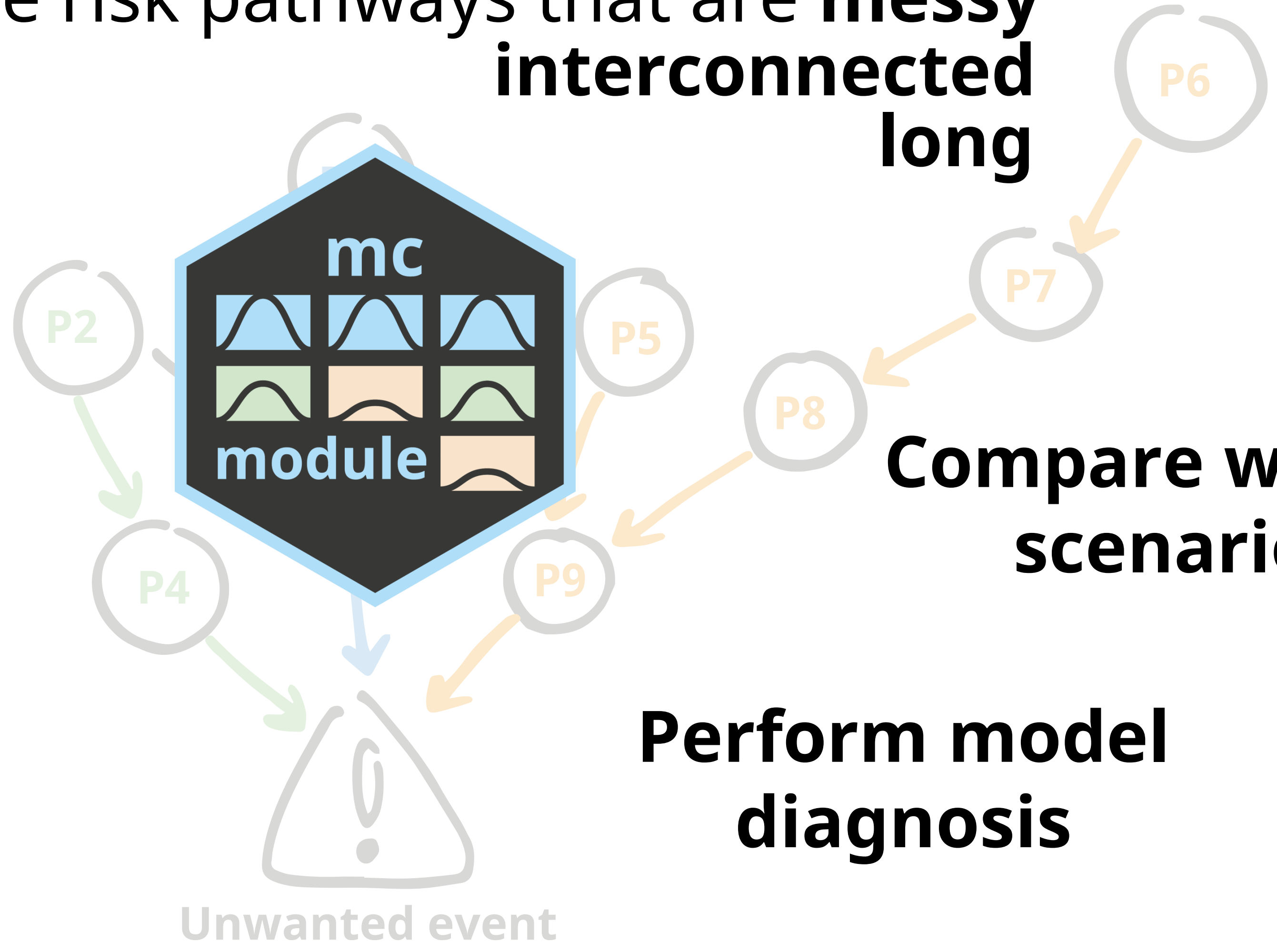
Compare what-if scenarios

Perform model diagnosis



Unwanted event

With `mcmodule`
Quantifying risk through stochastic probability steps
in real-life risk pathways that are **messy**
interconnected
long



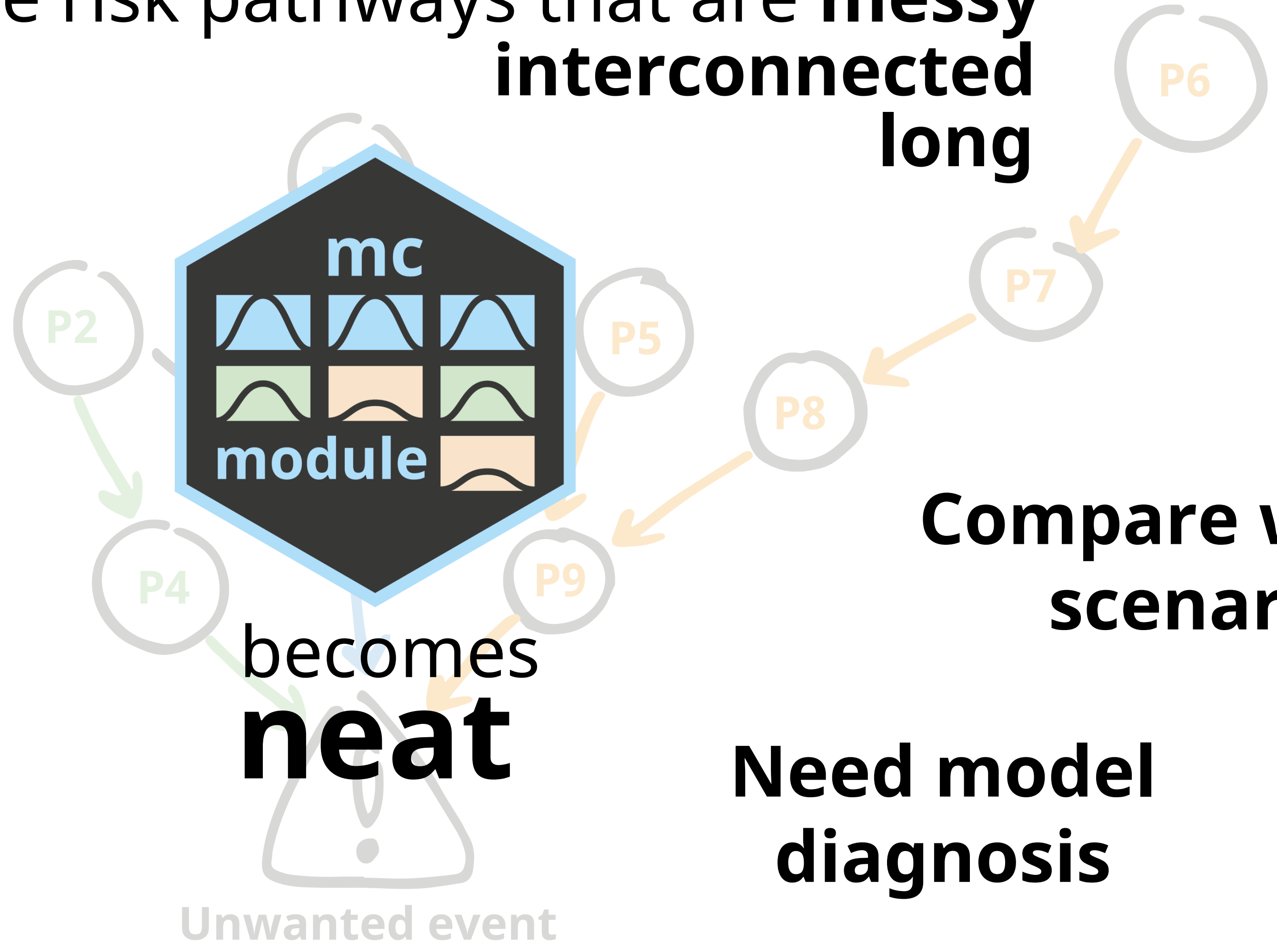
Multi-variate
(many rows)

Not-aligned pathway variates
(different rows)

Compare what-if scenarios

Perform model diagnosis

With `mcmodule`
Quantifying risk through stochastic probability steps
in real-life risk pathways that are **messy**
interconnected
long



Multi-variate
(many rows)

Not-aligned pathway variates
(different rows)

becomes
neat

Compare what-if scenarios

Need model diagnosis

mcmodule::

Core components

mcmodule:: core components

Component	Purpose
Data	Table(s) with one row per variate and columns for input distribution parameters
Data keys	Unique identifiers of rows/variates
mctable	Table with node specifications (distributions, transformations)
Expressions	Mathematical risk calculations

This creates **metadata-rich modules** where:

- Nodes are created automatically from data
- Variates are identified by keys (country, species, scenario…)
- Operations align automatically
- Pathways combine safely
- Scenarios propagate through the model

mcmodule:: case study

Question

What is the probability of introducing sheep/goat pox to Spain through imports?

Pathways

- Live animals (sheep and goat)
- Animal products with contamination risk (raw wool, hides)

Data

- Imports by commodity and country (COMTRADE + Spanish Cámara de Comercio)
- Disease status by country (WAHIS)
- Animal holdings and heads census by country (Eurostat + FAO)

mcmodule:: set-up

```r
# Load packages
library(mcmodule)
library(dplyr)
library(tidyr)
library(ggplot2)

ndvar(1001) # Uncertainty simulations

# Load data
imports_by_country_commodity <- read.csv("data/imports_by_country_commodity.csv", stringsAsFactors = TRUE)
disease_by_country <- read.csv("data/disease_by_country.csv", stringsAsFactors = TRUE)
census_by_country <- read.csv("data/census_by_country.csv", stringsAsFactors = TRUE)
```

mcmodule:: set-up

```r
# Load packages
library(mcmodule)
library(dplyr)
library(tidyr)
library(ggplot2)

ndvar(1001) # Uncertainty simulations

# Load data
imports_by_country_commodity <- read.csv("data/imports_by_country_commodity.csv", stringsAsFactors = TRUE)
disease_by_country <- read.csv("data/disease_by_country.csv", stringsAsFactors = TRUE)
census_by_country <- read.csv("data/census_by_country.csv", stringsAsFactors = TRUE)
```

If input parameters are provided at **different resolutions or aggregation levels**, it's recommended to load them in different tables. !

mcmodule:: set-up

```r
# Load packages
library(mcmodule)
library(dplyr)
library(tidyr)
library(ggplot2)

ndvar(1001) # Uncertainty simulations

# Load data
imports_by_country_commodity <- read.csv("data/imports_by_country_commodity.csv", stringsAsFactors = TRUE)
disease_by_country <- read.csv("data/disease_by_country.csv", stringsAsFactors = TRUE)
census_by_country <- read.csv("data/census_by_country.csv", stringsAsFactors = TRUE)

# Combine data
imports_data <- imports_by_country_commodity %>%
  left_join(disease_by_country, by = "country_code") %>%
  left_join(census_by_country, by = c("country_code", "species"))
```

If input parameters are provided at **different resolutions or aggregation levels**, it's recommended to load them in different tables. !

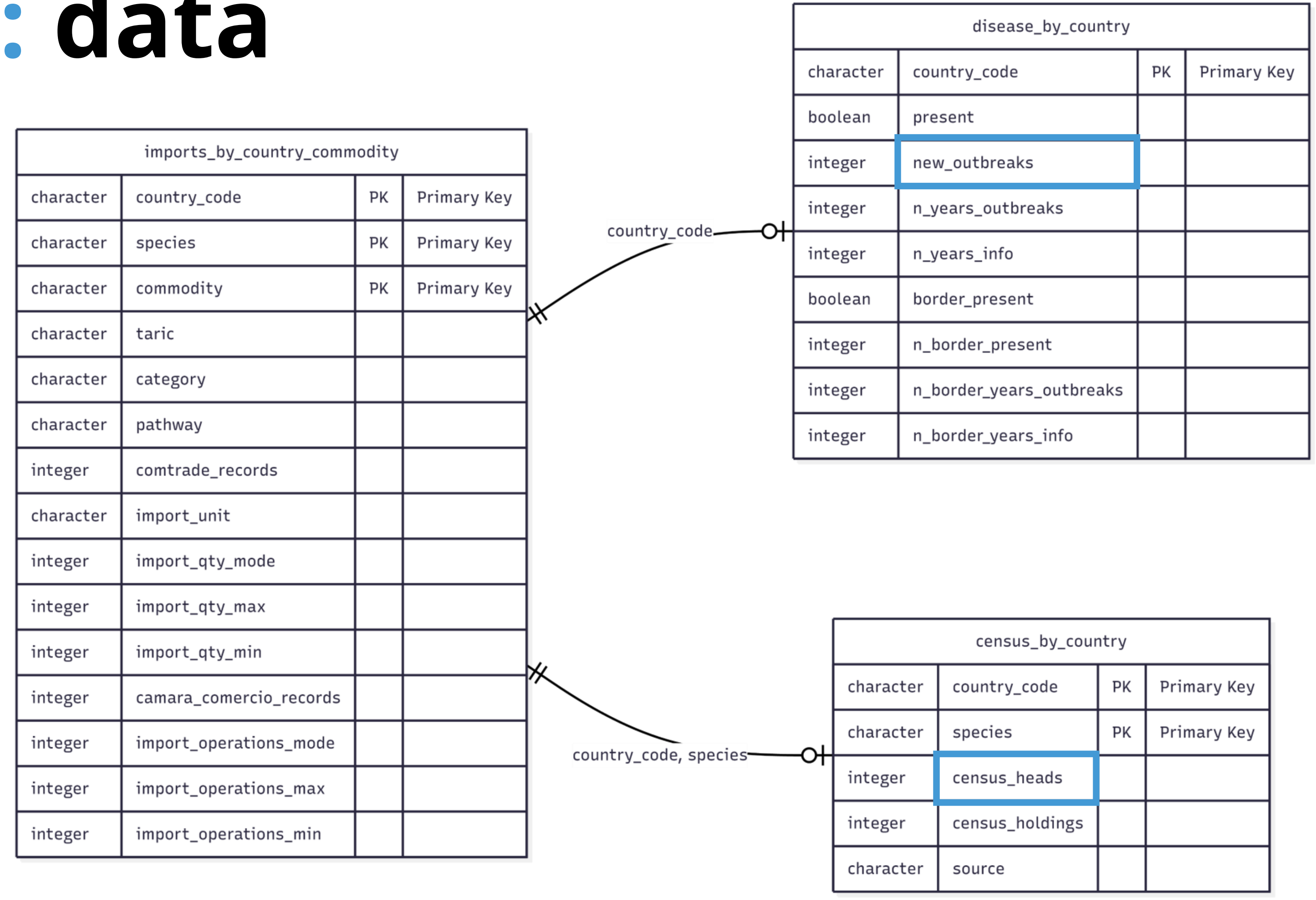
mcmodule:: **data**

Trace input **data resolution**

Good practice!

For example

- Parameter `census_heads` was provided at country and species level
- However `new_outbreaks` was provided at country level, no distinction between species



imports_by_country_commodity			
character	country_code	PK	Primary Key
character	species	PK	Primary Key
character	commodity	PK	Primary Key
character	taric		
character	category		
character	pathway		
integer	comtrade_records		
character	import_unit		
integer	import_qty_mode		
integer	import_qty_max		
integer	import_qty_min		
integer	camara_comercio_records		
integer	import_operations_mode		
integer	import_operations_max		
integer	import_operations_min		

disease_by_country			
character	country_code	PK	Primary Key
boolean	present		
integer	new_outbreaks		
integer	n_years_outbreaks		
integer	n_years_info		
boolean	border_present		
integer	n_border_present		
integer	n_border_years_outbreaks		
integer	n_border_years_info		

census_by_country			
character	country_code	PK	Primary Key
character	species	PK	Primary Key
integer	census_heads		
integer	census_holdings		
character	source		

mcmodule:: data

Trace input **data resolution**

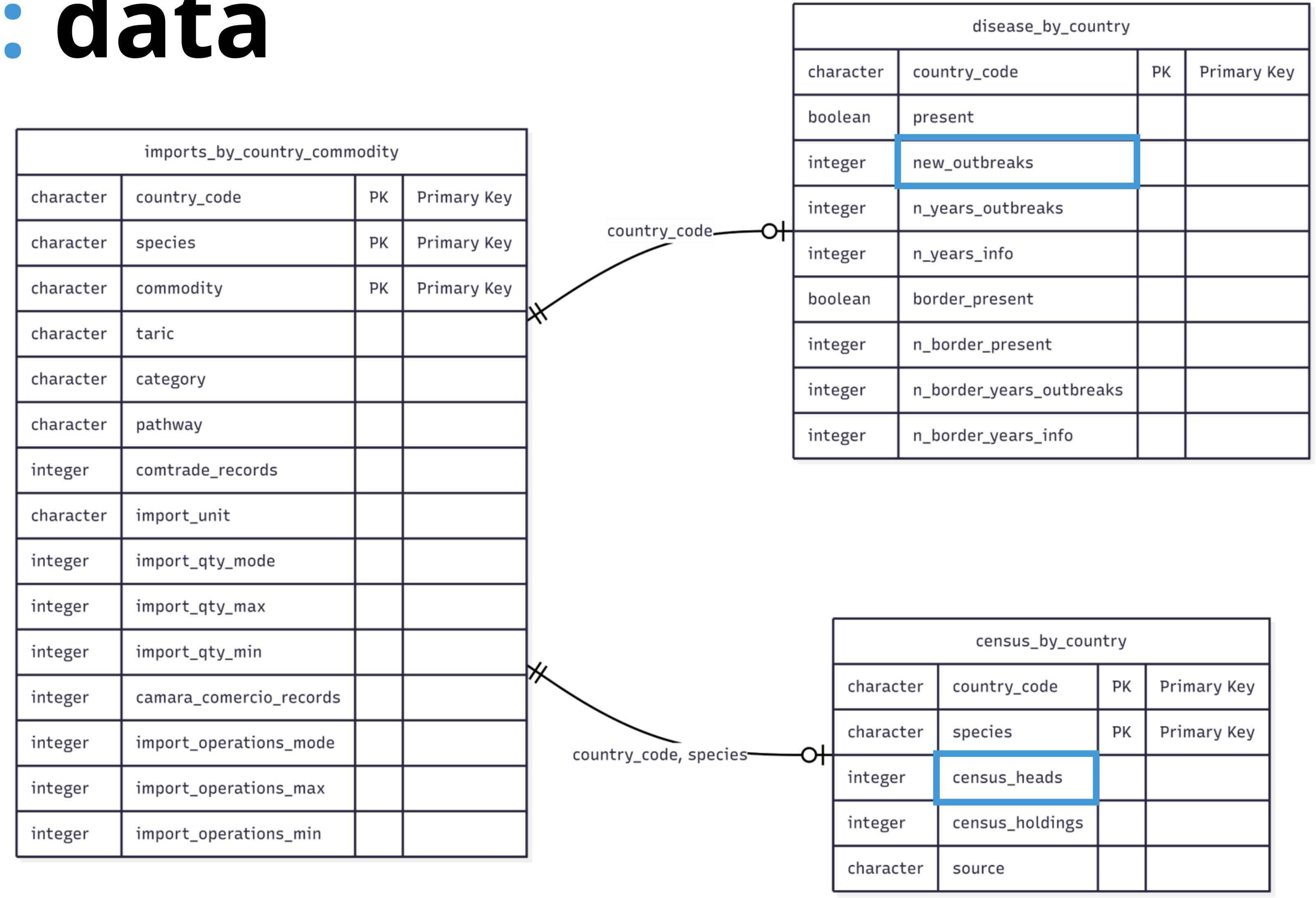
Good practice!

For example

- Parameter `census_heads` was provided at country and species level
- However `new_outbreaks` was provided at country level, no distinction between species

Saves memory use

Good practice!

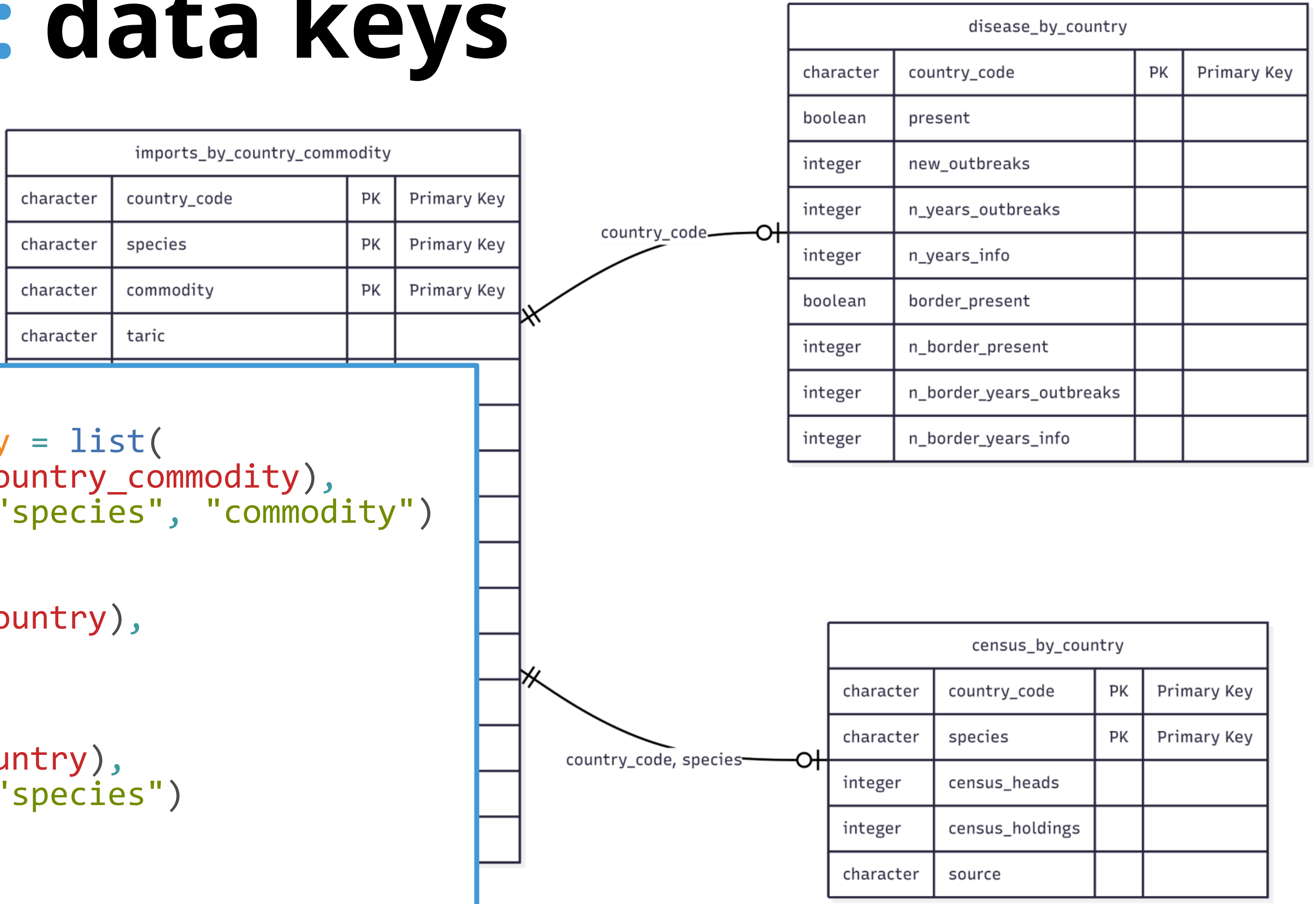


imports_by_country_commodity			
character	country_code	PK	Primary Key
character	species	PK	Primary Key
character	commodity	PK	Primary Key
character	taric		
character	category		
character	pathway		
integer	comtrade_records		
character	import_unit		
integer	import_qty_mode		
integer	import_qty_max		
integer	import_qty_min		
integer	camara_comercio_records		
integer	import_operations_mode		
integer	import_operations_max		
integer	import_operations_min		

disease_by_country			
character	country_code	PK	Primary Key
boolean	present		
integer	new_outbreaks		
integer	n_years_outbreaks		
integer	n_years_info		
boolean	border_present		
integer	n_border_present		
integer	n_border_years_outbreaks		
integer	n_border_years_info		

census_by_country			
character	country_code	PK	Primary Key
character	species	PK	Primary Key
integer	census_heads		
integer	census_holdings		
character	source		

mcmodule:: data keys



imports_by_country_commodity			
character	country_code	PK	Primary Key
character	species	PK	Primary Key
character	commodity	PK	Primary Key
character	taric		

disease_by_country			
character	country_code	PK	Primary Key
boolean	present		
integer	new_outbreaks		
integer	n_years_outbreaks		
integer	n_years_info		
boolean	border_present		
integer	n_border_present		
integer	n_border_years_outbreaks		
integer	n_border_years_info		

census_by_country			
character	country_code	PK	Primary Key
character	species	PK	Primary Key
integer	census_heads		
integer	census_holdings		
character	source		

Declare data keys

```r
sgp_data_keys <- list(
  imports_by_country_commodity = list(
    cols = names(imports_by_country_commodity),
    keys = c("country_code", "species", "commodity")
  ),
  disease_by_country = list(
    cols = names(disease_by_country),
    keys = "country_code"
  ),
  census_by_country = list(
    cols = names(census_by_country),
    keys = c("country_code", "species")
  )
)

set_data_keys(sgp_data_keys) # Set in the envir
```

mcmodule:: mctable

mcnode	description	mc_func	from_variable	transformation	sensi_baseline	sensi_variation
import_qty	Annual imports quantity (units for live animals, kilograms for animal products)	rpert	NA	NA	min = 2000, max = 8000	value * 1.5
import_operations	Annual imports operations	rpert	NA	NA	min = 30, max = 70	value * 1.5
n_years_outbreaks	Number of years with new outbreaks	NA	NA	NA	2	value * 1.5
n_years_info	Number of years with reported health status	NA	NA	NA	10	value * 1.5
census_heads	Number of animal heads in the country	NA	NA	NA	10000	value * 1.5
census_holdings	Number of animal holdings in the country	NA	NA	NA	500	value * 1.5
present	Disease was present in the country the last year reported	NA	NA	NA	0	pmin(1, pmax(0, value * 1.5))
border_present	TRUE if disease is present in any neighboring country	NA	NA	NA	1	pmin(1, pmax(0, value * 1.5))
n_border_present	Number of neighboring countries with disease present	NA	NA	NA	1	value * 1.5
n_border_years_info	Total years of health status information for neighboring countries	NA	NA	NA	10	value * 1.5
n_border_years_outbreaks	Total years of outbreaks in neighboring countries	NA	NA	NA	2	value * 1.5
live_animals	Live animals imports (TRUE if live animals, FALSE if animal products)	NA	pathway	value=='live_animals'	NA	pmin(1, pmax(0, value * 1.5))
p_herds_surveillance	Proportion of herds visited for surveillance at origin	runif	NA	NA	min = 0.25, max = 0.75	pmin(1, pmax(0, value * 1.5))
p_animals_inspected	Proportion of animals inspected per herd at origin	runif	NA	NA	min = 0.2, max = 0.4	pmin(1, pmax(0, value * 1.5))
clinical_se_origin	Sensitivity of clinical inspection at origin	runif	NA	NA	min = 0.5, max = 0.7	pmin(1, pmax(0, value * 1.5))
clinical_se_dest	Sensitivity of clinical inspection at destination	runif	NA	NA	min = 0.55, max = 0.75	pmin(1, pmax(0, value * 1.5))

mcmodule:: mctable

mcnode	description	mc_func	from_variable	transformation	sensi_baseline	sensi_variation
import_qty	Annual imports quantity (units for live animals, kilograms for animal products)	rpert	NA	NA	min = 2000, max = 8000	value * 1.5
import_operations	Annual imports operations	rpert	NA	NA	min = 30, max = 70	value * 1.5
n_years_outbreaks	Number of years with new outbreaks	NA	NA	NA	2	value * 1.5
n_years_info	Number of years with reported health status	NA	NA	NA	10	value * 1.5
census_heads	Number of animal heads in the country	NA	NA	NA	10000	value * 1.5
census_holdings	Number of animal holdings in the country	NA	N			
present	Disease was present in the country the last year reported	NA	N			
border_present	TRUE if disease is present in any neighboring country	NA	N			
n_border_present	Number of neighboring countries with disease present	NA	N			
n_border_years_info	Total years of health status information for neighboring countries	NA	N			
n_border_years_outbreaks	Total years of outbreaks in neighboring countries	NA	NA	NA	2	value * 1.5
live_animals	Live animals imports (TRUE if live animals, FALSE if animal products)	NA	pathway	value=='live_animals'	NA	pmin(1, pmax(0, value * 1.5))
p_herds_surveillance	Proportion of herds visited for surveillance at origin	runif	NA	NA	min = 0.25, max = 0.75	pmin(1, pmax(0, value * 1.5))
p_animals_inspected	Proportion of animals inspected per herd at origin	runif	NA	NA	min = 0.2, max = 0.4	pmin(1, pmax(0, value * 1.5))
clinical_se_origin	Sensitivity of clinical inspection at origin	runif	NA	NA	min = 0.5, max = 0.7	pmin(1, pmax(0, value * 1.5))
clinical_se_dest	Sensitivity of clinical inspection at destination	runif	NA	NA	min = 0.55, max = 0.75	pmin(1, pmax(0, value * 1.5))

Declare mctable

```r
sgp_mctable <- read.csv(
  "data/sgp_mctable.csv",
  stringsAsFactors = TRUE
)

set_mctable(sgp_mctable) # Set in the envir
```

mcmodule:: expressions

```r
origin_inf_exp <- quote({
  # Proportion of years with documented outbreaks
  p_years_outbreaks <- n_years_outbreaks / n_years_info

  # Surveillance sensitivity
  surveillance_se <- mcstoc(runif, min = 0.5, max = 0.9)

  # Probability origin is infected and not detected
  p_not_detected_origin <- p_years_outbreaks * (1 - surveillance_se)
})
```

mcmodule:: expressions

Input mcnodes will be created from data and mctable specifications

```
origin_inf_exp <- quote({
  # Proportion of years with documented outbreaks
  p_years_outbreaks <- n_years_outbreaks / n_years_info

  # Surveillance sensitivity
  surveillance_se <- mcstoc(runif, min = 0.5, max = 0.9)

  # Probability origin is infected and not detected
  p_not_detected_origin <- p_years_outbreaks * (1 - surveillance_se)
})
```

mcmodule:: expressions

Input mcnodes will be created from data and mctable specifications

```
origin_inf_exp <- quote({
  # Proportion of years with documented outbreaks
  p_years_outbreaks <- n_years_outbreaks / n_years_info

  # Surveillance sensitivity
  surveillance_se <- mcstoc(runif, min = 0.5, max = 0.9)

  # Probability origin is infected and not detected
  p_not_detected_origin <- p_years_outbreaks * (1 - surveillance_se)
})
```

When they are evaluated, they will keep their metadata with them (their keys, the table they come from…)

mcmodule:: expressions

output mcnodes metadata will record their inputs and the combined keys of their inputs

```
origin_inf_exp <- quote({
  # Proportion of years with documented outbreaks
  p_years_outbreaks <- n_years_outbreaks / n_years_info

  # Surveillance sensitivity
  surveillance_se <- mcstoc(runif, min = 0.5, max = 0.9)

  # Probability origin is infected and not detected
  p_not_detected_origin <- p_years_outbreaks * (1 - surveillance_se)
})
```

mcmodule:: expressions

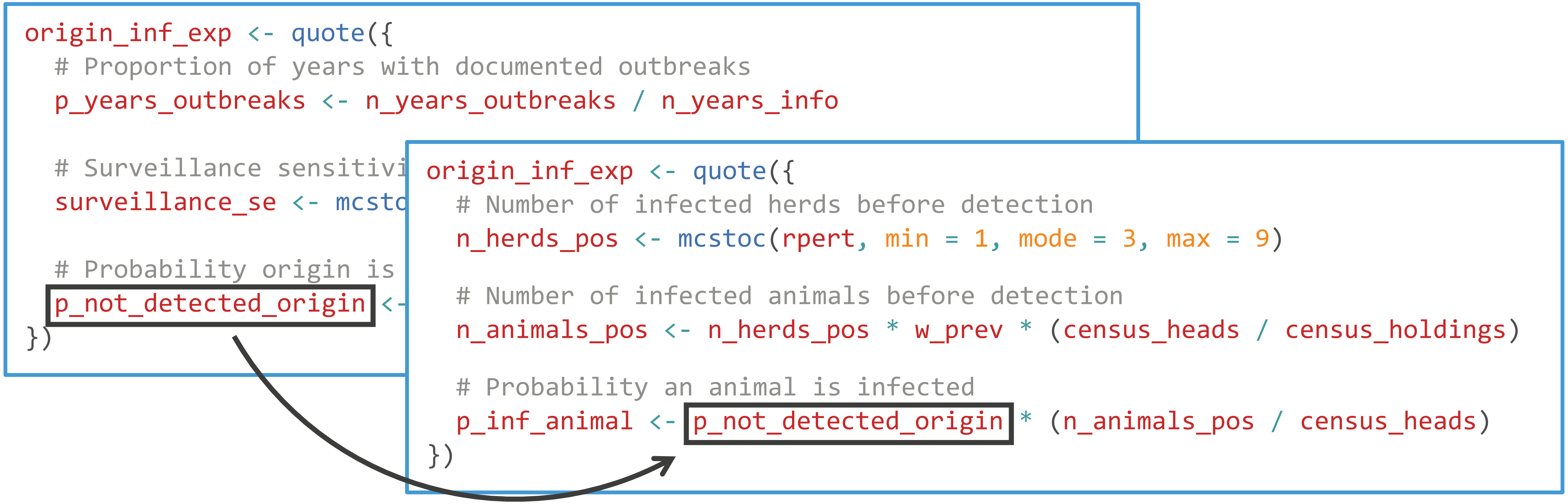
mcnodes can also be created **inline**

But variates are automatically assigned by data rows!

```
origin_inf_exp <- quote({
  # Proportion of years with documented outbreaks
  p_years_outbreaks <- n_years_outbreaks / n_years_info

  # Surveillance sensitivity
  surveillance_se <- mcstoc(runif, min = 0.5, max = 0.9)

  # Probability origin is infected and not detected
  p_not_detected_origin <- p_years_outbreaks * (1 - surveillance_se)
})
```

mcmodule:: expressions

One mcmodule can have **multiple expressions**



```
origin_inf_exp <- quote({
  # Proportion of years with documented outbreaks
  p_years_outbreaks <- n_years_outbreaks / n_years_info

  # Surveillance sensitivi
  surveillance_se <- mcsto

  # Probability origin is
  p_not_detected_origin <-
})
```

```
origin_inf_exp <- quote({
  # Number of infected herds before detection
  n_herds_pos <- mcstoc(rpert, min = 1, mode = 3, max = 9)

  # Number of infected animals before detection
  n_animals_pos <- n_herds_pos * w_prev * (census_heads / census_holdings)

  # Probability an animal is infected
  p_inf_animal <- p_not_detected_origin * (n_animals_pos / census_heads)
})
```

Expressions can use mcnodes from the previous expressions

mcmodule:: expressions

```r
origin_inf_exp <- quote({
  # Proport
  p_years_ou

  # Surveill
  surveillar

  # Probabil
  p_not_dete
})
```

```r
products_inf_exp <- quote({
  # Number c
  n_herds_pc

  # Number c
  n_products

  # Probabil
  p_inf_ pro
})
```

```r
animal_inf_exp <- quote({
  # Number of infected herds before detection
  n_herds_pos <- mcstoc(rpert, min = 1, mode = 3, max = 9)

  # Number of infected animals before detection
  n_animals_pos <- n_herds_pos * w_prev * (census_heads / census_holdings)

  # Probability an animal is infected
  p_inf_animal <- p_not_detected_origin * (n_animals_pos / census_heads)
})
```

```r
animal_import_exp <- list(
  origin_inf = origin_inf_exp,
  animal_inf = animal_inf_exp
)
```

```r
products_import_exp <- list(
  origin_inf = origin_inf_exp,
  products_inf = products_inf_exp
)
```

mcmodule:: expressions – case study

```r
# Probability that the origin country is infected with sheep/goat pox
inf_origin_exp <- quote({
  # Proportion of years with documented outbreaks
  p_years_outbreaks <- n_years_outbreaks / n_years_info


  # Number of herds visited for surveillance
  n_herd_surveillance <- p_herds_surveillance * census_holdings


  # Average herd size
  herd_size <- census_heads / census_holdings


  # Number of animals inspected
  n_animals_inspected <- p_animals_inspected * herd_size


  # Surveillance programme design prevalences
  herd_design_prev <- 0.1 # Herd-level target
  system_design_prev <- 0.001 # System-level target


  # Herd-level surveillance sensitivity
  herd_se <- 1 -
    (1 -
      (n_animals_inspected *
        clinical_se_origin /
```

mcmodule:: scenarios

```r
# Create baseline scenario with current surveillance parameters
baseline_data <- imports_by_country_commodity %>%
  left_join(disease_by_country, by = "country_code") %>%
  left_join(census_by_country, by = c("country_code", "species")) %>%
  mutate(scenario_id = "0") # Baseline must always be “0”


# Enhanced surveillance in destination scenario
enhanced_dest_data <- baseline_data %>%
  mutate(
    scenario_id = "Destination surveillance",
    clinical_se_dest_min = 0.85,
    clinical_se_dest_max = 0.95
  )
```

mcmodule::

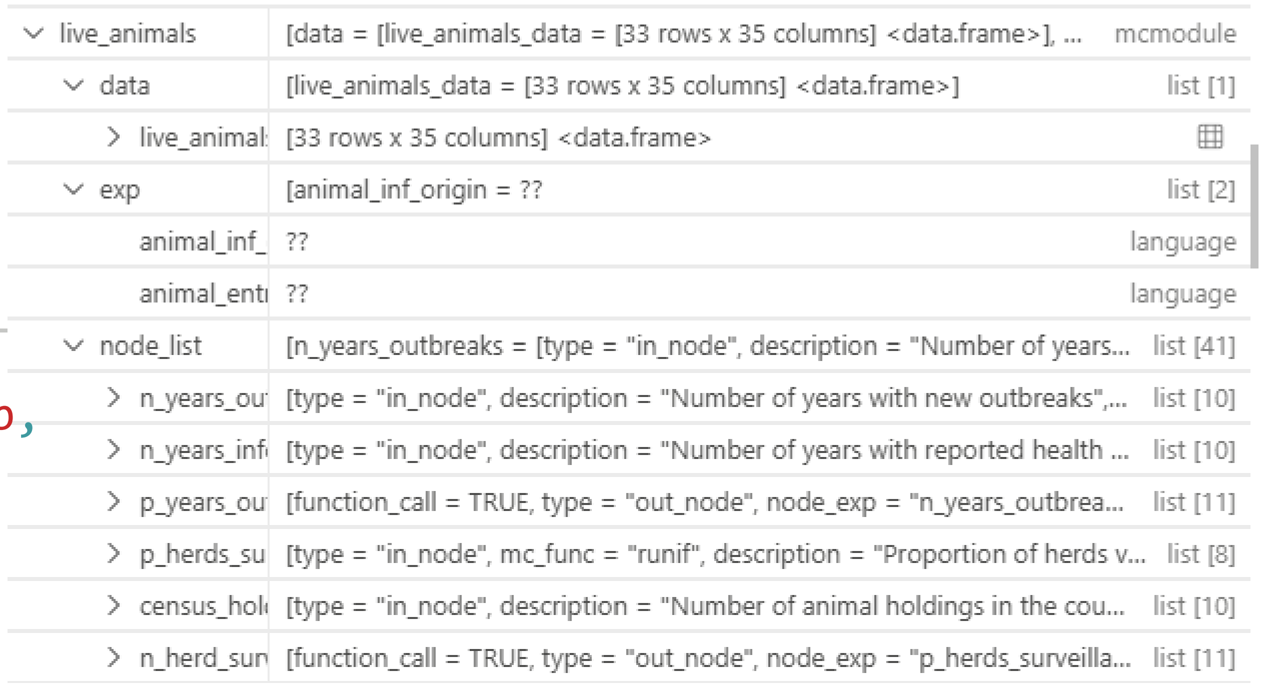
Evaluate a module

mcmodule::eval_module()

```r
# Filter live animals data
live_animals_data <- bind_rows(
  baseline_data,
  enhanced_dest_data,
  enhanced_origin_data
) %>%
  filter(pathway == "live_animals")

# Combine expressions
live_animals_exp <- list(
  animal_inf_origin = inf_origin_exp,
  animal_entry = animal_entry_exp
)

# Evaluate live animals model
live_animals <- eval_module(
  exp = live_animals_exp,
  data = live_animals_data
)
```

mcmodule::eval_module()

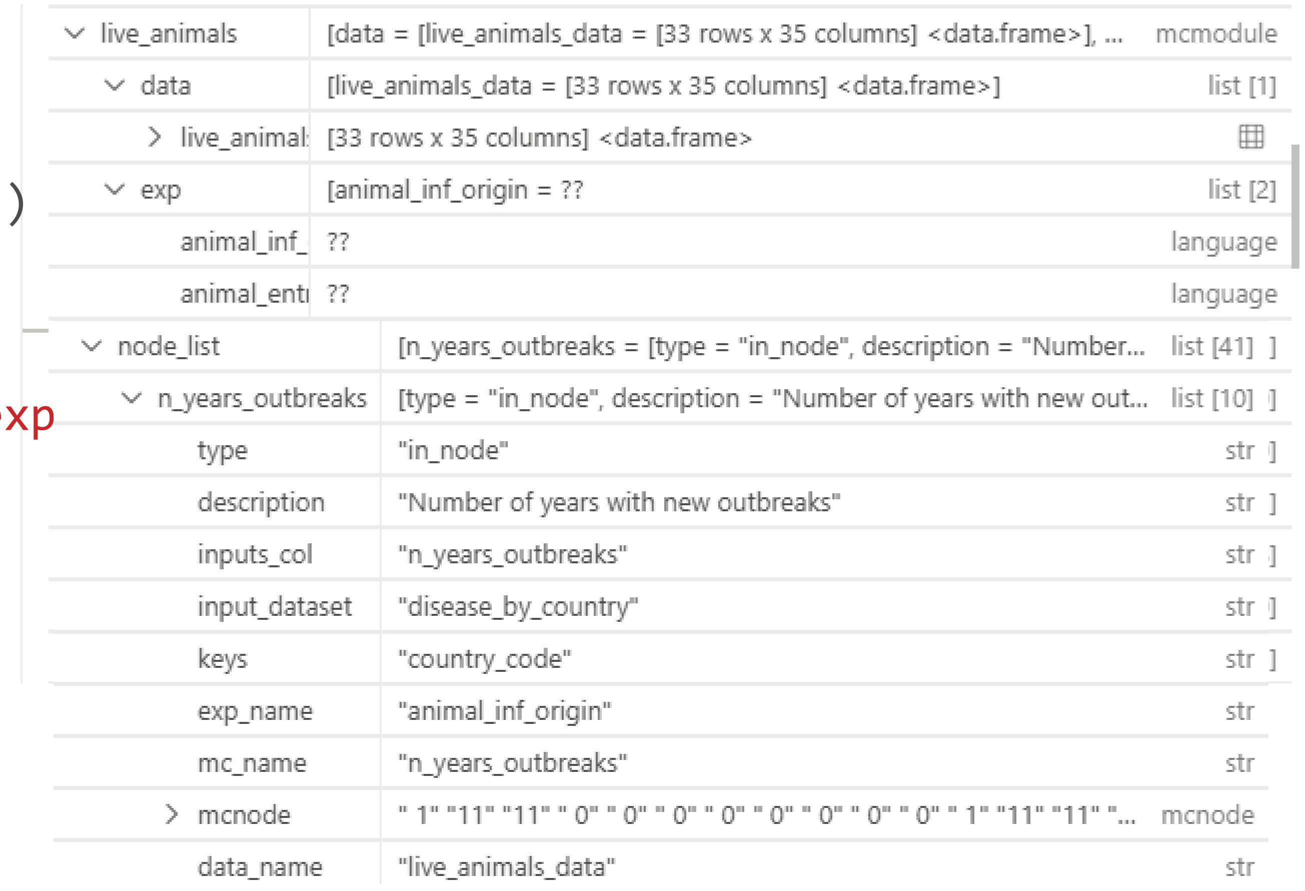
```r
# Filter live animals data
live_animals_data <- bind_rows(
  baseline_data,
  enhanced_dest_data,
  enhanced_origin_data
) %>%
  filter(pathway == "live_animals")

# Combine expressions
live_animals_exp <- list(
  animal_inf_origin = inf_origin_exp,
  animal_entry = animal_entry_exp
)

# Evaluate live animals model
live_animals <- eval_module(
  exp = live_animals_exp,
  data = live_animals_data
)
```



⌄ live_animals	[data = [live_animals_data = [33 rows x 35 columns] <data.frame>], ...	mcmodule
⌄ data	[live_animals_data = [33 rows x 35 columns] <data.frame>]	list [1]
› live_animal	[33 rows x 35 columns] <data.frame>	▦
⌄ exp	[animal_inf_origin = ??	list [2]
animal_inf_	??	language
animal_entı	??	language
⌄ node_list	[n_years_outbreaks = [type = "in_node", description = "Number of years...	list [41]
› n_years_ou	[type = "in_node", description = "Number of years with new outbreaks",...	list [10]
› n_years_inf	[type = "in_node", description = "Number of years with reported health ...	list [10]
› p_years_ou	[function_call = TRUE, type = "out_node", node_exp = "n_years_outbrea...	list [11]
› p_herds_su	[type = "in_node", mc_func = "runif", description = "Proportion of herds v...	list [8]
› census_hol	[type = "in_node", description = "Number of animal holdings in the cou...	list [10]
› n_herd_sur	[function_call = TRUE, type = "out_node", node_exp = "p_herds_surveilla...	list [11]

mcmodule::eval_module()

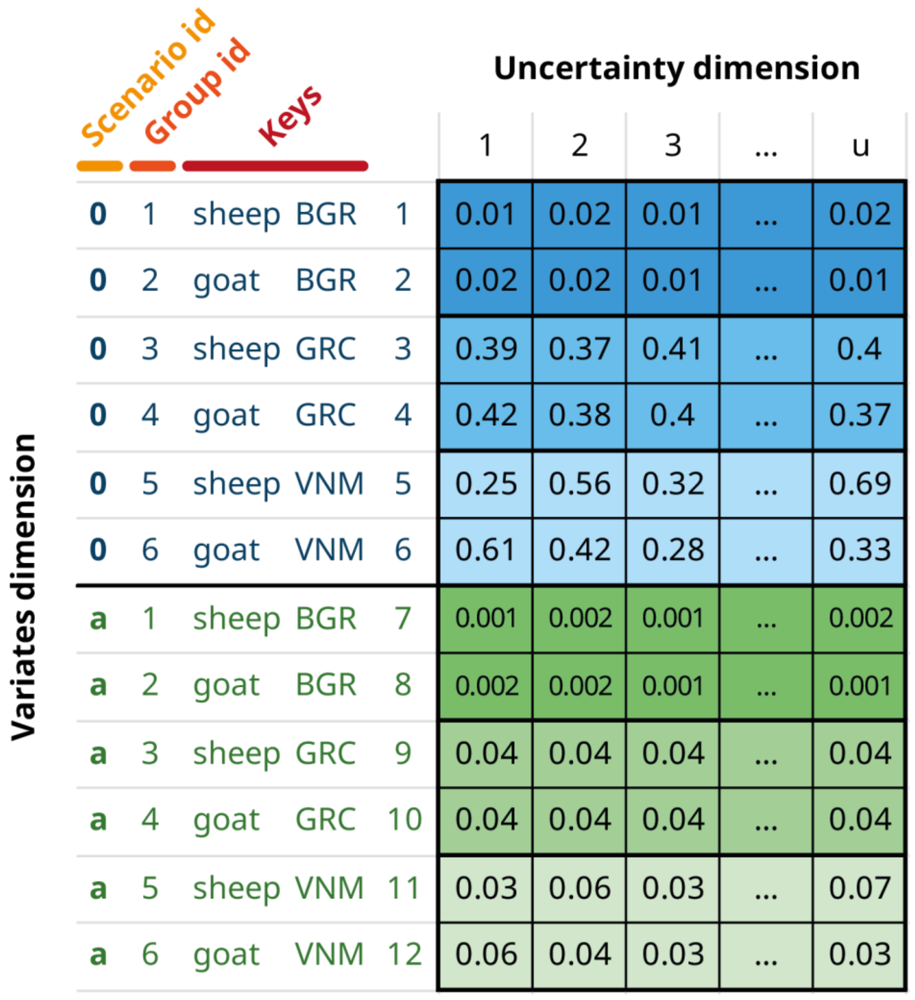
```r
# Filter live animals data
live_animals_data <- bind_rows(
  baseline_data,
  enhanced_dest_data,
  enhanced_origin_data
) %>%
  filter(pathway == "live_animals")

# Combine expressions
live_animals_exp <- list(
  animal_inf_origin = inf_origin_exp
  animal_entry = animal_entry_exp
)

# Evaluate live animals model
live_animals <- eval_module(
  exp = live_animals_exp,
  data = live_animals_data
)
```



⌄ live_animals	[data = [live_animals_data = [33 rows x 35 columns] <data.frame>], ...	mcmodule
⌄ data	[live_animals_data = [33 rows x 35 columns] <data.frame>]	list [1]
› live_animal	[33 rows x 35 columns] <data.frame>	▦
⌄ exp	[animal_inf_origin = ??	list [2]
animal_inf_	??	language
animal_entı	??	language
⌄ node_list	[n_years_outbreaks = [type = "in_node", description = "Number...	list [41]]
⌄ n_years_outbreaks	[type = "in_node", description = "Number of years with new out...	list [10]]
type	"in_node"	str]
description	"Number of years with new outbreaks"	str]
inputs_col	"n_years_outbreaks"	str]
input_dataset	"disease_by_country"	str]
keys	"country_code"	str]
exp_name	"animal_inf_origin"	str
mc_name	"n_years_outbreaks"	str
› mcnode	" 1" "11" "11" " 0" " 0" " 0" " 0" " 0" " 0" " 0" " 0" " 1" "11" "11" "...	mcnode
data_name	"live_animals_data"	str

mcmodule::eval_module()

```r
# Filter live animals data
live_animals_data <- bind_rows(
  baseline_data,
  enhanced_dest_data,
  enhanced_origin_data
) %>%
  filter(pathway == "live_animals")

# Combine expressions
live_animals_exp <- list(
  animal_inf_origin = inf_origin_exp,
  animal_entry = animal_entry_exp
)

# Evaluate live animals model
live_animals <- eval_module(
  exp = live_animals_exp,
  data = live_animals_data
)
```



Scenario id
Group id
Keys
Uncertainty dimension
Variates dimension

Scenario id	Group id	Keys			1	2	3	...	u
0	1	sheep	BGR	1	0.01	0.02	0.01	...	0.02
0	2	goat	BGR	2	0.02	0.02	0.01	...	0.01
0	3	sheep	GRC	3	0.39	0.37	0.41	...	0.4
0	4	goat	GRC	4	0.42	0.38	0.4	...	0.37
0	5	sheep	VNM	5	0.25	0.56	0.32	...	0.69
0	6	goat	VNM	6	0.61	0.42	0.28	...	0.33
a	1	sheep	BGR	7	0.001	0.002	0.001	...	0.002
a	2	goat	BGR	8	0.002	0.002	0.001	...	0.001
a	3	sheep	GRC	9	0.04	0.04	0.04	...	0.04
a	4	goat	GRC	10	0.04	0.04	0.04	...	0.04
a	5	sheep	VNM	11	0.03	0.06	0.03	...	0.07
a	6	goat	VNM	12	0.06	0.04	0.03	...	0.03

mcmodule::

Calculate trial totals

mcmodule:: simple trial totals

```r
# Probability for introducing at least one infected animal
# in one import operations
live_animals <- trial_totals(
  mcmodule = live_animals,
  mc_names = "p_entry_animal",
  trials_n = "import_size”
)
```

mcmodule:: simple trial totals

```r
# Probability for introducing at least one infected animal
# in one import operations
live_animals <- trial_totals(
  mcmodule = live_animals,
  mc_names = "p_entry_animal",
  trials_n = "import_size”
)
```

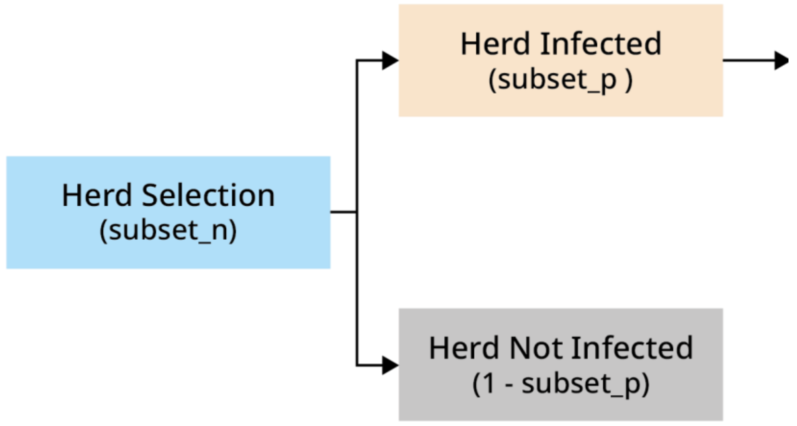
```
> animal_products$node_list$p_product_inf_animal_set$summary
                  mc_name country_code species       commodity         mean           sd Min 2.5% 25%
1  p_product_inf_animal_set          BGR   sheep Raw wool, sheep 1.165756e-14 4.858792e-14   0    0   0 0.0
2  p_product_inf_animal_set          EGY   sheep Raw wool, sheep 0.000000e+00 0.000000e+00   0    0   0 0.0
3  p_product_inf_animal_set          GRC    goat  Raw hide, goat 2.084691e-15 6.891559e-15   0    0   0 0.0
4  p_product_inf_animal_set          GRC   sheep Raw wool, sheep 1.774582e-18 1.560022e-17   0    0   0 0.0
5  p_product_inf_animal_set          TUN   sheep Raw wool, sheep 0.000000e+00 0.000000e+00   0    0   0 0.0
```

mcmodule:: hierarchical trial totals

```r
# Probability for introducing at least one infected animal
# in at least one import operations
live_animals <- trial_totals(
  mcmodule = live_animals,
  mc_names = "p_entry_animal",
  trials_n = "import_size",
  subsets_p = "p_inf_origin",
  subsets_n = "import_operations"
)
```

mcmodule:: hierarchical trial totals

```r
# Probability for introducing at least one infected animal
# in at least one import operations
live_animals <- trial_totals(
  mcmodule = live_animals,
  mc_names = "p_entry_animal",
  trials_n = "import_size",
  subsets_p = "p_inf_origin",
  subsets_n = "import_operations"
)
```

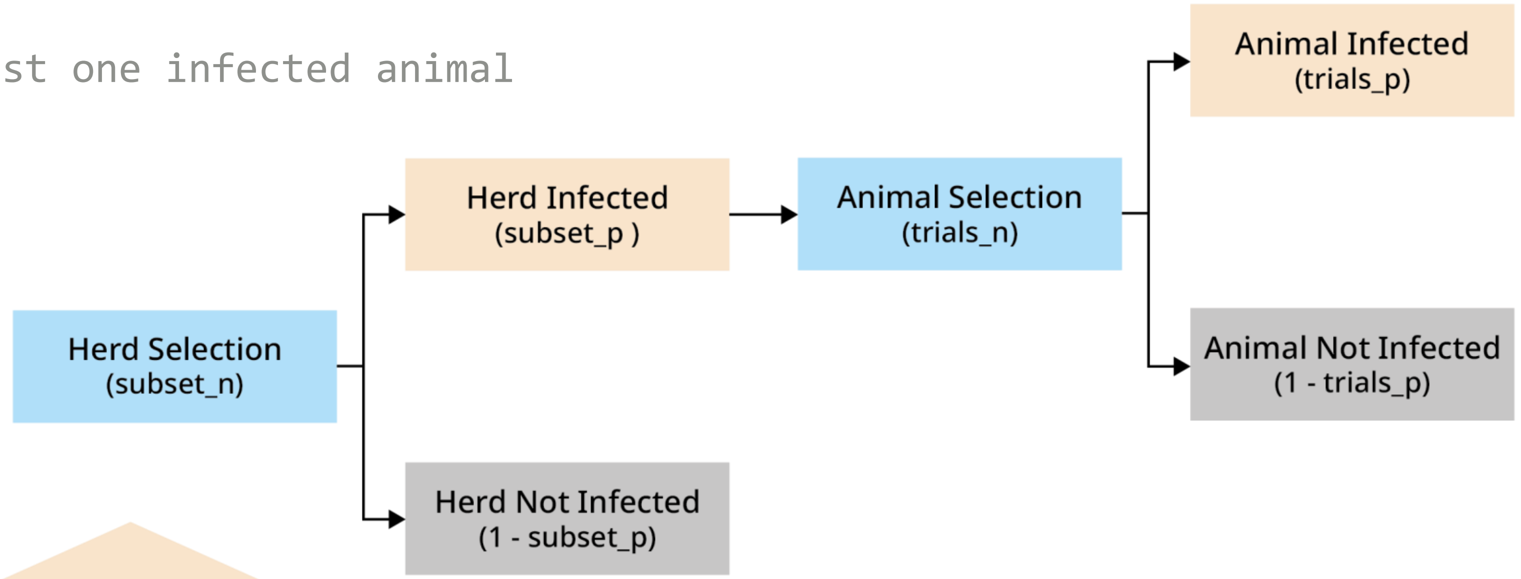


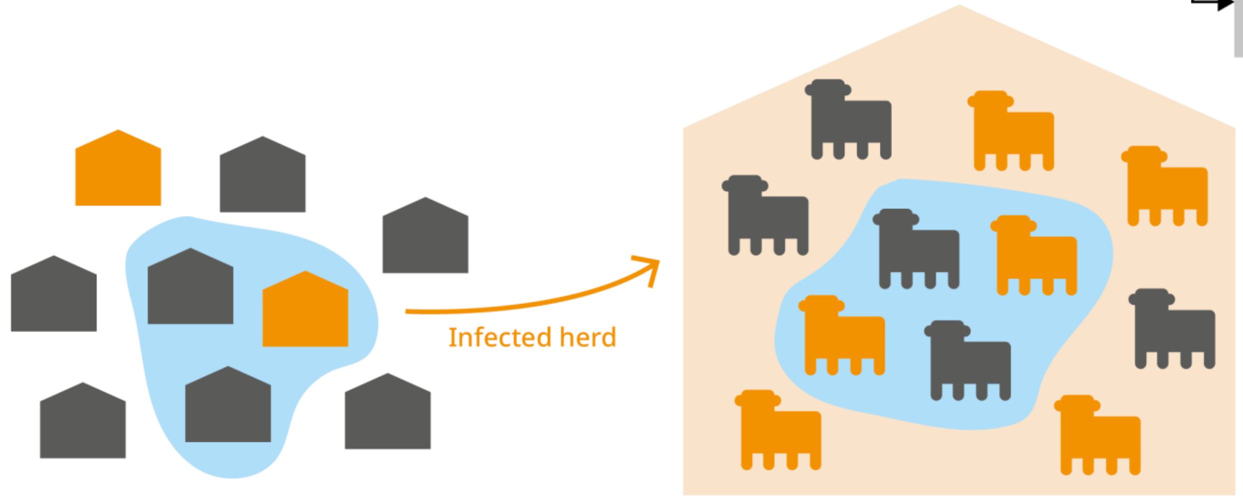
Infected herd

subset_n = 3, subset_p = 0.2

mcmodule:: hierarchical trial totals

```r
# Probability for introducing at least one infected animal
# in at least one import operations
live_animals <- trial_totals(
  mcmodule = live_animals,
  mc_names = "p_entry_animal",
  trials_n = "import_size",
  subsets_p = "p_inf_origin",
  subsets_n = "import_operations"
)
```





subset_n = 3, subset_p = 0.2

trials_n = 4, trials_p = 0.5

mcmodule::

Combine modules

mcmodule:: add_prefix()

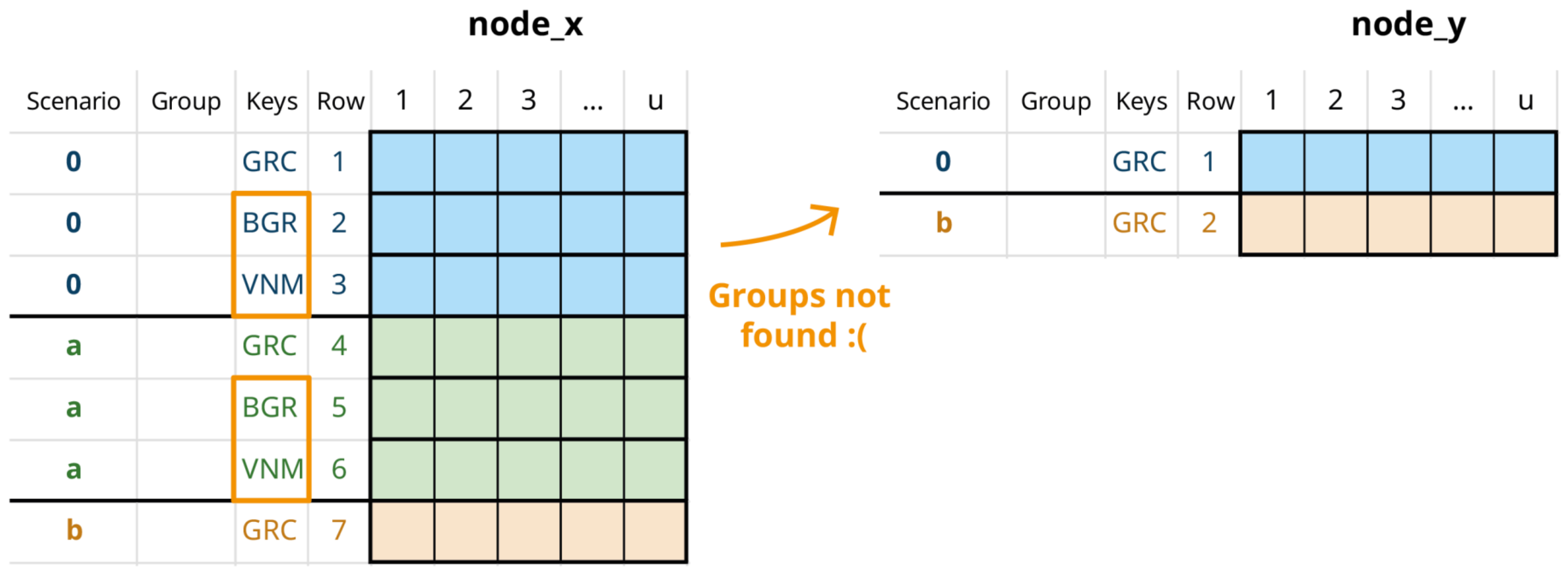
```r
# Evaluate live animals pathway
live_animals <- eval_module(
  exp = live_animals_exp,
  data = live_animals_data
)

# Evaluate animal products pathway
animal_products <- eval_module(
  exp = animal_products_exp,
  data = animal_products_data
)

# Add prefixes to distinguish pathway nodes
live_animals <- add_prefix(live_animals, prefix = "live")
animal_products <- add_prefix(animal_products, prefix = "products")
```

mcmodule:: combine_modules()

```r
# Evaluate live animals pathway
live_animals <- eval_module(
  exp = live_animals_exp,
  data = live_animals_data
)

# Evaluate animal products pathway
animal_products <- eval_module(
  exp = animal_products_exp,
  data = animal_products_data
)

# Add prefixes to distinguish pathway nodes
live_animals <- add_prefix(live_animals, prefix = "live")
animal_products <- add_prefix(animal_products, prefix = "products")


# Combine modules
animal_imports <- combine_modules(live_animals, animal_products)
```

mcmodule:: at_least_one()

```r
# Calculate combined probability of entry from either pathway
animal_imports <- at_least_one(
  mcmodule = animal_imports,
  mc_names = c("products_p_product_inf_animal_set", "live_p_entry_animal_set"),
  name = "p_entry_total"
)
```



node_x

Scenario	Group	Keys	Row	1	2	3	...	u
0		GRC	1	🟦	🟦	🟦	🟦	🟦
0		BGR	2	🟦	🟦	🟦	🟦	🟦
0		VNM	3	🟦	🟦	🟦	🟦	🟦
a		GRC	4	🟩	🟩	🟩	🟩	🟩
a		BGR	5	🟩	🟩	🟩	🟩	🟩
a		VNM	6	🟩	🟩	🟩	🟩	🟩
b		GRC	7	🟧	🟧	🟧	🟧	🟧

node_y

Scenario	Group	Keys	Row	1	2	3	...	u
0		GRC	1	🟦	🟦	🟦	🟦	🟦
b		GRC	2	🟧	🟧	🟧	🟧	🟧

mcmodule:: at_least_one()

```r
# Calculate combined probability of entry from either pathway
animal_imports <- at_least_one(
  mcmodule = animal_imports,
  mc_names = c("products_p_product_inf_animal_set", "live_p_entry_animal_set"),
  name = "p_entry_total"
)
```

Node x			Node y		
Scenario	Group	Row	Scenario	Group	Row
0	1	1	**0**	1	1
0	2	2	**null**	2	
0	3	3	**null**	3	
a	1	4	**0**	1	1
a	2	5	**null**	2	
a	3	6	**null**	3	
0	1	1	**b**	1	2

mcmodule:: at_least_one()

```r
# Calculate combined probability of entry from either pathway
animal_imports <- at_least_one(
  mcmodule = animal_imports,
  mc_names = c("products_p_product_inf_animal_set", "live_p_entry_animal_set"),
  name = "p_entry_total"
)
```

Node x			Node y		
Scenario	Group	Row	Scenario	Group	Row
0	1	1	**0**	1	1
0	2	2	**null**	2	
0	3	3	**null**	3	
a	1	4	**0**	1	1
a	2	5	**null**	2	
a	3	6	**null**	3	
0	1	1	**b**	1	2

node_xy

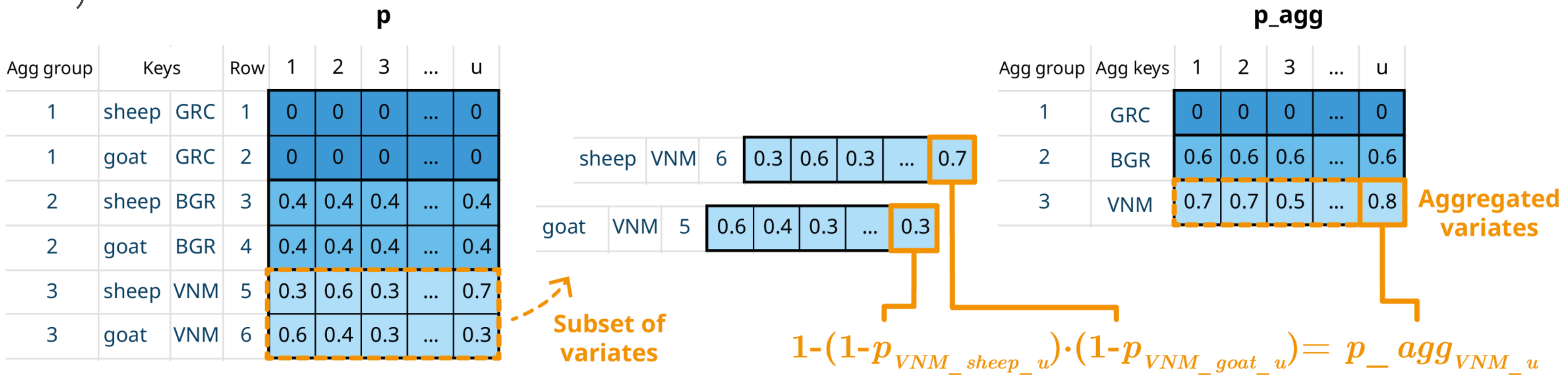
Scenario	Group	Keys	Row	1	2	3	...	u
0	1	GRC	1	🟦	🟦	🟦	🟦	🟦
0	2	BGR	2	🟦	🟦	🟦	🟦	🟦
0	3	VNM	3	🟦	🟦	🟦	🟦	🟦
a	1	GRC	4	🟩	🟩	🟩	🟩	🟩
a	2	BGR	5	🟩	🟩	🟩	🟩	🟩
a	3	VNM	6	🟩	🟩	🟩	🟩	🟩
b	1	GRC	7	🟧	🟧	🟧	🟧	🟧

mcmodule::

Calculate aggregation totals

mcmodule:: aggregation totals

```r
# 1. Total probability by country and scenario
animal_imports <- agg_totals(
  mcmodule = animal_imports,
  mc_name = "p_entry_total",
  agg_keys = c("country_code", "scenario_id"),
  name = "p_entry_total_by_country"
)
```



p

Agg group	Keys		Row	1	2	3	...	u
1	sheep	GRC	1	0	0	0	...	0
1	goat	GRC	2	0	0	0	...	0
2	sheep	BGR	3	0.4	0.4	0.4	...	0.4
2	goat	BGR	4	0.4	0.4	0.4	...	0.4
3	sheep	VNM	5	0.3	0.6	0.3	...	0.7
3	goat	VNM	6	0.6	0.4	0.3	...	0.3

sheep	VNM	6	0.3	0.6	0.3	...	0.7
goat	VNM	5	0.6	0.4	0.3	...	0.3

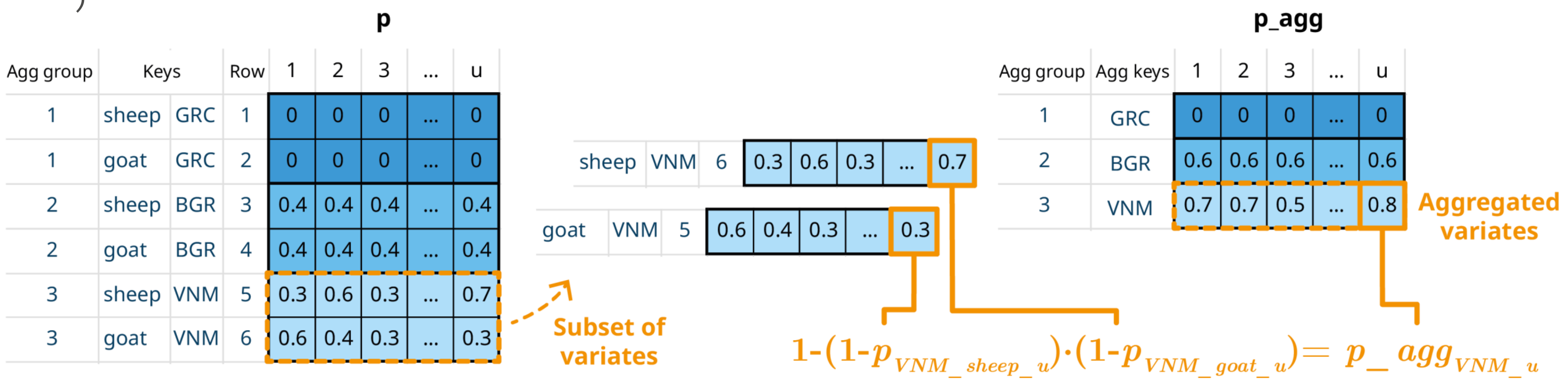
p_agg

Agg group	Agg keys	1	2	3	...	u
1	GRC	0	0	0	...	0
2	BGR	0.6	0.6	0.6	...	0.6
3	VNM	0.7	0.7	0.5	...	0.8

$$1\text{-}(1\text{-}p_{VNM\_sheep\_u})\cdot(1\text{-}p_{VNM\_goat\_u}) = p\_agg_{VNM\_u}$$

mcmodule:: aggregation totals

```r
# 1. Total probability by country and scenario
animal_imports <- agg_totals(
  mcmodule = animal_imports,
  mc_name = "p_entry_total",
  agg_keys = c("country_code", "scenario_id"),
  name = "p_entry_total_by_country"
)
```



p

Agg group	Keys		Row	1	2	3	...	u
1	sheep	GRC	1	0	0	0	...	0
1	goat	GRC	2	0	0	0	...	0
2	sheep	BGR	3	0.4	0.4	0.4	...	0.4
2	goat	BGR	4	0.4	0.4	0.4	...	0.4
3	sheep	VNM	5	0.3	0.6	0.3	...	0.7
3	goat	VNM	6	0.6	0.4	0.3	...	0.3

sheep	VNM	6	0.3	0.6	0.3	...	0.7
goat	VNM	5	0.6	0.4	0.3	...	0.3

p_agg

Agg group	Agg keys	1	2	3	...	u
1	GRC	0	0	0	...	0
2	BGR	0.6	0.6	0.6	...	0.6
3	VNM	0.7	0.7	0.5	...	0.8

$$1\text{-}(1\text{-}p_{VNM\_sheep\_u})\cdot(1\text{-}p_{VNM\_goat\_u}) = p\_agg_{VNM\_u}$$

By default, it calculates the **combined probability of independent events**, but it can take functions such as sum, product, average...

mcmodule:: aggregation totals

```r
# 1. Total probability by country and scenario
animal_imports <- agg_totals(
  mcmodule = animal_imports,
  mc_name = "p_entry_total",
  agg_keys = c("country_code", "scenario_id"),
  name = "p_entry_total_by_country"
)

# 2. Total probability by commodity and scenario
animal_imports <- agg_totals(
  mcmodule = animal_imports,
  mc_name = "p_entry_total",
  agg_keys = c("commodity", "scenario_id"),
  name = "p_entry_total_by_commodity"
)

# 3. Total probability by scenario only (overall risk aggregation)
animal_imports <- agg_totals(
  mcmodule = animal_imports,
  mc_name = "p_entry_total",
  agg_keys = "scenario_id",
  name = "p_entry_total_by_scenario"
)
```

mcmodule::

Analysis and visualization

mcmodule::mc_summary()

```
> # Summary statistics by scenario (overall risk)
+ mc_summary(animal_imports, "p_entry_total_by_scenario")
```

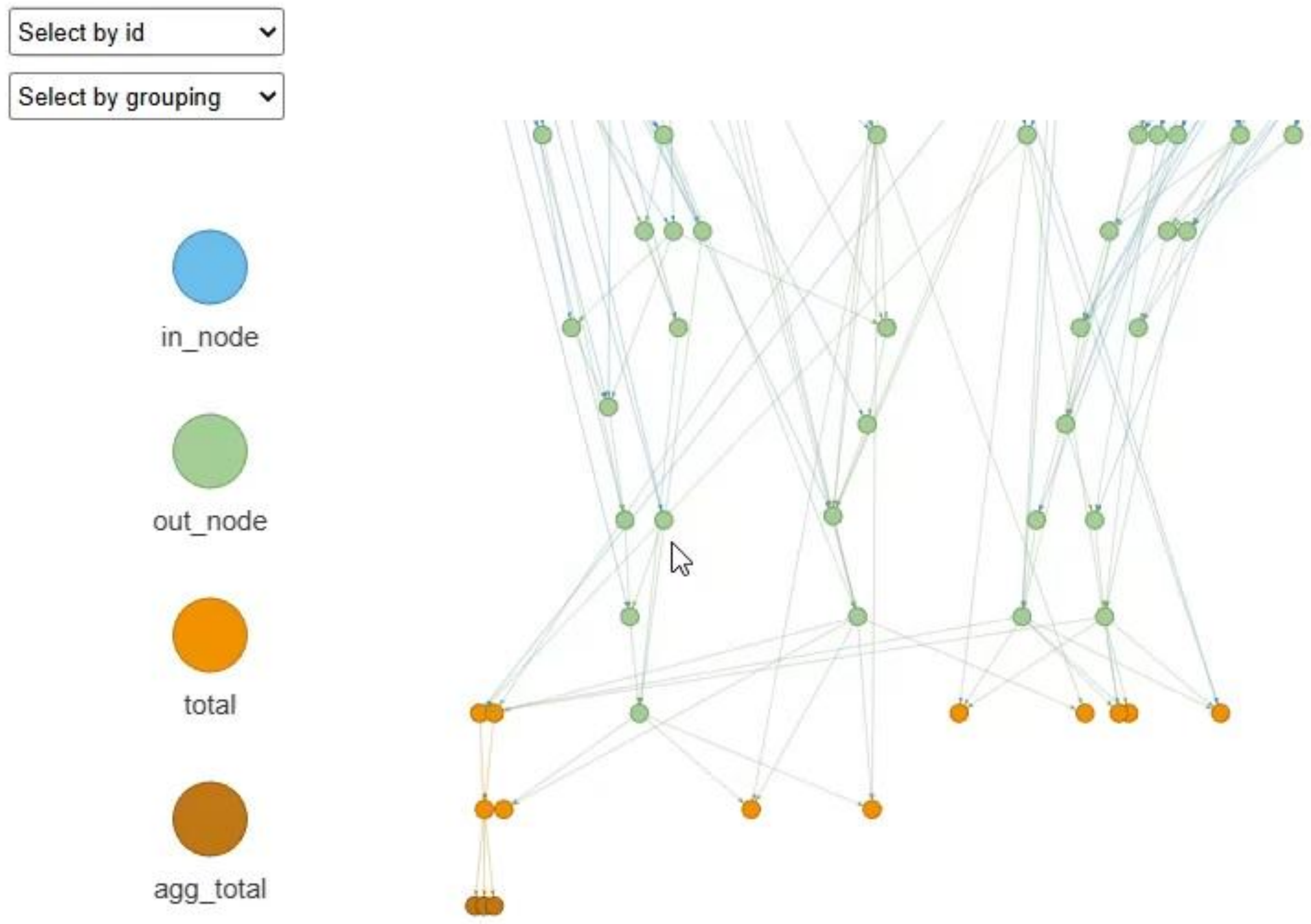
```
                   mc_name              scenario_id         mean           sd          Min         2.5%          25%          50%
1  p_entry_total_by_scenario                       0 1.461221e-06 1.366041e-06 3.860955e-08 1.439198e-07 5.241053e-07 1.048014e-06
41 p_entry_total_by_scenario Destination surveillance 4.097420e-07 4.304603e-07 3.552278e-09 3.282649e-08 1.436553e-07 2.842297e-07
81 p_entry_total_by_scenario      Origin surveillance 1.515106e-06 1.530098e-06 4.697465e-08 1.826068e-07 5.593250e-07 1.085713e-06
            75%        97.5%          Max  nsv Na's
1  1.954708e-06 5.084258e-06 1.014417e-05 1001    0
41 5.271032e-07 1.572215e-06 5.311798e-06 1001    0
81 1.983311e-06 5.188272e-06 2.017941e-05 1001    0
```

mcmodule::mc_summary()

```
> # Summary statistics by scenario (overall risk)
+ mc_summary(animal_imports, "p_entry_total_by_scenario")
```

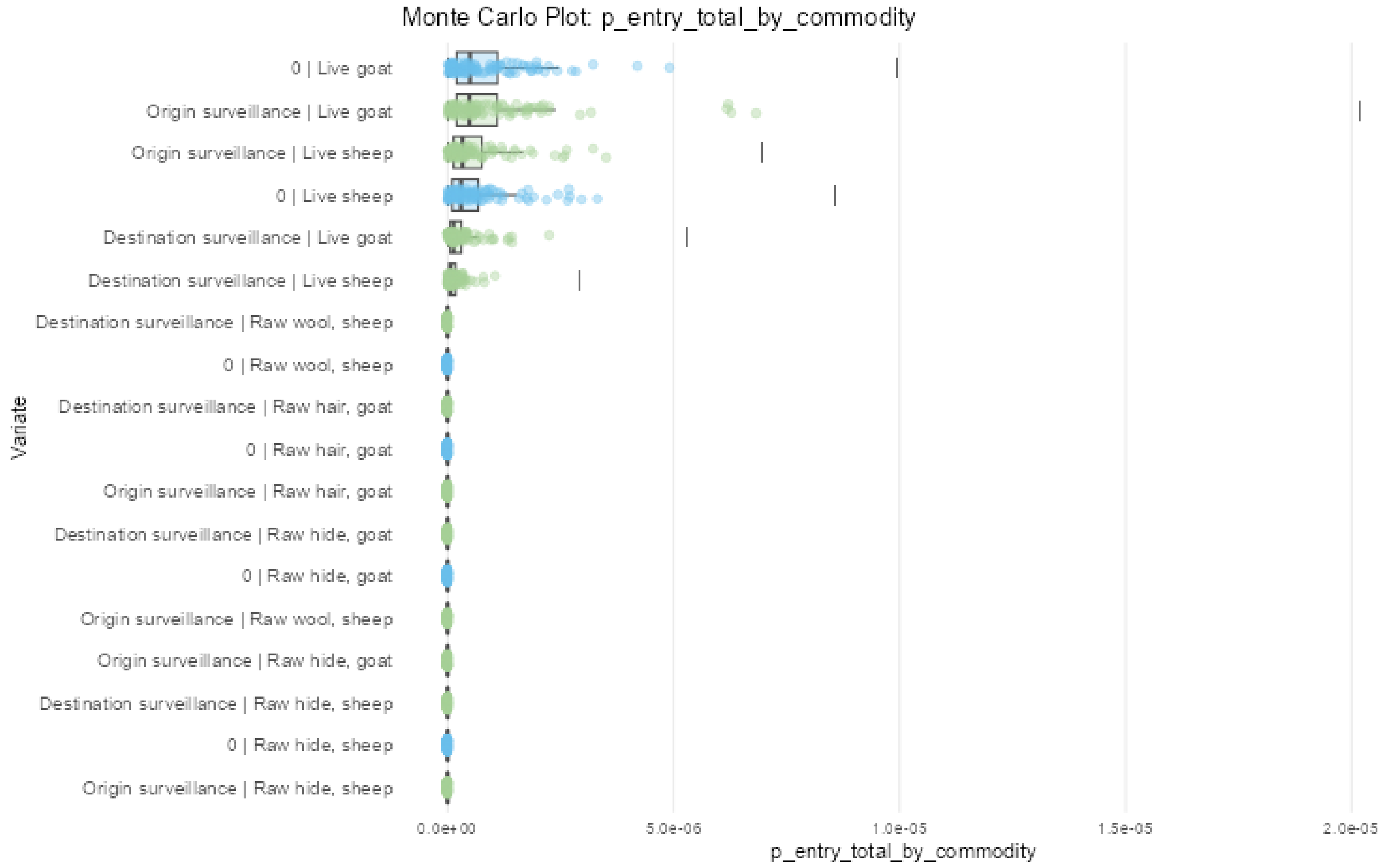
```
                     mc_name             scenario_id         mean           sd          Min         2.5%          25%          50%
1  p_entry_total_by_scenario                       0 1.461221e-06 1.366041e-06 3.860955e-08 1.439198e-07 5.241053e-07 1.048014e-06
41 p_entry_total_by_scenario Destination surveillance 4.097420e-07 4.304603e-07 3.552278e-09 3.282649e-08 1.436553e-07 2.842297e-07
81 p_entry_total_by_scenario      Origin surveillance 1.515106e-06 1.530098e-06 4.697465e-08 1.826068e-07 5.593250e-07 1.085713e-06
            75%        97.5%          Max  nsv Na's
1  1.954708e-06 5.084258e-06 1.014417e-05 1001    0
41 5.271032e-07 1.572215e-06 5.311798e-06 1001    0
81 1.983311e-06 5.188272e-06 2.017941e-05 1001    0
```

mcmodule::mcmodule_network()



mcmodule::mc_plot()

```r
# Plot total probability of entry by commodity
mc_plot(animal_imports, "p_entry_total_by_commodity", order_by = "median")
```



mcmodule::mcmodule_corr()

```r
# Correlation betwen inputs
# and total output
corr_analysis <- mcmodule_corr(
  mcmodule = animal_imports,
  output = "p_entry_total",
  progress = TRUE,
  print_summary = TRUE,
  method = "spearman"
)
```

```
Analysis Parameters:
- Analysis type: Global output
- Output node: p_entry_total
- Correlation method(s): spearman
- Missing value handling: all.obs

Results Summary:
- Total correlations calculated: 1836
- Top 5 most influential inputs (by absolute
mean correlation):
1. products_import_qty: 0.0790
2. products_clinical_se_origin: -0.0383
3. live_import_operations: 0.0120
4. live_clinical_se_origin: -0.0105
5. live_clinical_se_dest: -0.0103

Inputs by Correlation Strength:
- Moderate: products_import_qty
- Weak: products_import_qty
```

mcmodule::mcmodule_converg()

```r
# Analyse convergence of live animals pathway
converg_live <- mcmodule_converg(
  mcmodule = live_animals,
  from_quantile = 0.95,
  to_quantile = 1.0,
  print_summary = TRUE
)
```

```
     Analysis Parameters:
     - Number of simulations: 1001
     - Simulation quantile range: 0.95 to 1
     - Simulations range: 950 to 1001 (51 simulations)

     Convergence Results:
     - Total nodes analyzed: 711
     - Nodes with divergence below 0.001: 427 (60.06%)
     - Nodes with divergence below 1% of their mean: 520 (73.14%)
     - Nodes with divergence below 0.001 or 1% of their mean: 613 (86.22%)
     - Nodes with divergence below 0.001 or 2.5% of their mean: 679 (95.50%)
     - Nodes with divergence below 0.001 or 5% of their mean: 705 (99.16%)

     6 (0.84%) nodes did not converge at 5% threshold :(
```

mcmodule::

Tricks and tweaks

mcmodule:: tricks and tweaks

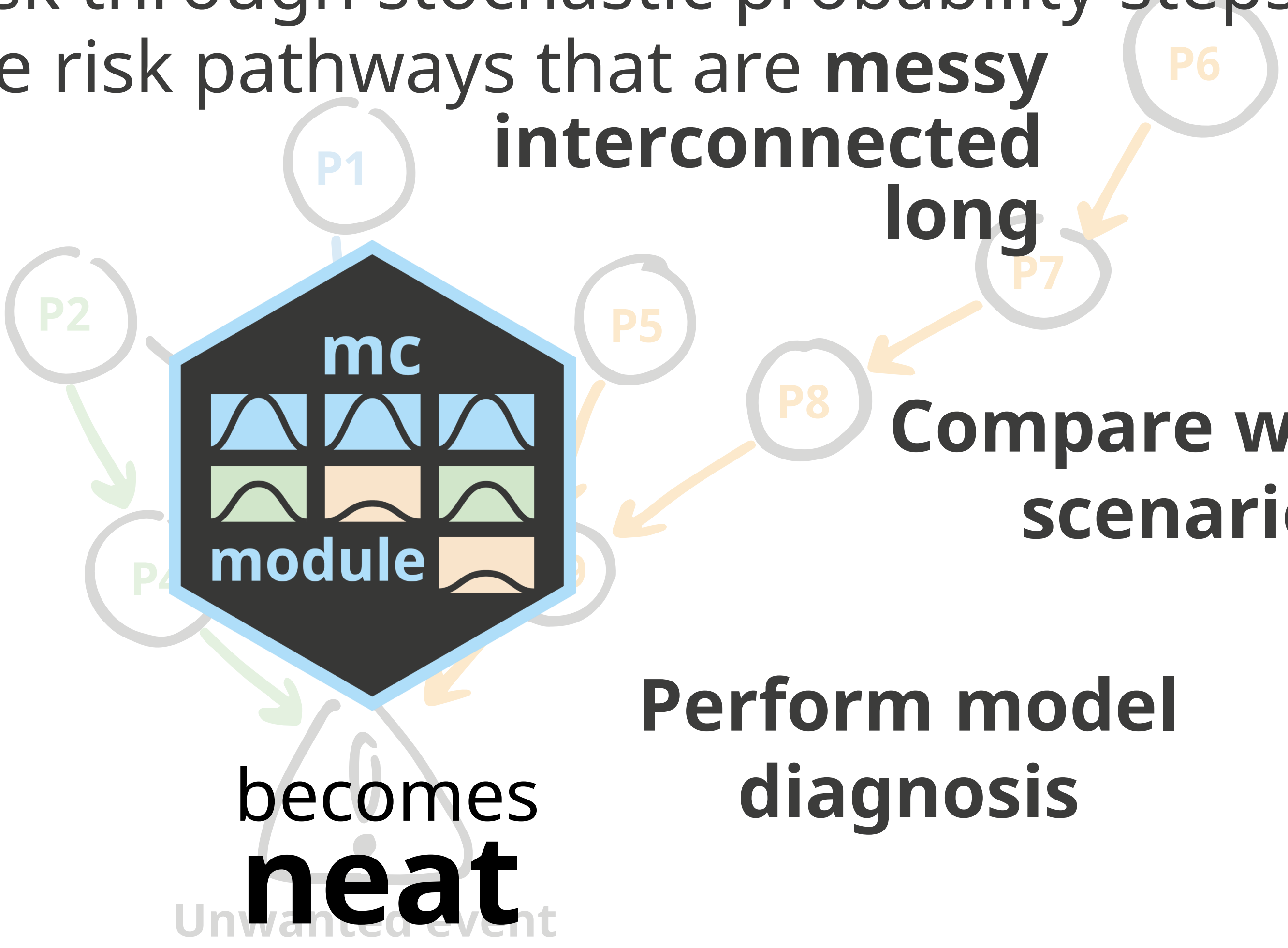
- Replace missing or infinite values to prevent calculation errors using `mcnode_na_rm()`
- Use the `name` parameter in functions like `at_least_one()` and `agg_totals()` to customize output node name
- Some functions can work outside modules: `create_mcnodes()` and `mc_summary()` (useful for prototyping)
- Convert to other formats: Transform mcmodules into matrices or mc objects with `mcmodule_to_matrices()` and `mcmodule_to_mc()`

mcmodule::

Conclusion

mcmodule:: conclusion

Quantifying risk through stochastic probability steps in real-life risk pathways that are **messy**
interconnected
long



Multi-variate

Not-aligned pathway variates

Compare what-if scenarios

Perform model diagnosis

becomes
neat

mcmodule:: conclusion

Modularity

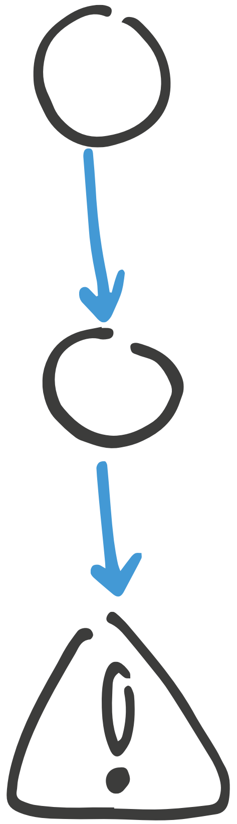
Scalability

Transparency & traceability

Communication

Diagnosis

mcmodule:: next directions



- We are currently planning to add **global sensitivity analysis** based on Morris screening, using the existing local sensitivity (OAT) support
- We are working on building **compatibility with other R packages**, to facilitate integration with broader risk assessment workflows
- To stay updated watch our repository: https://github.com/NataliaCiria/mcmodule
- We encourage you to explore the package further, adapt it to your own use cases, report bugs, and contribute feedback or improvements.

mcmodule:: acknowledgments

Natalia
Ciria

Dr. Alberto
Allepuz

Dr. Giovanna
Ciaravino

Funded by
the European Union

And thank you to all my colleagues and friends for listening to me complain about my bugs

https://nataliaciria.com/mcmodule/

Contact: natalia.ciria@uab.cat

Website: https://nataliaciria.com
LinkedIn: https://www.linkedin.com/in/natalia-ciria-artiga/
GitHub: https://github.com/NataliaCiria