Probabilistic Risk Analysis and Bayesian Decision Theory

Marcel van Oijen[*] Mark Brewer[†]

```
knitr::opts_chunk$set( collapse=TRUE, comment=">" )
library(copula)
library(DiagrammeR)
library(DiagrammeRsvg)
library(fields)
library(MCMCpack)
library(mvtnorm)
library(nimble)
library(plot.matrix)
library(readxl)
library(rsvg)
library(scales)
library(terra)
library(truncnorm)
library(VineCopula)
```

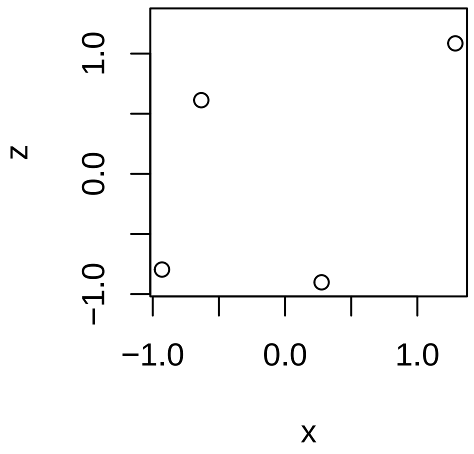
[*]Independent researcher, Edinburgh, UK - VanOijenMarcel@gmail.com
[†]BioSS Office, The James Hutton Institute, Craigiebuckler, Aberdeen AB15 8QH

Data

Sparse Dataset: {x_4,z_4}

```
set.seed(1)

n_4  <- 4              ; Sigma_4 <- matrix( c(1,0.5,0.5,1), nrow=2 )
xz_4 <- rmvnorm( n_4, c(0,0), Sigma_4 ) %*% chol( Sigma_4 )
xz_4 <- sweep( xz_4, 2, colMeans(xz_4) )
xz_4 <- xz_4 %*% solve( chol(cov(xz_4)) ) %*% chol(Sigma_4)
x_4  <- xz_4[,1]       ; z_4 <- xz_4[,2]
m_4  <- colMeans(xz_4) ; S_4 <- cov(xz_4)
```



A Collection of Linear Datasets: l_xz.L

```
set.seed(1)

mu  <- c(0,0) ; Sigma  <- diag(1,2)          ; Sigma[1,2] <- Sigma[2,1] <- 0.5
n_d <- 1e2    ; l_xz.L <- vector("list",n_d) ; n <- 1e3
for(d in 1:n_d) { l_xz.L[[d]] <- rmvnorm( n, mu, Sigma ) }
```

A Collection of Nonlinear Datasets: l_xz.NL

```
set.seed(1)

n_d <- 1e2 ; l_xz.NL <- vector("list",n_d) ; n <- 1e3 ; sz <- 0.1
for(d in 1:n_d) {
  x <- runif( n, 0, 3 ) ; ez <- rnorm( n, 0, sz) ; z <- 1-exp(-x) + ez
  l_xz.NL[[d]] <- cbind(x,z) }
```

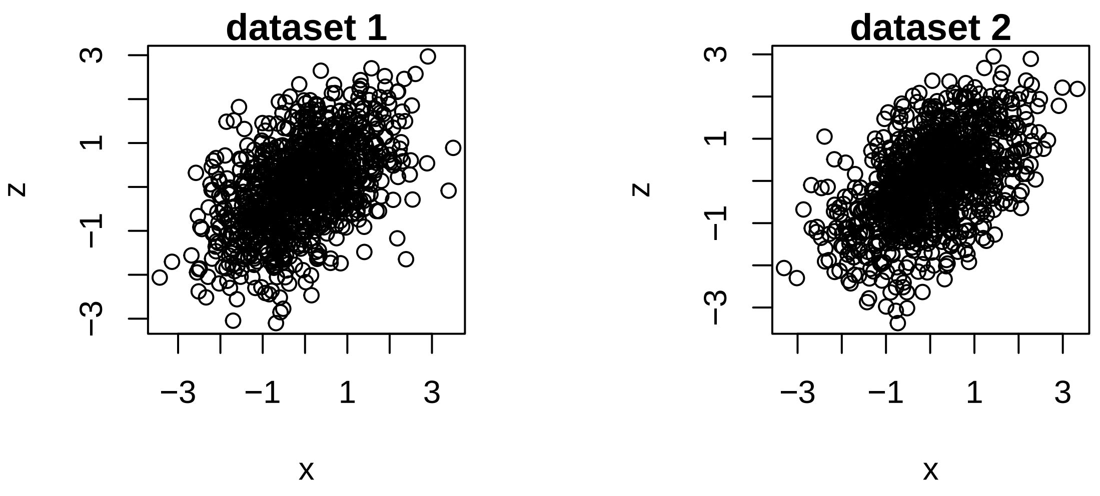


Figure 1: The first two data sets generated from a noisy linear relationship.

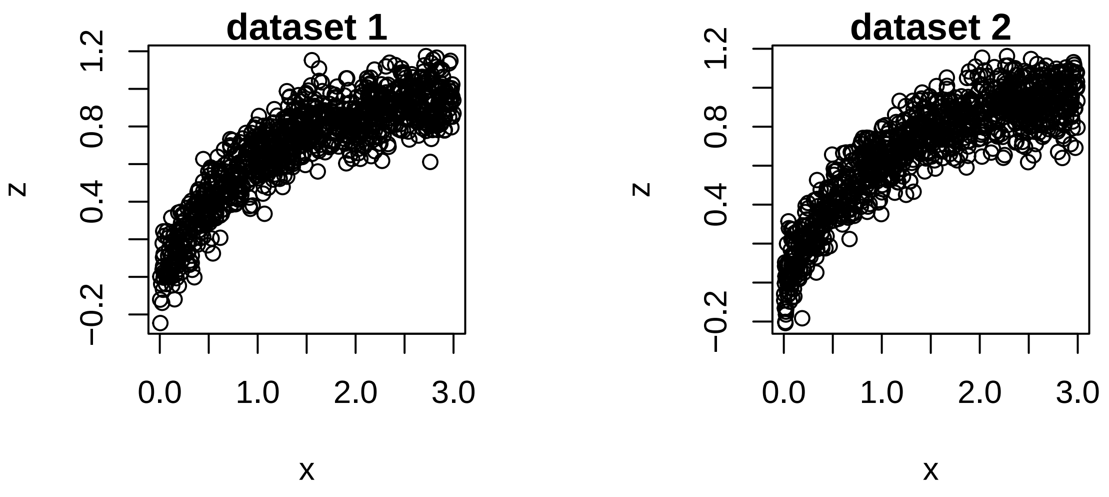


Figure 2: The first two data sets generated from a noisy negatively exponential relationship.

German Forestry Data: `x_r3`, `z_Fs`, `z_Q`, `z_Pa`, `z_Ps`

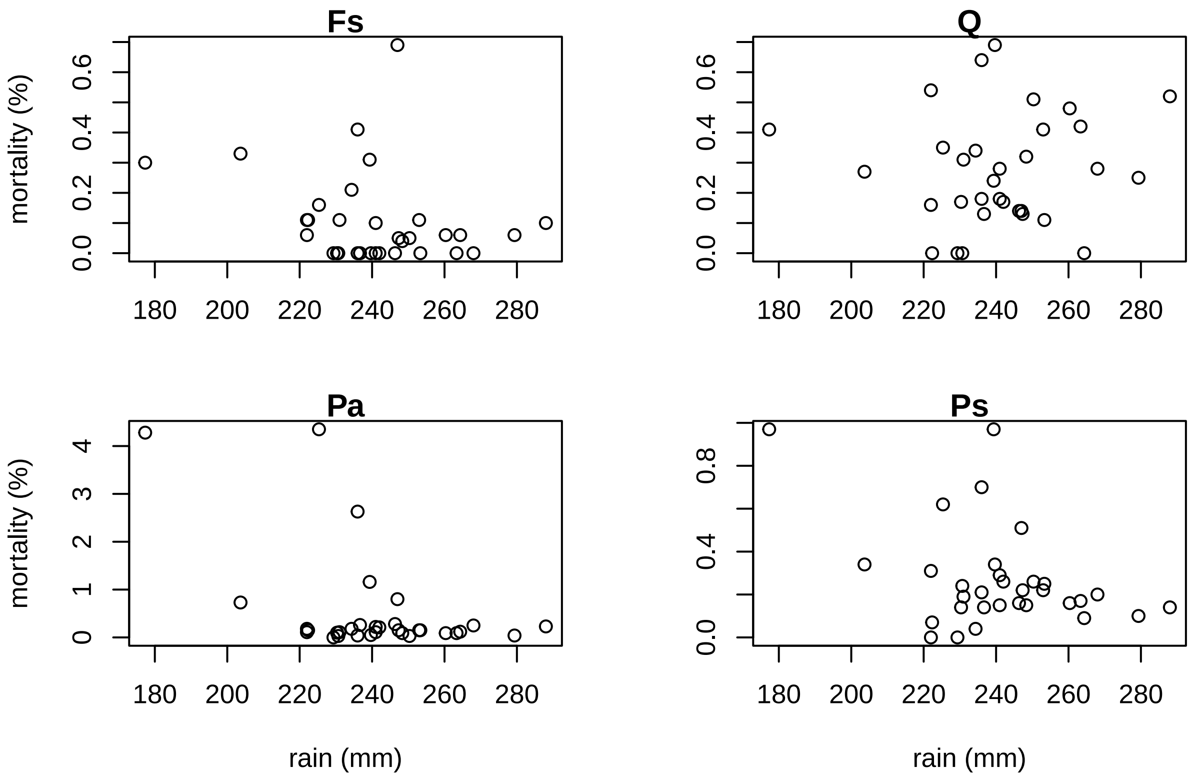
```
  d.data        <- "data_Germany/"

  file.data     <- paste0( d.data, "2-bis-7_abb-tab_nsh_2024-04-08.xlsx" )
  sheet.data    <- "5_Daten"
  r.data        <- "B120:C153"
  y_r           <- read_excel( file.data, sheet=sheet.data, range=r.data, col_names=F )
> New names:
> * `` -> `...1`
> * `` -> `...2`
  y_r           <- round( as.matrix(y_r) )
  colnames(y_r) <- c( "year", "rain.mm" )

  file.data     <- paste0( d.data, "absterberate_EI_GFI_GKI_RBU_zeitreihe.csv" )
  y_m           <- read.csv2( file.data )
  colnames(y_m) <- c( "year", "mort.%_Fs", "mort.%_Q", "mort.%_Pa", "mort.%_Ps" )
  y_r_m         <- cbind( y_r, y_m[ , startsWith( colnames(y_m), "mort" ) ] )

  # Annual mortality vs. summer rain of last three years
  meanlast <- function(v,k){as.vector(filter(v,f=rep(1/k,k),s=1))}
  n   <- dim(y_r_m)[1] ; k <- 3
  x_r3 <- meanlast(y_r_m[,"rain.mm"],k)[k:n]

  par ( mfrow=c(2,2), mar=c(4,4,1,2) )
  z_Fs <- y_r_m[k:n,"mort.%_Fs"]
  plot( x_r3, z_Fs, main="Fs", xlab=""         , ylab="mortality (%)" )
  z_Q <- y_r_m[k:n,"mort.%_Q"]
  plot( x_r3, z_Q , main="Q" , xlab=""         , ylab=""              )
  z_Pa <- y_r_m[k:n,"mort.%_Pa"]
  plot( x_r3, z_Pa, main="Pa", xlab="rain (mm)", ylab="mortality (%)" )
  z_Ps <- y_r_m[k:n,"mort.%_Ps"]
  plot( x_r3, z_Ps, main="Ps", xlab="rain (mm)", ylab=""              )
```



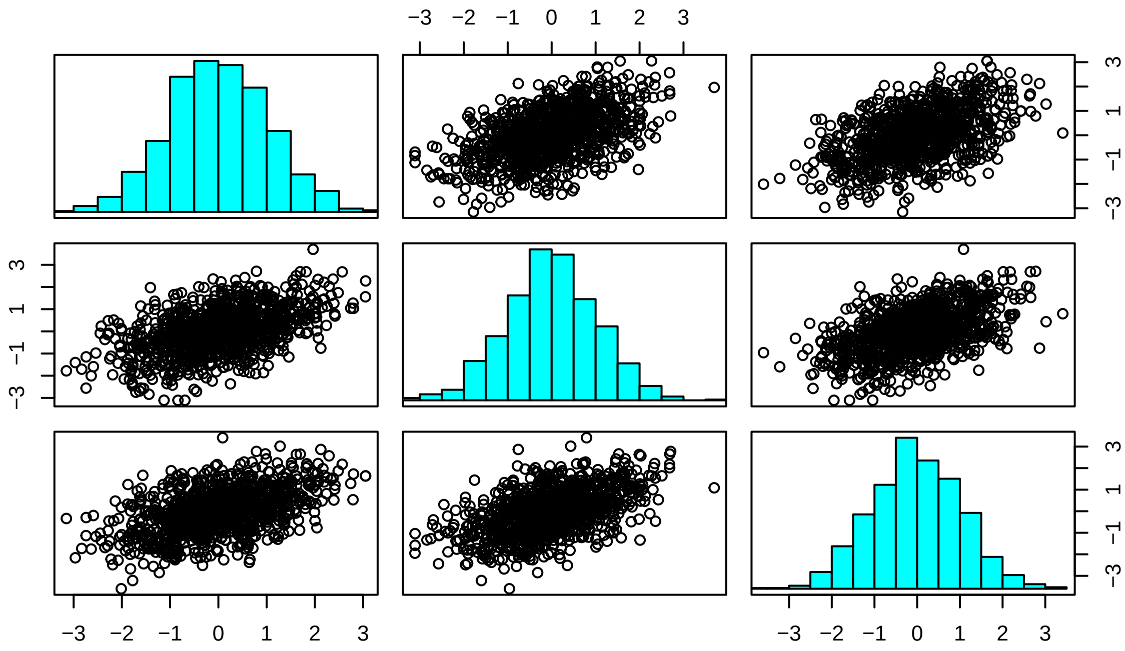
Trivariate Gaussian Dataset

```
panel.hist <- function(x, ...) {
  h      <- hist(x, plot=F)
  breaks <- h$breaks; n_b <- length(breaks)
  y      <- h$counts; y   <- min(x) + (max(x)-min(x)) * y/max(y)
  rect(breaks[-n_b], min(x), breaks[-1], y, col="cyan", ...) }
```

```
set.seed(1)

n_G3  <- 1e3 ; Sigma_G3 <- matrix(0.5,nrow=3,ncol=3) ; diag(Sigma_G3) <- 1
xz_G3 <- rmvnorm( n_G3, c(0,0,0), Sigma_G3 )

pairs(xz_G3, diag.panel=panel.hist, label="" ) ; cor(xz_G3)
```



```
>             [,1]      [,2]      [,3]
> [1,] 1.0000000 0.4993916 0.4933134
> [2,] 0.4993916 1.0000000 0.5190580
> [3,] 0.4933134 0.5190580 1.0000000
```

Introduction to Probabilistic Risk Analysis (PRA)

From risk matrices to PRA

Table 1: Typical risk matrix.

Consequence | Probability	1 (unlikely)	2 (medium)	3 (likely)
1 (minor)	1	2	3
2 (medium)	2	4	6
3 (major)	3	6	9

Basic equations for PRA

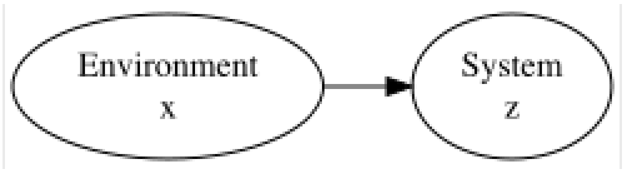


Figure 3: PRA as a probabilistic network.

$$R = E[z|\neg H] - E[z], \tag{1}$$

where $\neg H$ stands for non-hazardous environmental conditions.

$$\begin{aligned} V &= R\,/\,p[H] \\ &= (E[z|\neg H] - E[z])\,/\,p[H] \\ &= (E[z|\neg H] - p[H]E[z|H] - (1-p[H])E[z|\neg H])\,/\,p[H] \\ &= E[z|\neg H] - E[z|H]. \end{aligned} \tag{2}$$

Decomposition of risk: 2 or 3 components

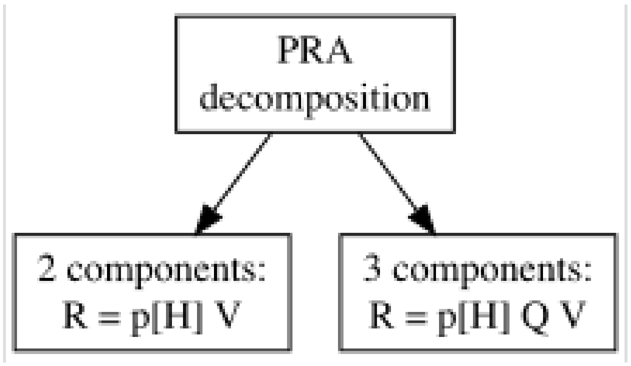


Figure 4: PRA-classification according to treatment of the hazardous region.

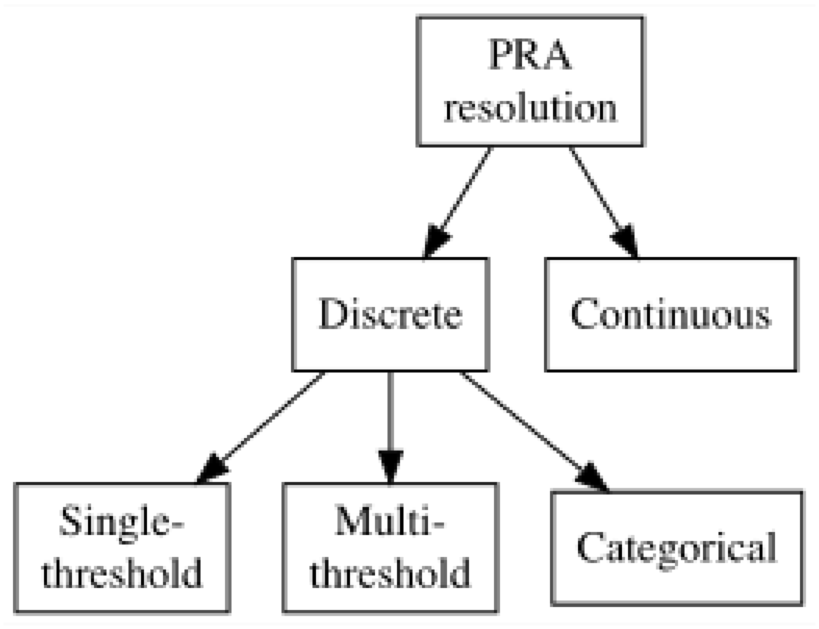


Figure 5: PRA-classification according to treatment of the hazardous region.

Resolution of PRA: single-threshold, multi-threshold, categorical, continuous

Single-threshold PRA

$$\begin{aligned} V &= E[z|x \geq thr] - E[z|x < thr], \\ R &= E[z|x \geq thr] - E[z] \\ &= p[H]\,V.(\#eq:PRAthr1) \end{aligned} \tag{3}$$

Multi-threshold PRA

$$\begin{aligned} p[H_i] &= p[thr_{i-1} \leq x < thr_i], \\ V_i &= E[z|\neg H] - E[z|H_i] \\ &= E[z|x \geq thr_n] - E[z|thr_{i-1} \leq x < thr_i], \\ R_i &= p[H_i]\,V_i, (\#eq:PRAthrnB) \end{aligned} \tag{4}$$

for $i = 1, .., n$ and where $thr_0 = -\infty$.

$$\begin{aligned} p[H] &= \sum p[H_i], \\ V &= \sum \frac{p[H_i]}{p[H]} V_i, \\ R &= \sum R_i, \\ &= p[H]\,V, (\#eq:PRAthrnA) \end{aligned} \tag{5}$$

with all summations running from $i = 1$ to n.

Categorical PRA

Continuous PRA

$$R = \int_{x=-\infty}^{thr} r(x) dx, (\#eq : PRAcontA) \tag{6}$$

where

$$\begin{aligned} r(x) &= p[x]\, v(x), \\ v(x) &= E[z|x \geq thr] - E[z|x].(\#eq : PRAcontB) \end{aligned} \tag{7}$$

Implementation of PRA: distribution-based, sampling-based, model-based

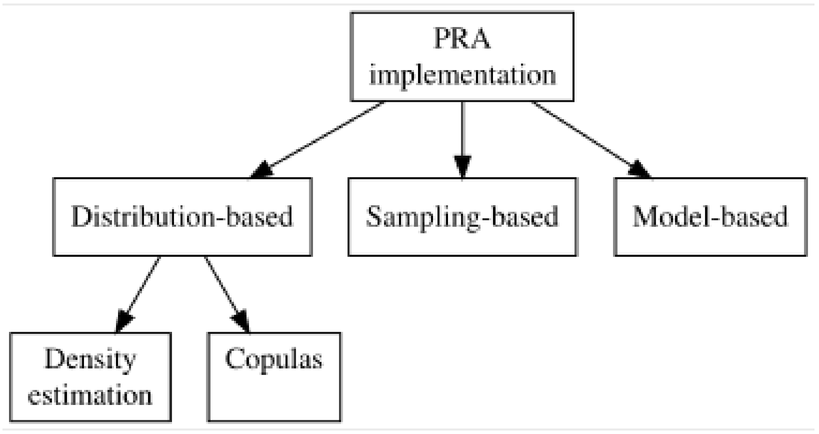


Figure 6: PRA-classification according to implementation method.

Distribution-based single-threshold PRA

Conditional distributions for z

$$\begin{aligned} p[z|x<thr] &= \frac{p[z,x<thr]}{p[x<thr]} \\ &= \frac{p[z]\int_{x=-\infty}^{thr} p[x|z]\,dx}{p[x<thr]} \\ &= \frac{p[z]F_{x|z}[thr]}{F_x[thr]}, \\ p[z|x\geq thr] &= \frac{p[z](1-F_{x|z}[thr])}{1-F_x[thr]}, (\#eq:pzGxbelowabove) \end{aligned} \tag{8}$$

where $F_x[thr]$ and $F_{x|z}[thr]$ are the cumulative distribution functions associated with $p[x]$ and $p[x|z]$, both evaluated at $x = thr$. Note that $F_x[thr] = p[x < thr]$ is the hazard probability $p[H]$.

$$\begin{aligned} E[z|x<thr] &= \int_{z=-\infty}^{\infty} z\,p[z|x<thr]\,dz \\ &= \frac{1}{F_x[thr]}\int_{z=-\infty}^{\infty} z\,p[z]\,F_{x|z}[thr]\,dz, \\ E[z|x\geq thr] &= \frac{1}{1-F_x[thr]}\int_{z=-\infty}^{\infty} z\,p[z]\,(1-F_{x|z}[thr])\,dz.(\#eq:EzGxbelowaboveA) \end{aligned} \tag{9}$$

$$\begin{aligned} E[z|x<thr] &= \frac{1}{F_x[thr]}\int_{x=-\infty}^{thr} p[x]\,E[z|x]\,dx, \\ E[z|x\geq thr] &= \frac{1}{1-F_x[thr]}\int_{x=thr}^{\infty} p[x]\,E[z|x]\,dx.(\#eq:EzGxbelowaboveB) \end{aligned} \tag{10}$$

Conditions for V being constant

$$\begin{aligned} \frac{dV}{d\,thr} &= \frac{d\,(E[z|x\geq thr]-E[z|x<thr])}{d\,thr} \\ &= p[thr]\,\{\frac{E[z|x\geq thr]-z[thr]}{1-F_x[thr]} - \frac{z[thr]-E[z|x<thr]}{F_x[thr]}\} \\ &= \frac{p[thr]}{F_x[thr](1-F_x[thr])}\,\{E[z|x\geq thr]F_x[thr]-z[thr]+E[z|x<thr](1-F_x[thr])\}, (\#eq:EzdVdthr) \end{aligned} \tag{11}$$

where $z[thr] = E[z|x = thr]$.

Example of distribution-based PRA: Gaussian $p[x, z]$

$$\begin{aligned} &\underline{\text{EXAMPLE: Bivariate Gaussian for } x \text{ and } z\text{:}} \\ &p[x,z] = N[\mu,\Sigma], \text{where} \\ &\mu = \begin{bmatrix}\mu_x\\ \mu_z\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix};\quad \Sigma = \begin{bmatrix}\sigma_x^2 & \rho\sigma_x\sigma_z\\ \rho\sigma_x\sigma_z & \sigma_z^2\end{bmatrix} = \begin{bmatrix}1 & 0.5\\ 0.5 & 1\end{bmatrix}.(\#eq:pxzGaussian) \end{aligned} \tag{12}$$

$$
\begin{aligned}
p[x] &= N[\mu_x, \sigma_x^2] = N[0,1], \\
p[z] &= N[\mu_z, \sigma_z^2] = N[0,1], \\
p[z|x] &= N[\mu_z + \rho(x-\mu_x)\frac{\sigma_z}{\sigma_x}, \sigma_z^2(1-\rho^2)] = N[0.5x, 0.75].
\end{aligned} \tag{13}
$$

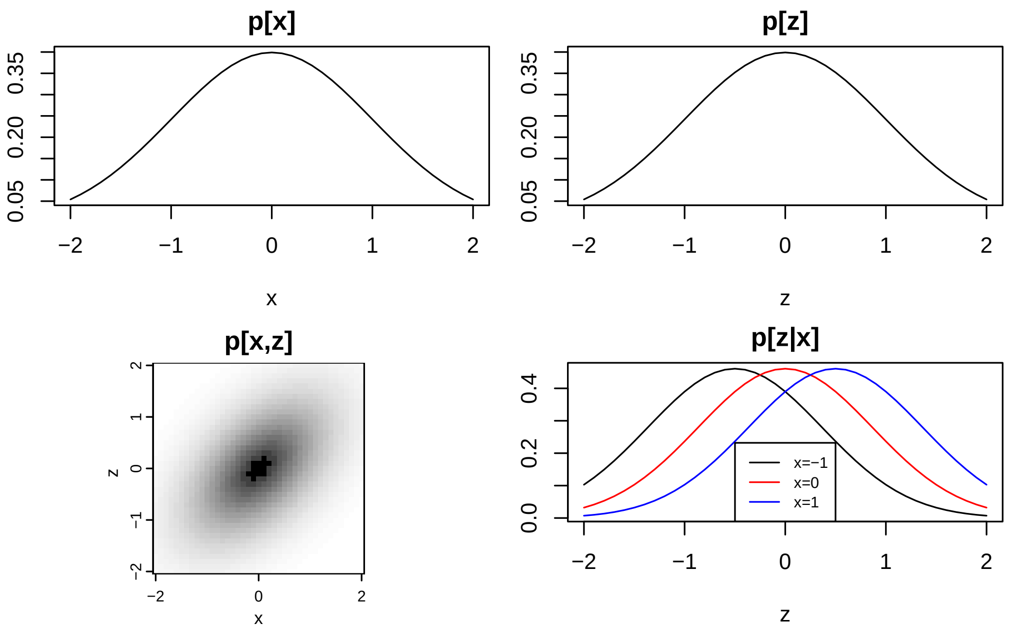


Figure 7: Marginal, joint, and conditional Gaussian distributions for x and z.

Hazard probability and conditional distributions

Conditional expectations and PRA

```r
Ez_xlo_NI <- function(thr=0, mz.=mz, mx.=mx, Vz.=Vz, Vx.=Vx, r=rxz) {
  x.seq    <- seq( thr-5*sqrt(Vx.), thr, length.out=101 )
  px.seq   <- px( x.seq, m=mx., V=Vx. ) ; px.sum <- sum(px.seq)
  Ez_x.seq <- mz. + r * (x.seq-mx.) * sqrt(Vz./Vx.)
  Ezxbelow <- sum(px.seq * Ez_x.seq) / px.sum
  return( Ezxbelow ) }
```

Approximation formulas for the conditional bivariate Gaussian expectations

$$
\begin{aligned}
&\underline{\text{APPRXOXIMATION FOR BIVARIATE GAUSSIAN p[x,z]:}} \\
&E[z|x < thr] \approx E[z] - \rho\,\sigma_x\,\sigma_z\,\frac{p_x[thr]}{F_x[thr]}, \\
&E[z|x \geq thr] \approx E[z] + \rho\,\sigma_x\,\sigma_z\,\frac{p_x[thr]}{1-F_x[thr]}.(\#eq:EzcondApproxGaussian)
\end{aligned} \tag{14}
$$

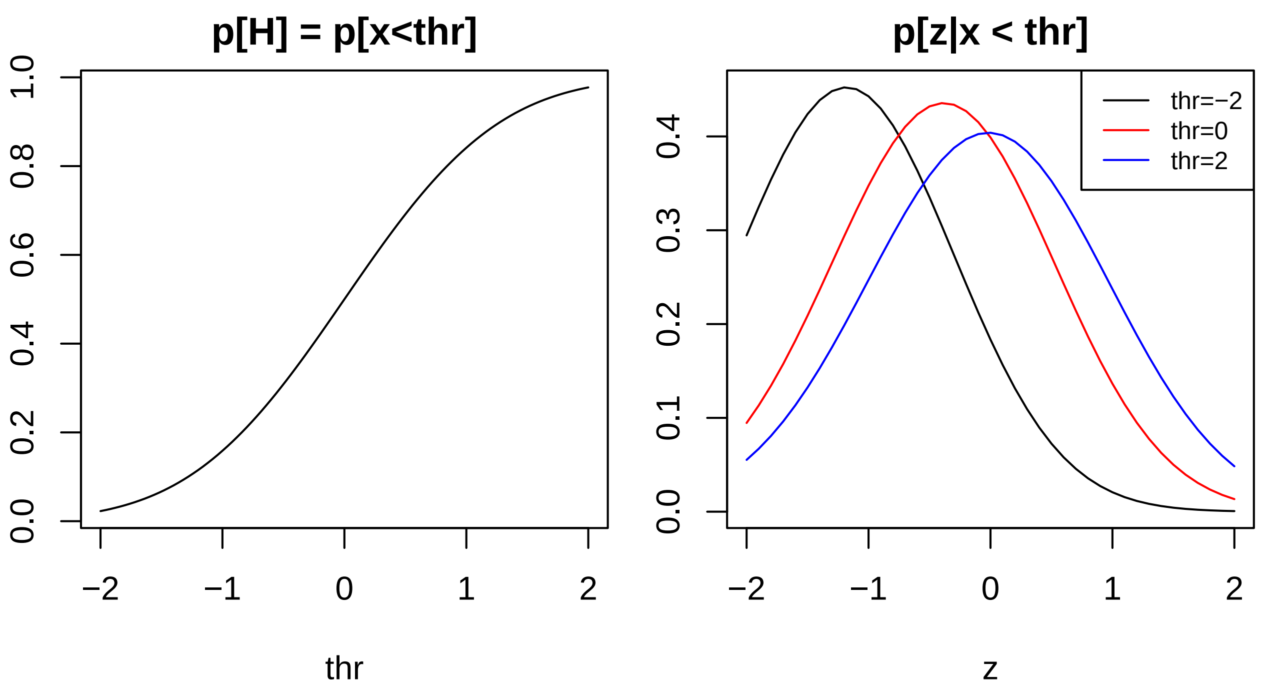


Figure 8: $p[H]$ and $p[z|x < thr]$ for the bivariate Gaussian.

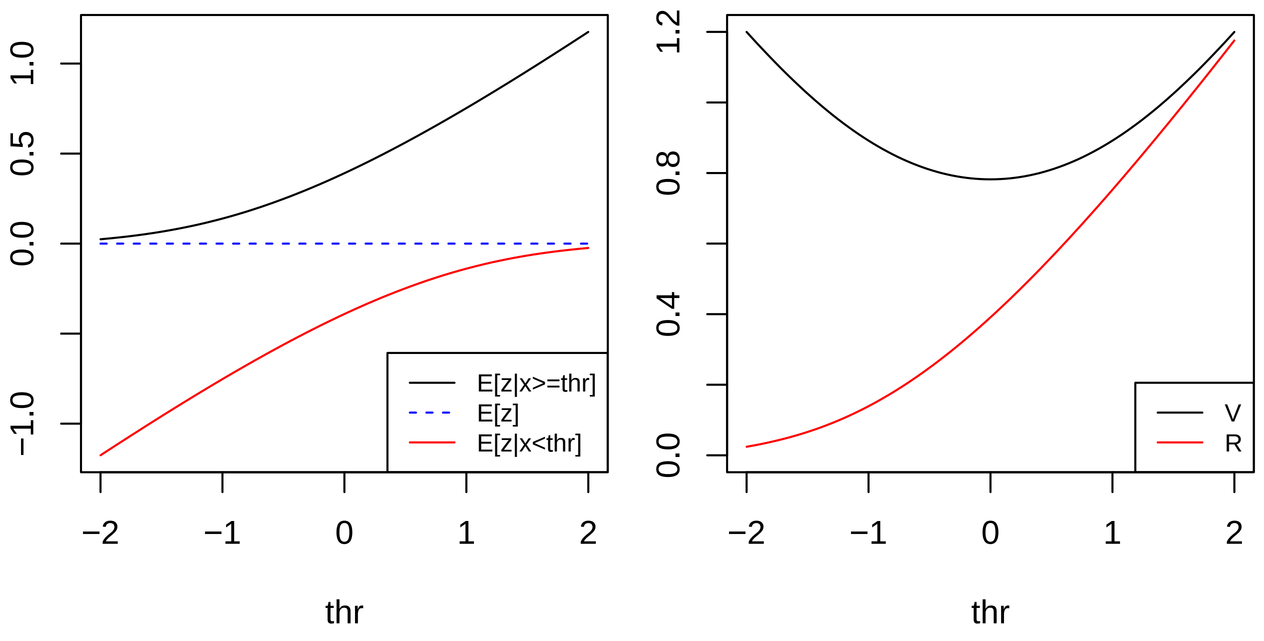


Figure 9: Distribution-based single-threshold PRA: bivariate Gaussian example. Left: conditional expectations for z as a function of threshold-value. Right: the corresponding values of V and R.

```
EzVz_Gauss <- function( m.=m, S.=S, thr.=thr ) {
  mx     <- m.[1]   ; mz <- m.[2]
  Vx     <- S.[1,1] ; Vz <- S.[2,2]   ; Vxz  <- S.[1,2] ; r <- Vxz/sqrt(Vx*Vz)
  pthr   <- dnorm(thr., mx, sqrt(Vx)) ; Fthr <- pnorm(thr., mx, sqrt(Vx))
  Ez_xlo <- mz - Vxz * pthr / Fthr
  Ez_xhi <- mz + Vxz * pthr / (1-Fthr)
  Vz_xlo <- Vz + r * (thr.-mx) * (Ez_xlo-mz) - (Ez_xlo-mz)^2
  Vz_xhi <- Vz + r * (thr.-mx) * (Ez_xhi-mz) - (Ez_xhi-mz)^2
  result <- c( Ez_xlo, Ez_xhi, Vz_xlo, Vz_xhi )
  names ( result ) <- c( "Ez_xlo", "Ez_xhi", "Vz_xlo", "Vz_xhi" )
  return( result ) }
```

```
xz <- l_xz.L[[1]] ; m <- colMeans(xz)            ; mx <- m[1]   ; mz <- m[2]
S  <- var(xz)     ; r <- S[1,2] / sqrt(Vx * Vz) ; Vx <- S[1,1] ; Vz <- S[2,2]
```

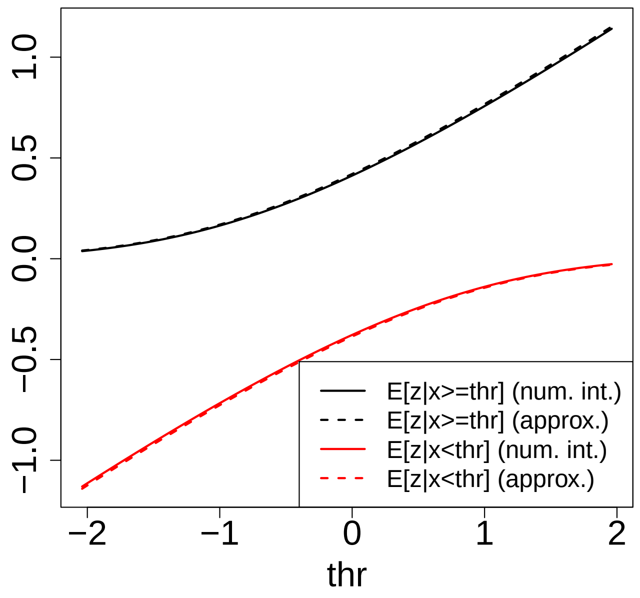


Figure 10: Conditional expectations for z as a function of threshold-value in the case of bivariate Gaussian $p[x, z]$. Solid lines: Numerical integration. Dashed lines: approximation formulas.

$$
\begin{aligned}
&\underline{\text{APPROXIMATE PRA FOR BIVARIATE GAUSSIAN p[x,z]:}}\\
&p[H] = F_x[thr],\\
&V \approx \frac{\rho\,\sigma_x\,\sigma_z\,p_x[thr]}{F_x[thr]\,(1 - F_x[thr])},\\
&R \approx \frac{\rho\,\sigma_x\,\sigma_z\,p_x[thr]}{1 - F_x[thr]}.(\#eq: PRAapproxGaussian)
\end{aligned} \tag{15}
$$

Sampling-based single-threshold PRA

$$\begin{aligned}
&\underline{\text{SAMPLE MEANS:}} && \\
&\hat{E}[z] &&= \overline{z}, \\
&\hat{E}[z|x < thr] &&= \overline{z_H}, \\
&\hat{E}[z|x \geq thr] &&= \overline{z_{\neg H}}.(\#eq:Esample)
\end{aligned} \tag{16}$$

```
PRA0 <- function( x, z, thr=0 ) {
  n    <- length(x)      ;  H       <- which(x < thr)  ;  n_H <- length(H)
  Ez_H <- mean( z[ H] )  ;  Ez_notH <- mean( z[-H] )
  pH   <- n_H / n        ;  V       <- Ez_notH - Ez_H  ;  R   <- pH * V
  return( c( pH=pH, V=V, R=R ) ) }
```

Example of sampling-based PRA: linear relationship

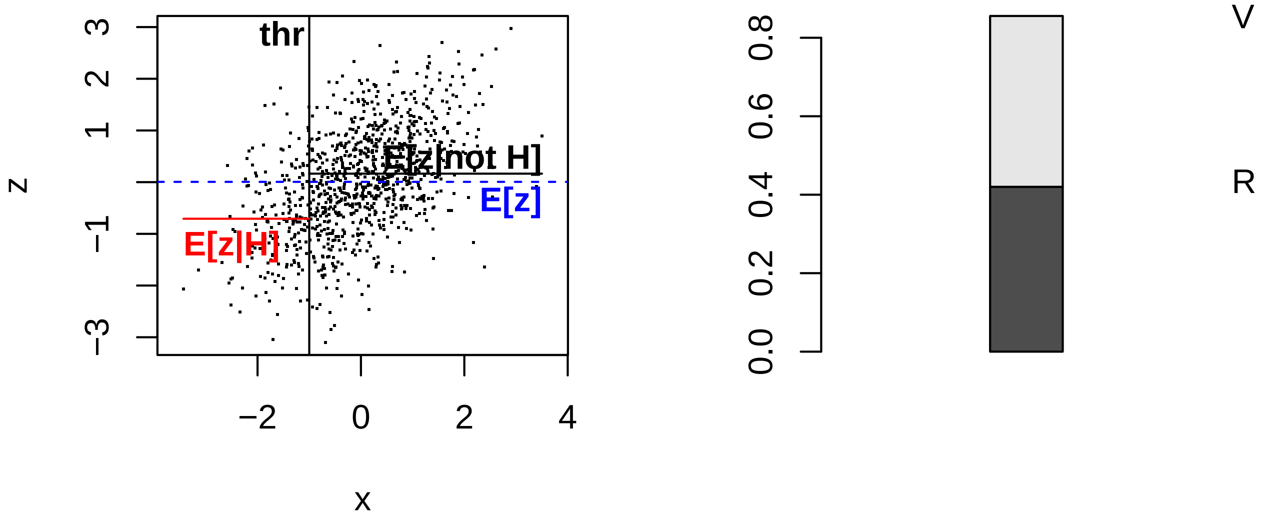


Figure 11: Sampling-based single-threshold PRA on 'data set 1' with thr=0. Left: data and expectation values. Right: V and R.

Varying the threshold

Example of sampling-based PRA: nonlinear relationship

$$\begin{aligned}
&\underline{\text{NONLINEAR RELATIONSHIP BETWEEN x AND z:}} \\
&z|x \sim N[f(x), \sigma_z^2], \quad x \geq 0, where \\
&f(x) = 1 - exp(-x).(\#eq:negexp)
\end{aligned} \tag{17}$$

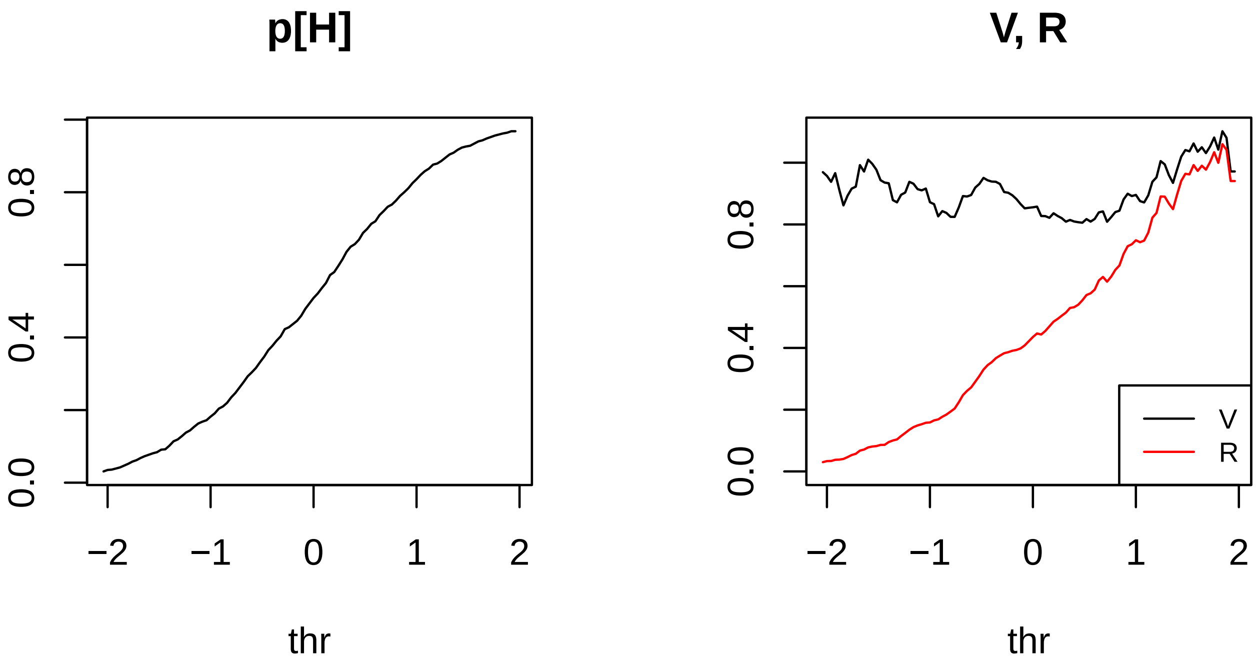


Figure 12: Sampling-based single-threshold PRA on data set 1 for a range of different thresholds. Left: $p[H]$. Right: V and R.

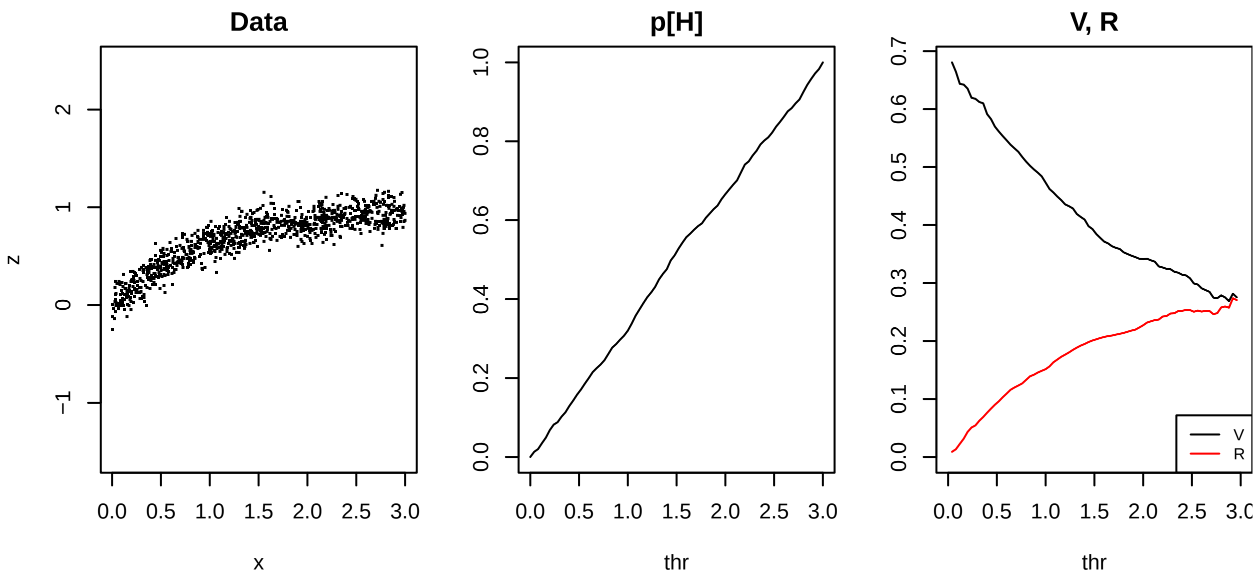


Figure 13: PRA on a data set from a nonlinear relationship.

Sampling-based single-threshold PRA: Uncertainty quantification (UQ)

$$\begin{aligned}
&\underline{\text{STANDARD DEVIATIONS OF SAMPLE MEANS:}}\\
\sigma_{\hat{E}[z|x<thr]} &= \sqrt{\frac{Var[z_H]}{n_H}},\\
\sigma_{\hat{E}[z|x\geq thr]} &= \sqrt{\frac{Var[z_{\neg H}]}{n-n_H}}.(\#eq:sigmasample)
\end{aligned} \tag{18}$$

Uncertainty in $p[H]$

$$\begin{aligned}
&\underline{\text{SAMPLING UNCERTAINTY FOR } p[H]:}\\
&\sigma_{p[H]} = \sqrt{p[H]\,(1-p[H])/n}, (\#eq:sigmapH)
\end{aligned} \tag{19}$$

$$\begin{aligned}
&\underline{\text{BAYESIAN POSTERIOR UNCERTAINTY FOR } p[H]:}\\
&p[H] \sim Be[a,b] \implies\\
&\sigma_{p[H]} = \frac{1}{a+b}\sqrt{\frac{ab}{a+b+1}}, (\#eq:sigmapHBayes)
\end{aligned} \tag{20}$$

where $a = 1 + n_H$ and $b = 1 + n - n_H$.

Uncertainty in V

$$\begin{aligned}
&\underline{\text{SAMPLING UNCERTAINTY FOR } V:}\\
&\sigma_V = \sqrt{\sigma^2_{\hat{E}[z|x\geq thr]} + \sigma^2_{\hat{E}[z|x<thr]}}, (\#eq:sigmaV)
\end{aligned} \tag{21}$$

where $\sigma_{\hat{E}[z|x\geq thr]}$ and $\sigma_{\hat{E}[z|x<thr]}$ are as defined in Eq. @ref(eq:sigmasample).

Uncertainty in R

$$\begin{aligned}
&\underline{\text{SAMPLING UNCERTAINTY FOR } R:}\\
&\sigma_R = \sqrt{\sigma^2_{p[H]}\sigma^2_V + \sigma^2_V p[H]^2 + \sigma^2_{p[H]} V^2}.(\#eq:sigmaR)
\end{aligned} \tag{22}$$

Extension of R-code for PRA: adding the UQ

```r
PRA <- function( x, z, thr=0 ) {
  n       <- length(x)     ; H         <- which(x < thr) ; n_H <- length(H)
  Ez_H    <- mean( z[ H] ) ; s_Ez_H    <- sqrt( var(z[ H]) /    n_H  )
  Ez_notH <- mean( z[-H] ) ; s_Ez_notH <- sqrt( var(z[-H]) / (n-n_H) )
  pH      <- n_H / n       ; V         <- Ez_notH - Ez_H ; R  <- pH * V
  s_pH    <- sqrt( pH*(1-pH) / n )
  s_V     <- sqrt( s_Ez_H^2 + s_Ez_notH^2 )
  s_R     <- sqrt( s_pH^2*s_V^2 + s_pH^2*V^2 + pH^2*s_V^2 )
  return( c(pH=pH,V=V,R=R,s_pH=s_pH,s_V=s_V,s_R=s_R) )
}
```

PRA with UQ on the nonlinear data set

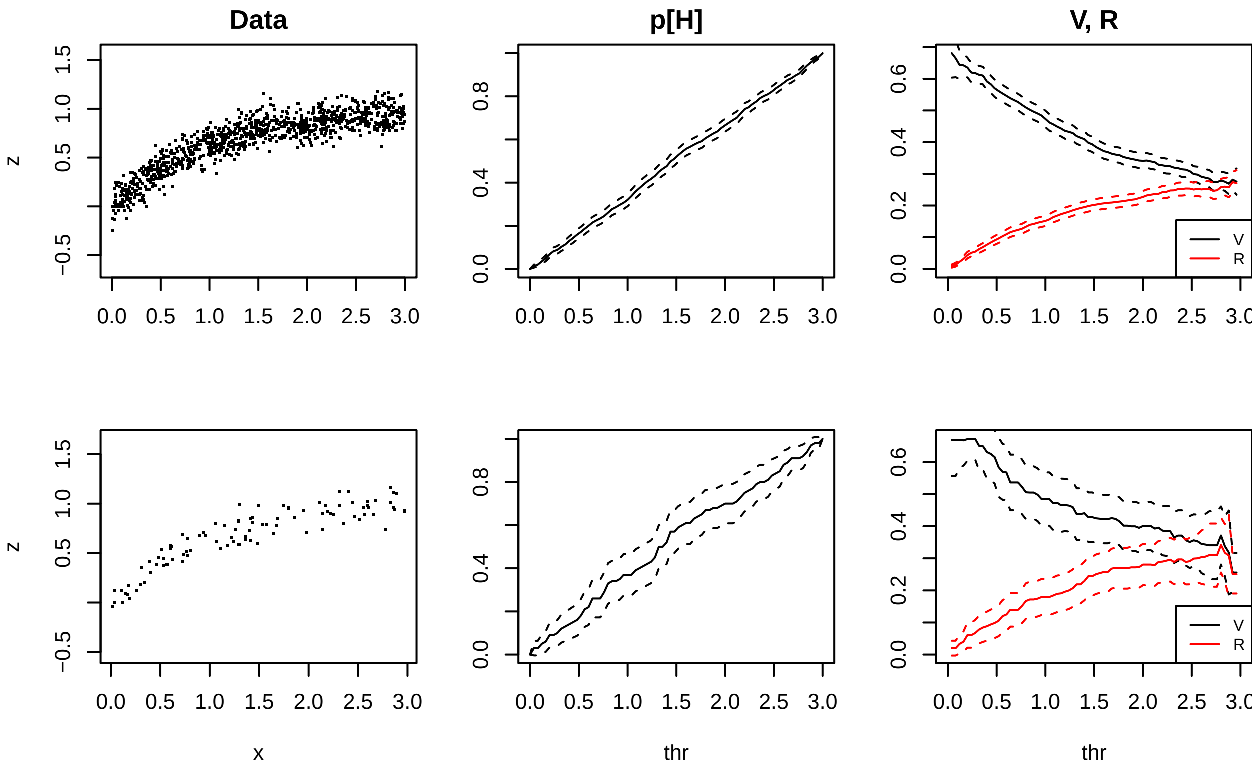


Figure 14: PRA on nonlinearly related x and z, with UQ. Top row: as previous example. Bottom row: same but just 10% of data being used. Dashed lines are 2 standard deviations away from the mean.

Verification of the UQ by simulating multiple data sets

UQ-verification: Nonlinear relationship

```
thr <- 1

PRA.tbl <- t( sapply( 1:length(l_xz.NL), function(d) {
  PRA( l_xz.NL[[d]][,1], l_xz.NL[[d]][,2], thr) } ) )

PRA.tbl[1:3,]
>         pH         V         R       s_pH        s_V         s_R
> [1,] 0.320 0.4732057 0.1514258 0.01475127 0.01298911 0.008126446
> [2,] 0.344 0.5008567 0.1722947 0.01502212 0.01328335 0.008805076
> [3,] 0.311 0.4681190 0.1455850 0.01463827 0.01250033 0.007880549
```

```
slist_pH <- sd( PRA.tbl[,"pH"] )
slist_V  <- sd( PRA.tbl[,"V" ] )
slist_R  <- sd( PRA.tbl[,"R" ] )
```

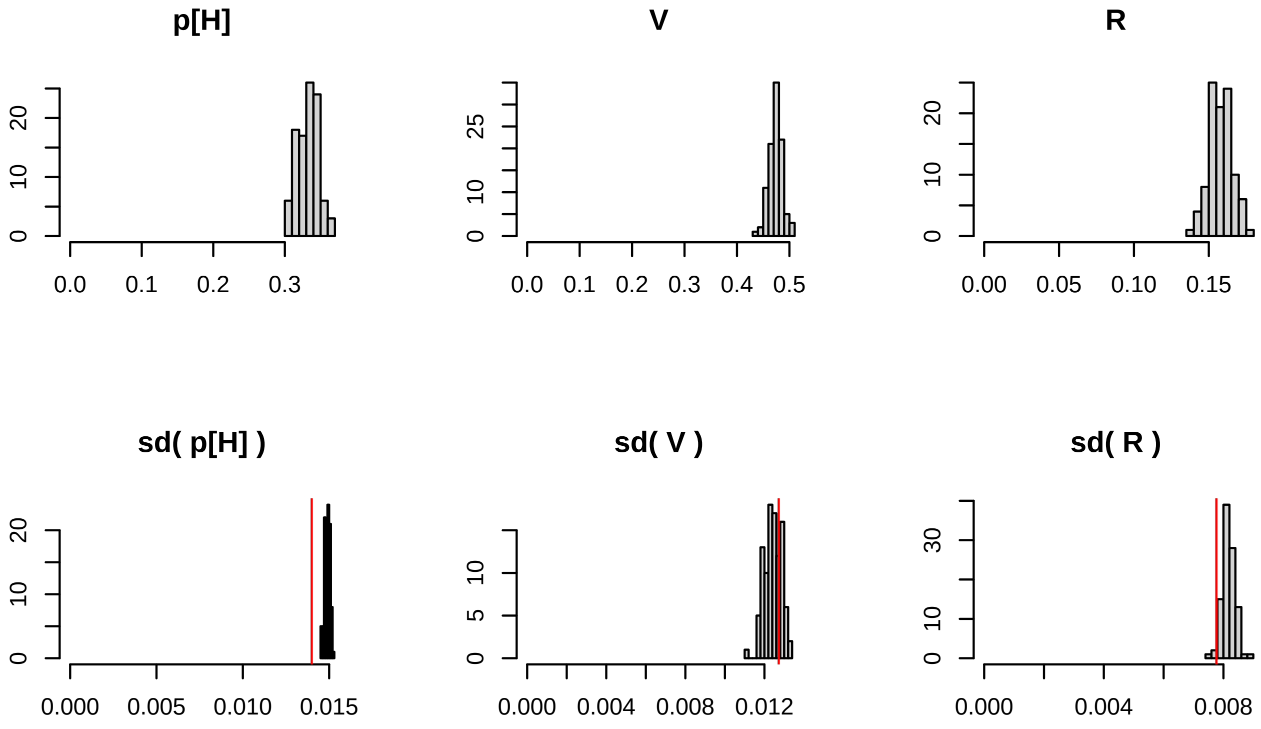


Figure 15: PRAs on 100 realisations (virtual data sets) from the same nonlinear relationship between x and z. Top row: distributions of the 100 estimates for $p[H]$, V and R. Bottom row: distributions of the 100 uncertainty estimates (sigma-values). The vertical red lines indicate the standard deviations of the top-row estimates (widths of top-row histograms).

UQ-verification: Linear relationship

```
>         pH         V         R       s_pH        s_V        s_R
> [1,] 0.182 0.8719316 0.1586916 0.01220148 0.07483537 0.01730677
> [2,] 0.143 1.0849957 0.1551544 0.01107028 0.07795630 0.01640994
> [3,] 0.166 0.9192960 0.1526031 0.01176622 0.07231166 0.01618065
```

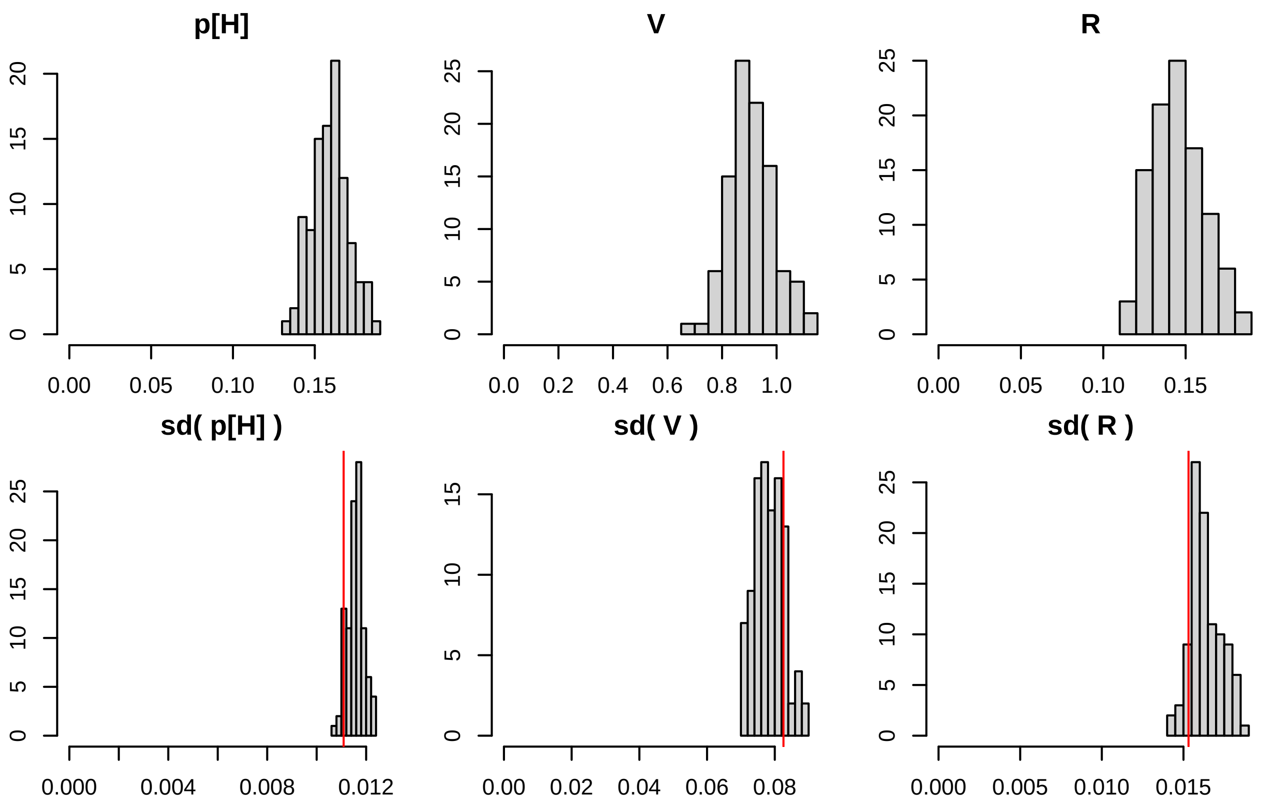


Figure 16: PRAs on 1000 realisations from the same linear relationship between x and z. Top row: 1000 estimates for $p[H]$, V and R. Bottom row: 1000 uncertainty estimates. Vertical red lines indicate standard deviations of the top-row estimates.

Approximation formulas for the conditional bivariate Gaussian variances

APPRXOXIMATION FOR BIVARIATE GAUSSIAN p[x,z]:

$$Var[z|x < thr] \approx \sigma_z^2 + \rho\,(thr - E[x])\,(\mu_1 - E[z]) - (\mu_1 - E[z])^2, \tag{23}$$

$$Var[z|x \geq thr] \approx \sigma_z^2 + \rho\,(thr - E[x])\,(\mu_2 - E[z]) - (\mu_2 - E[z])^2, (\#eq : VarzcondApproxGaussian)$$

where $\mu_1 = E[z|x < thr]$ and $\mu_2 = E[z|x \geq thr]$ for which the approximations were given in Eq. @ref(eq:EzcondApproxGaussian).

```
PRA0_Gauss <- function( m.=m, S.=S, thr.=thr ) {
  mx <- m.[1] ; sx <- sqrt(S.[1,1]) ; Vxz <- S.[1,2]
  pH <- pnorm( thr., mx, sx )
  V  <- Vxz * dnorm(thr., mx, sx) / (pH * (1-pH))
  R  <- pH * V
  return( c( pH=pH, V=V, R=R ) ) }
```

```
PRA_Gauss <- function( m.=m, S.=S, n.=n, thr.=thr ) {
  pH <- V <- R <- rep( NA, 1e3 )
  for(j in 1:1e3){
    S     <- riwish( n.-1, S. * (n.-1) ) ; m <- rmvnorm( 1, m., S/n. )
    PRA   <- PRA0_Gauss( m, S, thr. )
    pH[j] <- PRA["pH"] ; V[j] <- PRA["V"] ; R[j] <- PRA["R"]
  }
  return( c( pH=mean(pH),   V=mean(V),   R=mean(R),
           s_pH=sd  (pH), s_V=sd  (V), s_R=sd  (R) ) )
}
```

```
>           pH            V            R         s_pH          s_V          s_R
> 0.1588730000 0.9077771702 0.1442212824 0.0003655576 0.0024716987 0.0005141248
>           pH            V            R         s_pH          s_V          s_R
> 0.1586450576 0.9063497560 0.1437879427 0.0002921984 0.0016671928 0.0003869262
```

Density estimation to move from sampling- to distribution-based PRA

```
xz <- l_xz.L[[1]] ; m <- colMeans(xz) ; V <- var(xz)
```

```
mx     <- m[1]   ; mz <- m[2]
Vx     <- V[1,1] ; Vz <- V[2,2] ; rxz <- V[1,2] / sqrt(Vx * Vz)
thr    <- -1
Ez     <- mz
Ez_xlo <- Ez_xlo_NI(thr=thr, mz.=mz, mx.=mx, Vz.=Vz, Vx.=Vx, r=rxz)
Ez_xhi <- Ez_xhi_NI(thr=thr, mz.=mz, mx.=mx, Vz.=Vz, Vx.=Vx, r=rxz)
```

```
V  <- Ez_xhi - Ez_xlo
R  <- Ez_xhi - Ez
pH <- R / V
```

```
> PRA using density estimation and eqs for conditional expectations:
>  0.1803815 0.8810575 0.1589265
```

```
>         pH          V          R       s_pH        s_V        s_R
> 0.18200000 0.87193163 0.15869156 0.01220148 0.07483537 0.01730677
```

Copulas for distribution-based PRA

$$\underline{\text{SKLAR's THEOREM:}} \\ F_{xz}(x,z) = C(\,F_x(x),\ F_z(z)\,), (\#eq:Sklar) \tag{24}$$

Sampling from copulas and carrying out PRA

```
cpN    <- normalCopula( param=0.5, dim=2 )
mvN.NN <- mvdc( cpN, margins = c("norm", "norm"),
                paramMargins = list( list( mean=0, sd=1 ),
                                     list( mean=2, sd=1 ) ) )
```

```
n              <- 1e3
sample.cpN     <- rCopula( n, cpN )
         Fx.N <- sample.cpN[,1]     ; Fz.N    <- sample.cpN[,2]
sample.mvN.NN <- rMvdc    ( n, mvN.NN )
       x.N.NN <- sample.mvN.NN[,1] ; z.N.NN <- sample.mvN.NN[,2]
```

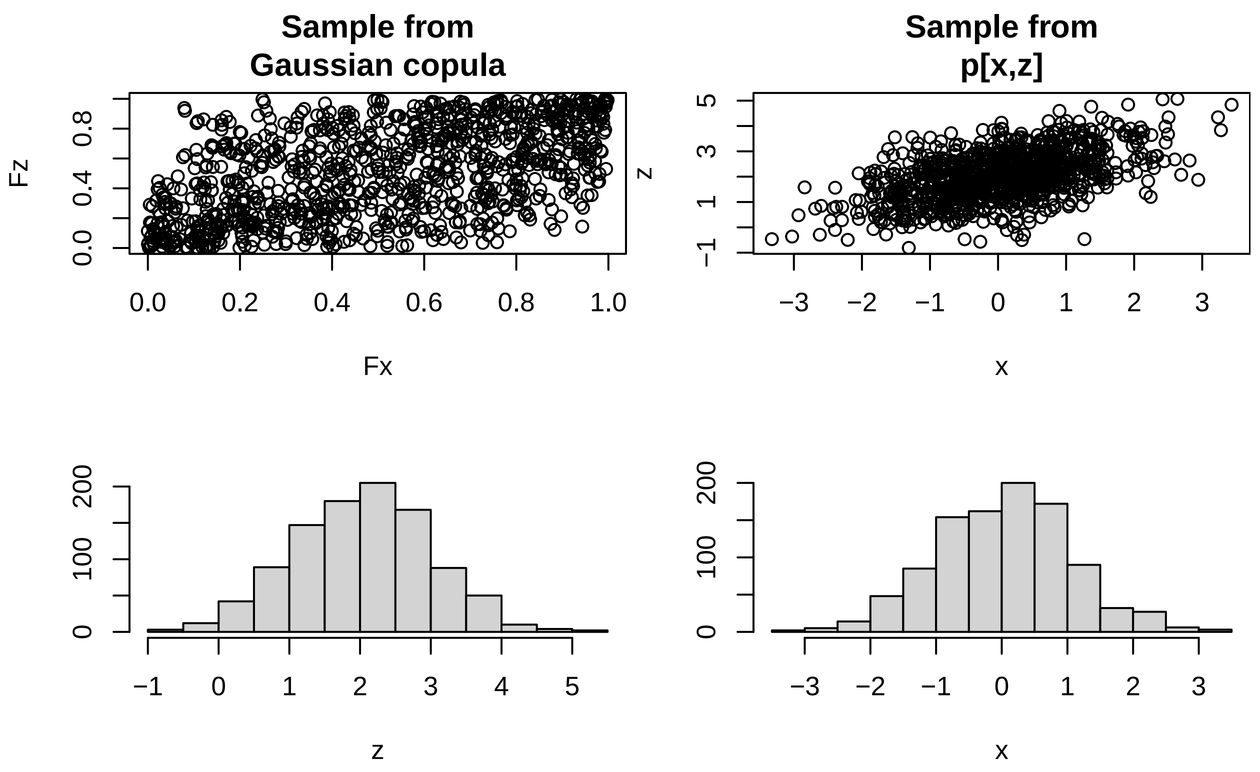


Figure 17: Top left: sample from a Gaussian copula with correlation parameter $\rho = 0.5$. Top right: sample from the joint distribution determined by this copula when the marginals for x and z are N[0,1] and N[2,1]. Bottom row: marginal samples.

```
mvN.NG        <- mvdc( cpN, margins = c("norm", "gamma"),
                       paramMargins = list( list( mean =0, sd   =1 ),
                                            list( shape=4, scale=2 ) ) )
```

```
sample.mvN.NG <- rMvdc( n, mvN.NG )
       x.N.NG <- sample.mvN.NG[,1] ; z.N.NG <- sample.mvN.NG[,2]
```

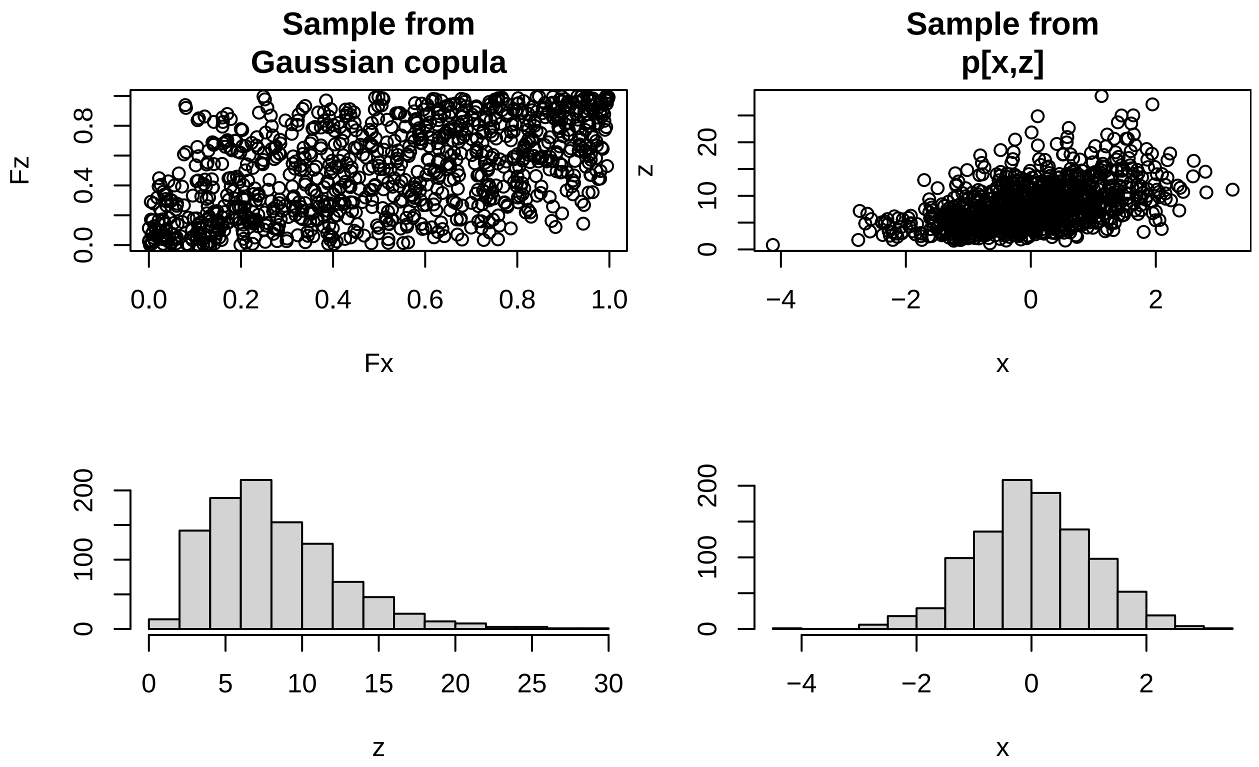


Figure 18: Sampling from a Gaussian copula with correlation parameter $\rho = 0.5$ combined with N[0,1] and Gamma[4,2] marginals for x and z.

```
>    pH     V     R  s_pH   s_V   s_R
> 0.150 0.880 0.140 0.011 0.077 0.016

>    pH     V     R  s_pH   s_V   s_R
> 0.150 3.300 0.500 0.011 0.260 0.055
```

```
  cpt <- tCopula( param=0.5, df=1 )
```

```
>    pH     V     R  s_pH   s_V   s_R
> 0.150 2.600 0.400 0.011 0.390 0.066
```

Copula selection

```
fitC <- function(x,z){ BiCopSelect( ecdf(x)(x)*n/(n+1), ecdf(z)(z)*n/(n+1),
                                    sel="BIC" ) }
fitC( x.N.NN, z.N.NN )
> Bivariate copula: Gaussian (par = 0.53, tau = 0.35)
fitC( x.N.NG, z.N.NG )
```

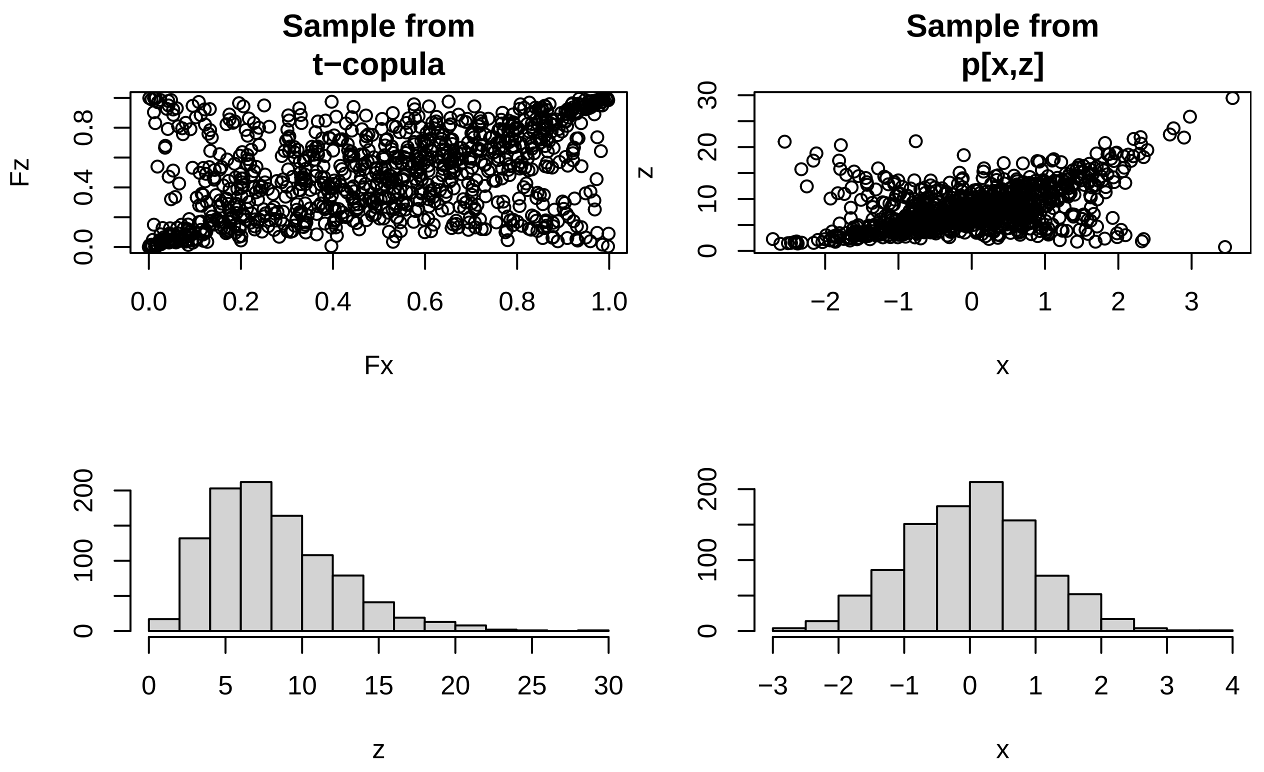


Figure 19: Sampling from a t-copula with correlation parameter $\rho = 0.5$ and $df = 1$ combined with N[0,1] and Gamma[4,2] marginals for x and z.

```
> Bivariate copula: Gaussian (par = 0.49, tau = 0.33)
fitC( x.t.NG, z.t.NG )
> Bivariate copula: t (par = 0.62, par2 = 2, tau = 0.42)
```

Using copulas in PRA

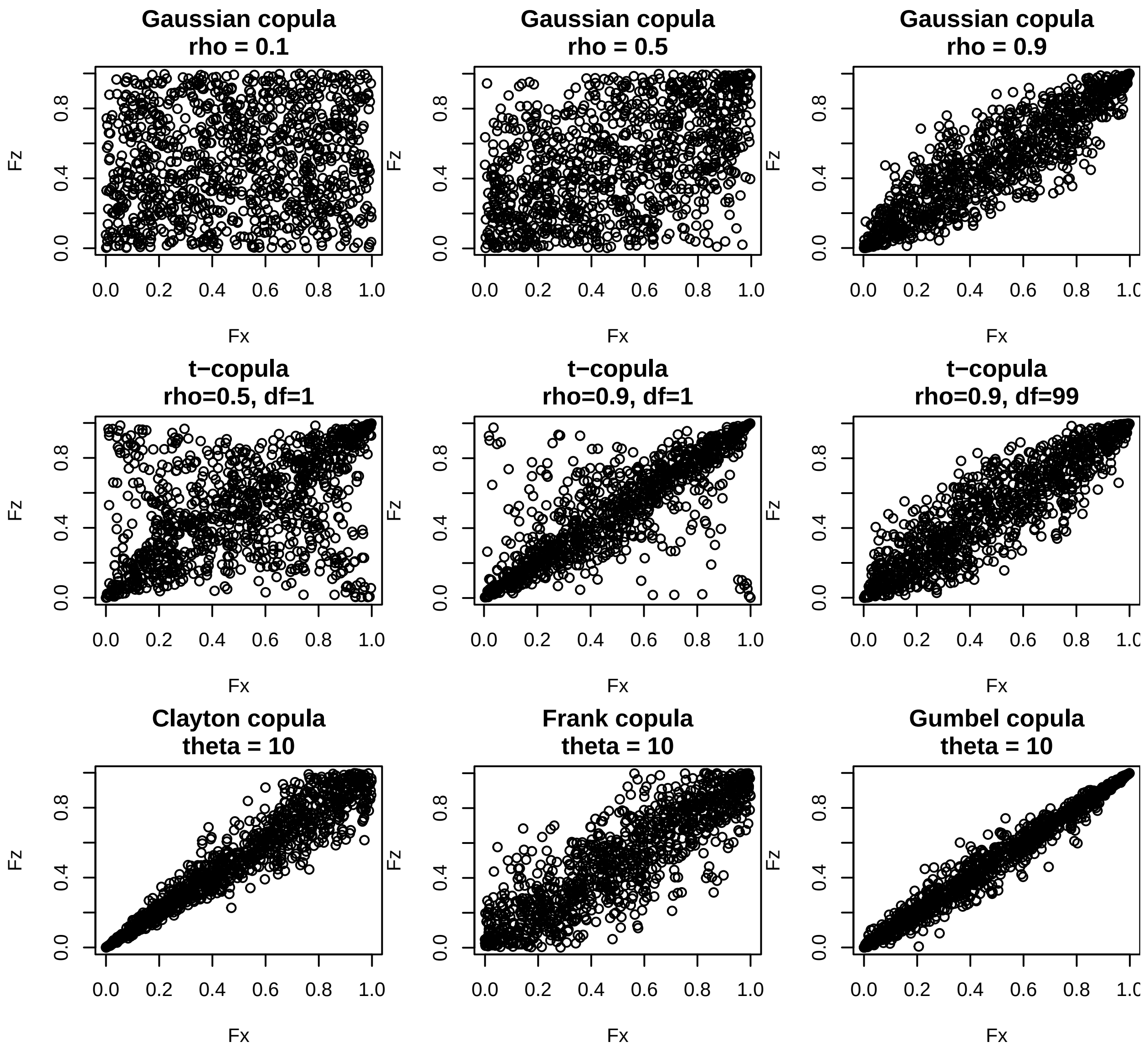


Figure 20: Sampling from different copula families and parameter settings. Top row: Gaussian copulas. Middle row: t-copulas. Bottom row: Clayton, Frank and Gumbel copulas.

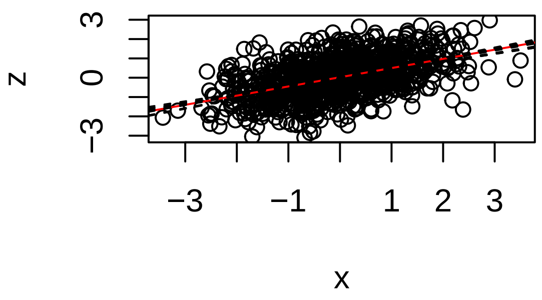
Bayesian model-based PRA

Linear example - Full Bayesian PRA with uncertainty

```r
xz <- l_xz.L[[1]] ; x  <- xz[,1] ; X  <- cbind( 1, x )
z  <- xz[,2]      ; Vz <- 1      ; Sz <- diag(Vz,n)
```

```r
# Prior:
mb         <- c(0,0) ; Vb <- c(1.e4,1.e4) ; Sb <- diag(Vb)
# Posterior:
Sb_y_LS72 <- solve( solve(Sb) + t(X) %*% solve(Sz) %*% X )
mb_y_LS72 <- Sb_y_LS72 %*% (solve(Sb) %*% mb + t(X) %*% solve(Sz) %*% z)
```

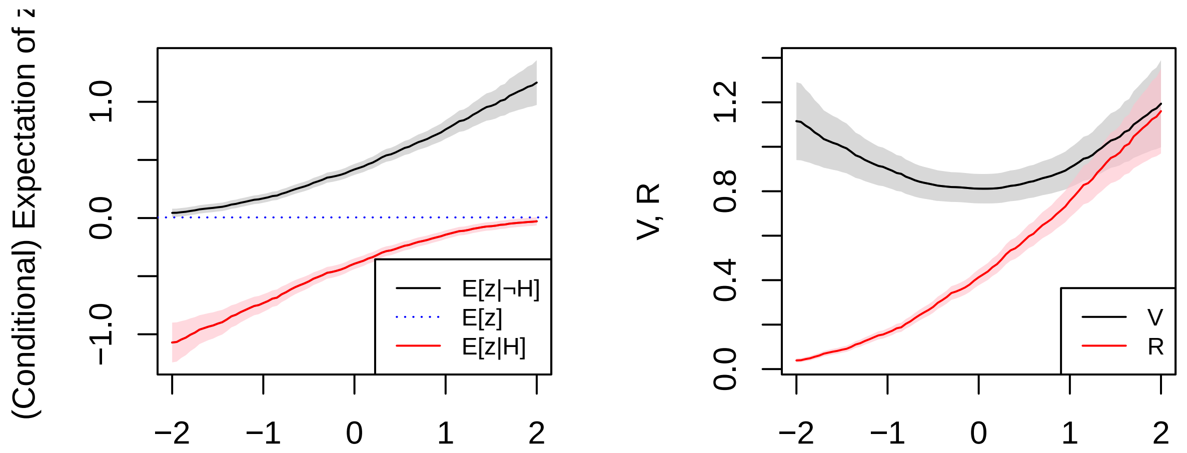
```r
par(mfrow=c(1,2))
plot( x, z ) ; abline( mb_y_LS72, col="red" )
nsmpl <- 10  ; smpl_b <- rmvnorm( nsmpl, mean=mb_y_LS72, sigma=Sb_y_LS72 )
for( i in 1:nsmpl) { abline( smpl_b[i,], lty=2 ) }
```



PRA

```r
thr  <- seq( -2, 2, by=0.05 ) ; n_thr <- length(thr)
n_H <- n_notH <- Ez_H <- Ez_NOTH <- Vz_H <- Vz_notH <- numeric(n_thr)
R   <- V      <- pH   <- s_R     <- s_V  <- s_pH    <- numeric(n_thr)
LCIR      <- UCIR      <- LCIV  <- UCIV              <- numeric(n_thr)
LCIzNOTH <- UCIzNOTH <- LCIzH <- UCIzH              <- numeric(n_thr)
for(i in 1:n_thr) {
  i_H    <- which( x <  thr[i] ) ; n_H[i]    <- length(i_H)
  i_notH <- which( x >= thr[i] ) ; n_notH[i] <- length(i_notH)
  pH[i]  <- n_H[i] / n           ; s_pH[i]   <- sqrt( pH[i]*(1-pH[i]) / n )
  # V, R from model expectations and UQ from law of total variance
  Ex_H       <- mean( x[i_H] )    ; Ez_H[i]    <- c(1,Ex_H)    %*% mb_y_LS72
  Ex_notH    <- mean( x[i_notH] ) ; Ez_notH[i] <- c(1,Ex_notH) %*% mb_y_LS72
  V[i]       <- Ez_notH[i] - Ez_H[i]
  R[i]       <- pH[i] * V[i]
  Vzi        <- function(i){ t(c(1,x[i])) %*% Sb_y_LS72 %*% c(1,x[i]) + Vz }
  Vz_H[i]    <- sum( sapply(i_H    ,Vzi) ) / n_H[i]    + mb_y_LS72[2]^2 * var(x[i_H])
  Vz_notH[i] <- sum( sapply(i_notH,Vzi) ) / n_notH[i] + mb_y_LS72[2]^2 * var(x[i_notH])
  s_V[i]     <- sqrt( Vz_H[i] / n_H[i] + Vz_notH[i] / n_notH[i] )
  s_R[i]     <- sqrt( s_pH[i]^2 * s_V[i]^2 + s_pH[i]^2 * V[i]^2 + pH[i]^2 * s_V[i]^2 )
}
```

```
LCIzNOTH <- Ez_notH - sqrt( Vz_notH / n_notH )
UCIzNOTH <- Ez_notH + sqrt( Vz_notH / n_notH )
LCIzH    <- Ez_H    - sqrt( Vz_H    / n_H    )
UCIzH    <- Ez_H    + sqrt( Vz_H    / n_H    )
LCIV     <- V - s_V ; UCIV <- V + s_V
LCIR     <- R - s_R ; UCIR <- R + s_R
```



Nonlinear example - Full Bayesian PRA with uncertainty

```
xz <- l_xz.NL[[1]] ; m <- colMeans(xz) ; S <- cov(xz)
x  <- xz[,1]       ; z <- xz[,2]       ; n <- length(x)
```

```
Model1.Code <- nimbleCode({
  lm.alpha  ~ dnorm( 0, sd=100 )
  lm.beta   ~ dnorm( 0, sd=100 )
  lm.tau    ~ dgamma( 0.01, 0.01 )
  lm.sigma <- 1 / sqrt(lm.tau)
  for(i in 1:ndata){
    lm.mu[i] <- lm.alpha + lm.beta*exp(-x[i])
    z[i]      ~ dnorm( lm.mu[i], sd=lm.sigma )
  }
} )
```

Checking the MCMC

```
>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
>  0.9795  0.9969  1.0004  1.0004  1.0041  1.0176
>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
> -1.0558 -1.0180 -1.0090 -1.0091 -1.0004 -0.9592
>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
> 0.09547 0.10201 0.10344 0.10353 0.10511 0.11215
>
> Call:
> lm(formula = z ~ exp(-x))
>
```

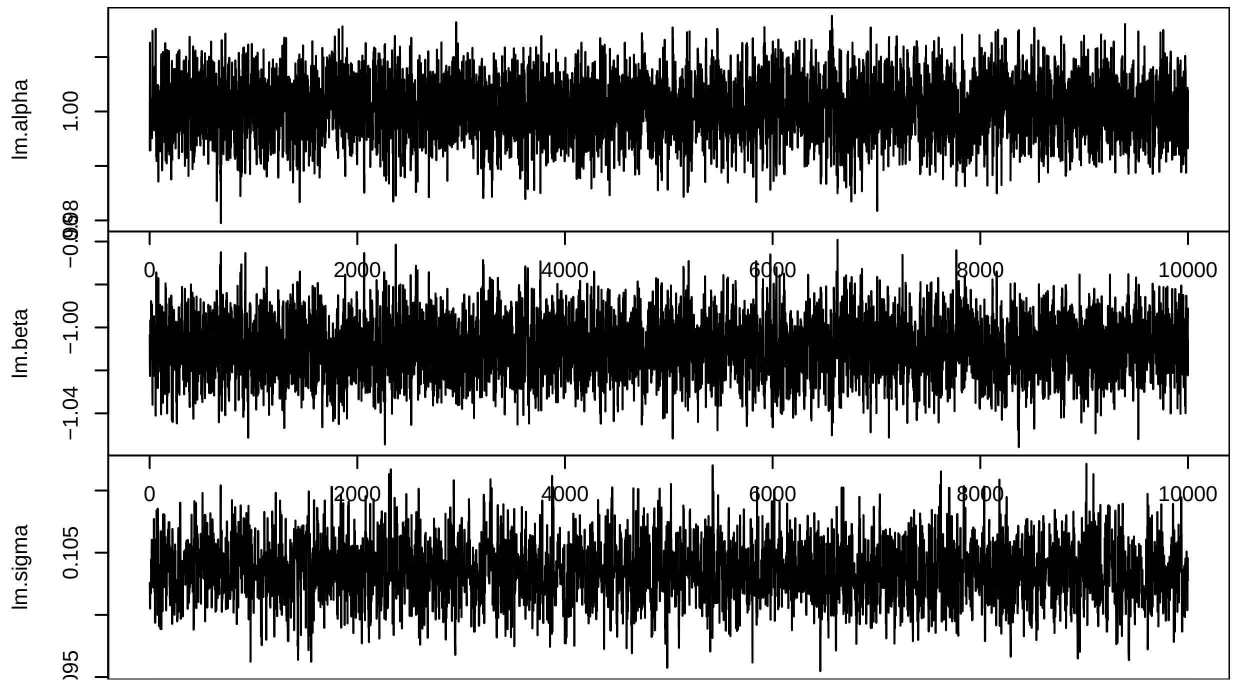


Figure 21: Parameter trace plots generated by Nimble for the nonlinear example.

```
> Residuals:
>      Min       1Q   Median       3Q      Max
> -0.32522 -0.07344 -0.00117  0.07431  0.36544
>
> Coefficients:
>              Estimate Std. Error t value Pr(>|t|)
> (Intercept)  1.000484   0.005226  191.46   <2e-16 ***
> exp(-x)     -1.009282   0.012912  -78.17   <2e-16 ***
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> Residual standard error: 0.1034 on 998 degrees of freedom
> Multiple R-squared:  0.8596,  Adjusted R-squared:  0.8595
> F-statistic:  6110 on 1 and 998 DF,  p-value: < 2.2e-16
```

PRA

```
n_unc     <- 1e3
itheta    <- sample( 1:ntheta, n_unc, replace=(ntheta<n_unc) )
thr       <- seq( 0.1, 2.9, by=0.05 )  ;   n_thr <- length(thr)
EzHj      <- EzNOTHj  <-   Rj  <-   Vj          <- numeric(n_unc)
EzH       <- EzNOTH   <-    R  <-    V  <-   pH <- numeric(n_thr)
                          s_R  <- s_V   <- s_pH <- numeric(n_thr)
LCIR      <- UCIR     <- LCIV  <- UCIV          <- numeric(n_thr)
LCIzNOTH  <- UCIzNOTH <- LCIzH <- UCIzH         <- numeric(n_thr)
lm.alpha  <- theta[,1] ; lm.beta <- theta[,2] ; lm.sigma <- theta[,3]
for(i in 1:n_thr) {
  i_H   <- which( x < thr[i] ) ; n_H     <- length(i_H)
  pH[i] <- n_H / n             ; s_pH[i] <- sqrt( pH[i]*(1-pH[i]) / n )
  for(j in 1:n_unc) {
    zj      <- lm.alpha[itheta[j]] + lm.beta[itheta[j]] * exp(-x) +
               rnorm( n, 0, lm.sigma[itheta[j]] )
    EzHj[j] <- mean( zj[i_H] ) ; EzNOTHj[j] <- mean( zj[-i_H] )
    Vj[j]   <- EzNOTHj[j] - EzHj[j]
    Rj[j]   <- pH[i] * Vj[j] }
  R[i]   <- mean( Rj ) ; V[i]   <- mean( Vj )
  s_R[i] <- sd  ( Rj ) ; s_V[i] <- sd  ( Vj )
  EzH[i]      <- mean( EzHj )  ; EzNOTH[i]   <- mean( EzNOTHj )
  qu          <- function( z, q=0.025 ){ quantile( z, q, na.rm=T ) }
  LCIzNOTH[i] <- qu( EzNOTHj ) ; UCIzNOTH[i] <- qu( EzNOTHj, 0.975 )
  LCIzH[i]    <- qu( EzHj    ) ; UCIzH[i]    <- qu( EzHj    , 0.975 )
  LCIV[i]     <- qu( Vj      ) ; UCIV[i]     <- qu( Vj      , 0.975 )
  LCIR[i]     <- qu( Rj      ) ; UCIR[i]     <- qu( Rj      , 0.975 )
}
```

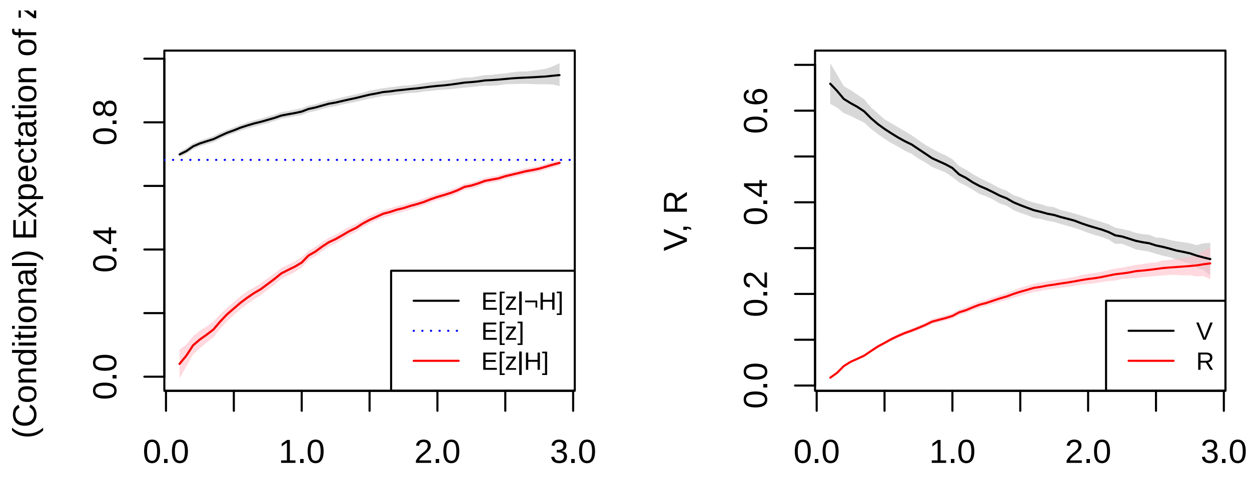


Figure 22: PRA on nonlinearly related x and z, with UQ following Bayesian modelling with Nimble.

Advantages of the Bayesian modelling approach

Sampling-based multi-threshold PRA: Gaussian linear example

```
PRAm <- function( x, z, thr=-1:1 ) {
  n   <- length(x) ; n_thr <- length(thr)
  H   <- vector("list",n_thr)
  n_H <- pH <- V <- R <- s_pH <- s_V <- s_R <- rep(NA,n_thr)
  H[[1]]  <- which( x < thr[1] ) ; n_H[1] <- length(H[[1]])
  for(i in 2:n_thr) { H[[i]] <- which( thr[i-1] <= x & x < thr[i])
                      n_H[i] <- length(H[[i]]) } ; n_notH <- n - sum(n_H)
  H.all   <- which( x < thr[n_thr] )
  pH      <- n_H / n             ; s_pH      <- sqrt( pH*(1-pH) / n )
  Ez_notH <- mean( z[-H.all] )   ; s_Ez_notH <- sqrt( var(z[-H.all] ) / n_notH )
  for(i in 1:n_thr) {
    Ez_Hi <- mean( z[ H[[i]] ] ) ; s_Ez_Hi <- sqrt( var(z[ H[[i]]]) /  n_H[i] )
    V[i]  <- Ez_notH - Ez_Hi     ; s_V[i]  <- sqrt( s_Ez_notH^2 + s_Ez_Hi^2 ) }
  R       <- pH * V
  s_R     <- sqrt( s_pH^2 * s_V^2 + s_pH^2 * V^2 + pH^2 * s_V^2 )
  R.sum   <- sum(R) ; pH.sum <- sum(pH) ; V.wsum <- R.sum / pH.sum
  return( list( sum = c( pH.sum=pH.sum, V.wsum=V.wsum, R.sum=R.sum ),
                seq = cbind( thr, pH, V, R, s_pH, s_V, s_R ) ) )
}
```

```
xz <- l_xz.L[[1]] ; x <- xz[,1] ; z <- xz[,2]
```

```
thr <- -1:-0 ; pram2 <- PRAm(x, z, thr)
```

```
> Overall values: pH.sum = 0.509 ; V.wsum = 0.855556 ; R.sum = 0.435478
> Vector values pH: 0.182 0.327
> Vector values V : 1.148718 0.6923893
> Vector values R : 0.2090667 0.2264113
```

```
thr <- seq(-2,1,0.1) ; pram30 <- PRAm (x, z, thr)
```

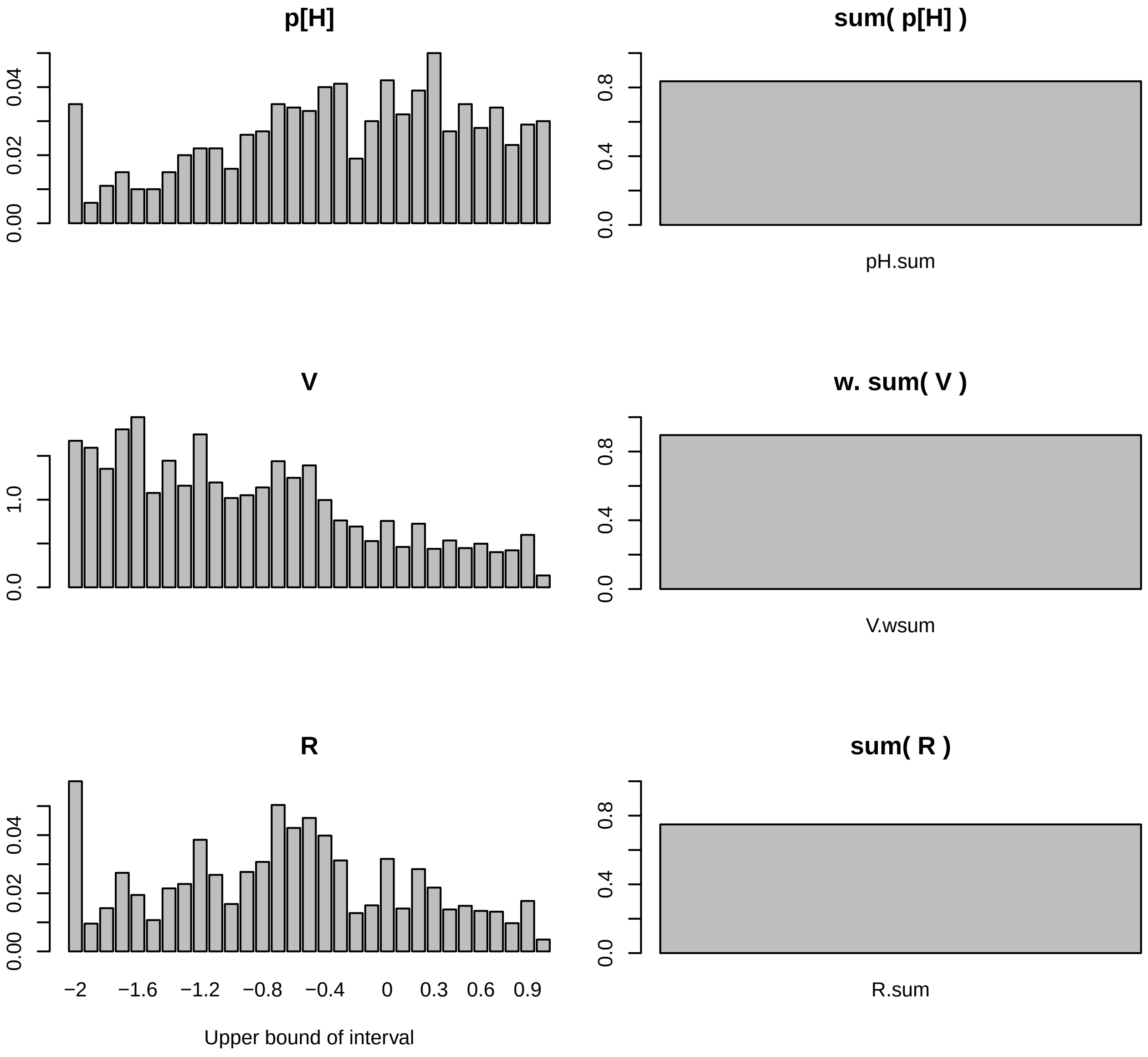


Figure 23: Multi-threshold PRA applied to a Gaussian linear data set. Left: results for 31 hazard-levels. Right: overall (summed) values for all hazard levels combined.

Distribution-based continuous PRA: Gaussian linear example

```r
p <- px # Gaussian density function as defined above
  mx   <- mz <- 0 ; rxz <- 0.5
  Ez_x <- function(x){ mz + (x-mx)*rxz }
v <- function(x,thr=0) { Ez_xhi_NI(thr) - Ez_x(x) }
r <- function(x,thr=0) { p(x) * v(x,thr) }

thr   <- 1
p.seq <- p(x.seq) ; v.seq <- v(x.seq,thr) ; r.seq <- r(x.seq,thr)
```

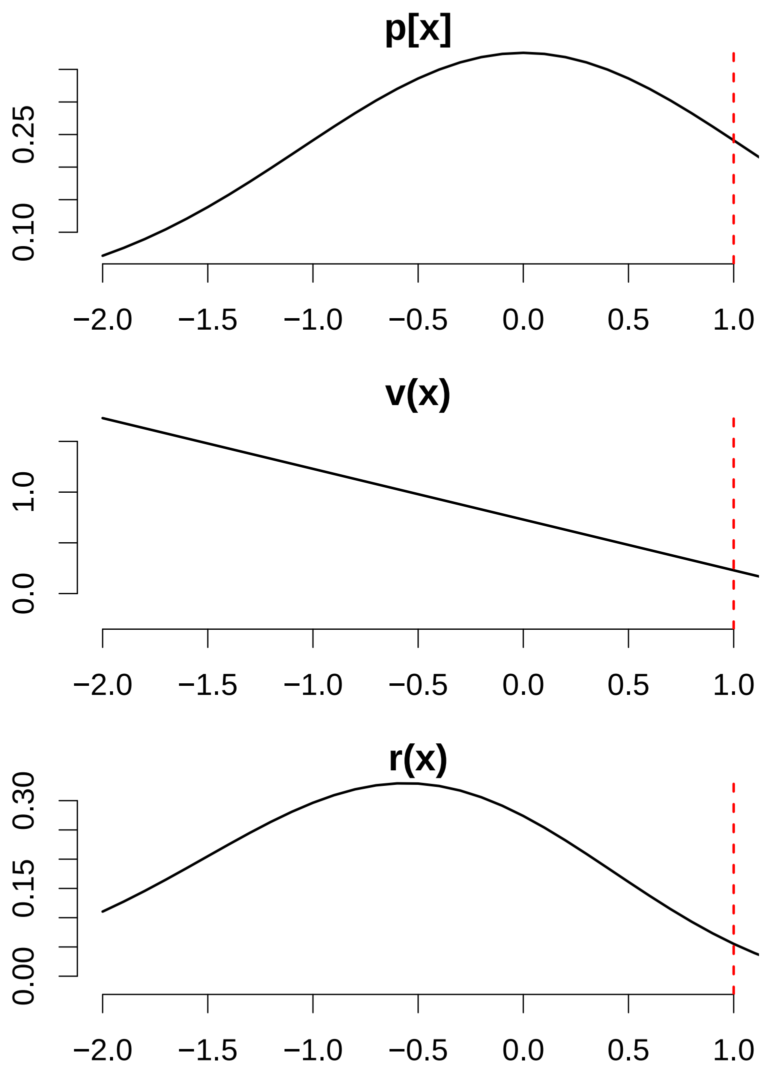


Figure 24: Continuous PRA applied to a bivariate Gaussian dsistribution.

Categorical PRA with other splits than for threshold-levels: spatio-temporal example

Spatio-temporal environmental data: $x(s, t)$

```
GP.AR <- function( s0, s, x, Vx.s, phi, x0past=0, Vx.t=0, alpha=0 ) {
  ds  <- as.matrix( dist(s) )
  rx  <- exp( -ds/phi )
  ds0 <- sapply(1:length(x),function(i){dist(rbind(s0,s[i,]))})
  r0  <- exp( -ds0/phi )
  m0  <- t(r0) %*% solve(rx) %*% x + alpha * x0past
  V0  <- Vx.s * (1 - t(r0) %*% solve(rx) %*% r0 ) + Vx.t
  return( c( m0=m0, V0=V0 ) ) }
```

```
ns1  <- 8 ; ns2 <- 8 ; nt <- 9
xst  <- array( NA, dim=c(ns1,ns2,nt) )

Vx.s <- 0.5 ; phi   <- 1
Vx.t <- 0.5 ; alpha <- 0.5
```

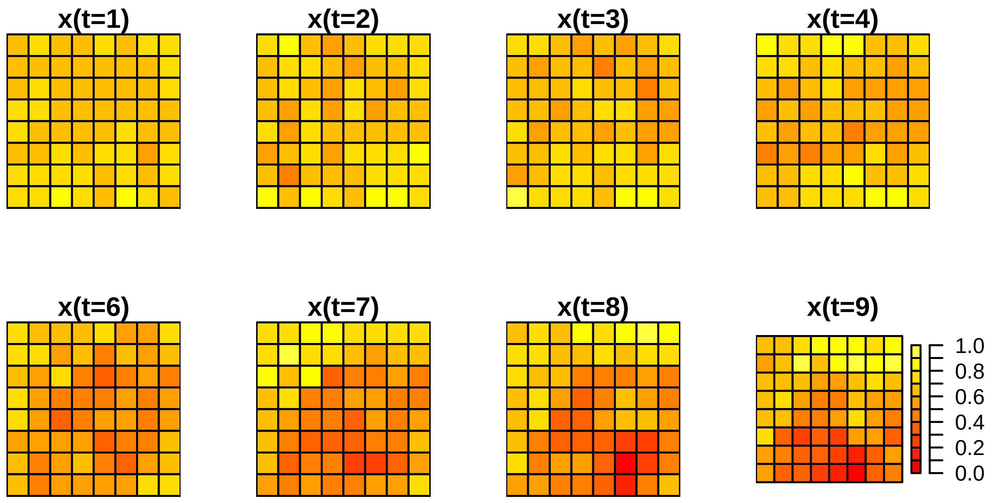


Figure 25: Evolution of the environmental variable $x(s, t)$ over time.

Spatio-temporal system data: $z(s, t)$

```
fz  <- function( x, xpast=1, k=10 ) {
  1 / (1 + exp( -k * (x + xpast/2 - 0.5 ) ) ) }
```

Single-category single-threshold PRA for the spatio-temporal data

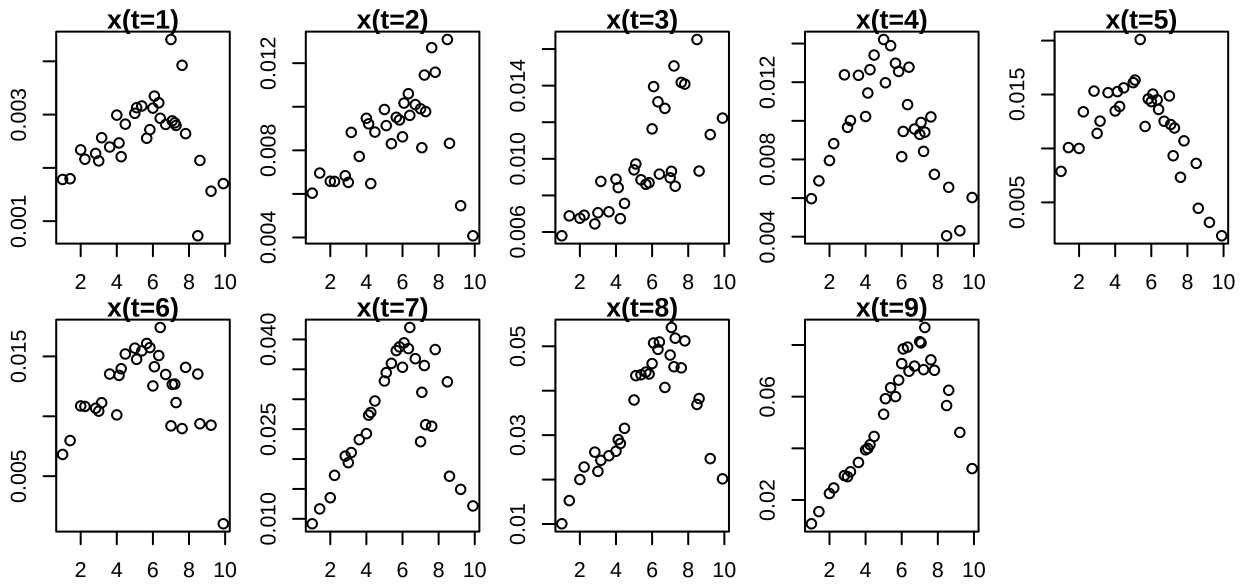


Figure 26: Spatial variograms for $x(z,t)$.

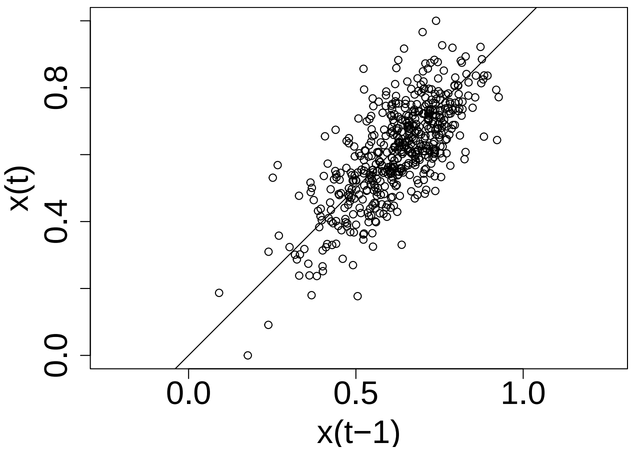


Figure 27: Temporal correlation of the environmental variable.

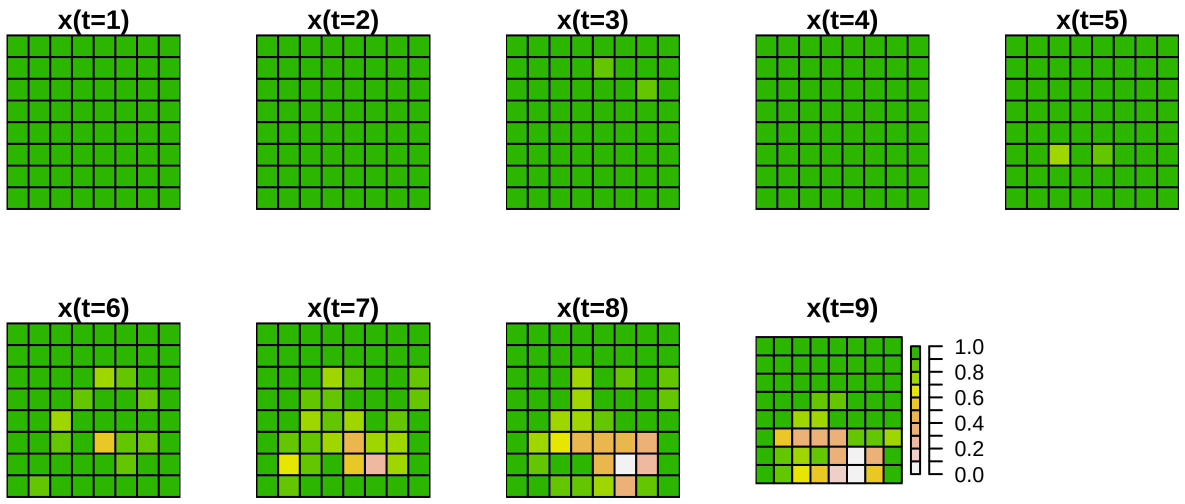


Figure 28: Evolution of the response variable $z(s,t)$ over time.

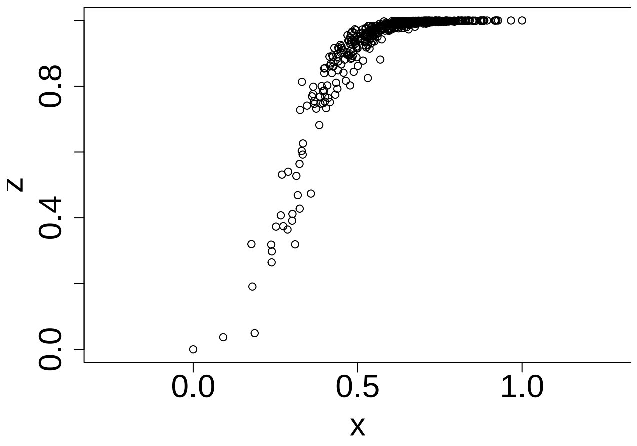


Figure 29: Responses of z to x for all locations and times.

```
thr.xst <- quantile( xst, pnorm(-1) )
```

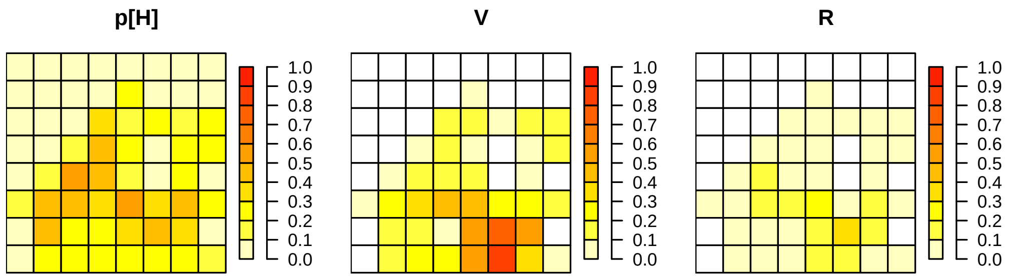


Figure 30: PRAs for all cells in a square region. Each of the 64 PRAs was based on a single cell's time series of (x, z).

Two-category single-threshold PRA for spatio-temporal data

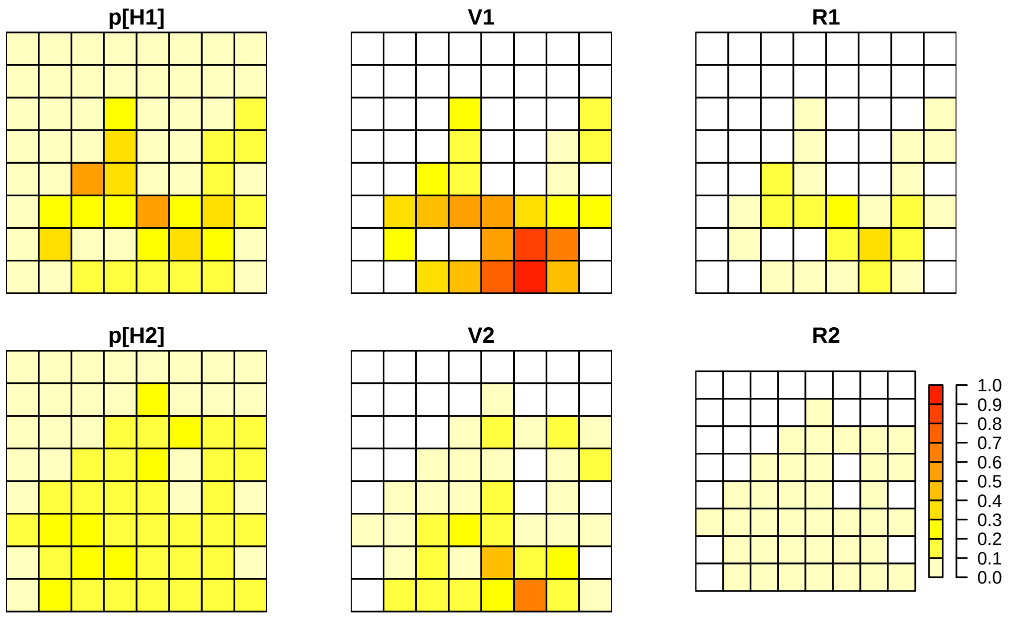


Figure 31: Two-category PRAs for all cells in a square region. Top row: category 1 droughts (preceded by drought in the time step before). Bottom row: category 2 droughts (not preceded by drought).

Three-component PRA

Three-component PRA for spatio-temporal data

```
> Three-component PRA:
>    Q = 37 ; pH = 0.2762763 ; V = 0.2604741
>    Rintensive = 0.07196282 ; Rextensive = 2.662624
```

Country-wide application of three-component PRA

UQ for three-component PRA

$$\sigma_{R_e} = \sqrt{\sigma_Q^2 \sigma_{R_i}^2 + \sigma_Q^2 E[R_i]^2 + \sigma_{R_i}^2 E[Q]^2}.(\#eq:sigmaRe) \tag{25}$$