

## @Norm2d

Implement the MATLAB class `@Norm2d` as a class folder

### Properties

#### Mean

A  $2 \times 1$  vector  $M = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$

#### Covariance

A  $2 \times 2$  positive definite matrix  $\Sigma = \begin{pmatrix} \sigma_1^2 & c_{12} \\ c_{12} & \sigma_2^2 \end{pmatrix}$

#### Precision

A  $2 \times 2$  matrix  $T$  that is the inverse of the covariance matrix:  $T = \Sigma^{-1}$

#### Correlation

A scalar  $\rho$  that is  $c_{12}/(\sigma_1\sigma_2)$

### Methods

#### Constructor `Norm2d(Mu, Sigma)`

Make sure that the user can make a Norm2d object without providing parameters. Also make sure  $T$  is always correct

#### Getters and setters

Make sure I can get and set Mean and Covariance, and that I can get Precision and Correlation, and that these are always consistent

#### Ordinary methods

**1. Probability density function `pdf(X,Mu,Sigma)`** The bivariate normal probability density function is given by

$$p(x_1, x_2 | M, \Sigma) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2} \frac{z}{1-\rho^2}\right),$$

where

$$z = \left(\frac{x_1 - \mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x_1 - \mu_1}{\sigma_1}\right)\left(\frac{x_2 - \mu_2}{\sigma_2}\right) + \left(\frac{x_2 - \mu_2}{\sigma_2}\right)^2,$$

and

$$\rho = \text{cor}(x_1, x_2) = \frac{c_{12}}{\sigma_1\sigma_2}$$

is the correlation of  $x_1$  and  $x_2$  and  $c_{12}$  is the covariance.

Make sure I can input a  $2 \times n$  matrix for  $X$  and obtain a  $1 \times n$  output

**2. Log-PDF `logpdf(X,Mu,Sigma)`** The log-pdf will be useful in the future. You'll want to implement it *efficiently*. Don't just compute the pdf and take the logarithm – first you should mathematically simplify this:

$$l(x_1, x_2 | M, \Sigma) = \log(p(x_1, x_2 | M, \Sigma))$$

and implement the result

**3. Cumulative distribution function `cdf(X,Mu,Sigma)`** Use the `mvncdf()` function from the Statistics toolbox

**4. Log-CDF `logcdf(X,Mu,Sigma)`** Use the `mvncdf()` function from the Statistics toolbox

**5. Random number generation `rng(Mu,Sigma,size)`** A sample from a bivariate normal distribution can be simulated by first simulating a point from the marginal distribution of one of the random variables and then simulating from the second random variable conditioned on the first:

$$\begin{aligned} x_1 &\sim N(\mu_1, \sigma_1) \\ x_2 &\sim N\left(\mu_2 + \sigma_2 \rho \left(\frac{x_1 - \mu_1}{\sigma_1}\right), \sigma_2^2 \sqrt{1 - \rho^2}\right) \end{aligned}$$

Make sure this function returns a matrix of size **size**

**6. Deviance `deviance(Data,Mu,Sigma)`** The data set  $D$  is a  $2 \times n$  matrix:

$$D = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \\ \vdots & \vdots \\ d_{n,1} & d_{n,2} \end{pmatrix}$$

The deviance is the summed logpdf of all points in a data set, times -2:

$$dev(D | M, \Sigma) = -2 \sum_{i=1}^n l(d_{i,1}, d_{i,2} | M, \Sigma)$$

Make sure this returns a scalar