Carnegie Mellon

School of Computer Science

Deep Reinforcement Learning and Control

Bayesian Optimization with Gaussian Processes

Spring 2021, CMU 10-403

Katerina Fragkiadaki

Used Materials

- **Disclaimer:** Some material and slides for this lecture were borrowed from Nando de Freitas lecture on Gaussian processes and Bayesian Optimization, from Richard Turner's lecture on Gaussian processes, and from Kirthevasan Kandasamy's lecture on Bayesian optimization.

This lecture - Motivation

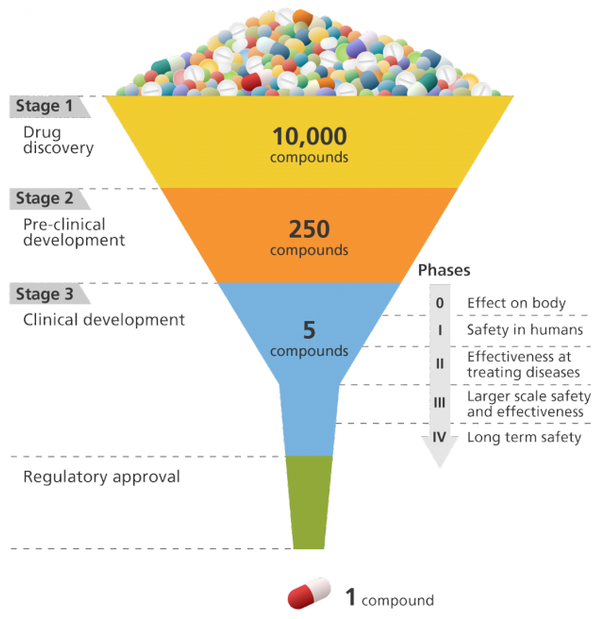
Learning to act in a non-sequential setup with continuous actions:

- Each action returns an immediate reward.
- We want to choose actions that maximize our expected (immediate) reward.

Example: drug discovery

Actions: the compounds to mix

Rewards: drug effectiveness/safety (e.g., as measured in mice).



This lecture - Motivation

Learning to act in a non-sequential setup:

- Each action returns an immediate reward.
- We want to choose actions that maximize our expected (immediate) reward.

Example: drilling for oil

Actions: where to drill next

Rewards: how much oil I found

This lecture - Motivation

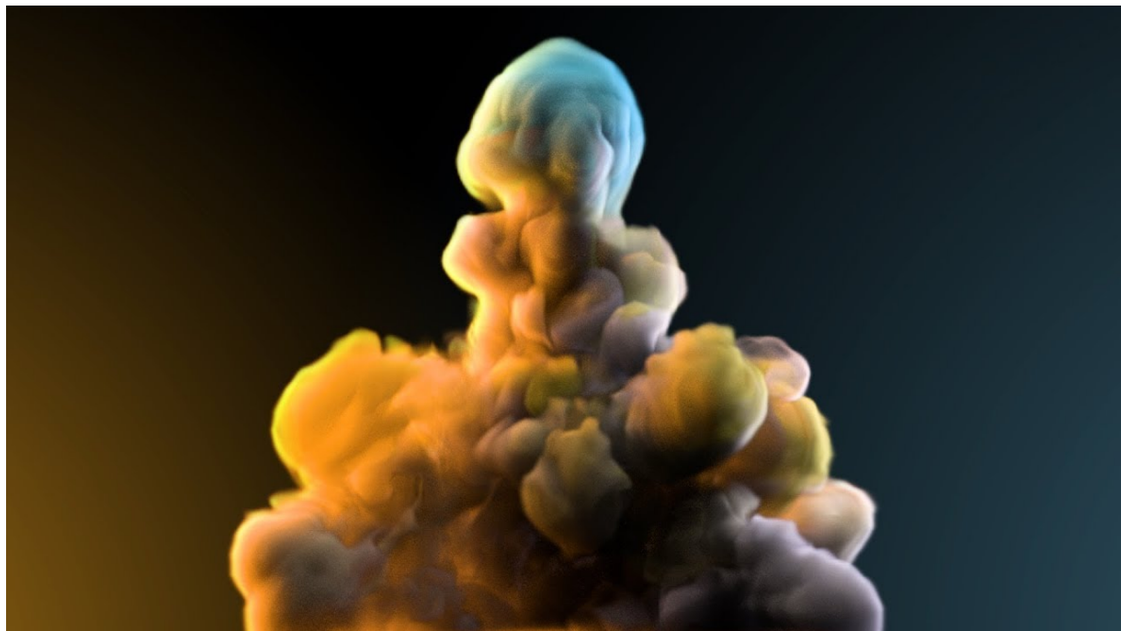
Learning to act in a non-sequential setup:

- Each action returns an immediate reward.
- We want to choose actions that maximize our expected (immediate) reward.

Example: simulating smoke

Actions: what simulation parameters to use

Rewards: how realistic the resulting smoke looks



This lecture - Motivation

Learning to act in a non-sequential setup:

- Each action returns an immediate reward.
- We want to choose actions that maximize our expected (immediate) reward.

Example: walking after breaking your ankle

Actions: what walking policy to use

Rewards: how fast you ‘ll move

This lecture - Motivation

Learning to act in a non-sequential setup:

- Each action returns an immediate reward.
- We want to choose actions that maximize our expected (immediate) reward.

It turns out, this is equivalent to maximizing a function for which:

- We do not have an explicit parametric form, e.g., we do not know the mapping from smoke simulation parameters to realism/human pleasure from watching the smoke
- We may have a parametric form but function evaluation is very expensive.

In both cases, we cannot use gradient information.

This lecture - Motivation

Learning to act in a non-sequential setup:

- Each action returns an immediate reward.
- We want to choose actions that maximize our expected (immediate) reward.

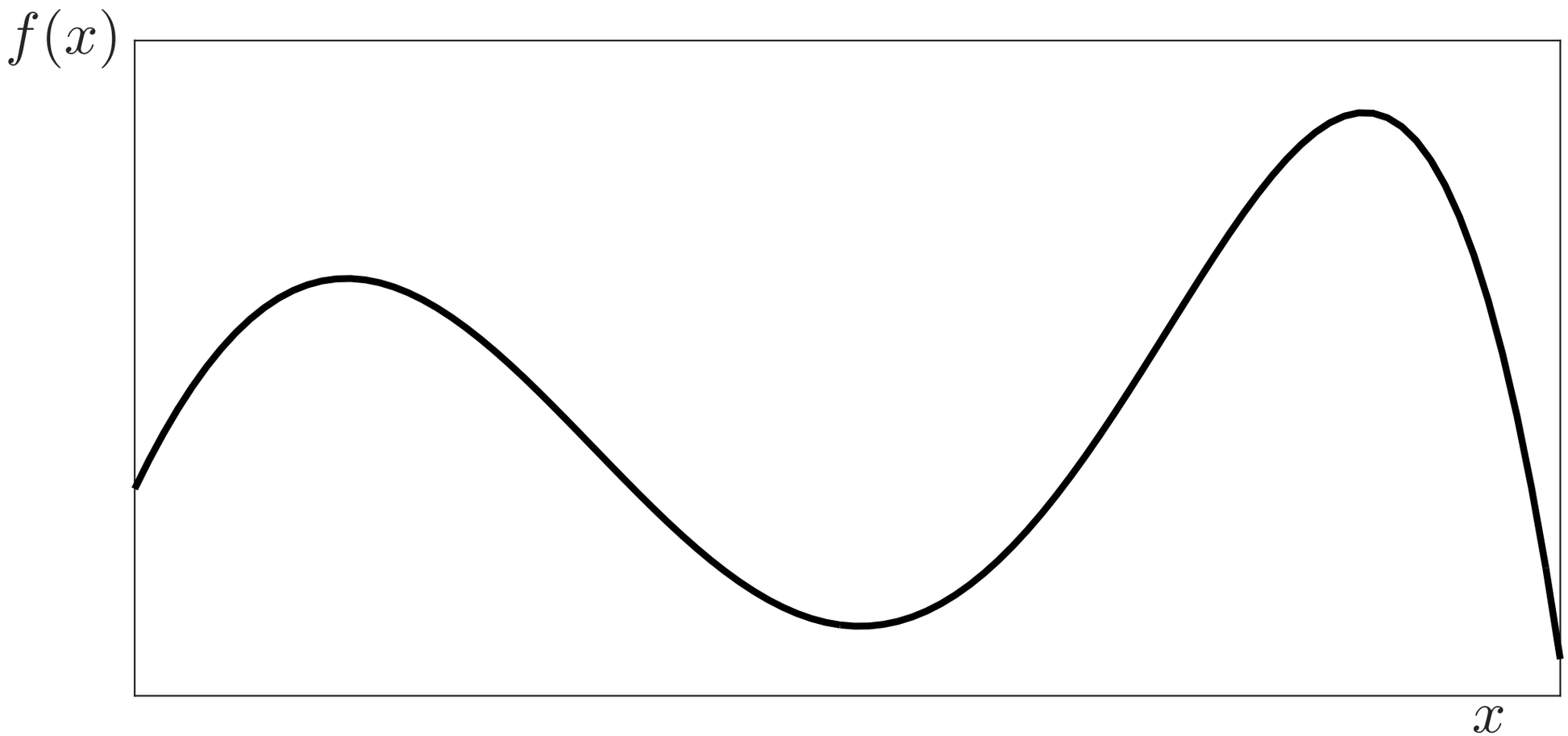
It turns out, this is equivalent to black-box (no gradients) optimization of functions.

Actions: places to evaluate the function.

Rewards: the value of the function.

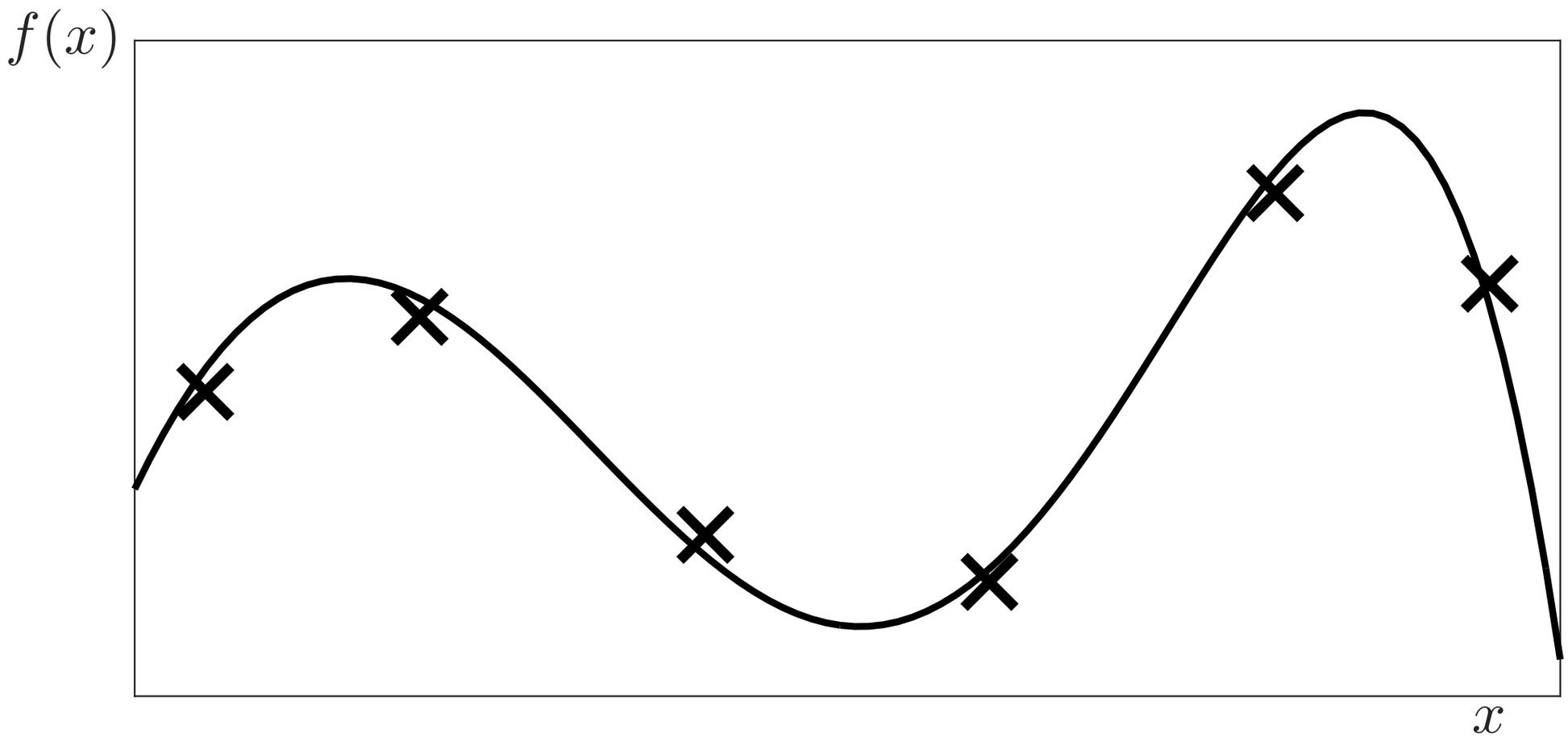
Black-box Optimization

$f : \mathcal{X} \to \mathbb{R}$ is an expensive black-box function, accessible only via noisy evaluations.



Black-box Optimization

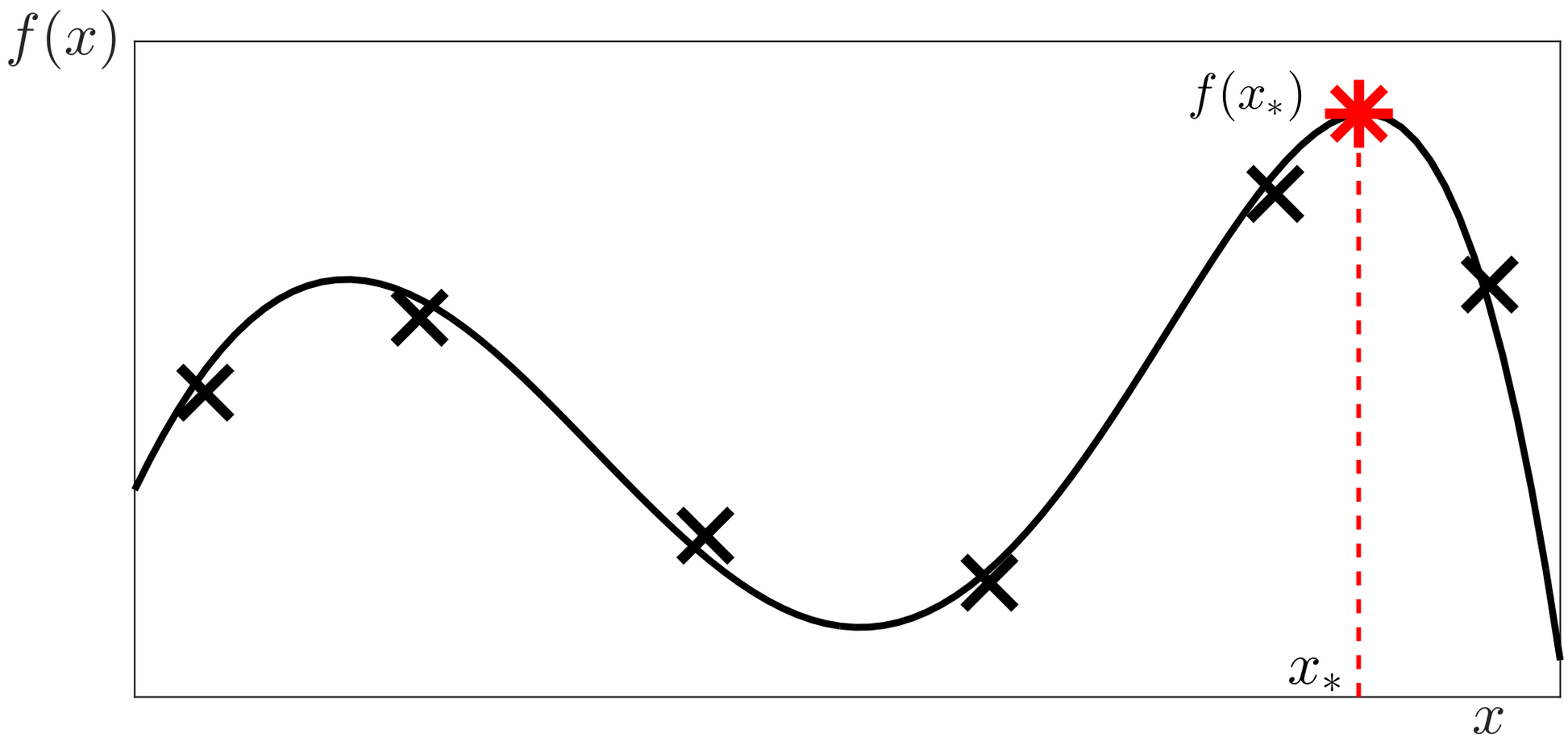
$f : \mathcal{X} \to \mathbb{R}$ is an expensive black-box function, accessible only via noisy evaluations.



Black-box Optimization

$f : \mathcal{X} \to \mathbb{R}$ is an expensive black-box function, accessible only via noisy evaluations.

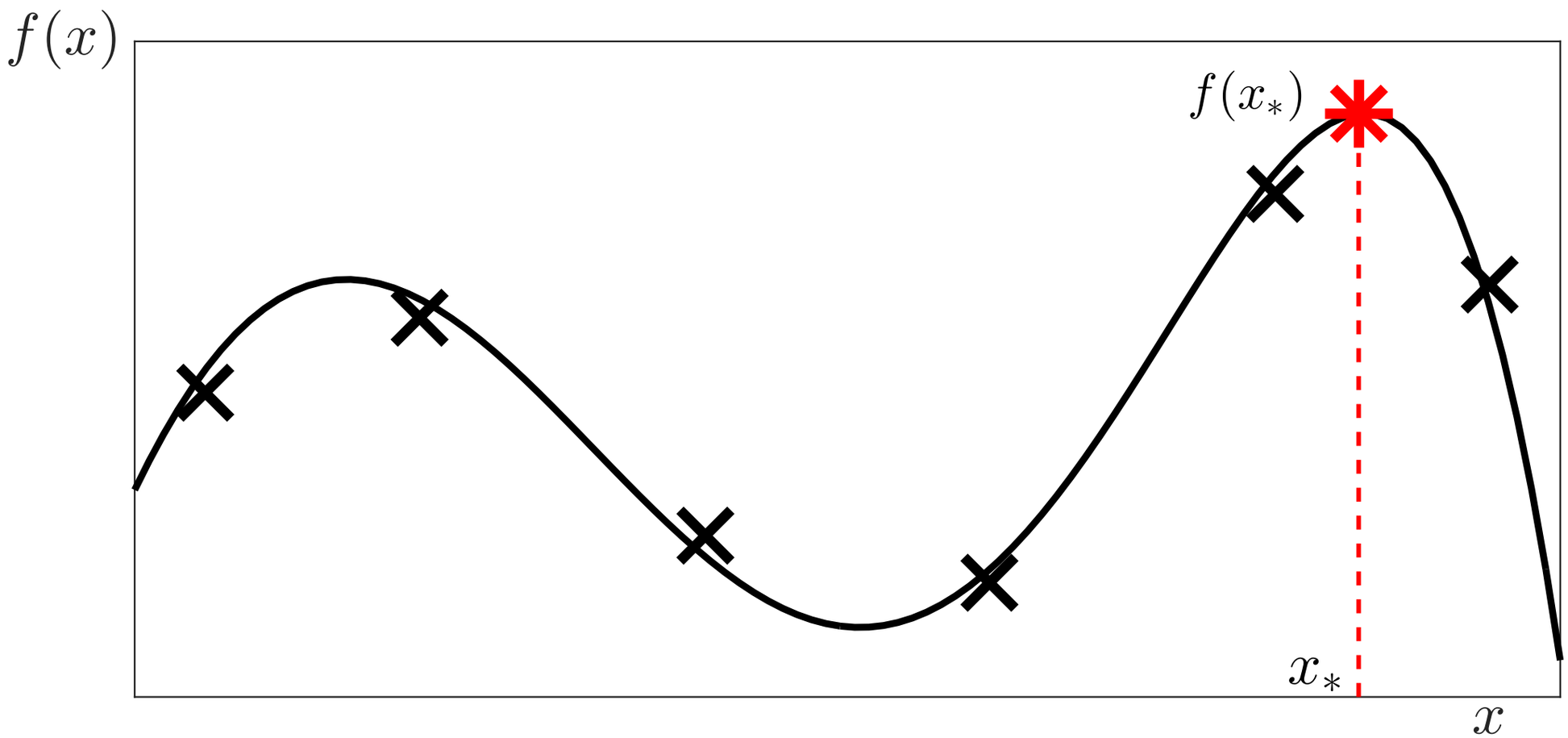
Let $x_* = \text{argmax}_x f(x)$



Black-box Optimization

$f : \mathcal{X} \to \mathbb{R}$ is an expensive black-box function, accessible only via noisy evaluations.

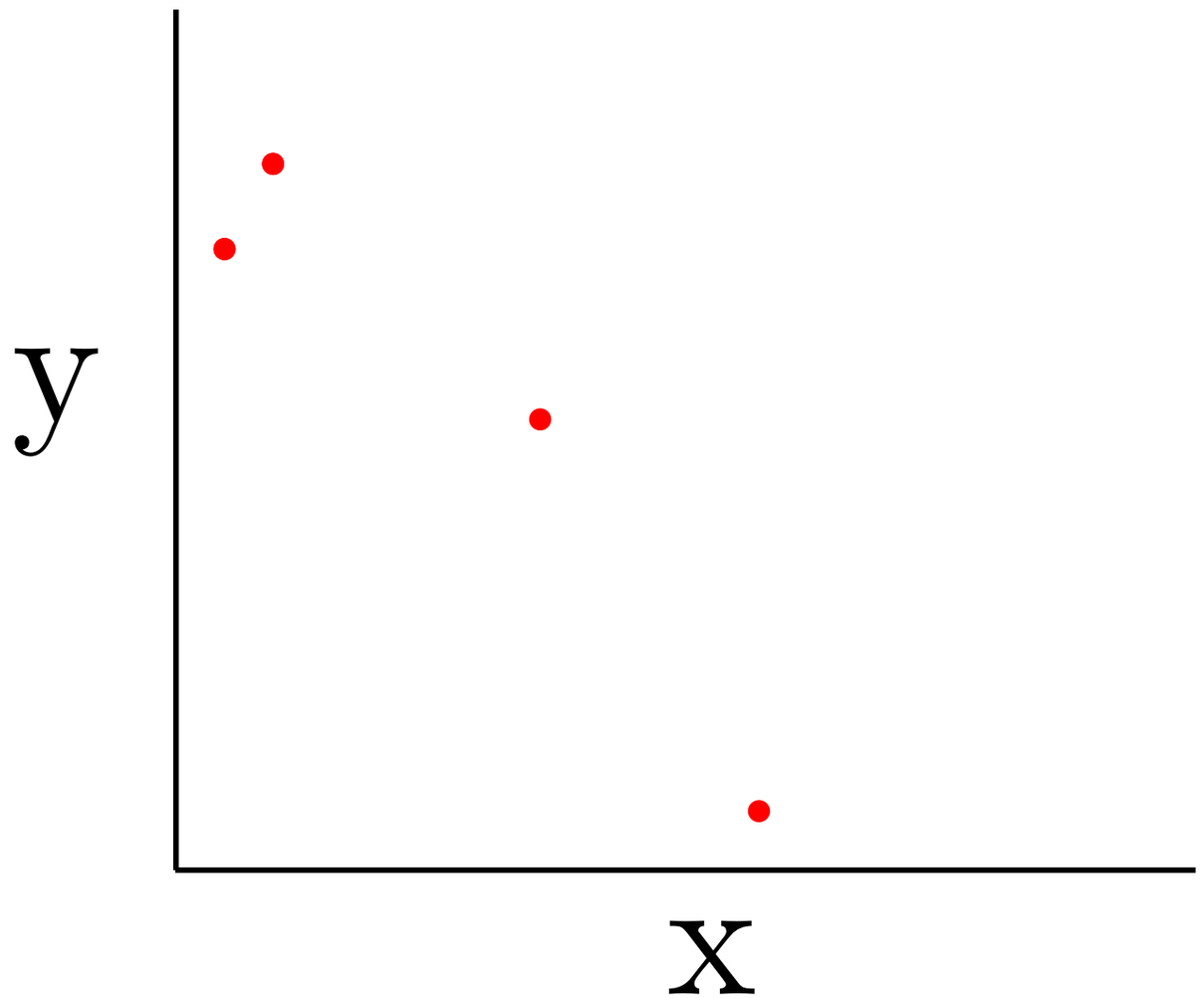
Let $x_* = \text{argmax}_x f(x)$



We want to find the point x^* with as few function evaluations as possible.

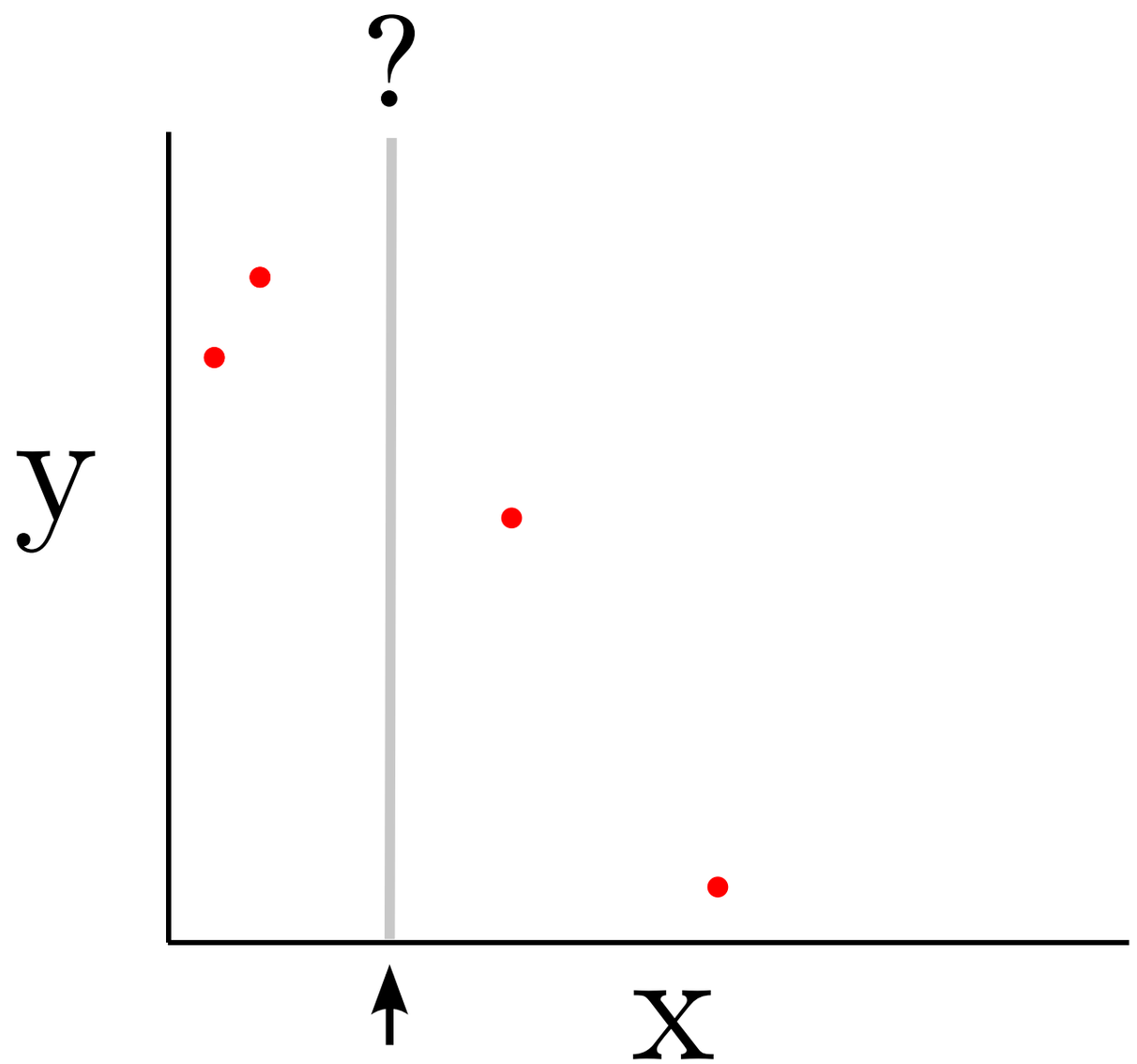
My action space: where in the x axis I should evaluate the function next.

Non-linear regression

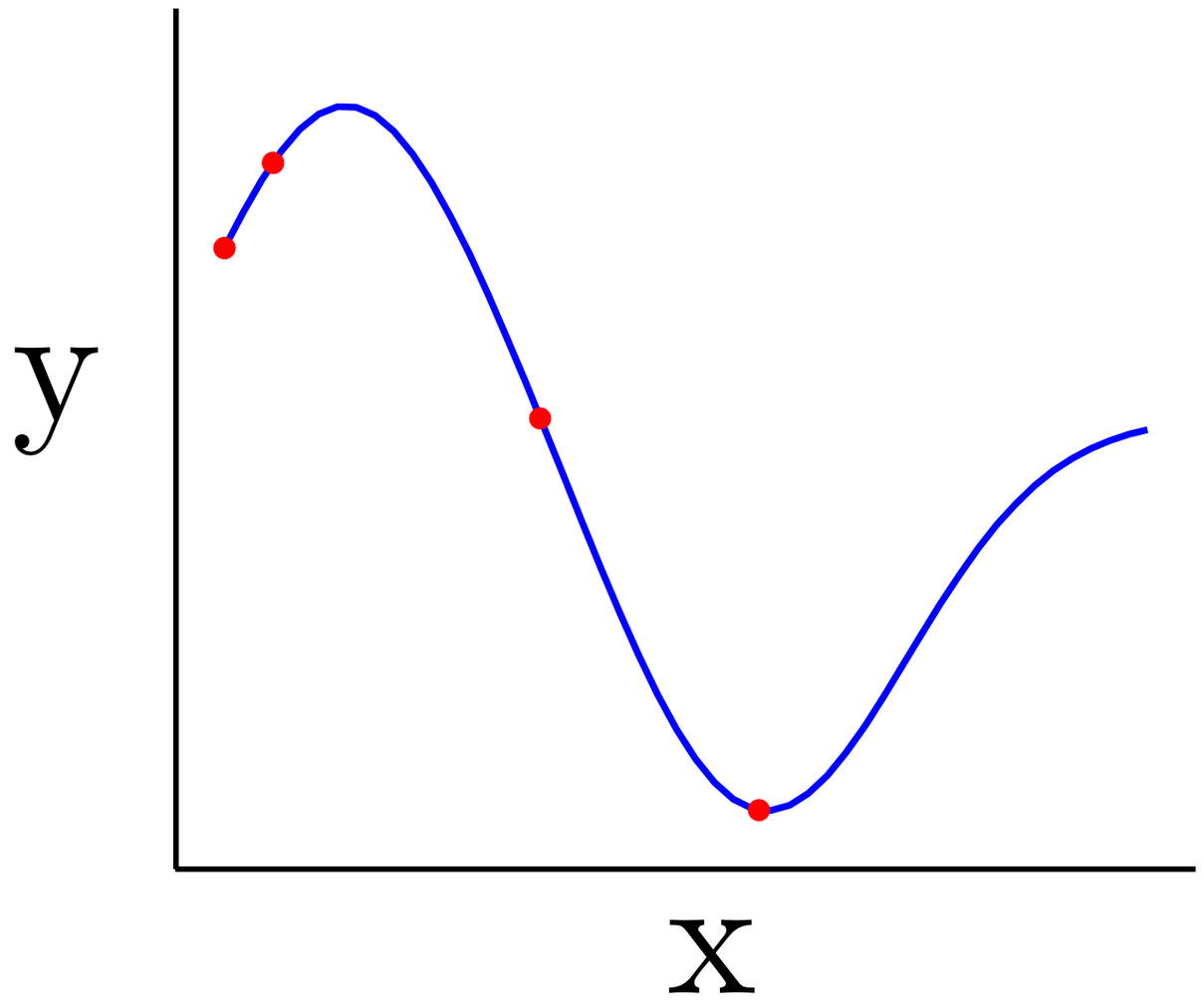


Which x location would you select next?

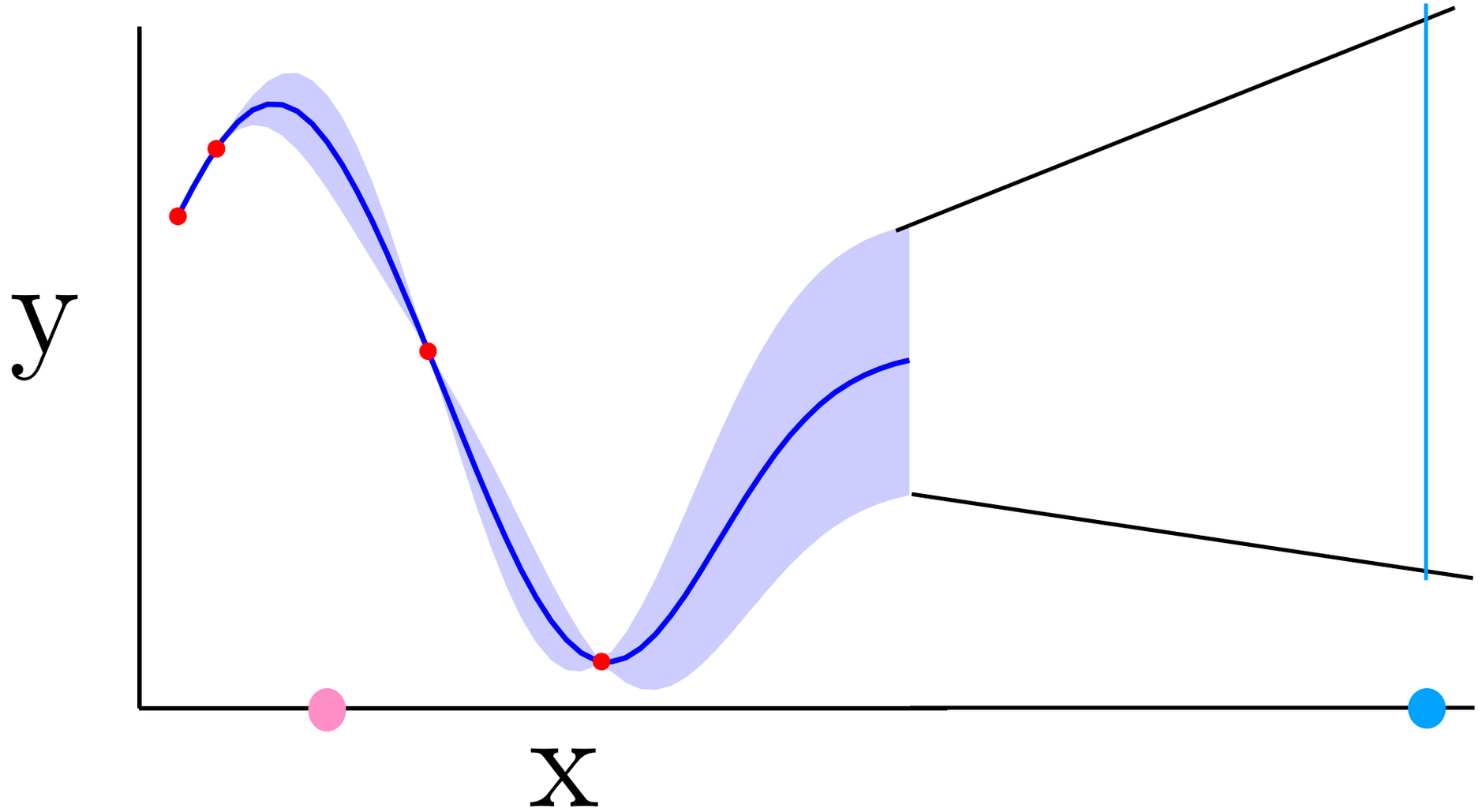
Non-linear regression



Non-linear regression



Non-linear regression with uncertainty



- This point seems the most promising from what I know so far (exploit)
- This point seems the point I am most uncertain about (explore)

Next: Non-linear regression with error bars using Gaussian processes

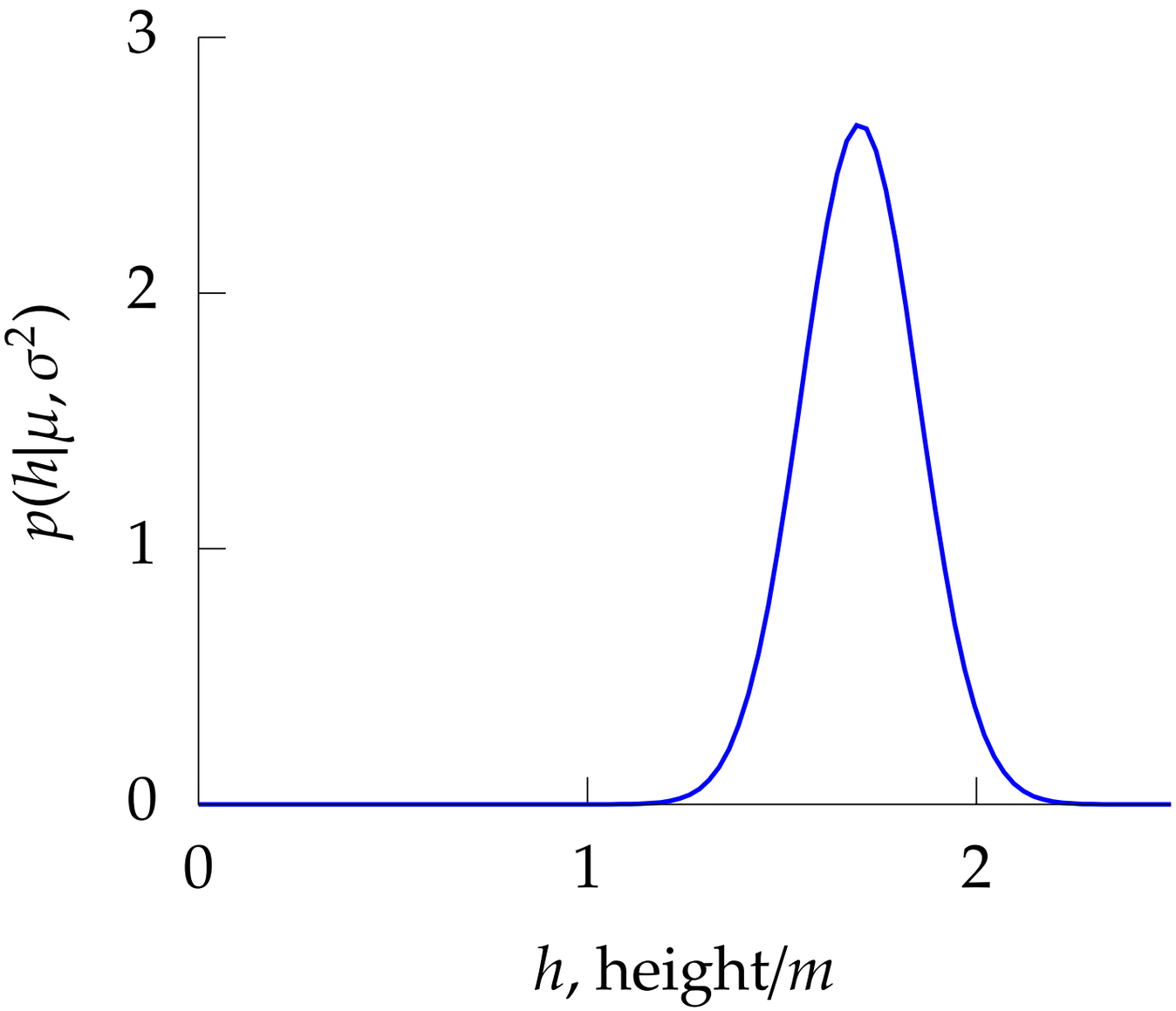
Gaussian Density

Perhaps the most common probability density

$$\mathcal{N}\left(y|\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

σ^2 is the variance of the density and μ is the mean.

Gaussian Density



Population of students distributed based on their height.

Two Important Gaussian Properties

Sum of Gaussians

- Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

Two Important Gaussian Properties

Sum of Gaussians

- Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

And the sum is distributed as

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

Two Important Gaussian Properties

Sum of Gaussians

- Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

And the sum is distributed as

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

(*Aside*: As sum increases, sum of non-Gaussian, finite variance variables is also Gaussian [central limit theorem].)

Two Important Gaussian Properties

Scaling a Gaussian

- Scaling a Gaussian leads to a Gaussian.

Two Important Gaussian Properties

Scaling a Gaussian

- Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

Two Important Gaussian Properties

Scaling a Gaussian

- Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

And the scaled density is distributed as

$$wy \sim \mathcal{N}\left(w\mu, w^2\sigma^2\right)$$

Multivariate Consequence

- If

$$\mathbf{x} \sim \mathcal{N}\left(\mu, \Sigma\right)$$

Multivariate Consequence

- If

$$\mathbf{x} \sim \mathcal{N}\left(\mu, \Sigma\right)$$

- And

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

Multivariate Consequence

- If

$$\mathbf{x} \sim \mathcal{N}\left(\mu, \Sigma\right)$$

- And

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

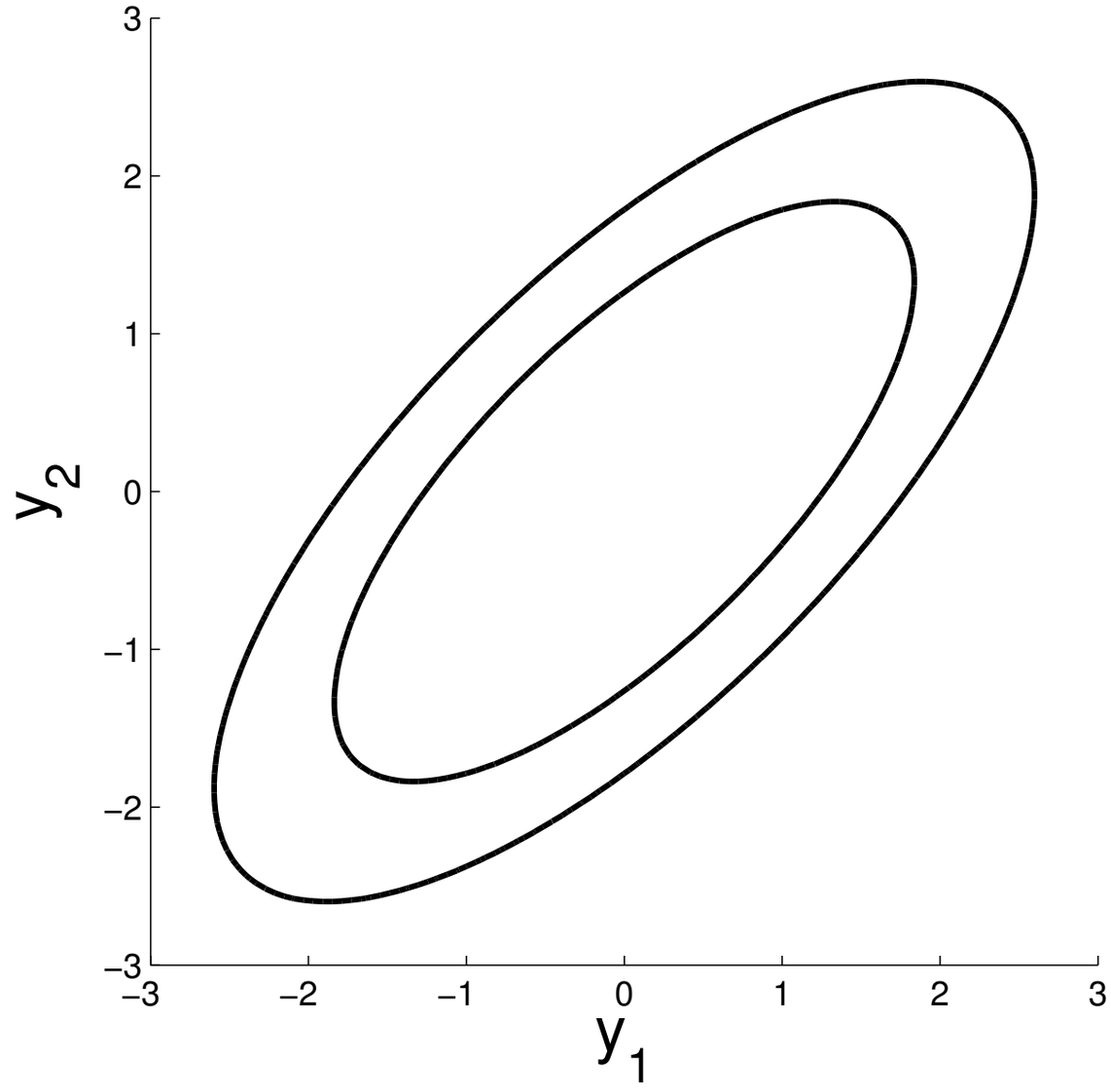
- Then

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{W}\mu, \mathbf{W}\Sigma\mathbf{W}^{\top}\right)$$

Gaussian Distribution

$$\Sigma_{i,j} = \mathbb{E}\left[(y_i - \mathbb{E}(y_i))(y_j - \mathbb{E}(y_j))\right]$$

$$p(\mathbf{y}|\Sigma) \propto \exp\left(-\tfrac{1}{2}\mathbf{y}^\mathsf{T}\Sigma^{-1}\mathbf{y}\right) \qquad \Sigma = \begin{bmatrix} 1 & .7 \\ .7 & 1 \end{bmatrix}$$



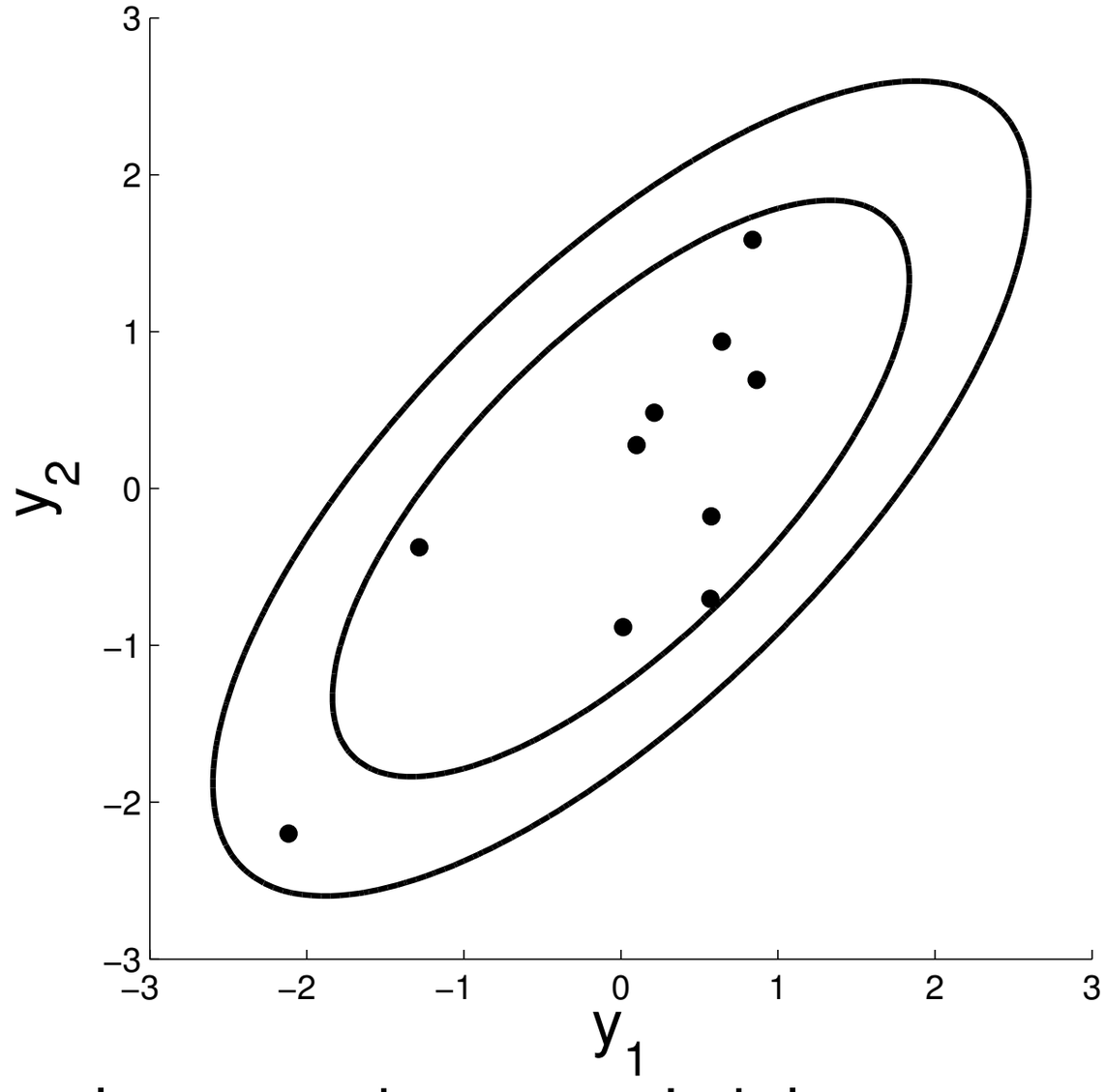
y_i : scalar random variable

$\mathbf{y}$: vector random variable

Gaussian Distribution

$$\Sigma_{i,j} = \mathbb{E}\left[(y_i - \mathbb{E}(y_i))(y_j - \mathbb{E}(y_j))\right]$$

$$p(\mathbf{y}|\Sigma) \propto \exp\left(-\tfrac{1}{2}\mathbf{y}^\mathsf{T}\Sigma^{-1}\mathbf{y}\right) \qquad \Sigma = \begin{bmatrix} 1 & .7 \\ .7 & 1 \end{bmatrix}$$

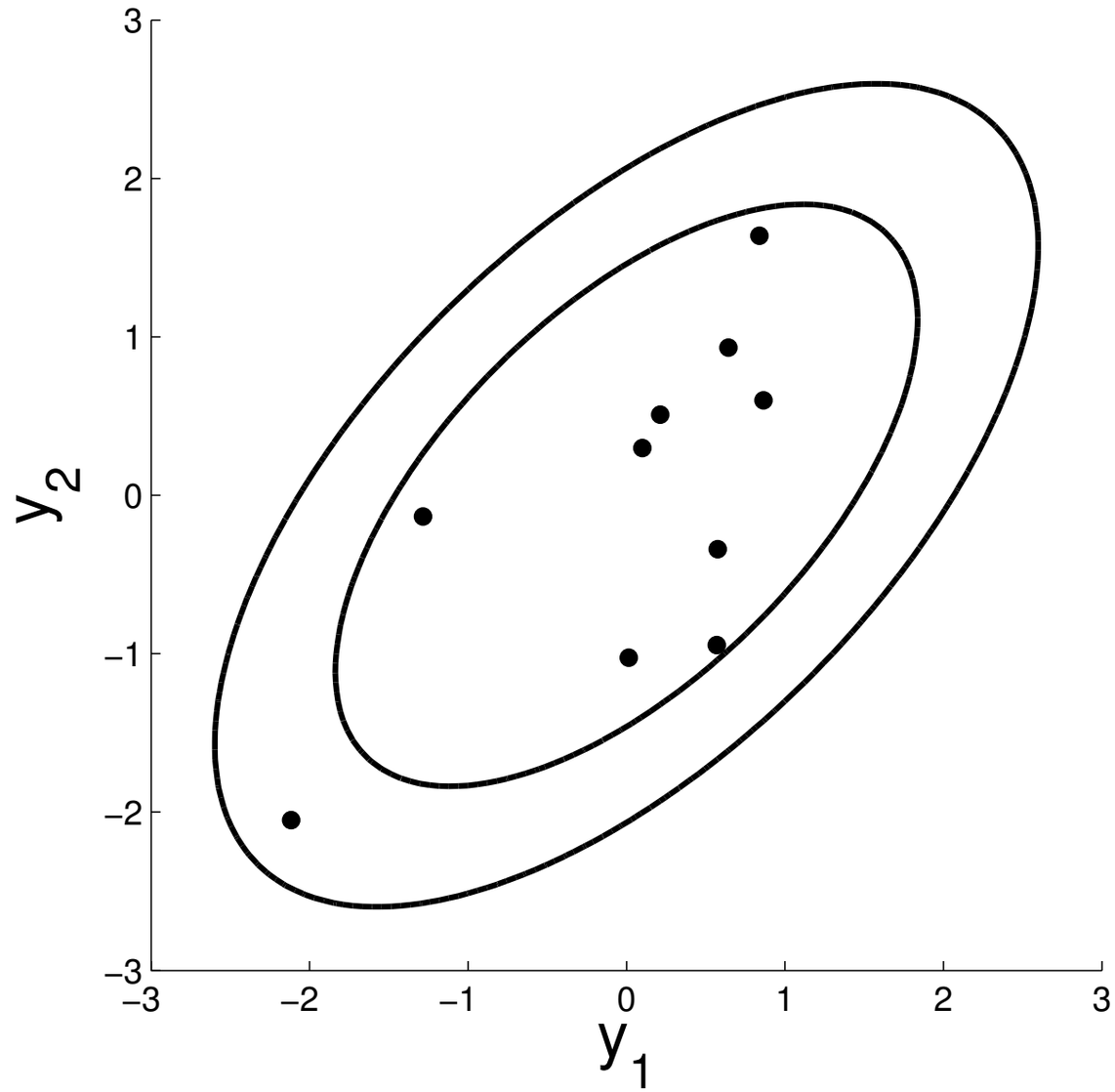


y_i : scalar random variable

$\mathbf{y}$: vector random variable

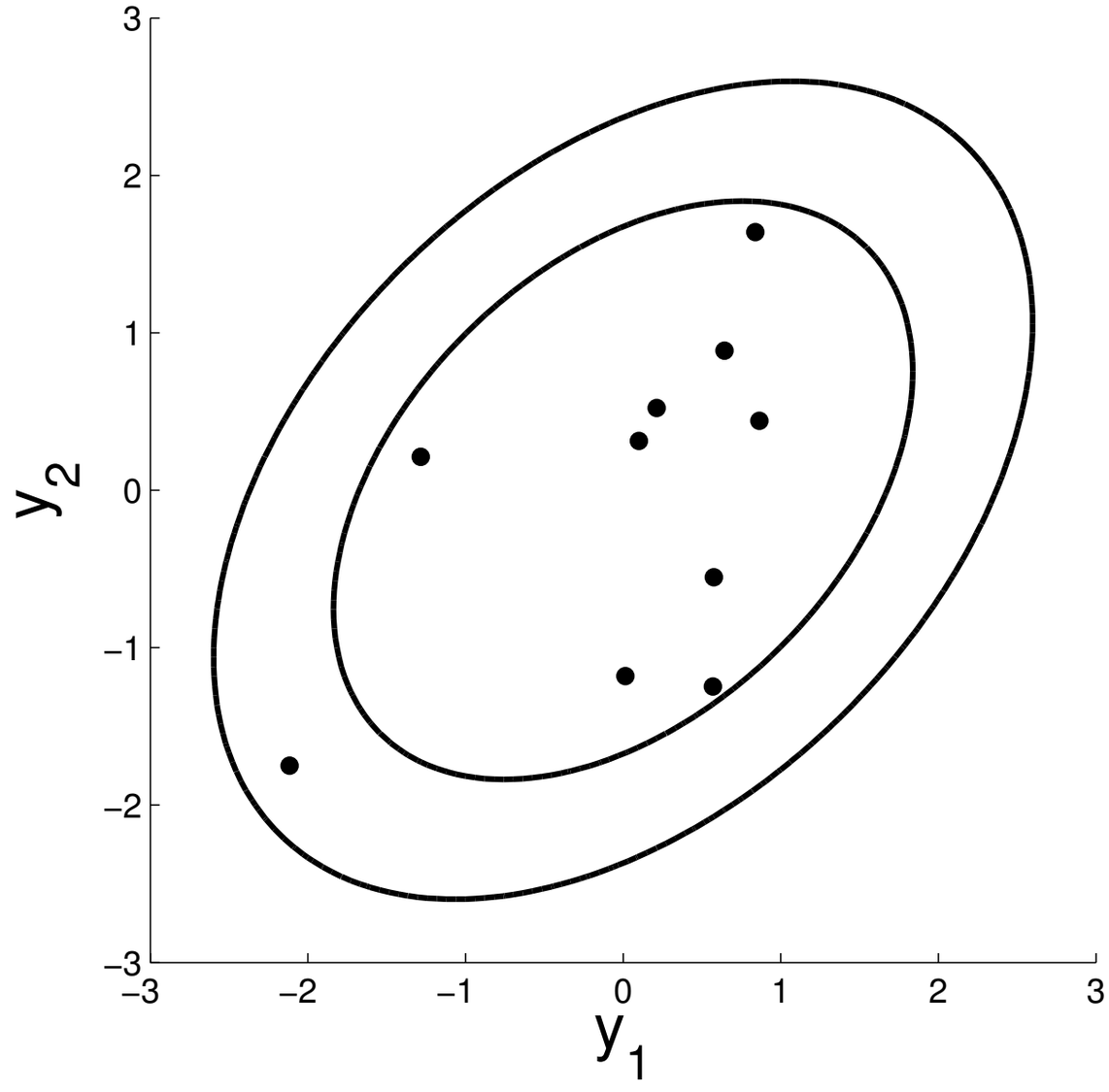
Gaussian Distribution

$$p(\mathbf{y}|\Sigma) \propto \exp\left(-\tfrac{1}{2}\mathbf{y}^\mathsf{T}\Sigma^{-1}\mathbf{y}\right) \qquad \Sigma = \begin{bmatrix} 1 & .6 \\ .6 & 1 \end{bmatrix}$$



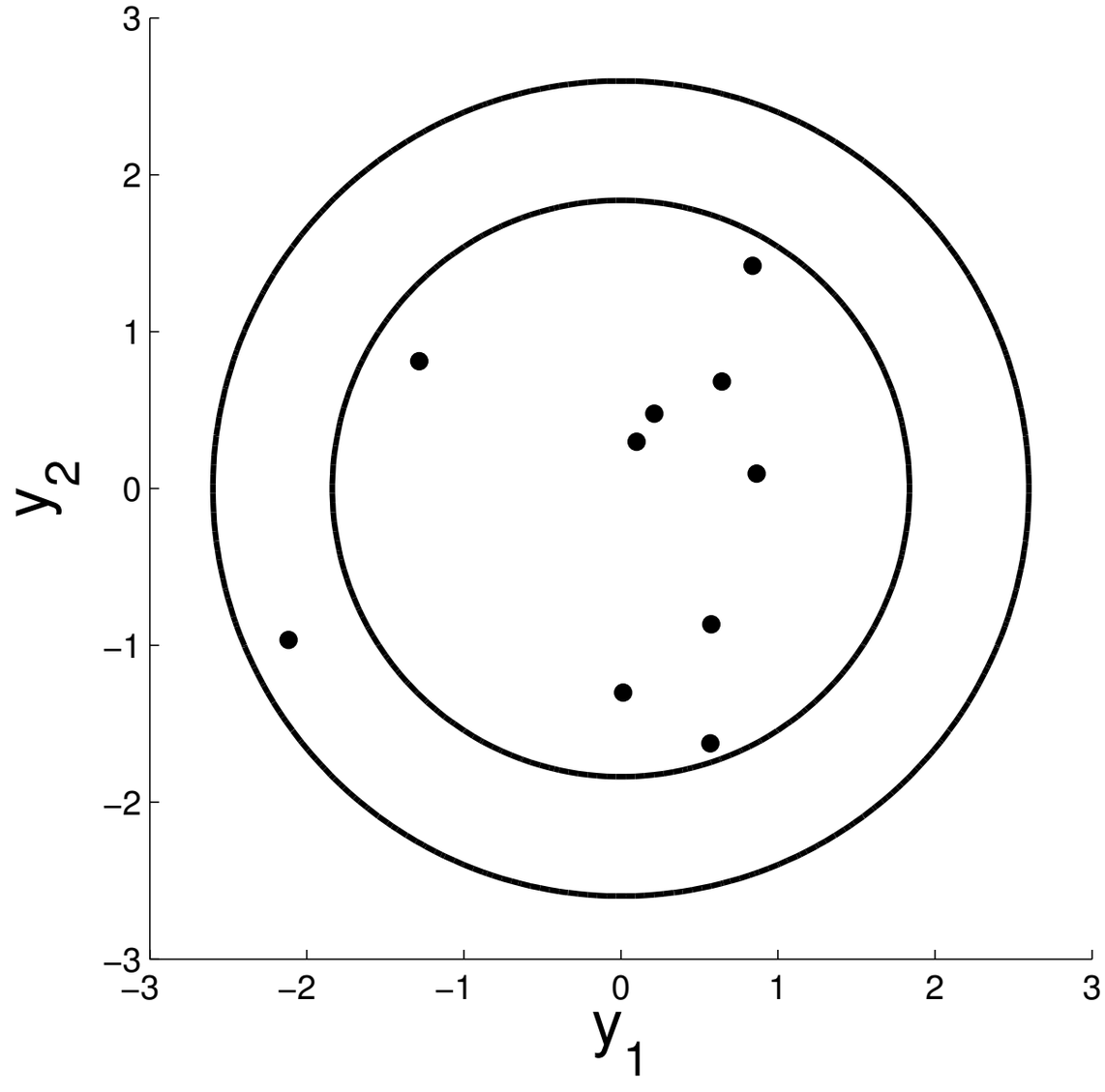
Gaussian Distribution

$$p(\mathbf{y}|\Sigma) \propto \exp\left(-\tfrac{1}{2}\mathbf{y}^\mathsf{T}\Sigma^{-1}\mathbf{y}\right) \qquad \Sigma = \begin{bmatrix} 1 & .4 \\ .4 & 1 \end{bmatrix}$$



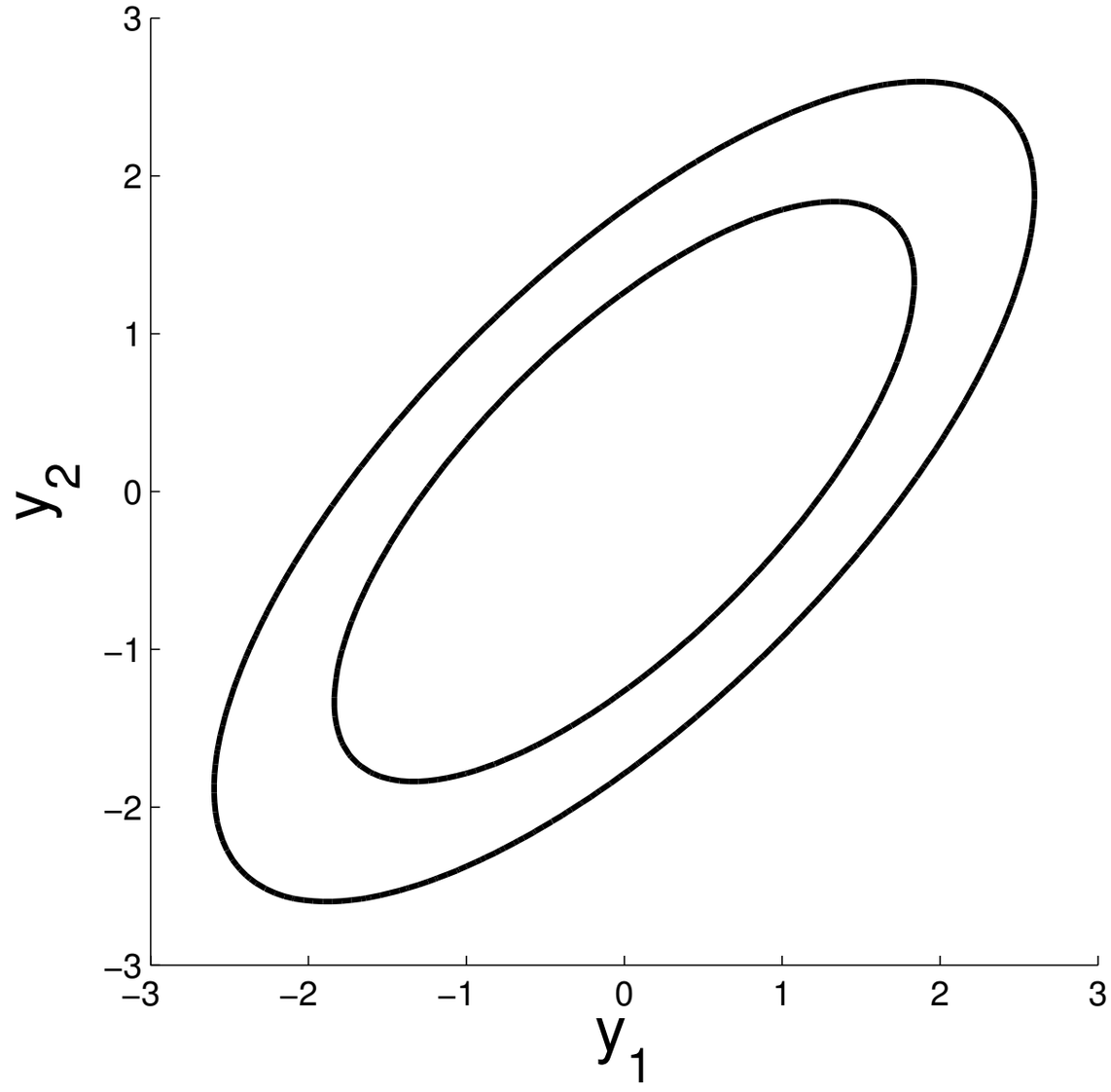
Gaussian Distribution

$$p(\mathbf{y}|\Sigma) \propto \exp\left(-\tfrac{1}{2}\mathbf{y}^{\mathsf{T}}\Sigma^{-1}\mathbf{y}\right) \qquad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



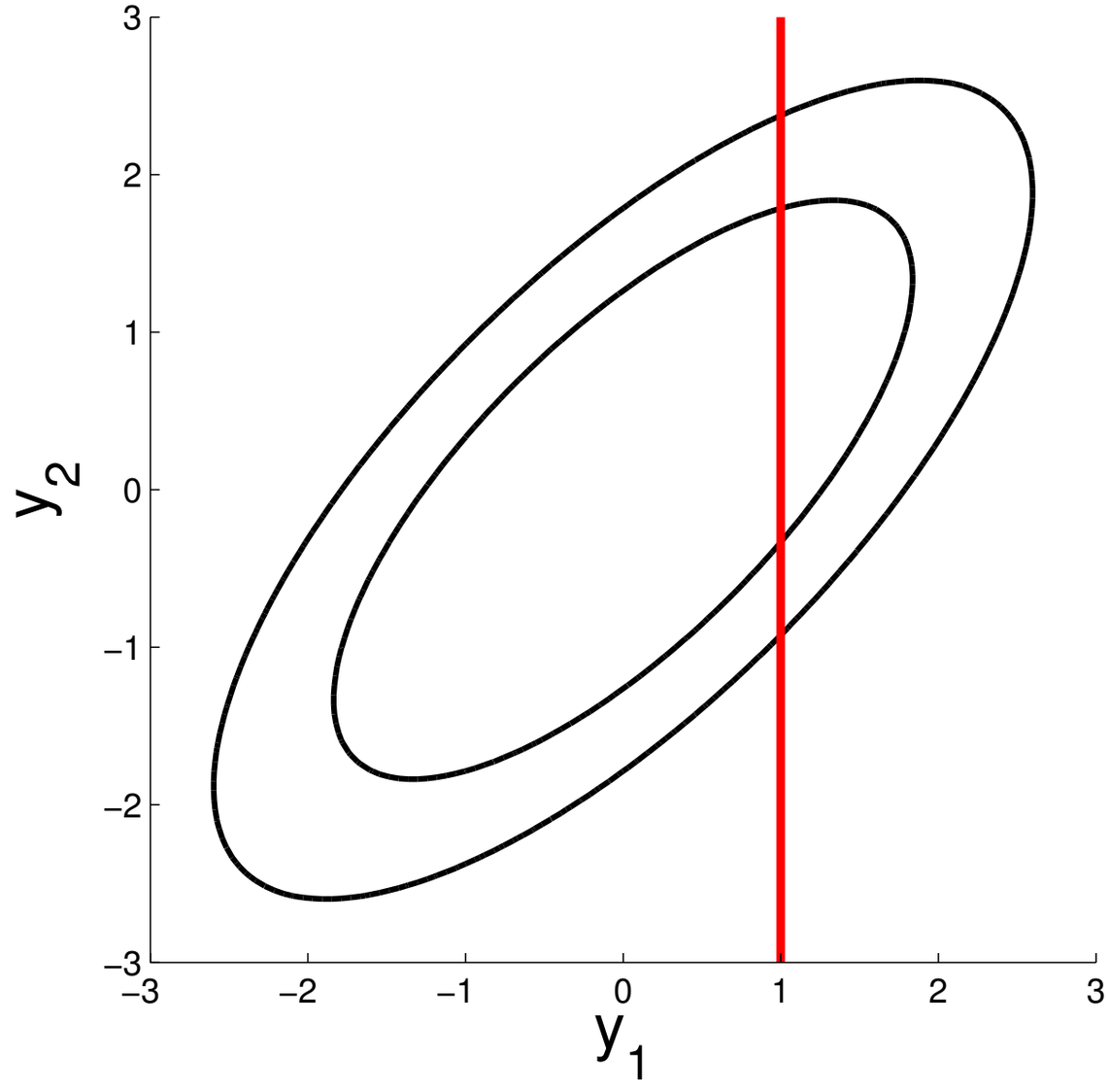
Gaussian Distribution

$$p(\mathbf{y}|\Sigma) \propto \exp\left(-\tfrac{1}{2}\mathbf{y}^\mathsf{T}\Sigma^{-1}\mathbf{y}\right) \qquad \Sigma = \begin{bmatrix} 1 & .7 \\ .7 & 1 \end{bmatrix}$$



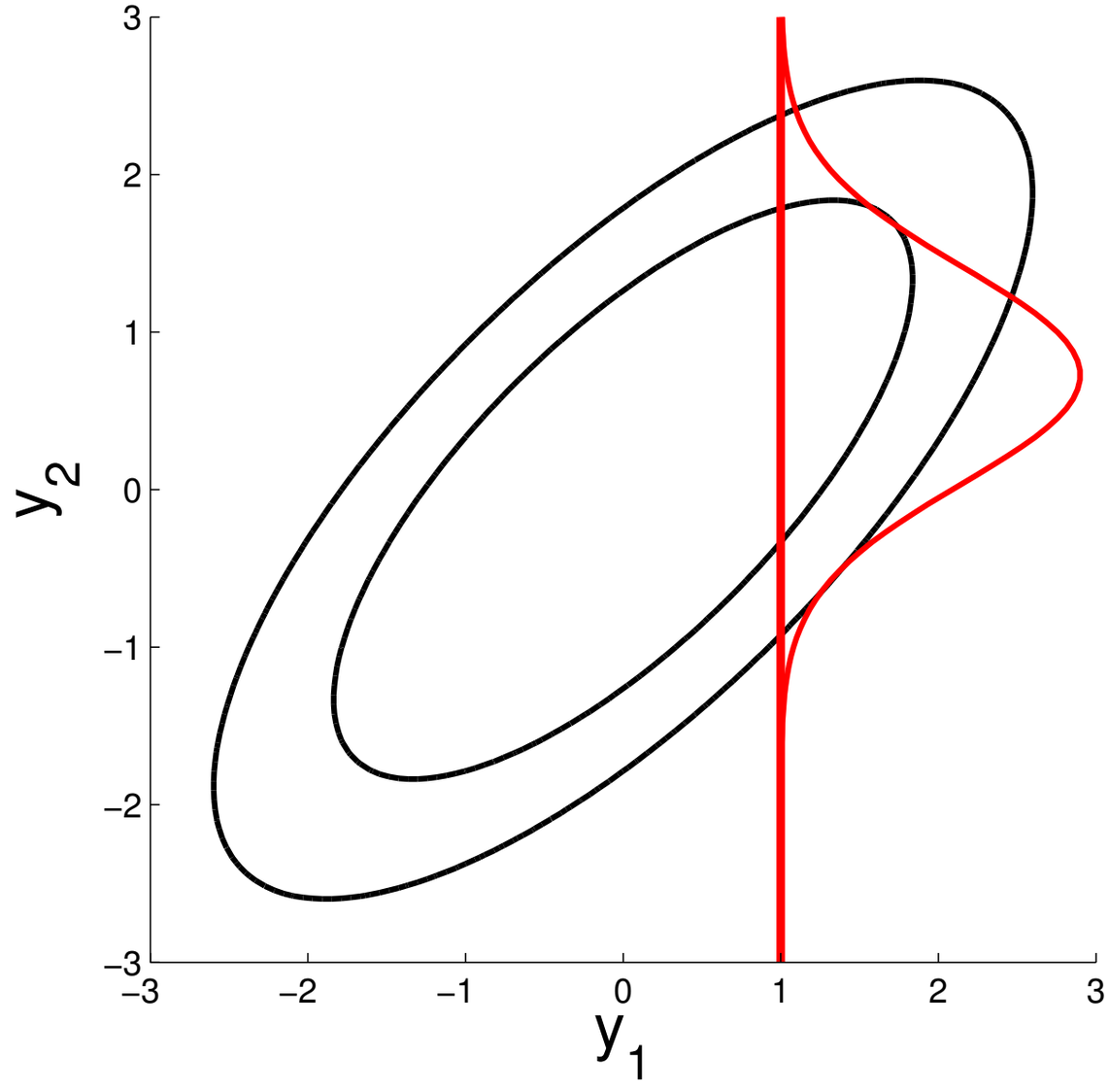
Gaussian distribution - Conditioning

$$p(\mathbf{y}|\Sigma) \propto \exp\left(-\tfrac{1}{2}\mathbf{y}^\intercal\Sigma^{-1}\mathbf{y}\right) \qquad \Sigma = \begin{bmatrix} 1 & .7 \\ .7 & 1 \end{bmatrix}$$



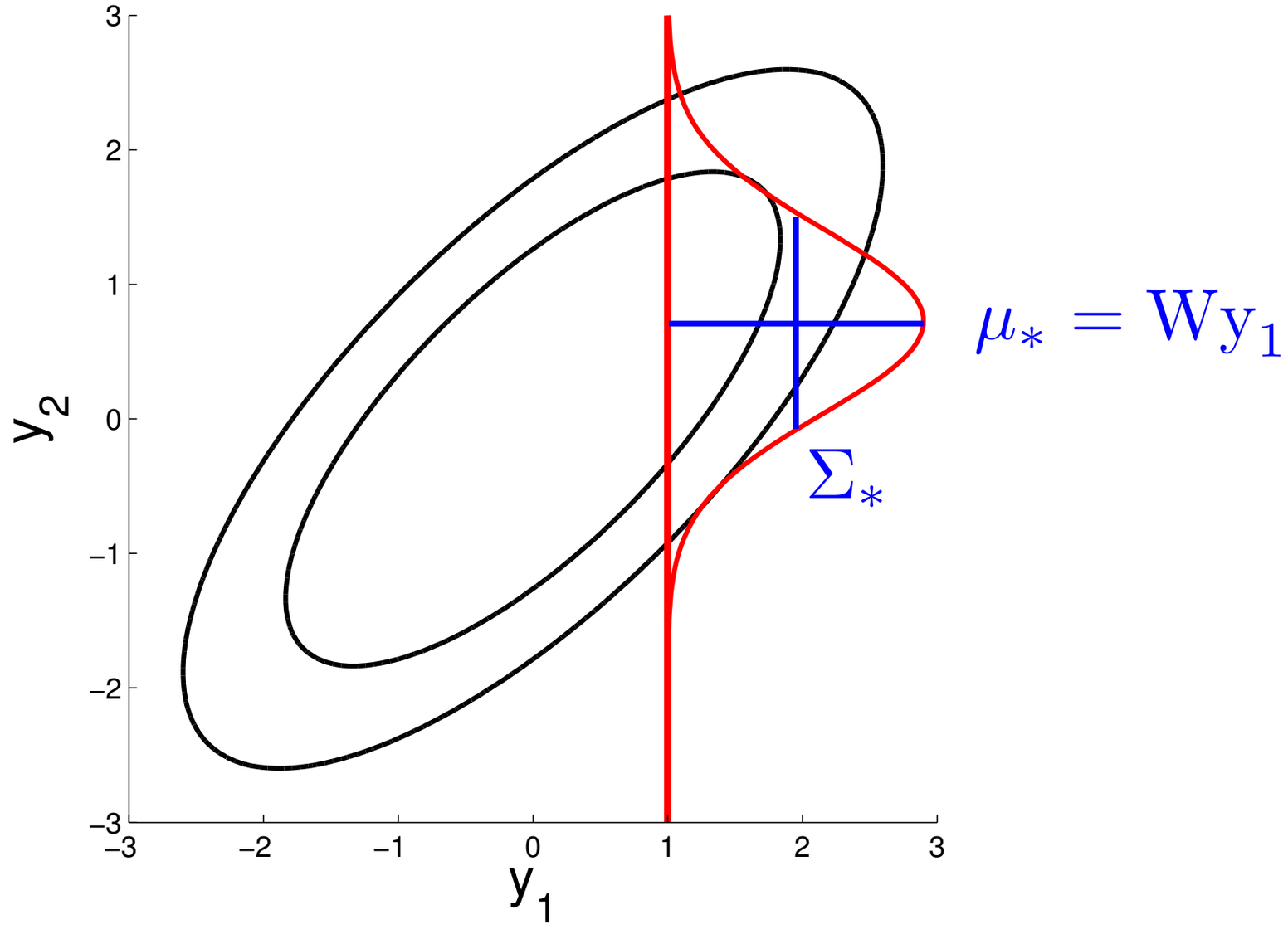
Gaussian distribution - Conditioning

$$p(\mathbf{y}_2|\mathbf{y}_1, \Sigma) \propto \exp\left(-\tfrac{1}{2}(\mathbf{y}_2 - \mu_*){\Sigma_*}^{-1}(\mathbf{y}_2 - \mu_*)\right)$$



Gaussian distribution - Conditioning

$$p(\mathbf{y}_2|\mathbf{y}_1, \Sigma) \propto \exp\left(-\tfrac{1}{2}(\mathbf{y}_2 - \mu_*){\Sigma_*}^{-1}(\mathbf{y}_2 - \mu_*)\right)$$



There are closed form solutions for μ_*, Σ_*

Multivariate Gaussian Theorem

Theorem 4.2.1 (Marginals and conditionals of an MVN). *Suppose* $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ *is jointly Gaussian with parameters*

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix} \tag{4.12}$$

Then the marginals are given by

$$\begin{aligned} p(\mathbf{x}_1) &= \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ p(\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_2 | \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \end{aligned}$$

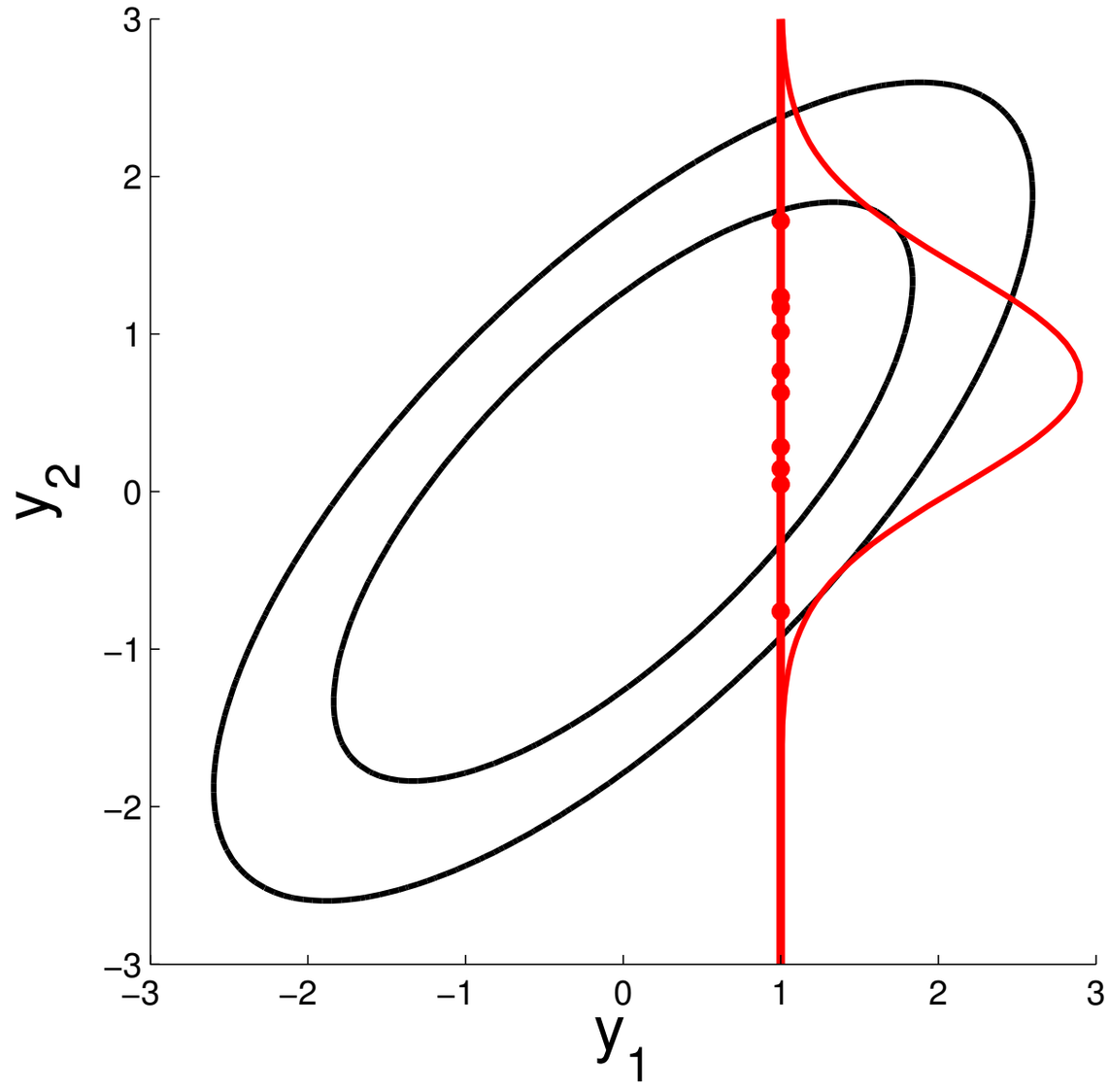
and the posterior conditional is given by

$$\begin{aligned} p(\mathbf{x}_1|\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\Sigma}_{1|2}\left(\boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2)\right) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1} \end{aligned}$$

Machine Learning: A Probabilistic Perspective, Kevin Murphy

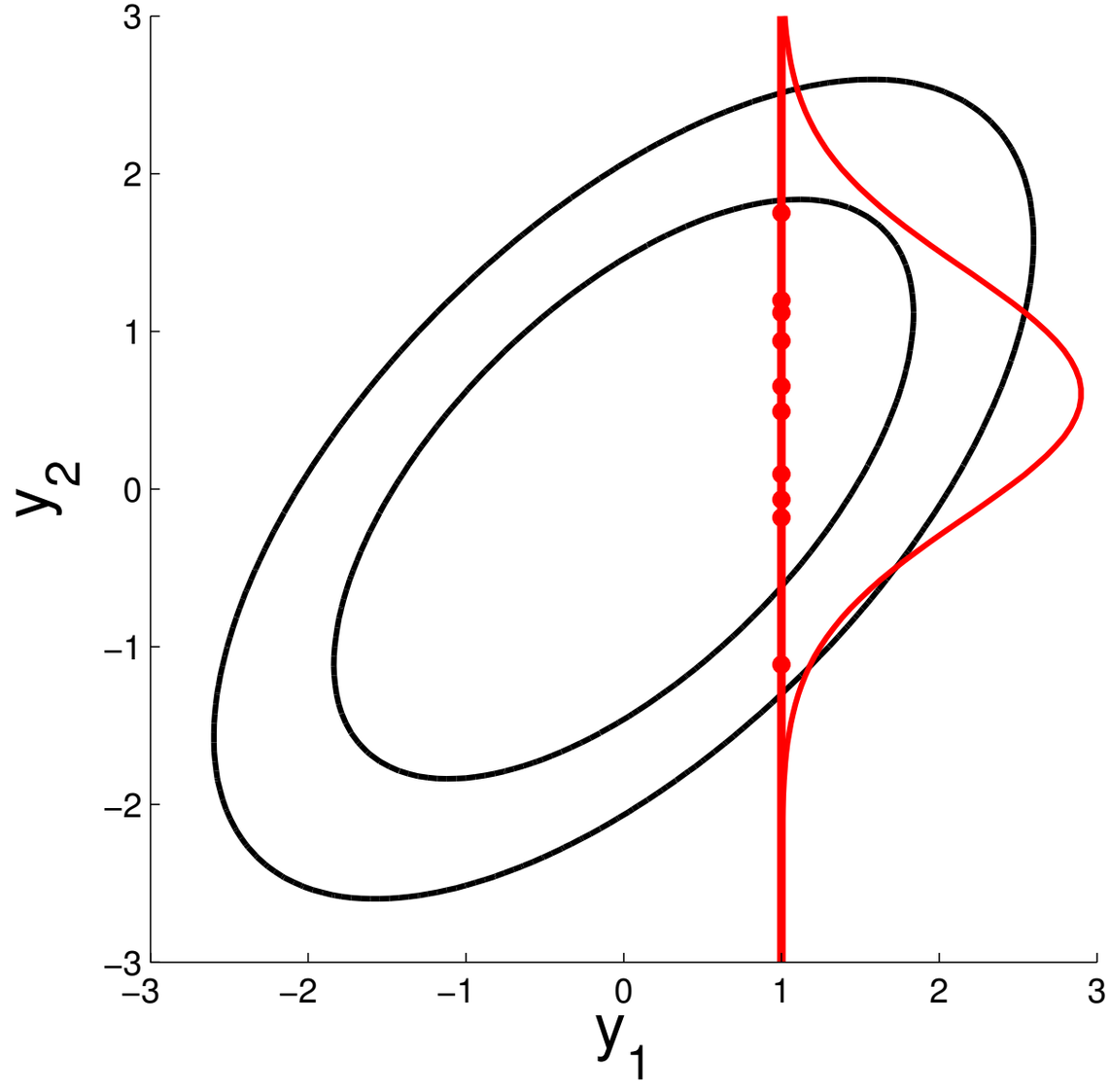
Gaussian distribution - Conditioning

$$p(\mathbf{y}_2|\mathbf{y}_1, \Sigma) \propto \exp\left(-\tfrac{1}{2}(\mathbf{y}_2 - \mu_*)\Sigma_*{}^{-1}(\mathbf{y}_2 - \mu_*)\right)$$



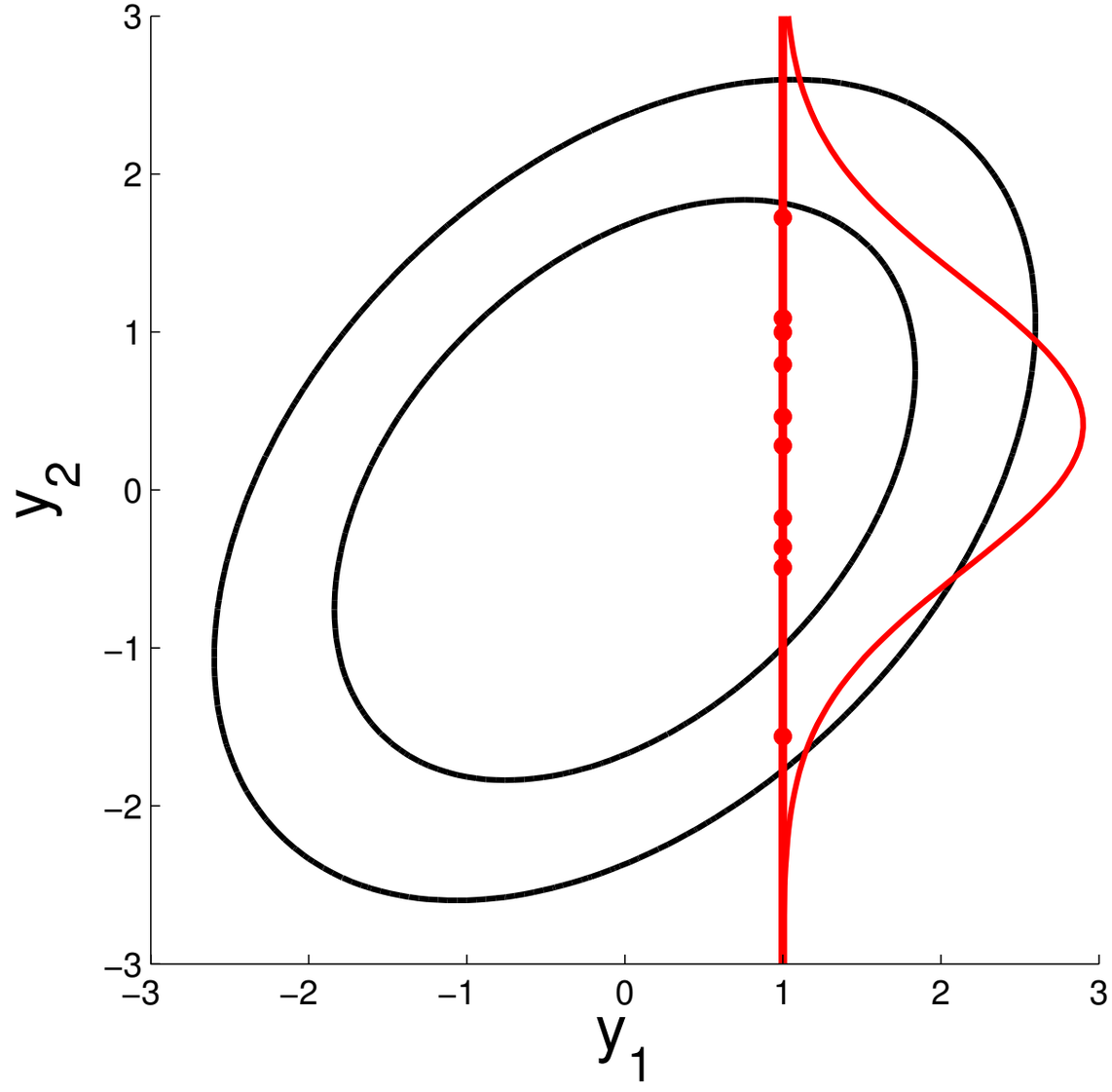
Gaussian distribution - Conditioning

$$p(\mathsf{y}_2|\mathsf{y}_1, \Sigma) \propto \exp\left(-\tfrac{1}{2}(\mathsf{y}_2 - \mu_*){\Sigma_*}^{-1}(\mathsf{y}_2 - \mu_*)\right)$$



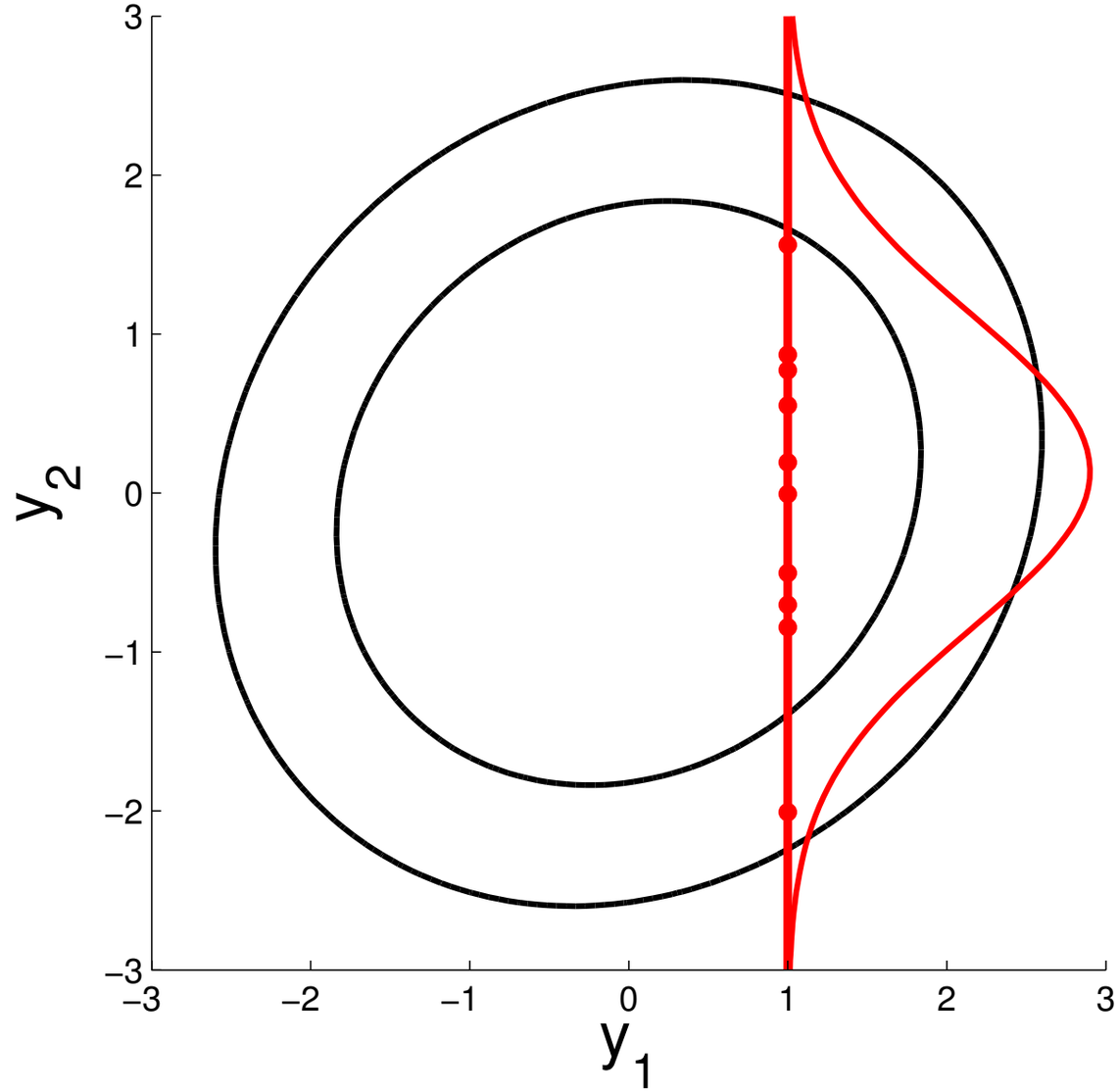
Gaussian distribution - Conditioning

$$p(\mathbf{y}_2|\mathbf{y}_1, \Sigma) \propto \exp\left(-\tfrac{1}{2}(\mathbf{y}_2 - \mu_*){\Sigma_*}^{-1}(\mathbf{y}_2 - \mu_*)\right)$$



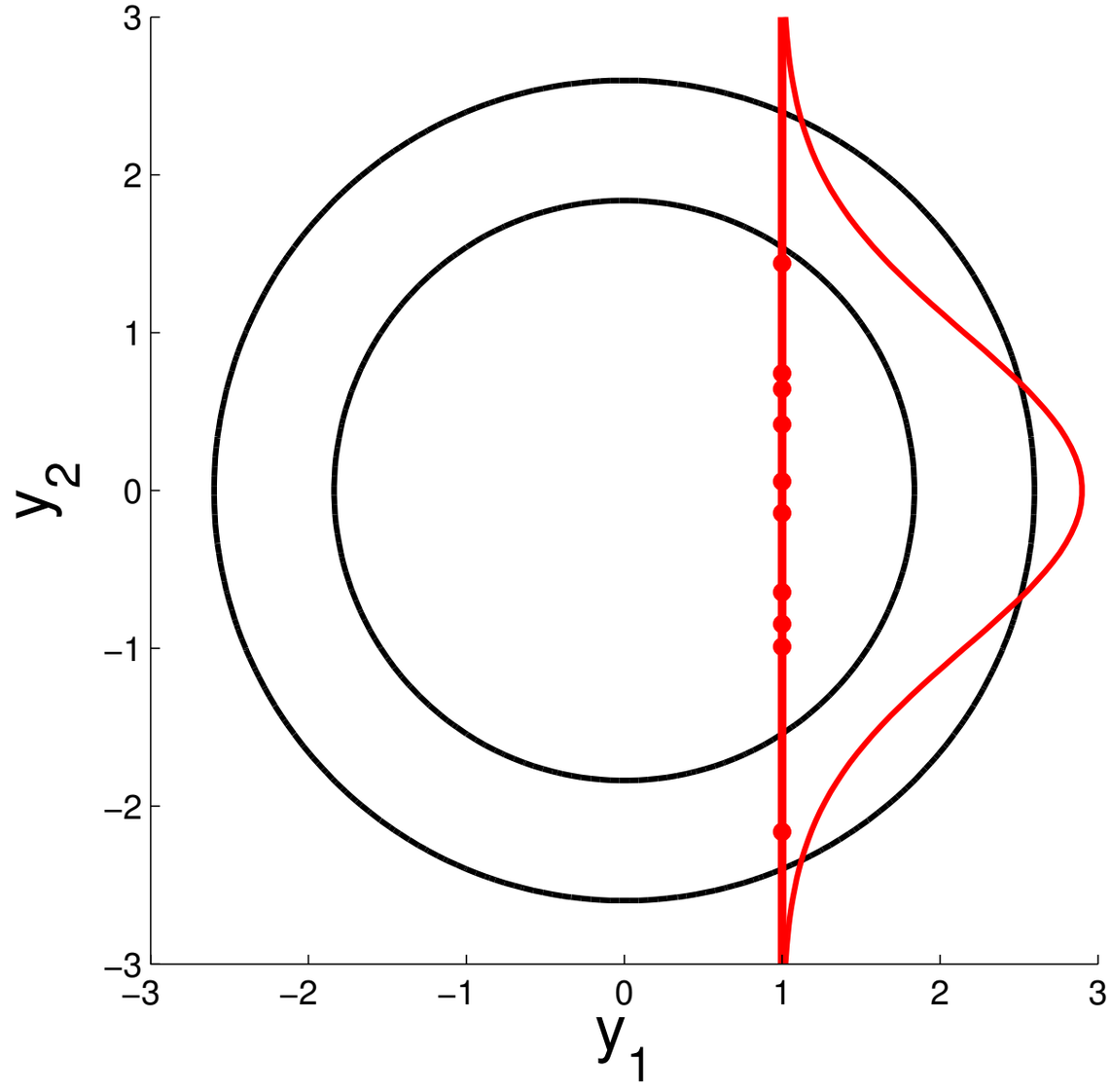
Gaussian distribution - Conditioning

$$p(\mathsf{y}_2|\mathsf{y}_1, \Sigma) \propto \exp\left(-\tfrac{1}{2}(\mathsf{y}_2 - \mu_*){\Sigma_*}^{-1}(\mathsf{y}_2 - \mu_*)\right)$$



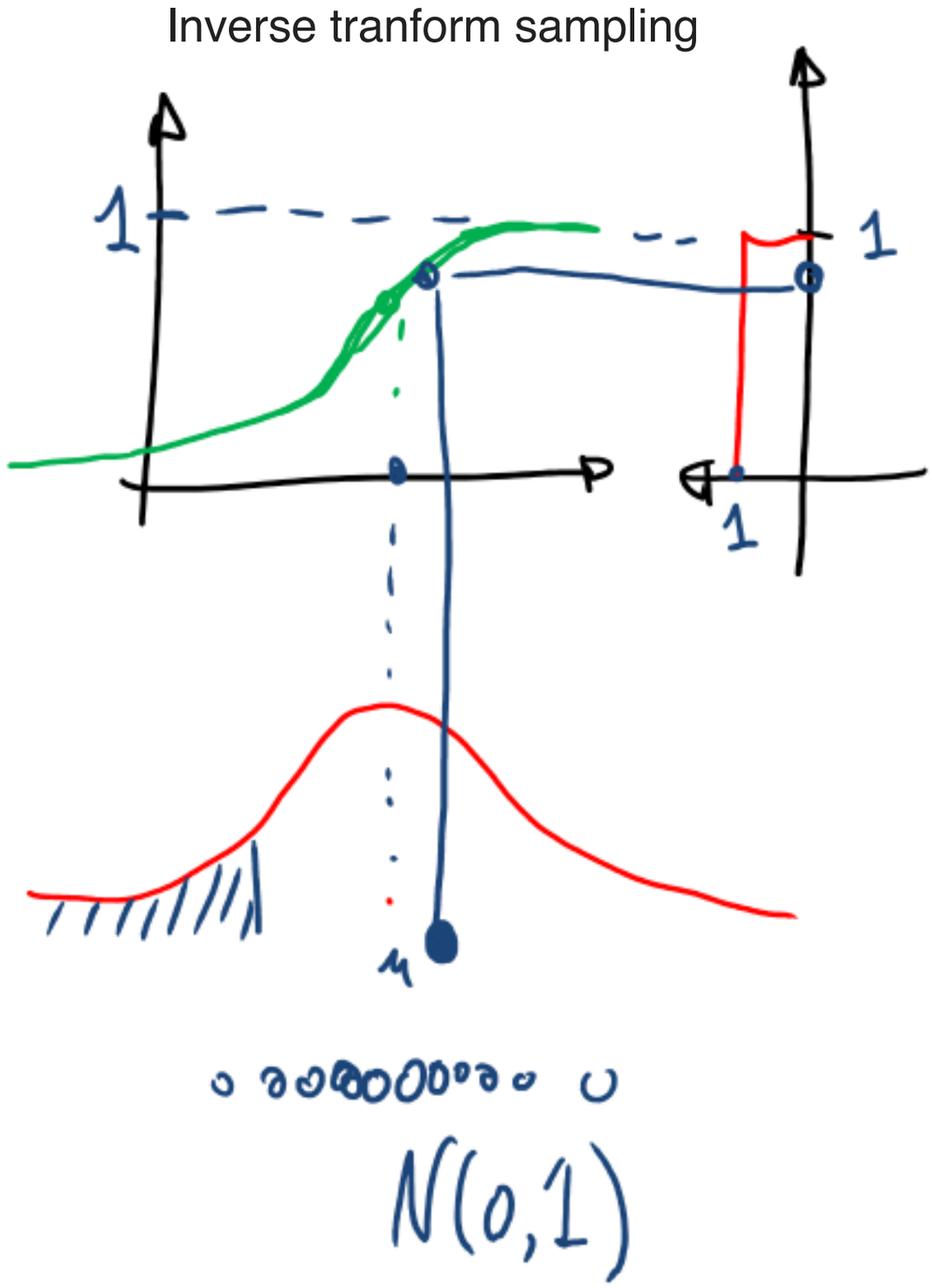
Gaussian distribution - Conditioning

$$p(\mathbf{y}_2|\mathbf{y}_1, \Sigma) \propto \exp\left(-\tfrac{1}{2}(\mathbf{y}_2 - \mu_*){\Sigma_*}^{-1}(\mathbf{y}_2 - \mu_*)\right)$$



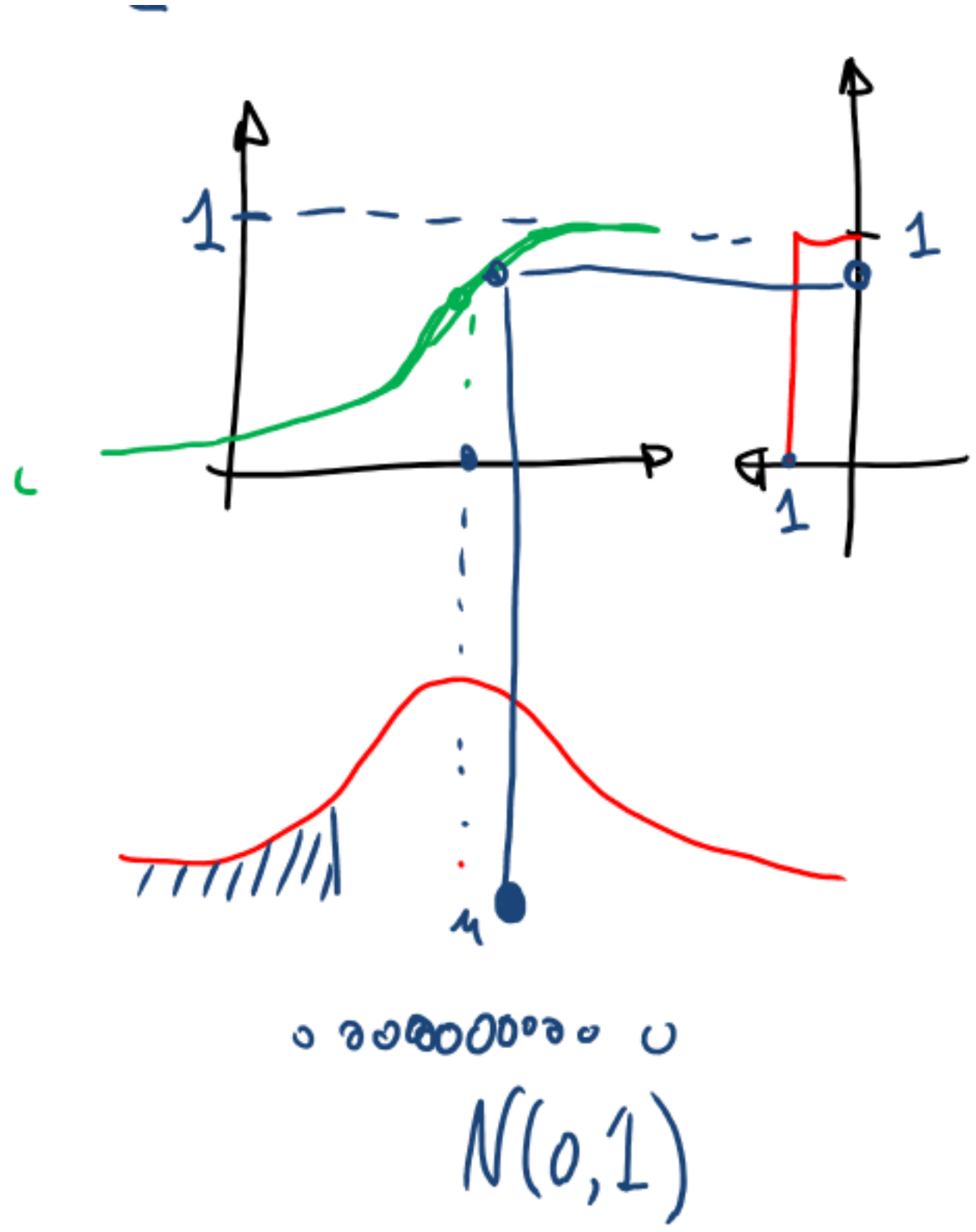
Sampling from $\mathcal{N}(0,1)$

- Integrate the Gaussian density to compute the Cumulative Density Function (CDF) F(x).
- Invert the function F(x). The resulting function is the inverse cumulative distribution function or quantile function F-1(x).
- Substitute the value of the uniformly distributed random number U into the inverse normal CDF.



Sampling from a Gaussian density

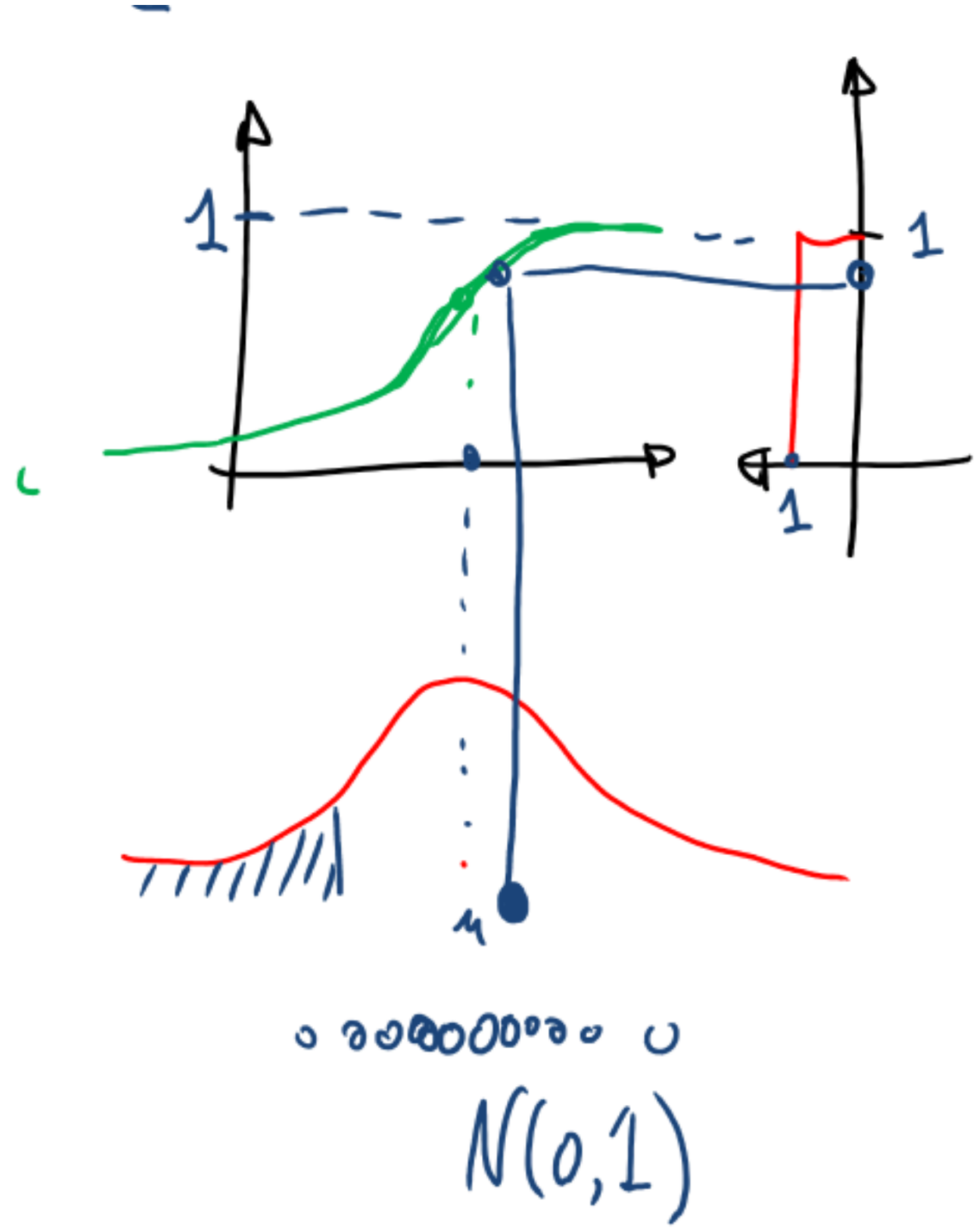
$$x_i \sim \mathcal{N}(0,1)$$



Sampling from a Gaussian density

$$x_i \sim \mathcal{N}(0,1)$$

$$x_i \sim \mathcal{N}(\mu, \sigma^2)$$
$$\sim \mu + \sigma\mathcal{N}(0,1)$$



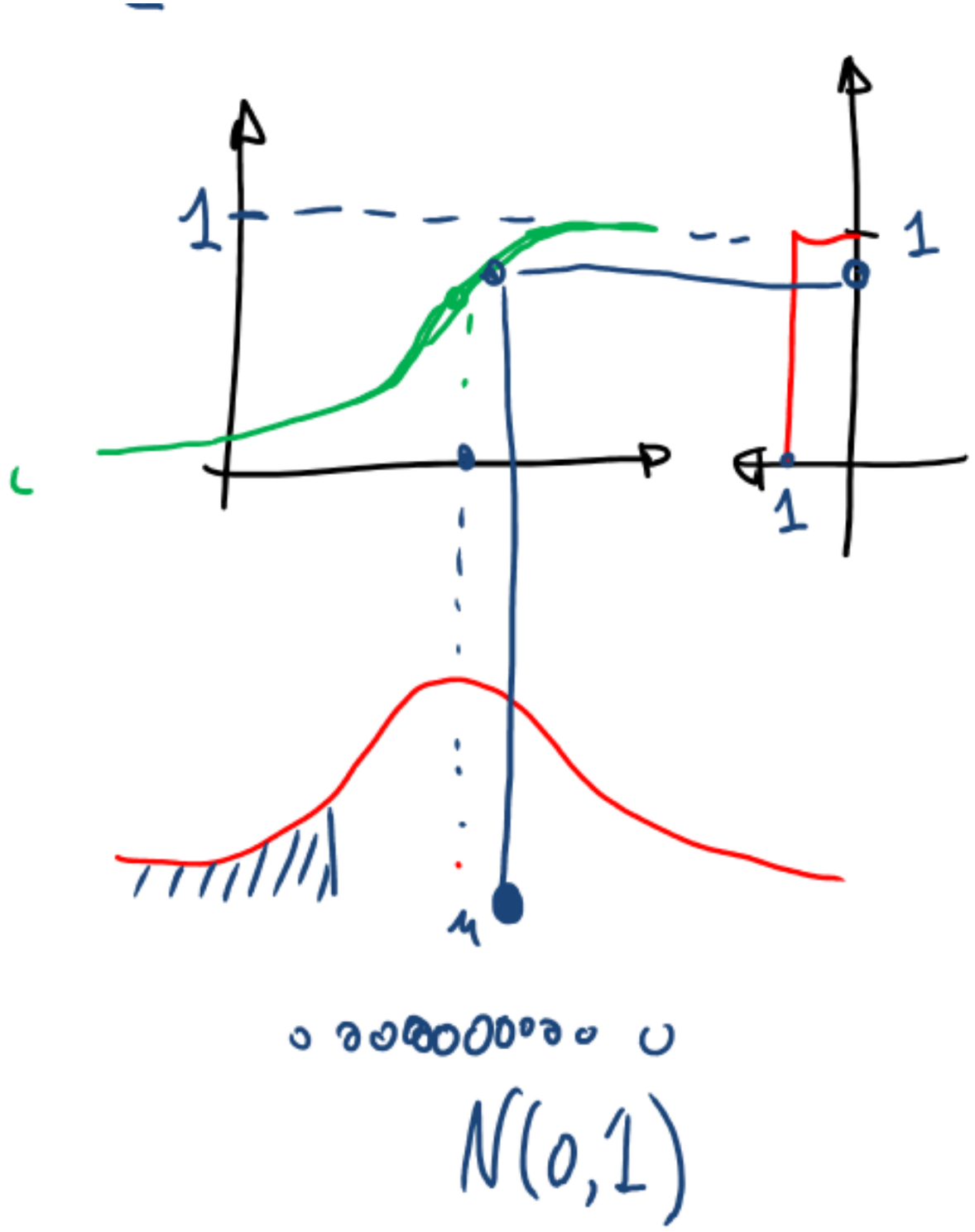
Sampling from a Gaussian density

$$x_i \sim \mathcal{N}(0,1)$$

$$x_i \sim \mathcal{N}(\mu, \sigma^2)$$
$$\sim \mu + \sigma\mathcal{N}(0,1)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$



Sampling from a general Gaussian density

$$x_i \sim \mathcal{N}(0,1)$$

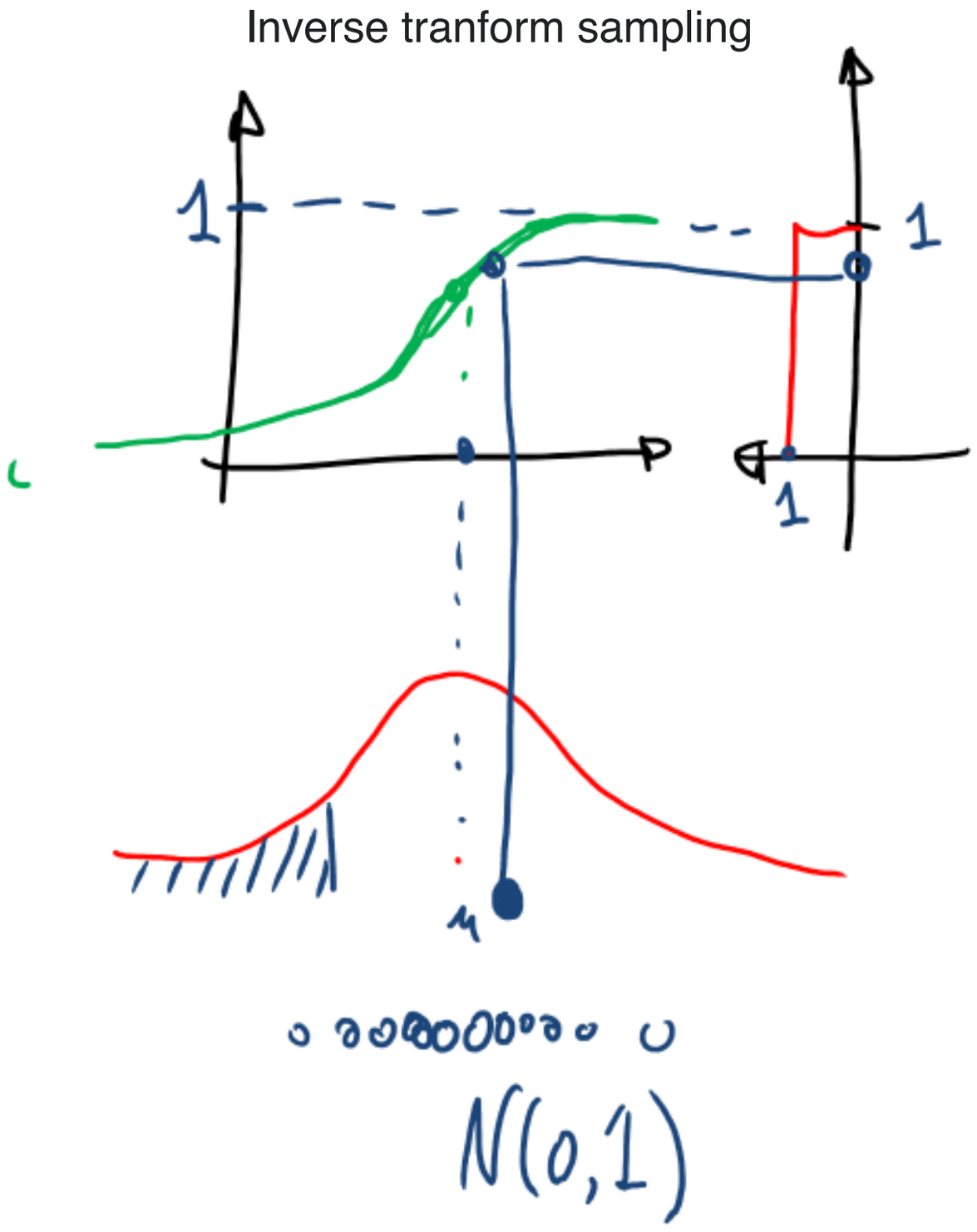
$$x_i \sim \mathcal{N}(\mu, \sigma^2)$$
$$\sim \mu + \sigma\mathcal{N}(0,1)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$

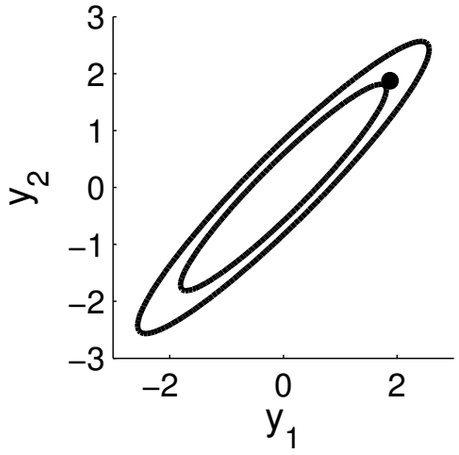
$\mathbf{x}$ μ Σ

$$\mathbf{x} \sim \mathcal{N}(\mu, \Sigma) \qquad x \sim \mu + L\mathcal{N}(0,I)$$



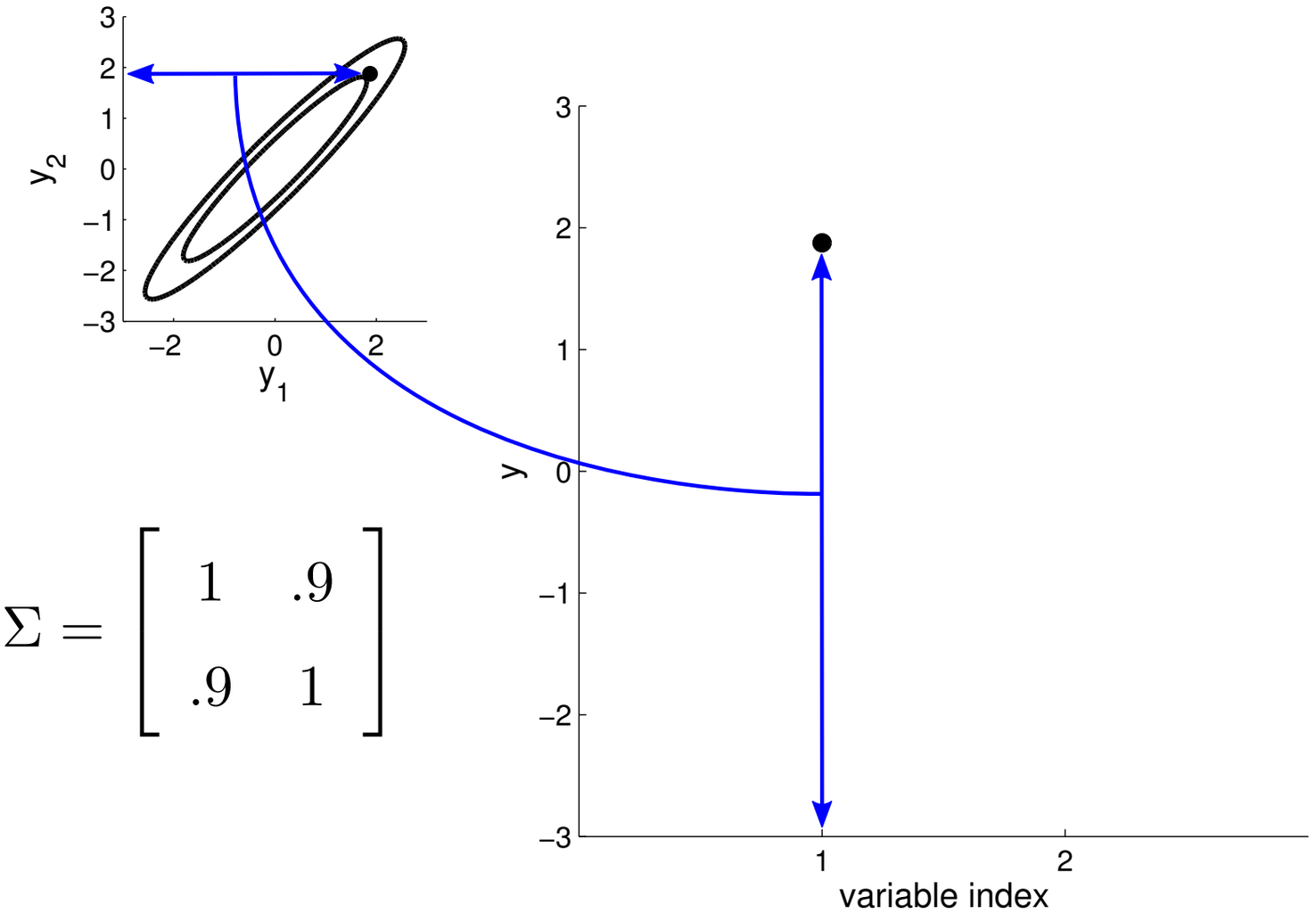
Cholesky decomposition $\Sigma = LL^T$

Towards higher dimensional Gaussians - New Visualization



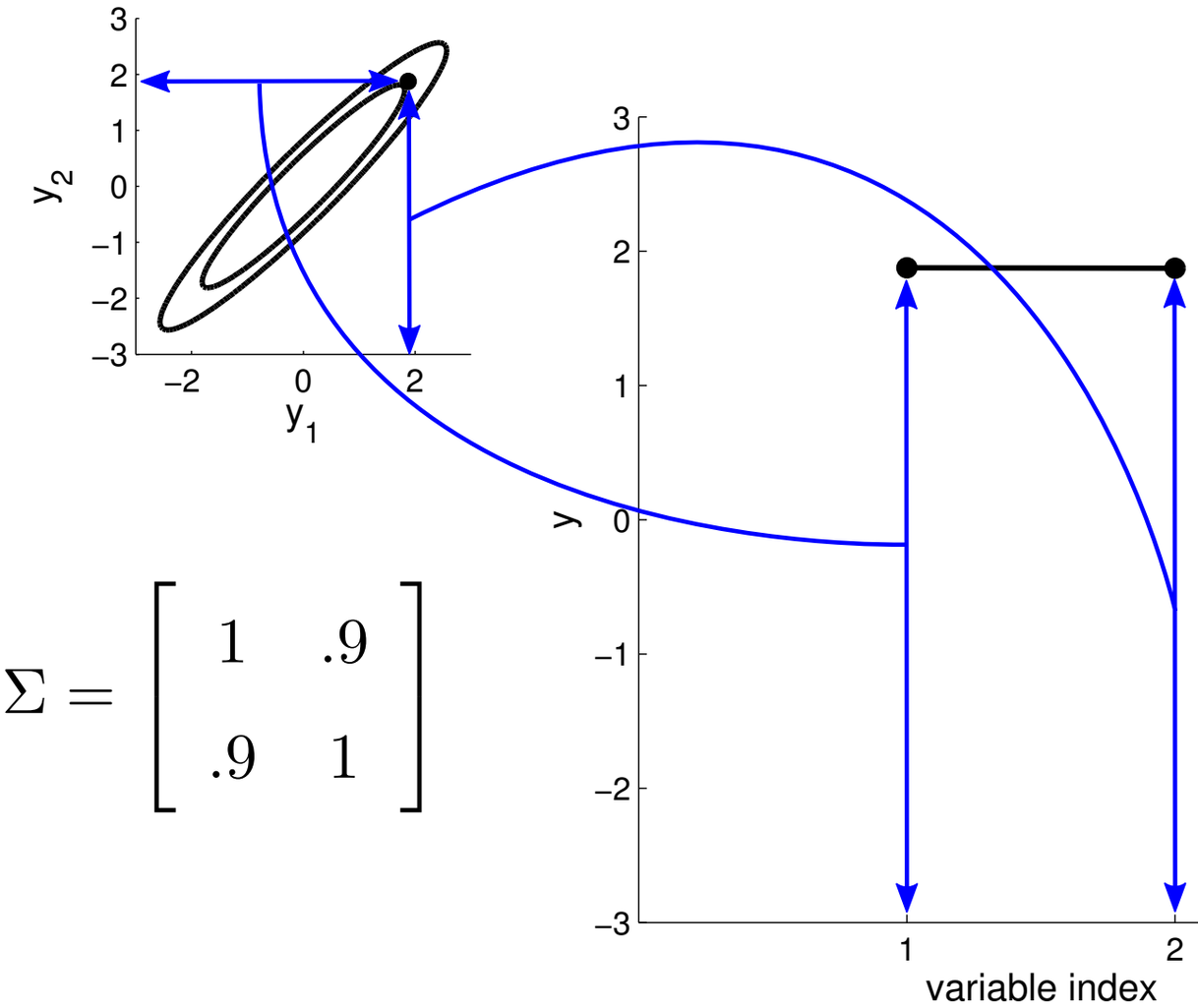
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$

New Visualisation



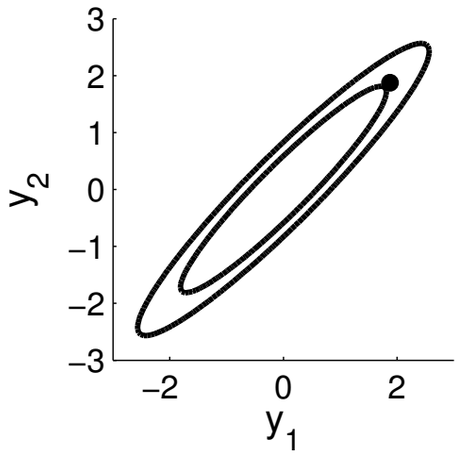
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$

New Visualisation

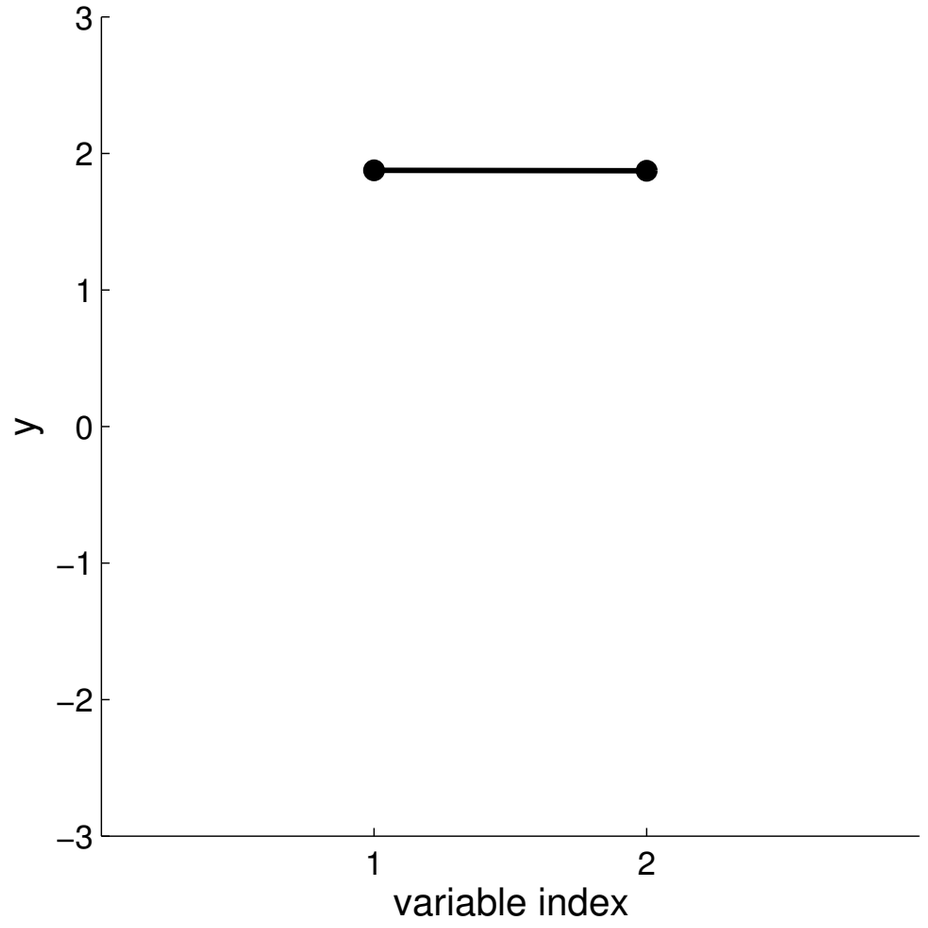


$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$

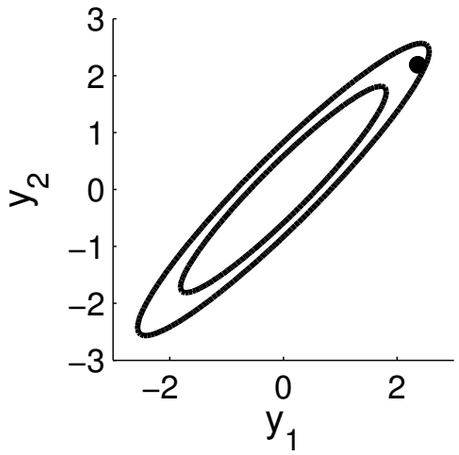
New Visualisation



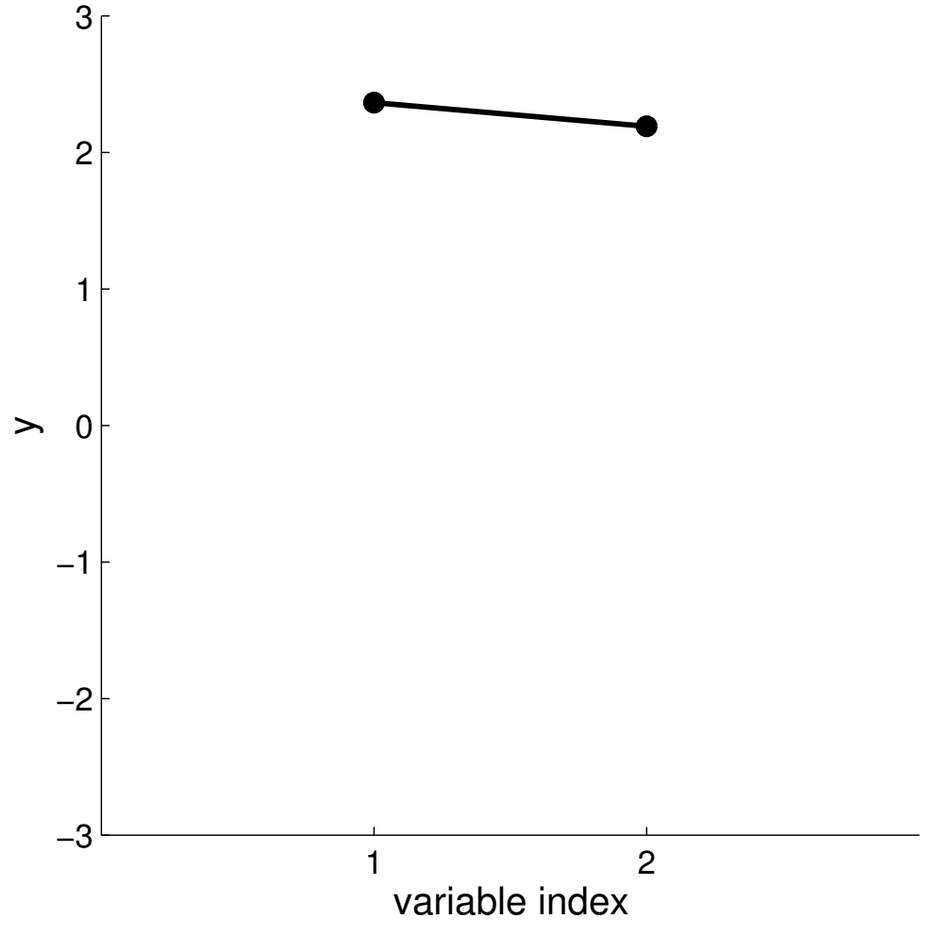
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



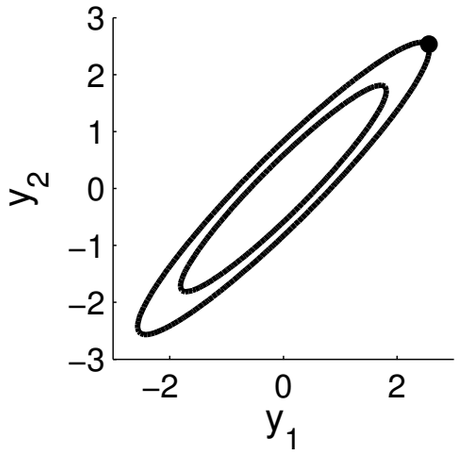
New Visualisation



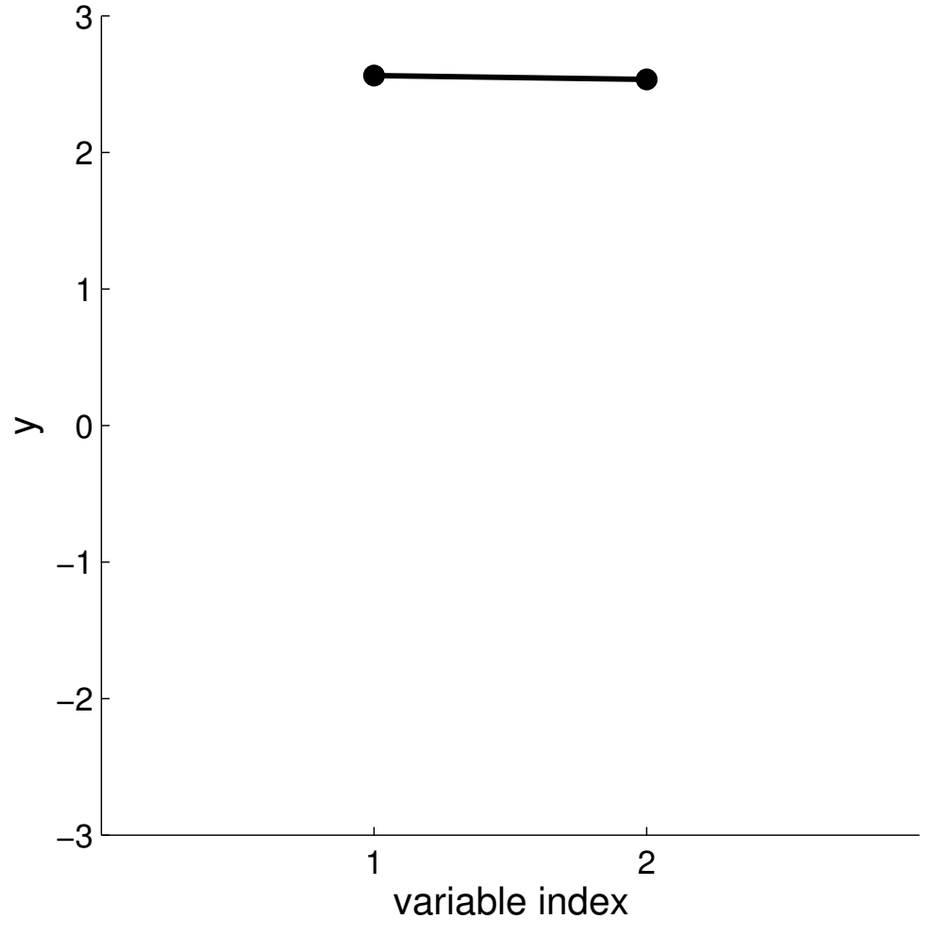
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



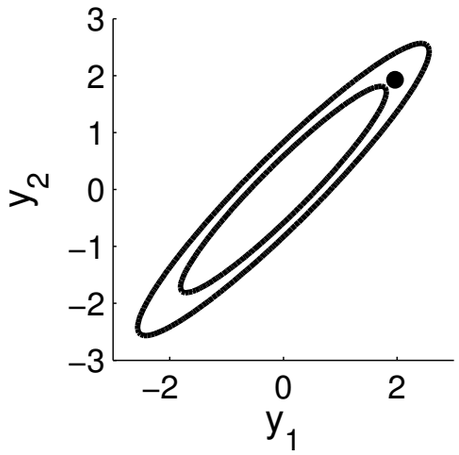
New Visualisation



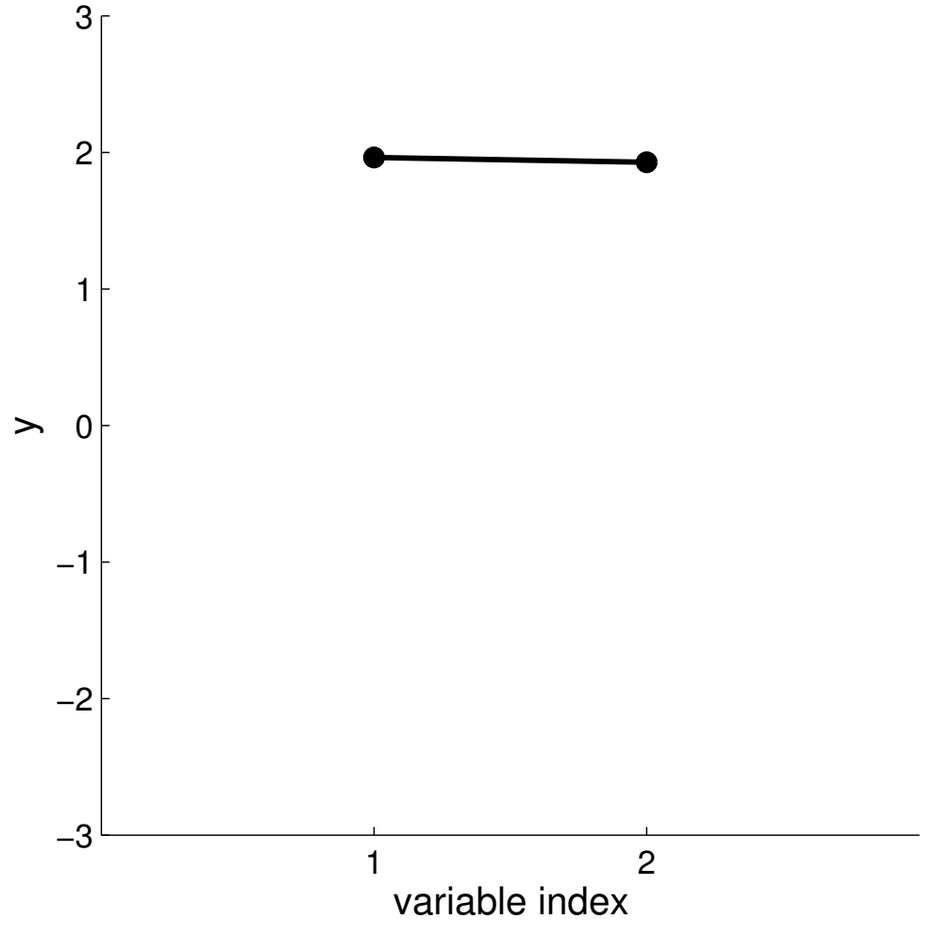
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



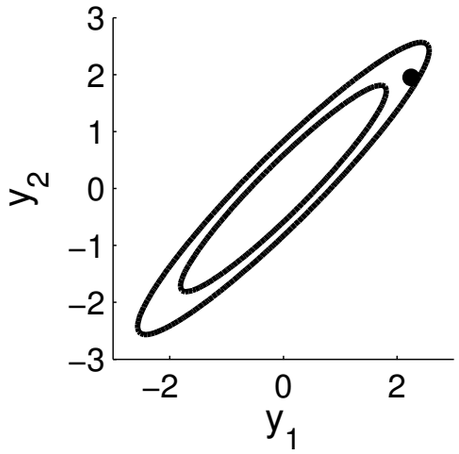
New Visualisation



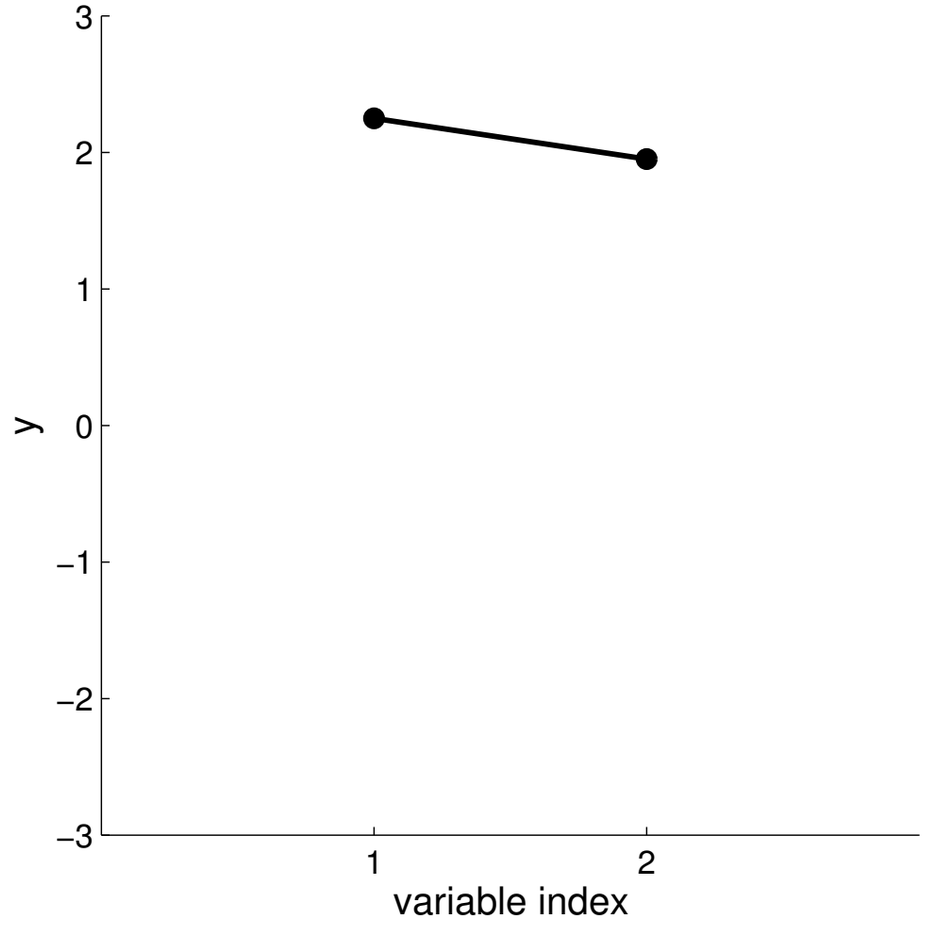
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



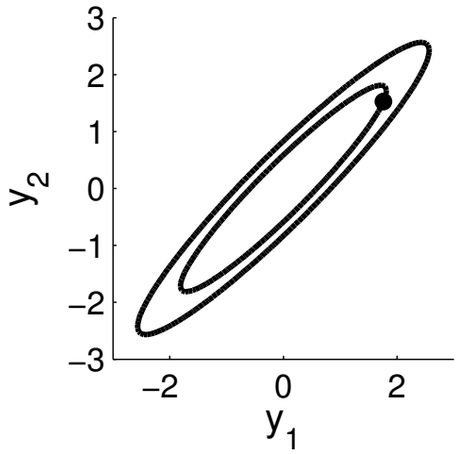
New Visualisation



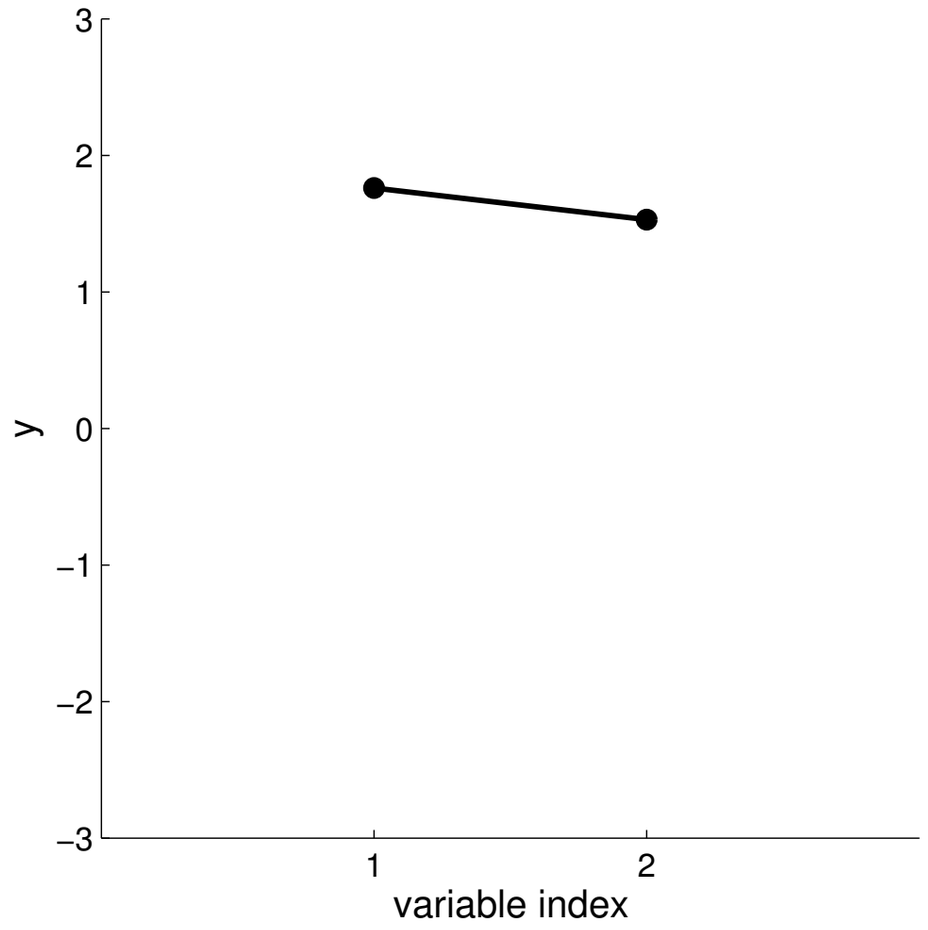
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



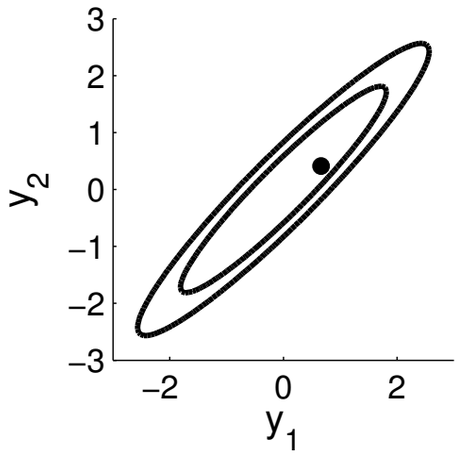
New Visualisation



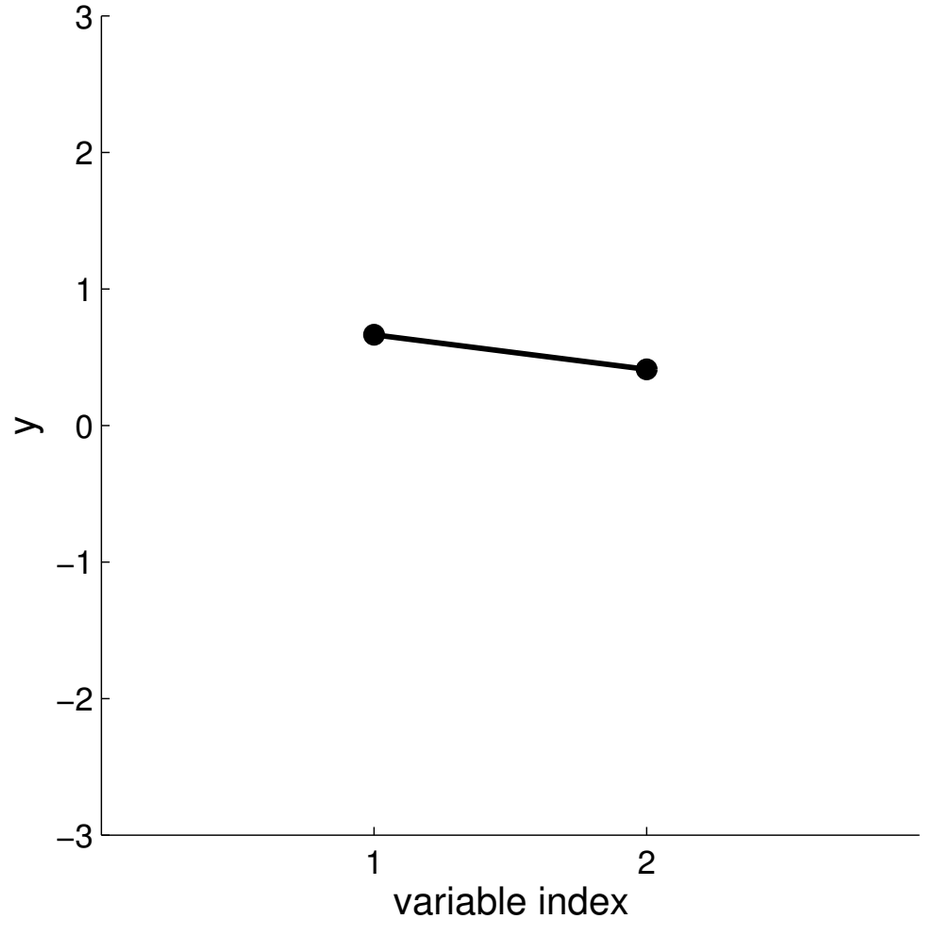
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



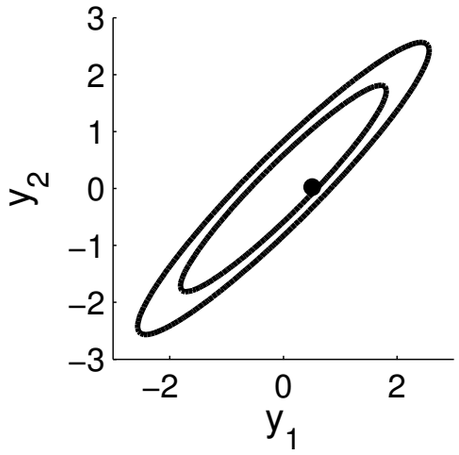
New Visualisation



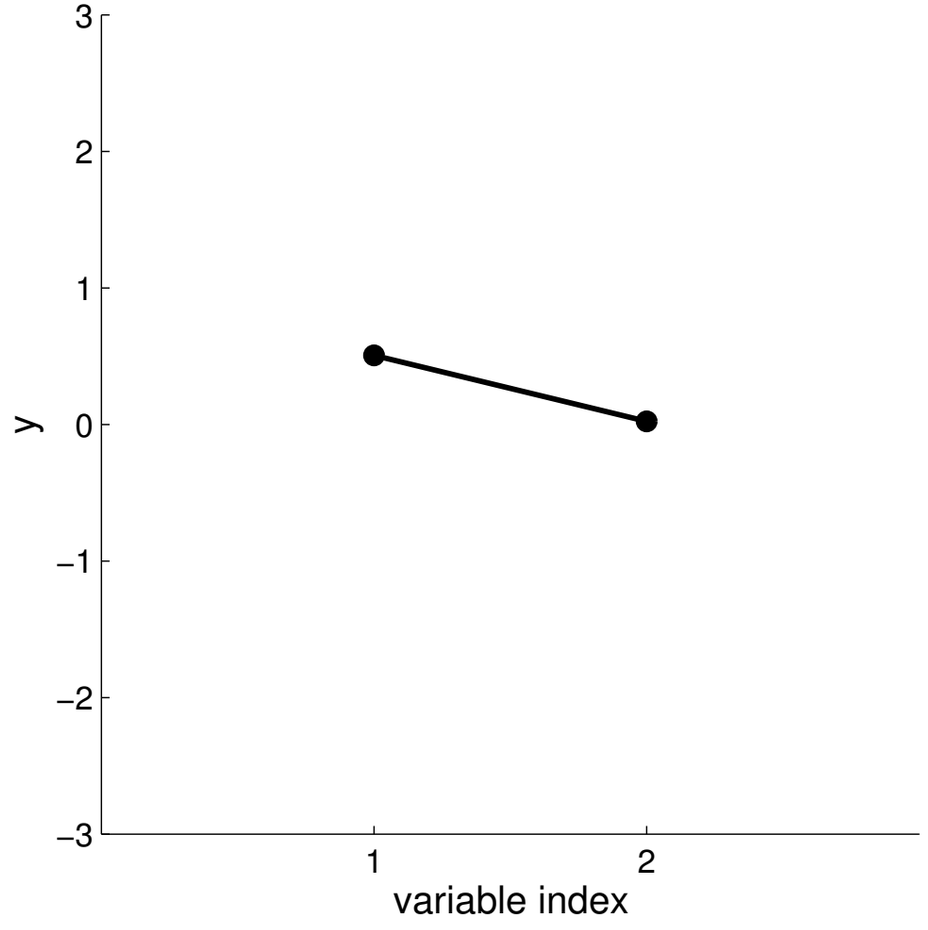
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



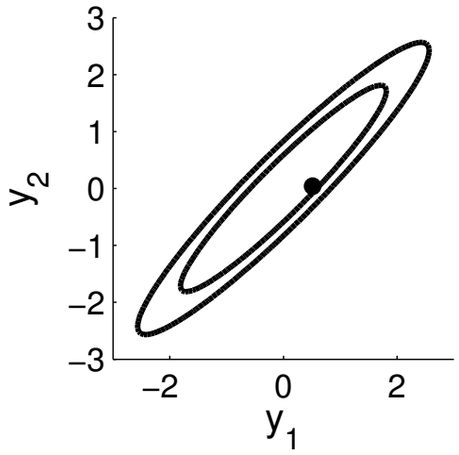
New Visualisation



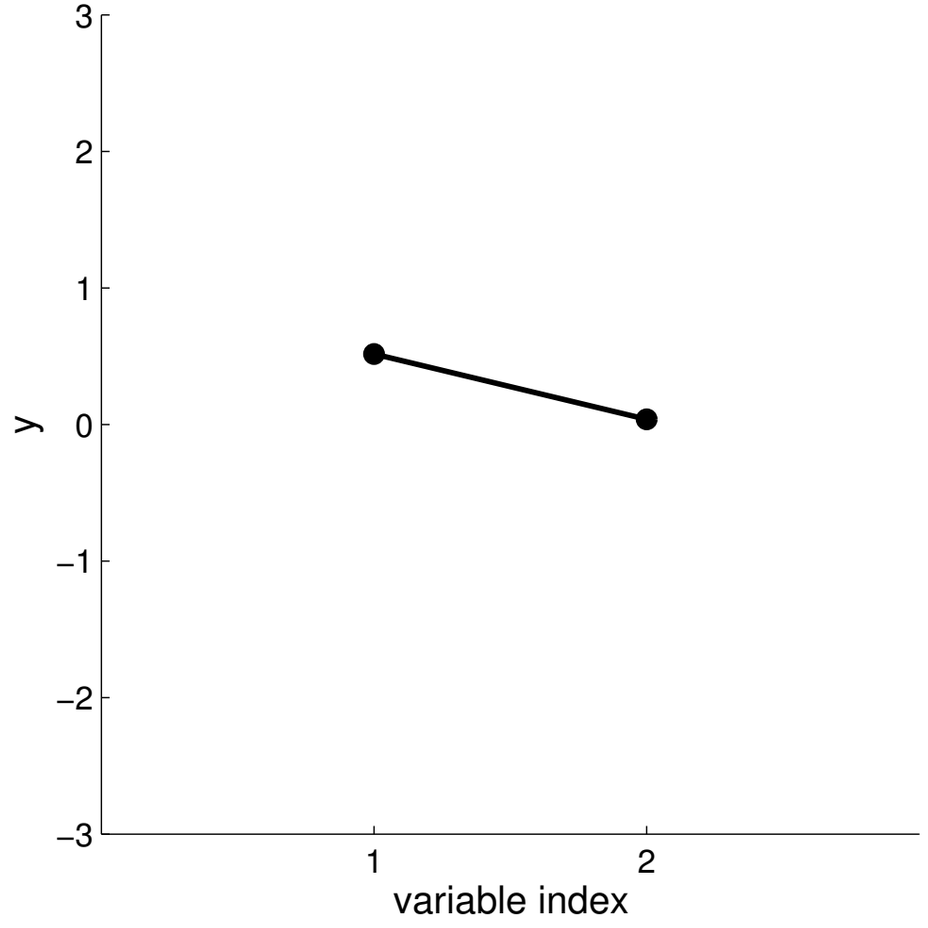
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



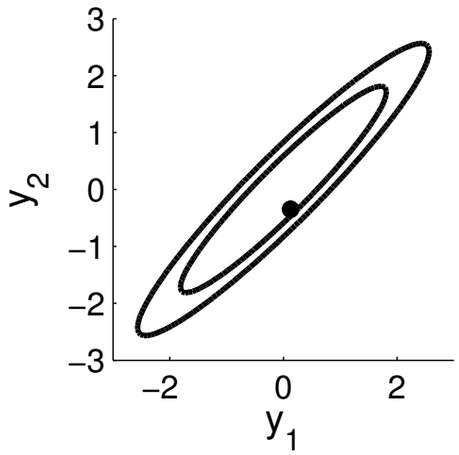
New Visualisation



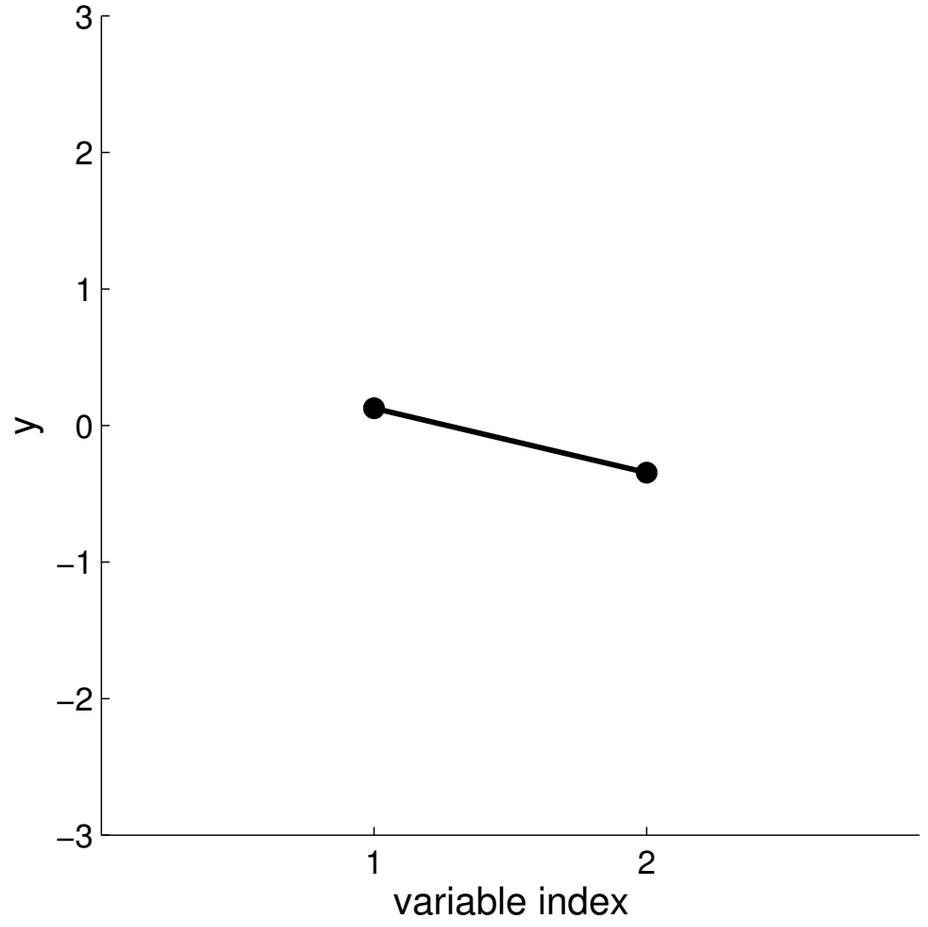
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



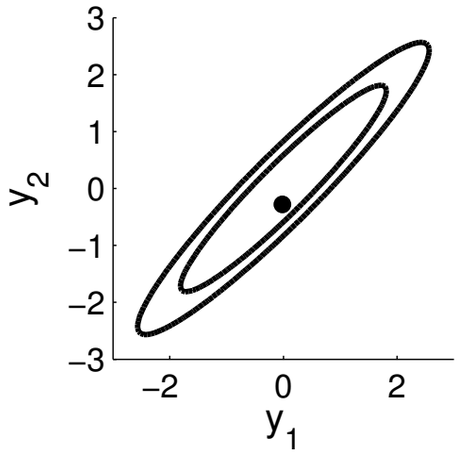
New Visualisation



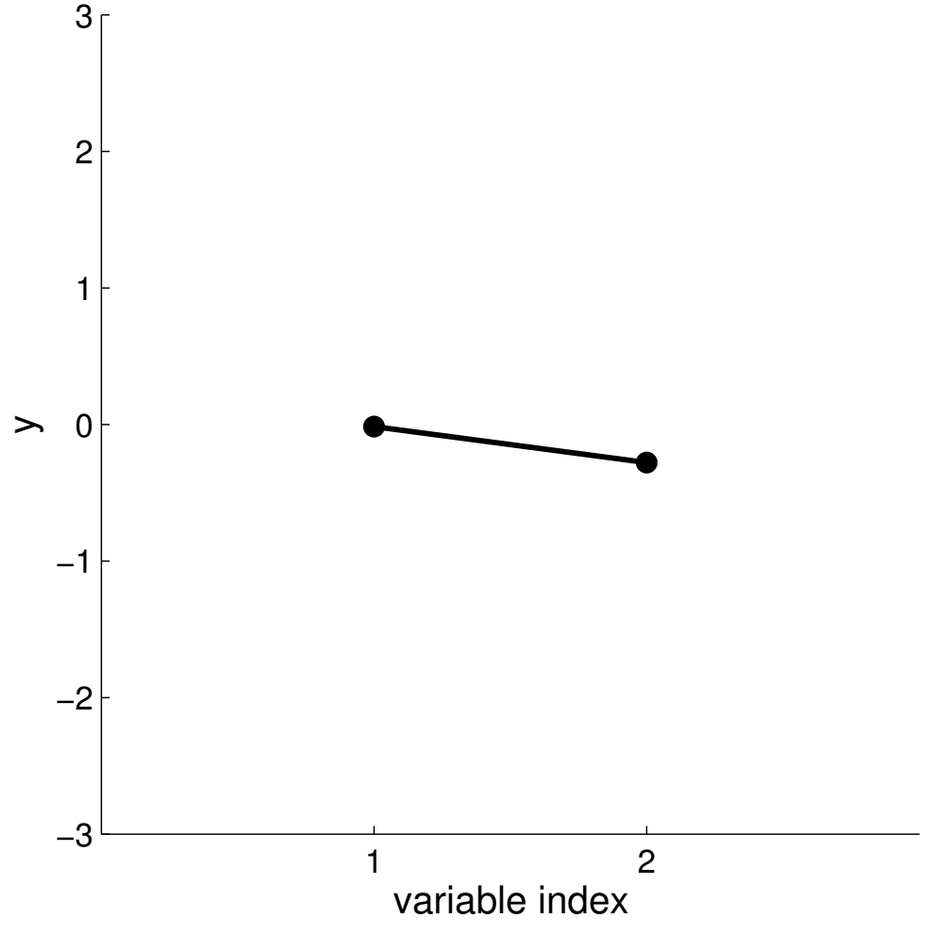
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



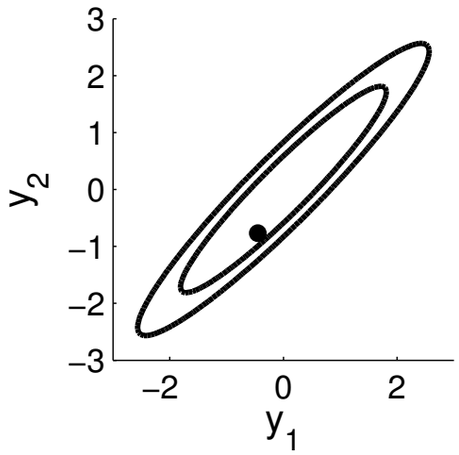
New Visualisation



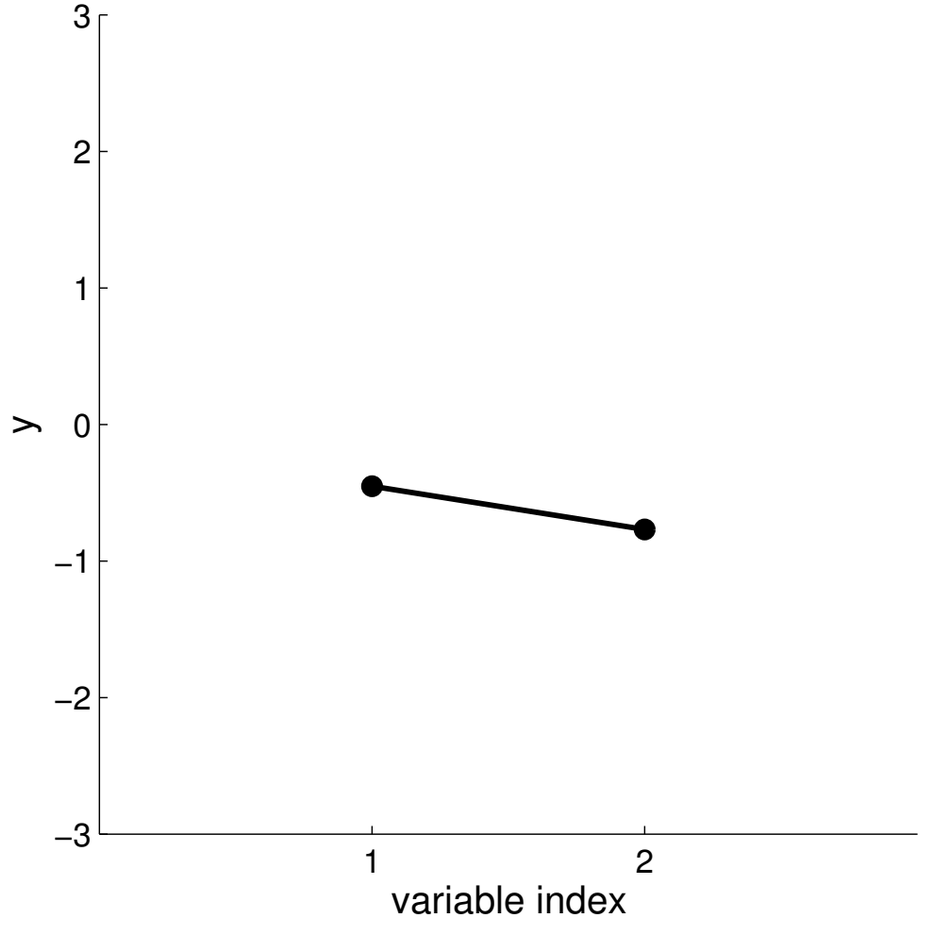
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



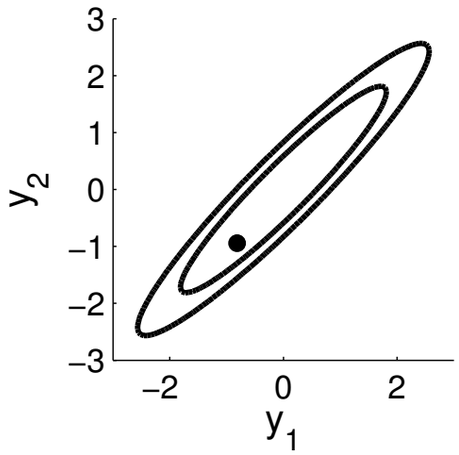
New Visualisation



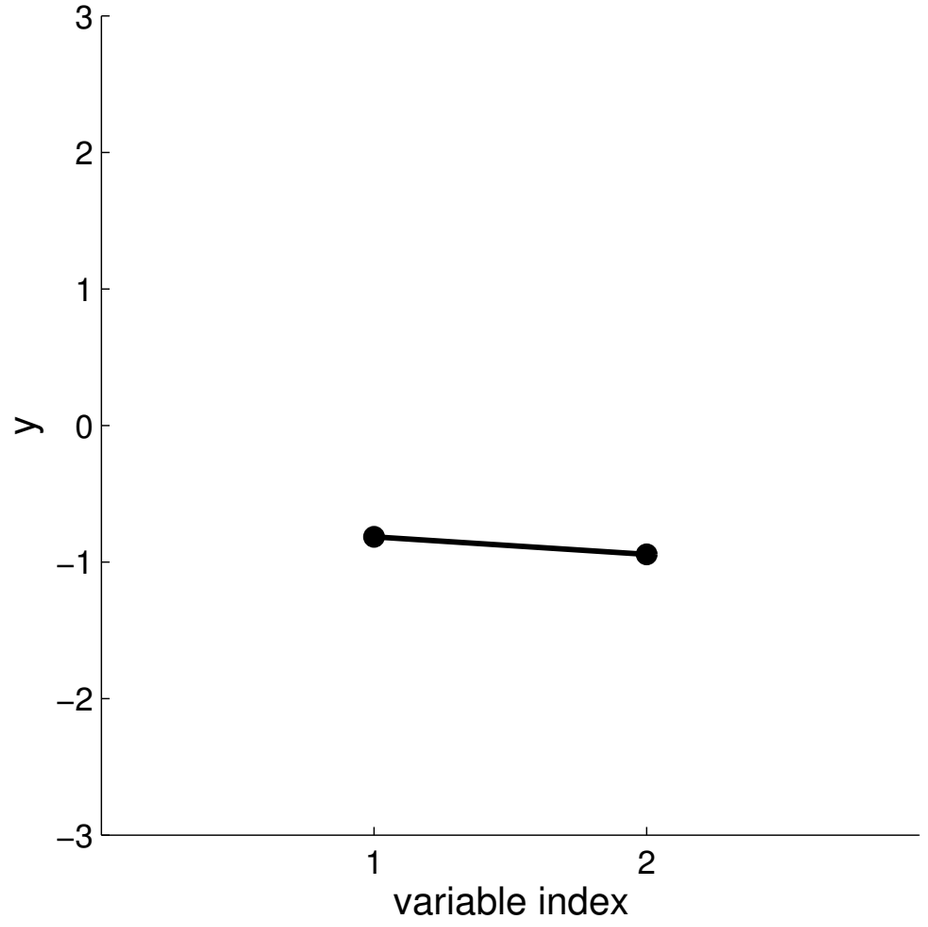
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



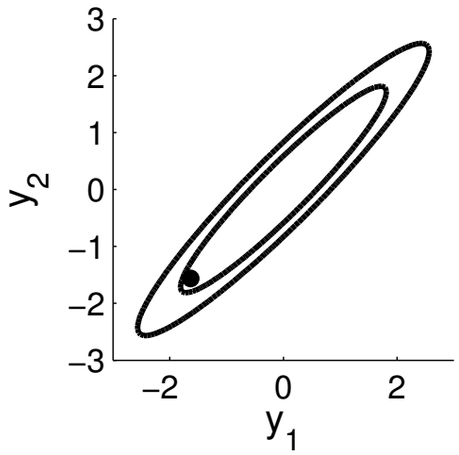
New Visualisation



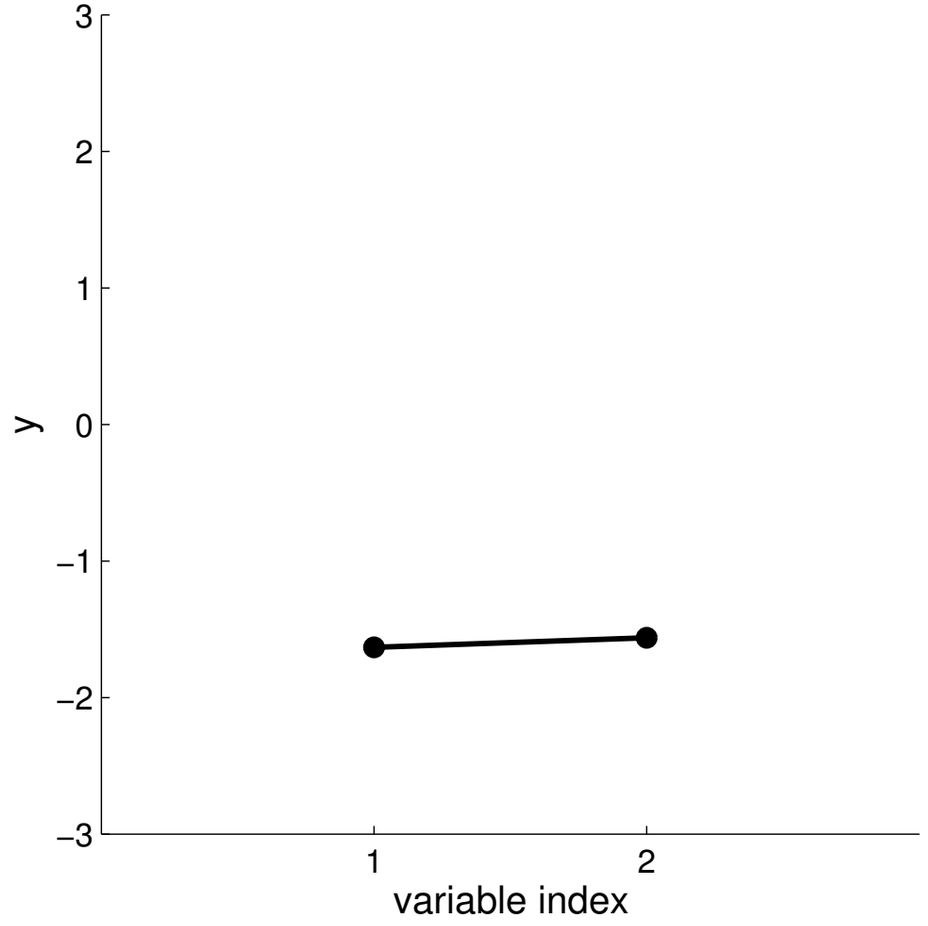
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



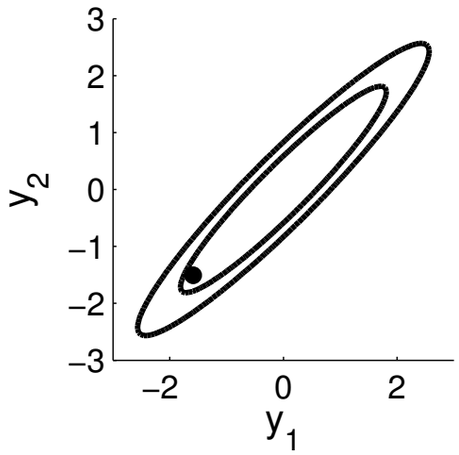
New Visualisation



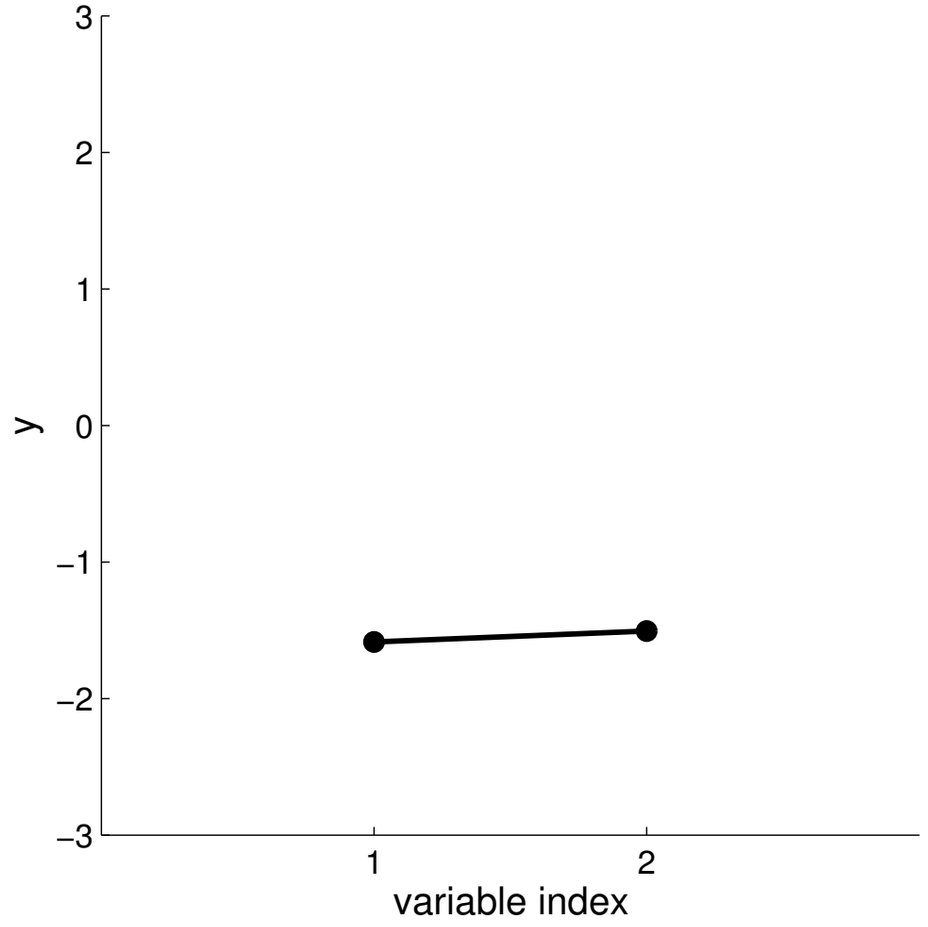
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



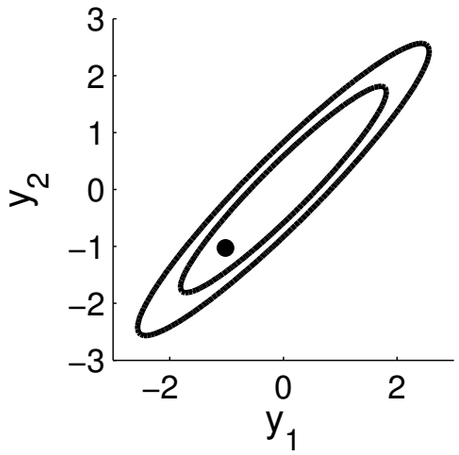
New Visualisation



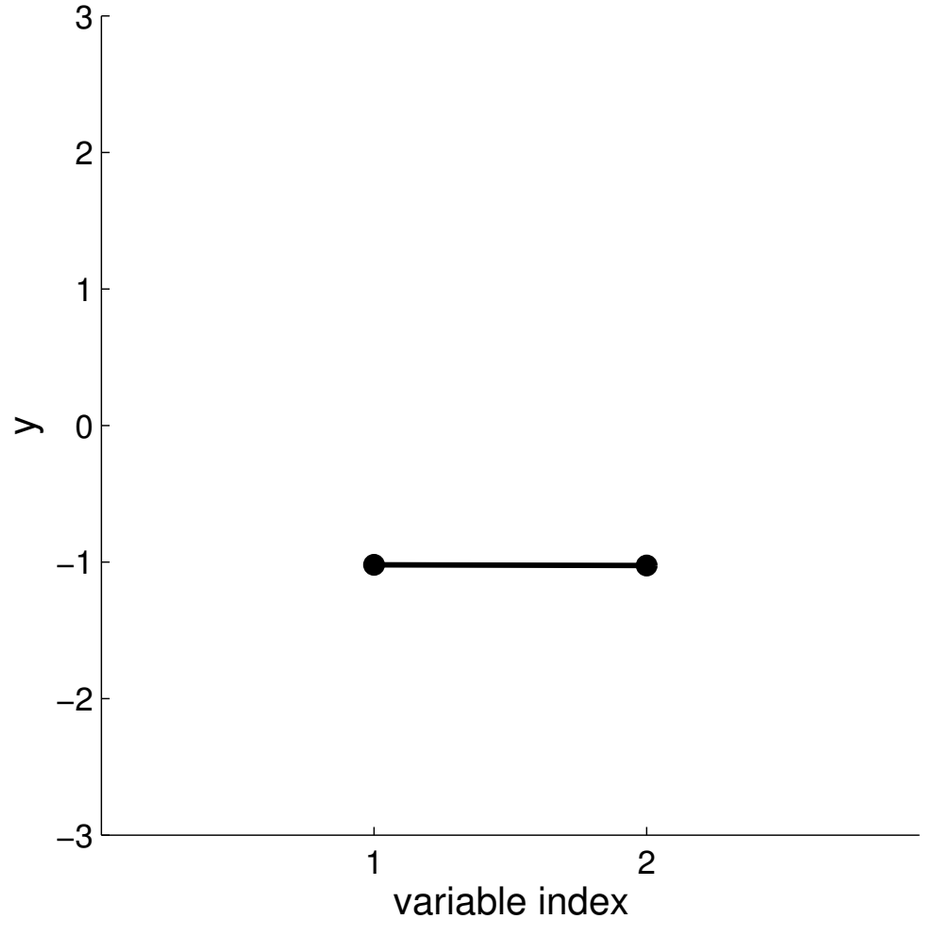
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



New Visualisation

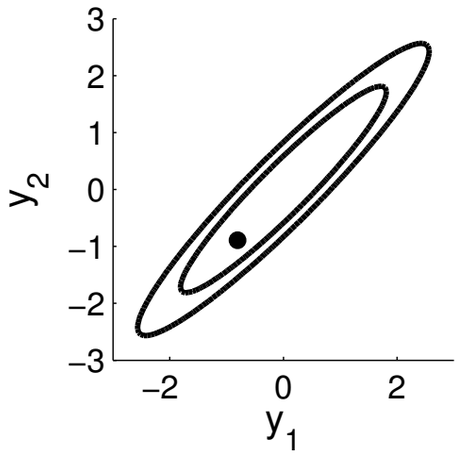


$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$

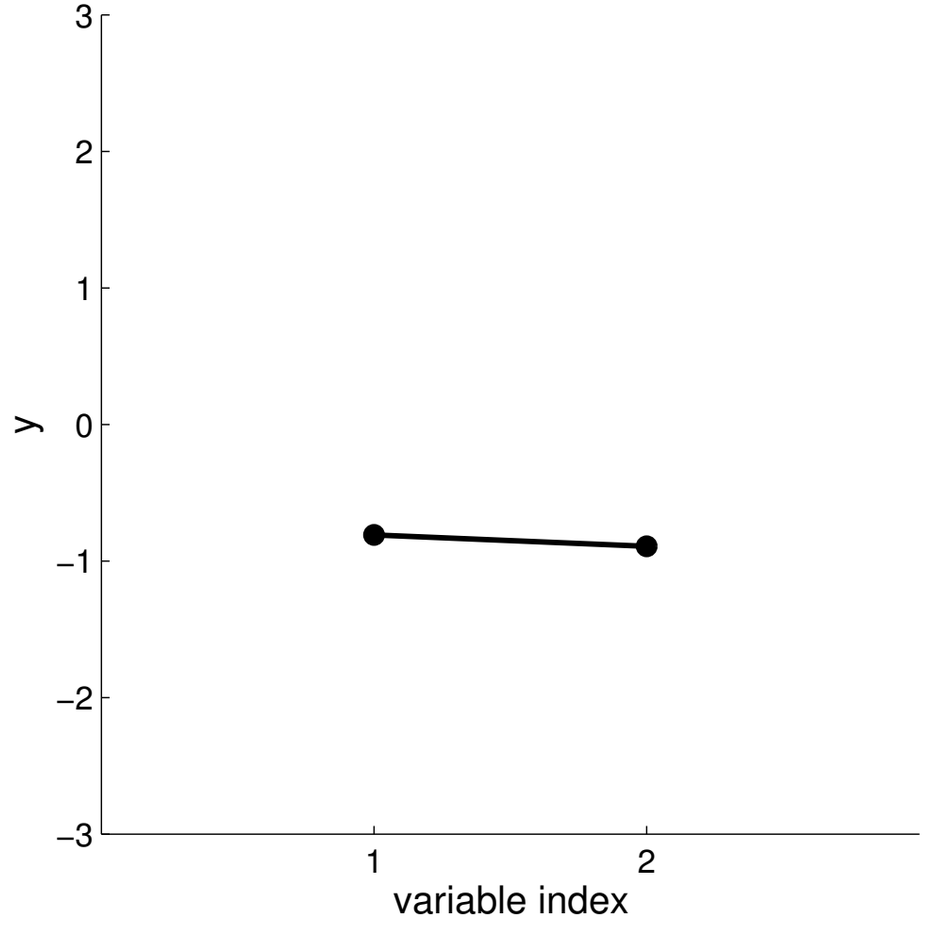


Q: How would the bar look like if the covariance of y_1,y_2 was the identity matrix?

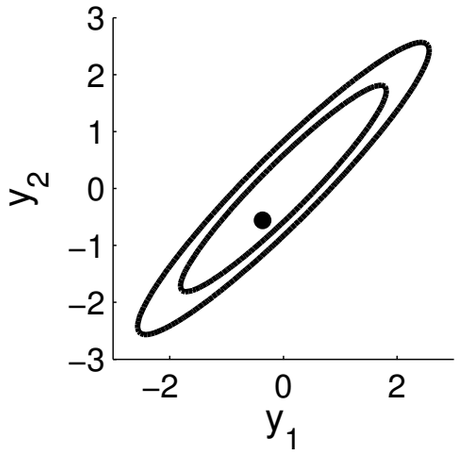
New Visualisation



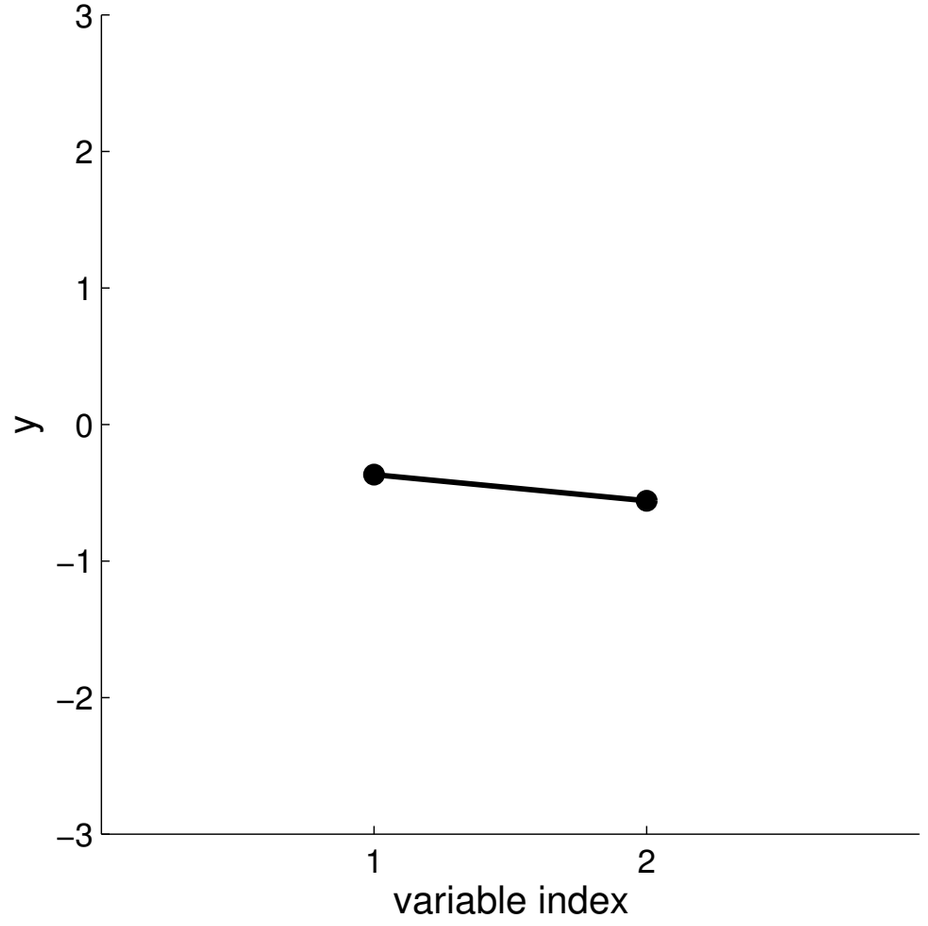
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



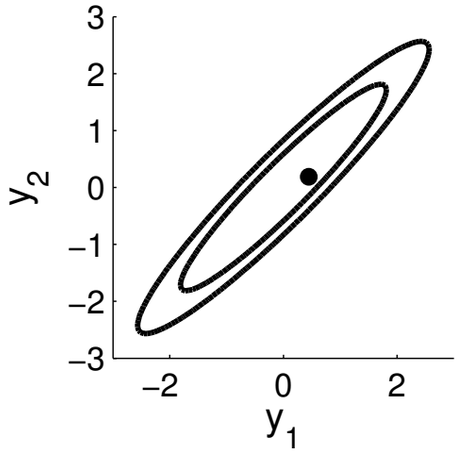
New Visualisation



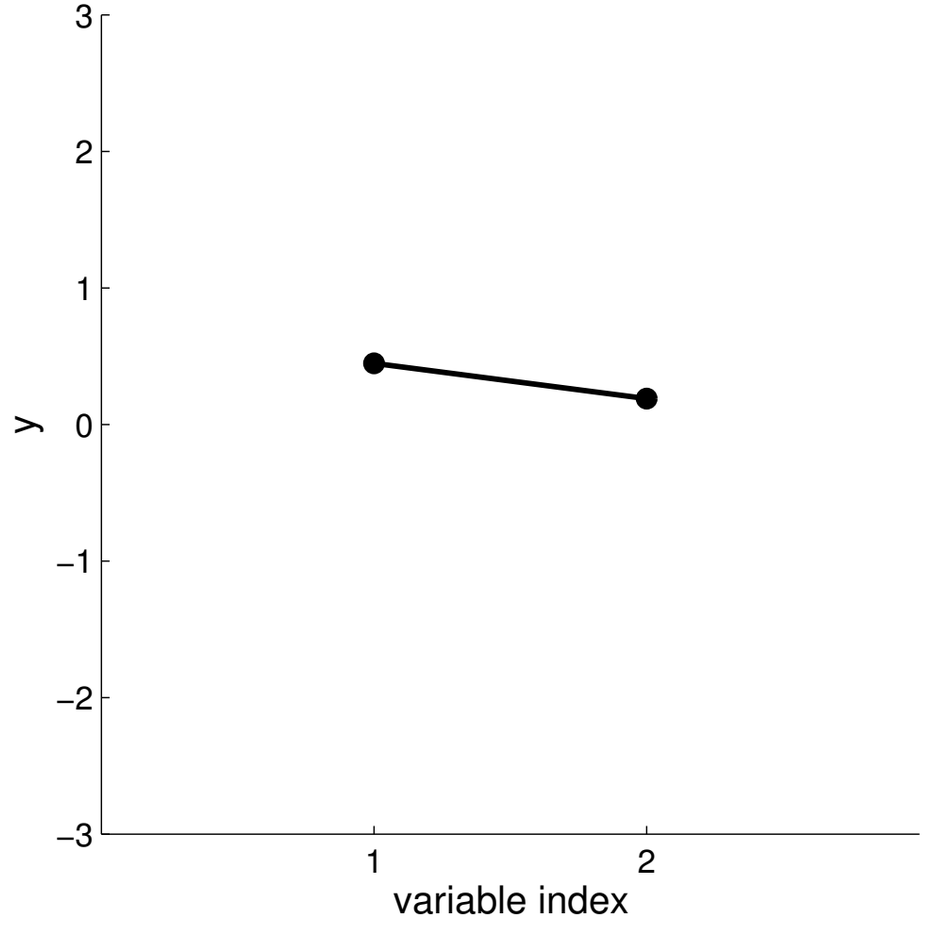
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



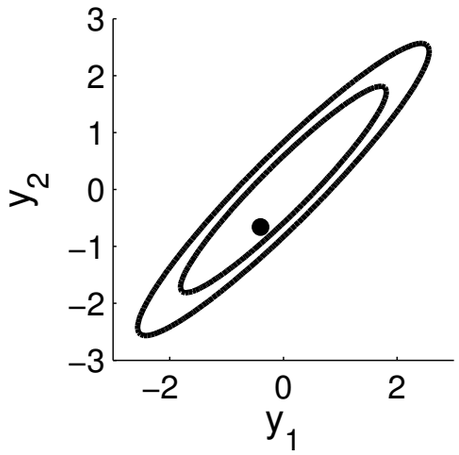
New Visualisation



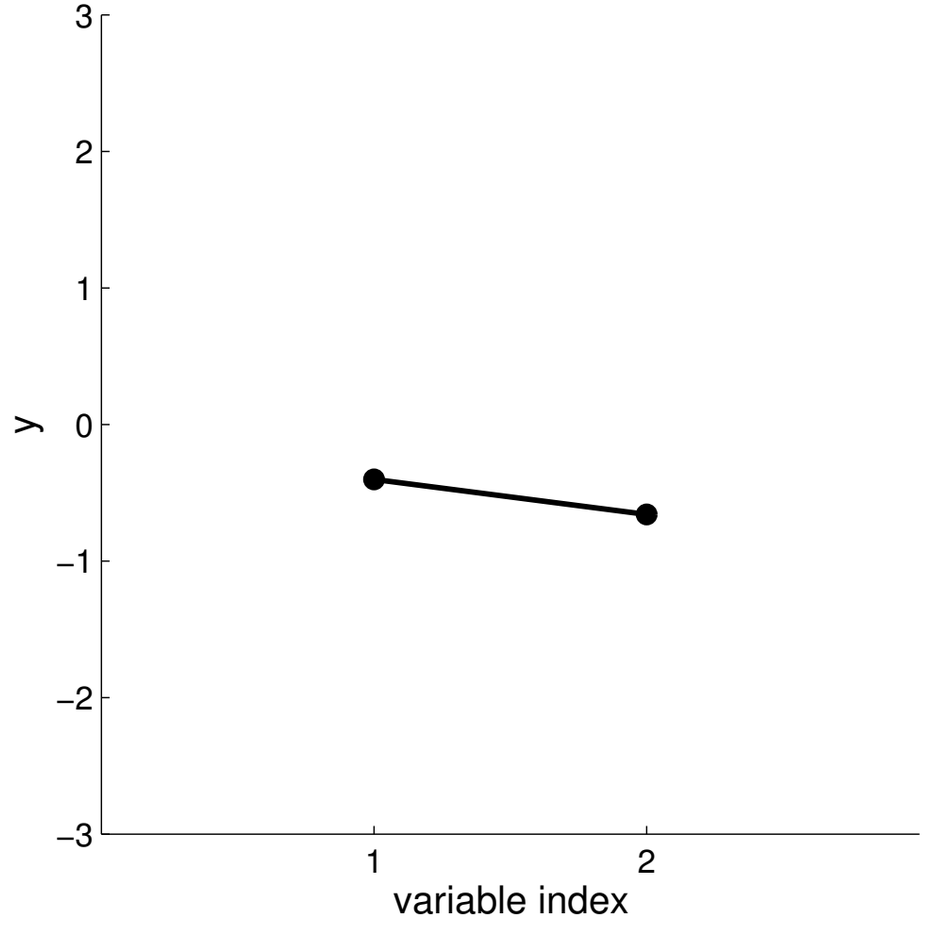
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



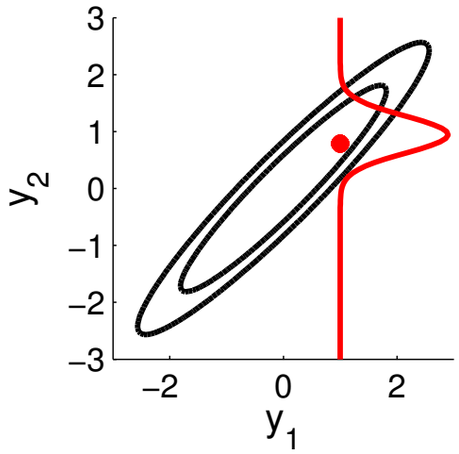
New Visualisation



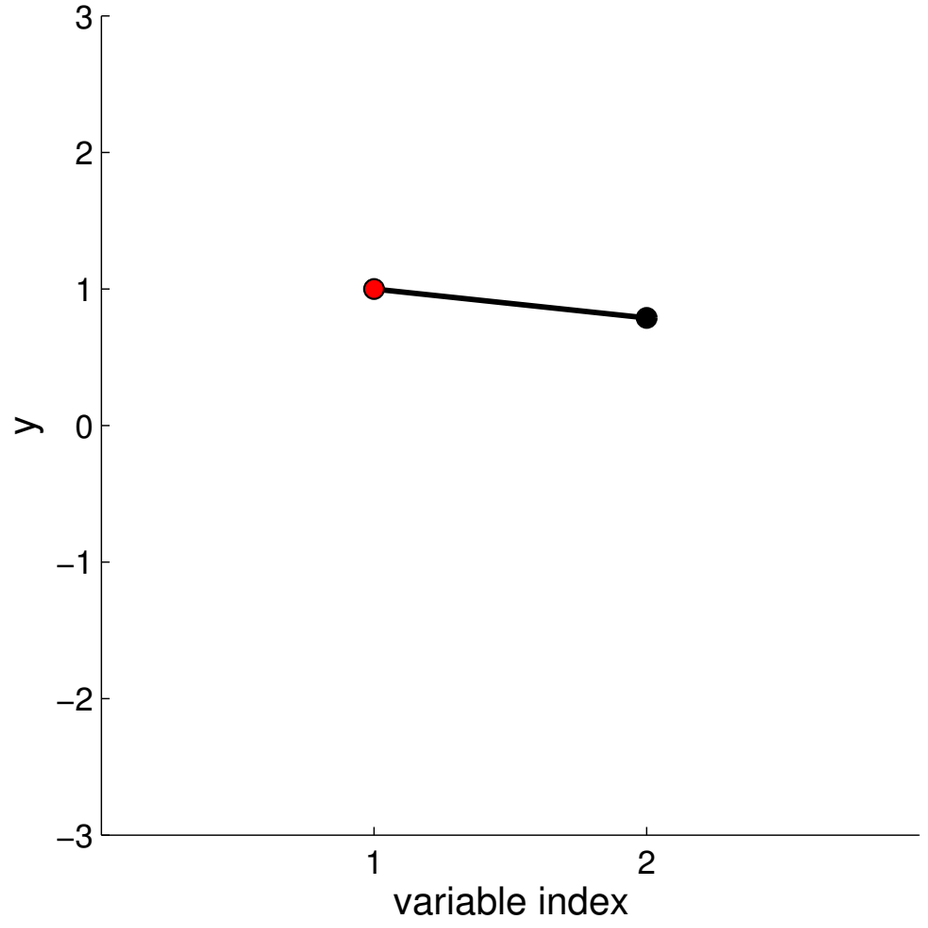
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



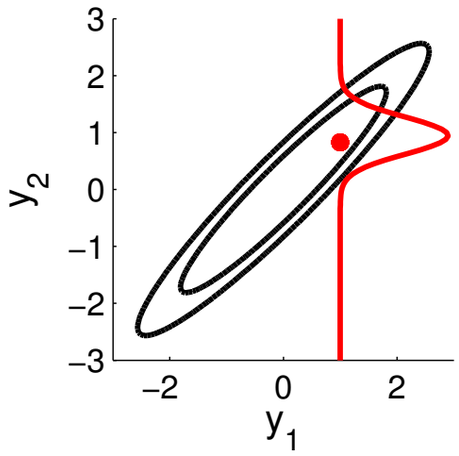
New Visualisation



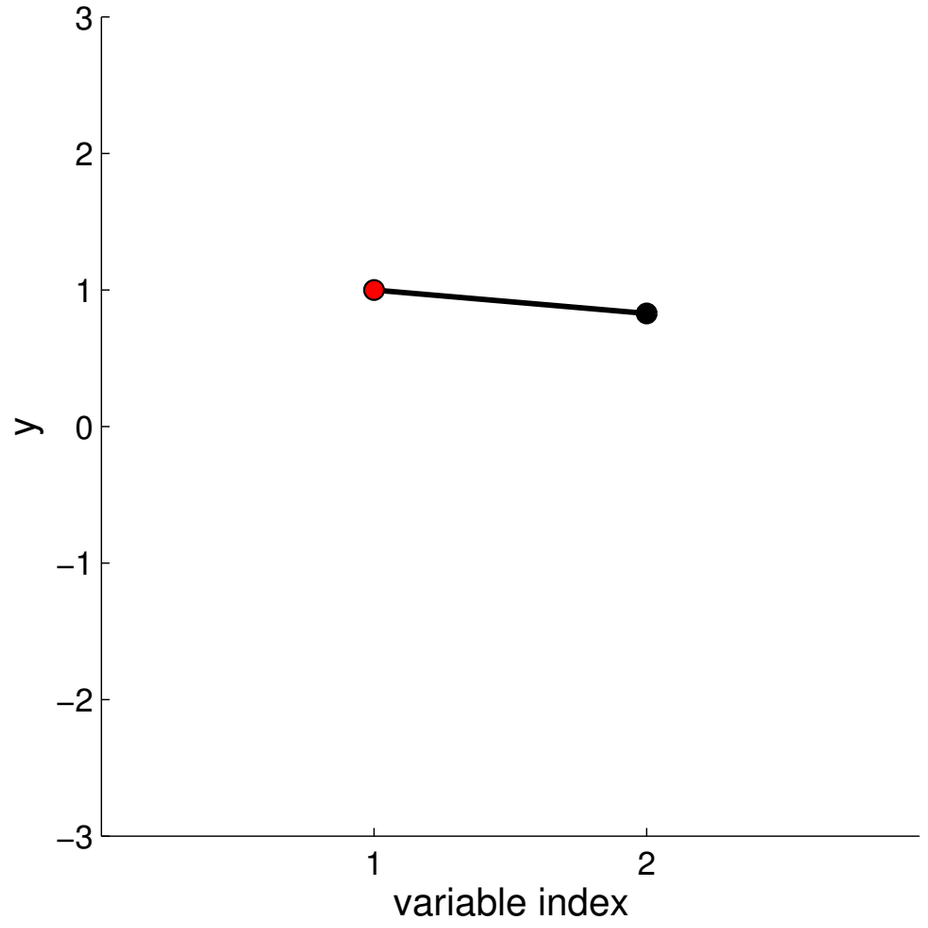
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



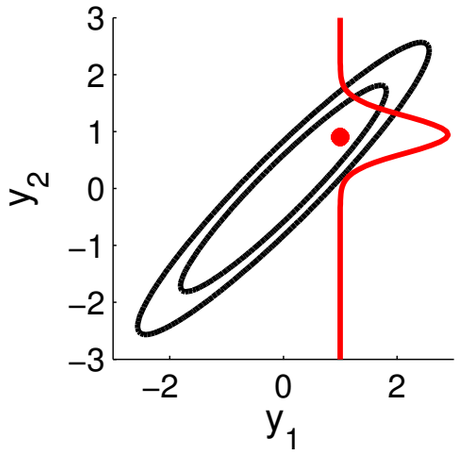
New Visualisation



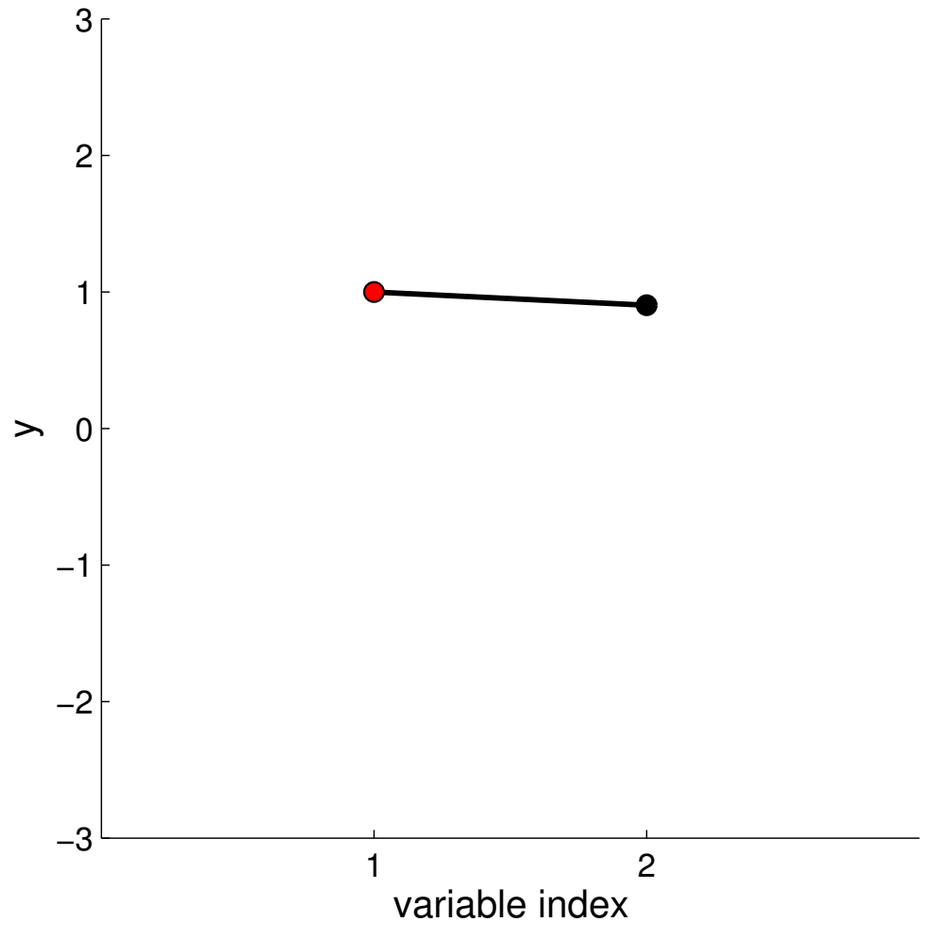
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



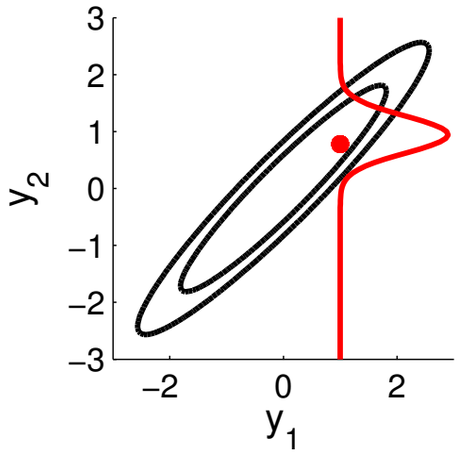
New Visualisation



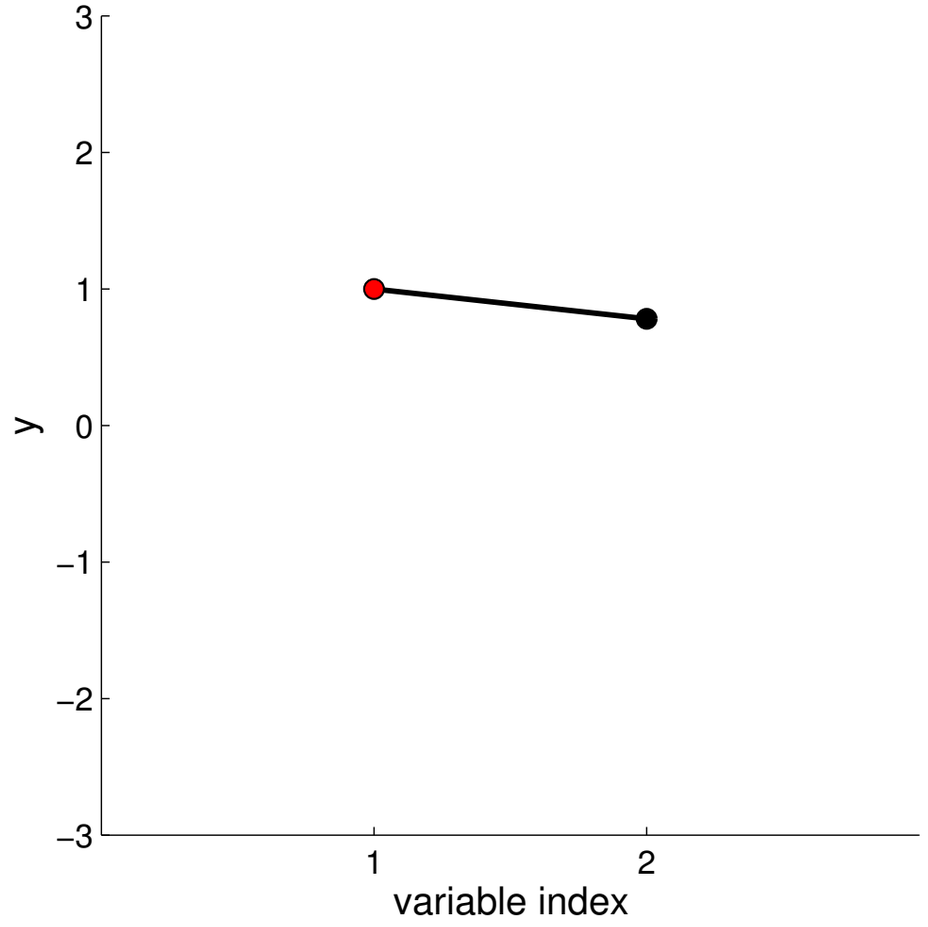
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



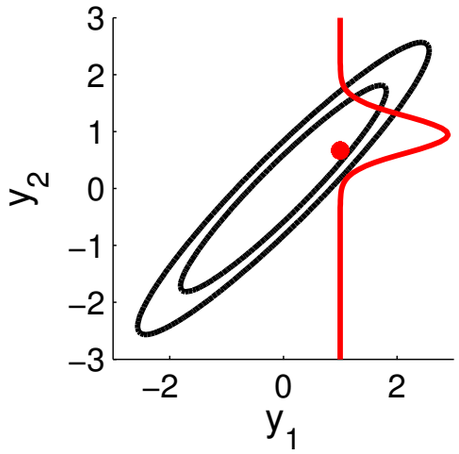
New Visualisation



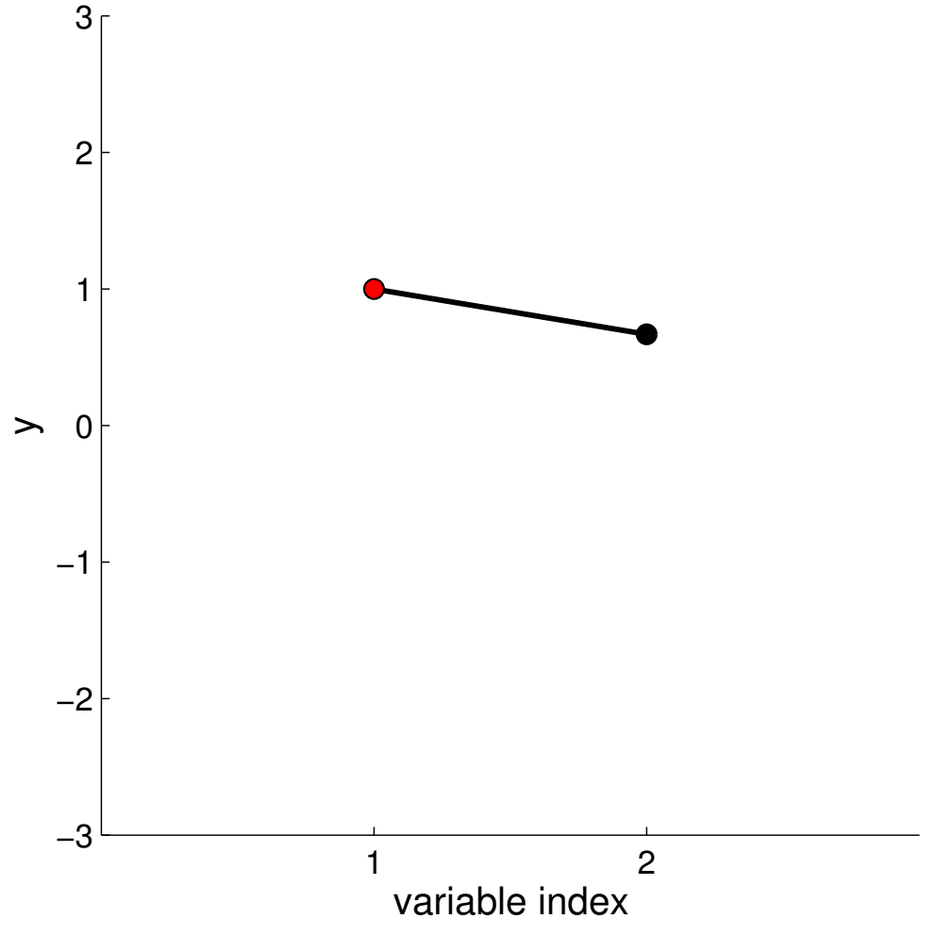
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



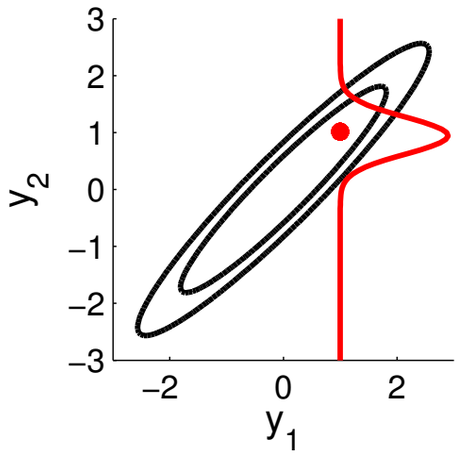
New Visualisation



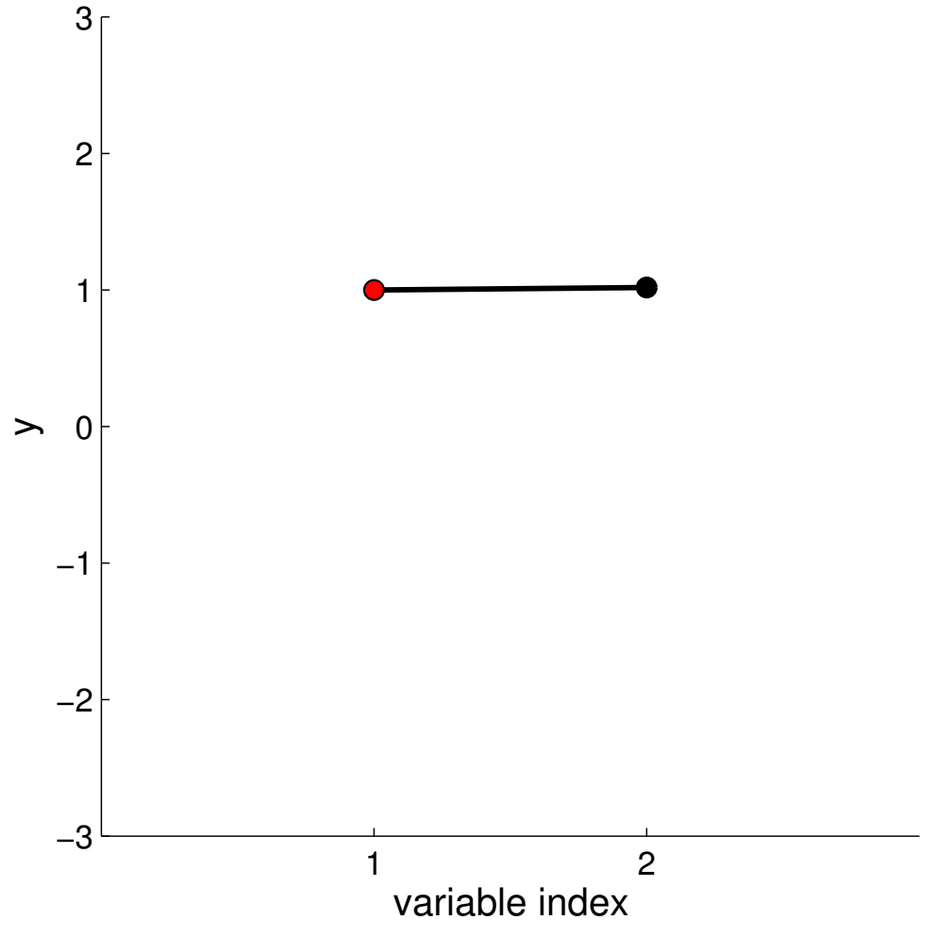
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



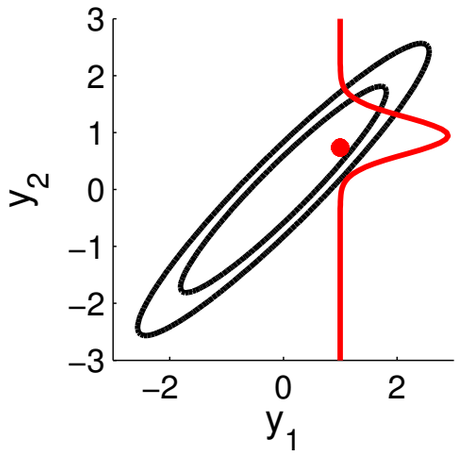
New Visualisation



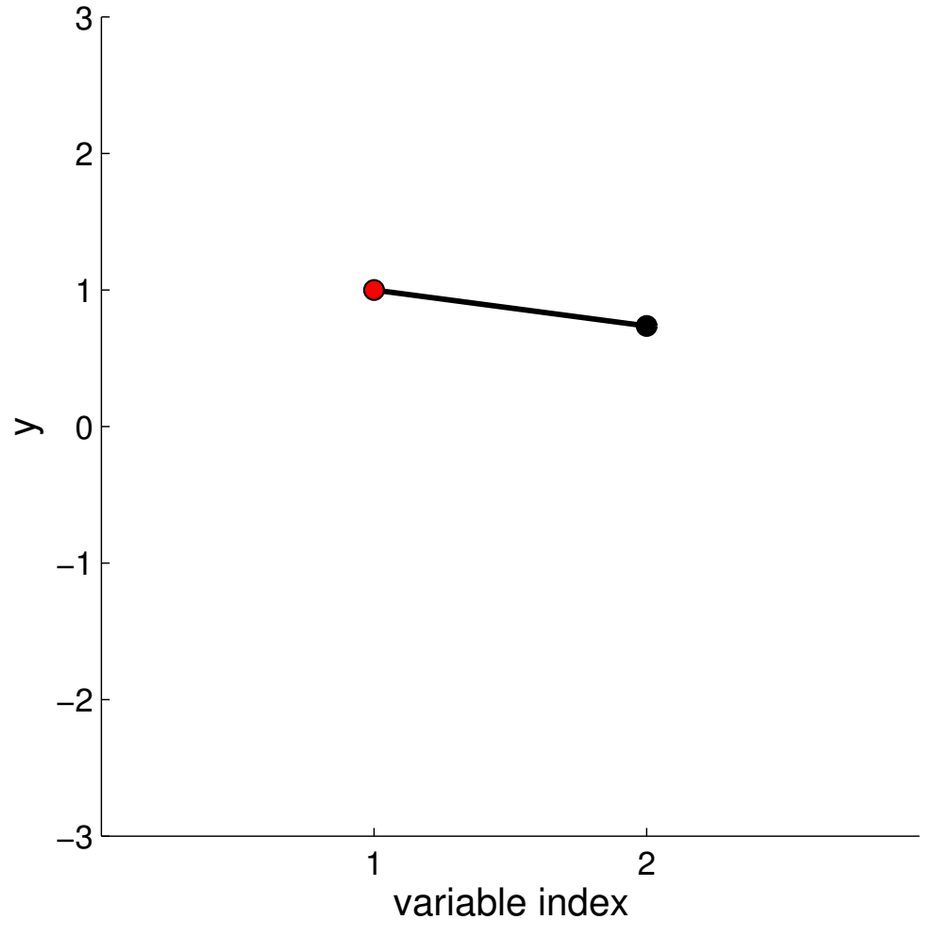
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



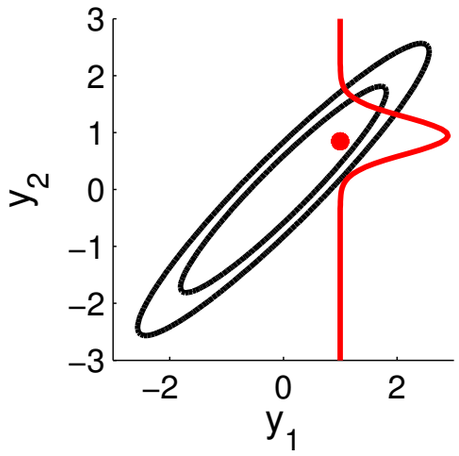
New Visualisation



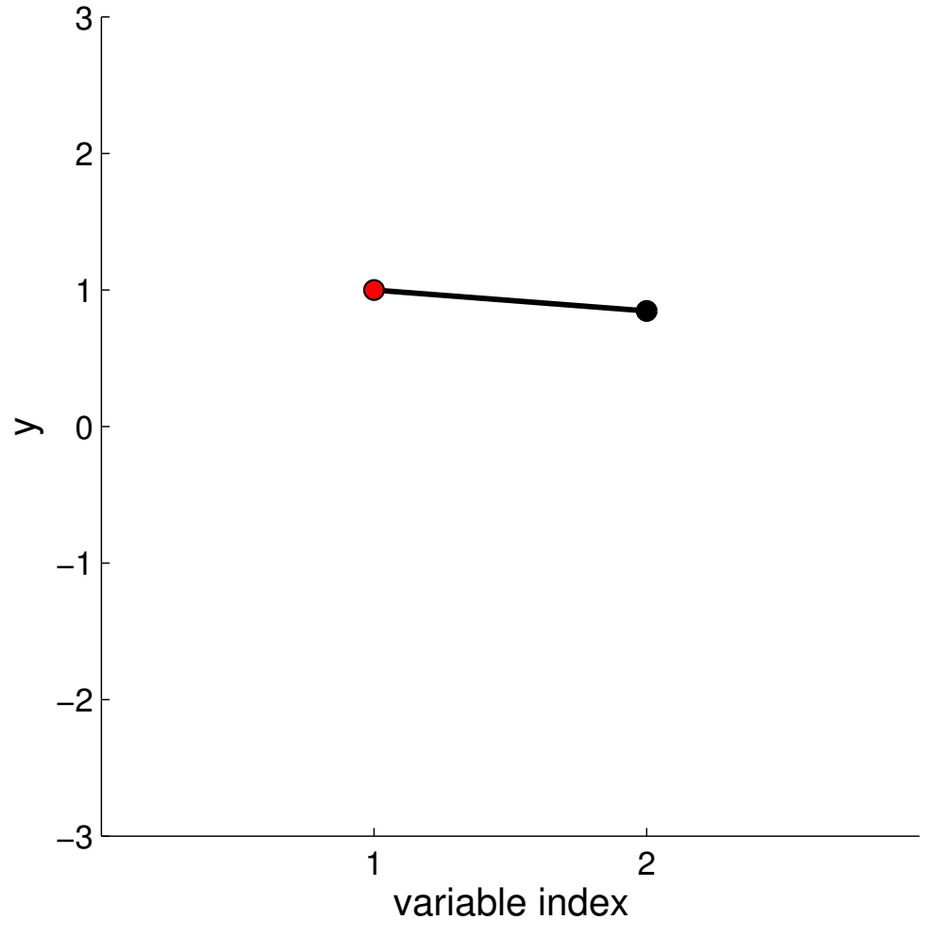
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



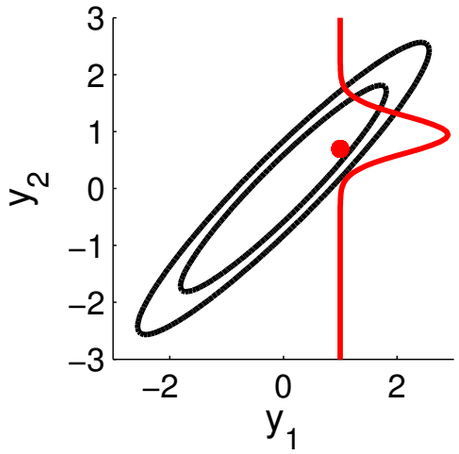
New Visualisation



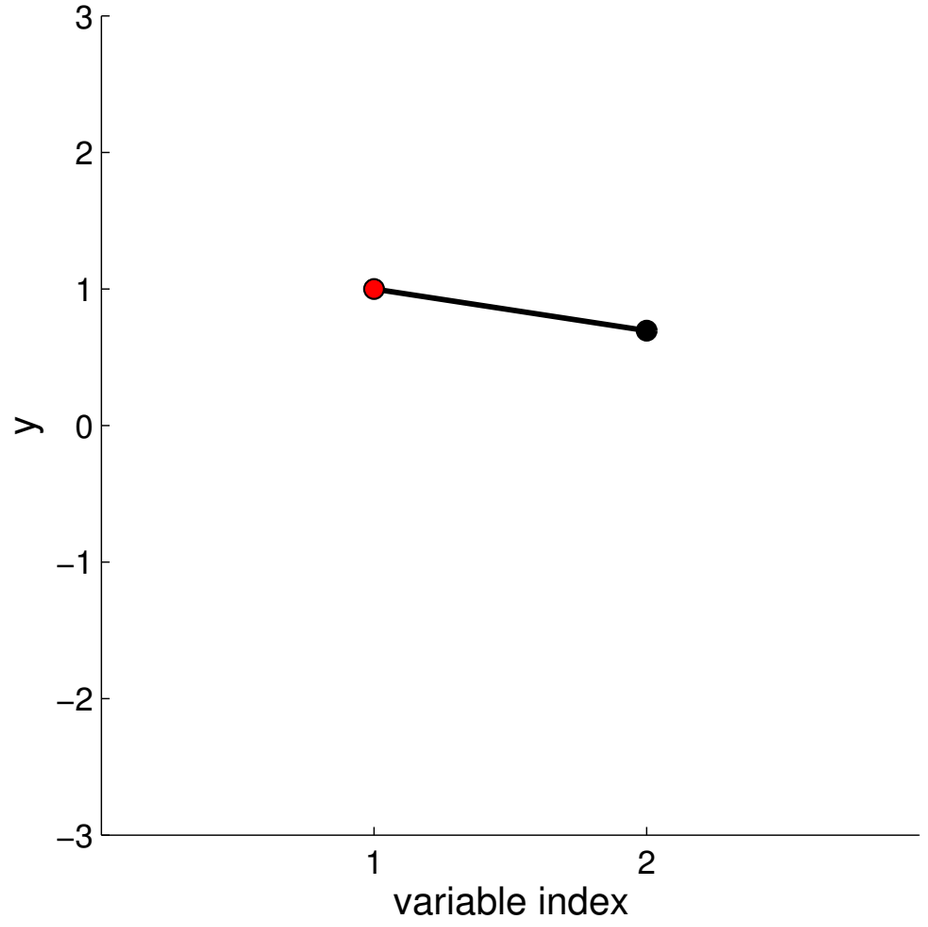
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



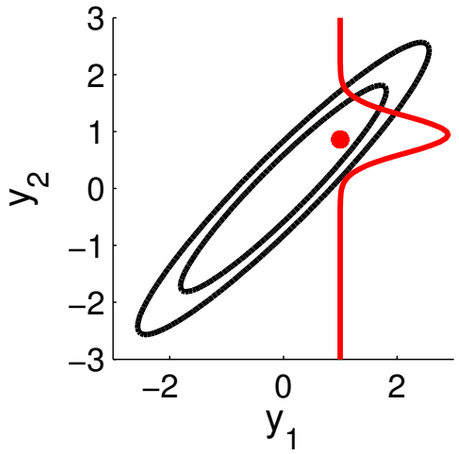
New Visualisation



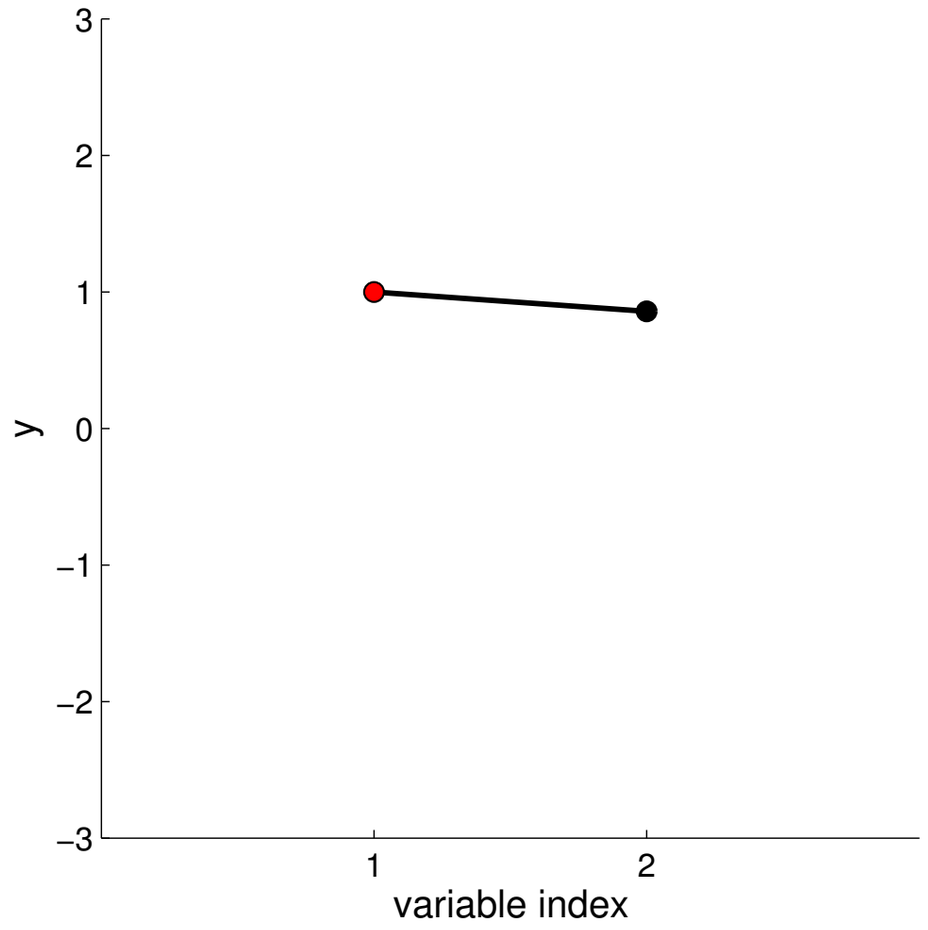
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



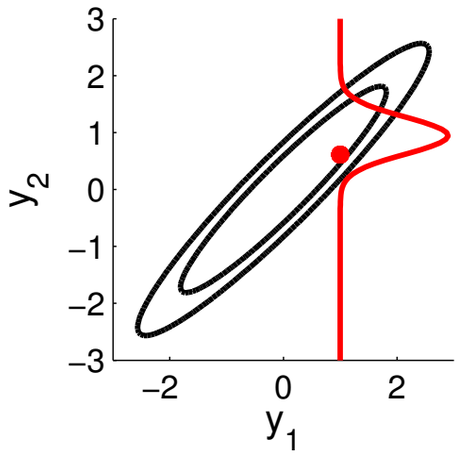
New Visualisation



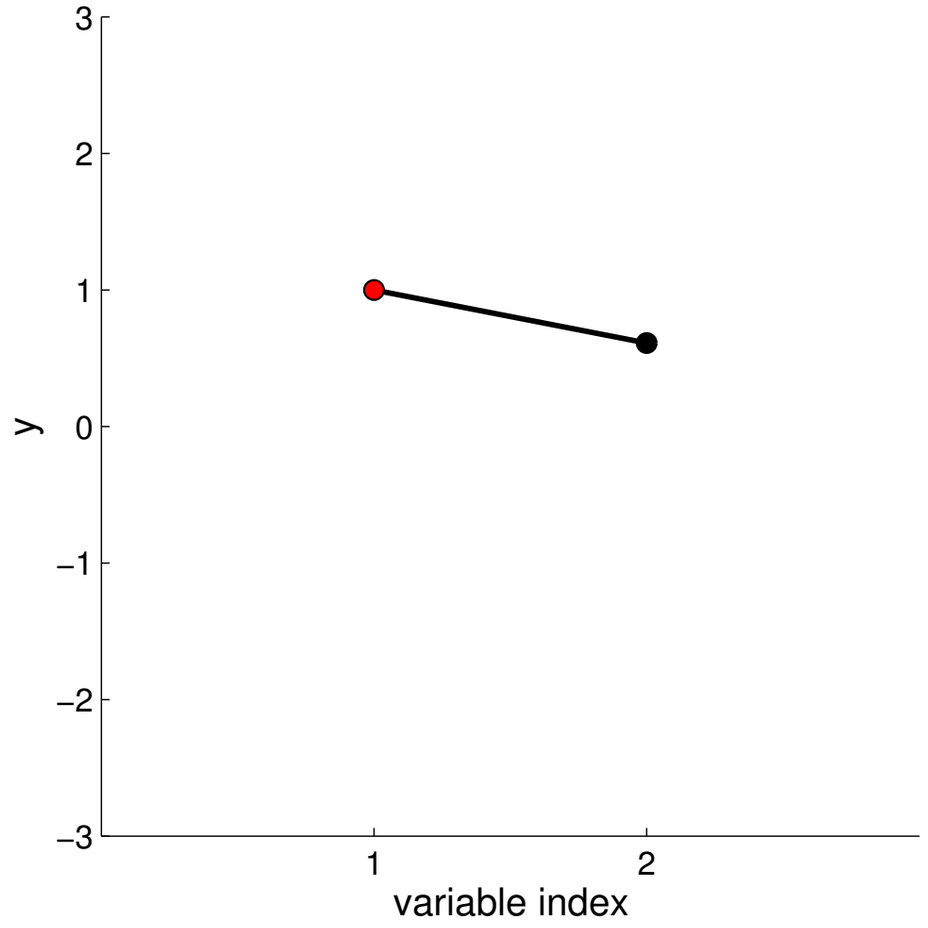
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



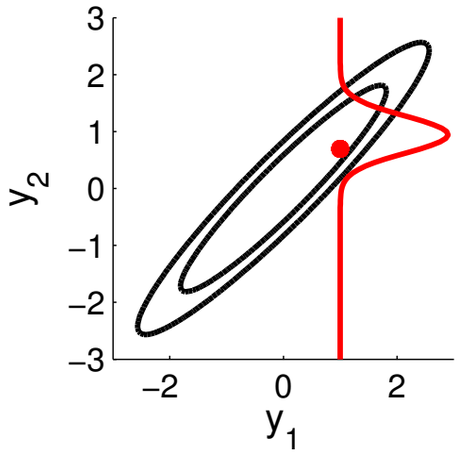
New Visualisation



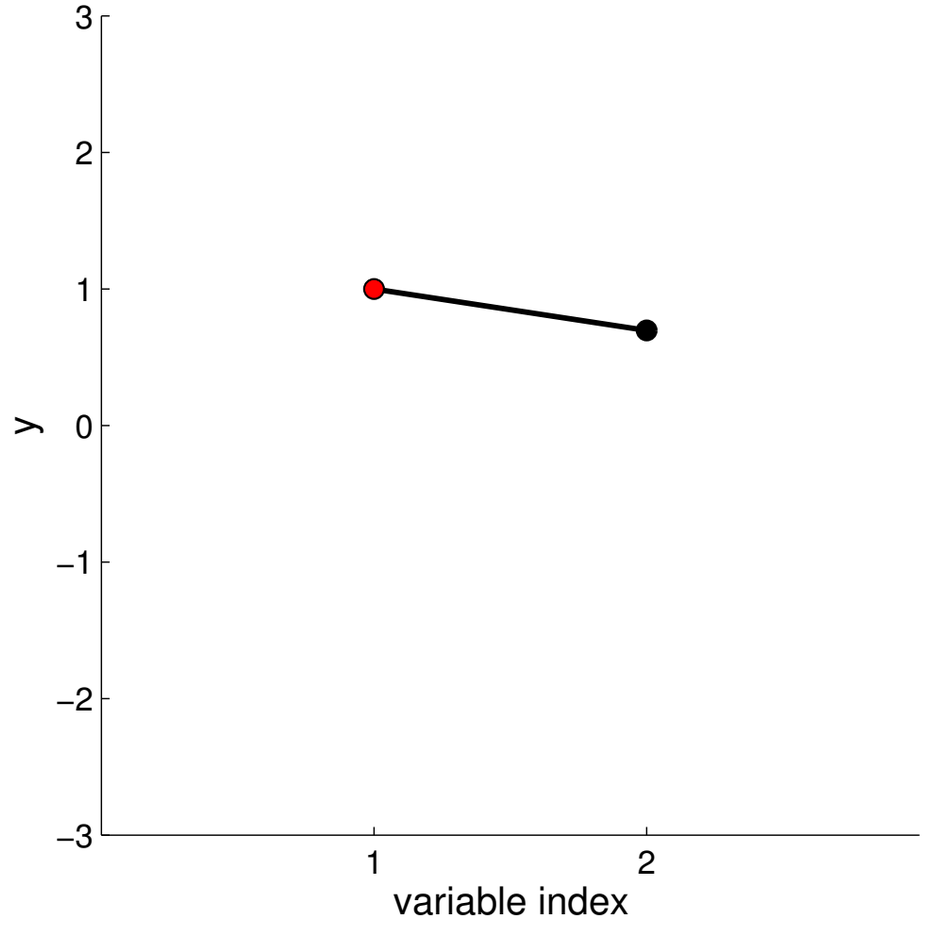
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



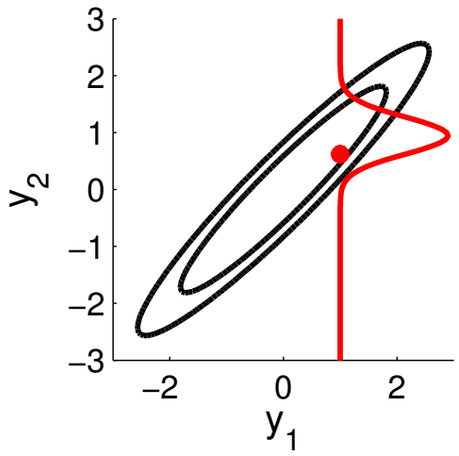
New Visualisation



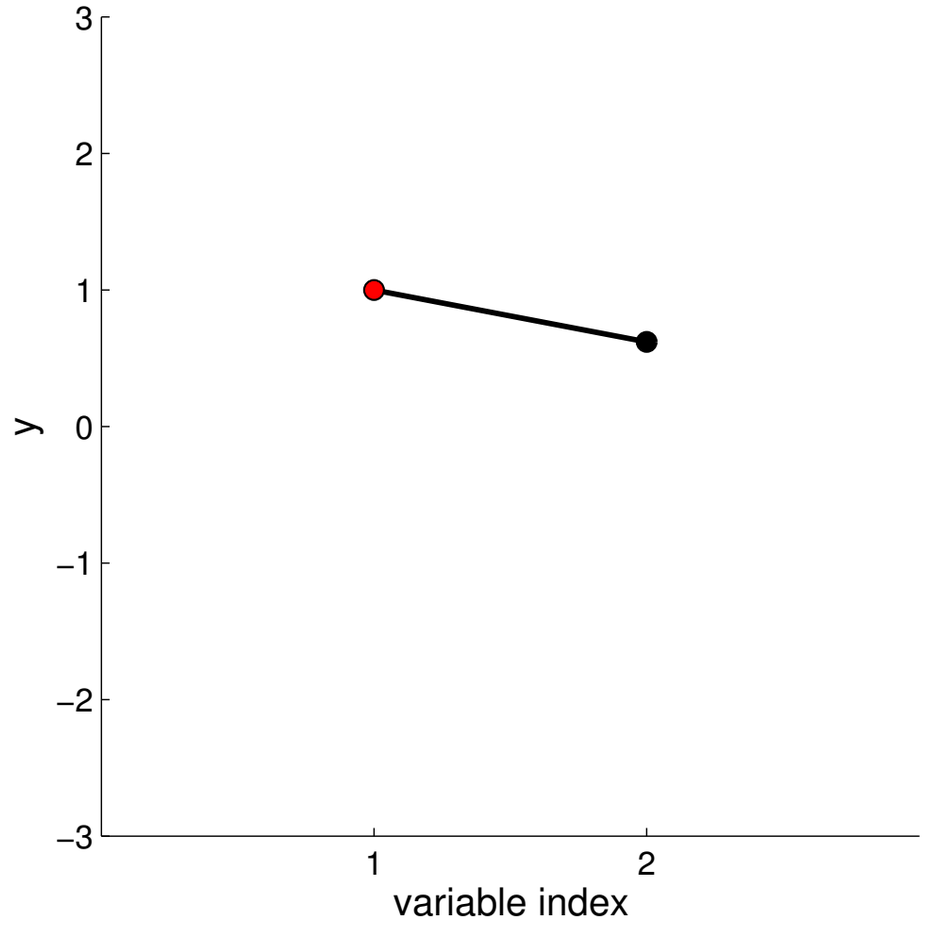
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



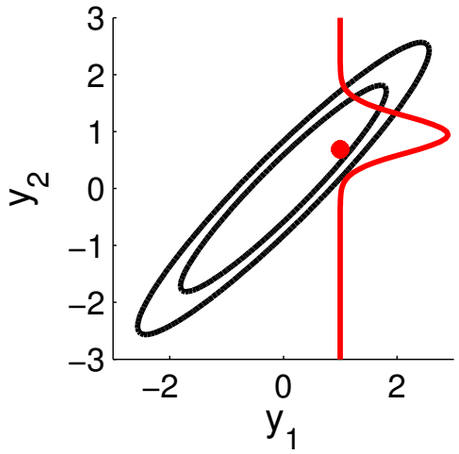
New Visualisation



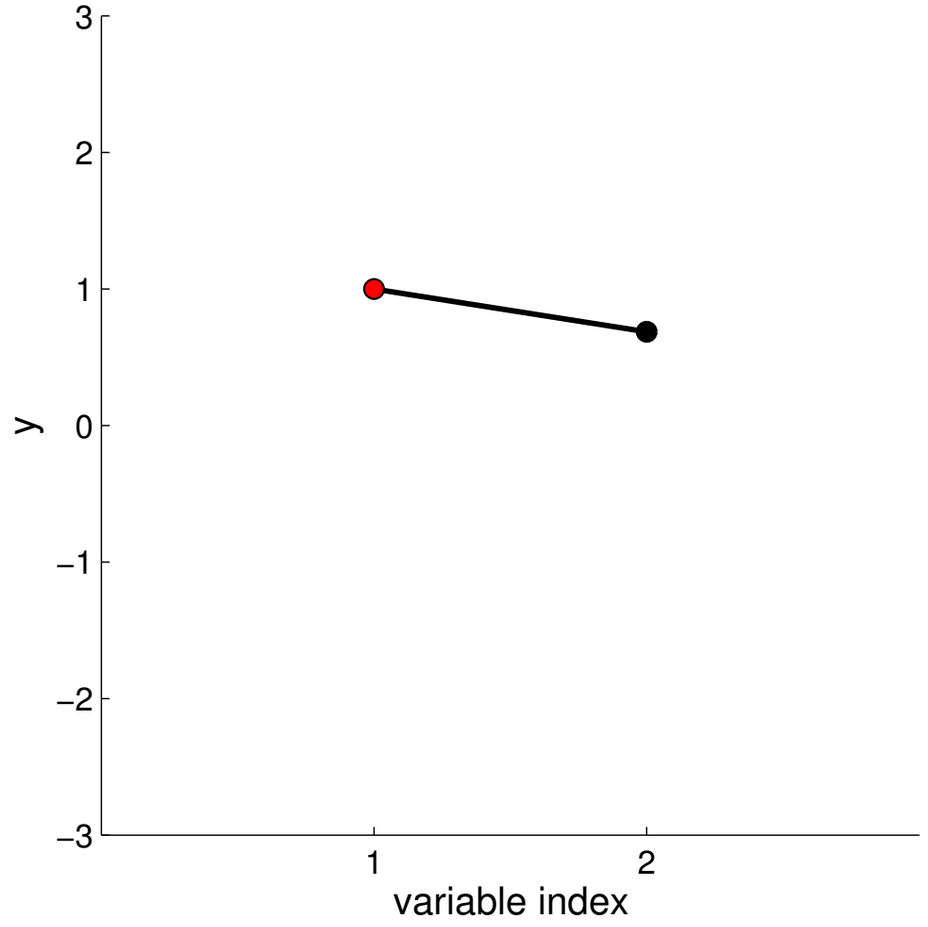
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



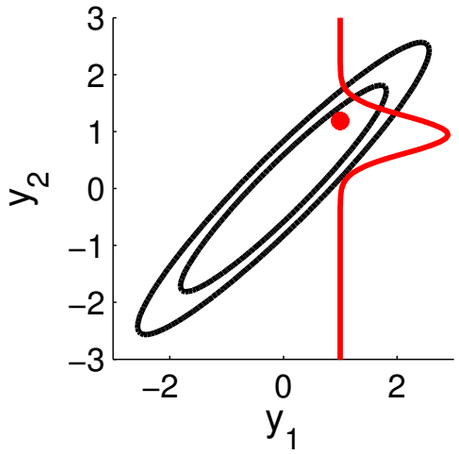
New Visualisation



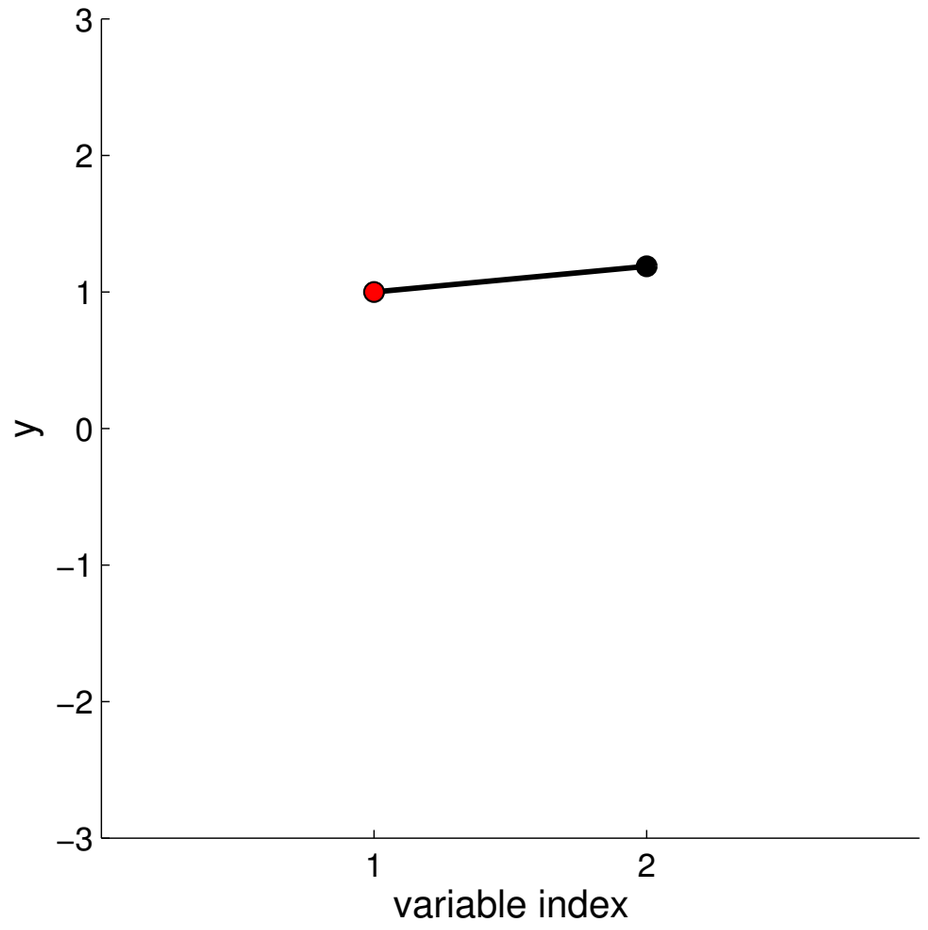
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



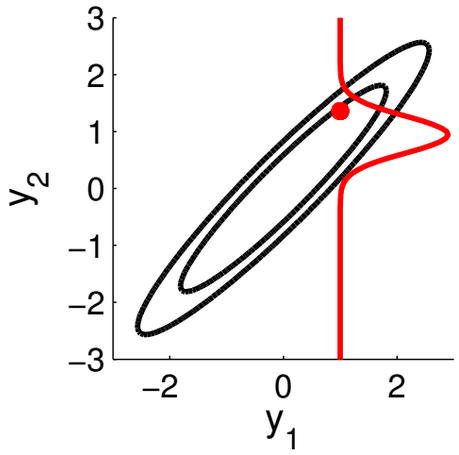
New Visualisation



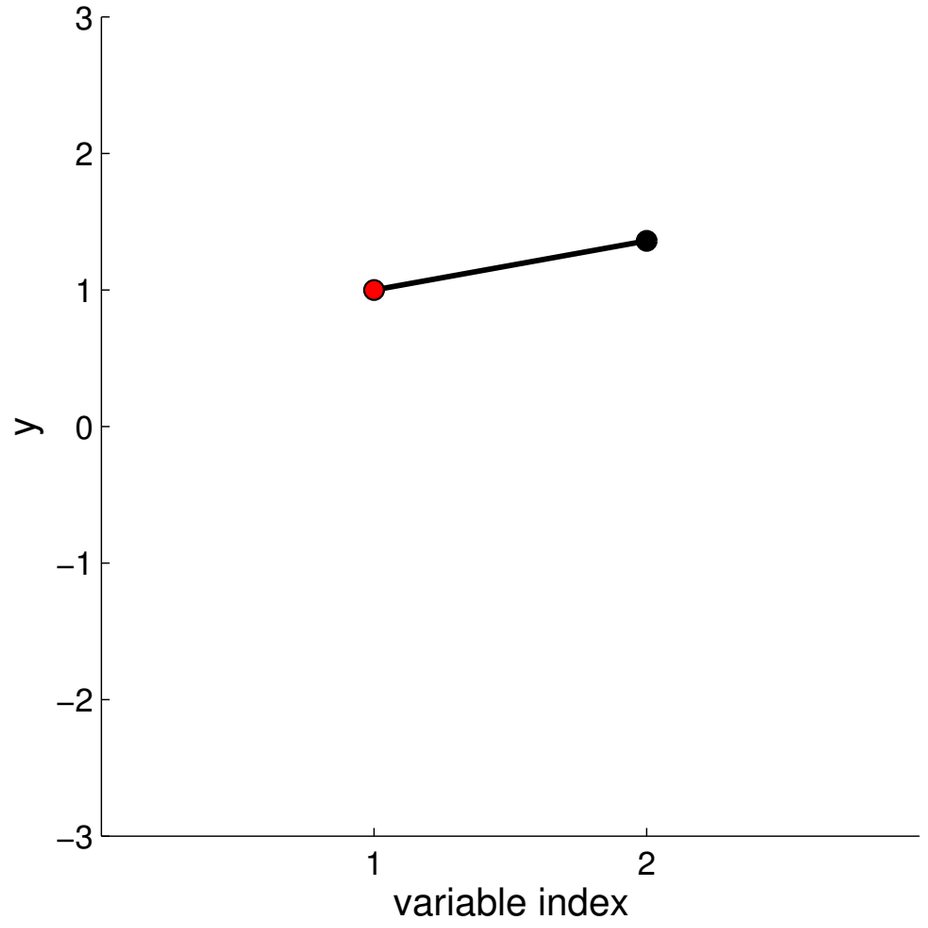
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



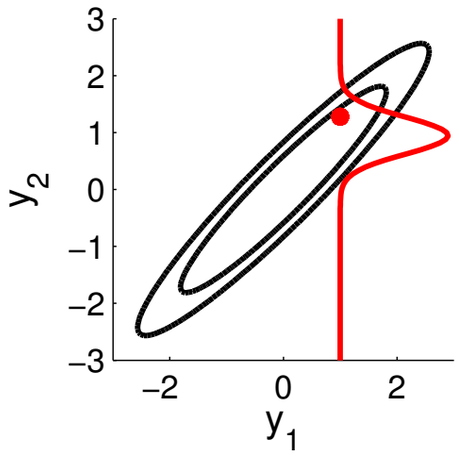
New Visualisation



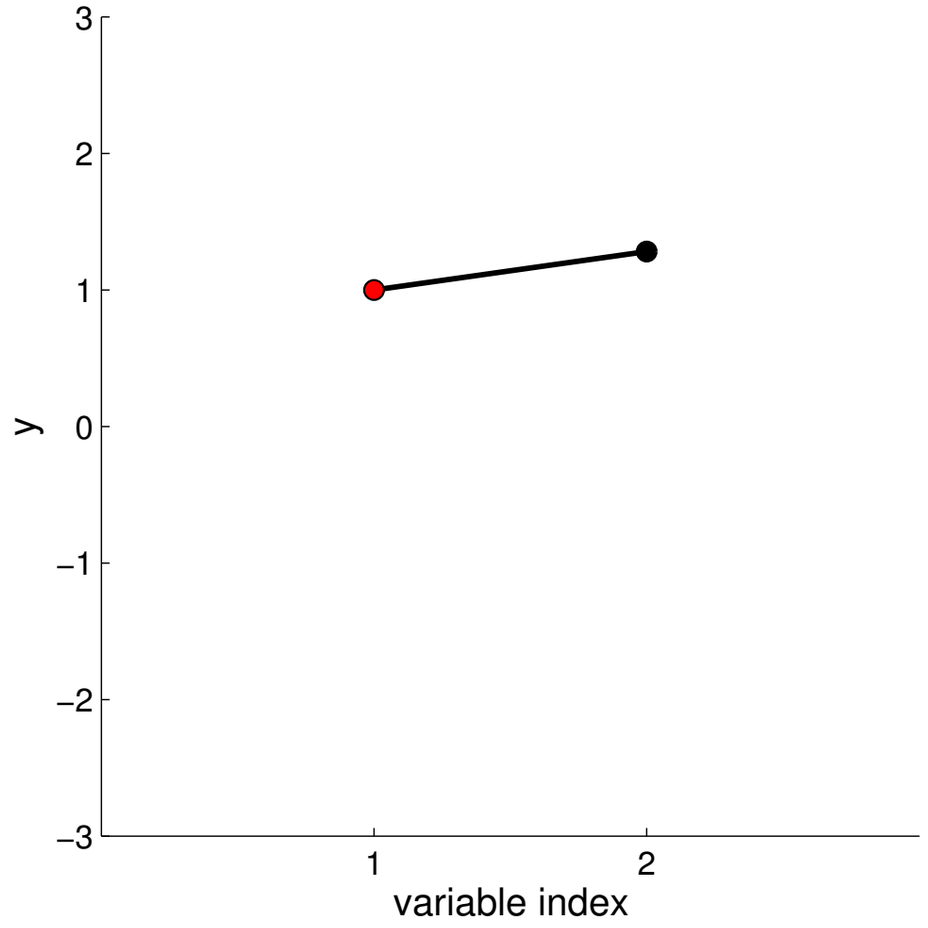
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



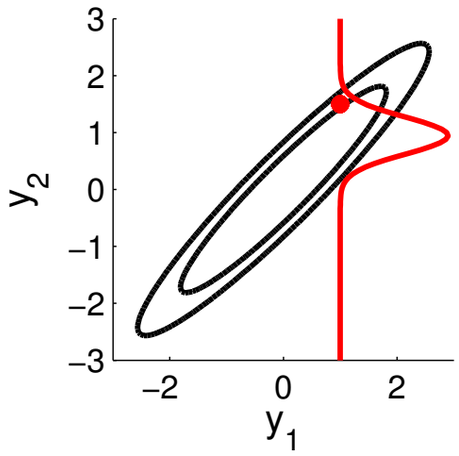
New Visualisation



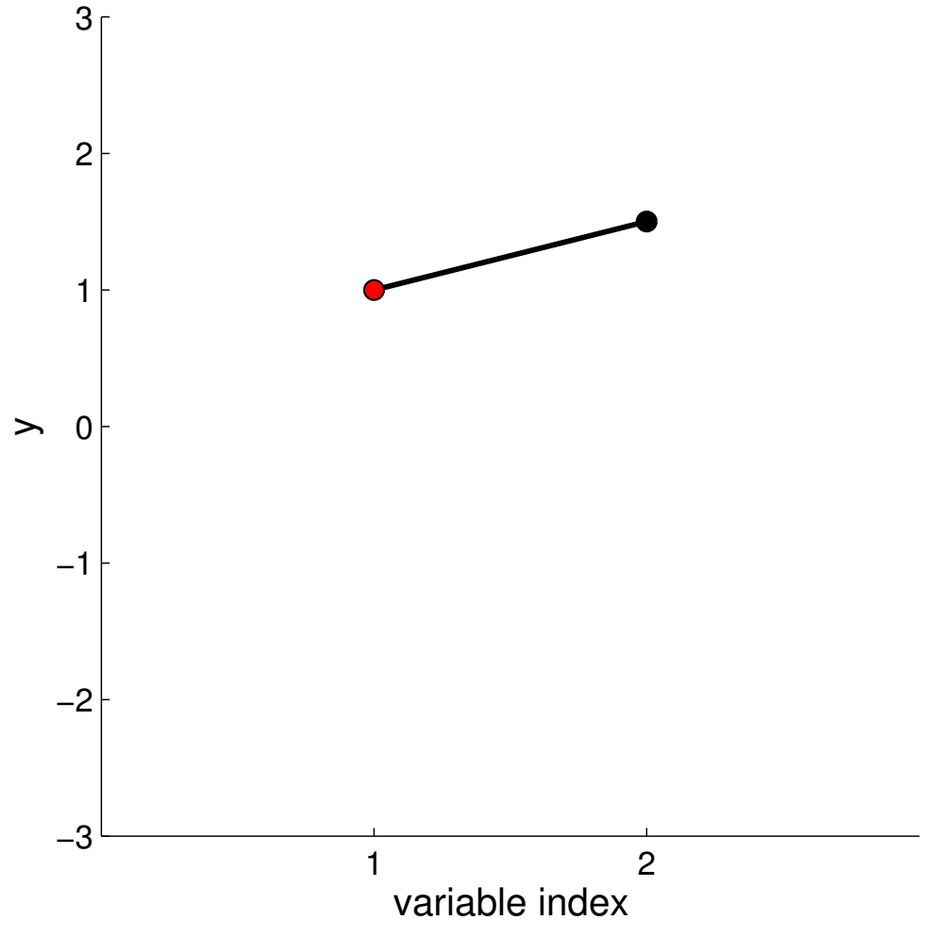
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



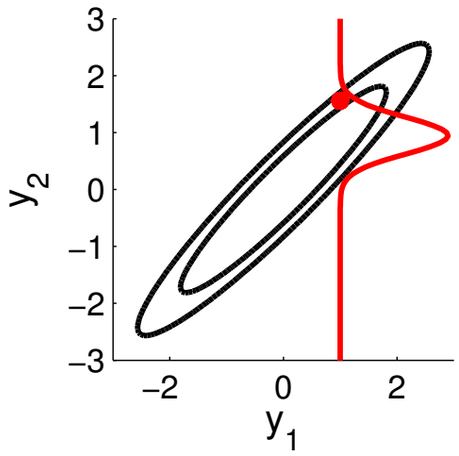
New Visualisation



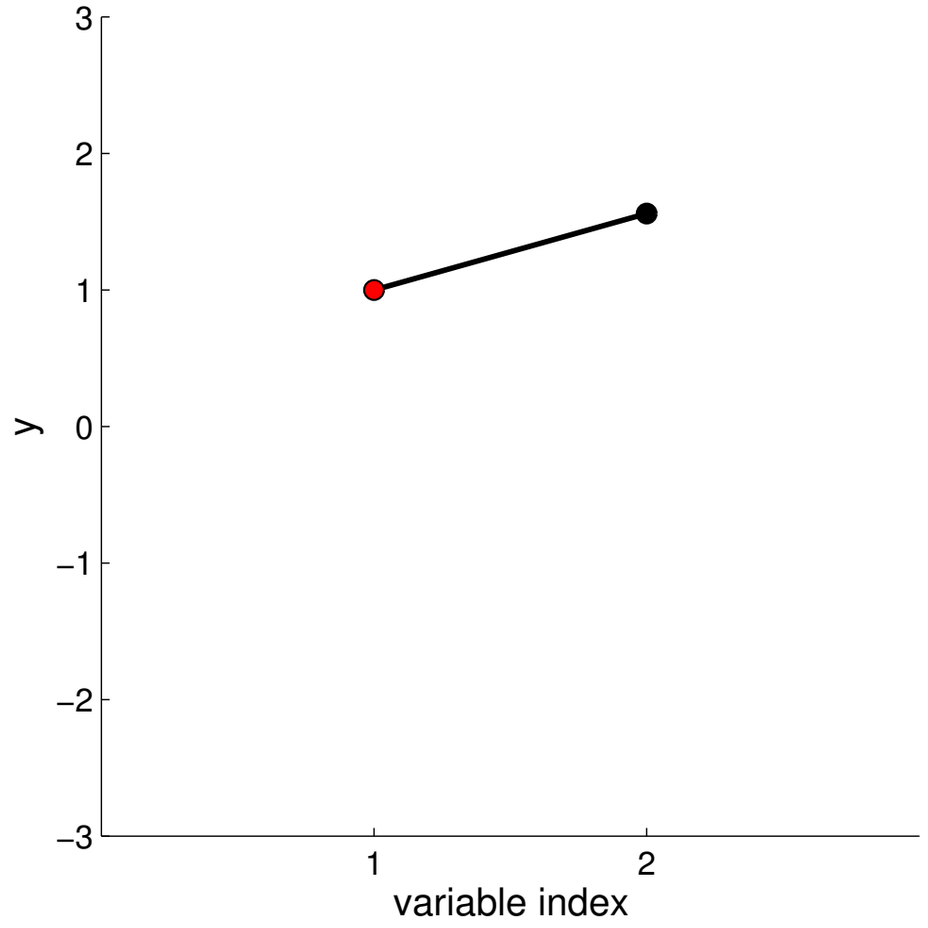
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



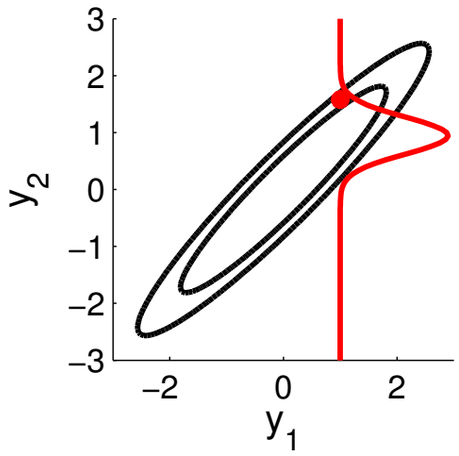
New Visualisation



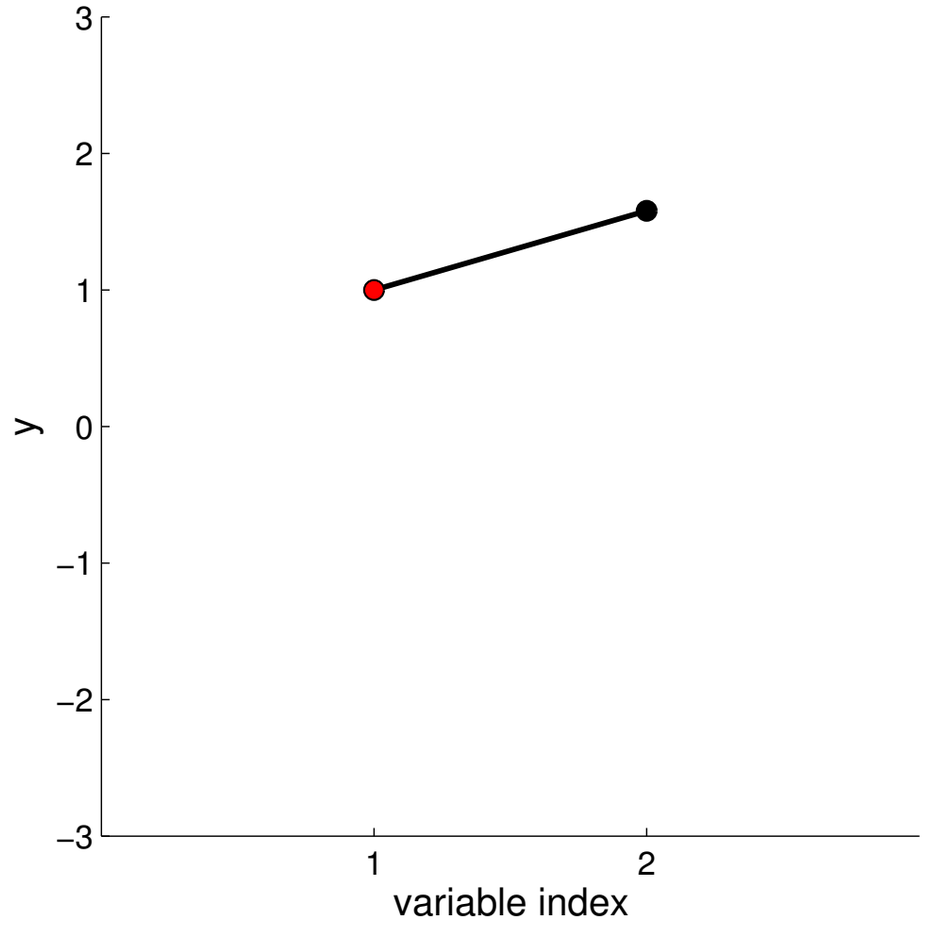
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



New Visualisation

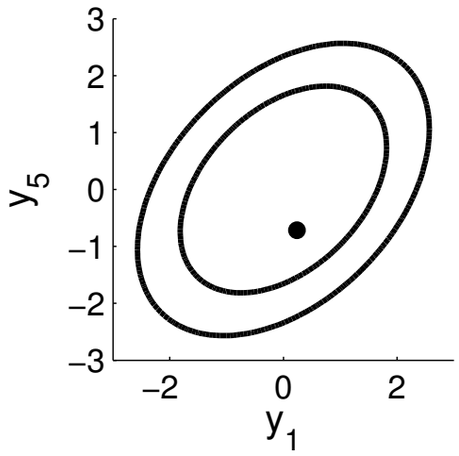


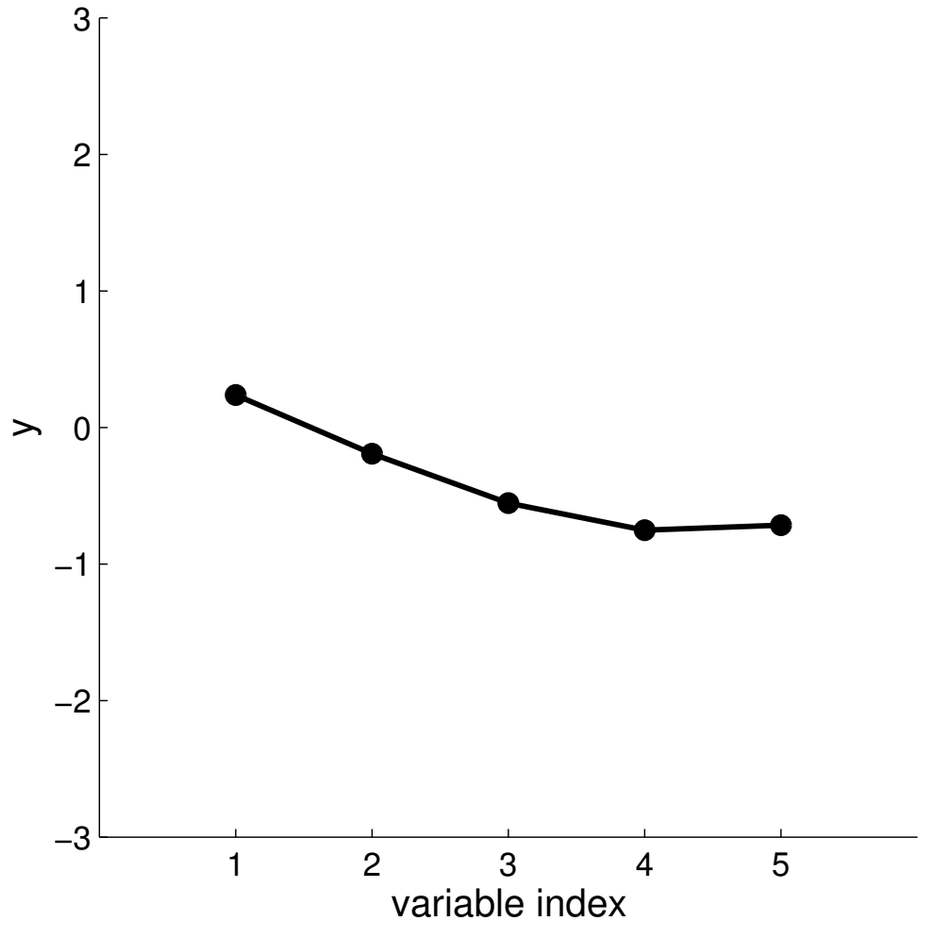
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



Special covariance matrix

- Correlations fall off the further the indices of the variables!



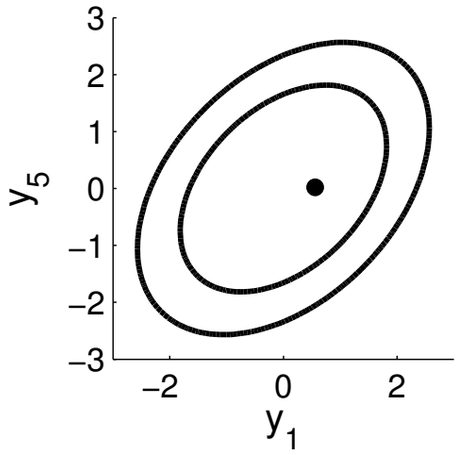


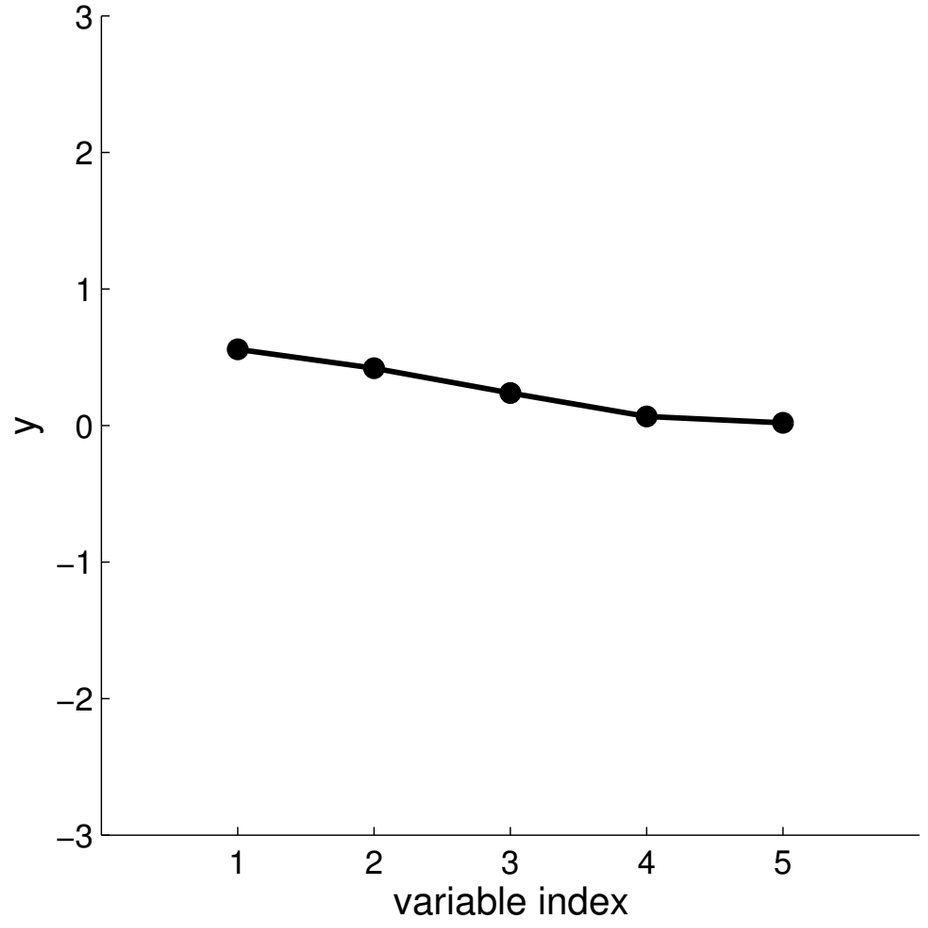
$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

$$k_{ij} = \exp(-\lambda \| x_i - x_j \|^2) = \begin{cases} 0 & \| x_i - x_j \| \to \infty \\ 1 & x_i = x_j \end{cases}$$

Special covariance matrix

- Correlations fall off the further the indices of the variables!

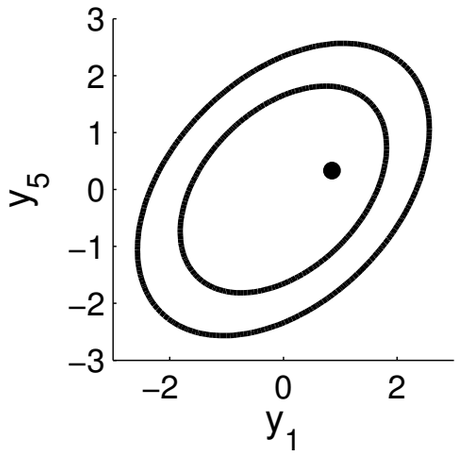


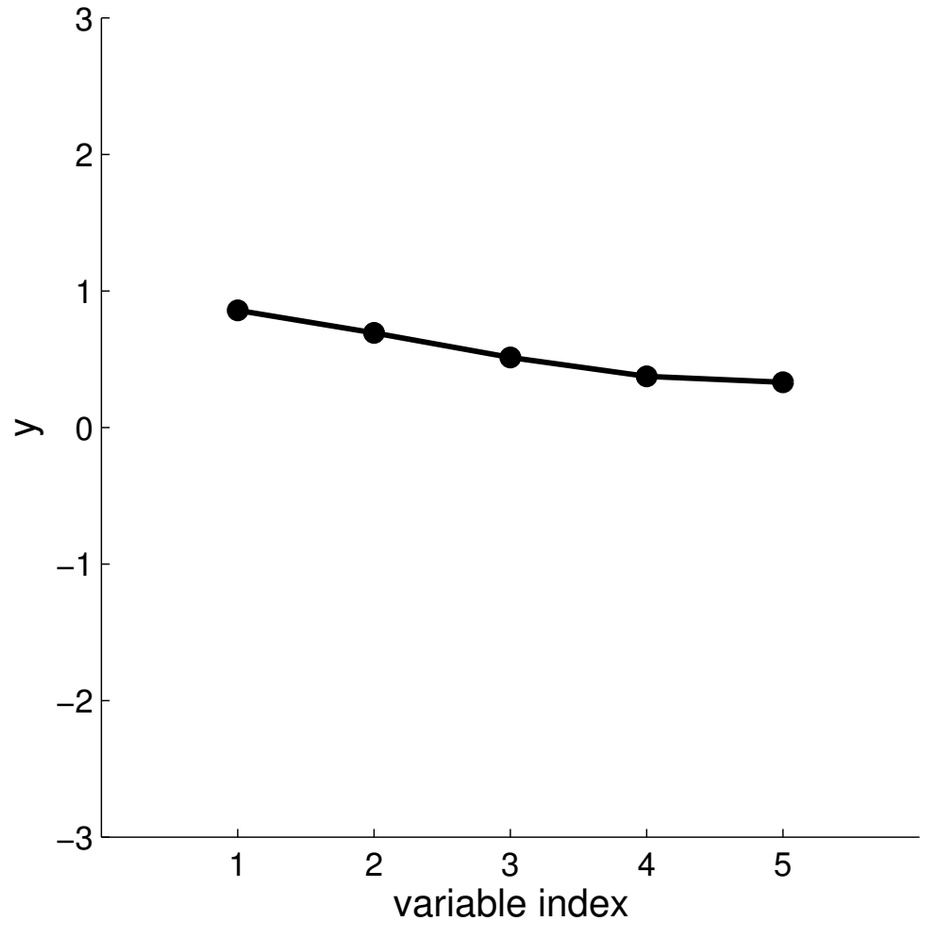


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

- Correlations fall off the further the indices of the variables!

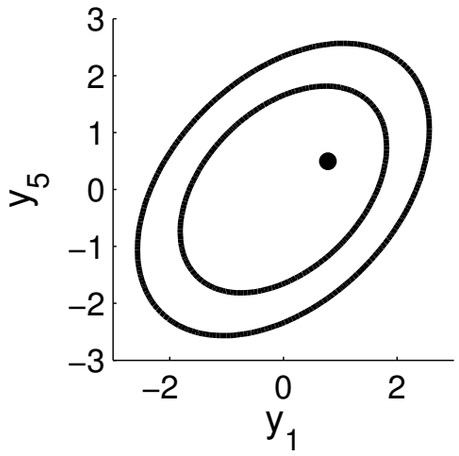


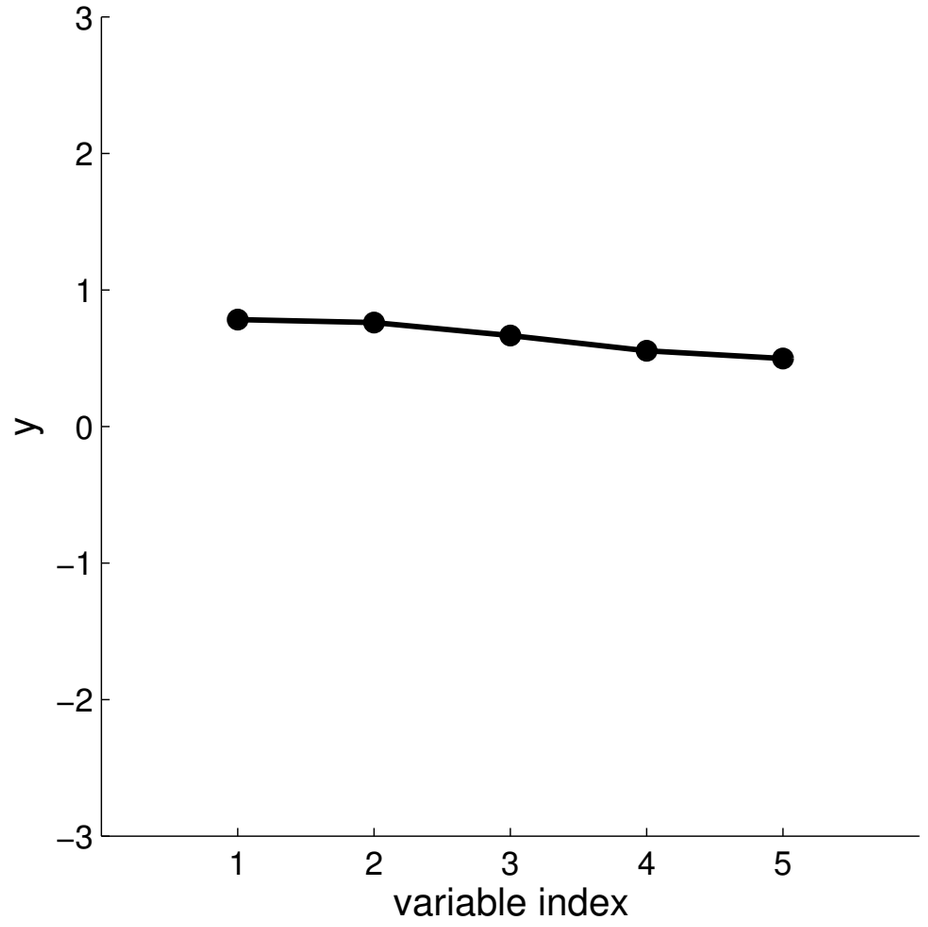


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

- Correlations fall off the further the indices of the variables!

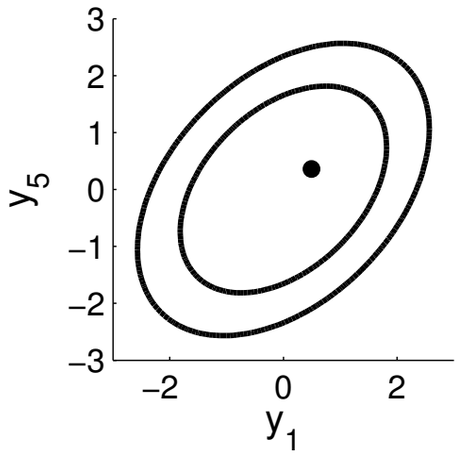


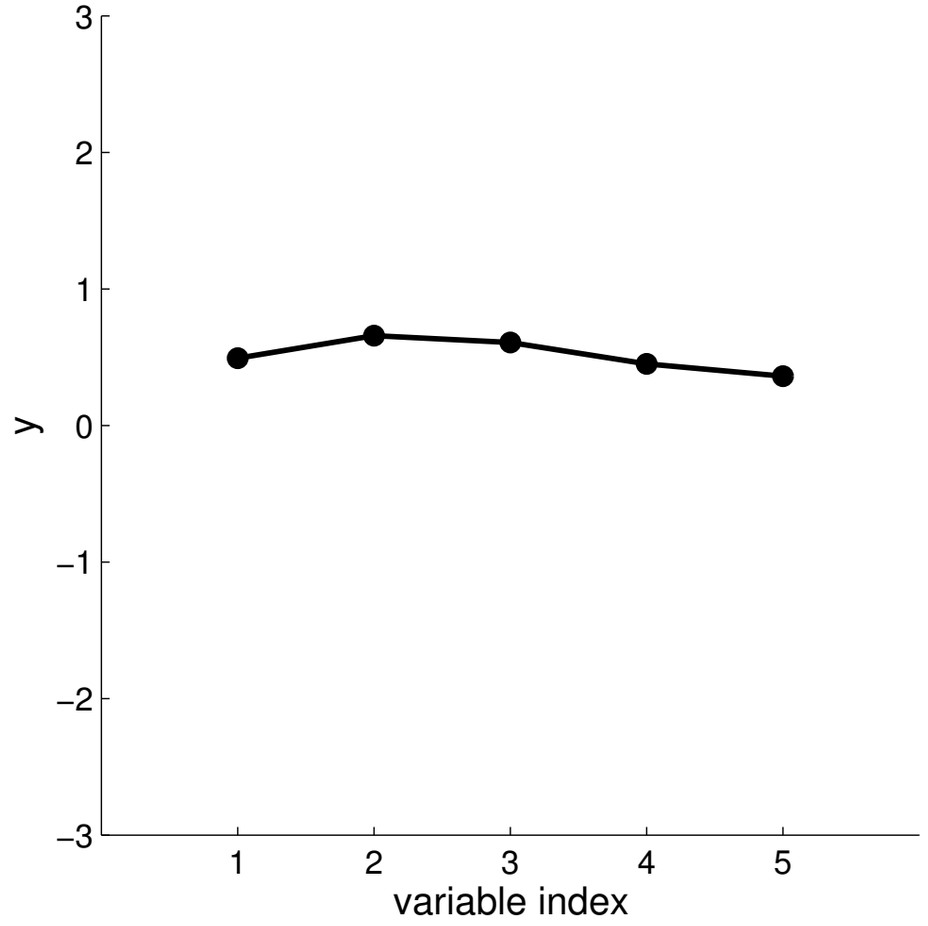


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

- Correlations fall off the further the indices of the variables!

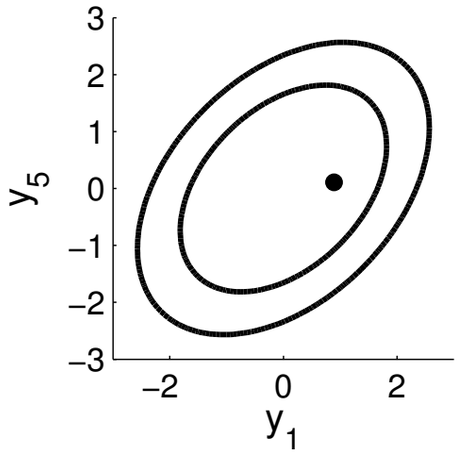


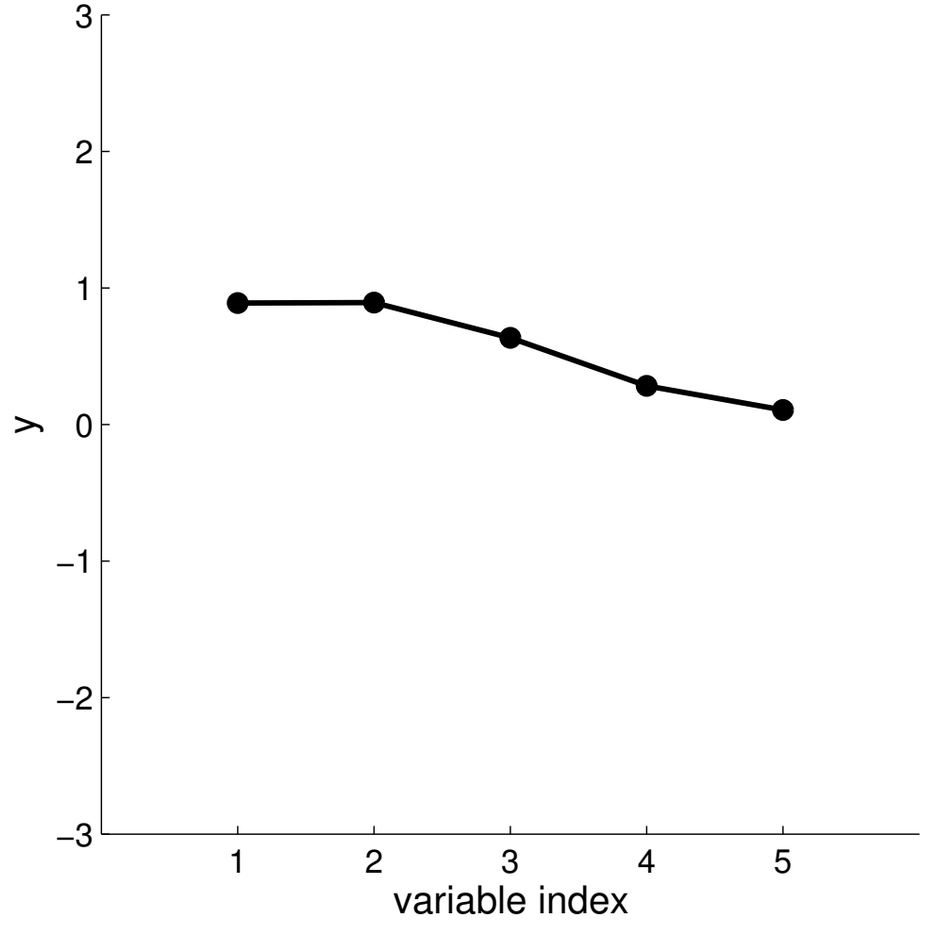


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

- Correlations fall off the further the indices of the variables!

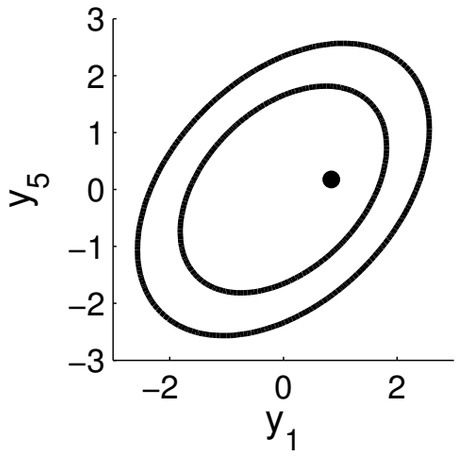


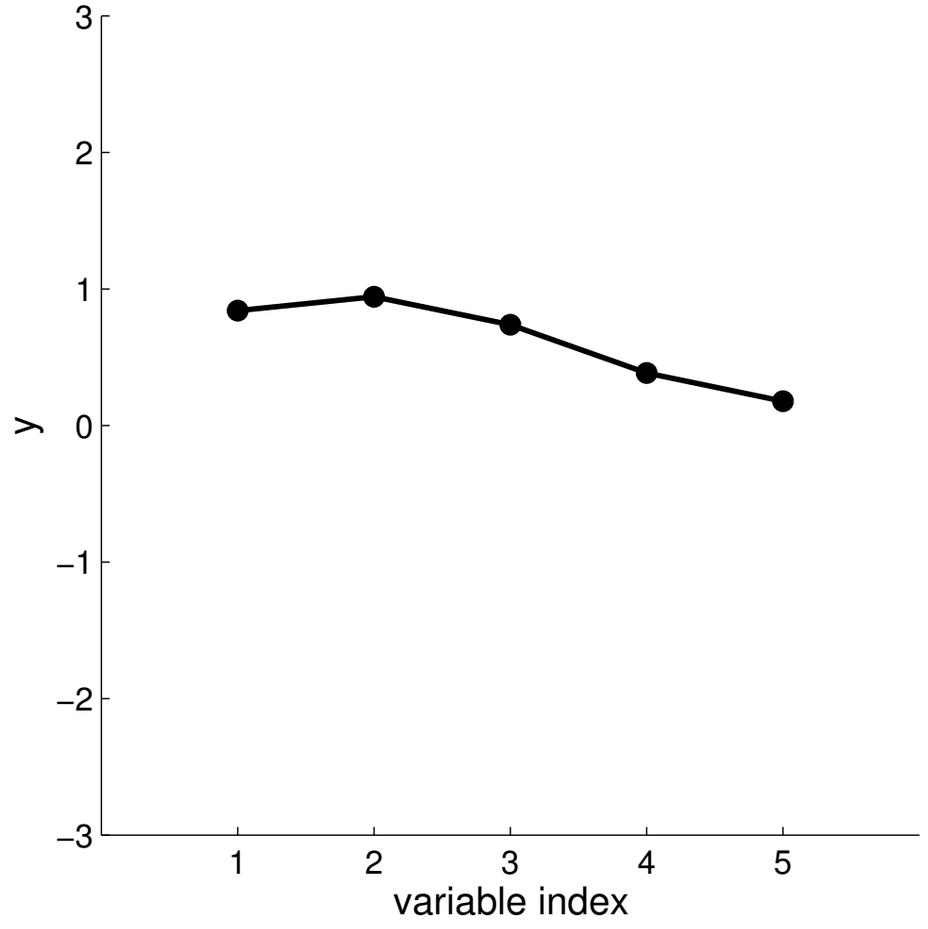


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

- Correlations fall off the further the indices of the variables!

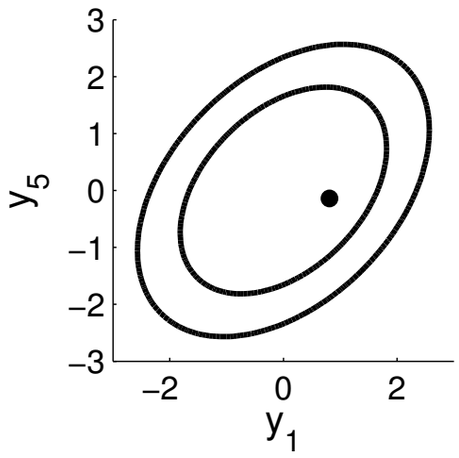


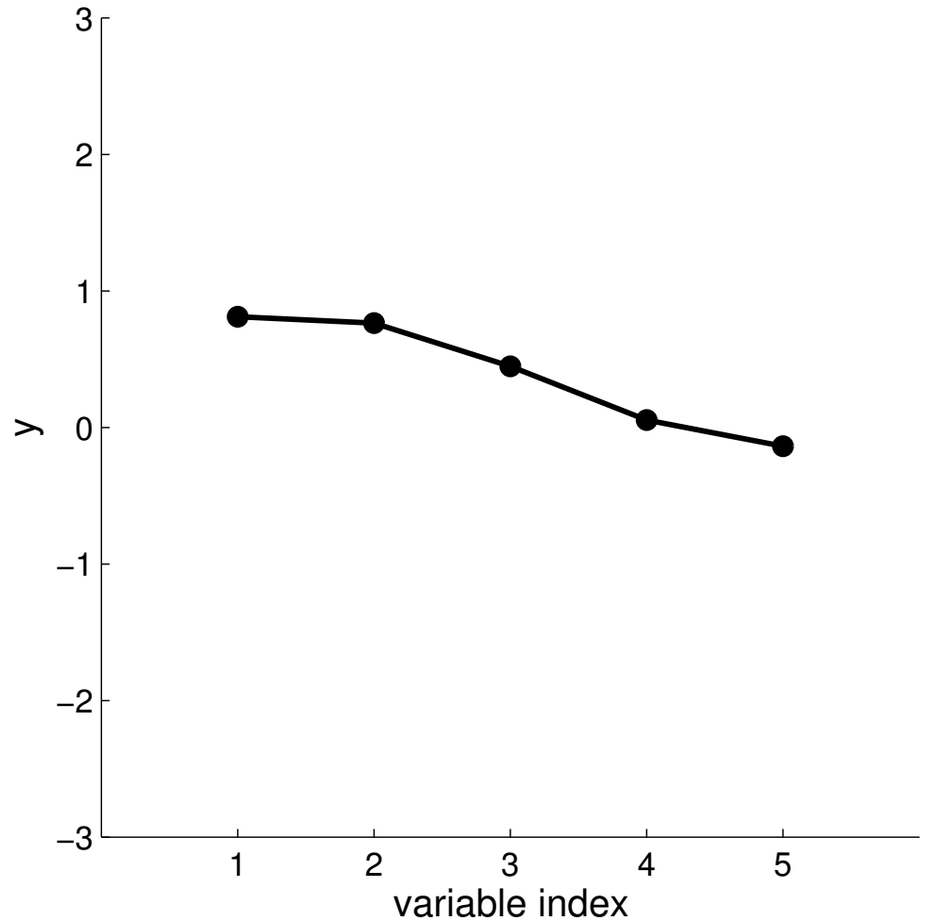


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

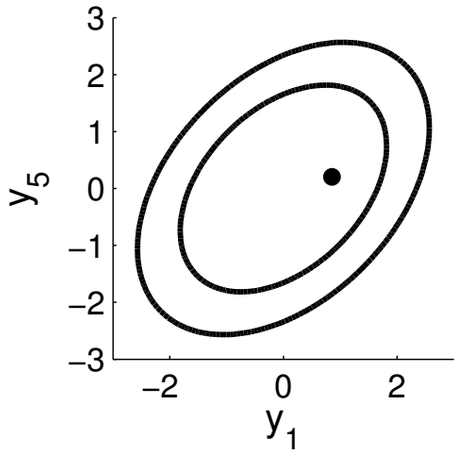


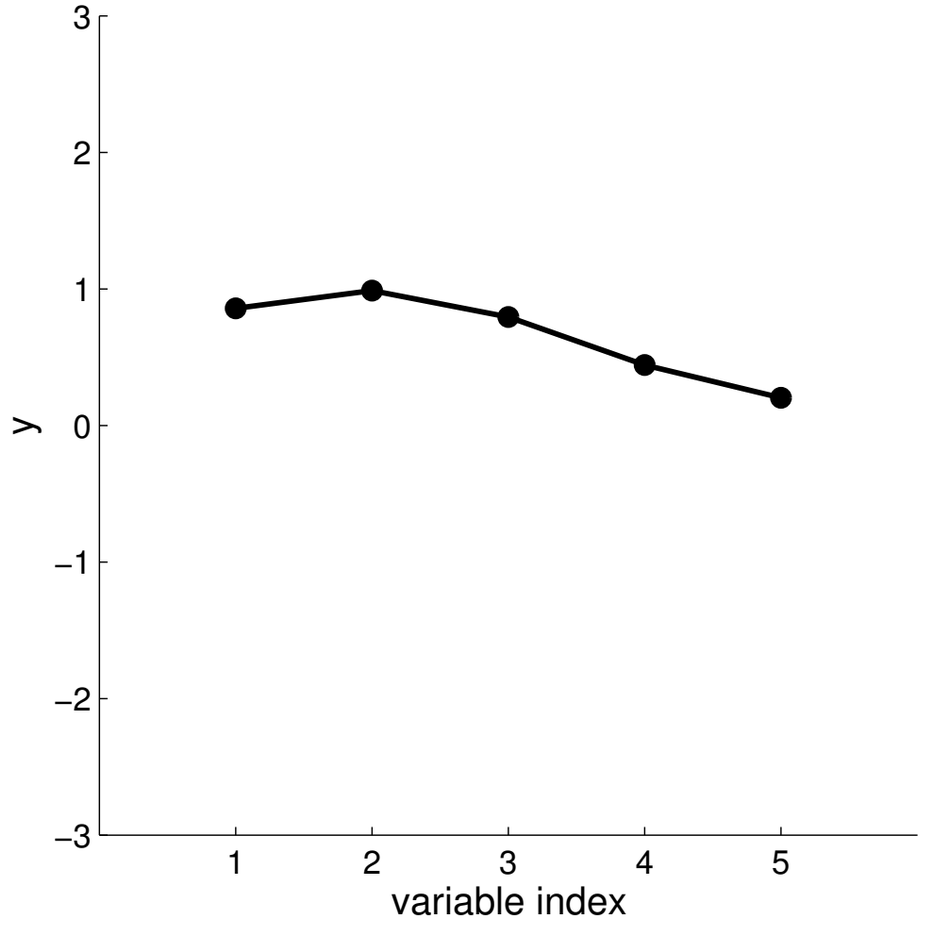


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

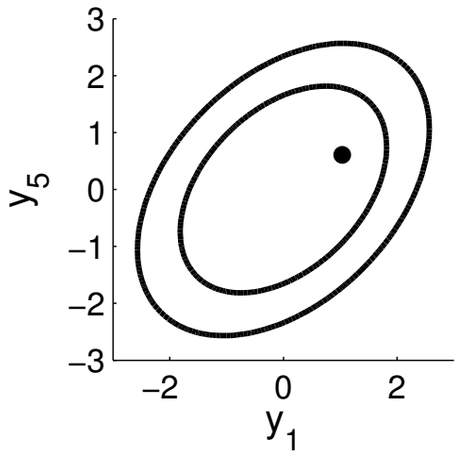


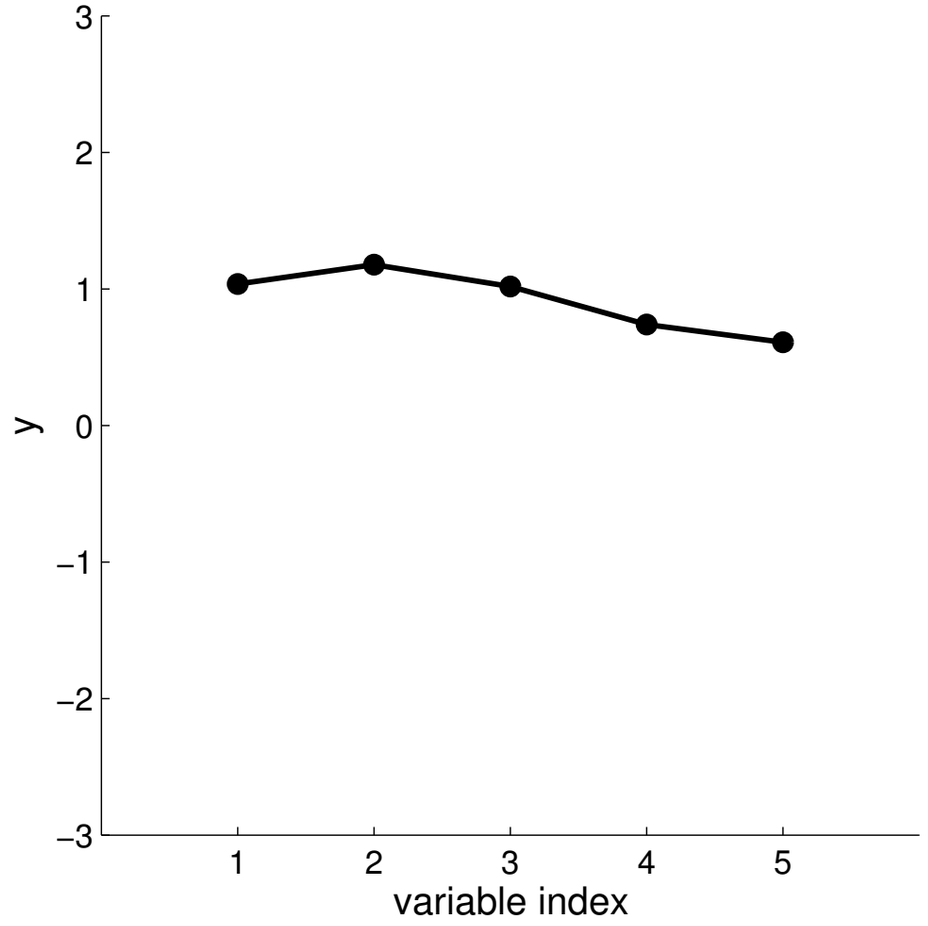


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

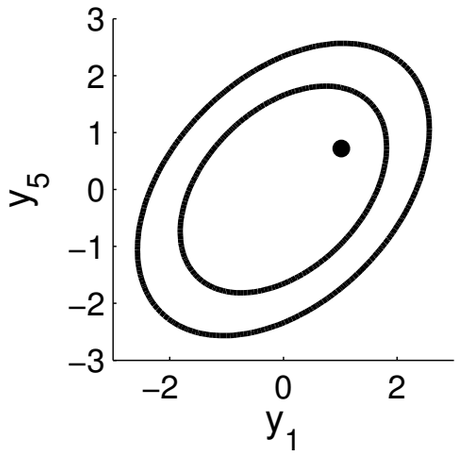


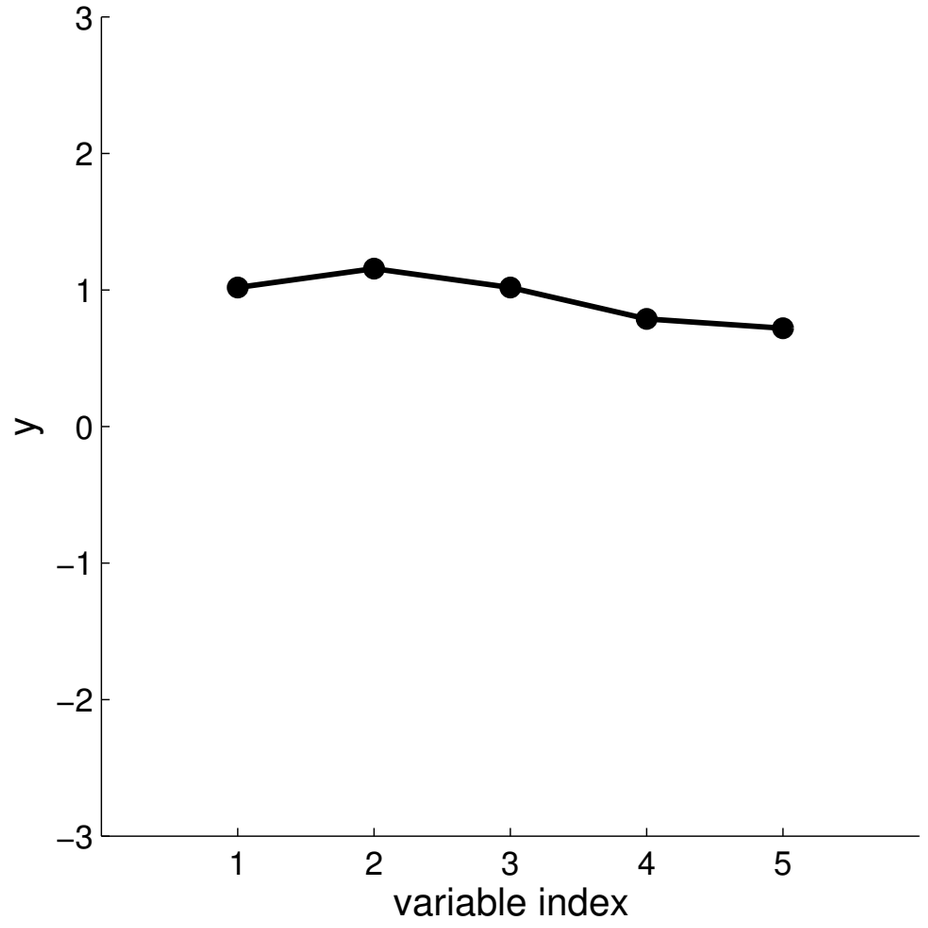


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

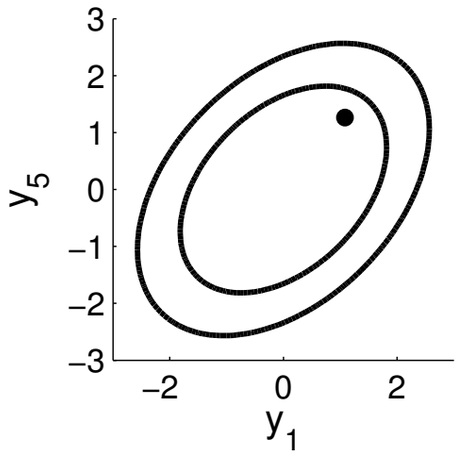


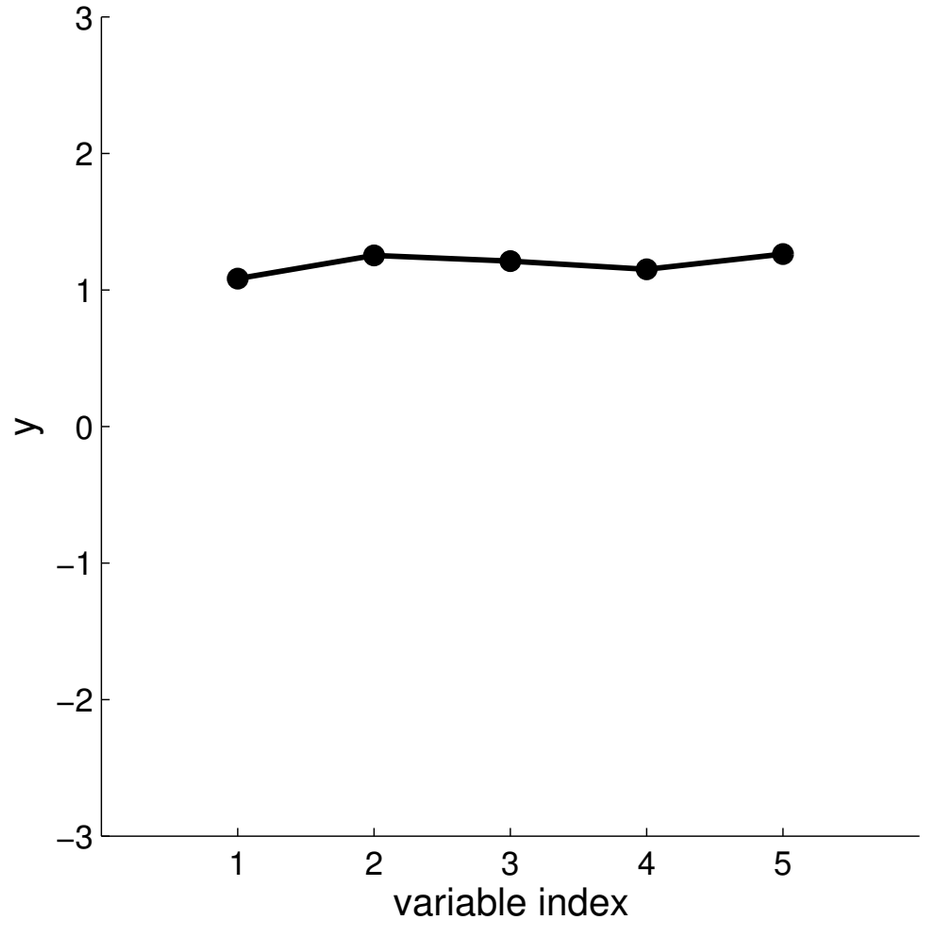


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

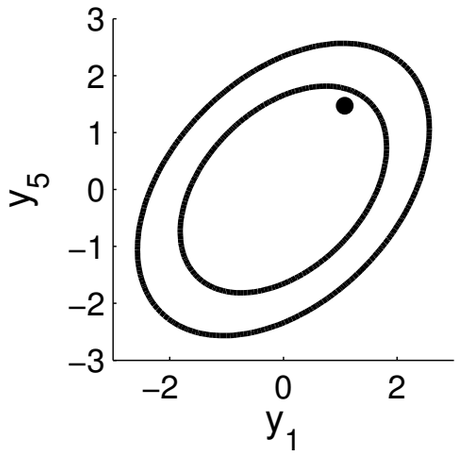


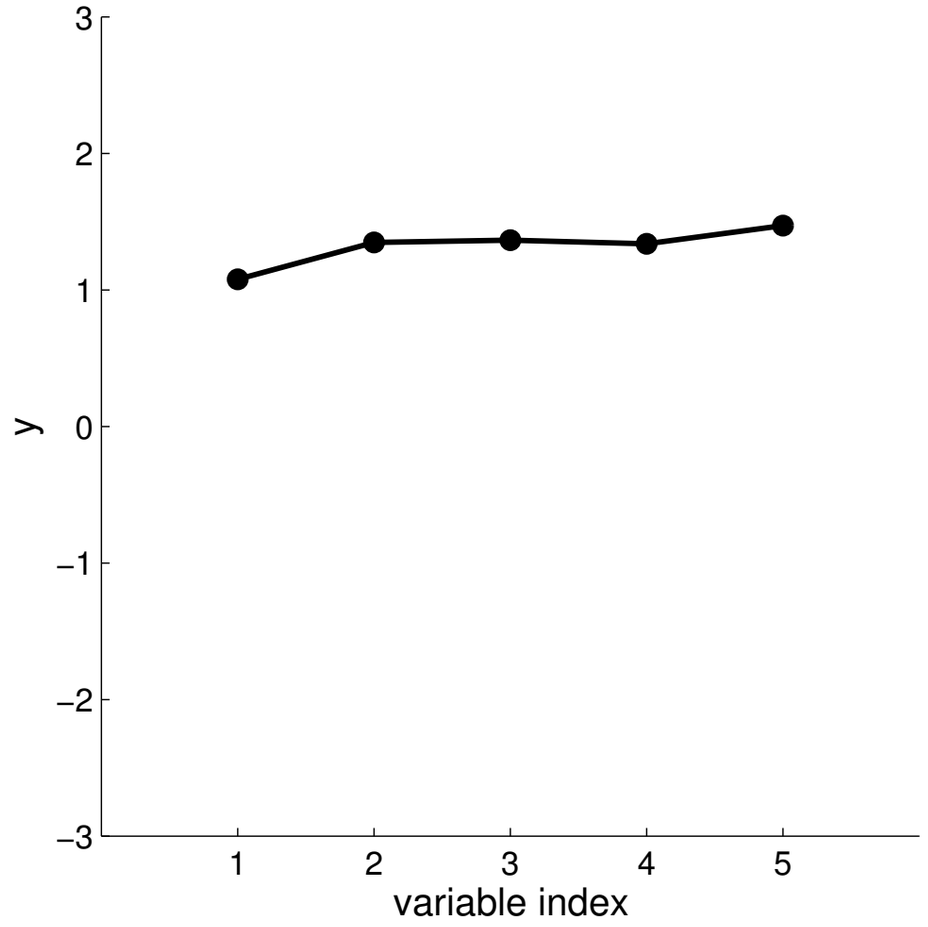


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

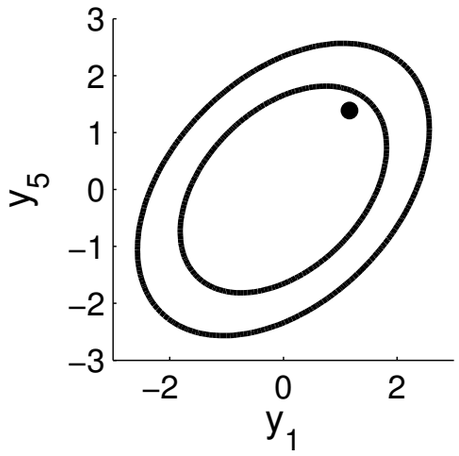


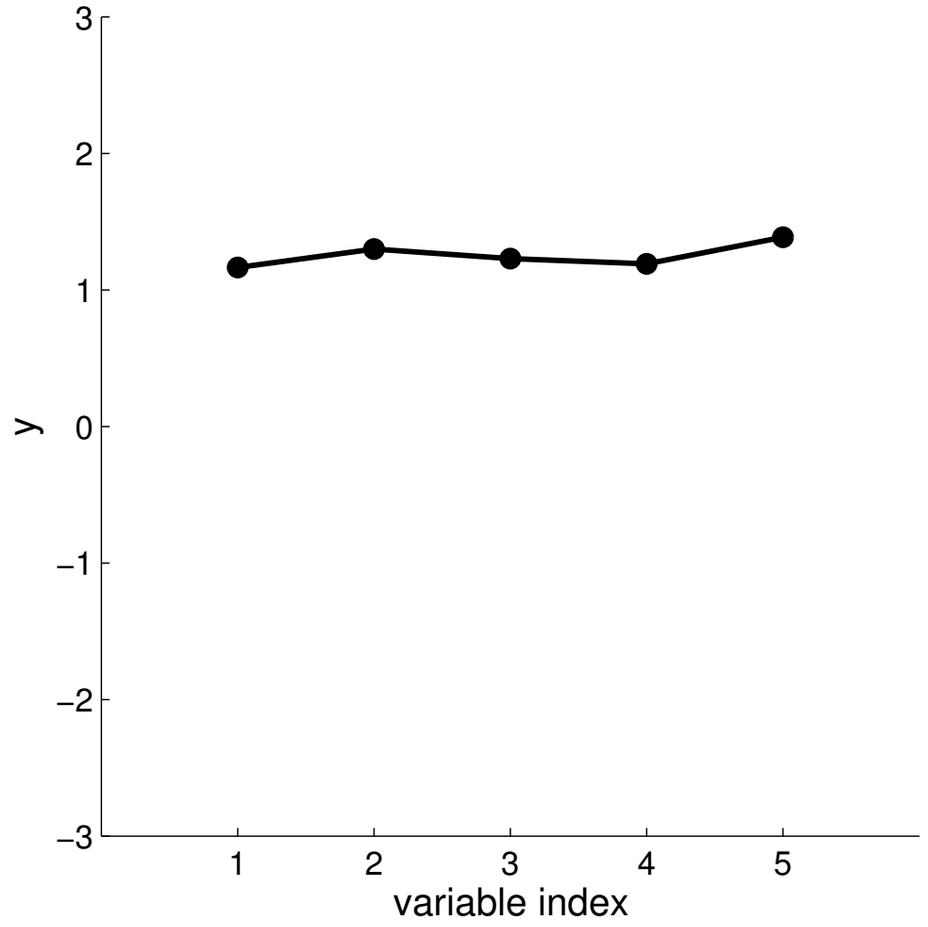


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

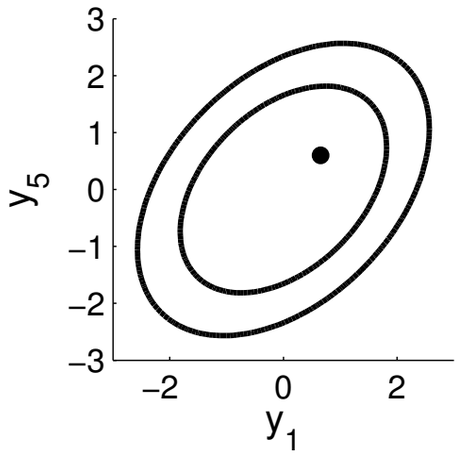


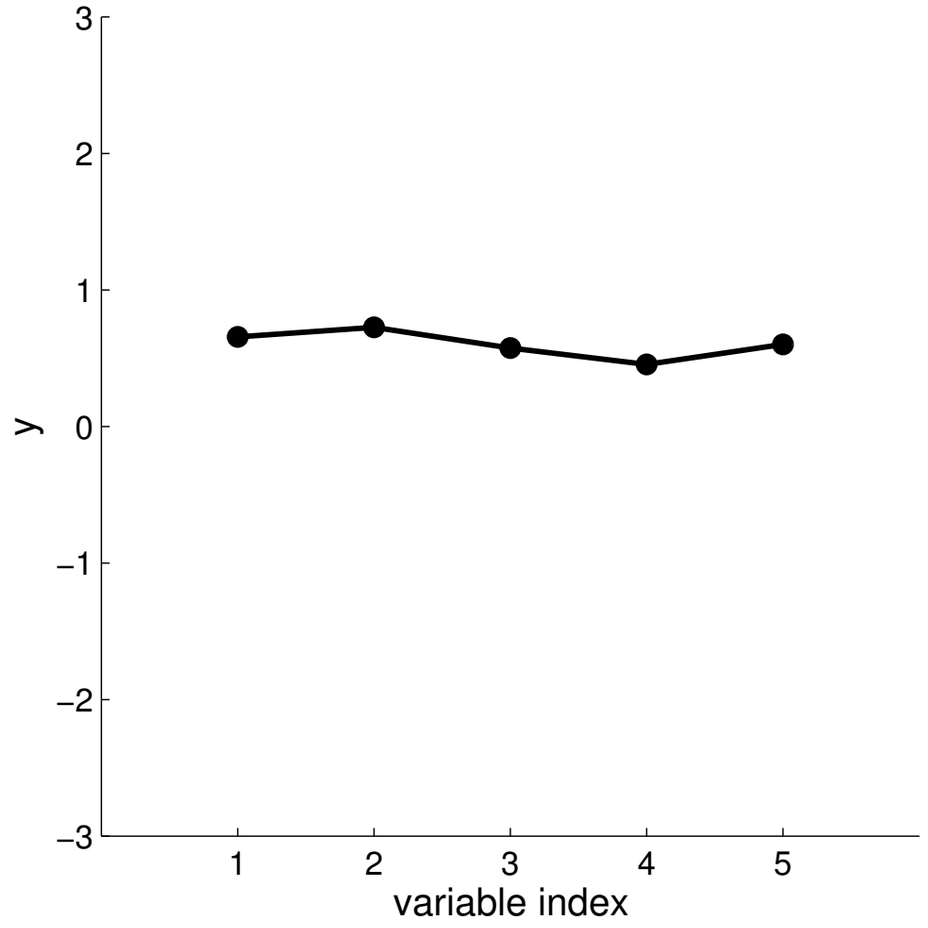


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

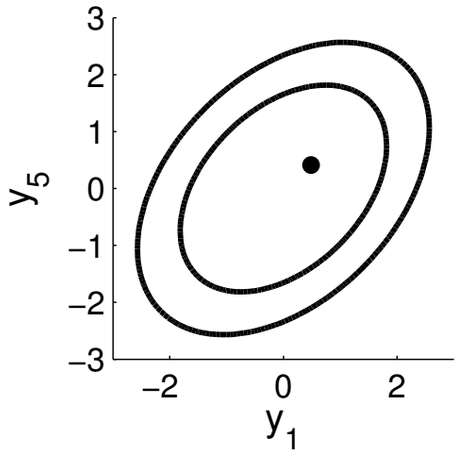


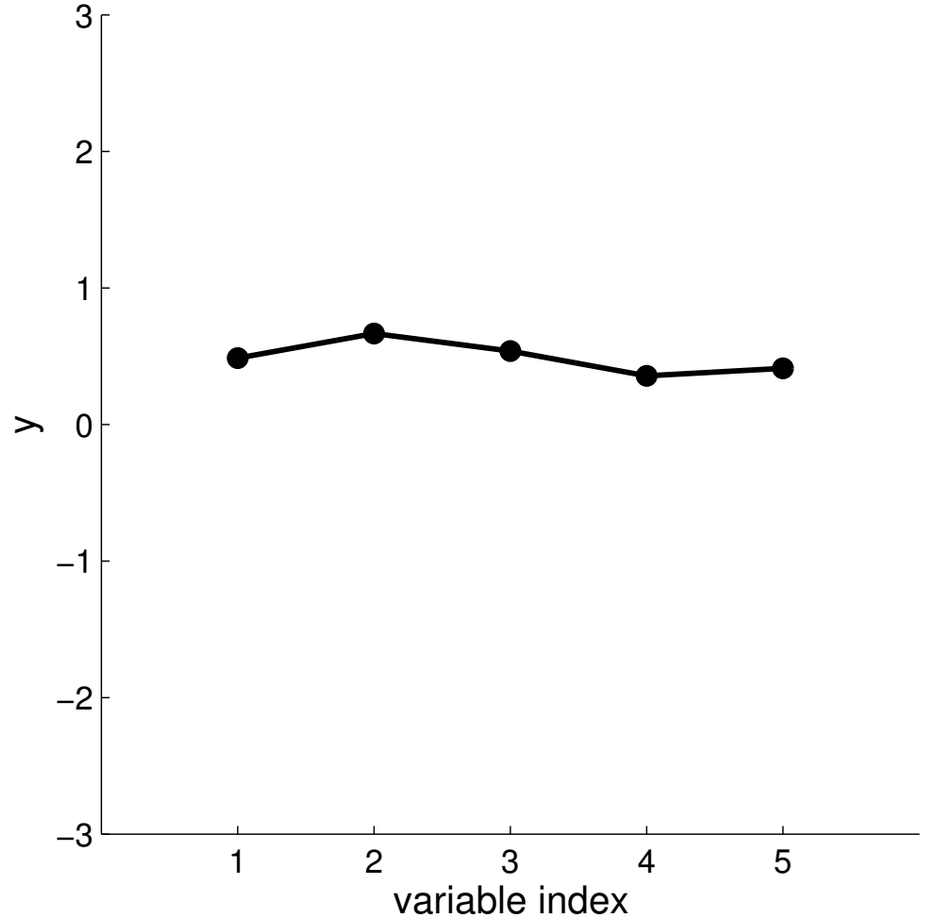


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

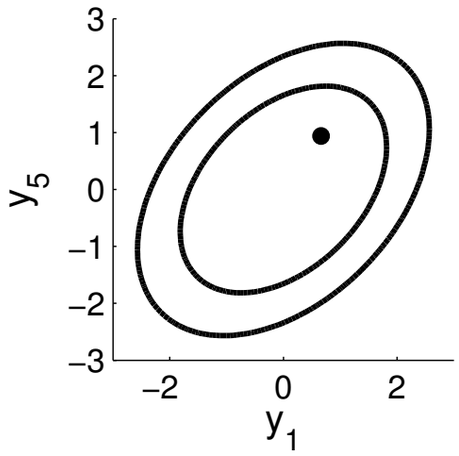


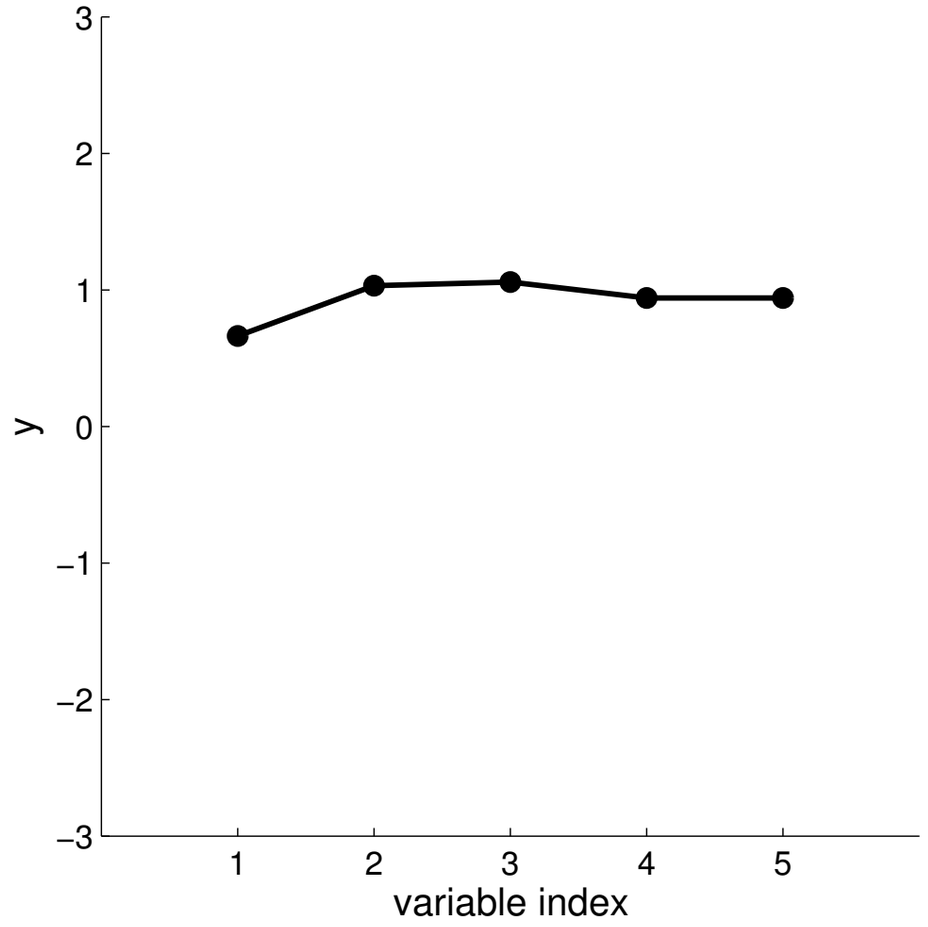


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

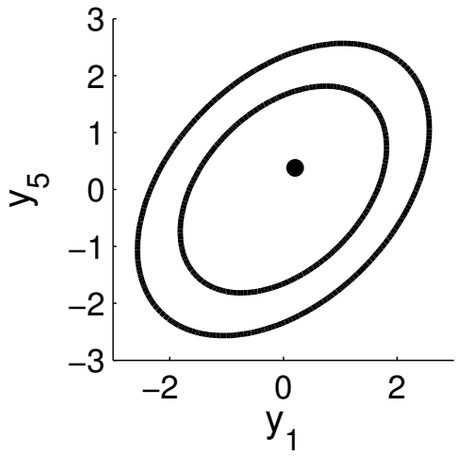


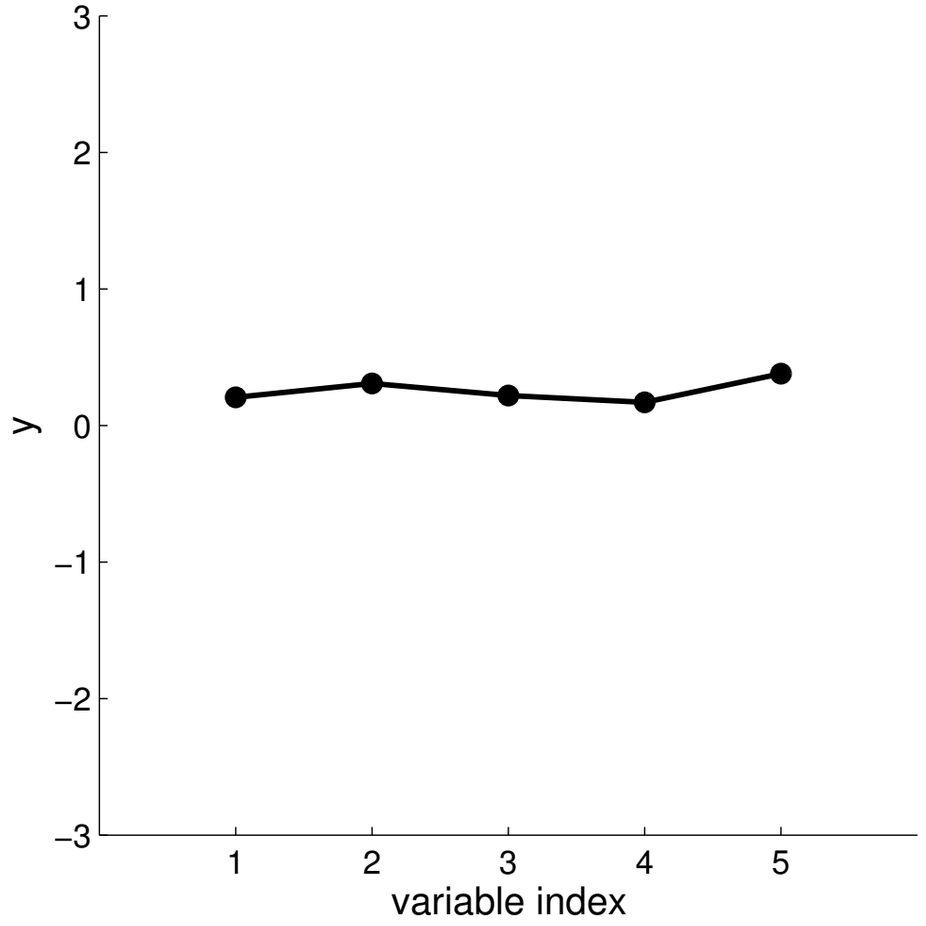


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

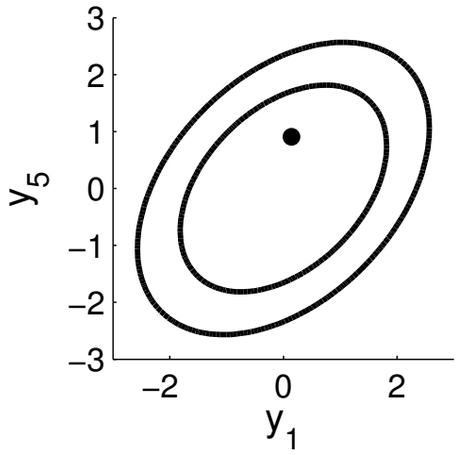


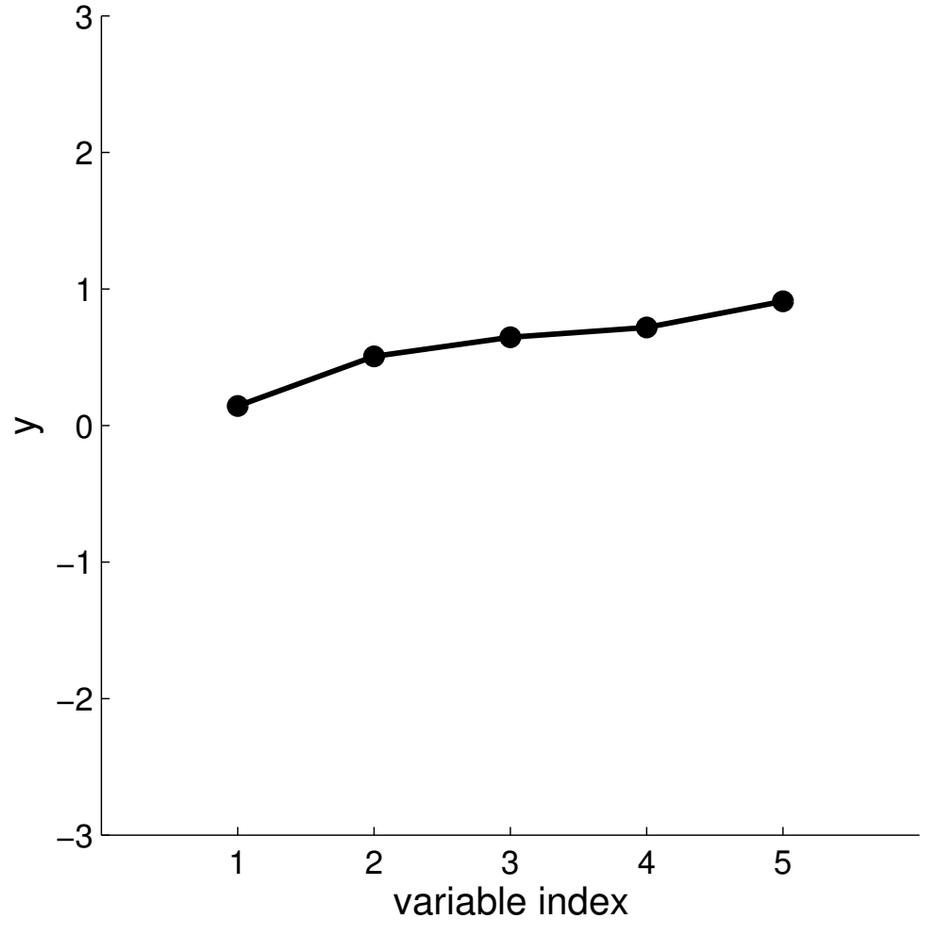


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

Correlations fall off the further the indices of the variables!

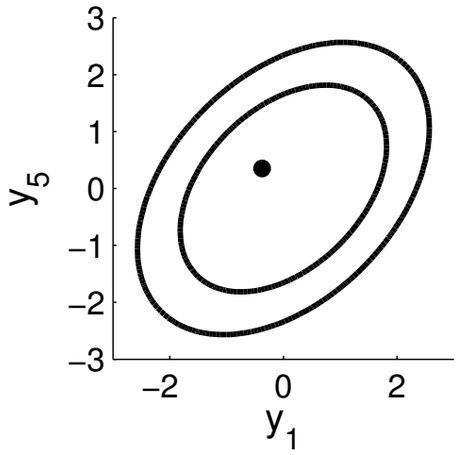


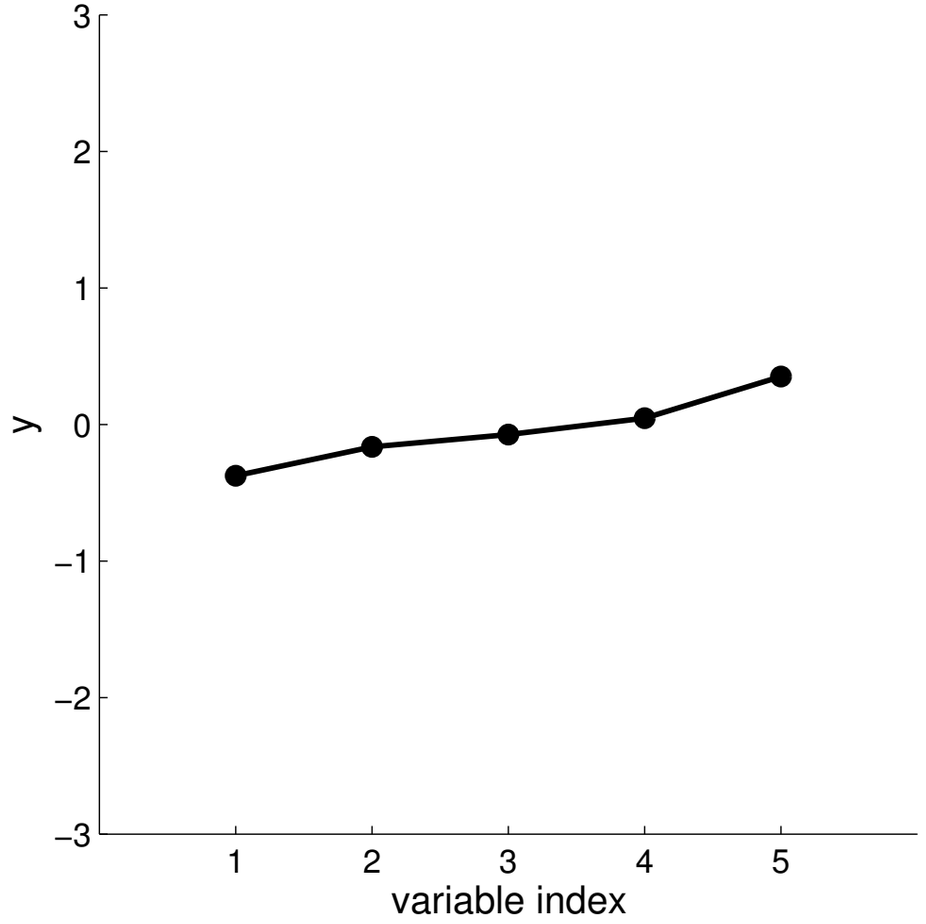


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix

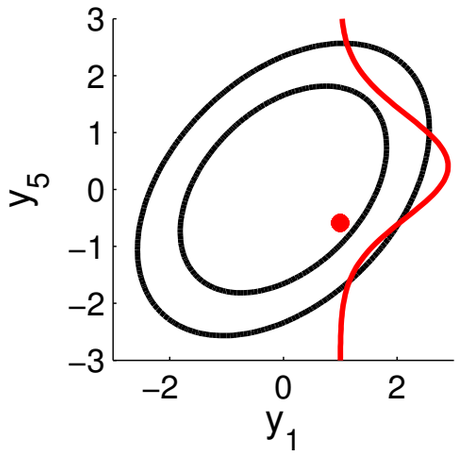
Correlations fall off the further the indices of the variables!

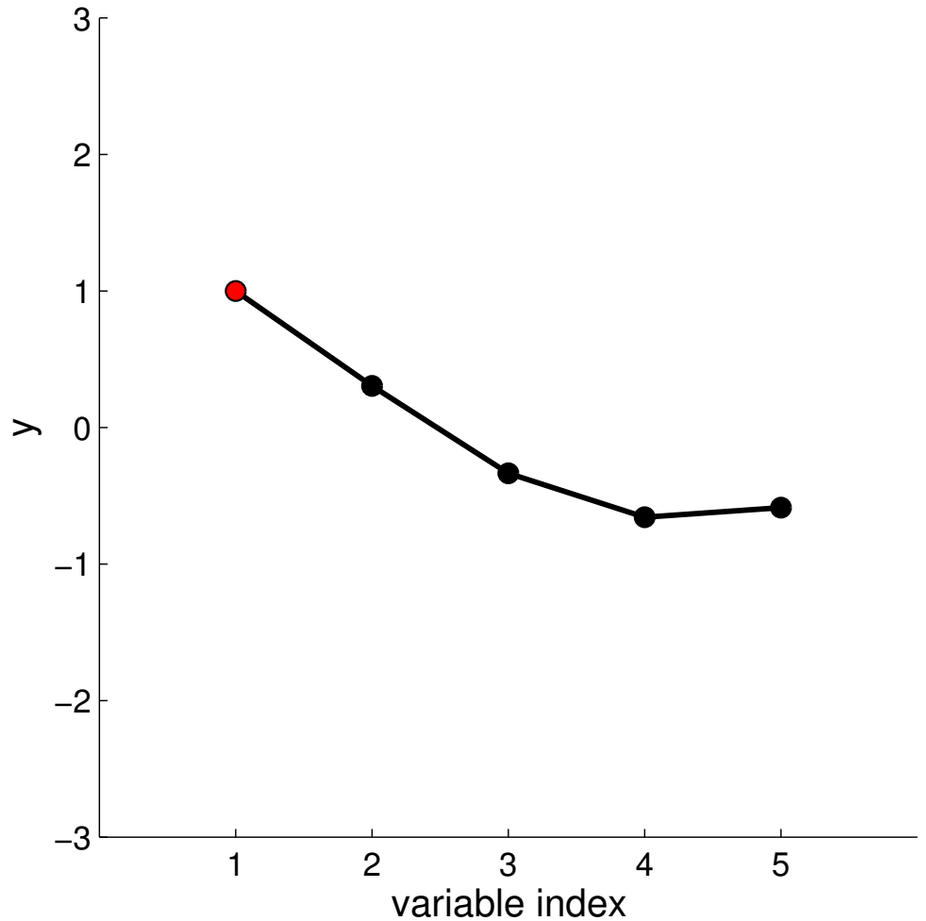




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

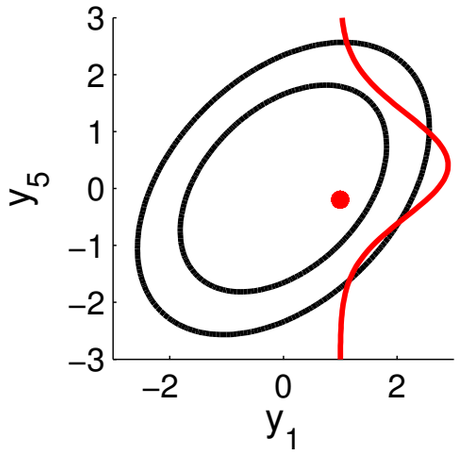
Special covariance matrix - conditioning

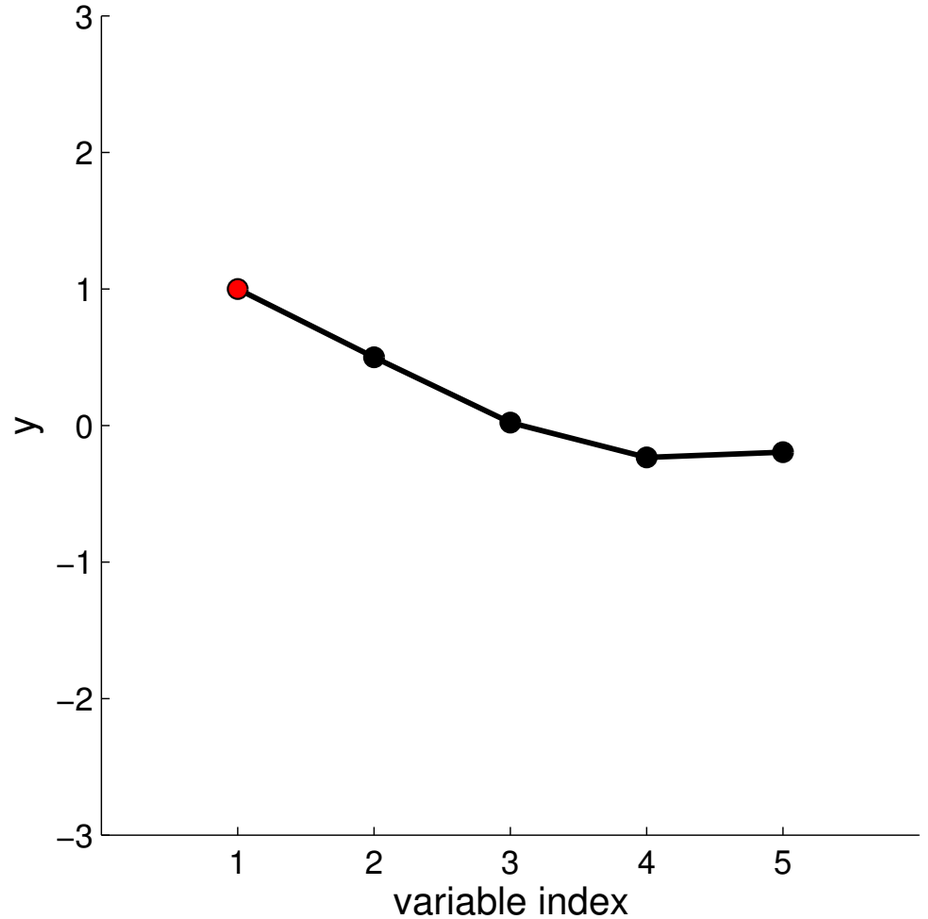




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

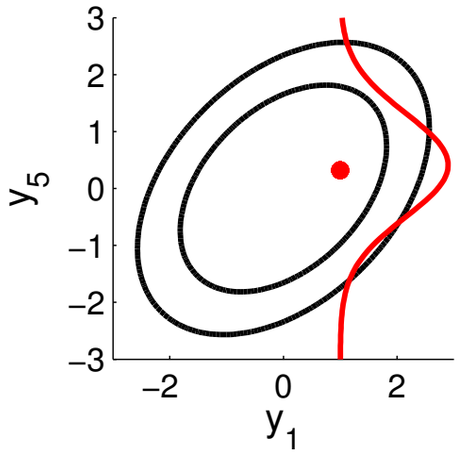
Special covariance matrix - conditioning

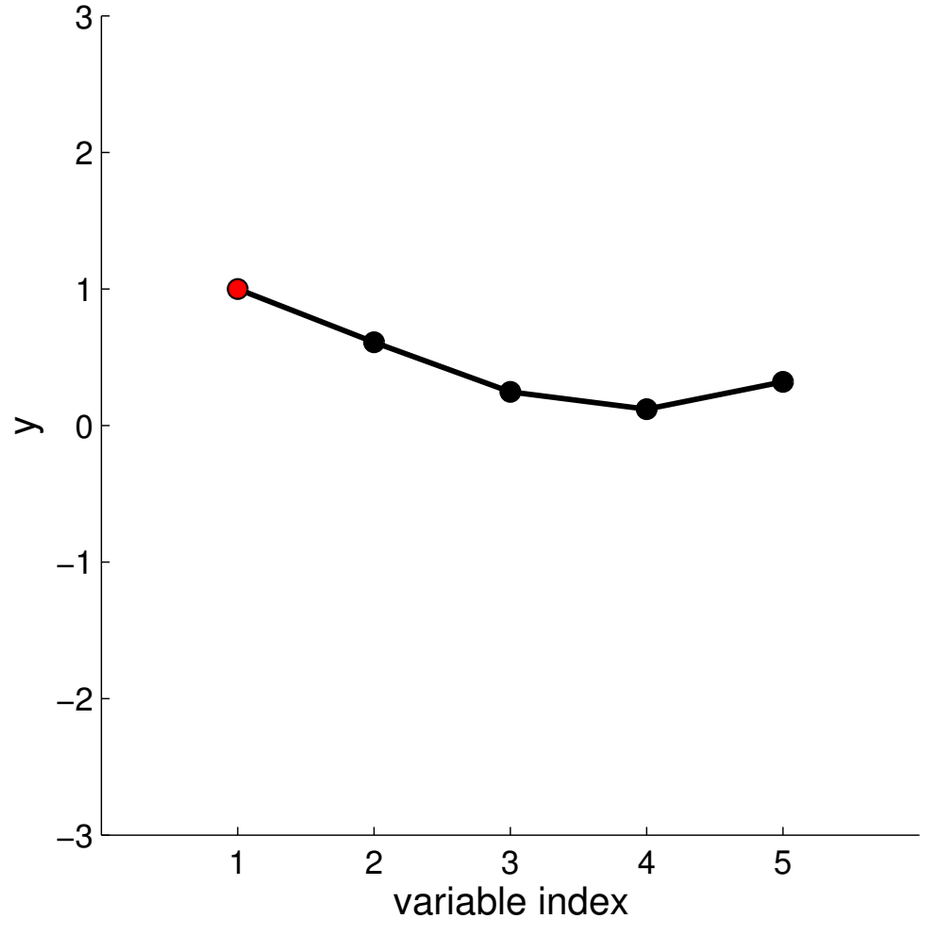




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

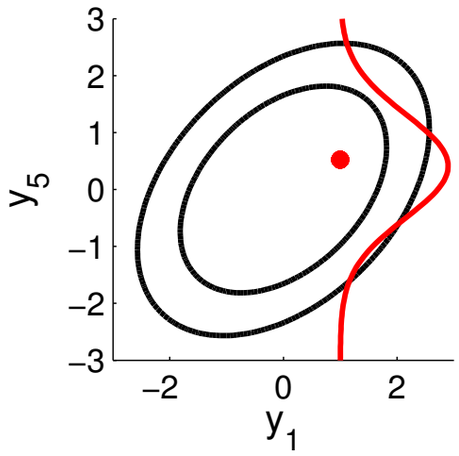
Special covariance matrix - conditioning

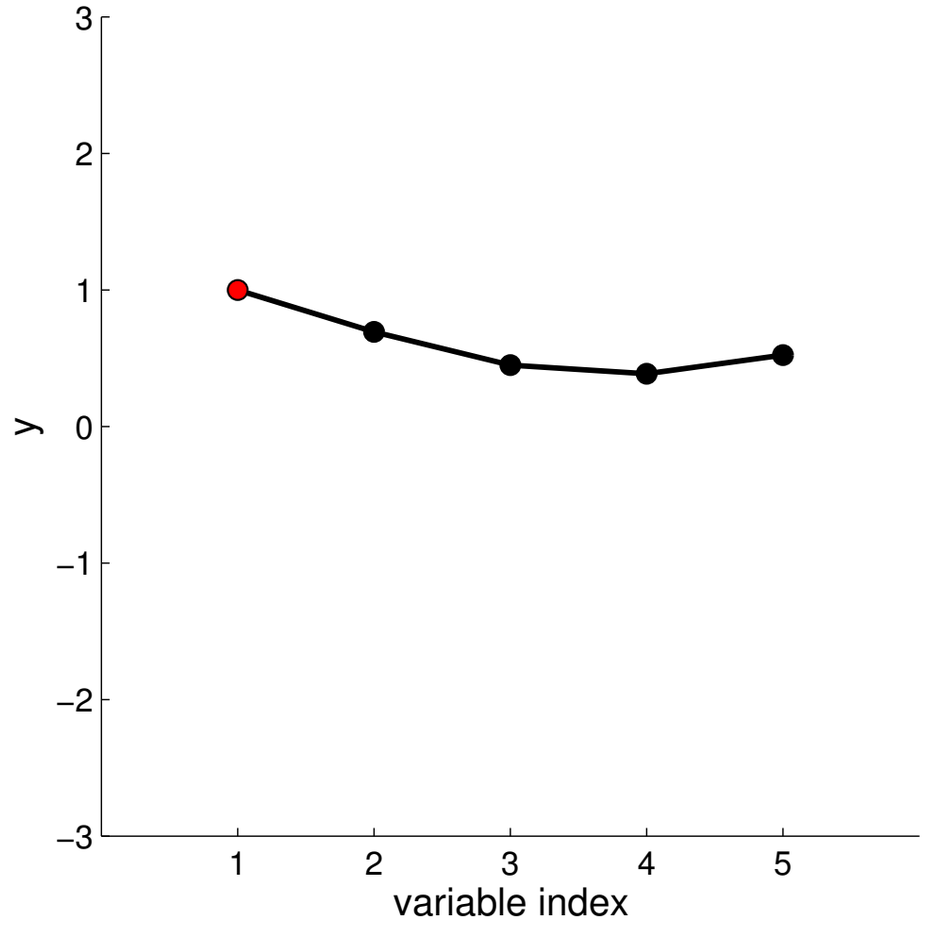




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

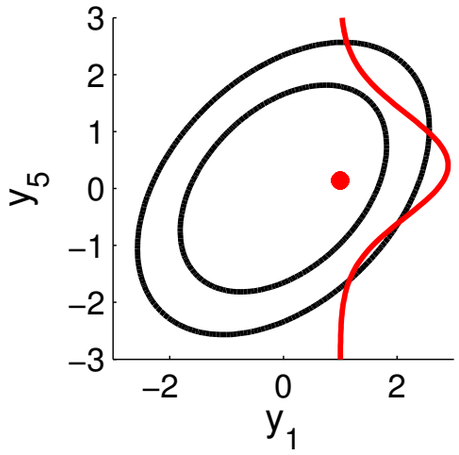
Special covariance matrix - conditioning

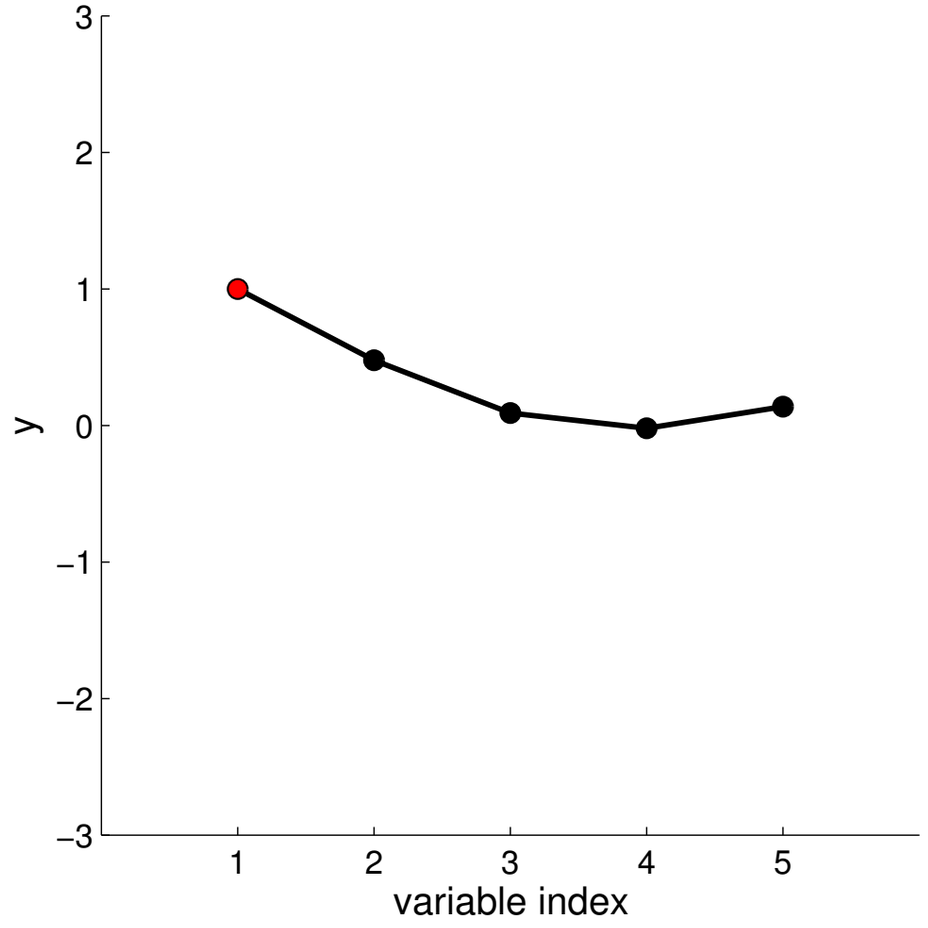




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

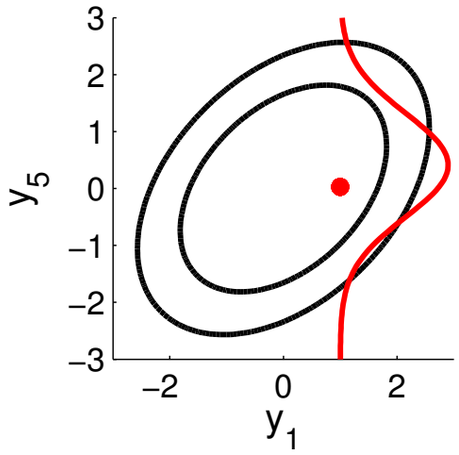
Special covariance matrix - conditioning

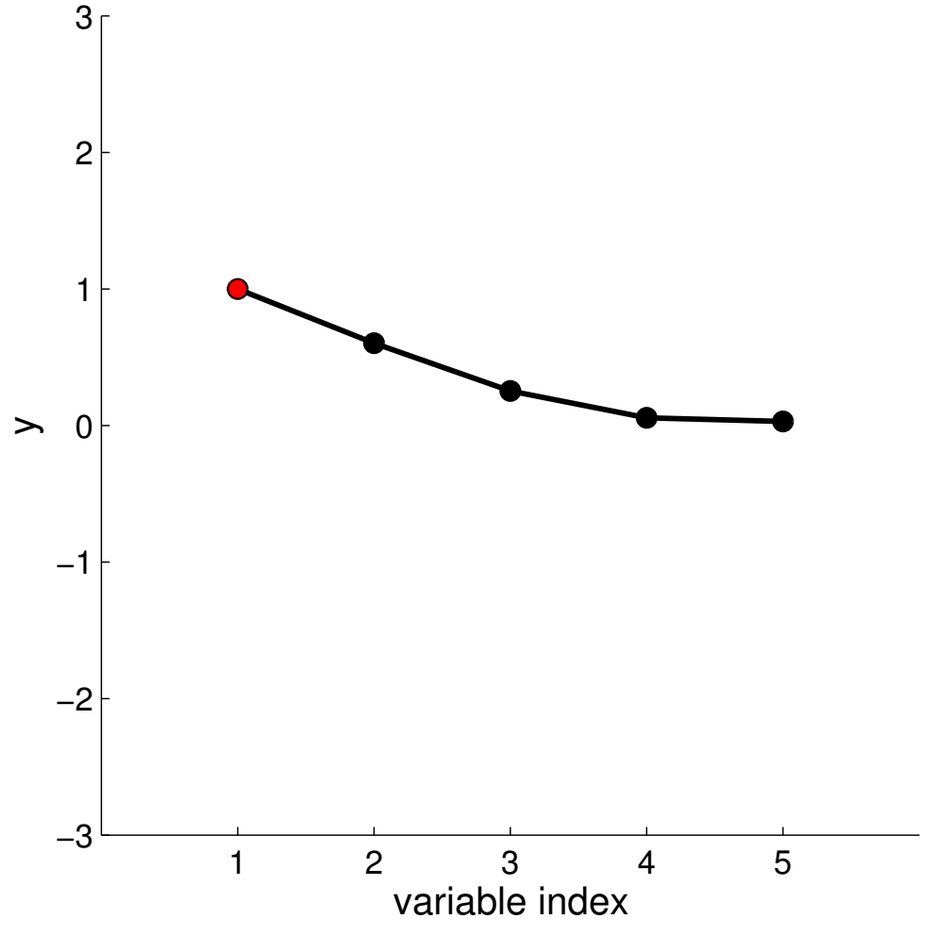




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

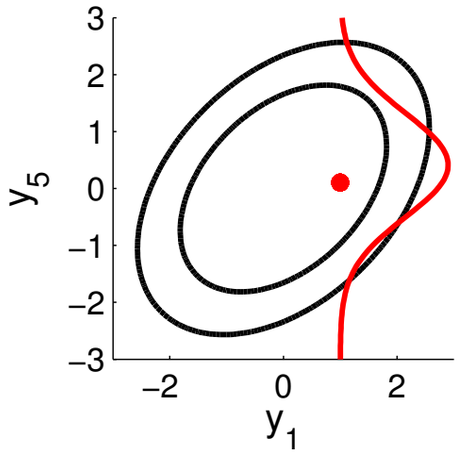
Special covariance matrix - conditioning

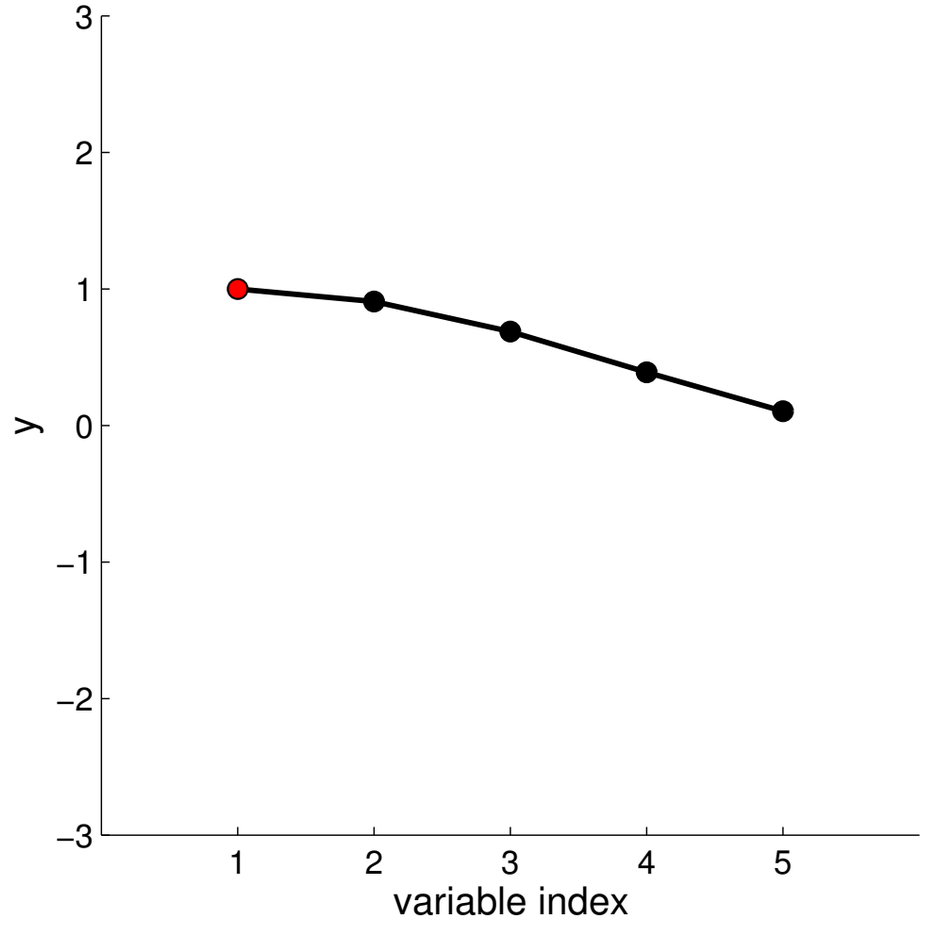




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

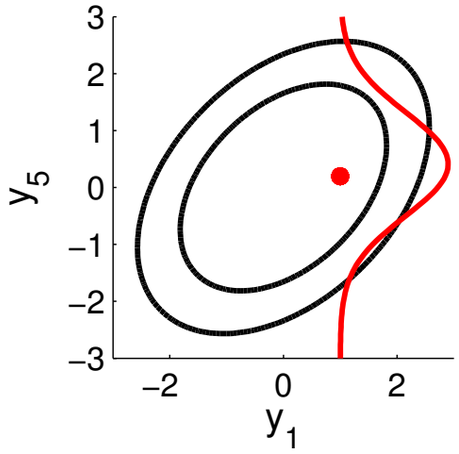
Special covariance matrix - conditioning

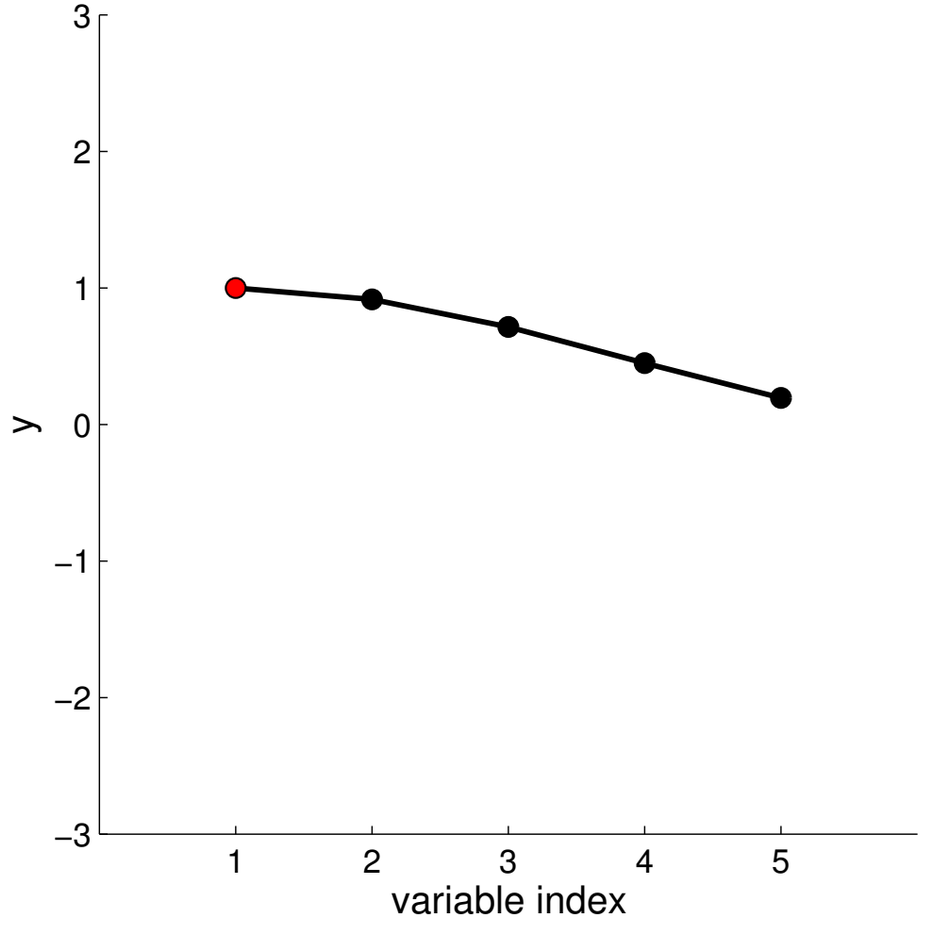




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

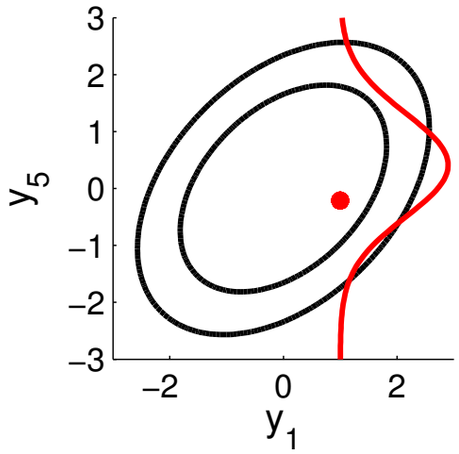
Special covariance matrix - conditioning

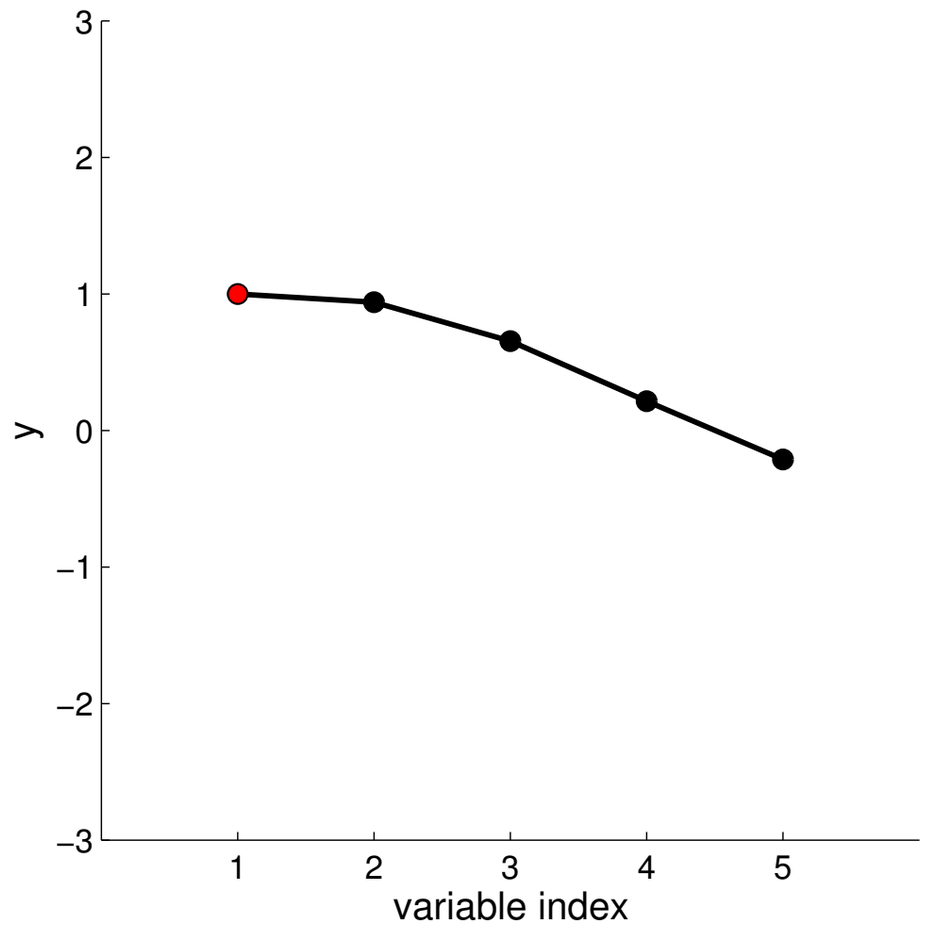




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

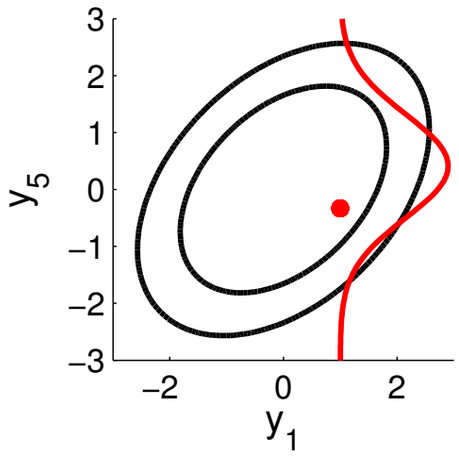
Special covariance matrix - conditioning

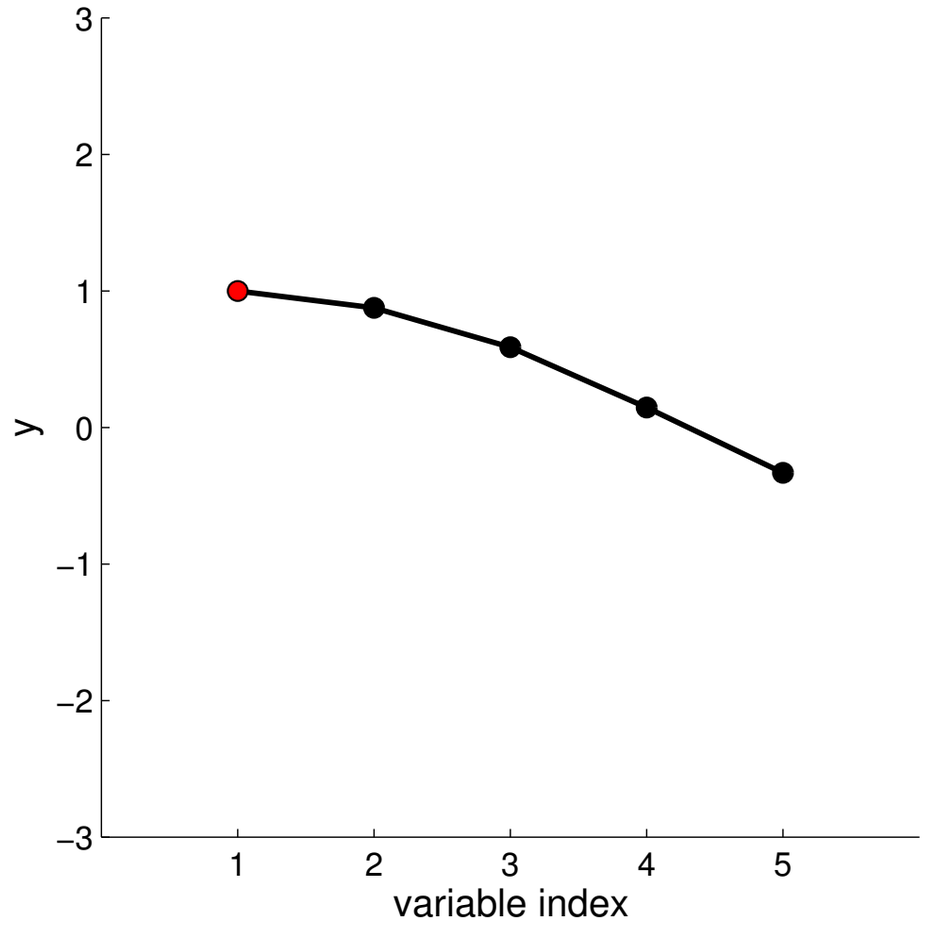




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

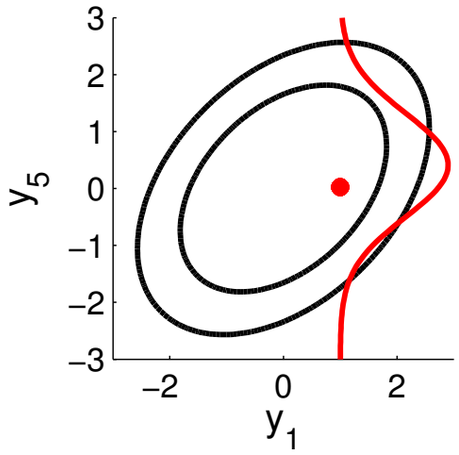
Special covariance matrix - conditioning

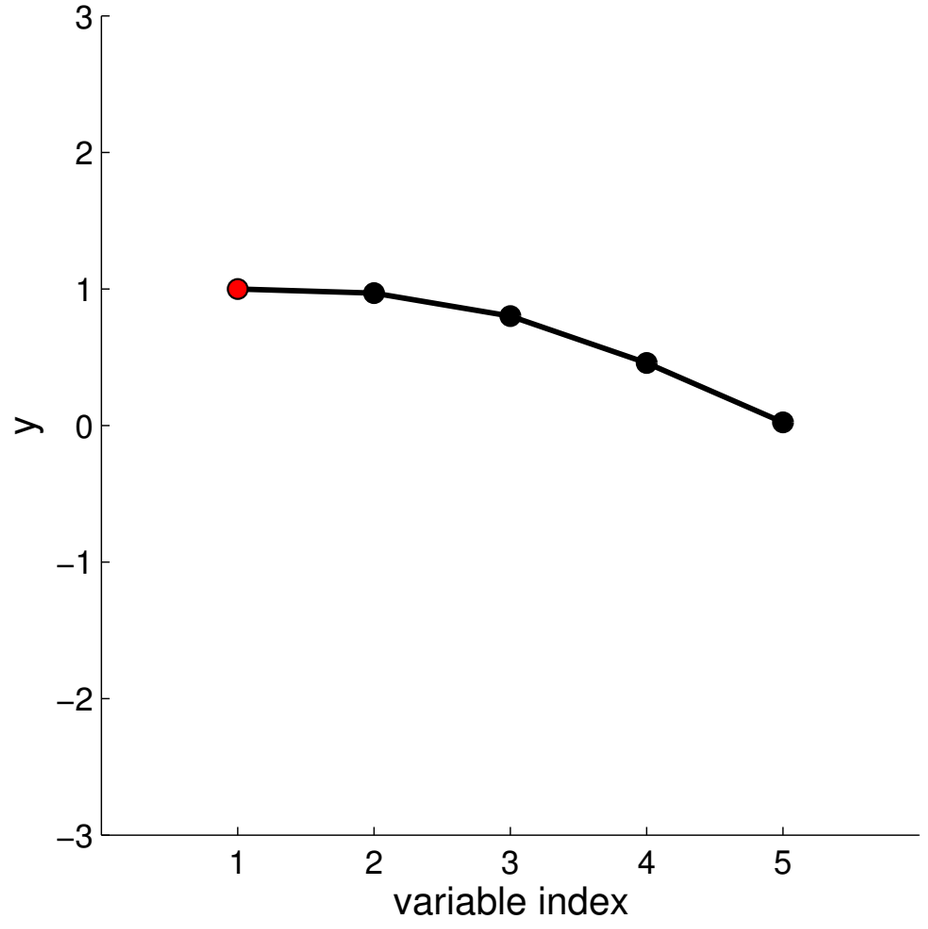




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

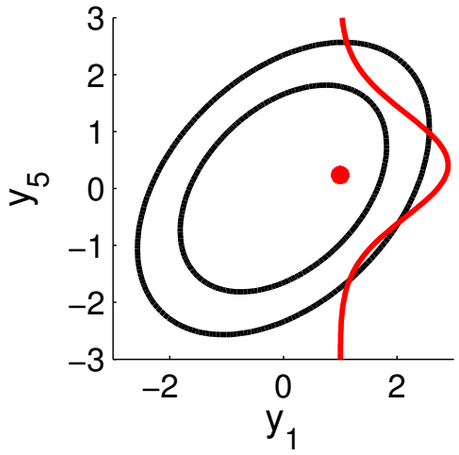
Special covariance matrix - conditioning

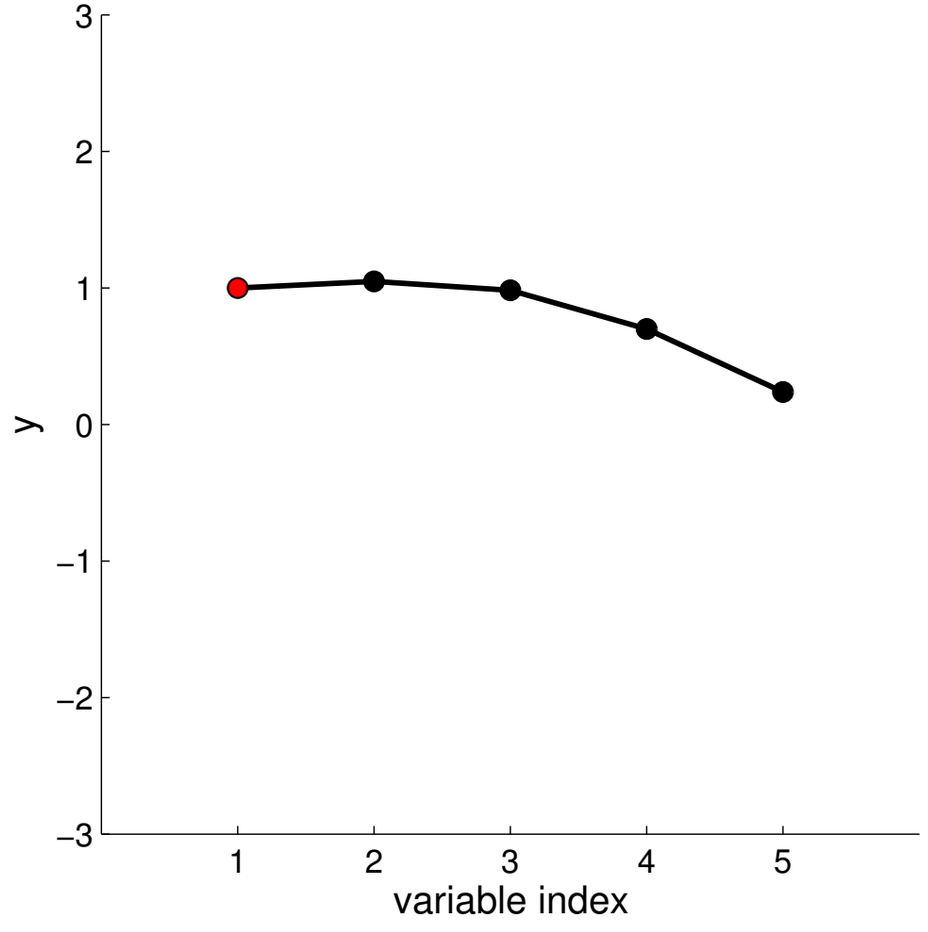




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

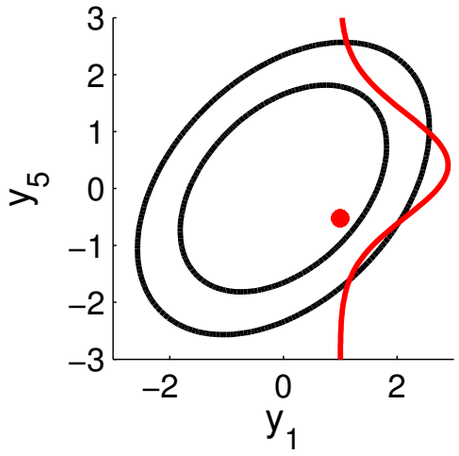
Special covariance matrix - conditioning

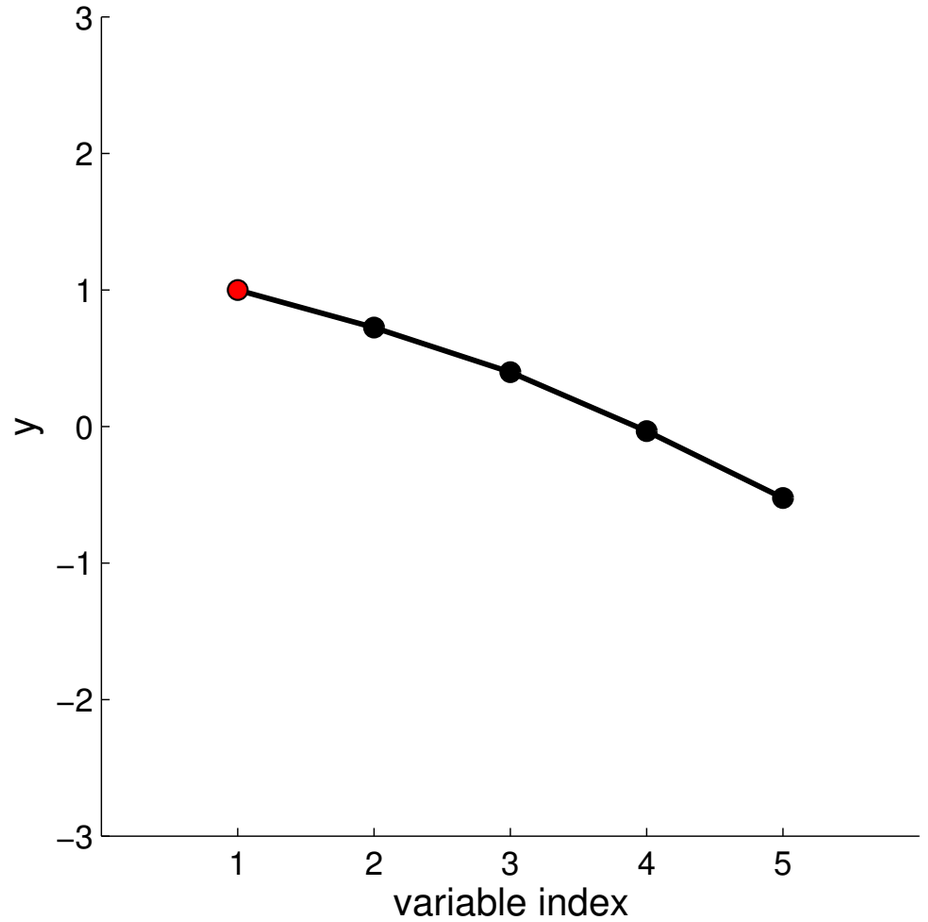




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

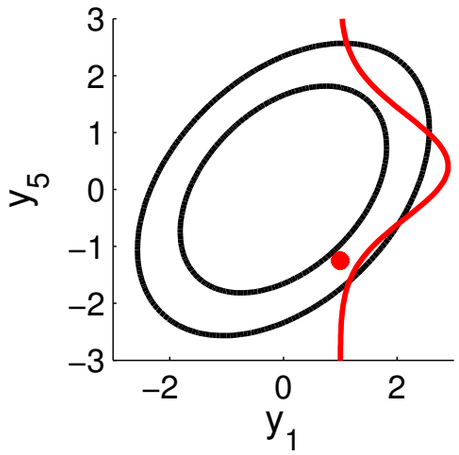
Special covariance matrix - conditioning

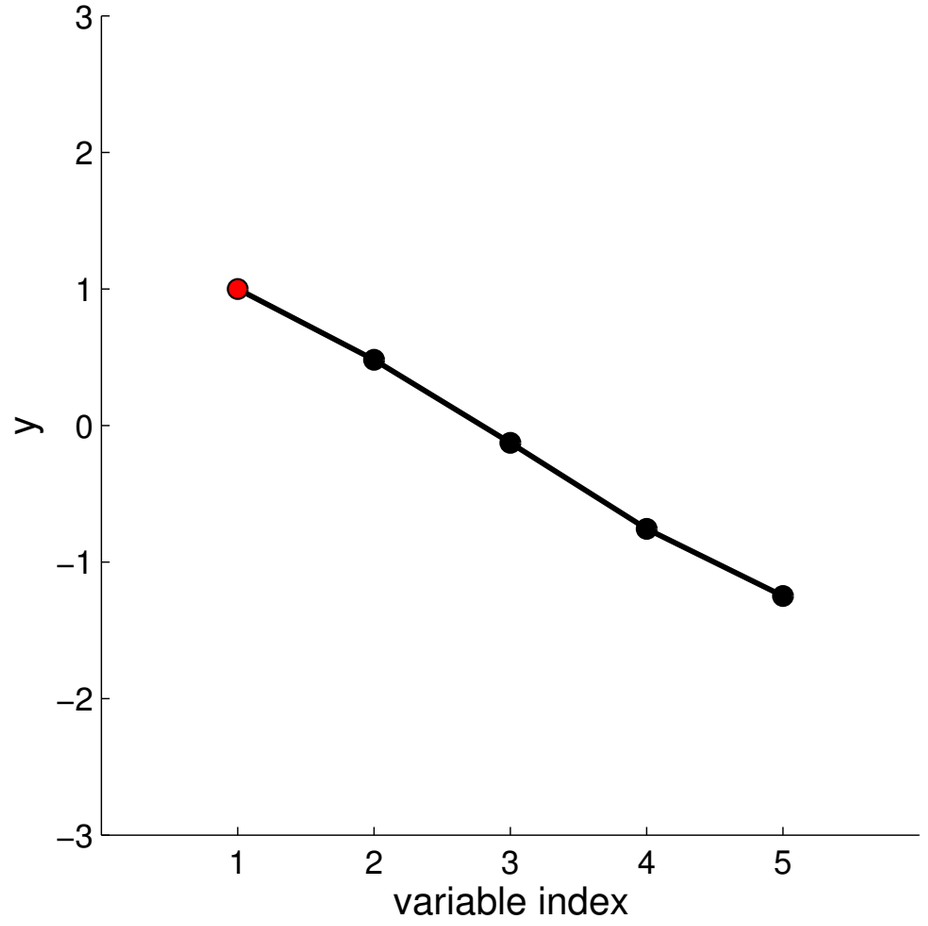




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

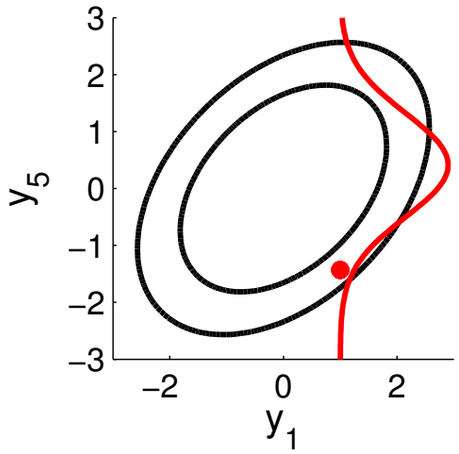
Special covariance matrix - conditioning

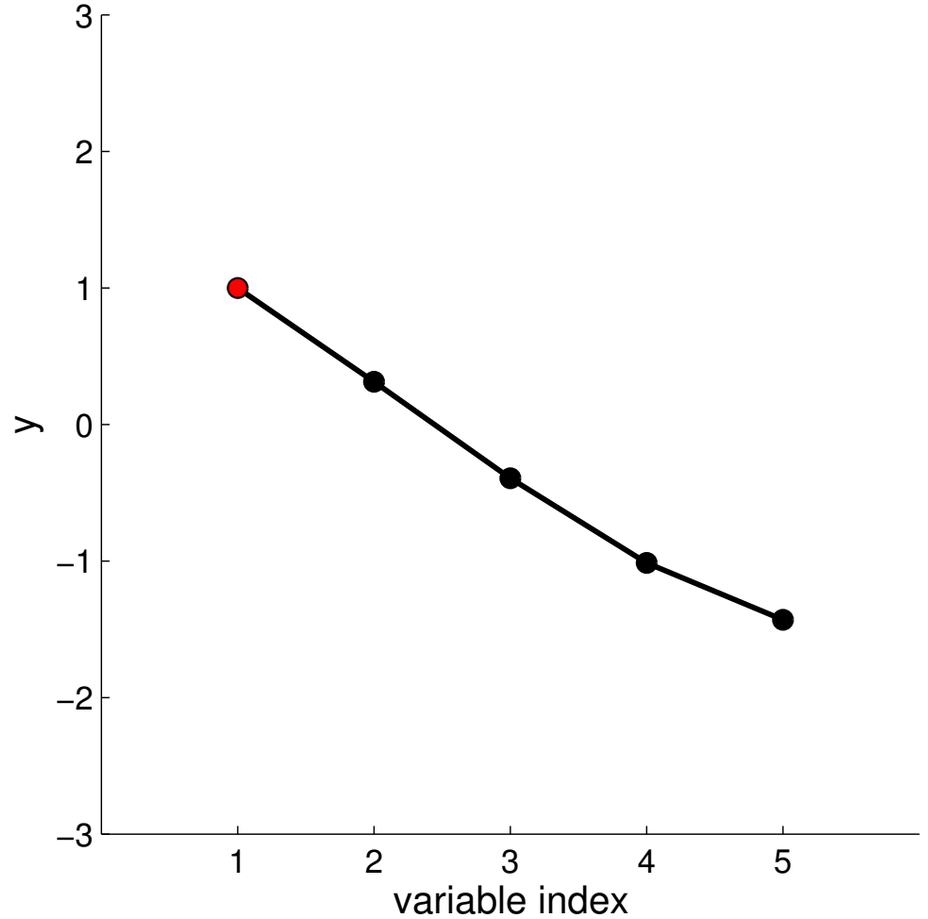




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

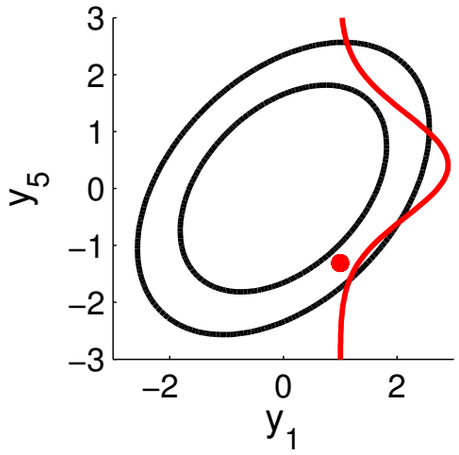
Special covariance matrix - conditioning

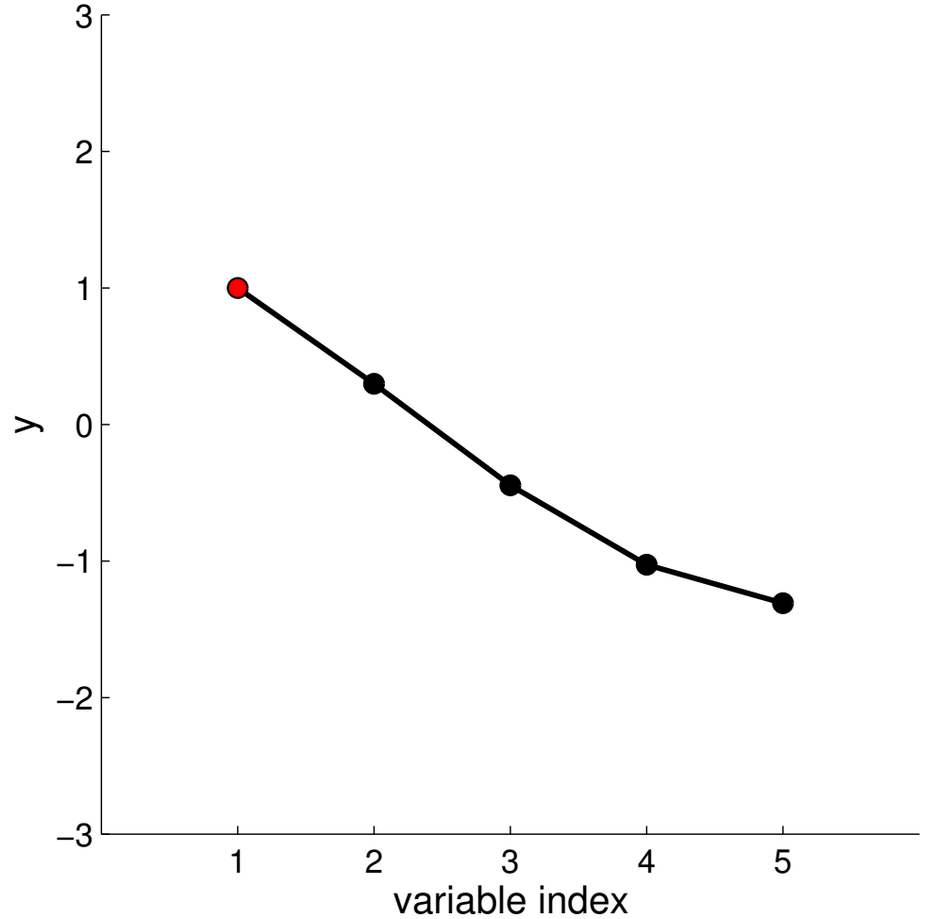




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

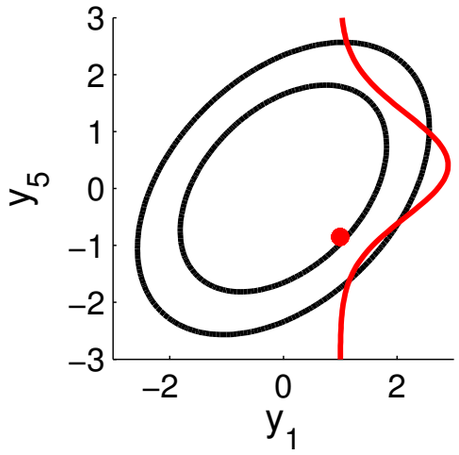
Special covariance matrix - conditioning

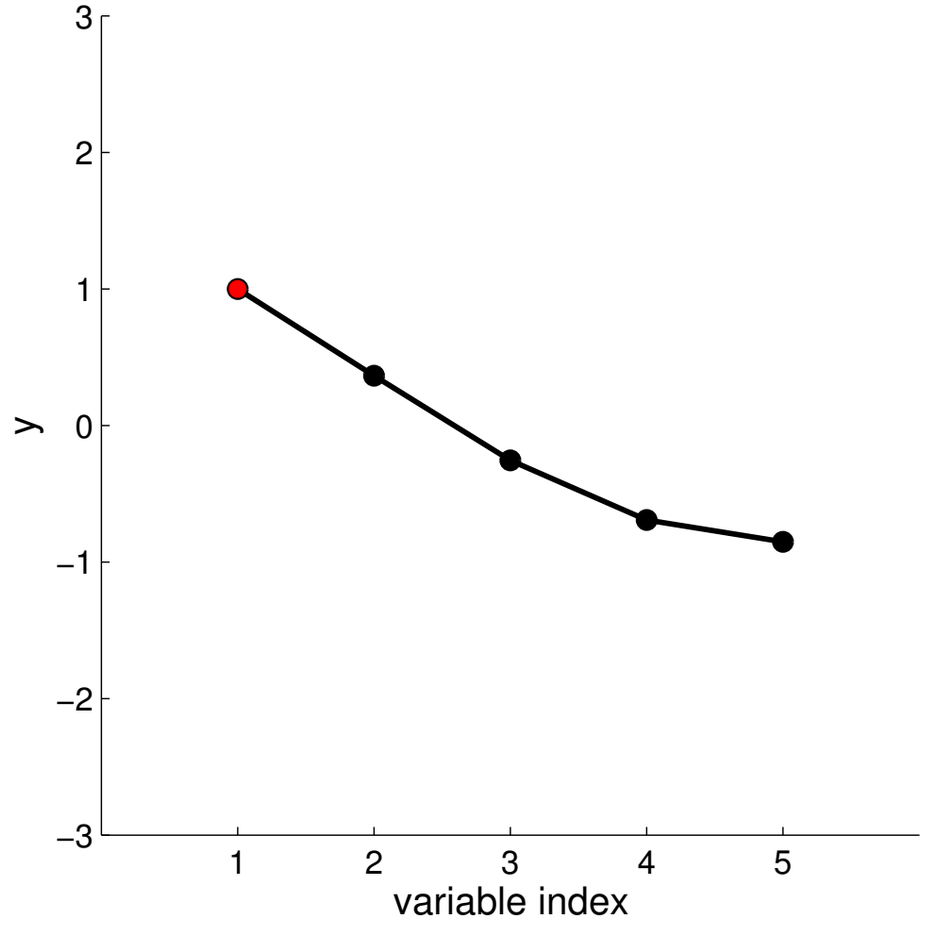




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

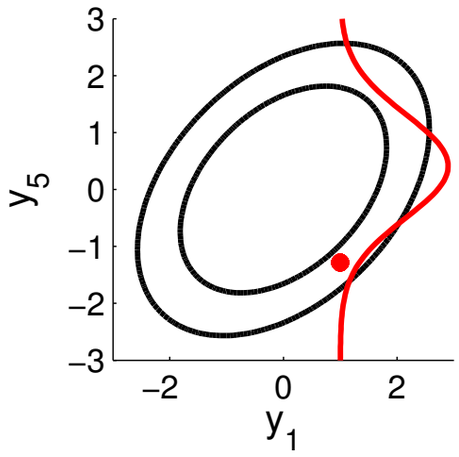
Special covariance matrix - conditioning

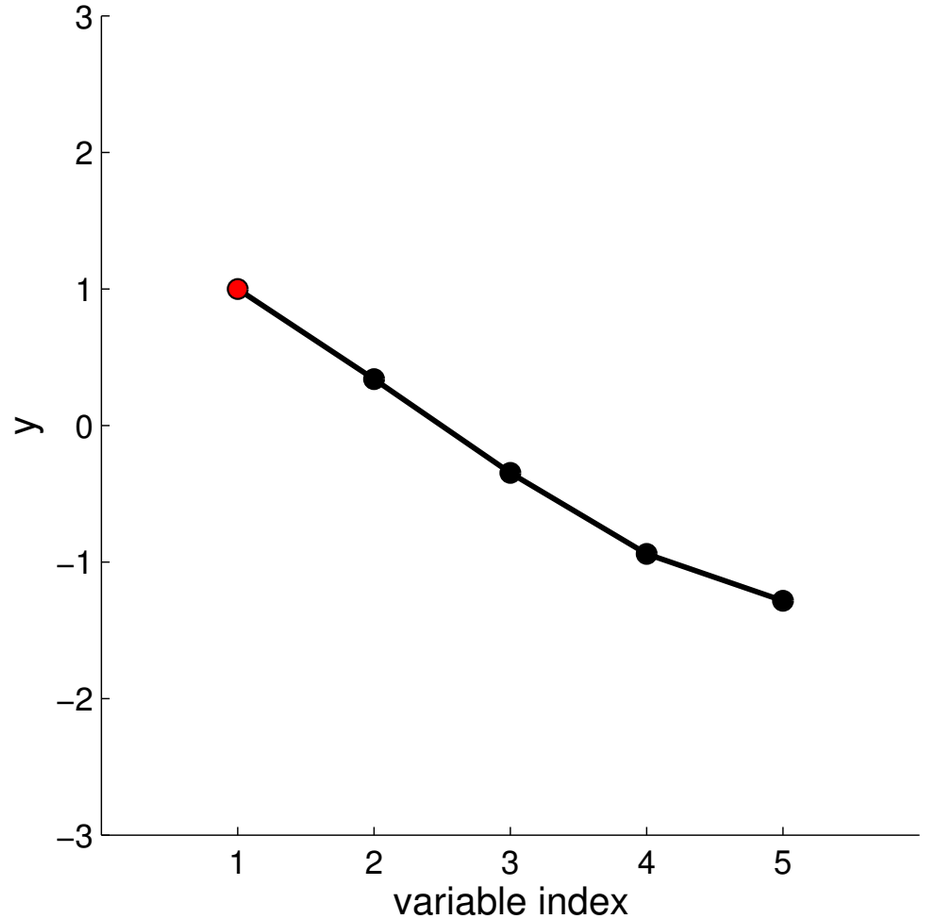




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

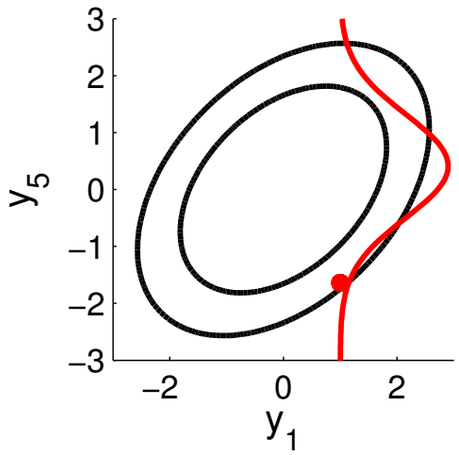
Special covariance matrix - conditioning

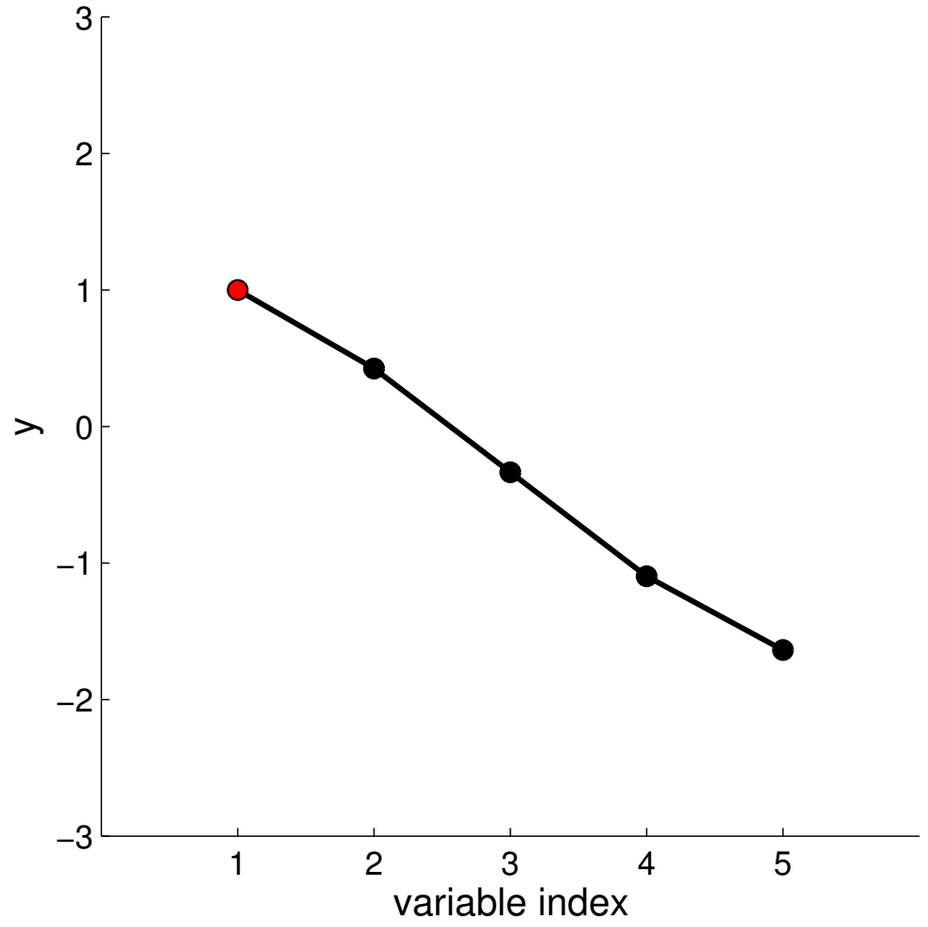




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

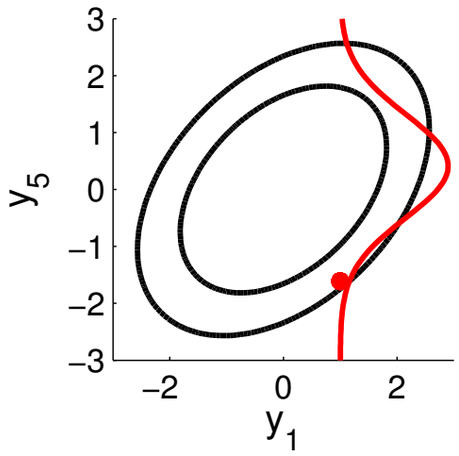
Special covariance matrix - conditioning

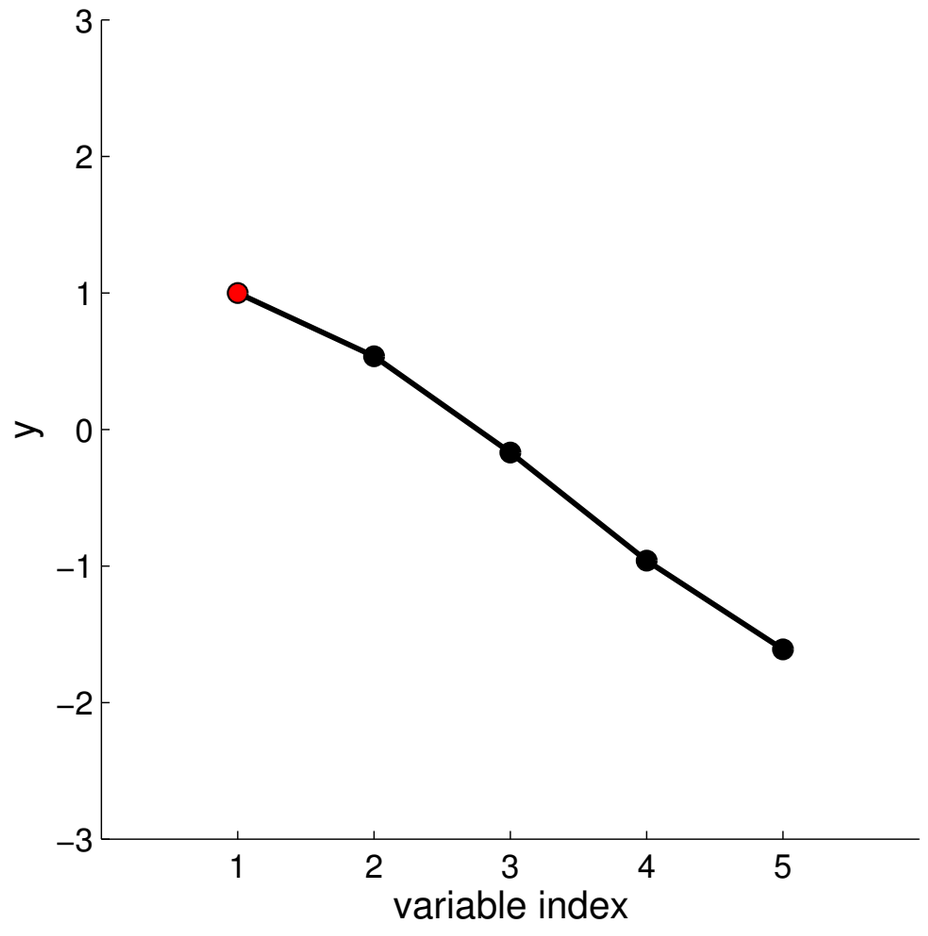




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

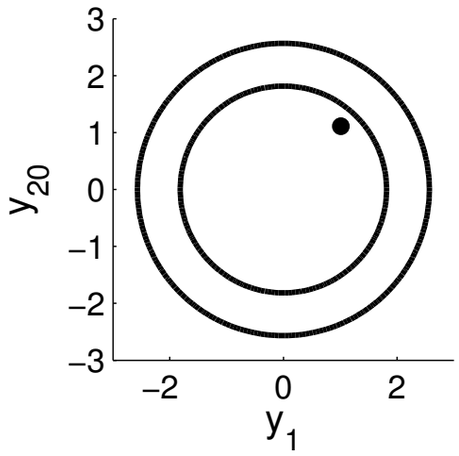
Special covariance matrix - conditioning

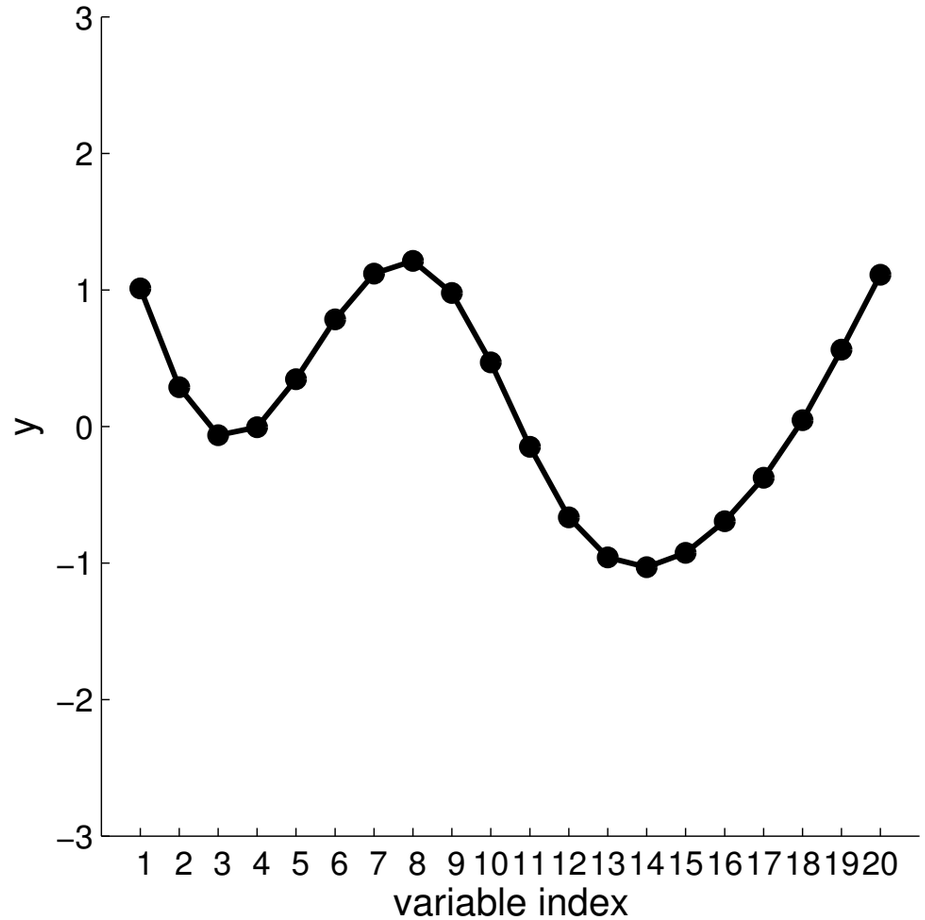




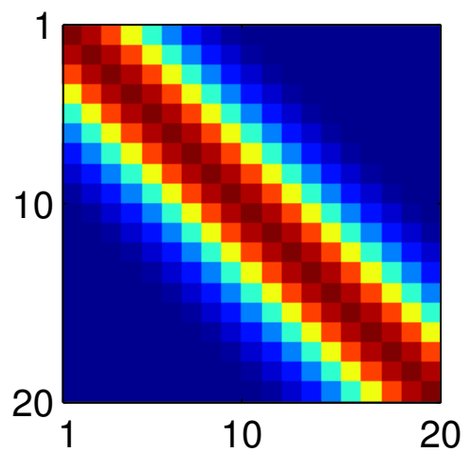
$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

Special covariance matrix



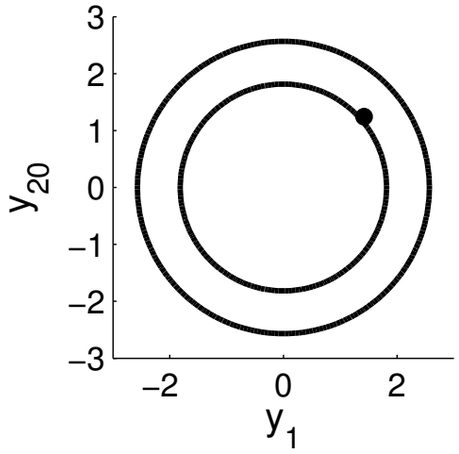


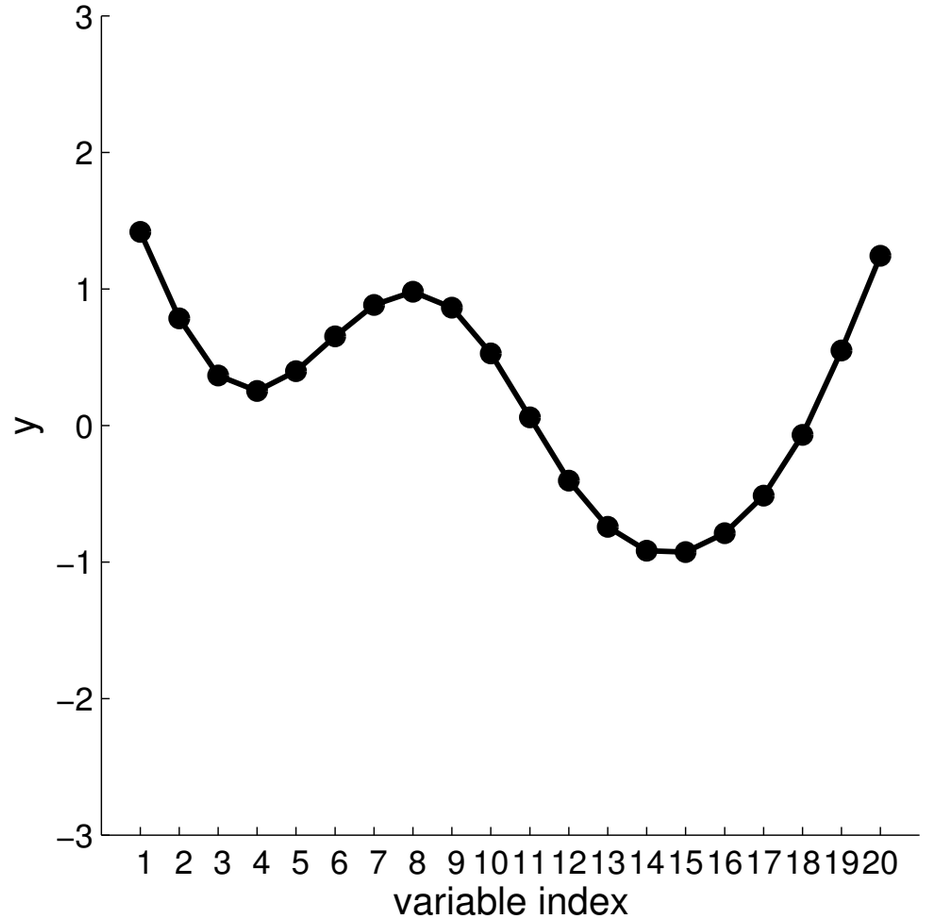
$\Sigma =$



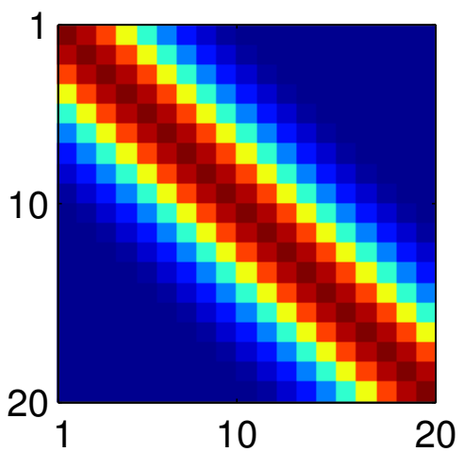
red means 1, blue means 0

Special covariance matrix

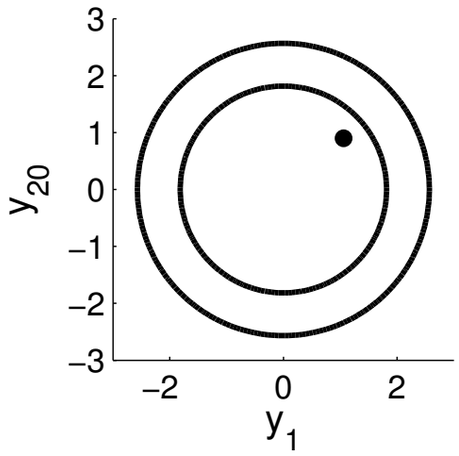


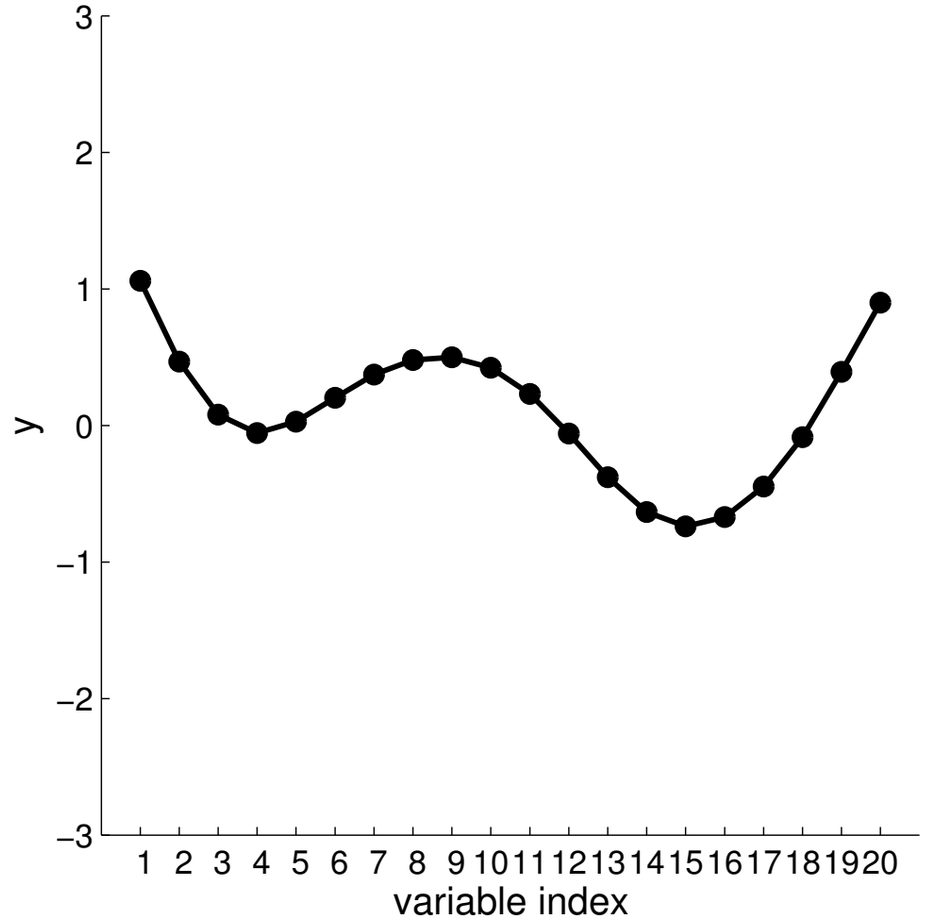


$\Sigma =$

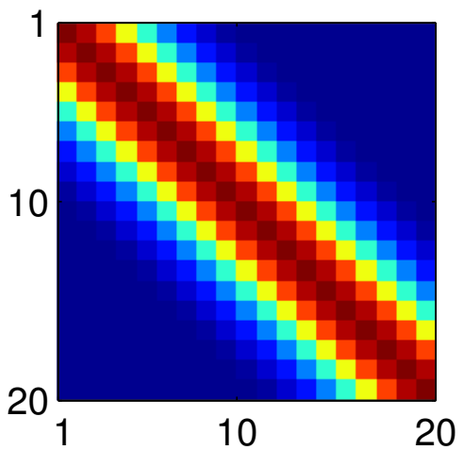


Special covariance matrix

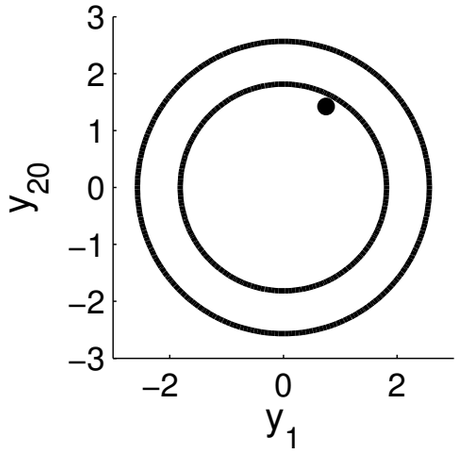


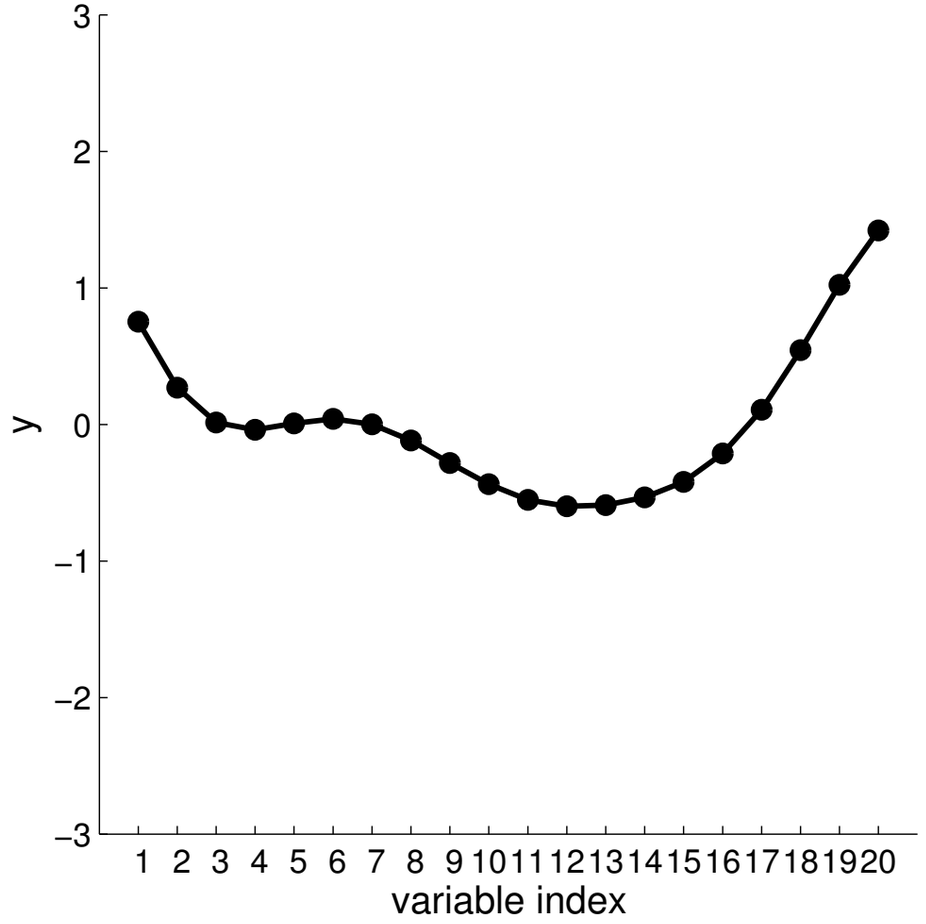


$\Sigma =$

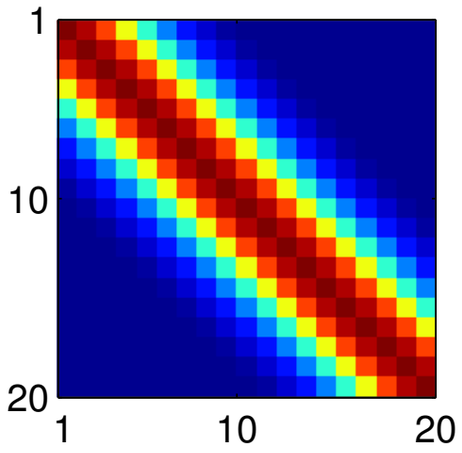


Special covariance matrix

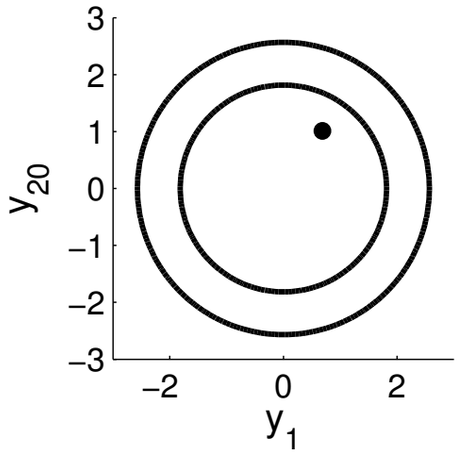


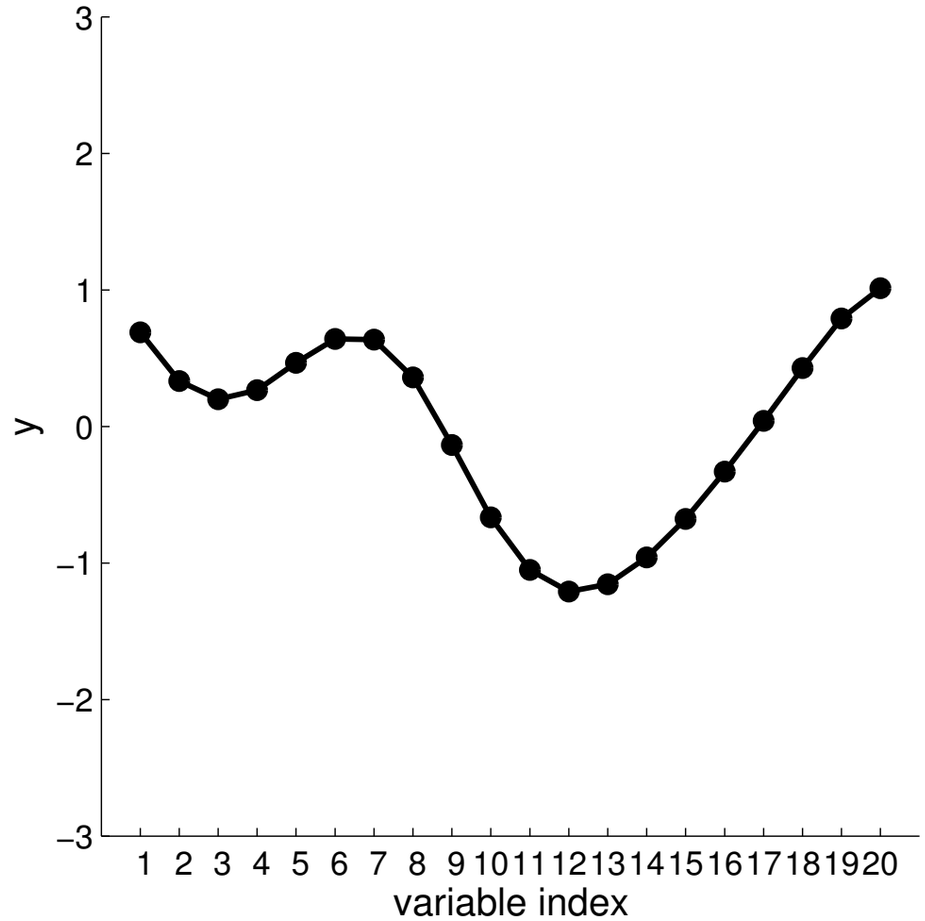


$\Sigma =$

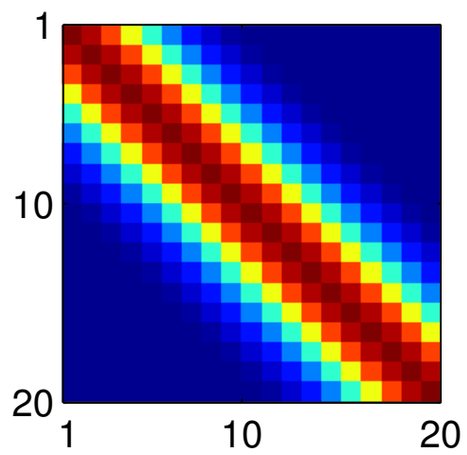


Special covariance matrix

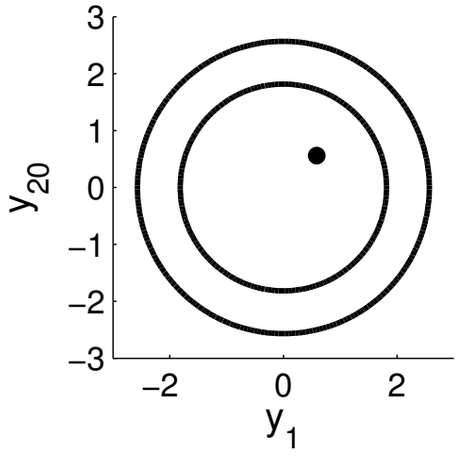


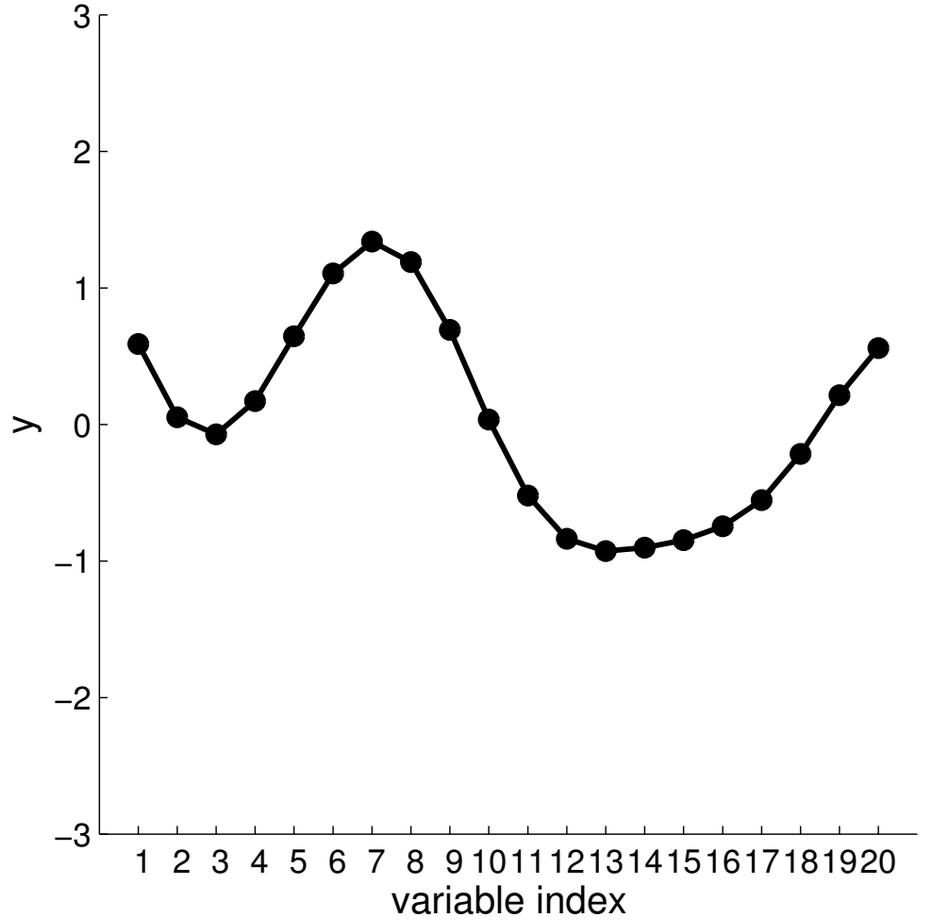


$\Sigma =$

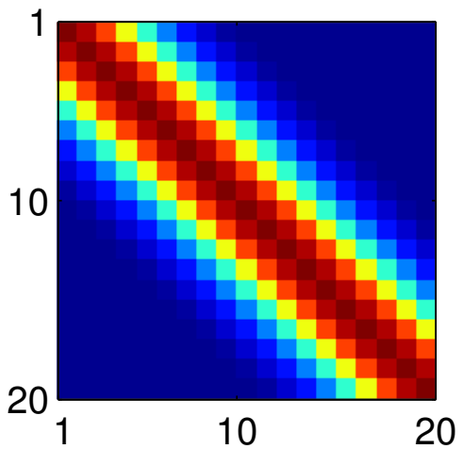


Special covariance matrix

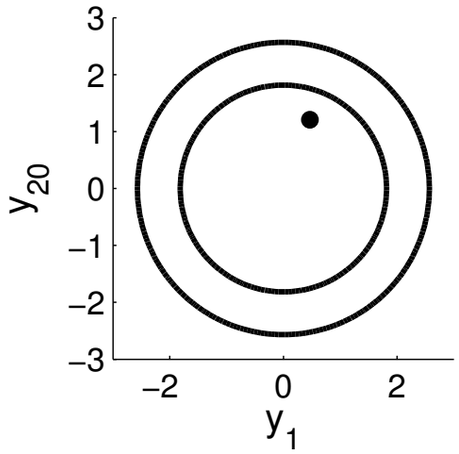


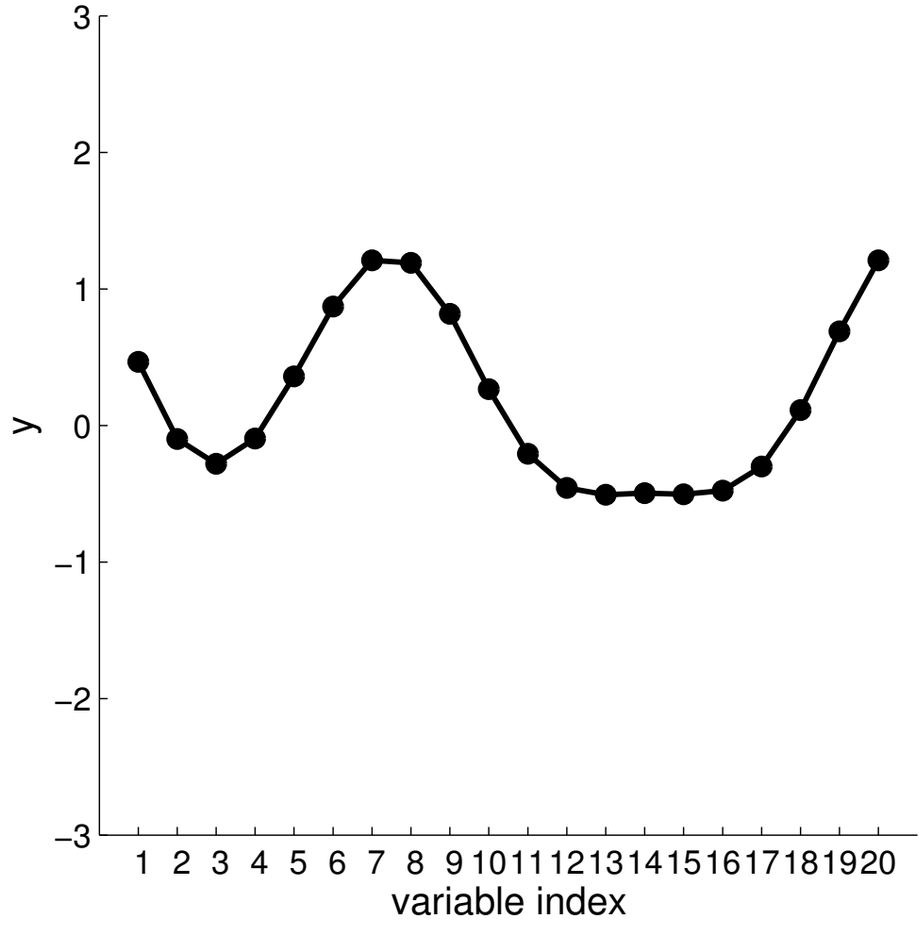


$\Sigma =$

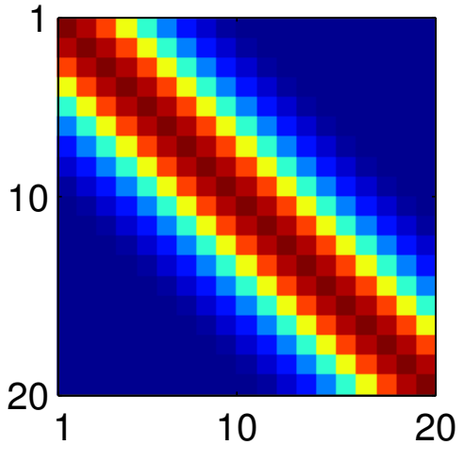


Special covariance matrix

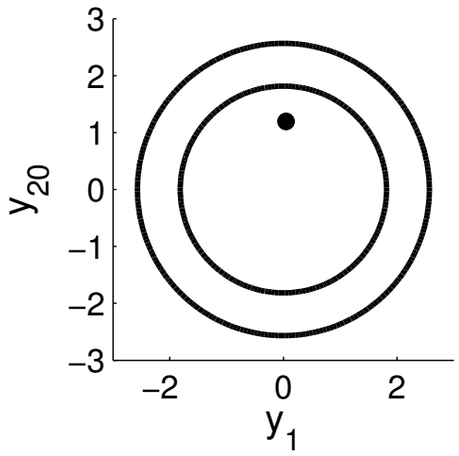


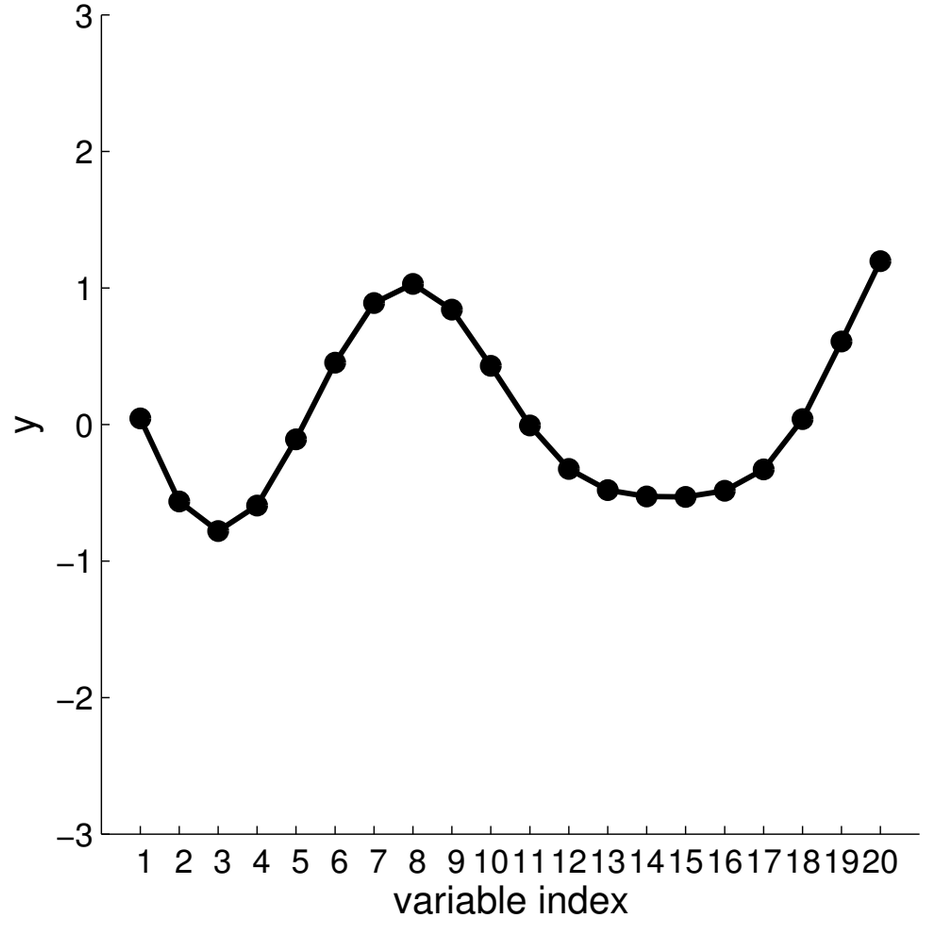


$\Sigma =$

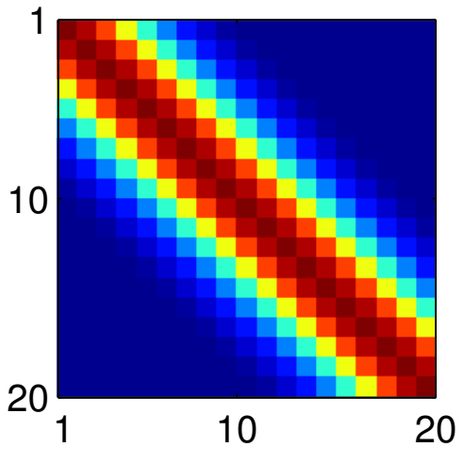


Special covariance matrix

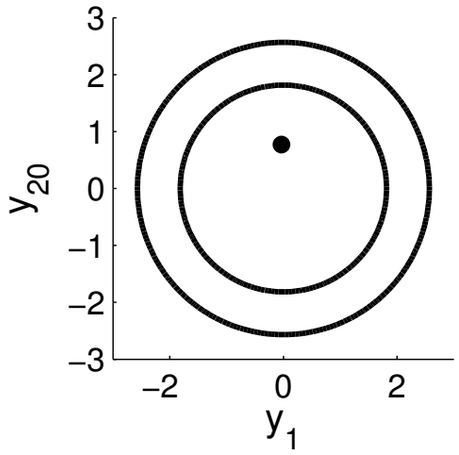


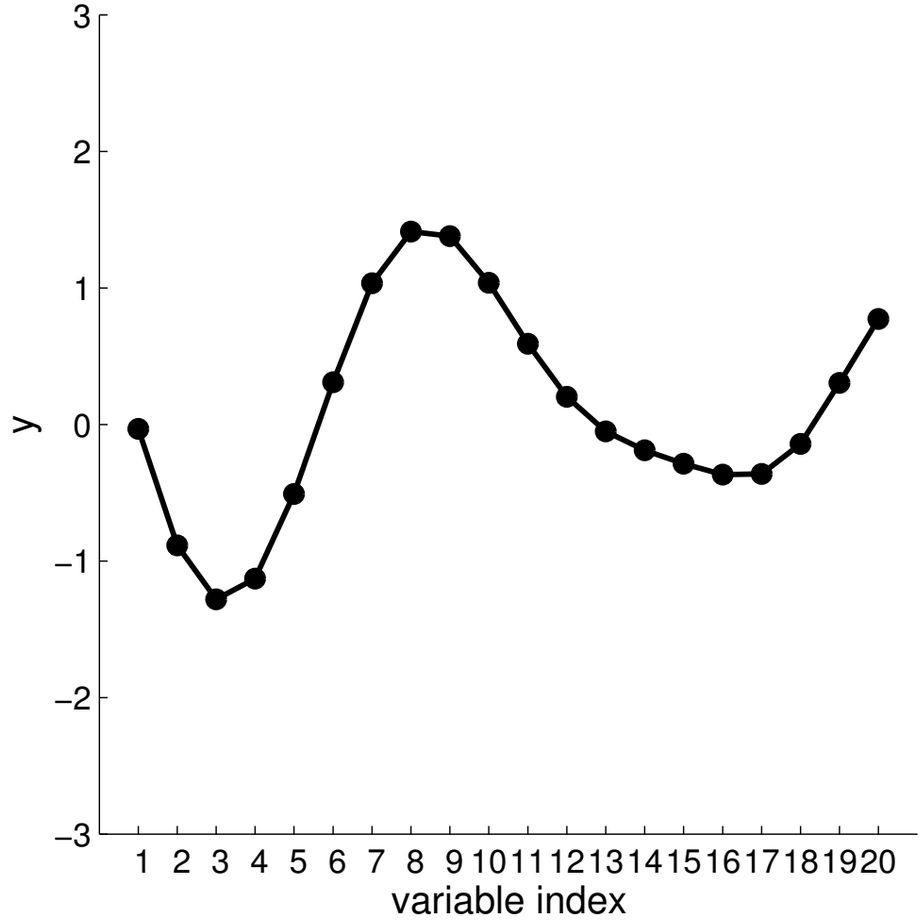


$\Sigma =$

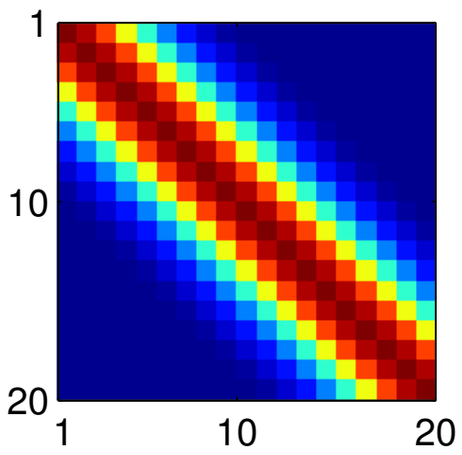


Special covariance matrix

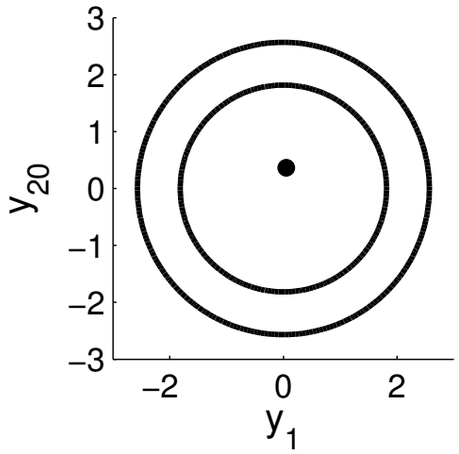


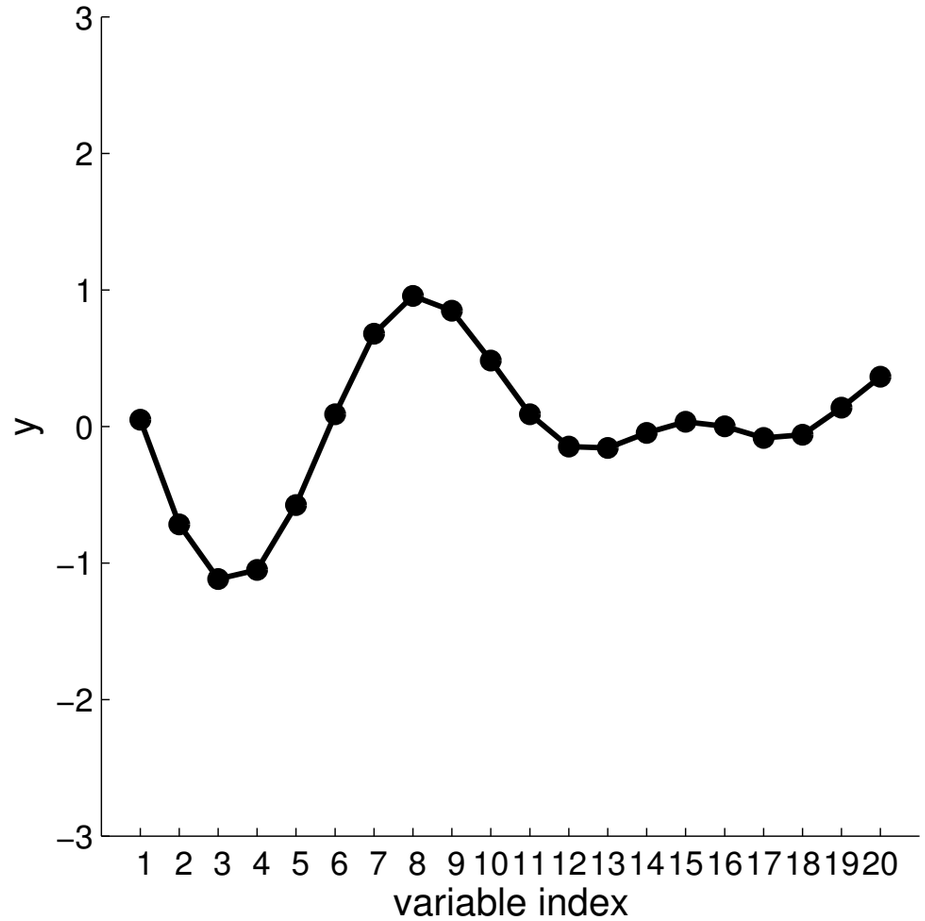


$\Sigma =$

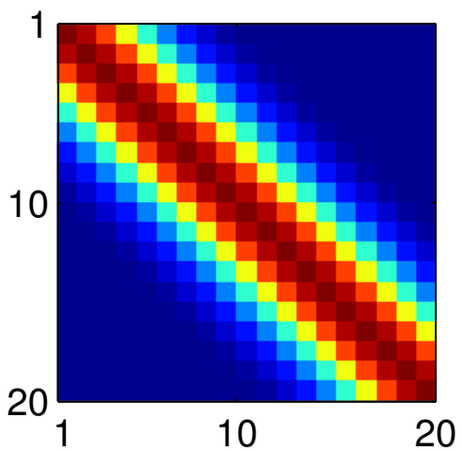


Special covariance matrix

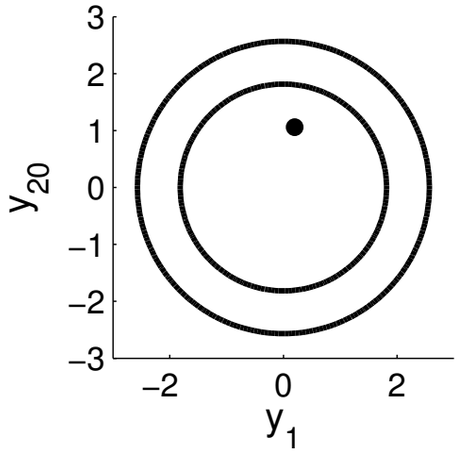


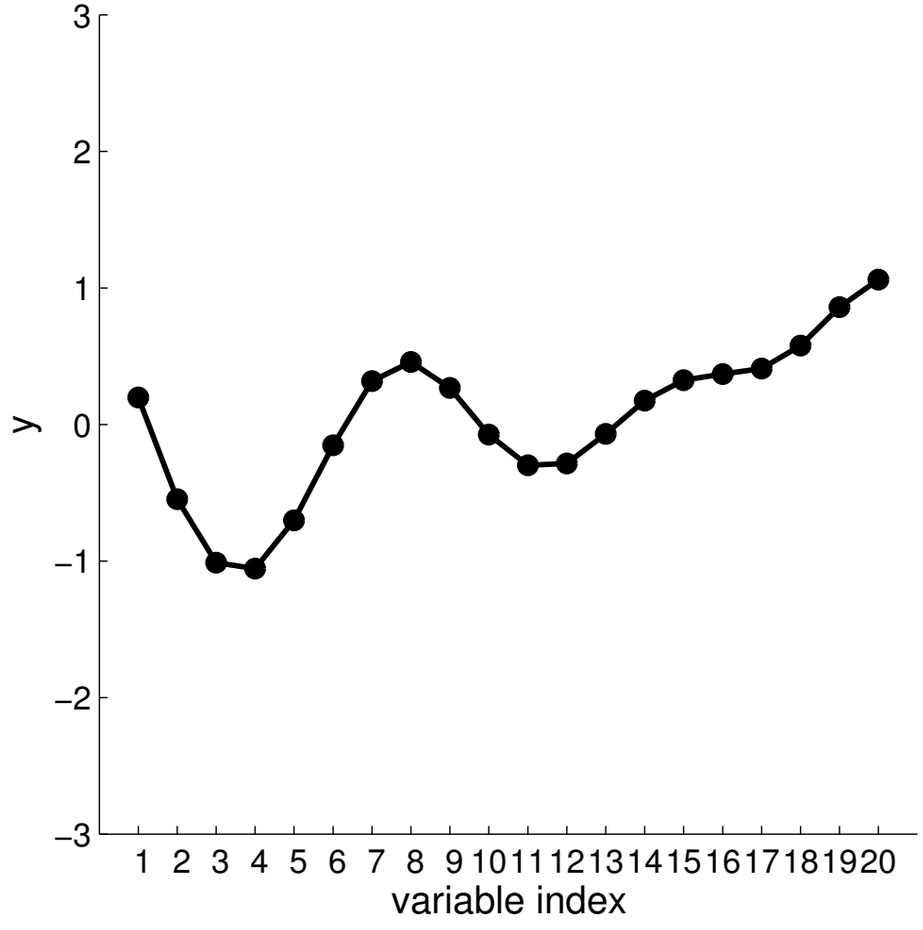


$\Sigma =$

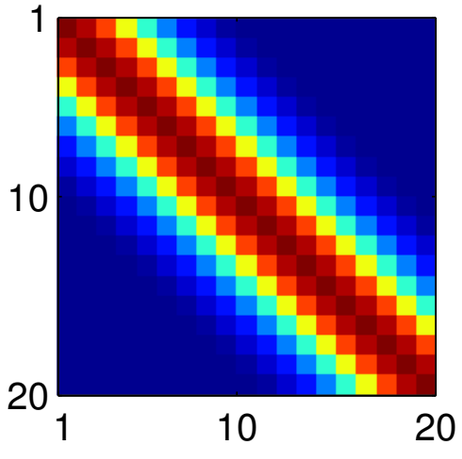


Special covariance matrix

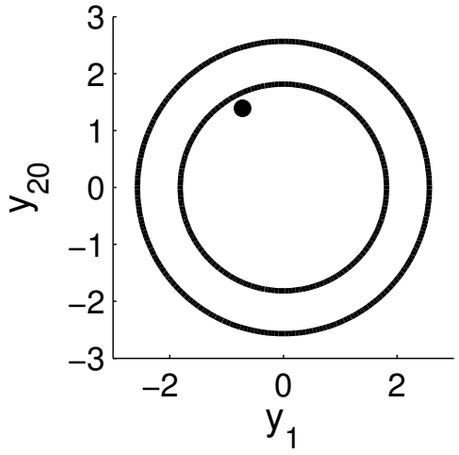


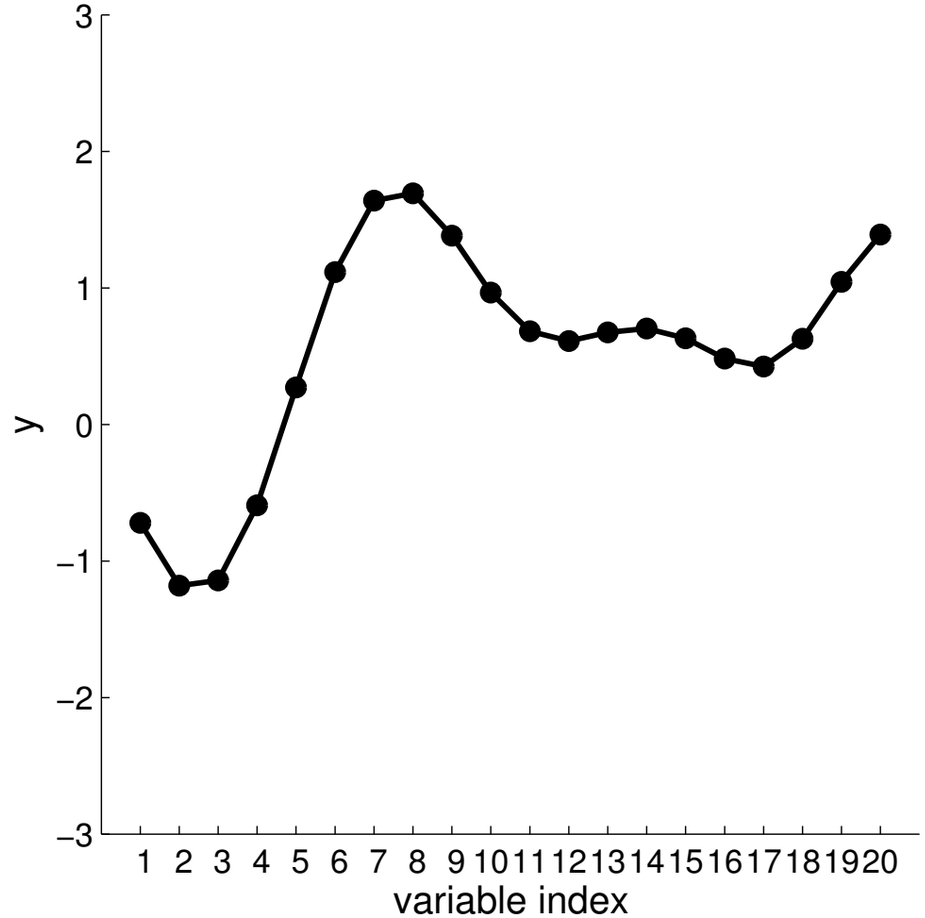


$\Sigma =$

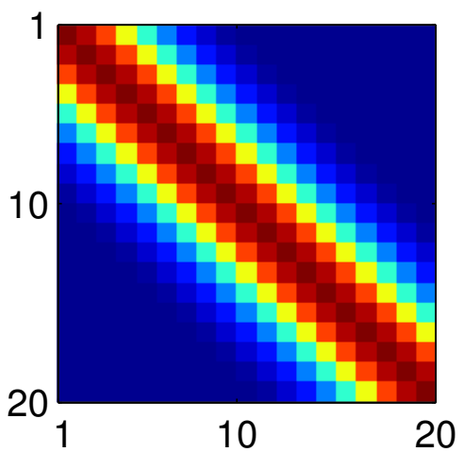


Special covariance matrix

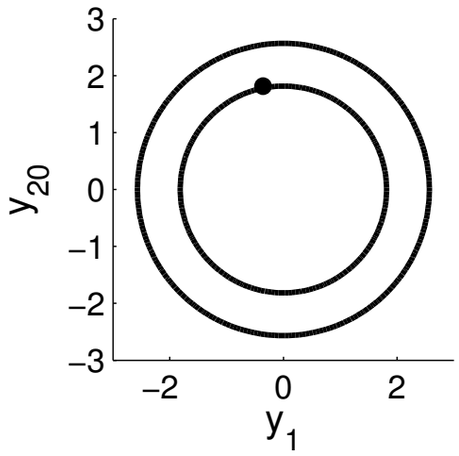


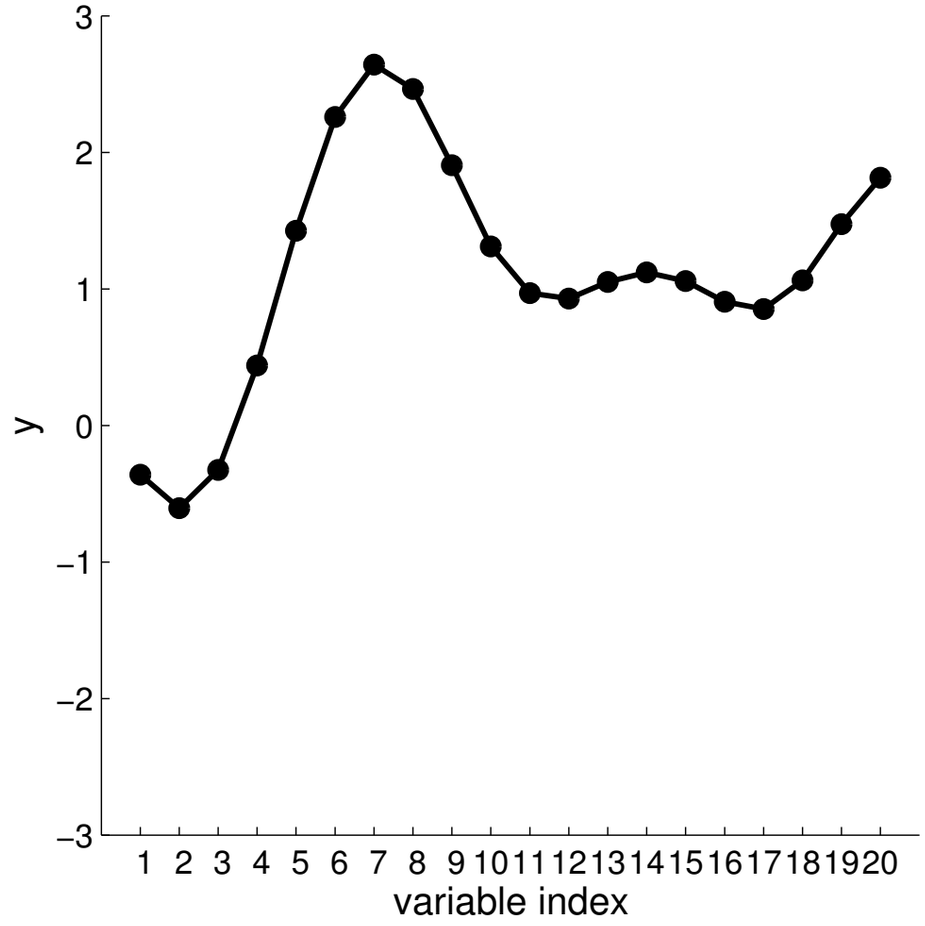


$\Sigma =$

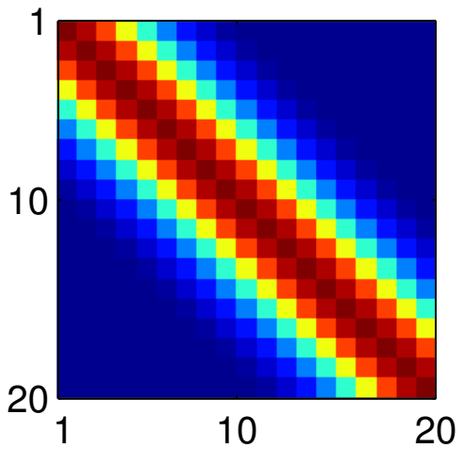


Special covariance matrix

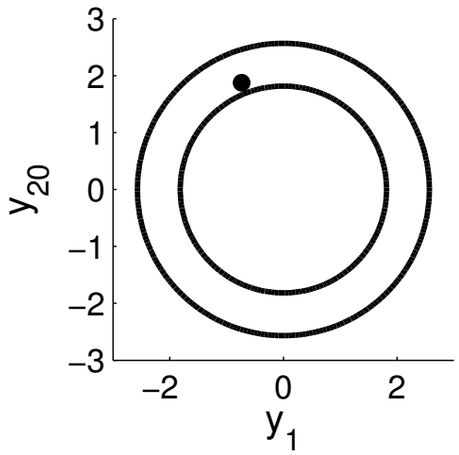


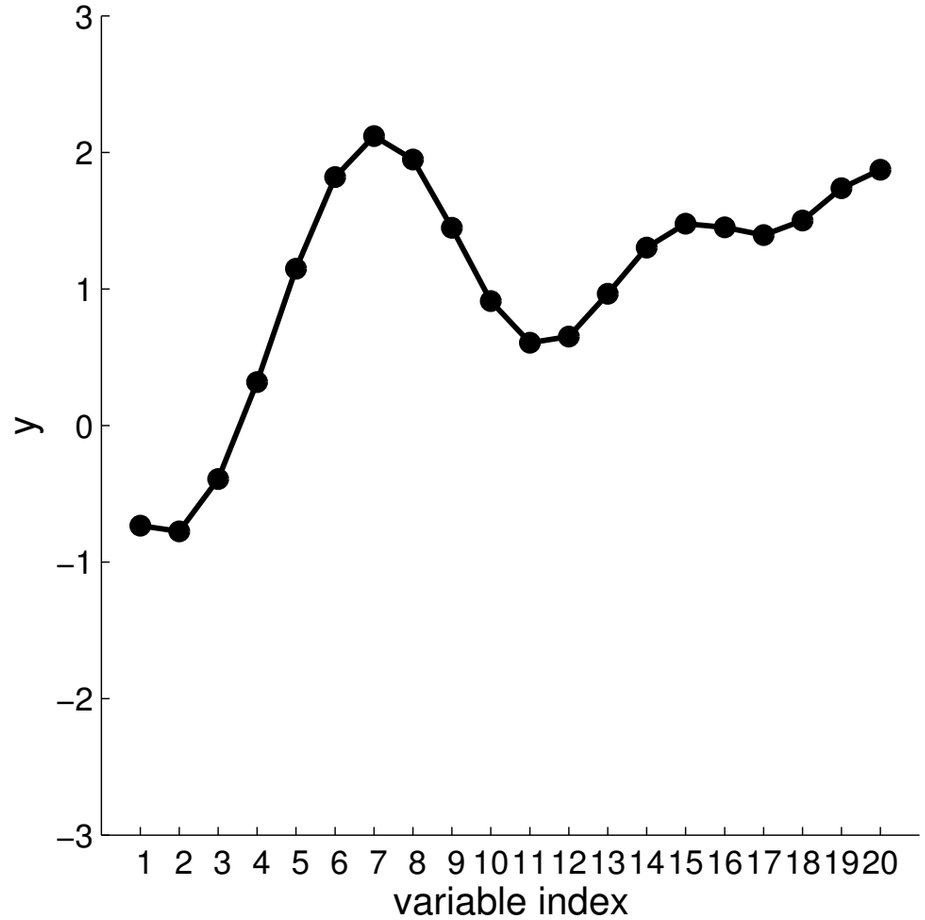


$\Sigma =$

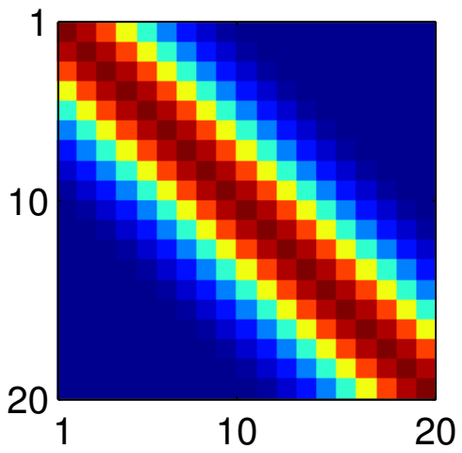


Special covariance matrix

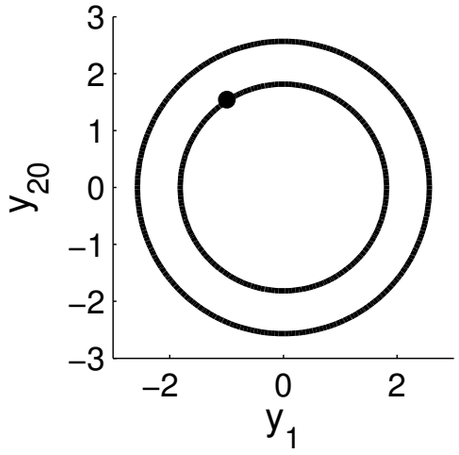


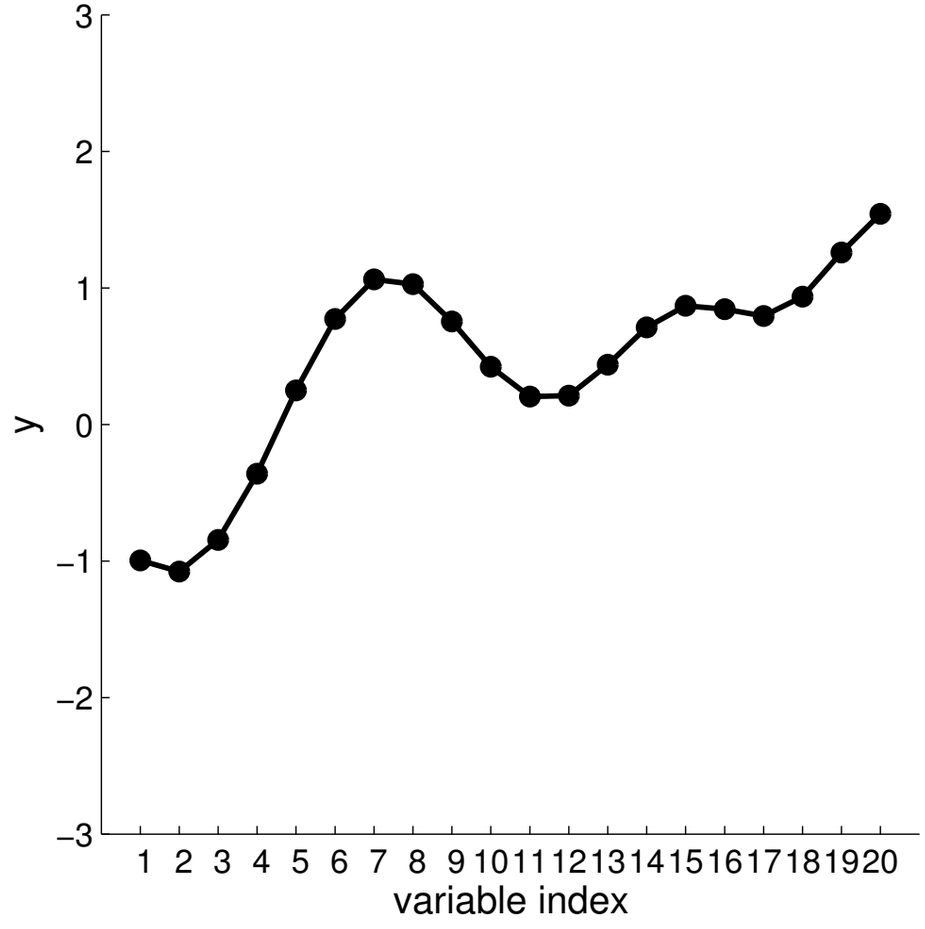


$\Sigma =$

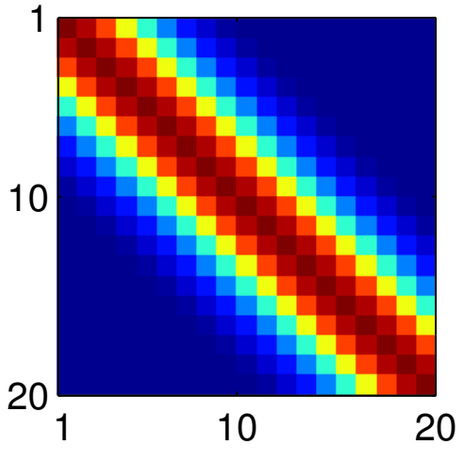


Special covariance matrix

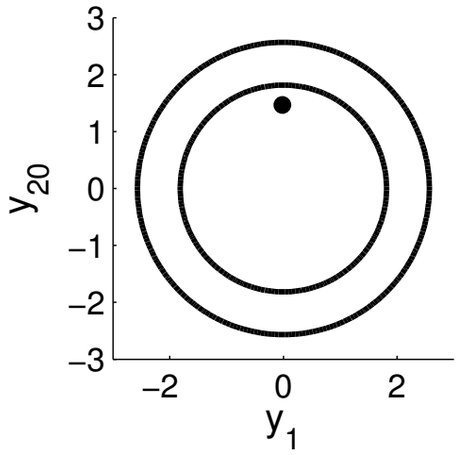


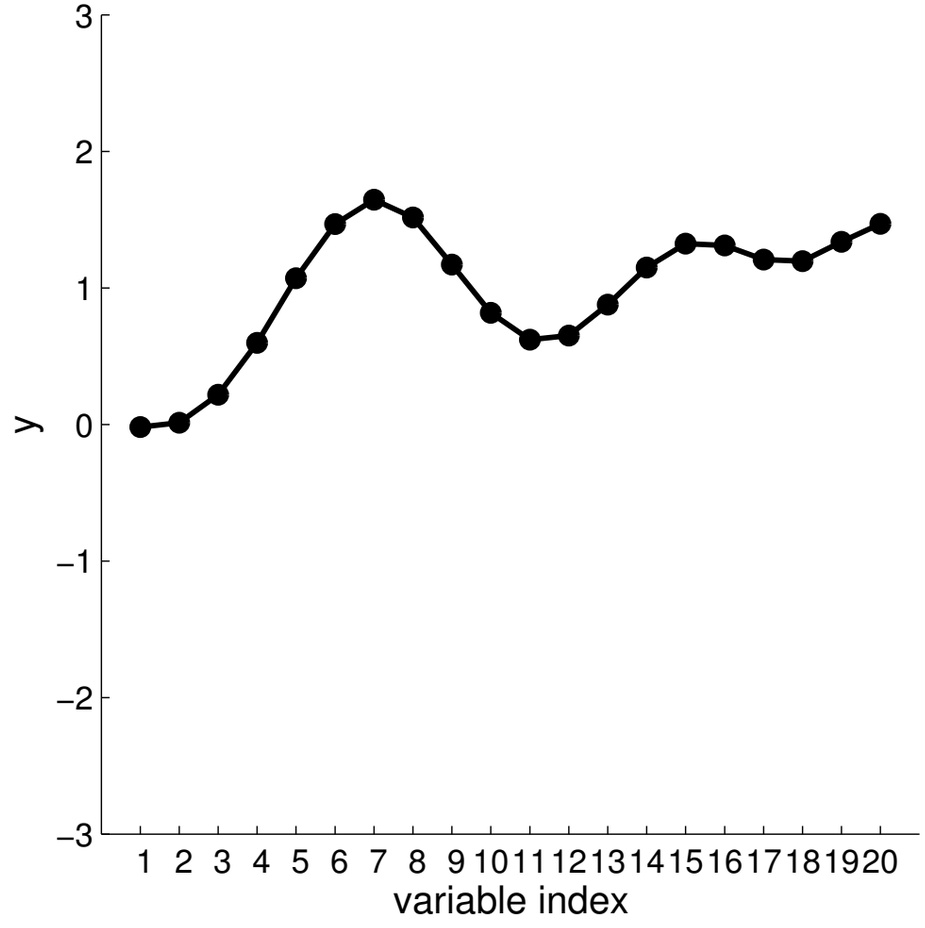


$\Sigma =$

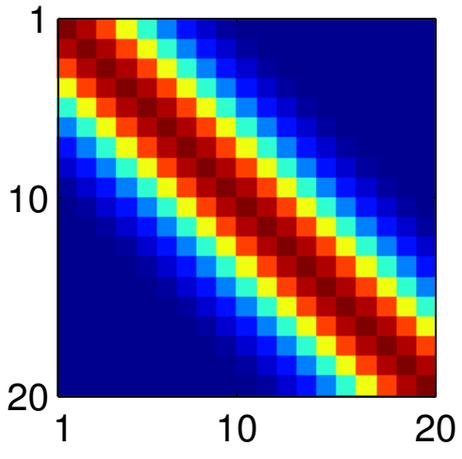


Special covariance matrix

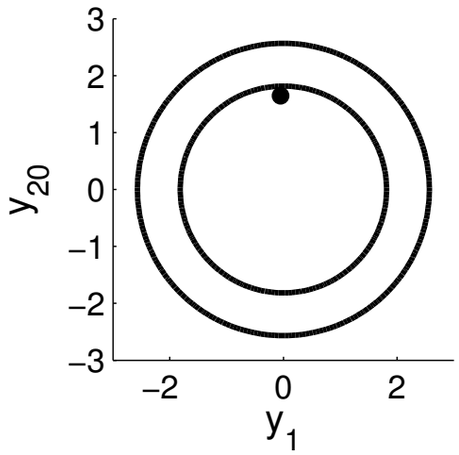


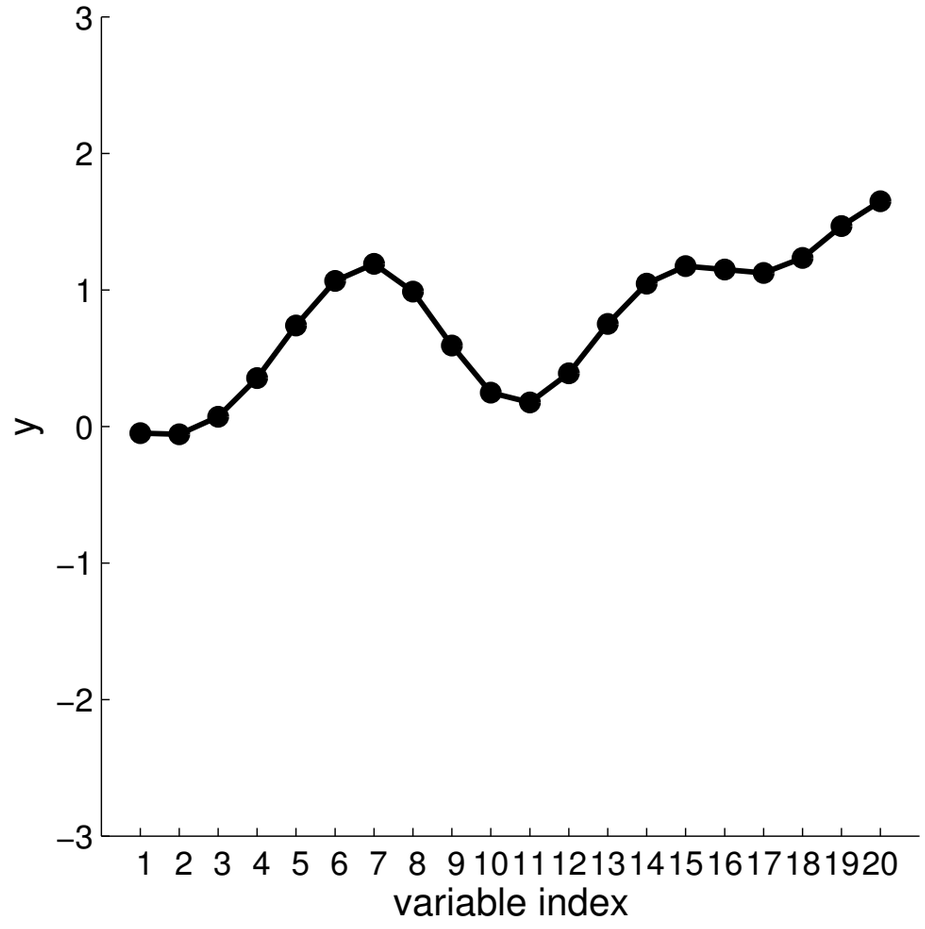


$\Sigma =$

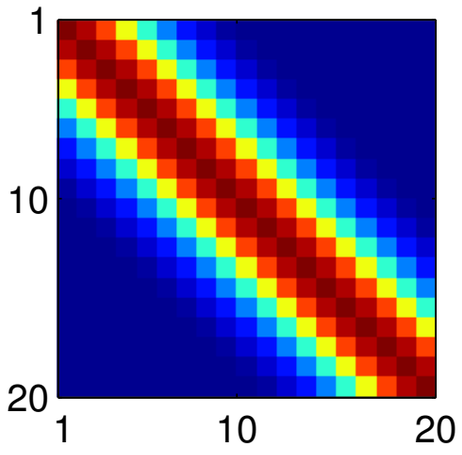


Special covariance matrix

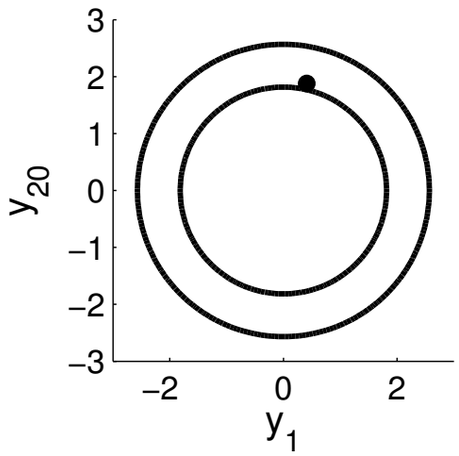


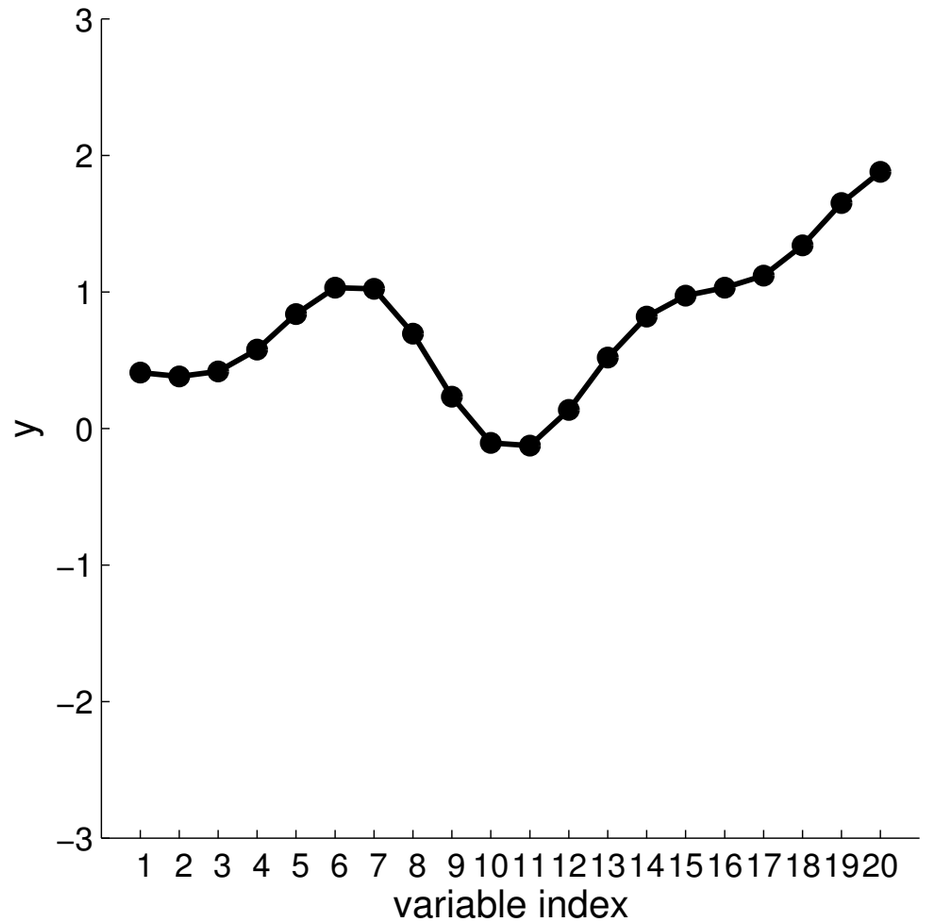


$\Sigma =$

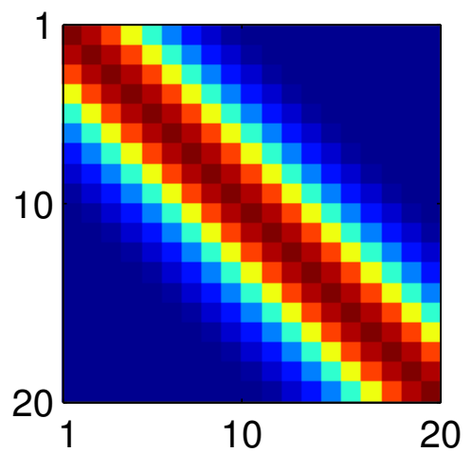


Special covariance matrix

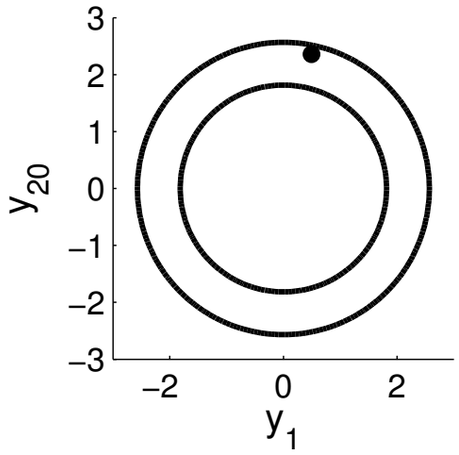


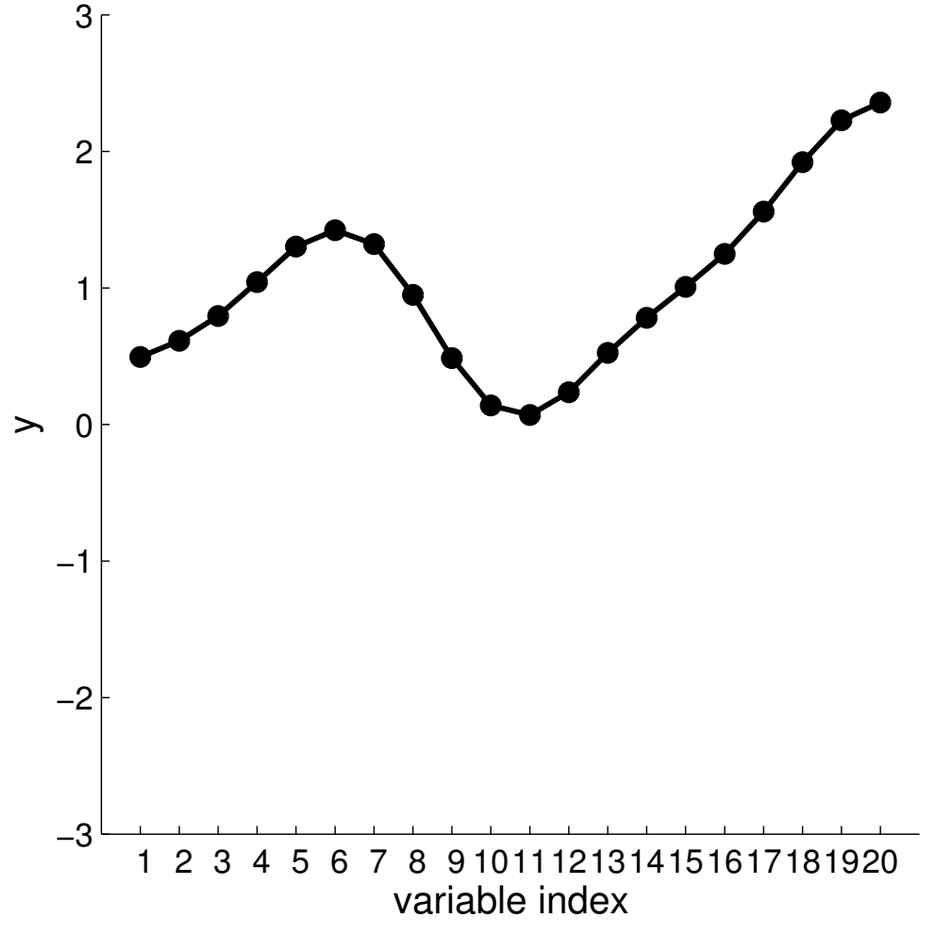


$\Sigma =$

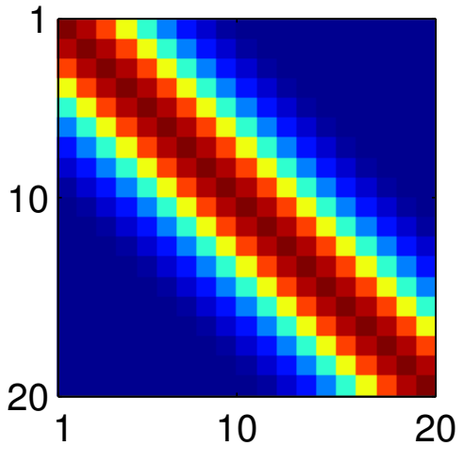


Special covariance matrix



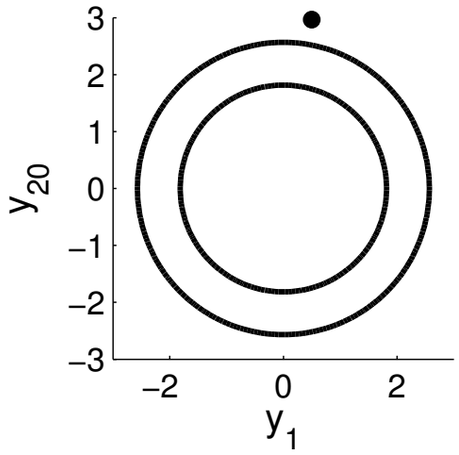


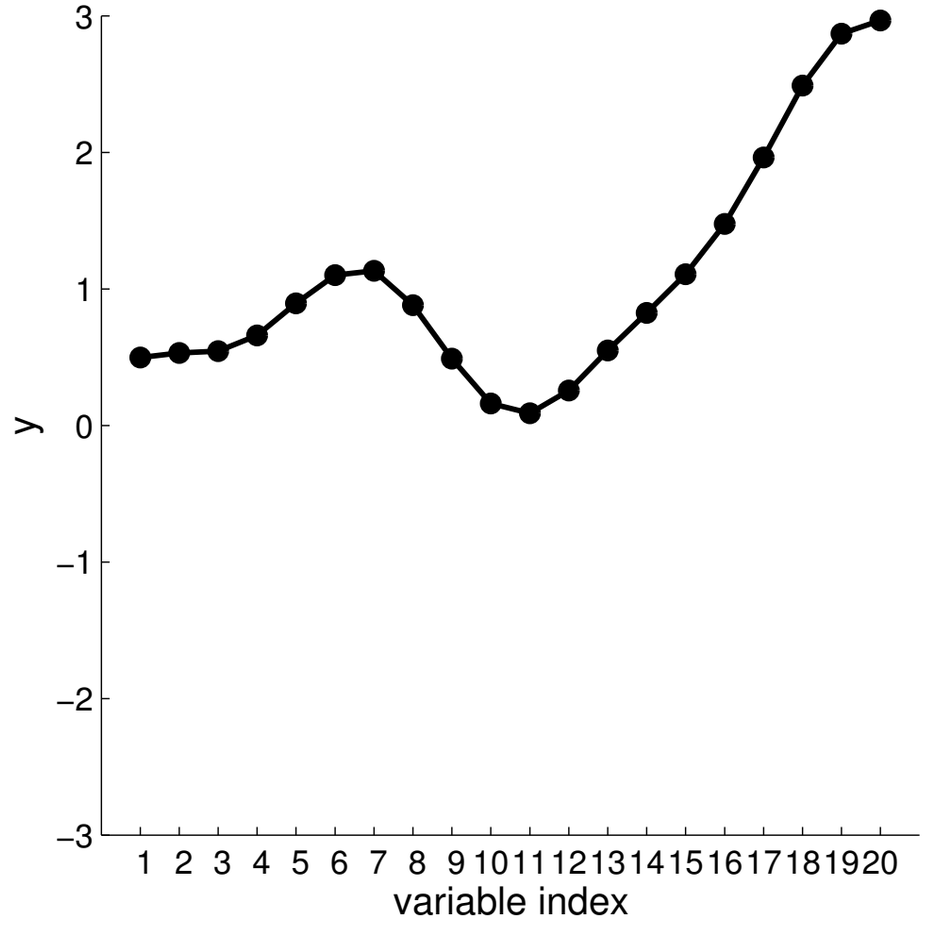
$\Sigma =$



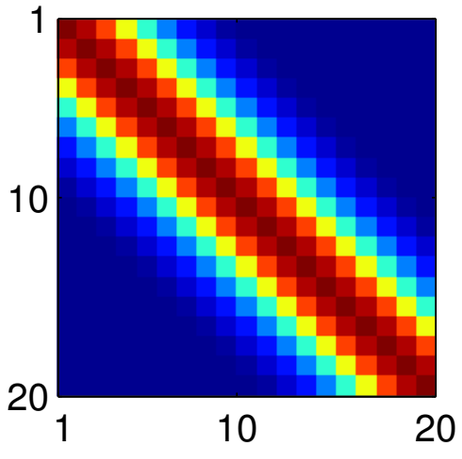
Special covariance matrix

What do those samples look like? Just smooth functions: in neabry points, we see similar values

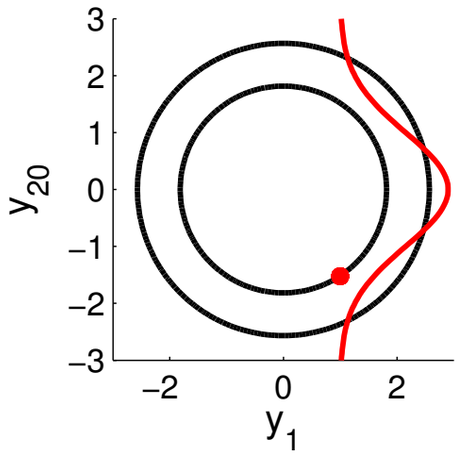


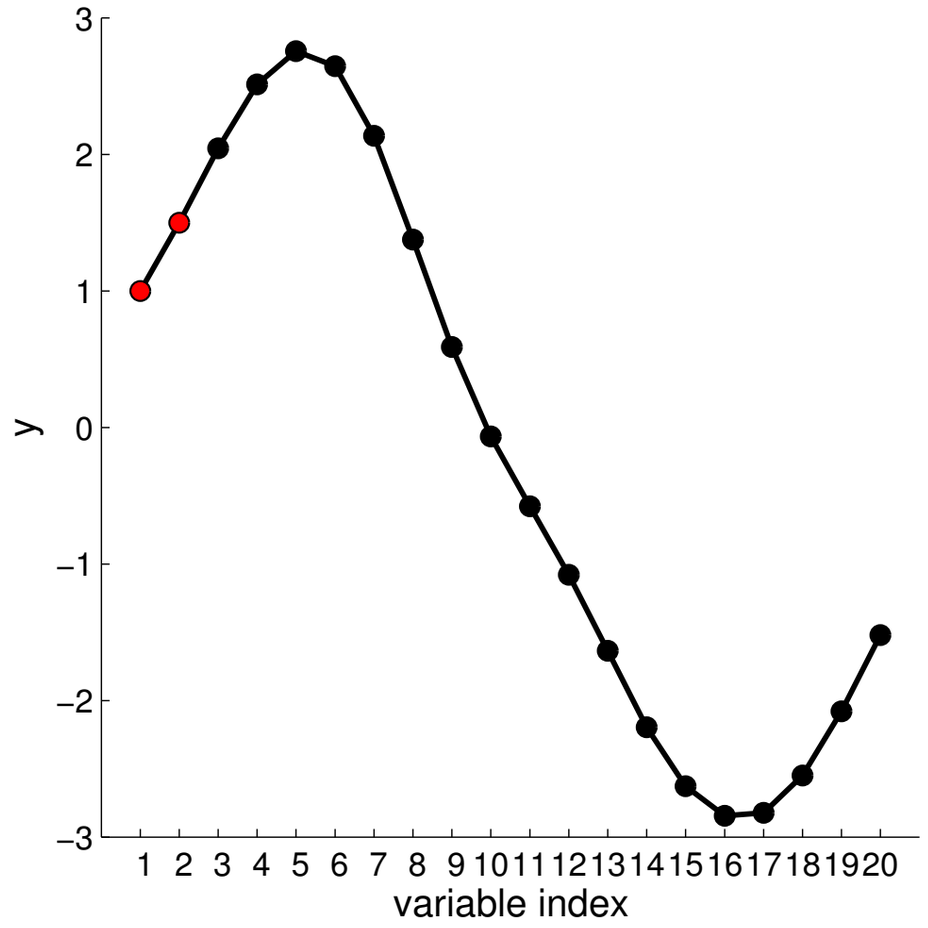


$\Sigma =$

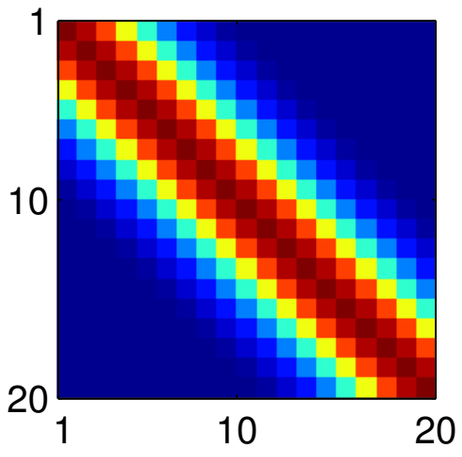


Special covariance matrix - conditioning



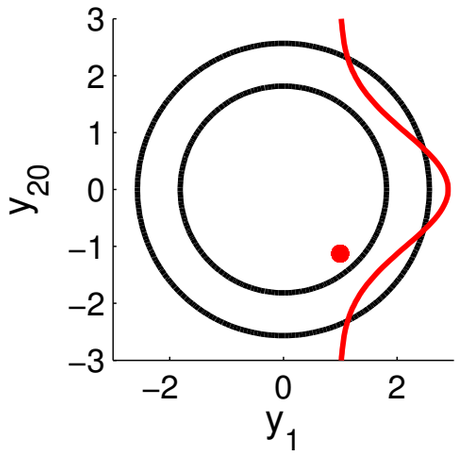


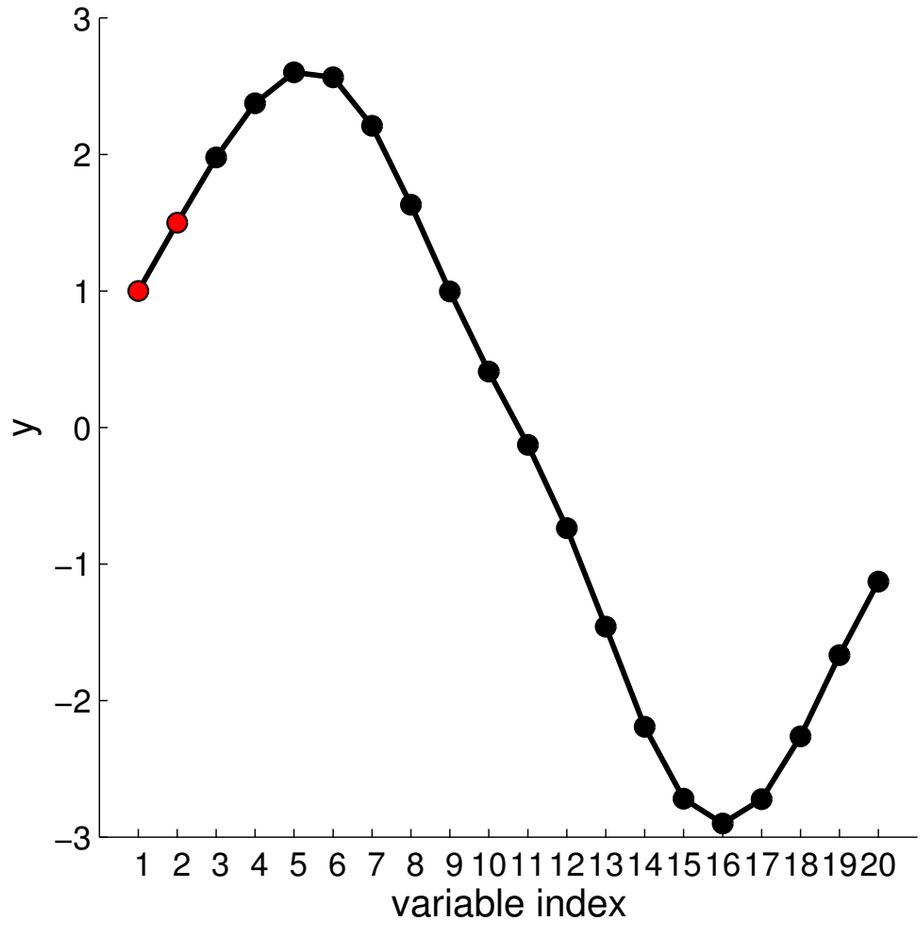
$\Sigma =$



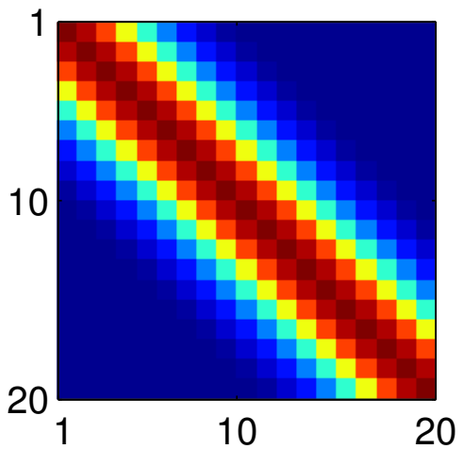
Conditioning on y_1 and y_2

Special covariance matrix - conditioning



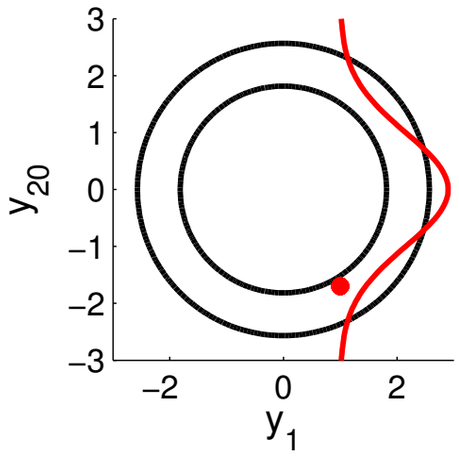


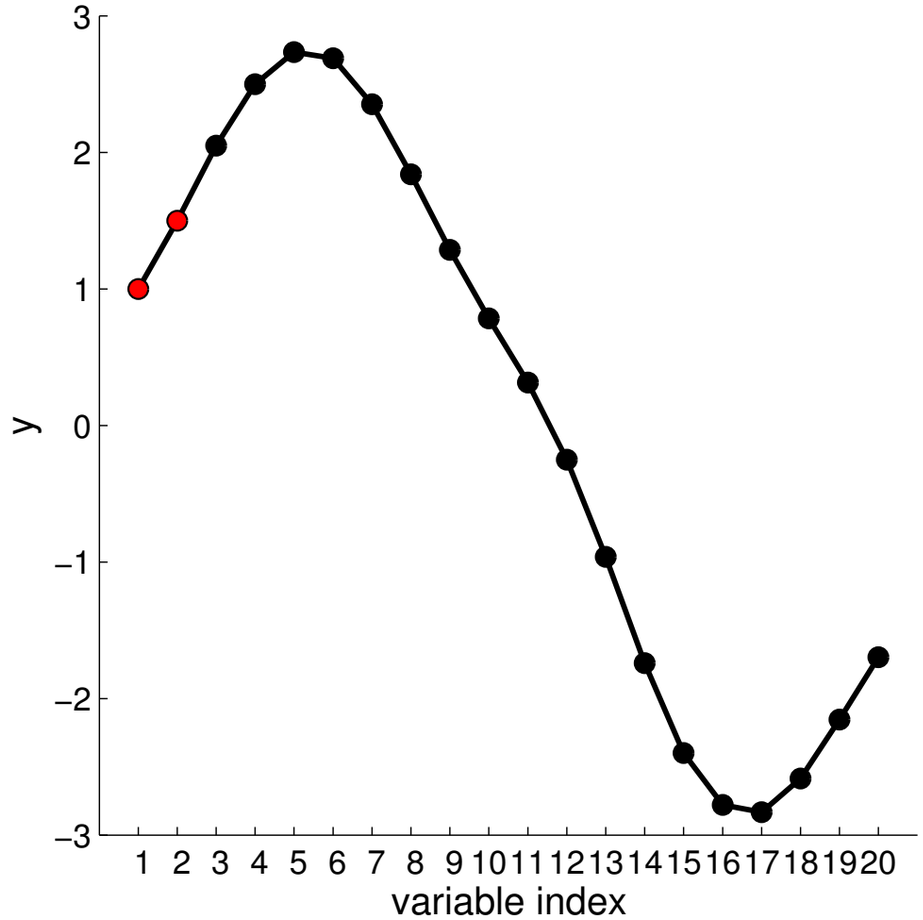
$\Sigma =$



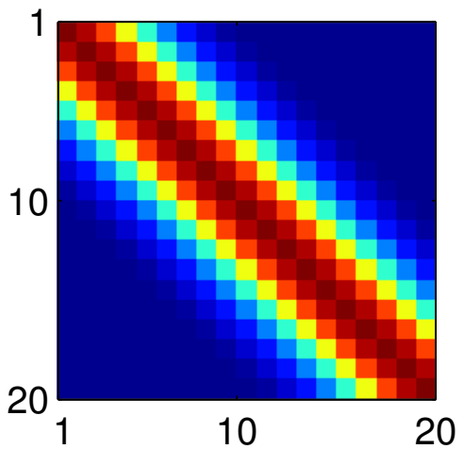
Conditioning on y_1 and y_2

Special covariance matrix - conditioning



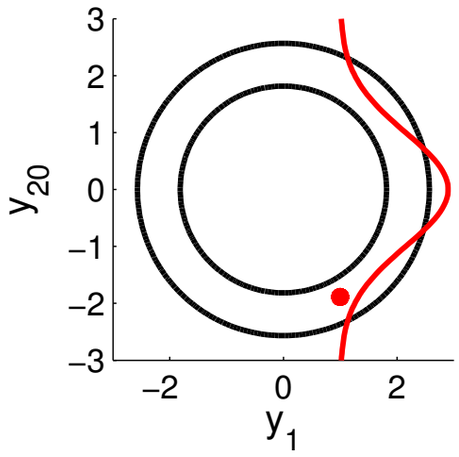


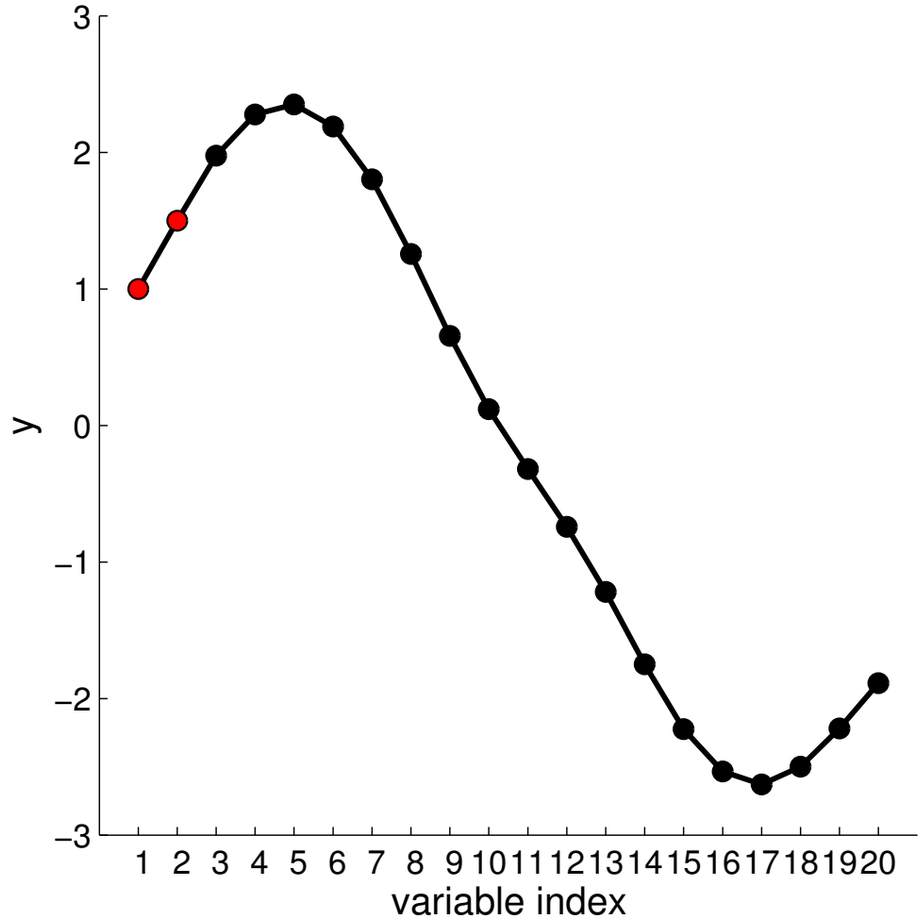
$\Sigma =$



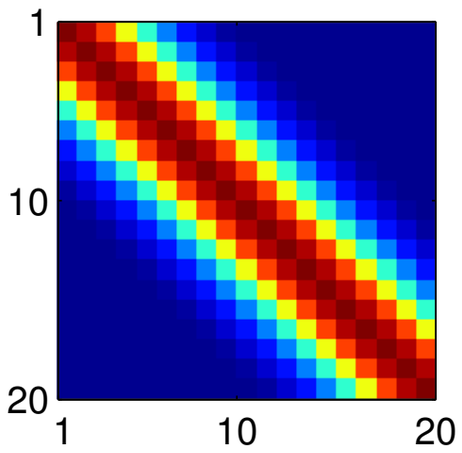
Conditioning on y_1 and y_2

Special covariance matrix - conditioning



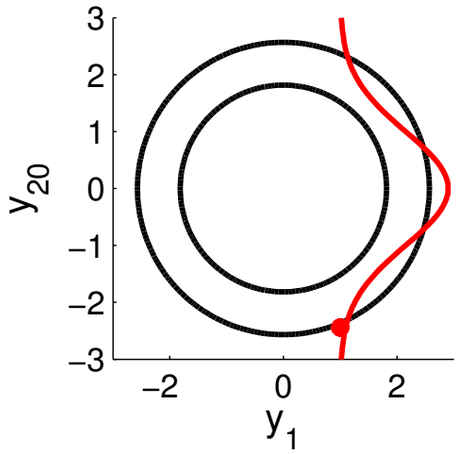


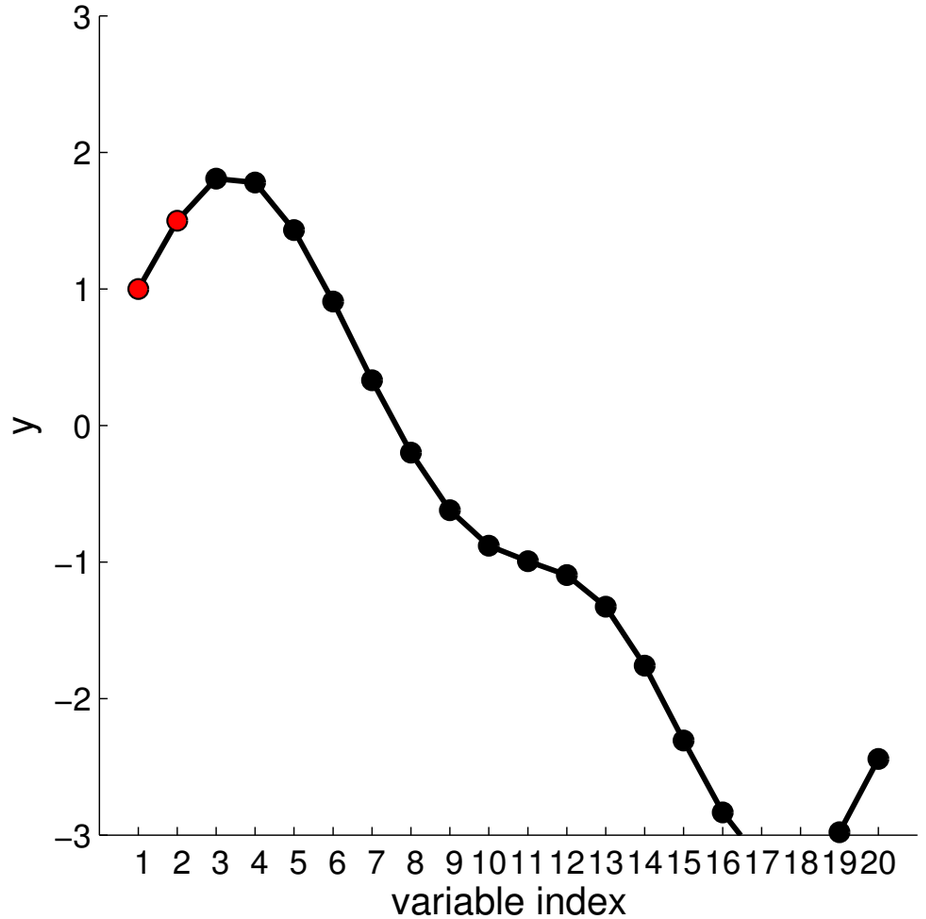
$\Sigma =$



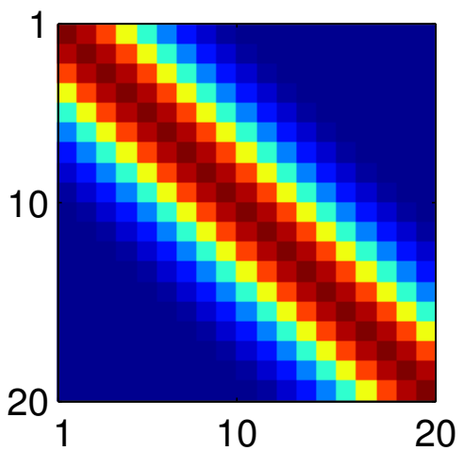
Conditioning on y_1 and y_2

Special covariance matrix - conditioning



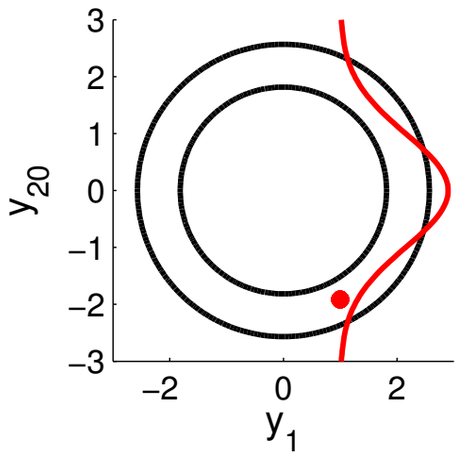


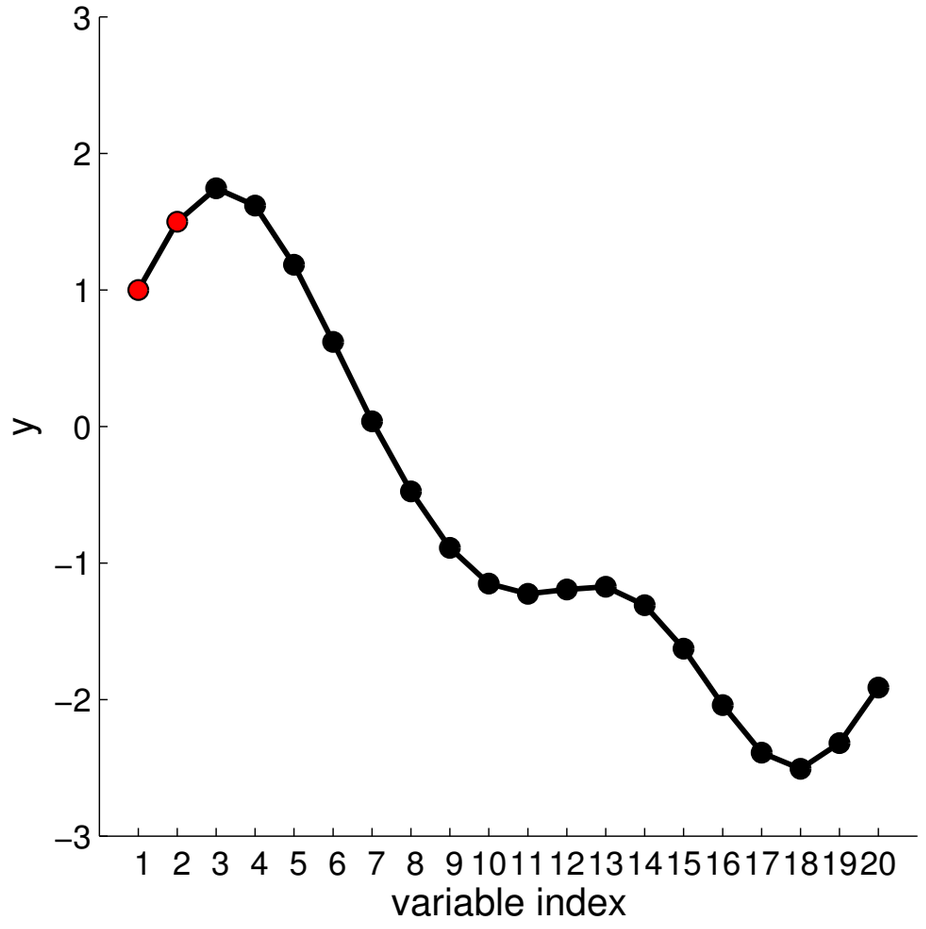
$\Sigma =$



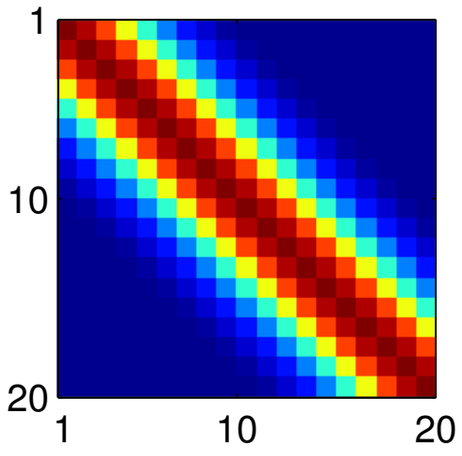
Conditioning on y_1 and y_2

Special covariance matrix - conditioning



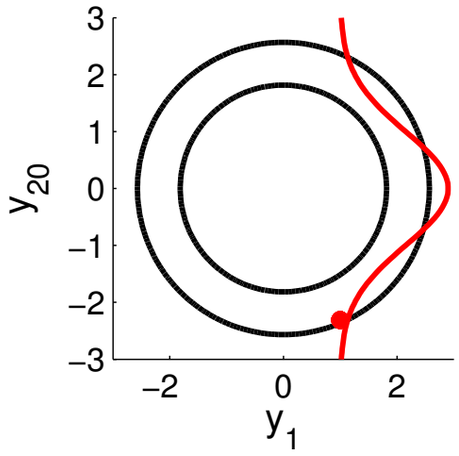


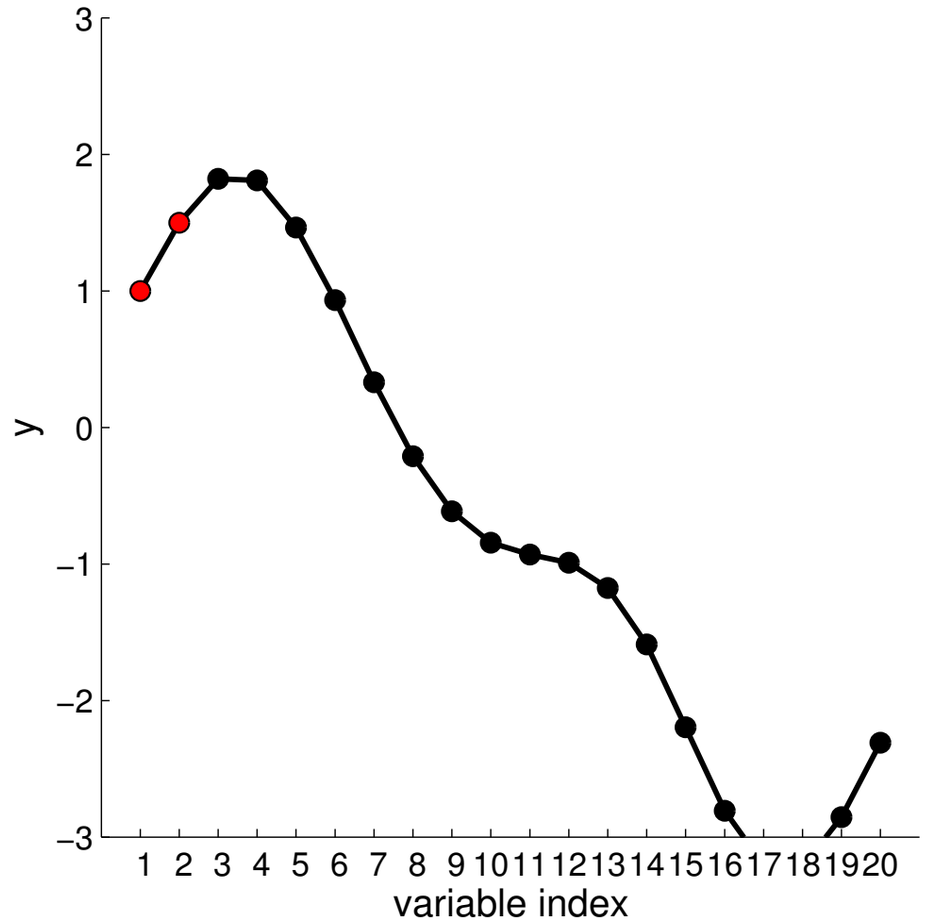
$\Sigma =$



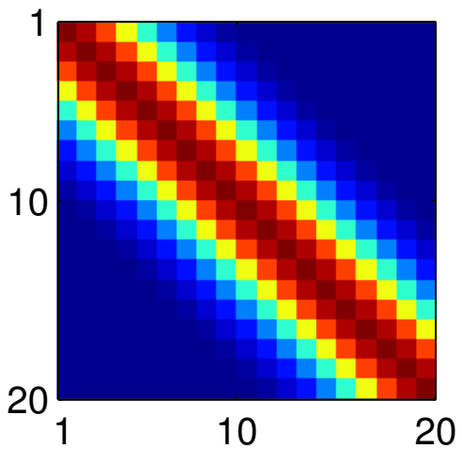
Conditioning on y_1 and y_2

Special covariance matrix - conditioning



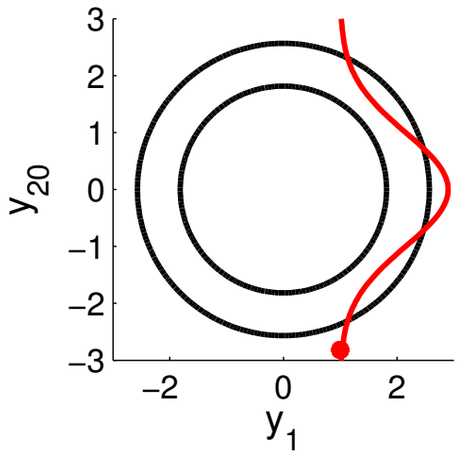


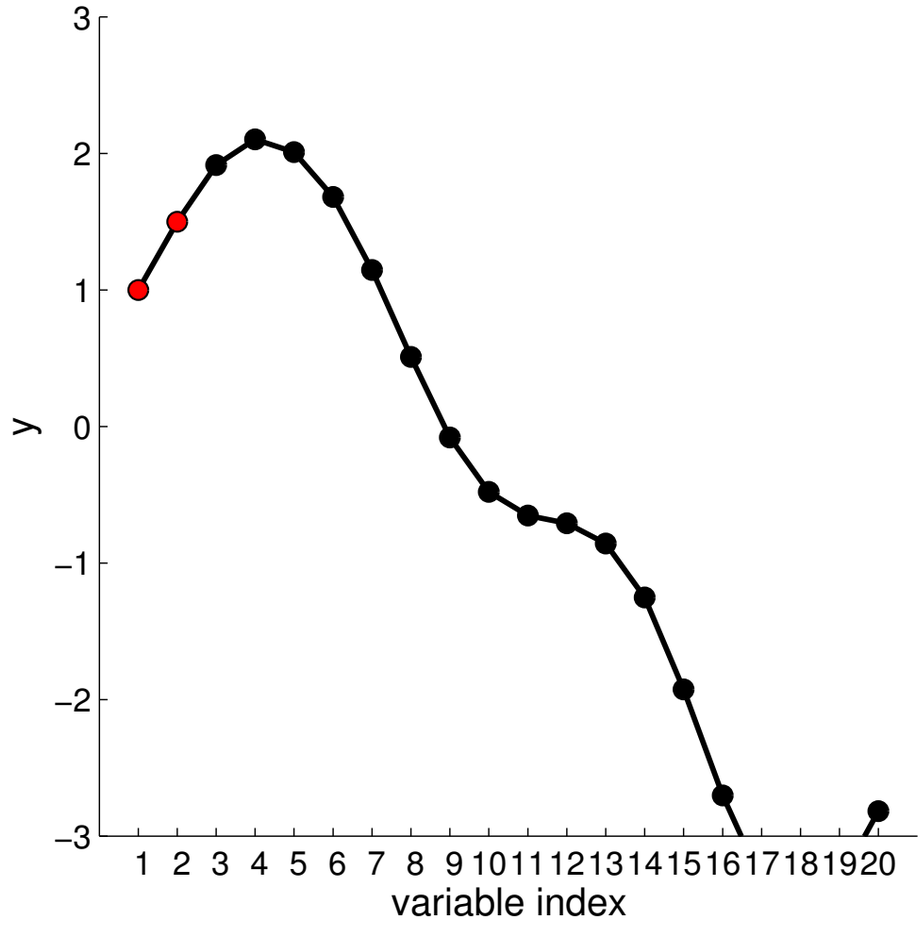
$\Sigma =$



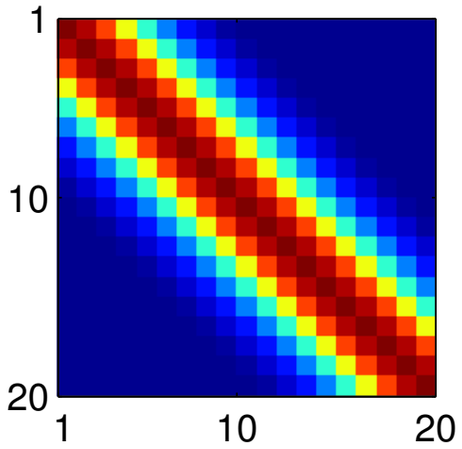
Conditioning on y_1 and y_2

Special covariance matrix - conditioning



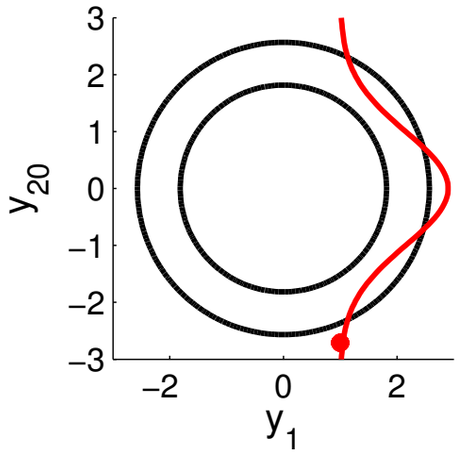


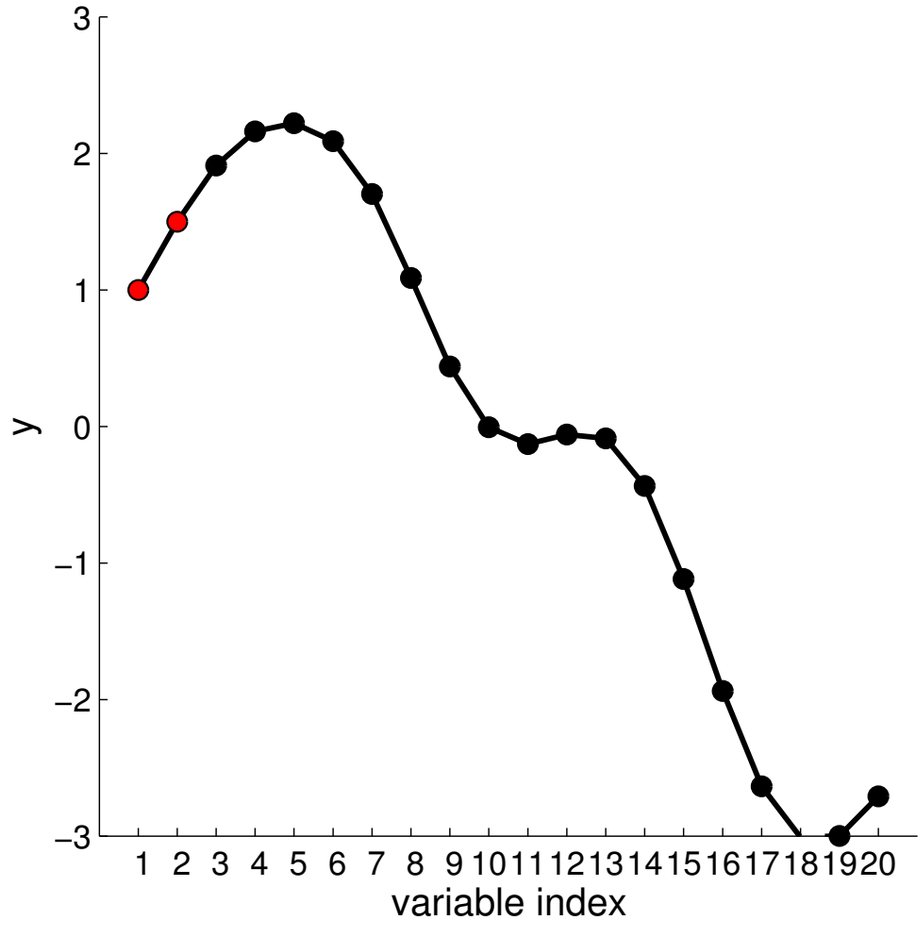
$\Sigma =$



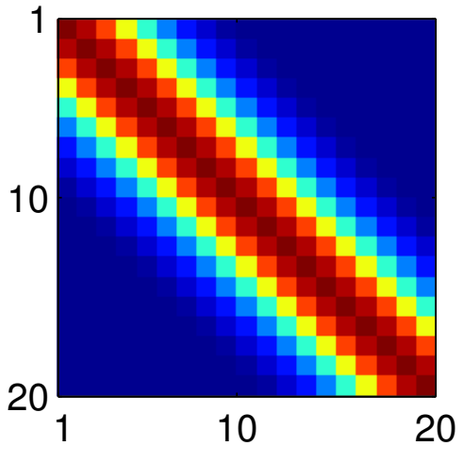
Conditioning on y_1 and y_2

Special covariance matrix - conditioning



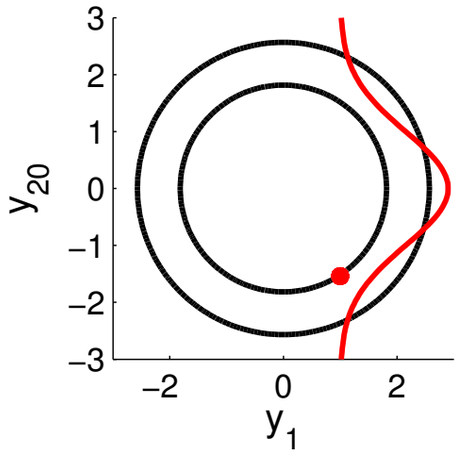


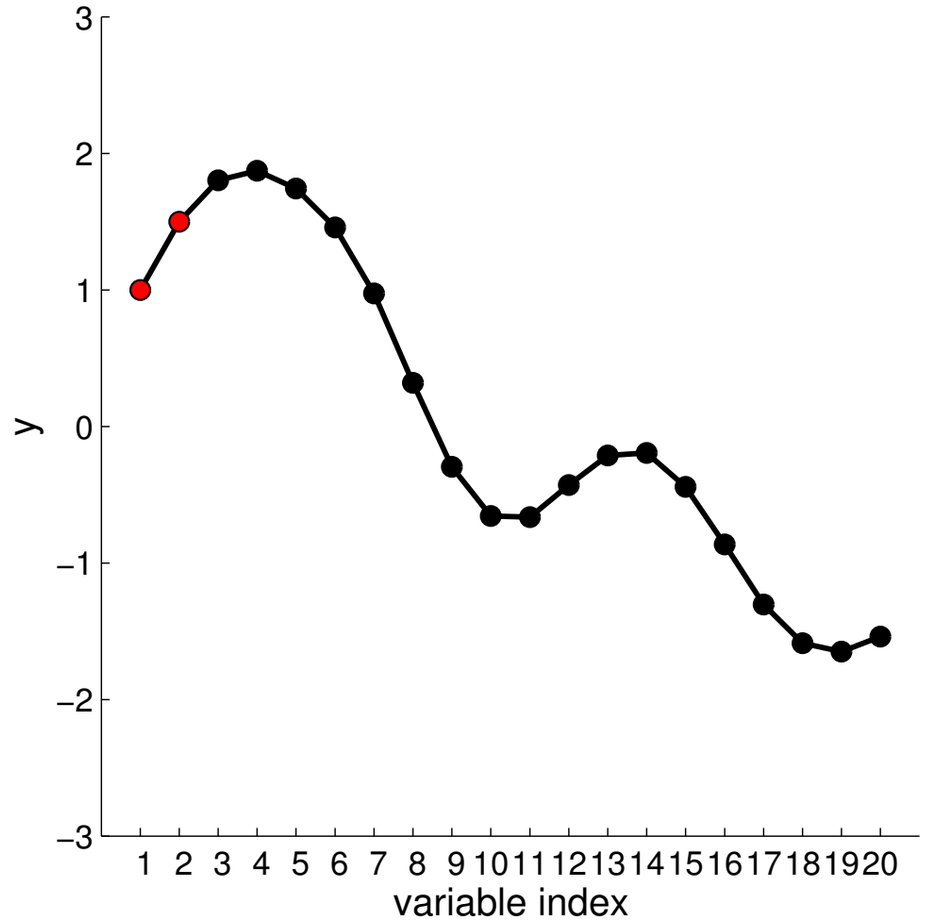
$\Sigma =$



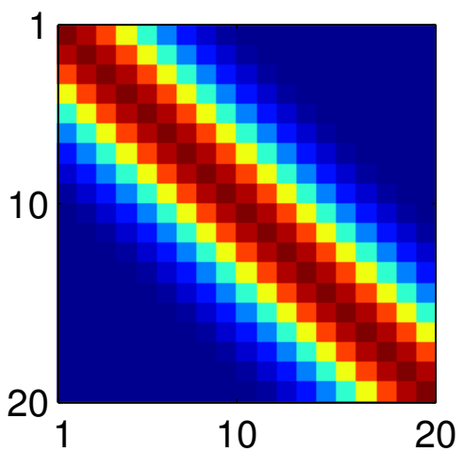
Conditioning on y_1 and y_2

Special covariance matrix - conditioning



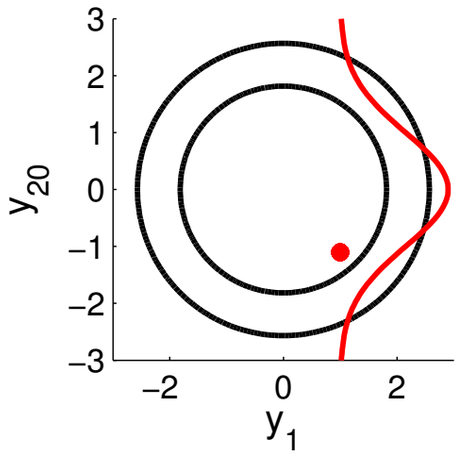


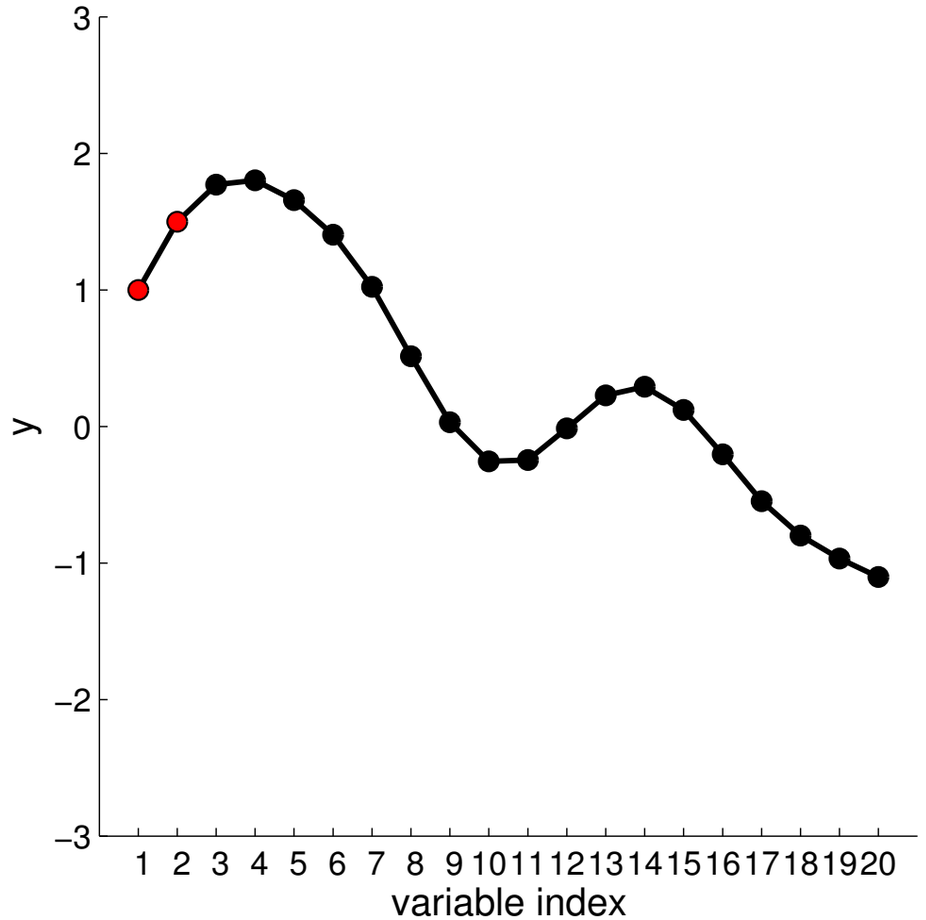
$\Sigma =$



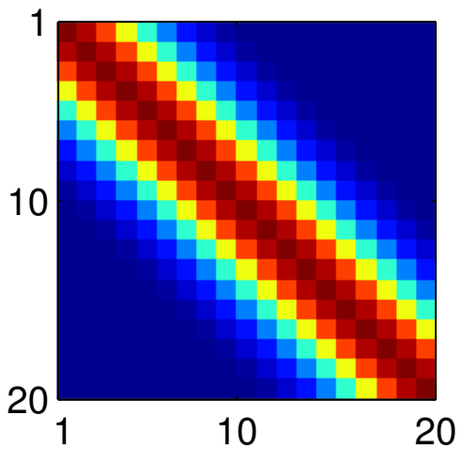
Conditioning on y1 and y2

Special covariance matrix - conditioning



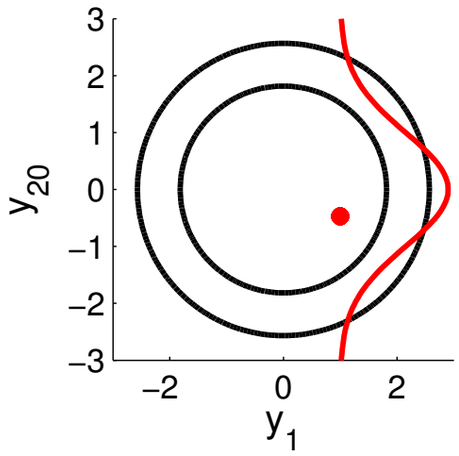


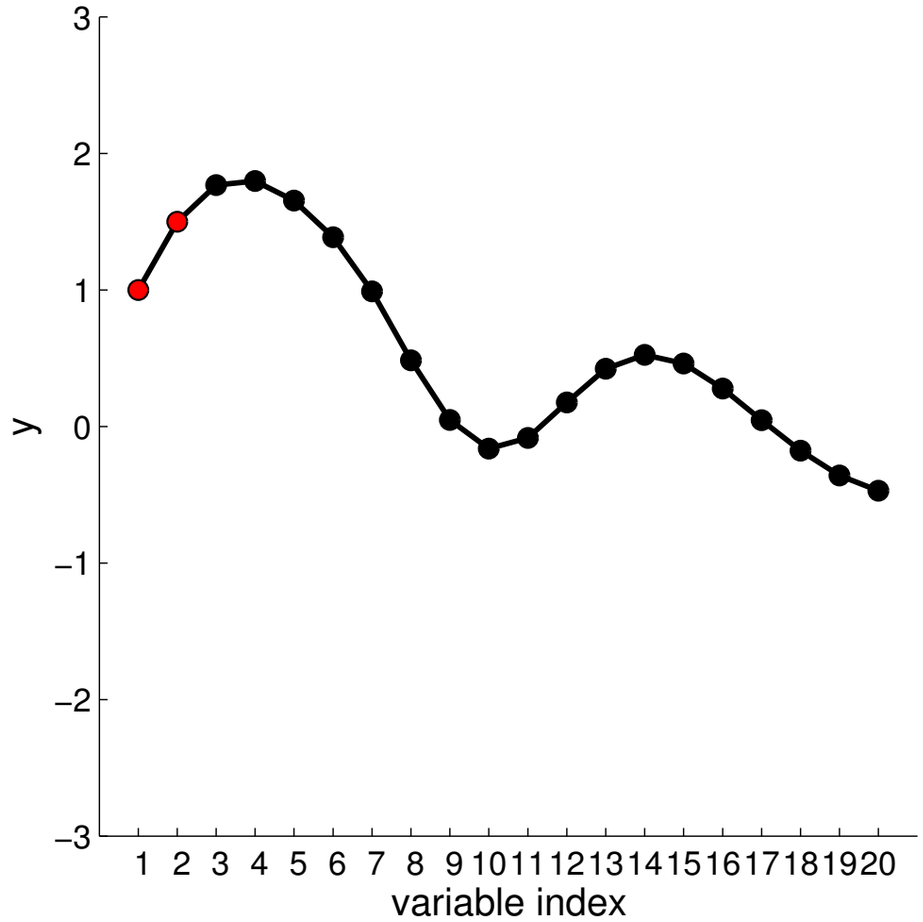
$\Sigma =$



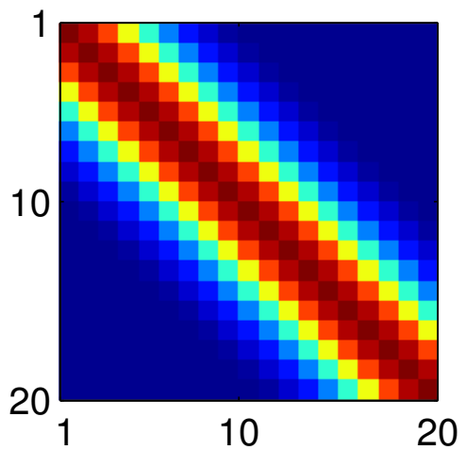
Conditioning on y1 and y2

Special covariance matrix - conditioning



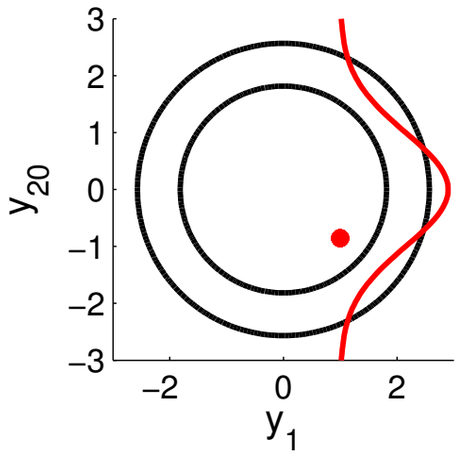


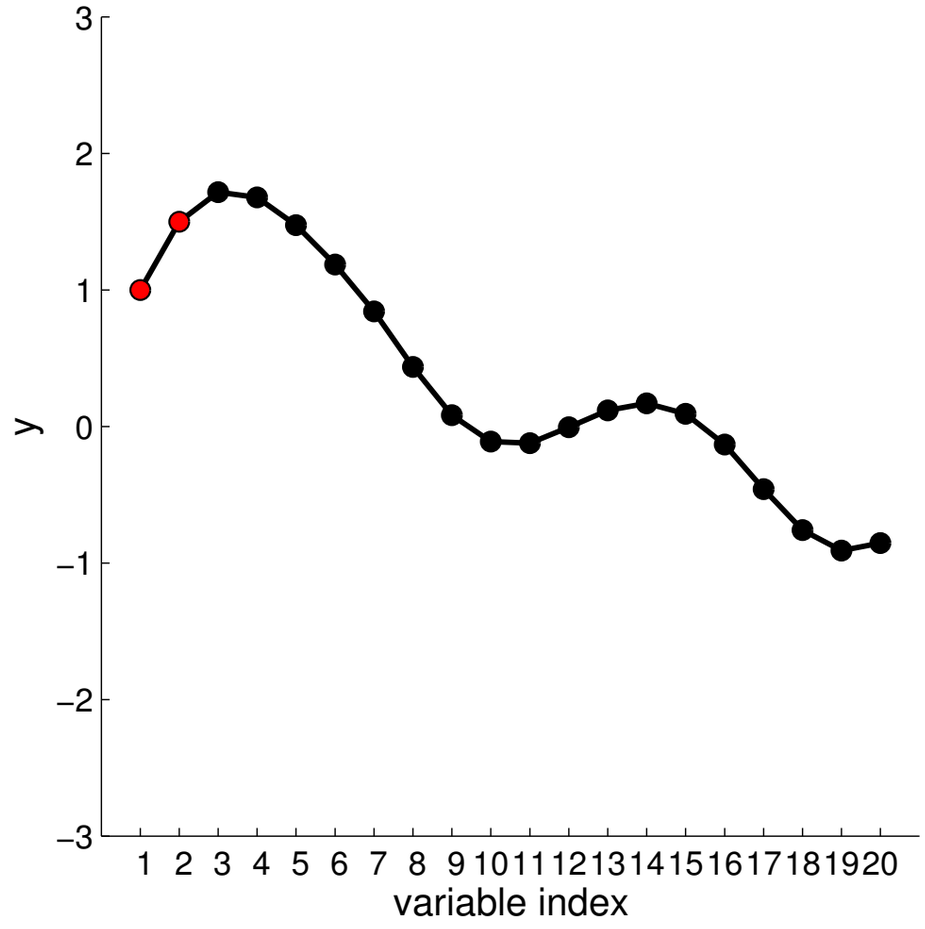
$\Sigma =$



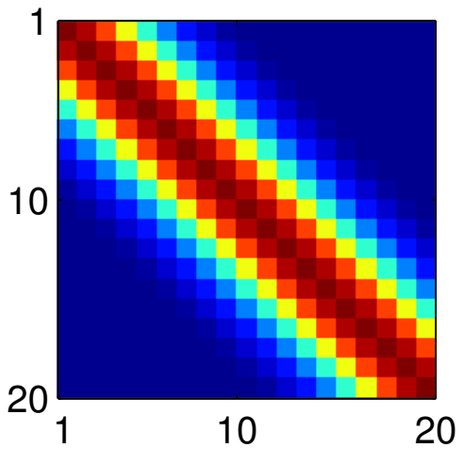
Conditioning on y1 and y2

Special covariance matrix - conditioning



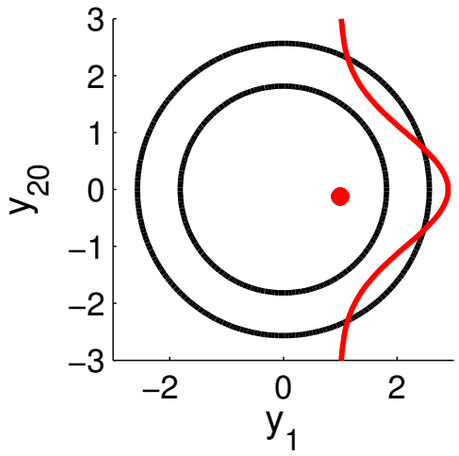


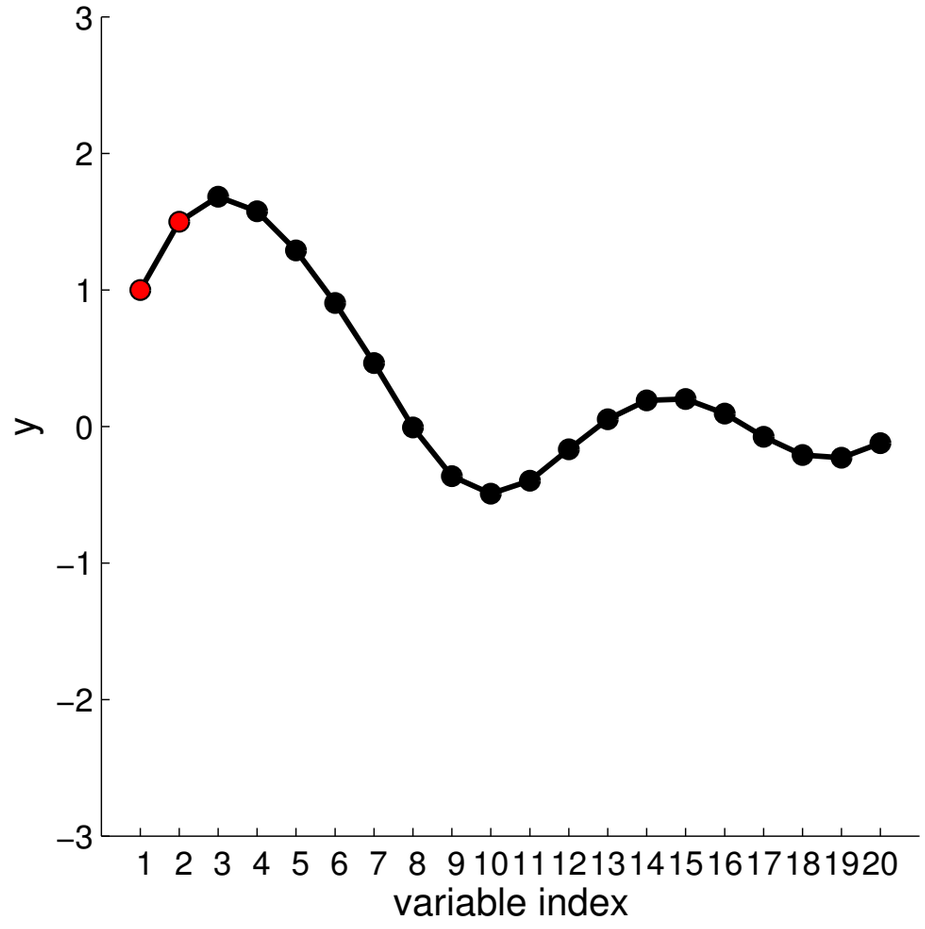
$\Sigma =$



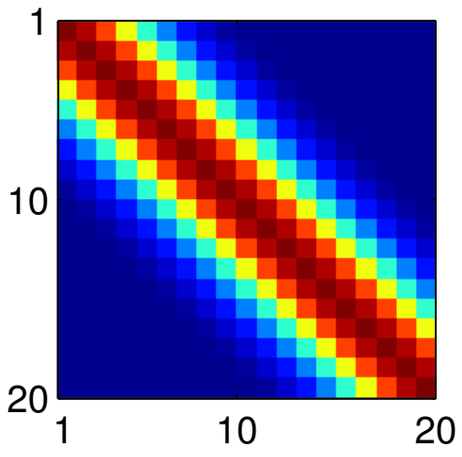
Conditioning on y1 and y2

Special covariance matrix - conditioning



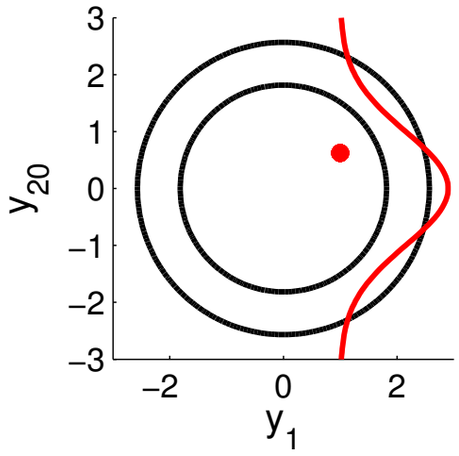


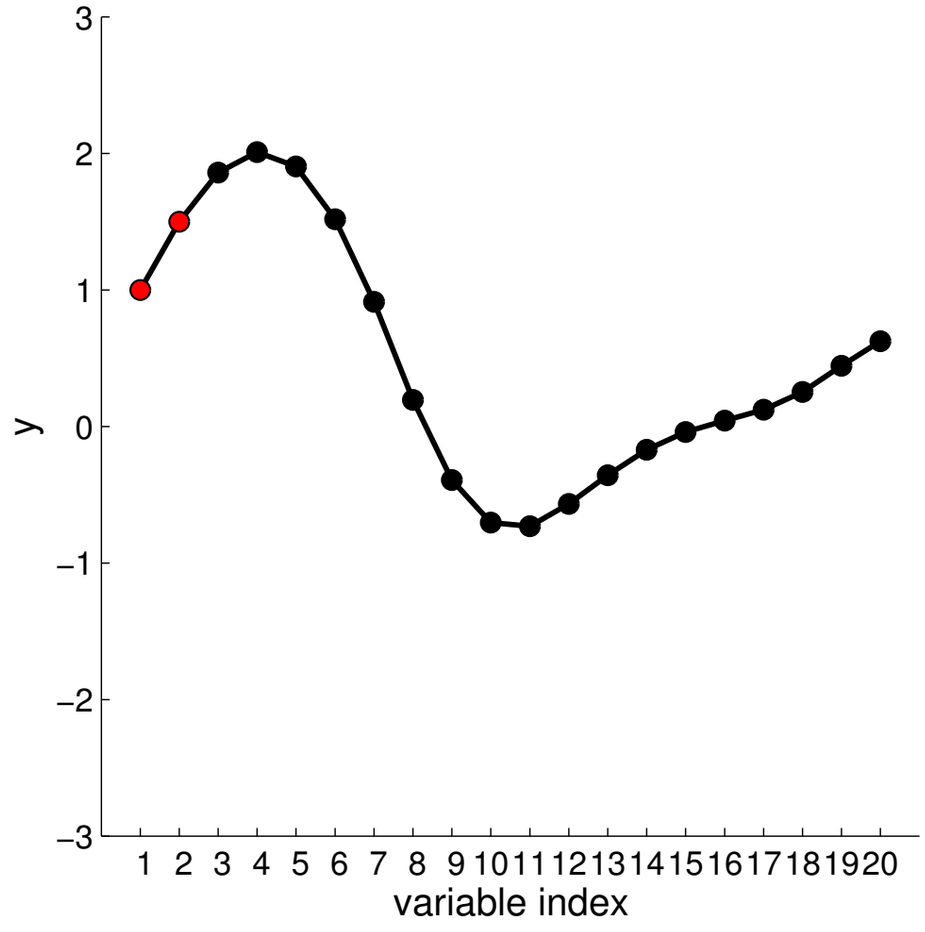
$\Sigma =$



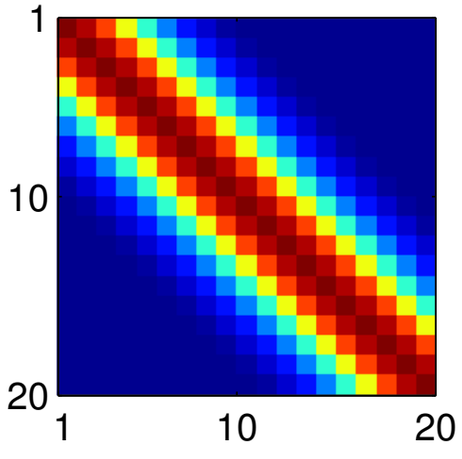
Conditioning on y1 and y2

Special covariance matrix - conditioning



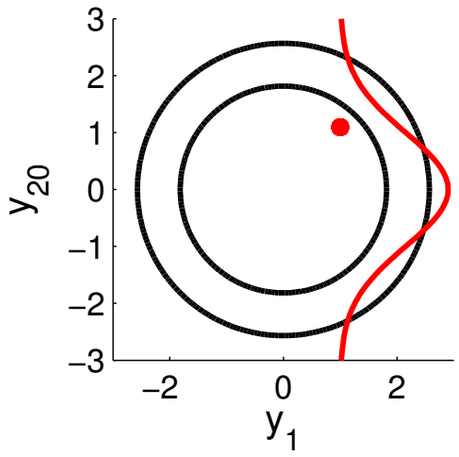


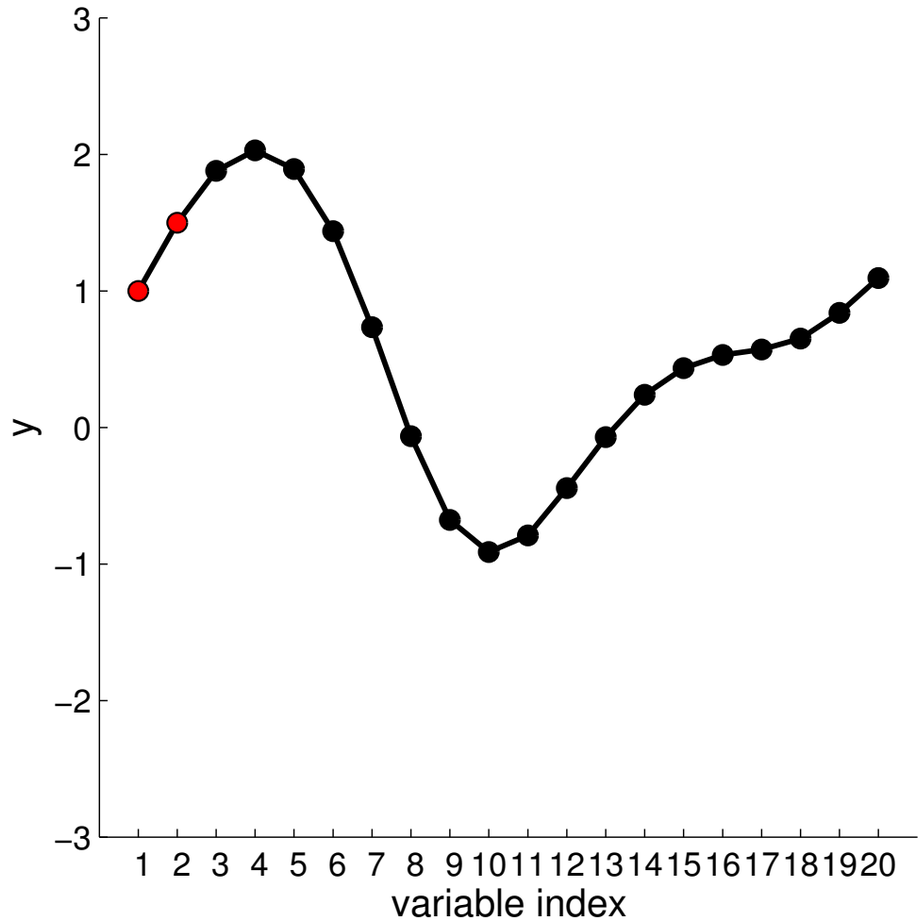
$\Sigma =$



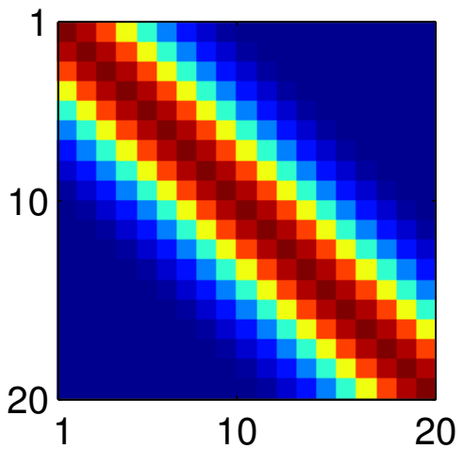
Conditioning on y1 and y2

Special covariance matrix - conditioning



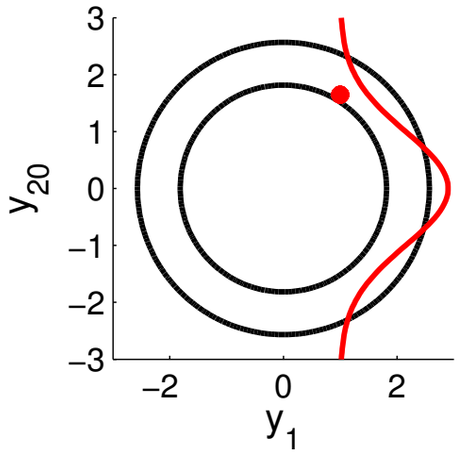


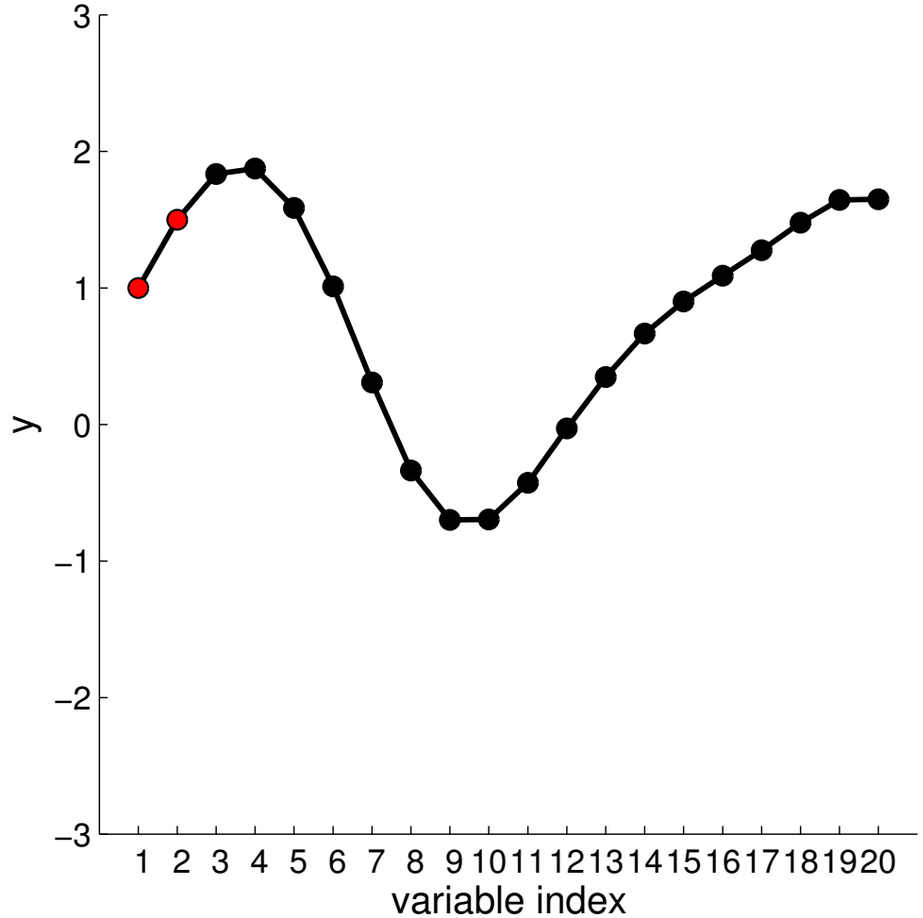
$\Sigma =$



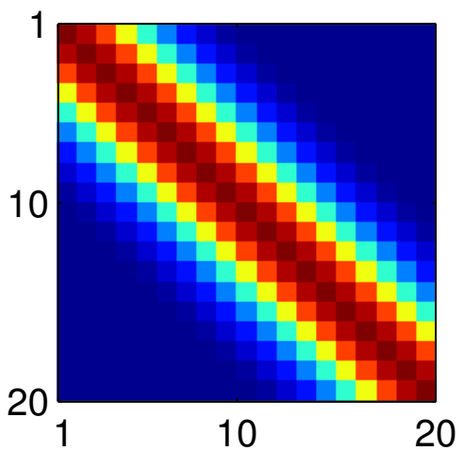
Conditioning on y1 and y2

Special covariance matrix - conditioning



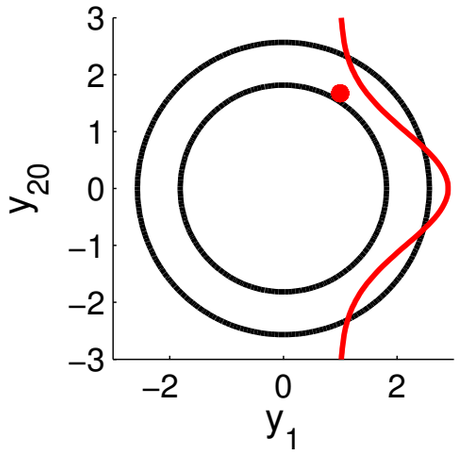


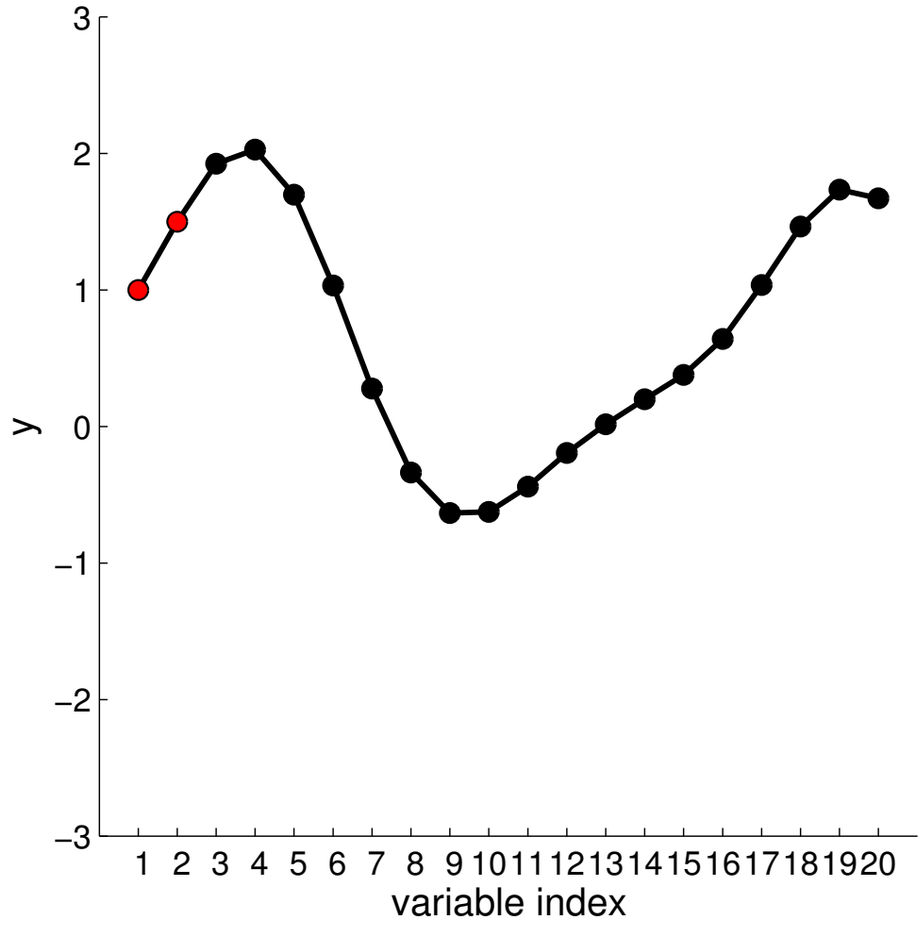
$\Sigma =$



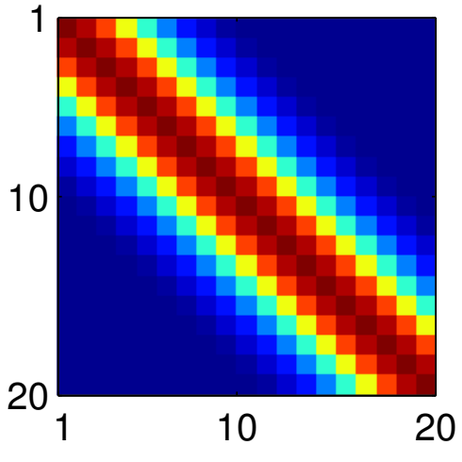
Conditioning on y1 and y2

Special covariance matrix - conditioning



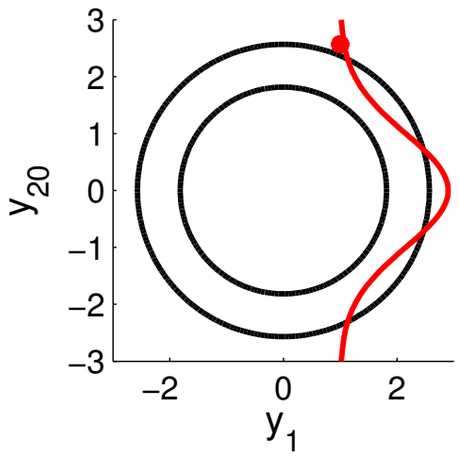


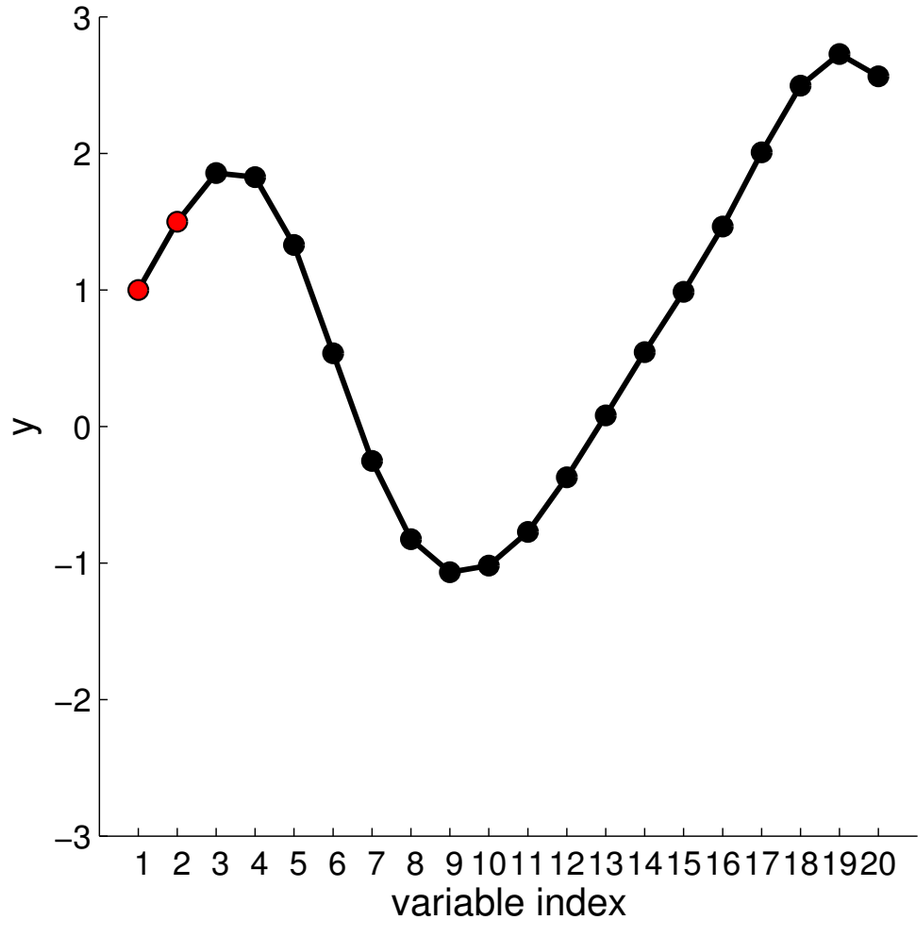
$\Sigma =$



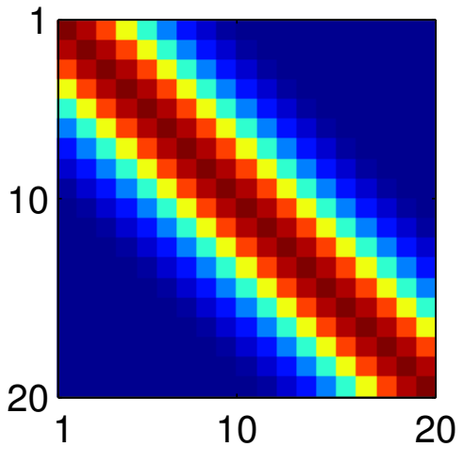
Conditioning on y1 and y2

Special covariance matrix - conditioning



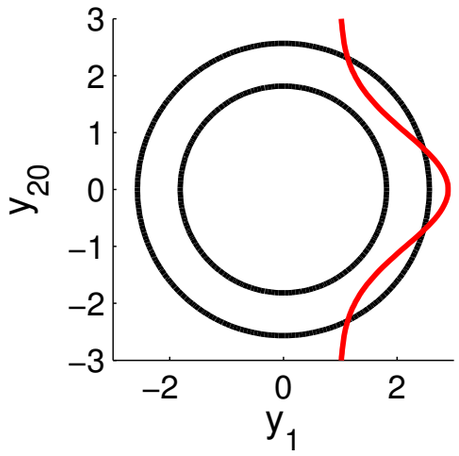


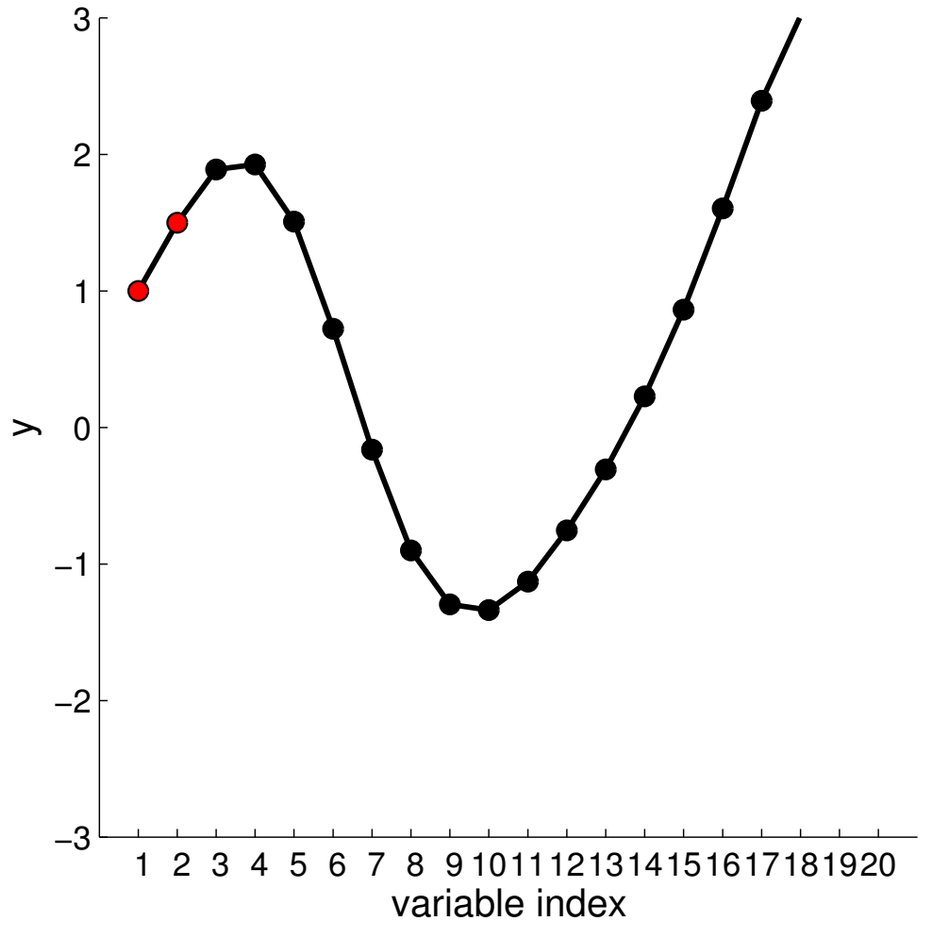
$\Sigma =$



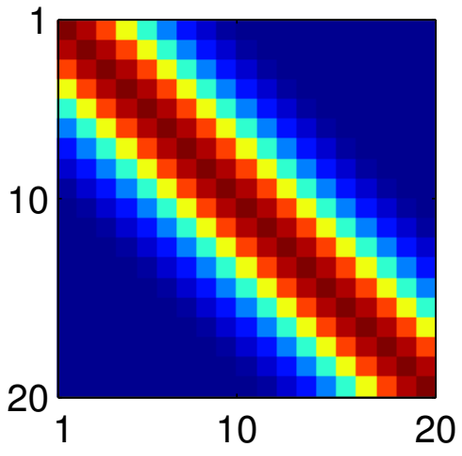
Conditioning on y1 and y2

Special covariance matrix - conditioning





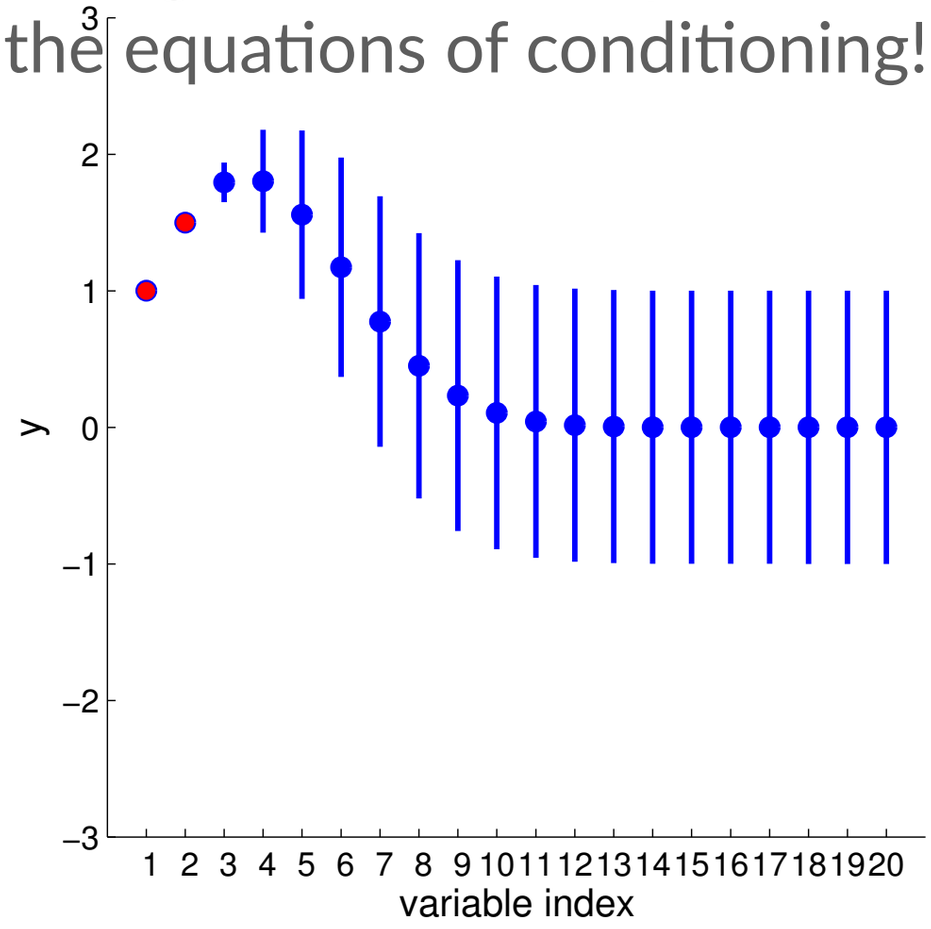
$\Sigma =$



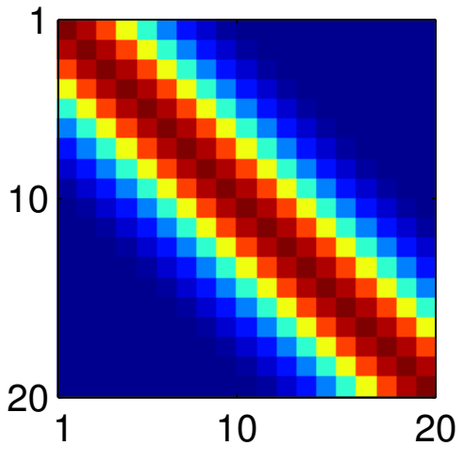
Conditioning on y1 and y2

Regression Using Gaussians

- If we average over lots of samples we can get mean and variance for each of the variables, conditioning on the observed red values! Exactly what we were looking for: Regression with error bars.
- Actually we do not need to average! We will compute means and variances analytically, using the equations of conditioning!



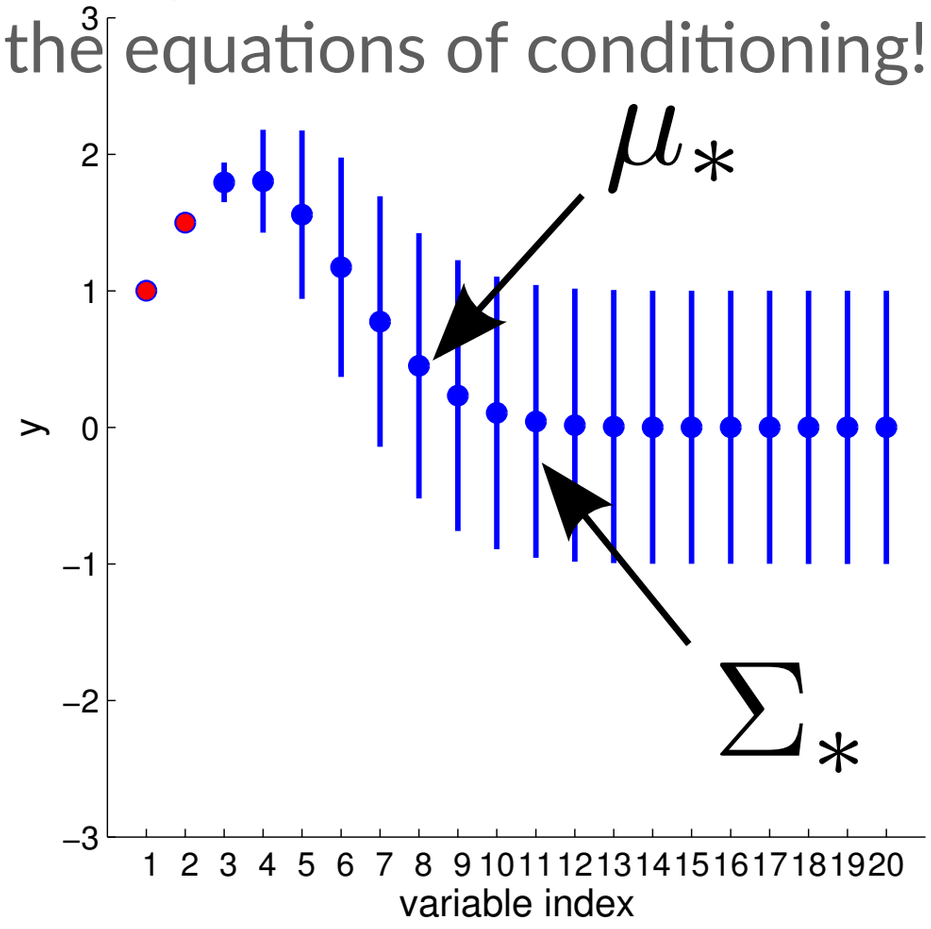
$\Sigma =$



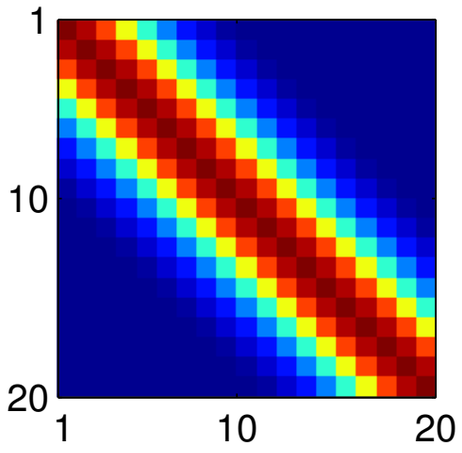
Conditioning on y1 and y2

Regression Using Gaussians

- If we average over the samples we can get mean and variance for each of the variables, conditioning on the observed red values! Exactly what we were looking for: Regression with error bars.
- Actually we do not need to average! We will compute means and variances analytically, using the equations of conditioning!



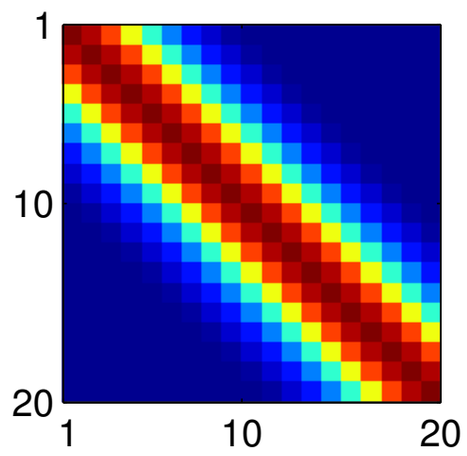
$\Sigma =$

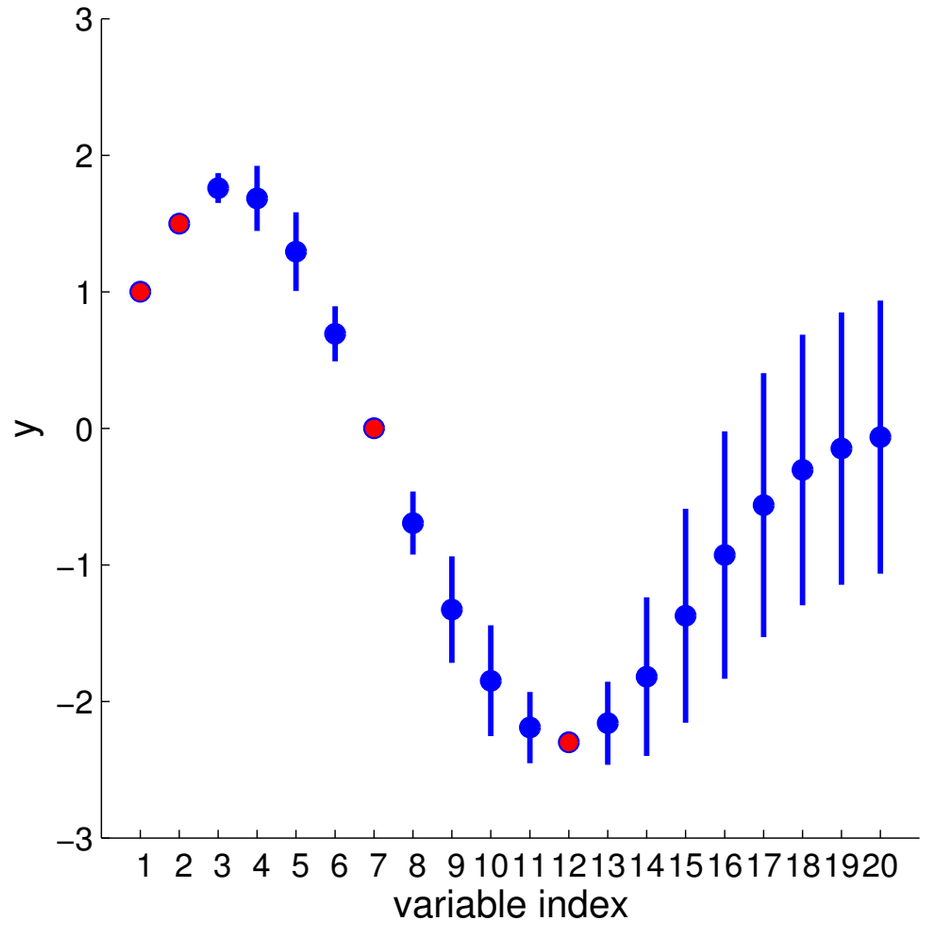


Regression Using Gaussians

We can also condition on non-contiguous indices

$\Sigma =$



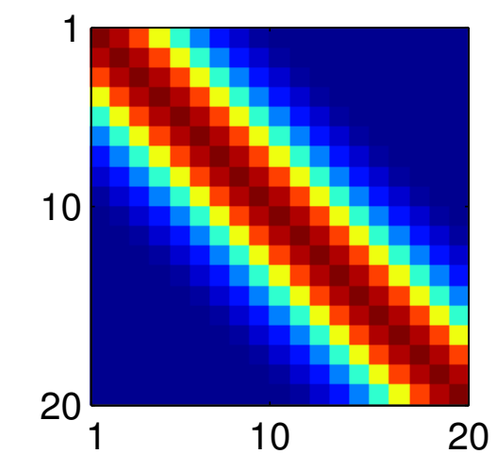


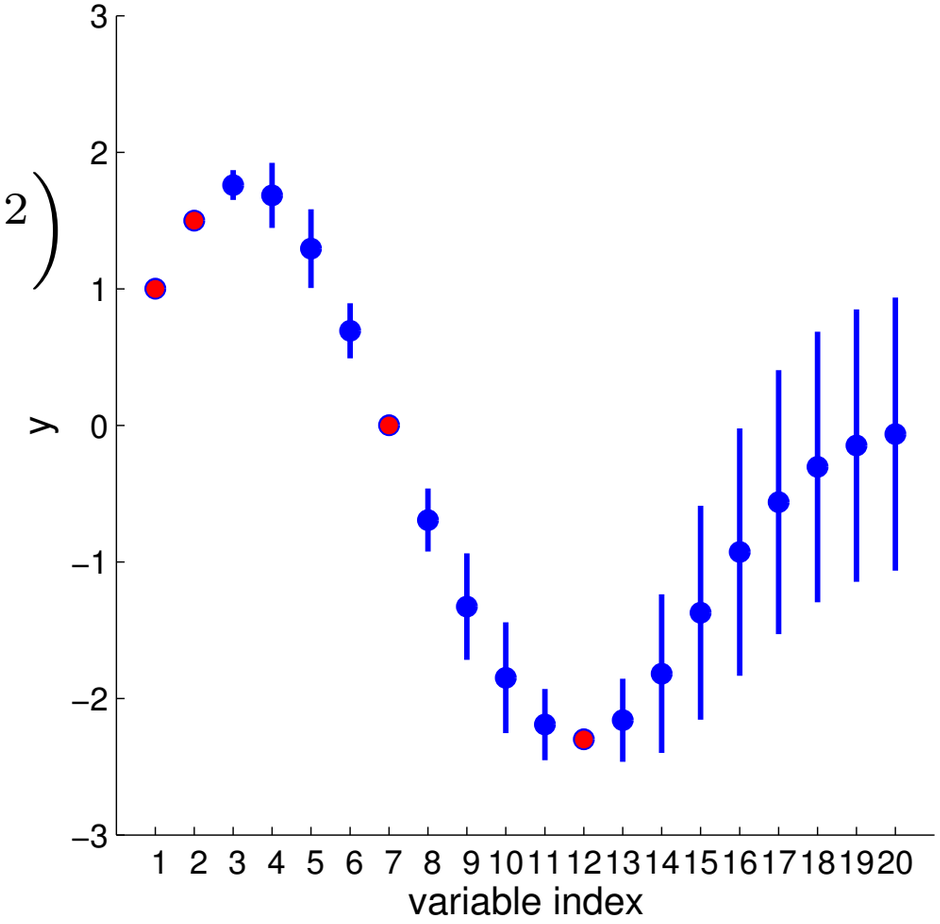
Regression Using Gaussians

$$\Sigma(\mathsf{x}_1, \mathsf{x}_2) = \mathsf{K}(\mathsf{x}_1, \mathsf{x}_2) + \mathsf{I}\sigma_{\mathsf{y}}^2$$

$$\mathsf{K}(\mathsf{x}_1, \mathsf{x}_2) = \sigma^2 \exp\left(-\frac{1}{2\mathsf{l}^2}(\mathsf{x}_1 - \mathsf{x}_2)^2\right)$$

$\Sigma =$





Q: Do x_1, x_2 need to be integers?

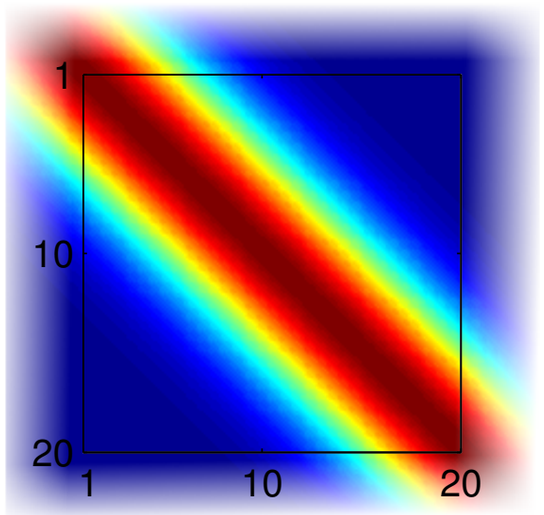
From multivariate Gaussian distributions to Gaussian Processes

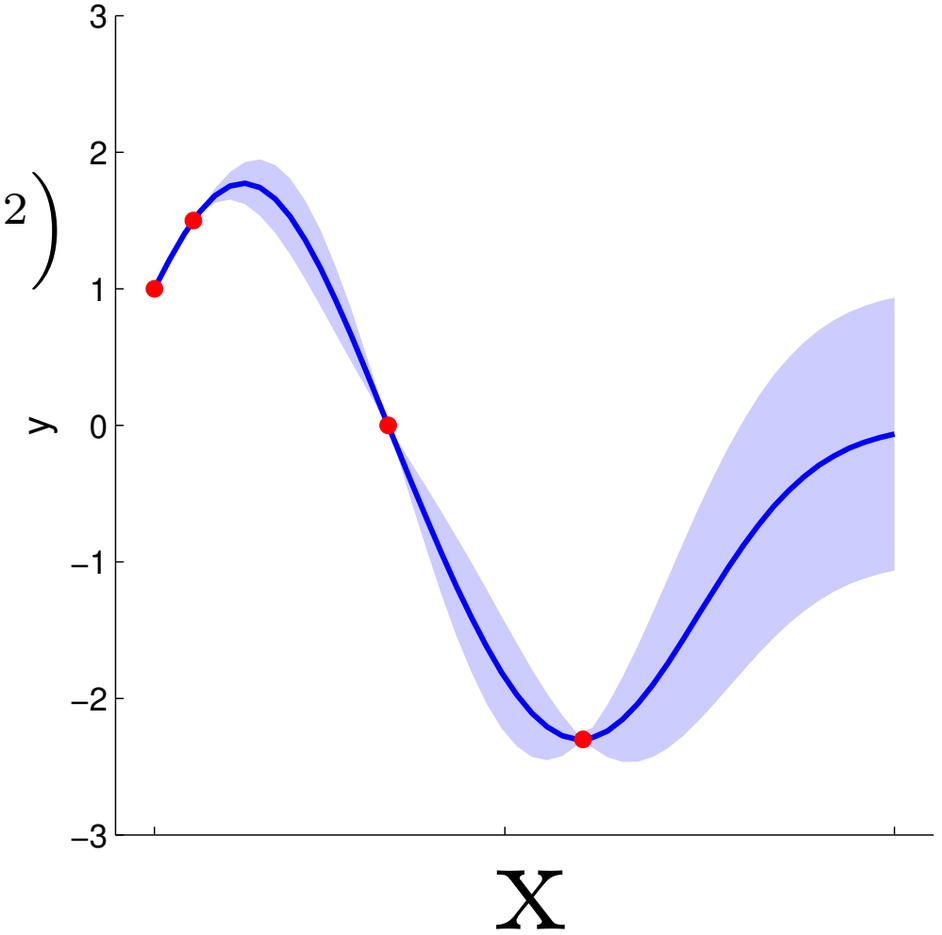
GP: a multivariate Gaussian over an uncountably infinite number of variables with infinite mean vector and infinite times infinite covariance matrix

$$\Sigma(\mathsf{x}_1, \mathsf{x}_2) = \mathsf{K}(\mathsf{x}_1, \mathsf{x}_2) + \mathsf{I}\sigma_{\mathsf{y}}^2$$

$$\mathsf{K}(\mathsf{x}_1, \mathsf{x}_2) = \sigma^2 \exp\left(-\frac{1}{2\mathsf{l}^2}(\mathsf{x}_1 - \mathsf{x}_2)^2\right)$$

$\Sigma =$





Gaussian Processes: Definition

Gaussian process = generalization of multivariate Gaussian distribution to infinitely many variables

Definition: a Gaussian process is a collection of random variables, any finite number of which have (consistent) Gaussian distributions

Gaussian Processes: Definition

Gaussian process = generalization of multivariate Gaussian distribution to infinitely many variables.

Definition: a Gaussian process is a collection of random variables, any finite number of which have (consistent) Gaussian distributions.

A Gaussian distribution is fully specified by a mean vector, μ, and a covariance matrix Σ:

$$\mathbf{f} = (f_1, \ldots, f_n) \sim \mathcal{N}(\mu, \Sigma), \text{ indices } i = 1, \ldots, n$$

Gaussian Processes: Definition

Gaussian process = generalization of multivariate Gaussian distribution to infinitely many variables

Definition: a Gaussian process is a collection of random variables, any finite number of which have (consistent) Gaussian distributions.

A Gaussian distribution is fully specified by a mean vector, μ, and a covariance matrix Σ:

$$\mathbf{f} = (f_1, \ldots, f_n) \sim \mathcal{N}(\mu, \Sigma), \text{ indices } i = 1, \ldots, n$$

A Gaussian process is fully specified by a mean function $m(\mathbf{x})$ and a covariance function $K(\mathbf{x}, \mathbf{x}')$:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), \mathbf{K}(\mathbf{x}, \mathbf{x}')\right), \text{ indices } \mathbf{x}$$

Mathematical justification

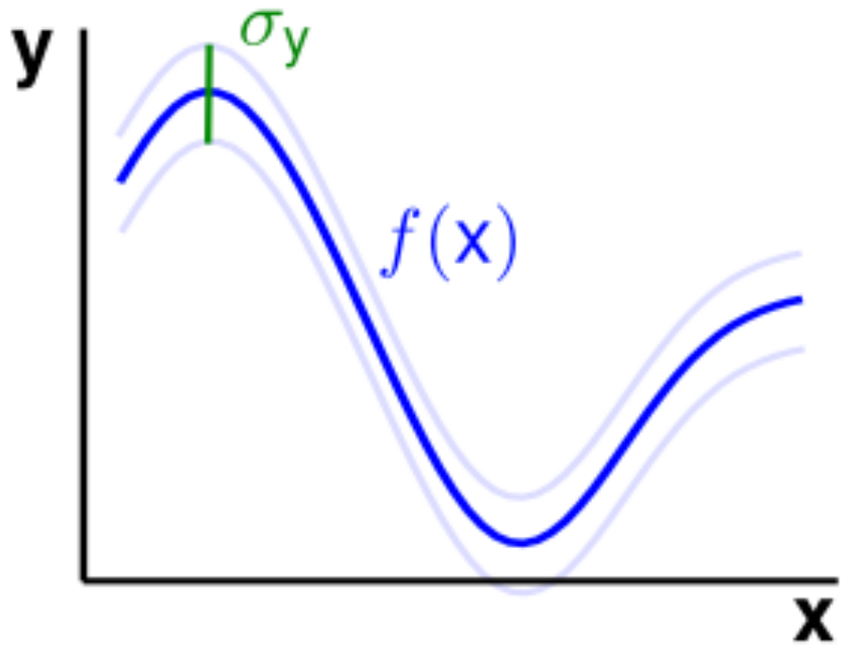
Q: What is a formal justification for how we are using GPs for regression?

Mathematical justification

Q: What is a formal justification for how we are using GPs for regression?

Generative model (like non-linear regression)

$$\mathrm{y}(\mathrm{x}) = f(\mathrm{x}) + \epsilon\sigma_{\mathrm{y}}$$

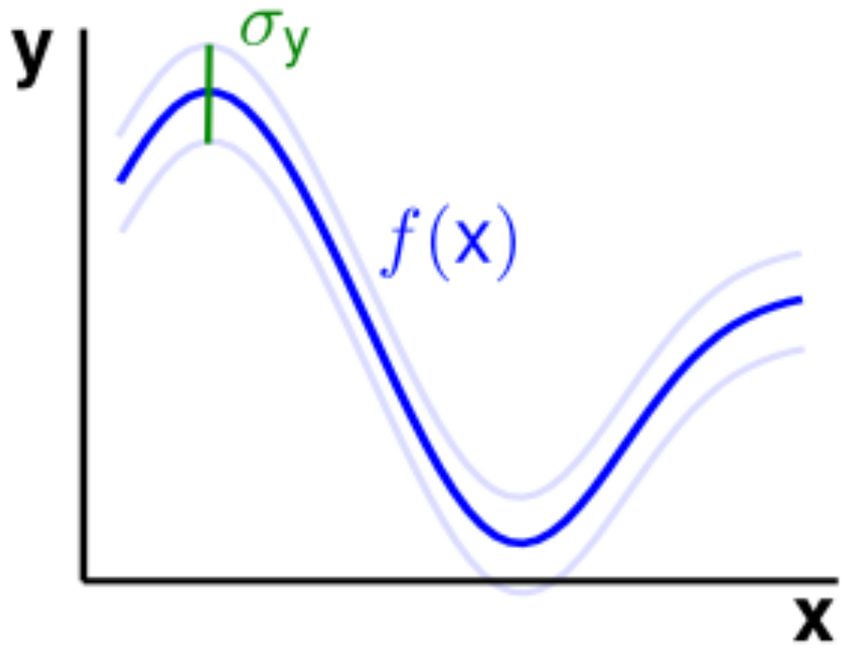


Mathematical justification

Q: What is a formal justification for how we are using GPs for regression?

Generative model (like non-linear regression)

$$\mathrm{y}(\mathrm{x}) = f(\mathrm{x}) + \epsilon\sigma_{\mathrm{y}}$$
$$p(\epsilon) = \mathcal{N}(0,1)$$

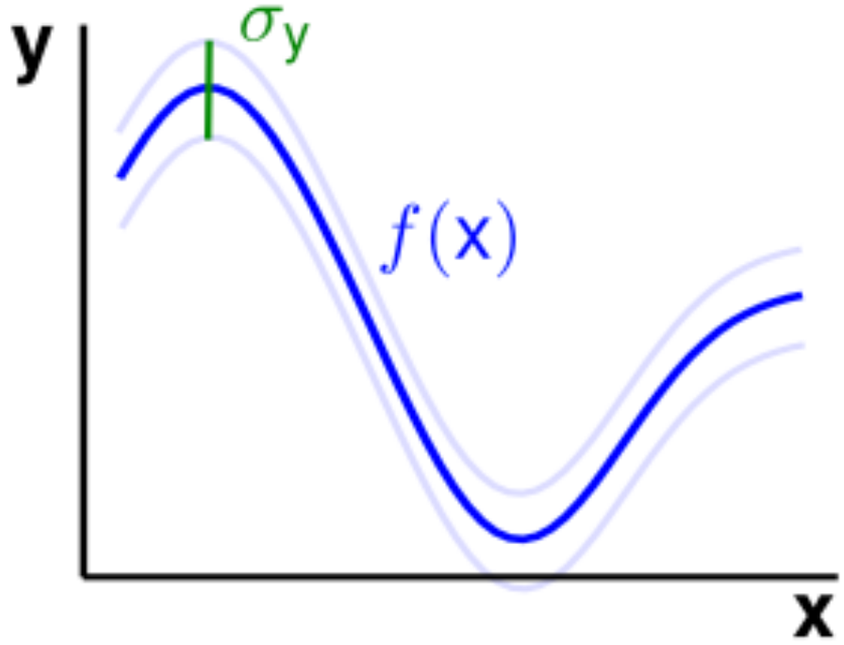


Mathematical justification

Q: What is a formal justification for how we are using GPs for regression?

Generative model (like non-linear regression)

$$\mathrm{y}(\mathrm{x}) = f(\mathrm{x}) + \epsilon\sigma_{\mathrm{y}}$$
$$p(\epsilon) = \mathcal{N}(0,1)$$



place GP prior over the non-linear function

$$p(f(\mathrm{x})\,|\,\theta) = \mathcal{GP}(0,\mathrm{K}(\mathrm{x},\mathrm{x}'))$$
$$\mathrm{K}(\mathrm{x},\mathrm{x}') = \sigma^2 \exp\left(-\frac{1}{2l^2}(\mathrm{x}-\mathrm{x}')^2\right)$$

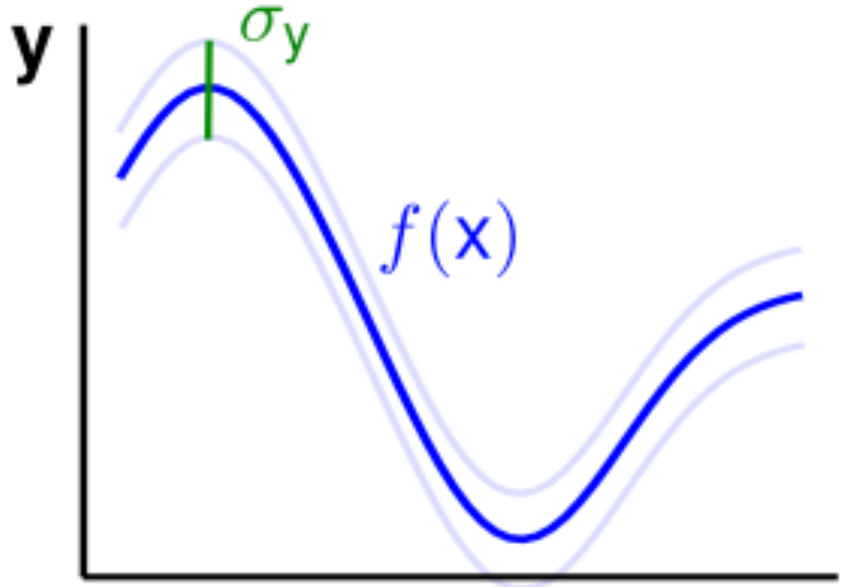
(smoothly wiggling functions expected)

Mathematical justification

Q: What is a formal justification for how we are using GPs for regression?

Generative model (like non-linear regression)

$$\mathrm{y}(\mathrm{x}) = f(\mathrm{x}) + \epsilon\sigma_{\mathrm{y}}$$
$$p(\epsilon) = \mathcal{N}(0,1)$$



place GP prior over the non-linear function

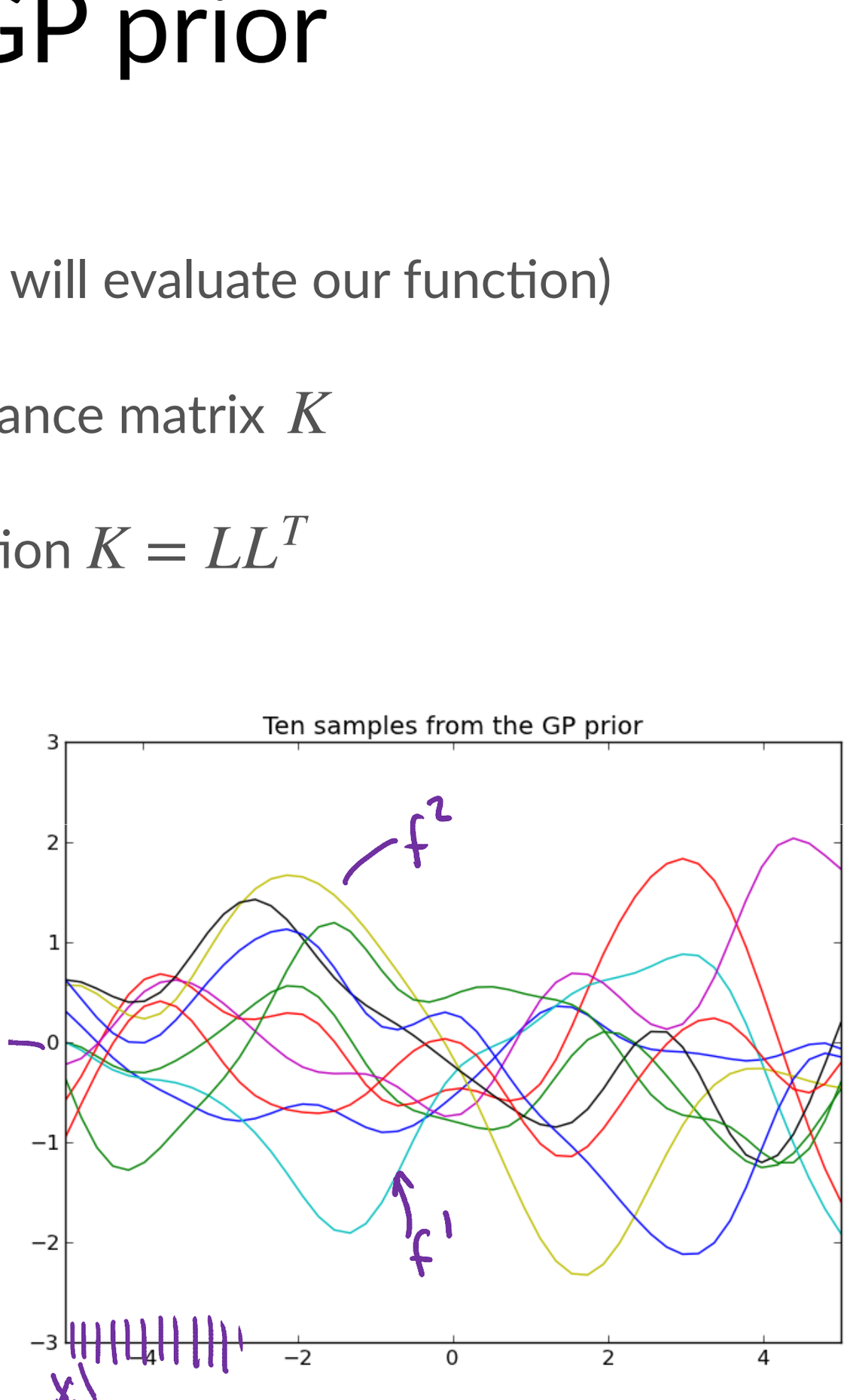
$$p(f(\mathrm{x})\,|\,\theta) = \mathcal{GP}(0,\mathrm{K}(\mathrm{x},\mathrm{x}'))$$
$$\mathrm{K}(\mathrm{x},\mathrm{x}') = \sigma^2 \exp\left(-\frac{1}{2l^2}(\mathrm{x}-\mathrm{x}')^2\right)$$ (smoothly wiggling functions expected)

since the sum of two Gaussians is a Gaussian, the model induces a GP over $y(x)$:

$$p(\mathbf{y}(\mathbf{x})\,|\,\theta) = \mathcal{GP}\left(0,\mathrm{K}(\mathrm{x},\mathrm{x}') + \mathrm{I}\sigma_{\mathrm{y}}^2\right)$$

Sampling from the GP prior

1. Create $x_{1:N}$ (the N points where we will evaluate our function)
2. Compute mean $\mu = 0_N$ and covariance matrix K
3. Compute the Cholesky decomposition $K = LL^T$
4. $$f^l \sim \mathcal{N}(\mu, K)$$
$$\sim L\mathcal{N}(0, I)$$



Sampling from the GP prior

```
from __future__ import division
import numpy as np
import matplotlib.pyplot as pl

def kernel(a, b):
    """ GP squared exponential kernel """
    sqdist = np.sum(a**2,1).reshape(-1,1) + np.sum(b**2,1) - 2*np.dot(a, b.T)
    return np.exp(-.5 * sqdist)

n = 50                                          # number of test points.
Xtest = np.linspace(-5, 5, n).reshape(-1,1)     # Test points.
K_ = kernel(Xtest, Xtest)                       # Kernel at test points.

# draw samples from the prior at our test points.
L = np.linalg.cholesky(K_ + 1e-6*np.eye(n))
f_prior = np.dot(L, np.random.normal(size=(n,10)))

pl.plot(Xtest, f_prior)
```

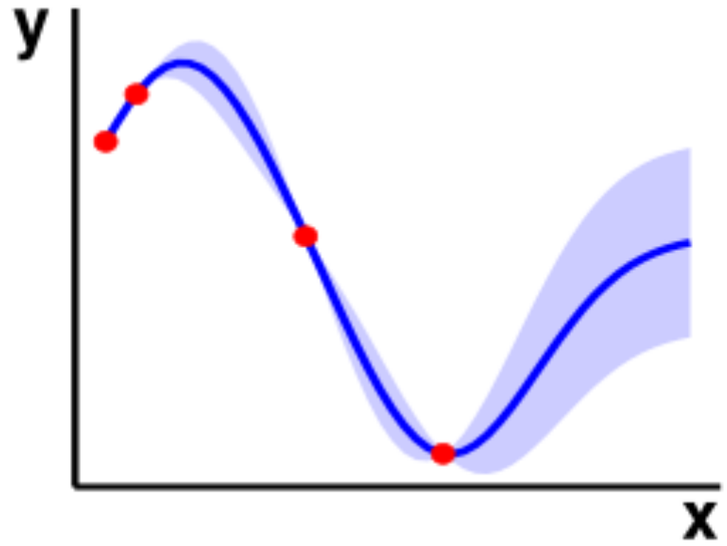
Gaussian Processes for regression

Gaussian Processes for regression

Reminder: the conditioning equations for multivariate Gaussian distributions

$$p\left(\mathbf{y}_1, \mathbf{y}_2\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^\top & \mathbf{C}\end{bmatrix}\right)$$

$$p\left(\mathbf{y}_1 \,|\, \mathbf{y}_2\right) = \frac{p\left(\mathbf{y}_1, \mathbf{y}_2\right)}{p\left(\mathbf{y}_2\right)}$$

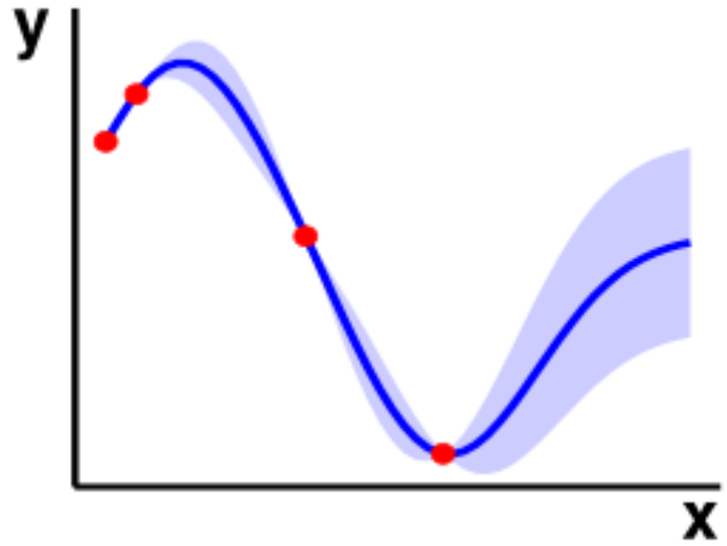


Gaussian Processes for regression

Reminder: the conditioning equations for multivariate Gaussian distributions

$$p\left(\mathbf{y}_1, \mathbf{y}_2\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^\top & \mathbf{C}\end{bmatrix}\right)$$

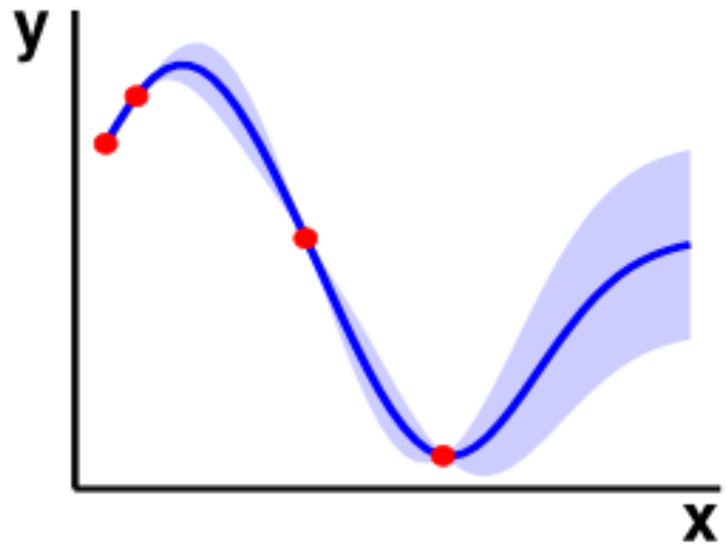
$$p\left(\mathbf{y}_1 \,|\, \mathbf{y}_2\right) = \frac{p\left(\mathbf{y}_1, \mathbf{y}_2\right)}{p\left(\mathbf{y}_2\right)} \qquad p\left(\mathbf{y}_2\right) = \mathcal{N}(\mathbf{b}, \mathbf{C})$$



Gaussian Processes for regression

Reminder: the conditioning equations for multivariate Gaussian distributions

$$p\left(\mathbf{y}_1, \mathbf{y}_2\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^\top & \mathbf{C}\end{bmatrix}\right)$$



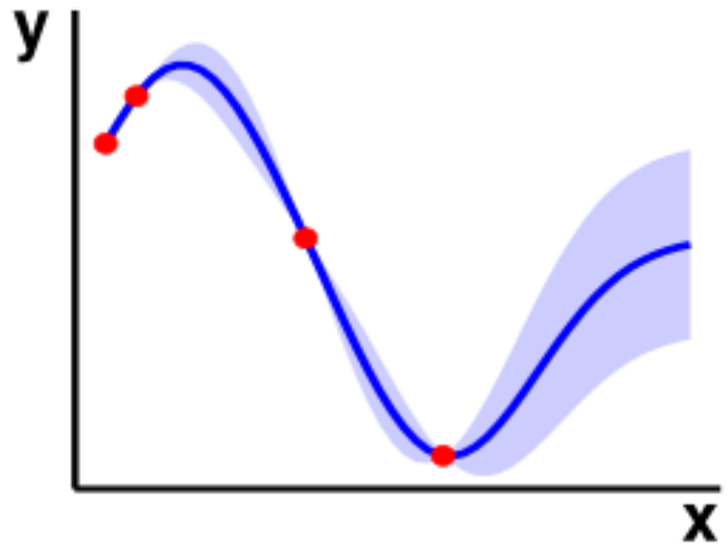
$$p\left(\mathbf{y}_1 \mid \mathbf{y}_2\right) = \frac{p\left(\mathbf{y}_1, \mathbf{y}_2\right)}{p\left(\mathbf{y}_2\right)} \qquad p\left(\mathbf{y}_2\right) = \mathcal{N}(\mathbf{b}, \mathrm{C})$$

$$p\left(\mathbf{y}_1 \mid \mathbf{y}_2\right) = \mathcal{N}\left(\mathbf{a} + \mathrm{B}\mathrm{C}^{-1}\left(\mathbf{y}_2 - \mathbf{b}\right), \mathbf{A} - \mathrm{B}\mathrm{C}^{-1}\mathrm{B}^{\mathrm{T}}\right)$$

Gaussian Processes for regression

Reminder: the conditioning equations for multivariate Gaussian distributions

$$p\left(\mathbf{y}_1, \mathbf{y}_2\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^\top & \mathbf{C}\end{bmatrix}\right)$$



$$p\left(\mathbf{y}_1 \mid \mathbf{y}_2\right) = \frac{p\left(\mathbf{y}_1, \mathbf{y}_2\right)}{p\left(\mathbf{y}_2\right)} \qquad p\left(\mathbf{y}_2\right) = \mathcal{N}(\mathbf{b}, \mathrm{C})$$

$$p\left(\mathbf{y}_1 \mid \mathbf{y}_2\right) = \mathcal{N}\left(\mathbf{a} + \mathrm{BC}^{-1}\left(\mathbf{y}_2 - \mathbf{b}\right), \mathbf{A} - \mathrm{BC}^{-1}\mathrm{B}^{\mathrm{T}}\right)$$

predictive mean: $\mu_{\mathbf{y}_1|\mathbf{y}_2} = \mathbf{a} + \mathbf{BC}^{-1}\left(\mathbf{y}_2 - \mathbf{b}\right)$

predictive covariance: $\Sigma_{\mathbf{y}_1|\mathbf{y}_2} = \mathrm{A} - \mathrm{BC}^{-1}\mathrm{B}^{\mathrm{T}}$

Gaussian Processes for regression: noiseless

What are the means and variances of the values f^* at points X*, given observed values f at points X.

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}(X) \\ \boldsymbol{\mu}(X_*) \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix} \right)$$

Conditioning:

$$p\left(\mathbf{f}_* \,|\, \mathbf{X}_*, \mathbf{X}, \mathbf{f}\right) = \mathcal{N}\left(\mathbf{f}_* \,|\, \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\right)$$

$$\boldsymbol{\mu}_* = \boldsymbol{\mu}\left(\mathbf{X}_*\right) + \mathbf{K}_*^T \mathbf{K}^{-1}(\mathbf{f} - \boldsymbol{\mu}(\mathbf{X}))$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*$$

Gaussian Processes for regression: noisy

What are the means and variances of the values f_* at points X_*,
given (noisy) observed values y at points X.

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}(X) \\ \boldsymbol{\mu}(X_*) \end{pmatrix}, \begin{pmatrix} \mathbf{K}_\mathbf{y} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix} \right) \qquad \mathbf{K}_y = \mathbf{K} + \sigma_y^2 \mathbf{I}_N$$

Conditioning:

$$p\left(\mathbf{f}_* \,|\, \mathbf{X}_*, \mathbf{X}, \mathbf{y}\right) = \mathcal{N}\left(\mathbf{f}_* \,|\, \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\right)$$

$$\boldsymbol{\mu}_* = \boldsymbol{\mu}\left(\mathbf{X}_*\right) + \mathbf{K}_*^T \mathbf{K}_\mathbf{y}^{-1} (y - \boldsymbol{\mu}(\mathbf{X}))$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_\mathbf{y}^{-1} \mathbf{K}_*$$

Gaussian Processes for regression: noisy

What is the mean and variance of the value f_* at single point x_*,
given (noisy) observed values y at points X.

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}(X) \\ \boldsymbol{\mu}(x_*) \end{pmatrix}, \begin{pmatrix} \mathbf{K_y} & \mathbf{k}_* \\ \mathbf{k}_*^T & \mathbf{k}_{**} \end{pmatrix} \right) \qquad \mathbf{K}_y = \mathbf{K} + \sigma_y^2 \mathbf{I}_N$$

Conditioning:

$$\mathbf{k}_{**} = K(\mathbf{x}, \mathbf{x}) + \sigma_y^2 \quad \text{scalar!}$$

$$p\left(\mathbf{f}_* \,|\, \mathbf{x}_*, \mathbf{X}, \mathbf{y}\right) = \mathcal{N}\left(\mathbf{f}_* \,|\, \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\right)$$

$$\boldsymbol{\mu}_* = \boldsymbol{\mu}\left(\mathbf{x}_*\right) + \mathbf{k}_*^T \mathbf{K_y}^{-1} (y - \boldsymbol{\mu}(\mathbf{X}))$$

$$\boldsymbol{\sigma}_*^2 = \mathbf{k}_{**} - \mathbf{k}_*^T \mathbf{K_y}^{-1} \mathbf{k}_*$$

Computational Cost

prediction task

- train on N points
- test on M points

prediction equations

$$\mu_M = \mathrm{K}_{MN}\mathrm{K}_{NN}^{-1}\mathrm{y}_N$$

$$\Sigma_{MM} = \mathrm{K}_{MM} - \mathrm{K}_{MN}\mathrm{K}_{NN}^{-1}\mathrm{K}_{NM}$$

Without special structure, computation is limited to $\mathcal{O}(1000)$ variables

Computational cost is a major limitation of GPs

Numerical Computations Considerations

For GP $\mu(x) = 0$

$$p\left(\mathbf{f}_* \,|\, \mathbf{x}_*, \mathbf{X}, \mathbf{y}\right) = \mathcal{N}\left(\mathbf{f}_* \,|\, \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\right)$$

$$\boldsymbol{\mu}_* = \mathbf{k}_*^T \mathbf{K_y}^{-1} y$$

$$\boldsymbol{\sigma}_*^2 = \mathbf{k}_{**} - \mathbf{k}_*^T \mathbf{K_y}^{-1} \mathbf{k}_*$$

$$\mathbf{k}_{**} = K(\mathbf{x}, \mathbf{x}) + \sigma_y^2$$

$$\mathbf{K}_y = \mathbf{L}\mathbf{L}^T$$

$$\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\mathbf{y} = \mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{y}$$

Algorithm: GP Regression

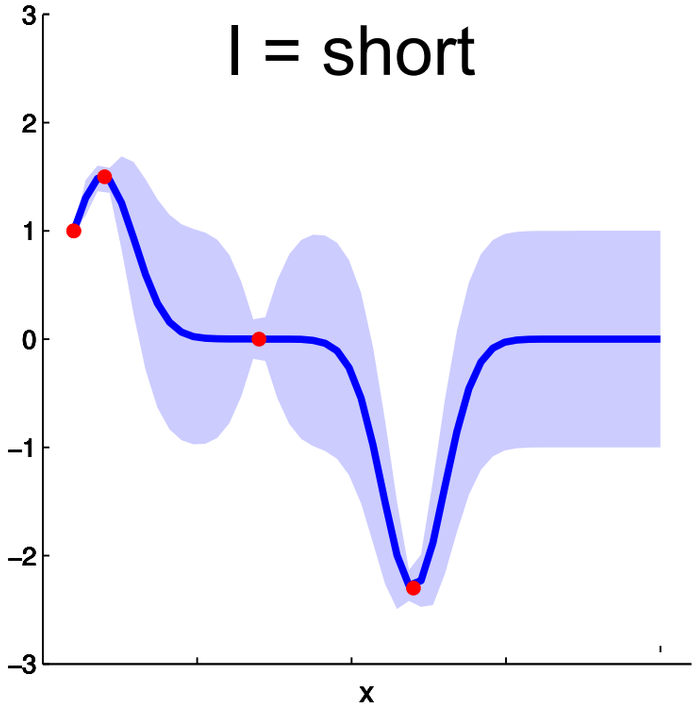
1. $\mathbf{L} =$ cholesky $\left(\mathbf{K}_y\right)$
2. $\boldsymbol{\alpha} = \mathbf{L}^T \backslash \left(\mathbf{L} \backslash \mathbf{y}\right)$
3. $\mu_* = \mathbb{E}\left[f_*\right] = \mathbf{k}_*^T \boldsymbol{\alpha}$
4. $\mathbf{v} = \mathbf{L} \backslash \mathbf{k}_*$
5. $\sigma_*^2 = \mathrm{var}\left[f_*\right] = \mathbf{k}_{**} - \mathbf{v}^T \mathbf{v}$

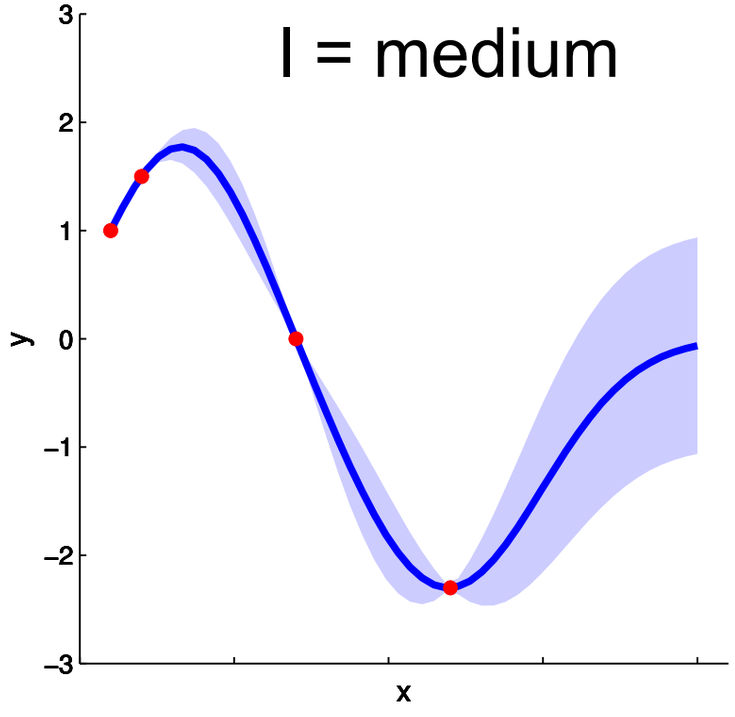
What effect do the hyper-parameters have?

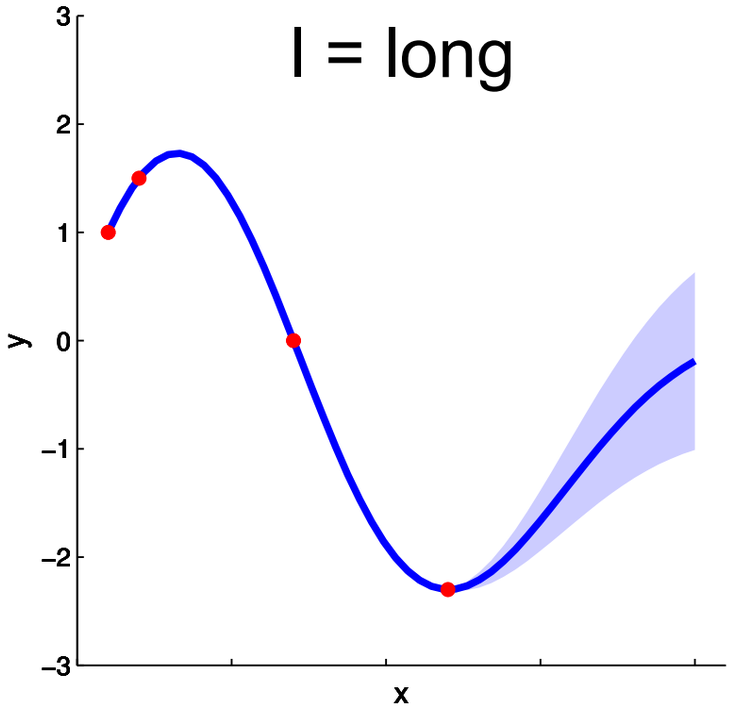
$$\mathbf{K}\left(\mathbf{x}_1, \mathbf{x}_2\right) = \sigma^2 \exp\left(-\frac{1}{2l^2}\left(\mathbf{x}_1 - \mathbf{x}_2\right)^2\right)$$

Hyper-parameters have a strong effect

- l controls the horizontal scaling
- σ^2 controls the vertical scale of the data

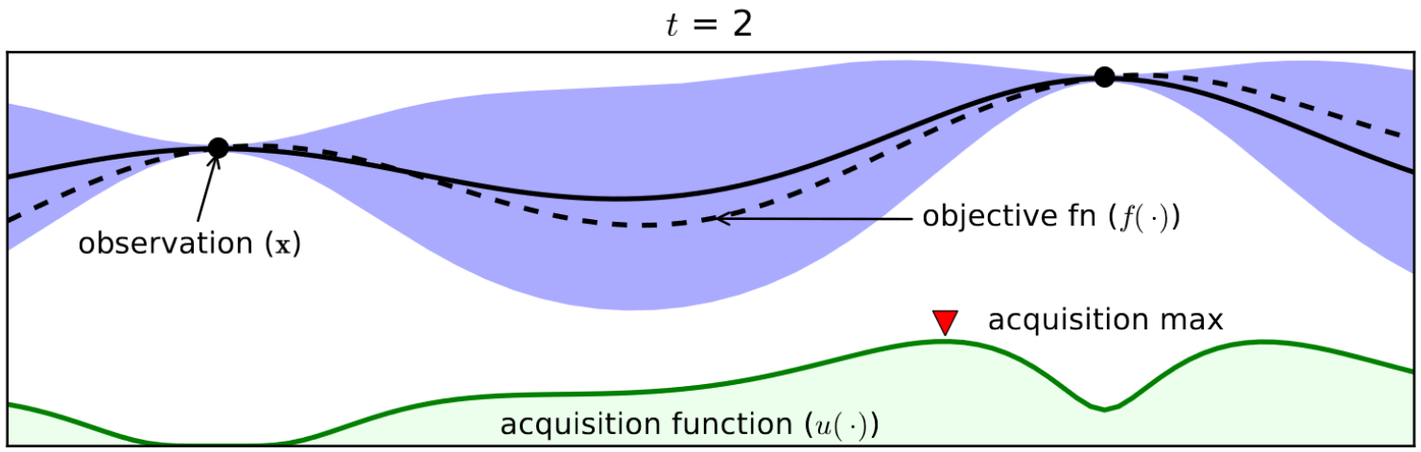






Bayesian Optimization with Gaussian processes

Bayesian Optimization with Gaussian processes

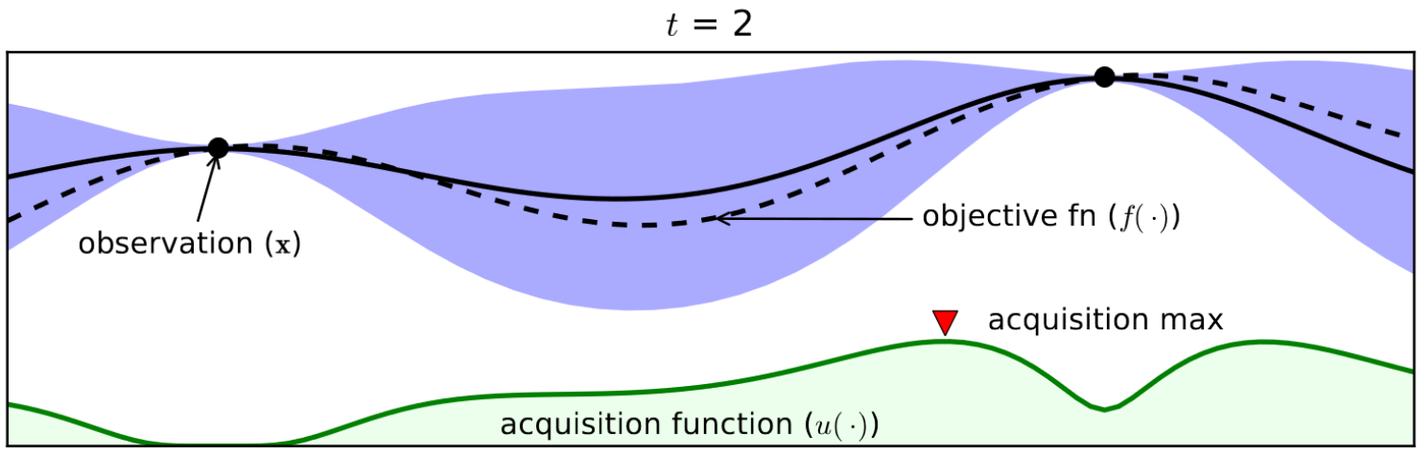


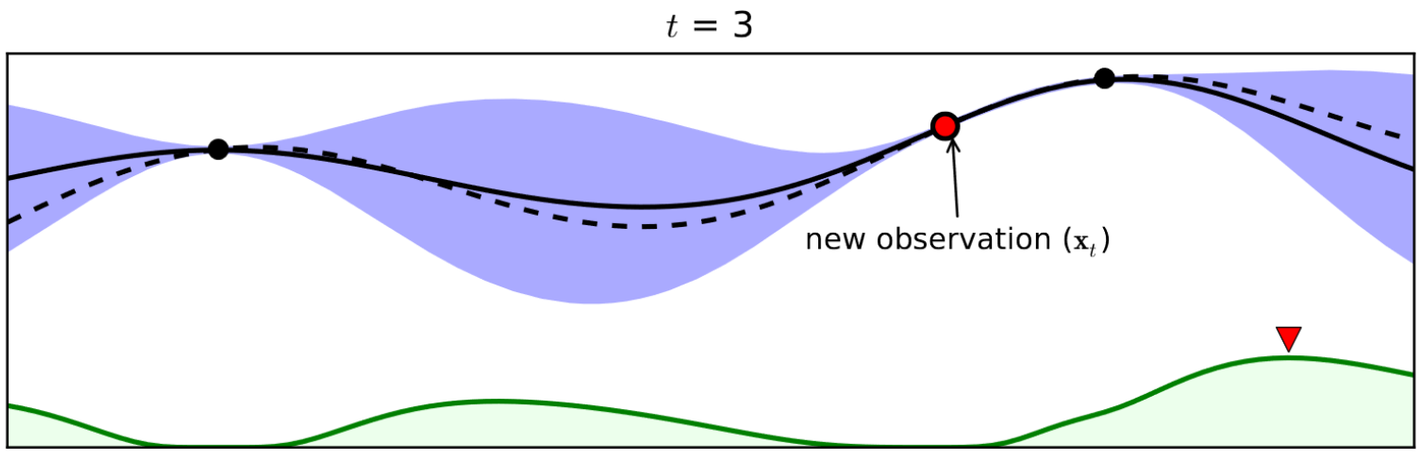
for $t = 1,2,\ldots$ **do**

1. Find $\mathbf{x}_t$ by combining attributes of the posterior distribution in a utility function u and maximizing:
   $\mathbf{x}_t = \arg\max_{\mathbf{x}} u\left(\mathbf{x} \mid \mathscr{D}_{1:t-1}\right)$
2. Sample the objective function:
   $y_t = f\left(x_t\right) + \varepsilon_t$
3. Augment the data :
   $\mathscr{D}_{1:t} = \left\{\mathscr{D}_{1:t-1}, \left(\mathrm{x}_t, y_t\right)\right\}$
   and update the GP.

end for

Bayesian Optimization with Gaussian processes



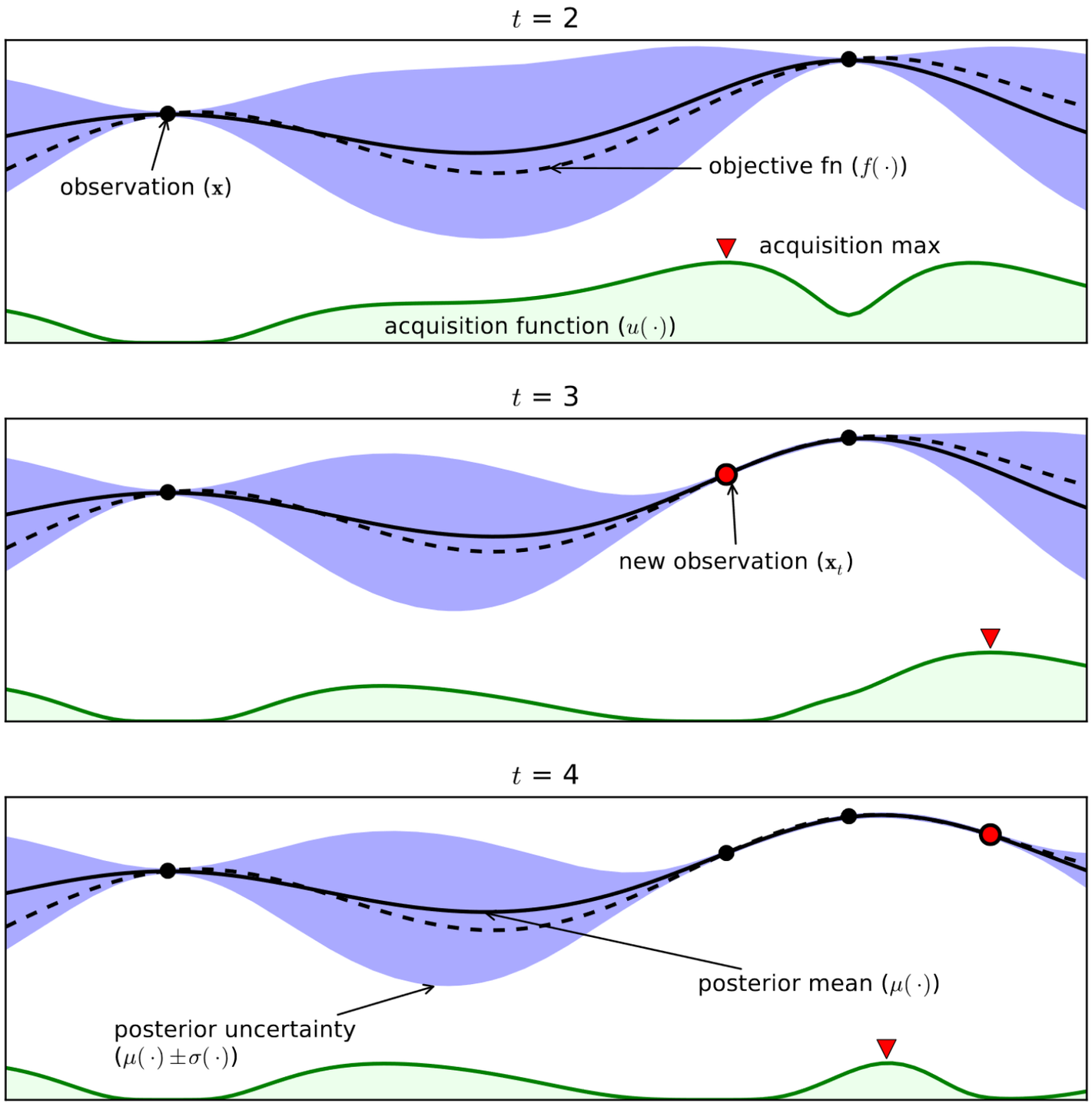


for $t = 1,2,\ldots$ **do**

1. Find $\mathbf{x}_t$ by combining attributes of the posterior distribution in a utility function u and maximising: $\mathbf{x}_t = \arg\max_{\mathbf{x}} u\left(\mathbf{x} \mid \mathscr{D}_{1:t-1}\right)$
2. Sample the objective function: $y_t = f\left(x_t\right) + \varepsilon_t$
3. Augment the data : $\mathscr{D}_{1:t} = \left\{\mathscr{D}_{1:t-1}, \left(\mathrm{x}_t, y_t\right)\right\}$ and update the GP.

end for

Bayesian Optimization with Gaussian processes



for $t = 1,2,\ldots$ **do**

1. Find $\mathbf{x}_t$ by combining attributes of the posterior distribution in a utility function u and maximising:
   $\mathbf{x}_t = \arg\max_{\mathbf{x}} u\left(\mathbf{x} \mid \mathscr{D}_{1:t-1}\right)$
2. Sample the objective function:
   $y_t = f\left(x_t\right) + \varepsilon_t$
3. Augment the data :
   $\mathscr{D}_{1:t} = \left\{\mathscr{D}_{1:t-1}, \left(\mathrm{x}_t, y_t\right)\right\}$
   and update the GP.

end for

Exploration-Exploitation Tradeoff

How should we pick the next point x to evaluate?

GP prediction in the special case of one test point x_{t+1}:

$$P\left(y_{t+1} \mid \mathscr{D}_{1:t}, \mathbf{x}_{t+1}\right) = \mathcal{N}\left(\mu_t\left(\mathbf{x}_{t+1}\right), \sigma_t^2\left(\mathbf{x}_{t+1}\right) + \sigma^2_{\text{noise}}\right)$$

$$\mu_t\left(\mathbf{x}_{t+1}\right) = \mathbf{k}^T\left[\mathbf{K} + \sigma^2_{\text{noise}}\,\mathbf{I}\right]^{-1}\mathbf{y}_{1:t}$$

$$\sigma_t^2\left(\mathbf{x}_{t+1}\right) = k\left(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}\right) - \mathbf{k}^T\left[\mathbf{K} + \sigma^2_{\text{noise}}\,\mathbf{I}\right]^{-1}\mathbf{k}$$

We should choose the next point x where the mean is high (exploitation) and the variance is high (exploration).

We can balance exploration and exploitation with an acquisition function u:

$$u(x) = \mu(x) + \kappa\sigma(x)$$

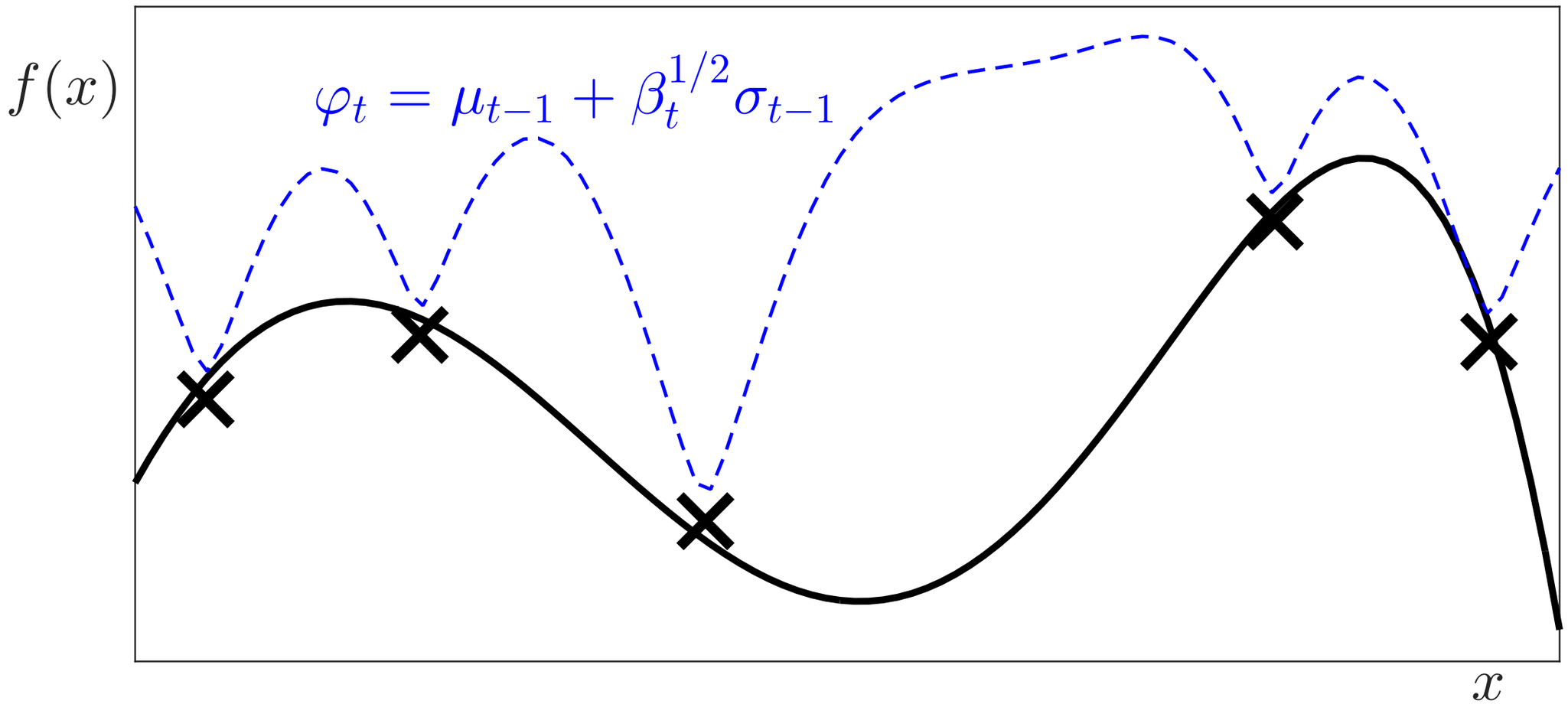
But then we need to maximize our acquisition function to pick the next point. For this we use some vanilla black box optimizer.

GP-Upper Confidence Bound (UCB)

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010)



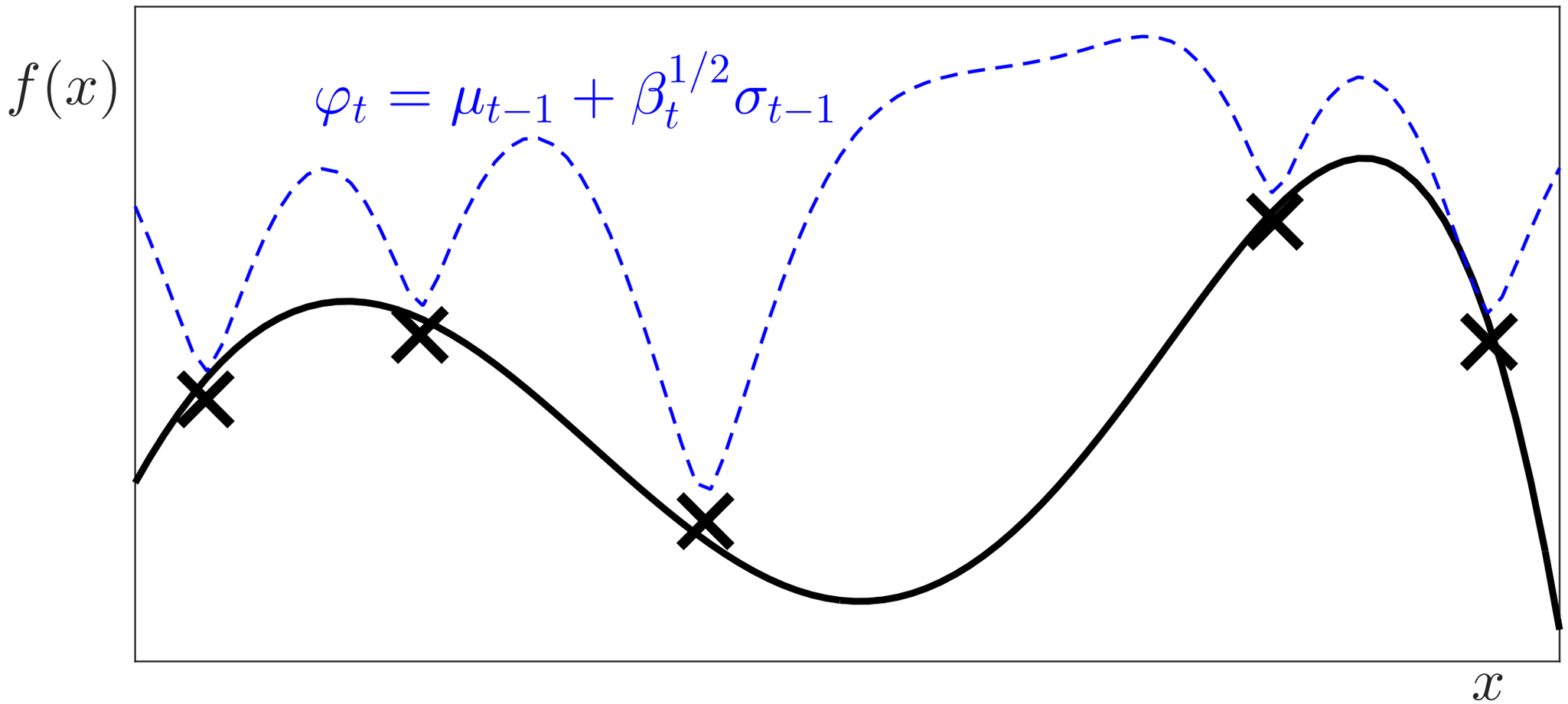
1) Compute posterior $\mathcal{GP}$

GP-Upper Confidence Bound (UCB)

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010)



1) Compute posterior $\mathcal{GP}$

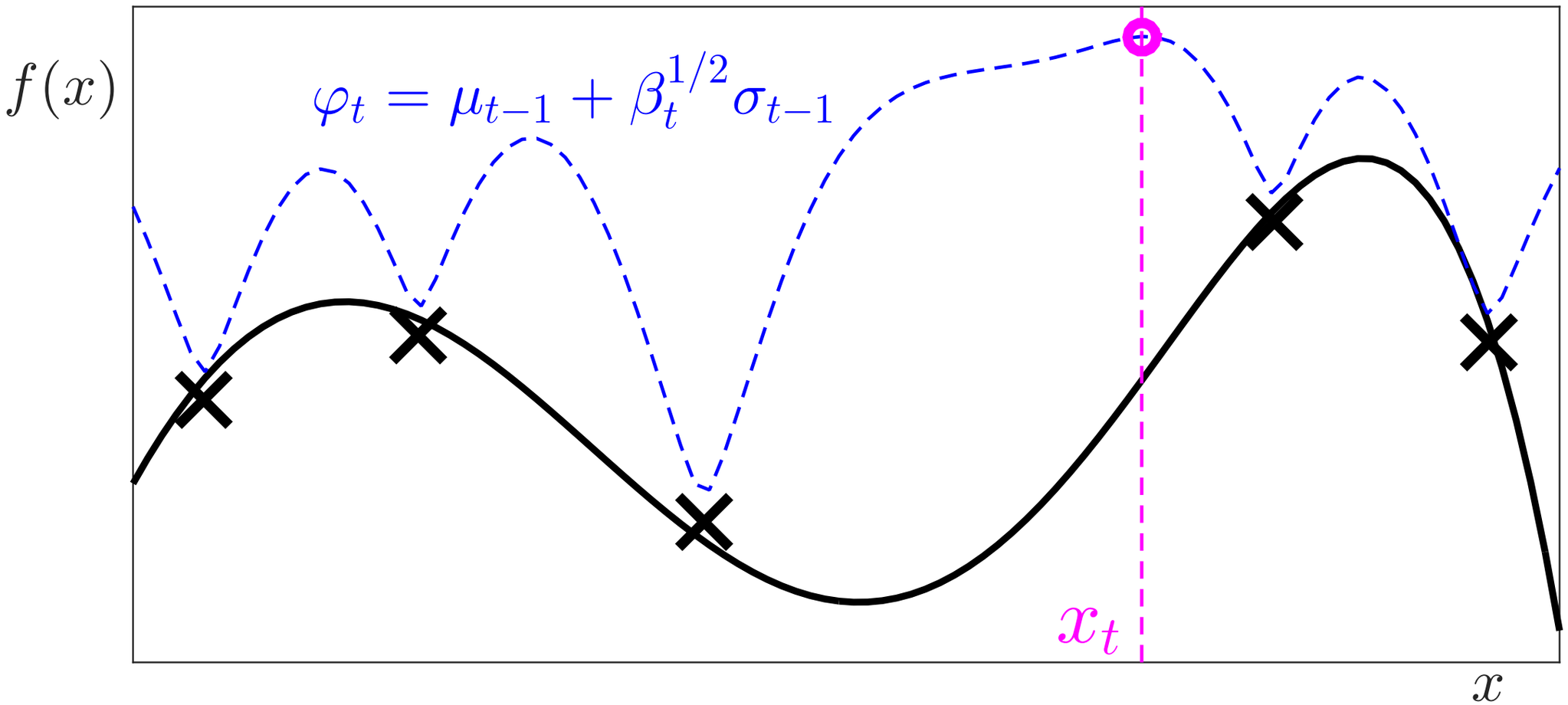
2) Construct UCB φ_t

GP-Upper Confidence Bound (UCB)

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010)



1) Compute posterior $\mathcal{GP}$

2) Construct UCB φ_t

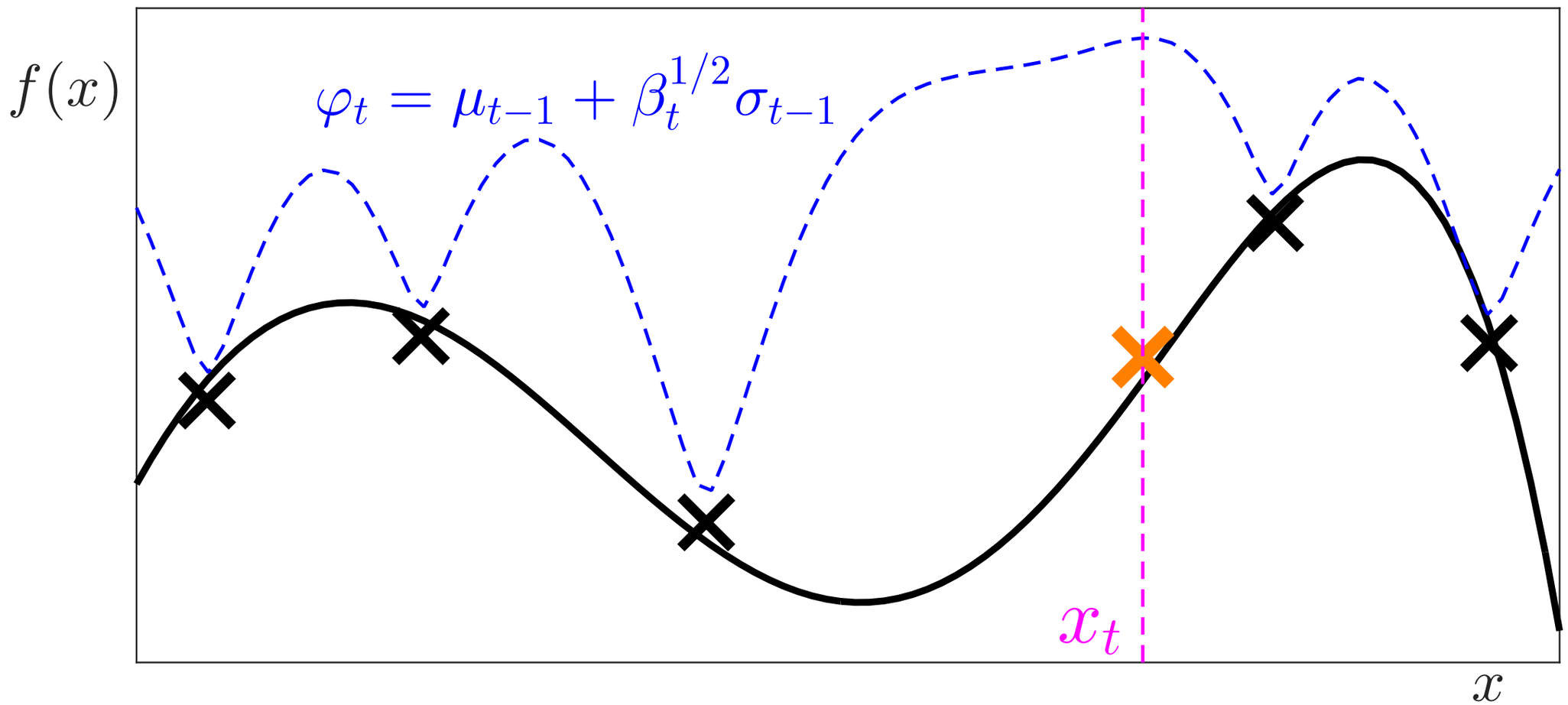
3) Choose $x_t = \underset{x \in D \subset (0,r)^d}{\operatorname{argmax}} \ \phi_t$

GP-Upper Confidence Bound (UCB)

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010)



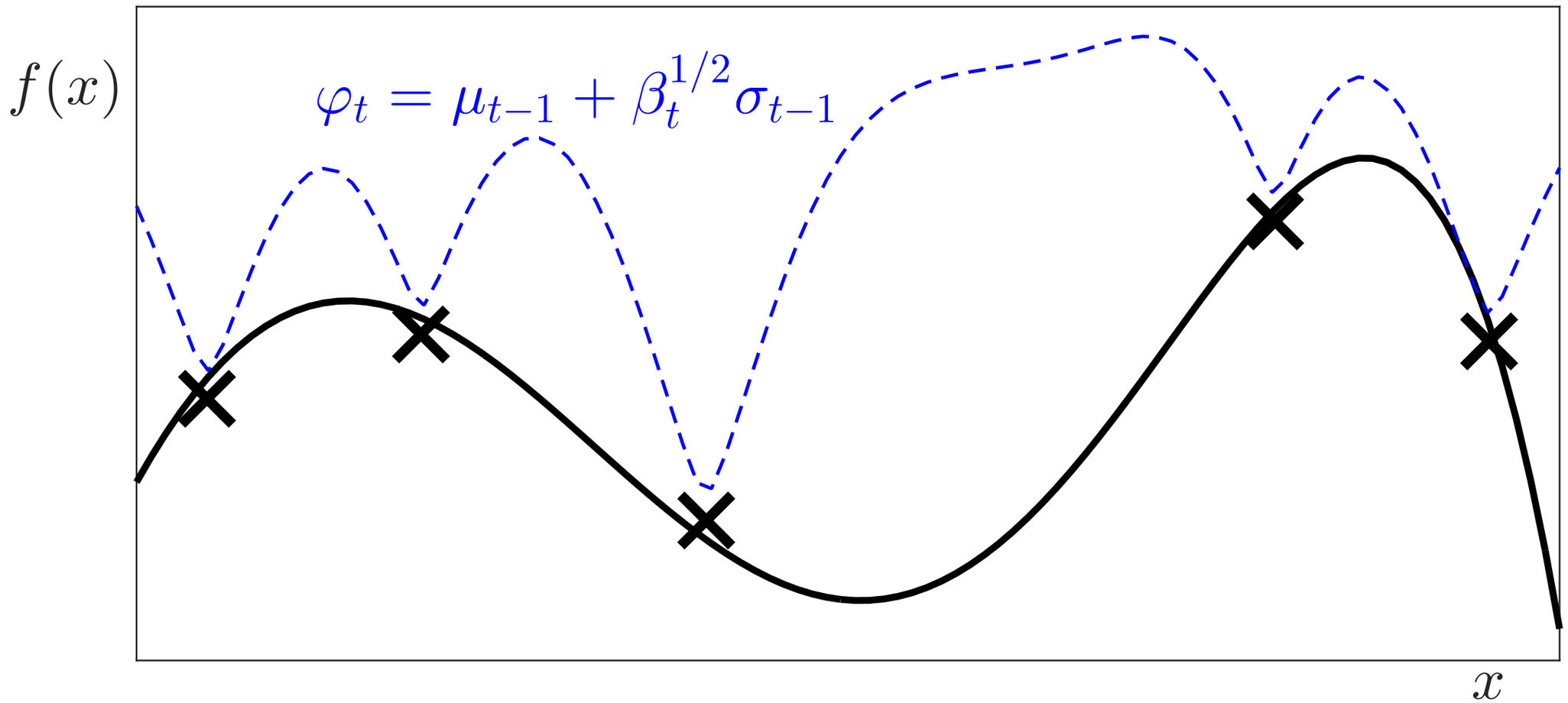
1) Compute posterior $\mathcal{GP}$
2) Construct UCB φ_t
3) Choose $x_t = \underset{x \in D \subset (0,r)^d}{\operatorname{argmax}} \ \phi_t$
4) Evaluate f at x_t

GP-Upper Confidence Bound (UCB)

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010)



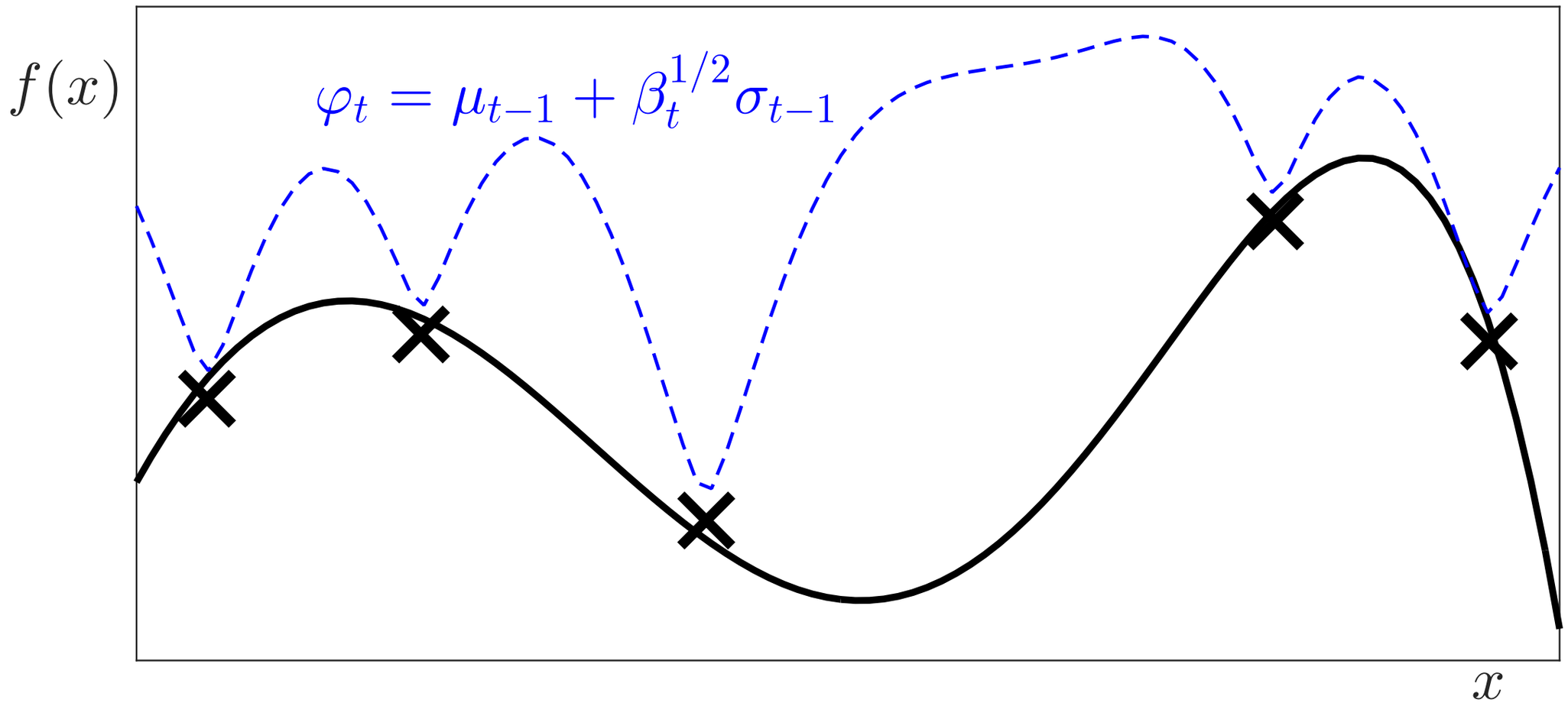
1) Compute posterior $\mathcal{GP}$

GP-Upper Confidence Bound (UCB)

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010)



1) Compute posterior $\mathcal{GP}$

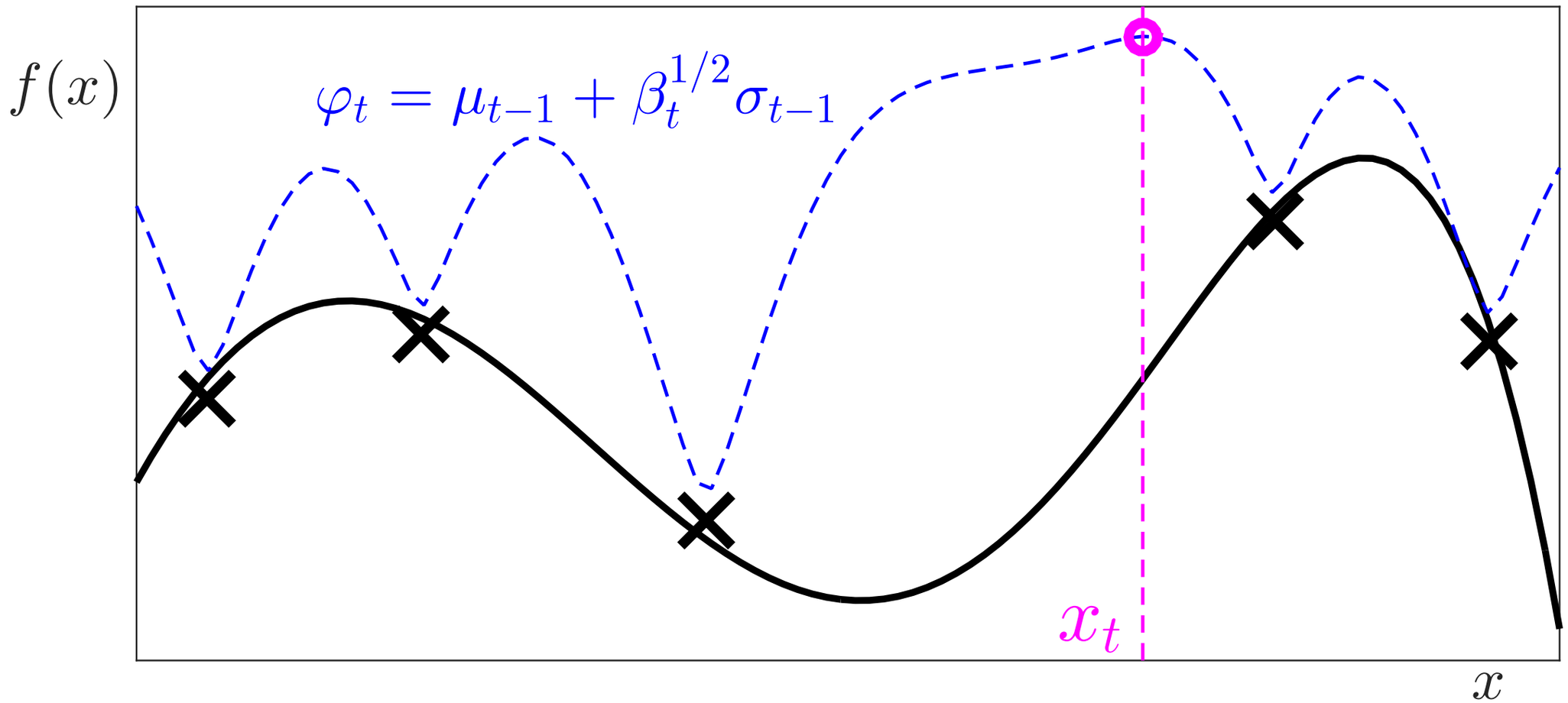
2) Construct UCB φ_t

GP-Upper Confidence Bound (UCB)

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010)



1) Compute posterior $\mathcal{GP}$

2) Construct UCB φ_t

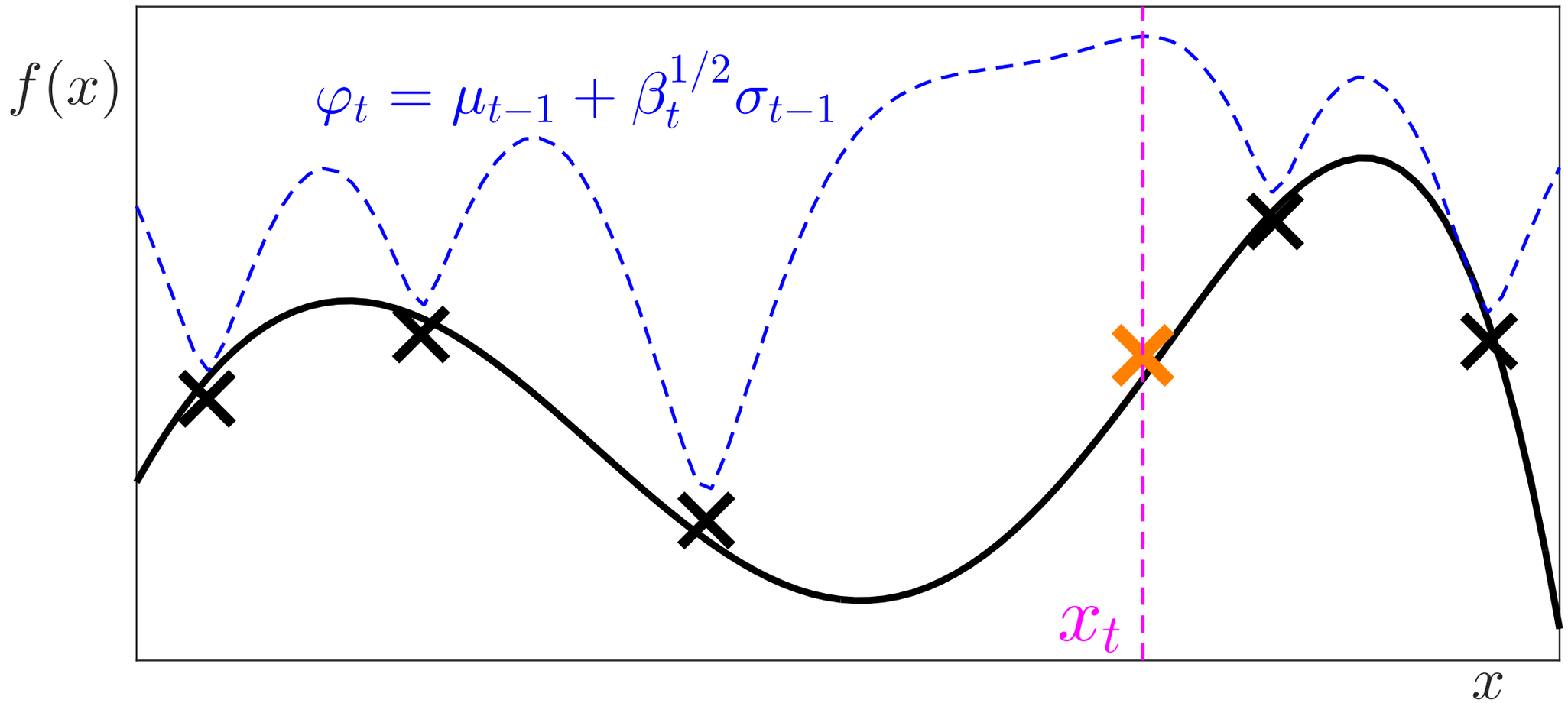
3) Choose $x_t = \underset{x \in D \subset (0,r)^d}{\operatorname{argmax}} \ \phi_t$

GP-Upper Confidence Bound (UCB)

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$.

Gaussian Process Upper Confidence Bound (GP-UCB)

(Srinivas et al. 2010)



1) Compute posterior $\mathcal{GP}$

2) Construct UCB φ_t

3) Choose $x_t = \underset{x \in D \subset (0,r)^d}{\operatorname{argmax}} \ \phi_t$

4) Evaluate f at x_t

GP-Upper Confidence Bound (UCB)

Algorithm 1 GP-UCB

Require: k

1: $\mu \leftarrow 0_d$
2: **for** $t \leftarrow 1$ to T **do**
3: $\beta_t = 2\log(t^{\frac{d}{2}+2}\pi^2/3\delta)$
4: Choose $a_t \leftarrow arg\max_i \mu_{t-1} + \sqrt{\beta_t}\sigma_{t-1}$
5: Observe $y_t = f(\mathbf{x}_t) + \epsilon_t$
6: $\mu_t = k_{t-1}(\mathbf{x})^T(K_{t-1} + \sigma^2 I_d)^{-1} y_t$
7: $k_t = k(\mathbf{x}, \mathbf{x}') - k_{t-1}(\mathbf{x})^T(K_{t-1} + \sigma^2 I_d)^{-1} k_{t-1}(\mathbf{x}')$
8: $\sigma_t^2 = k_t(\mathbf{x}, \mathbf{x})$
9: **end for**

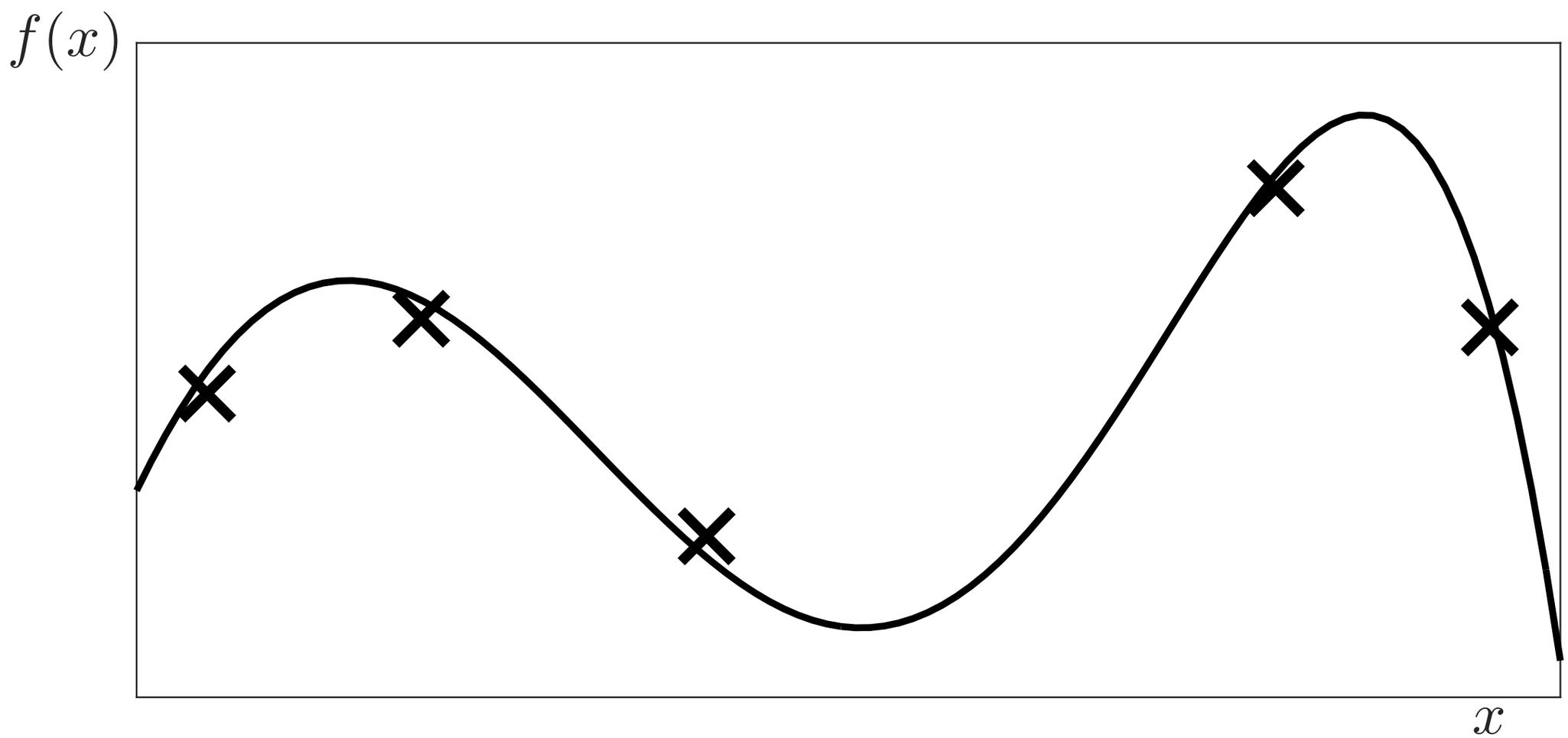
d: the dimensionality of the objective,

δ: the probability that $f(x)$ is bounded above by $\mu_t+\beta_t\sigma_t$ and below by $\mu_t - \beta_t\sigma_t$

GP-Thompson Sampling

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$

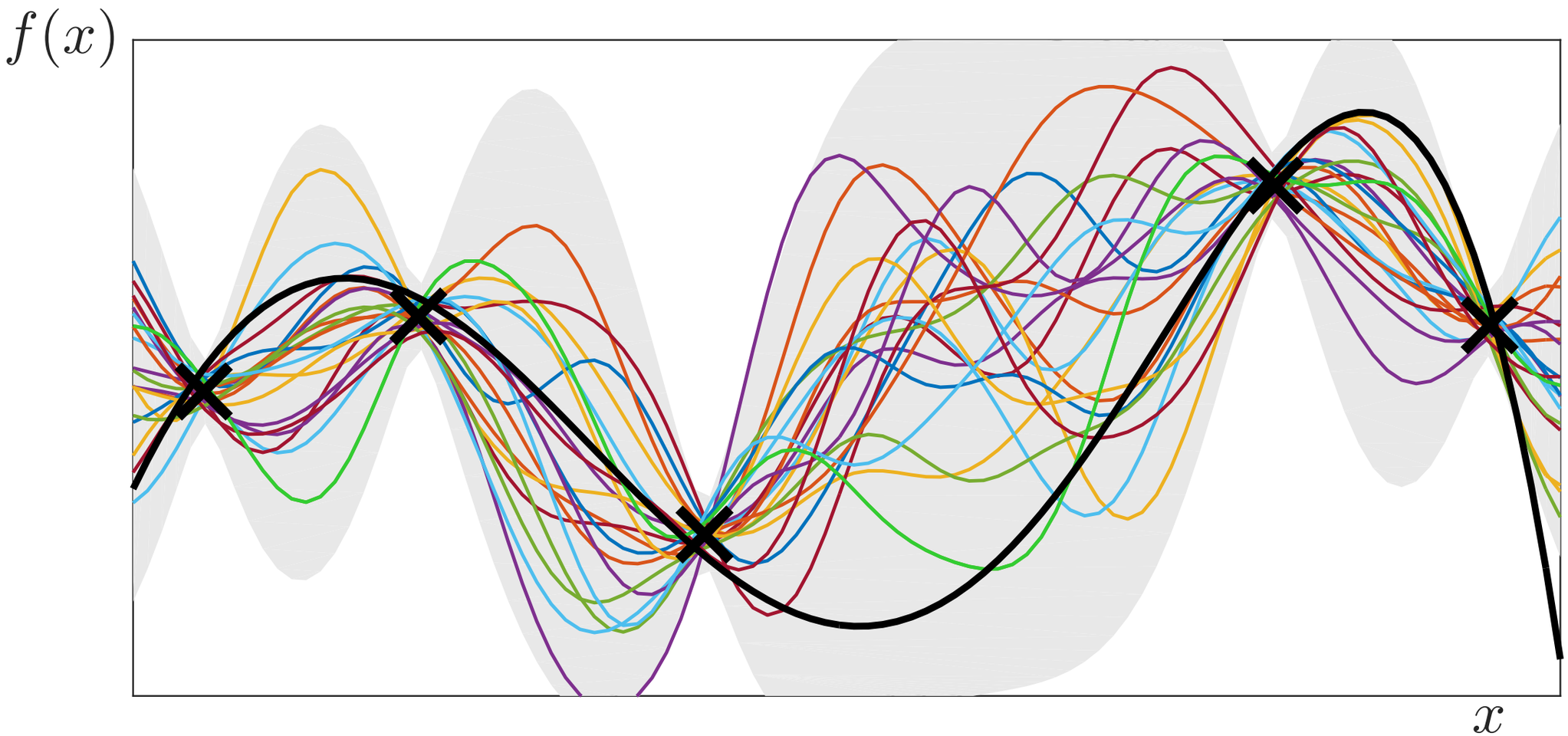
(Thompson, 1933)



GP-Thompson Sampling

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$

(Thompson, 1933)

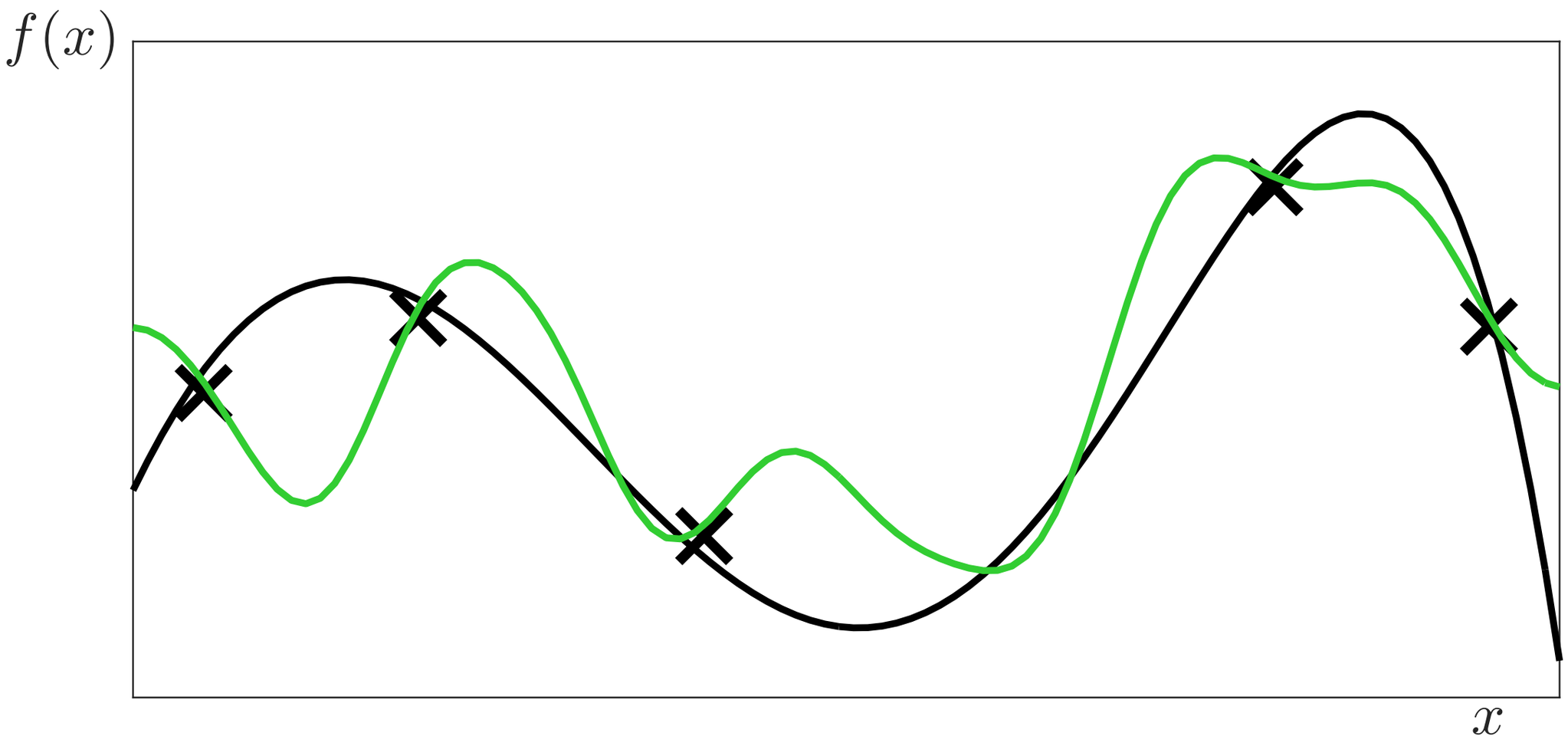


1) Construct posterior $\mathcal{GP}$

GP-Thompson Sampling

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$

(Thompson, 1933)



1) Construct posterior $\mathcal{GP}$

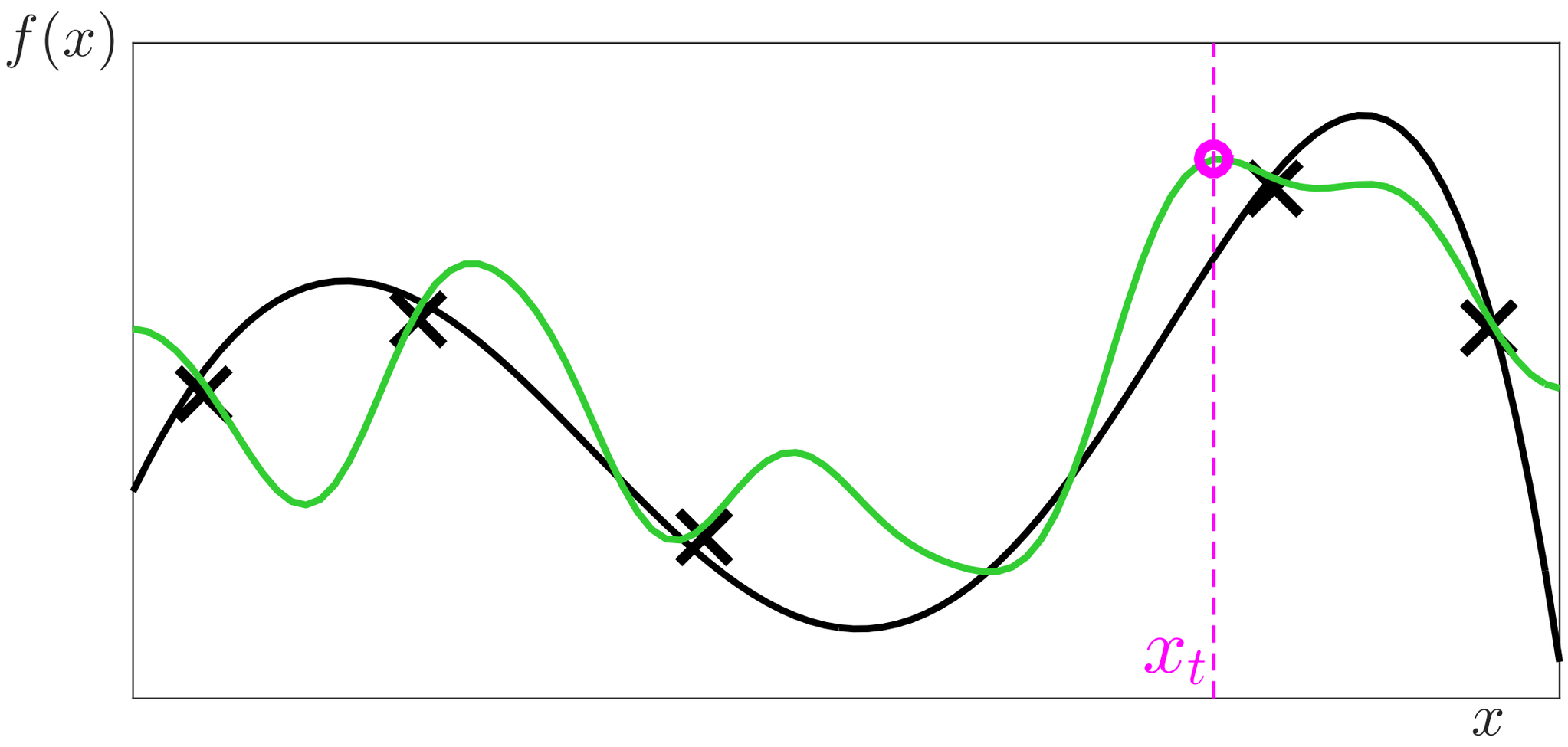
2) Draw sample g from posterior

r

GP-Thompson Sampling

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$

(Thompson, 1933)

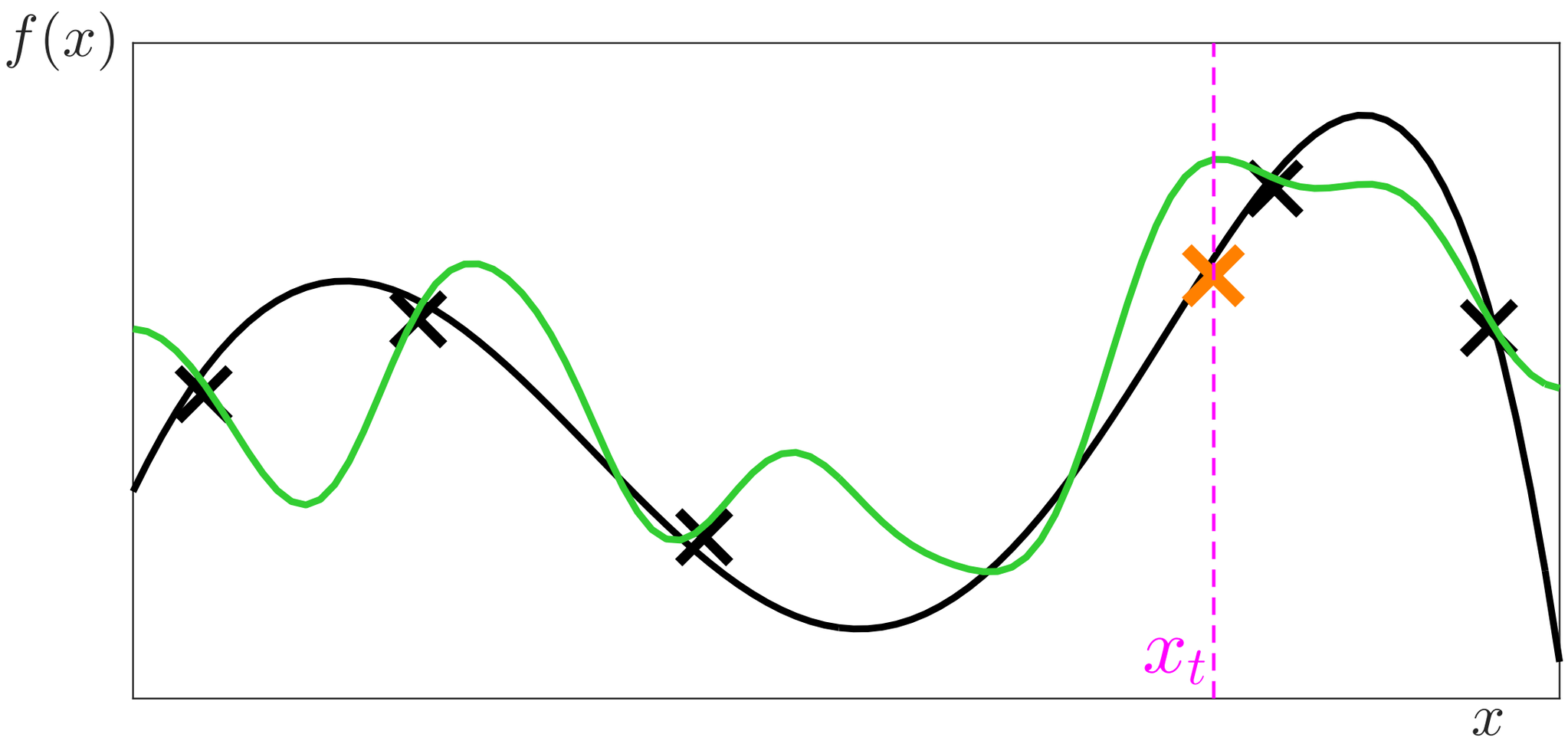


1) Construct posterior $\mathcal{GP}$
2) Draw sample g from posterior
3) Choose $x_t = \text{argmax}_x\, g(x)$

GP-Thompson Sampling

Model $f \sim \mathcal{GP}(\mathbf{0}, \kappa)$

(Thompson, 1933)



1) Construct posterior $\mathcal{GP}$
2) Draw sample g from posterior
3) Choose $x_t = \text{argmax}_x\, g(x)$
4) Evaluate f at x_t

Bayesian Optimization VS Bandits

- Both consider actions and maximization of immediate rewards
- BO has a continuous action space and bandits discrete
- Both use exploration and exploitation to select the best action to try out
- BO tries to approximate the reward function (by approximating the function to be maximized), while bandits approximate the bandit mean rewards.

Bayesian Optimization VS Evolutionary search

- Both consider maximization of a function
- BO tries to approximate the function to be maximized, while ES does not.
- BO with GPs works with small number of parameters (hundreds), while ES can scale to thousands using the tricks we discussed
- Their combination is possible, e.g., *Accelerating Evolutionary Algorithms with Gaussian Process Fitness Function Models Buche et al.*, but again low parametric

Is there a train and test phase in bandits?

No, the setup we explored was: given a set of K arms, how do we select actions to minimize our cumulative regret.

Q: what would be the learning based equivalent of the multi-armed bandit problem?

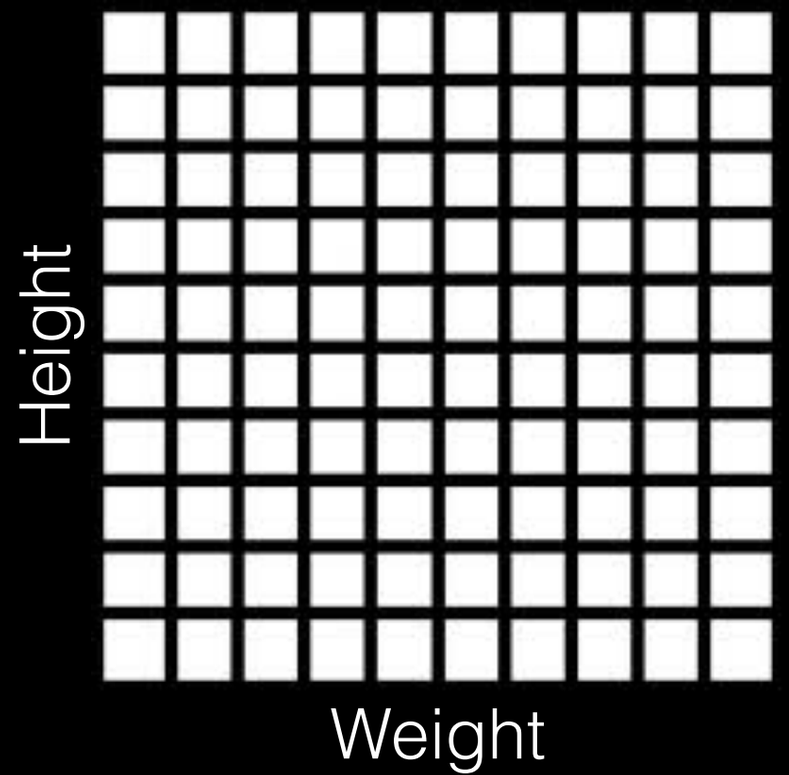
A:

- We have a training set of N multi-armed bandit instantiations.
- Each K-armed bandit is one training example.
- The agent gets n number of interactions, and obtains a final reward (-regret).
- The agent learns a policy —mapping from its set of actions taken thus far and their outcomes, to a probability over what actions to try next

We will visit this setup in the meta-learning lecture.

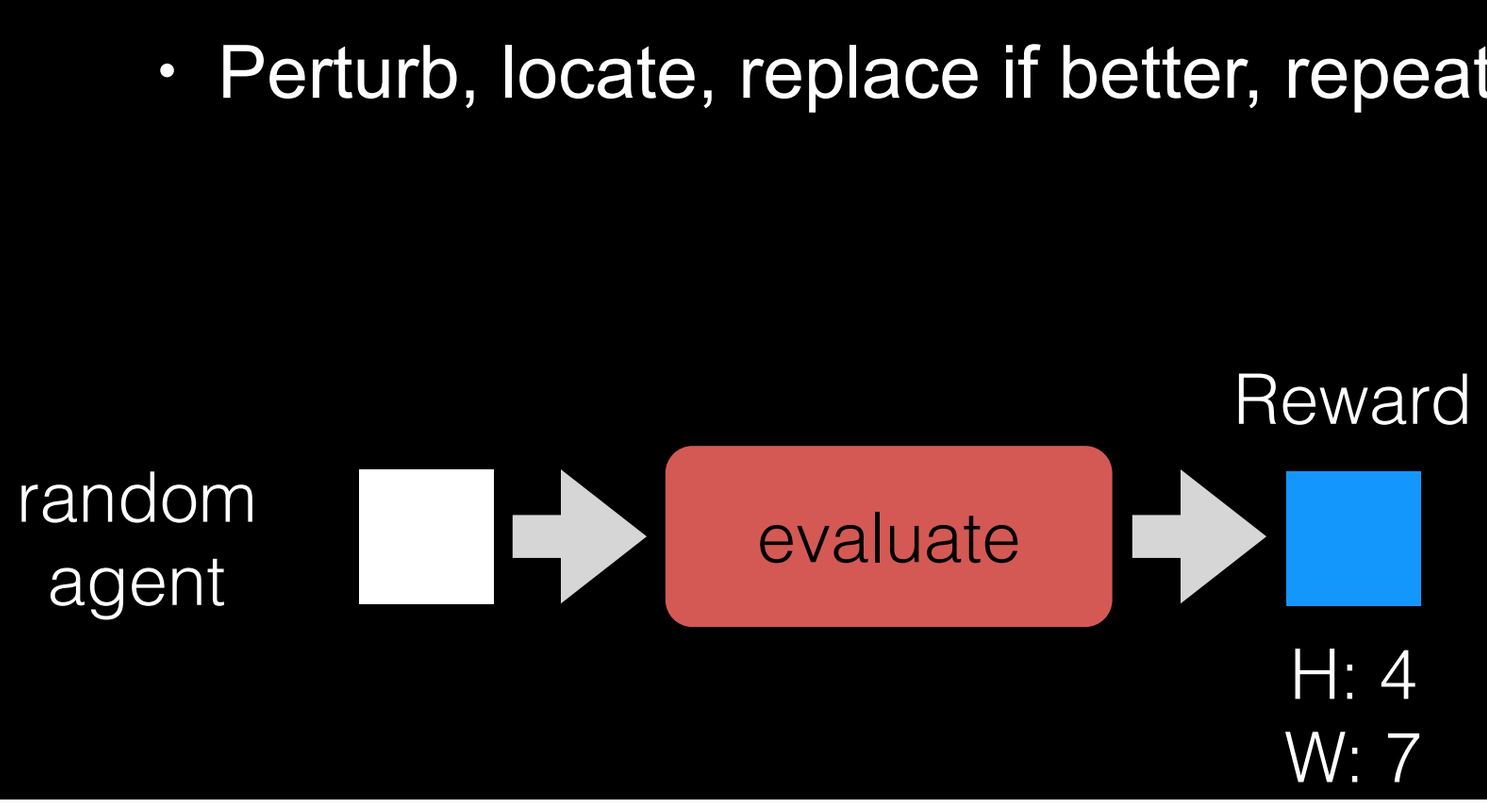
Joint learning of diverse policies with ES

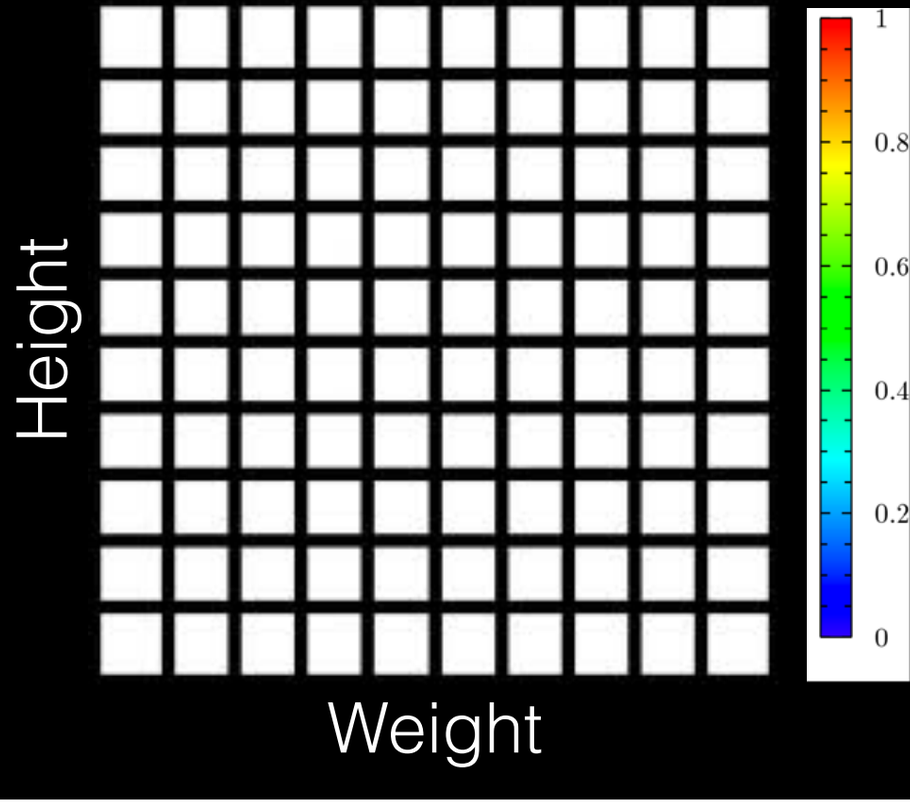
- Multi-dimensional archive of phenotypic elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Perturb, locate, replace if better, repeat



Joint learning of diverse policies with ES

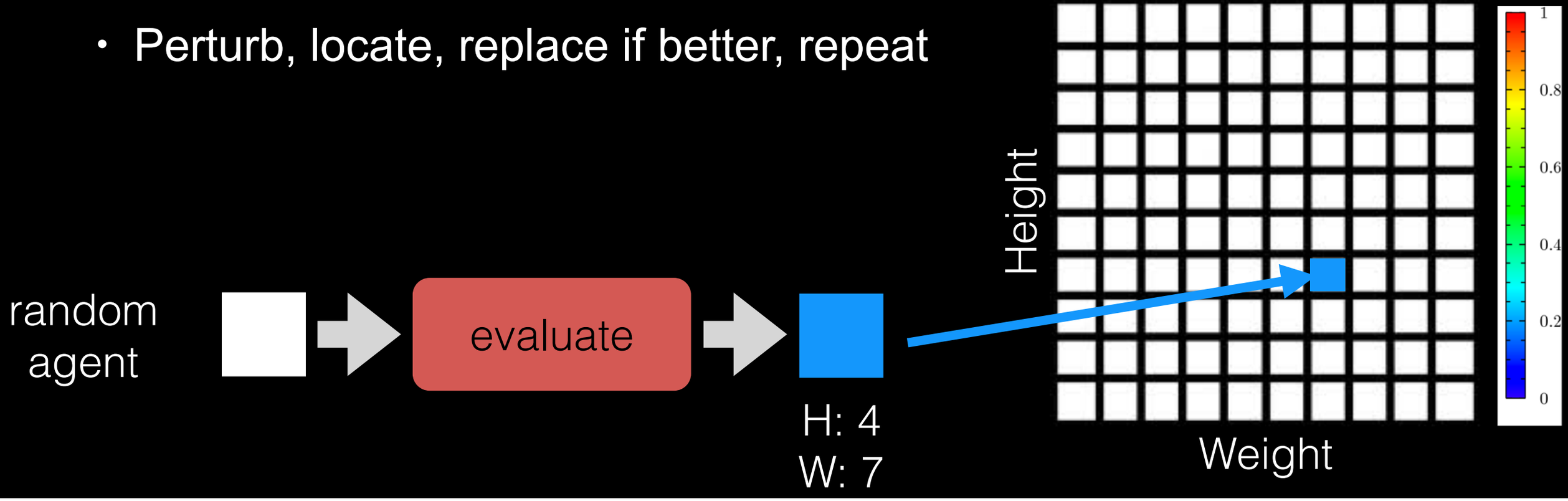
- Multi-dimensional archive of phenotypic elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Perturb, locate, replace if better, repeat





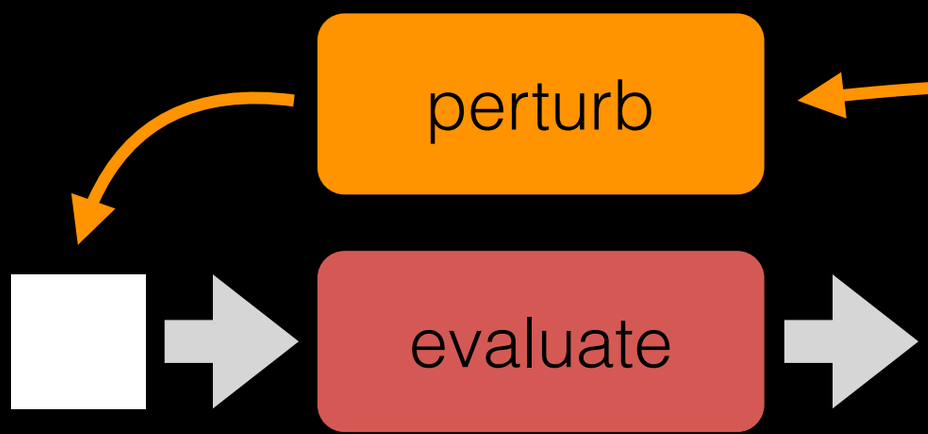
Joint learning of diverse policies with ES

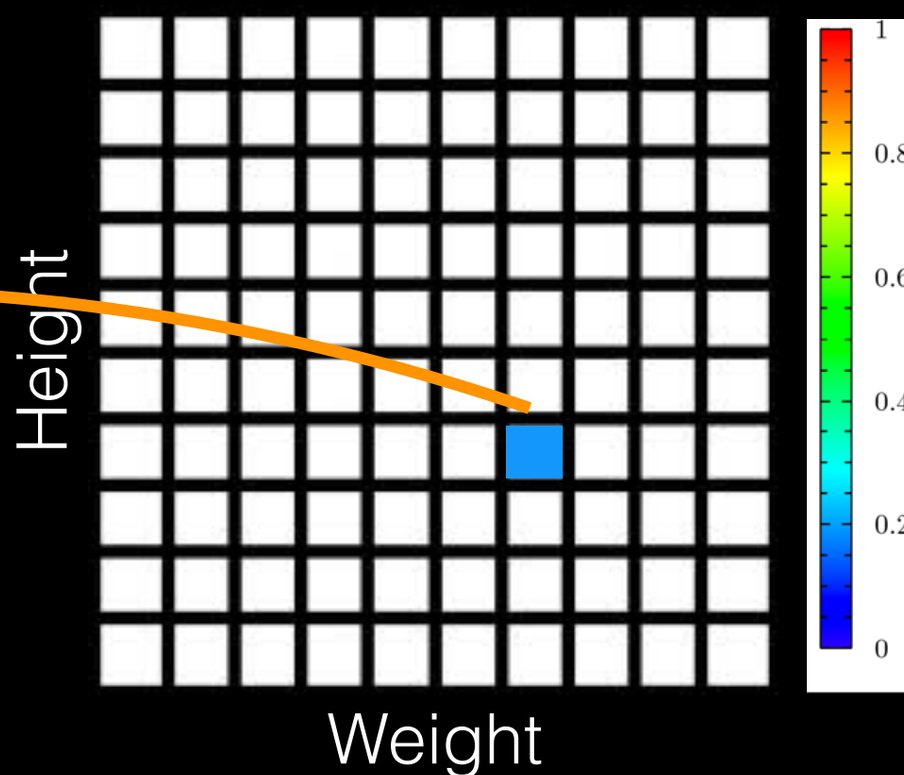
- Multi-dimensional archive of phenotypic elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Perturb, locate, replace if better, repeat



Joint learning of diverse policies with ES

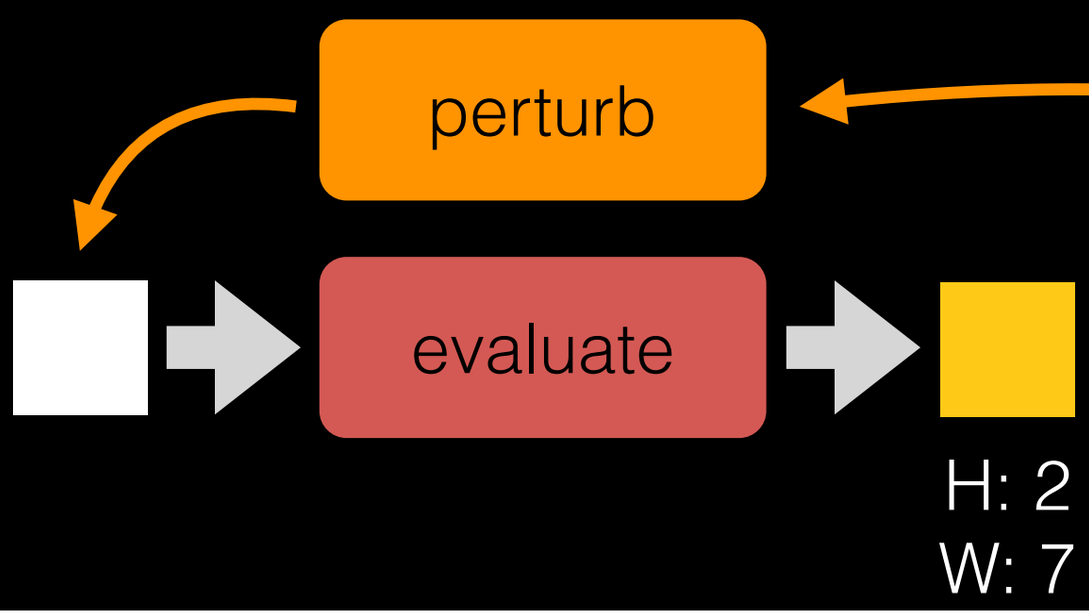
- Multi-dimensional archive of phenotypic elites
  - Choose dimensions of interest in behavior space
  - Discretize
- Perturb, locate, replace if better, repeat

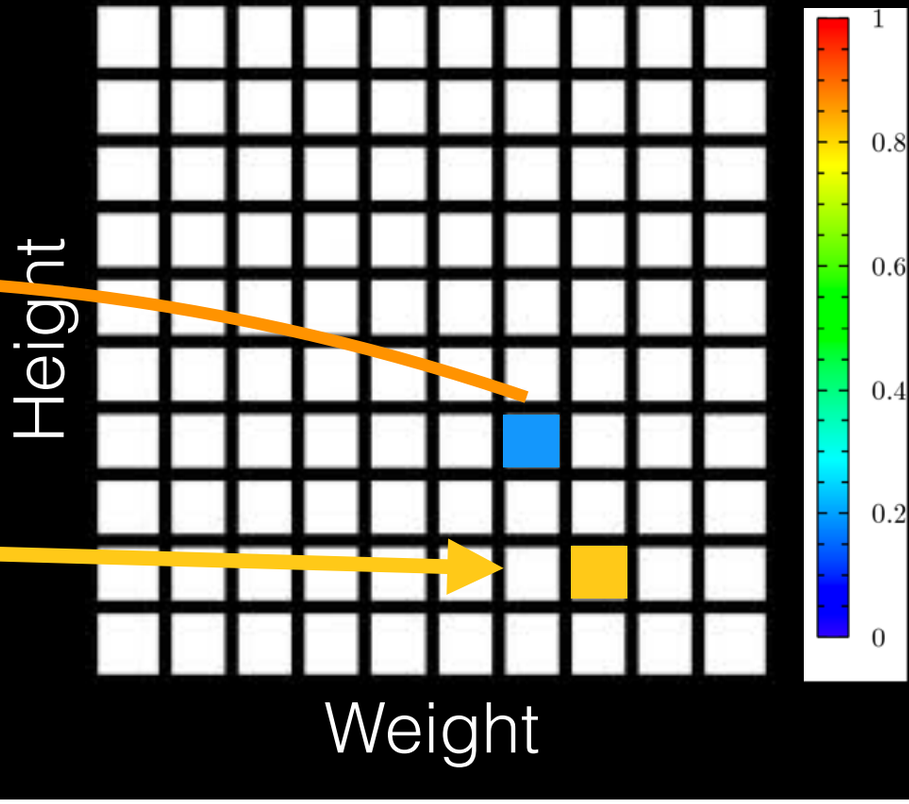




Joint learning of diverse policies with ES

- Multi-dimensional archive of phenotypic elites
  - Choose dimensions of interest in behavior space
  - Discretize
- Perturb, locate, replace if better, repeat

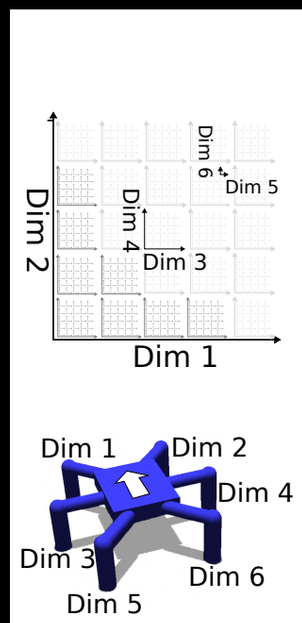


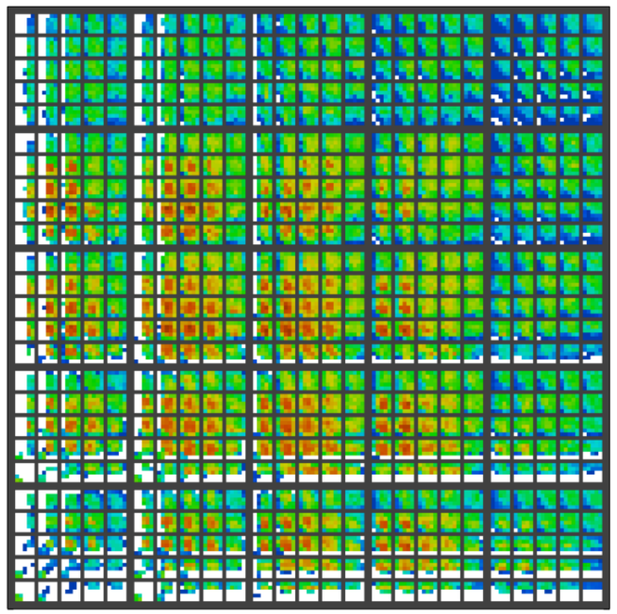


Joint learning of diverse policies with ES

intuitions about
different ways to move

- MAP-Elites
- Behavioral characterization
  - % of time each leg touches the ground (6-dimensional)
- Massive search space
- MAP-Elites map has ~13,000 diverse, high-performing gaits





Initial Map

Robots that can adapt like animals, Cully et al.

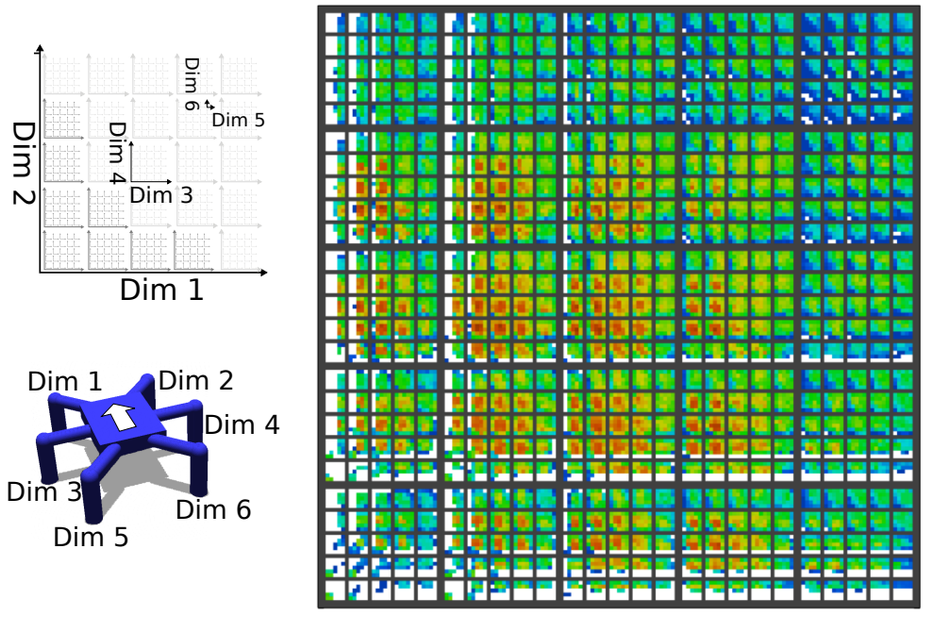
Task diversity

1.The engineer comes up with a low-dimensional characterization of the behaviour.

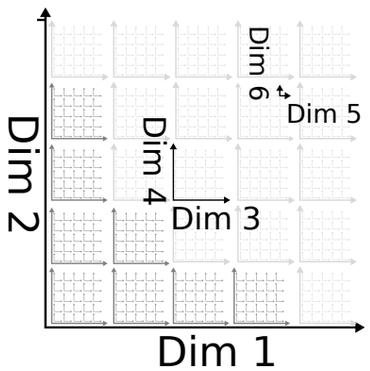
- Example: walking, dimensions: the percentage each robot leg is being activated
- Example: grasping, dimensions: orientation of the gripper, number of hands involved,gripper opening

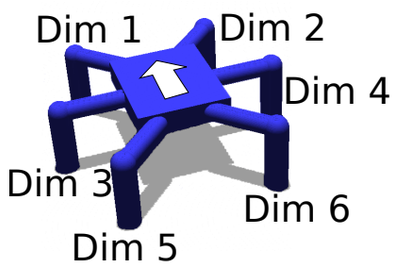
2.Training time: ES for searching for high performing policies in a grid multi-dimensional map: we keep the highest performing policy for each behaviour cell.

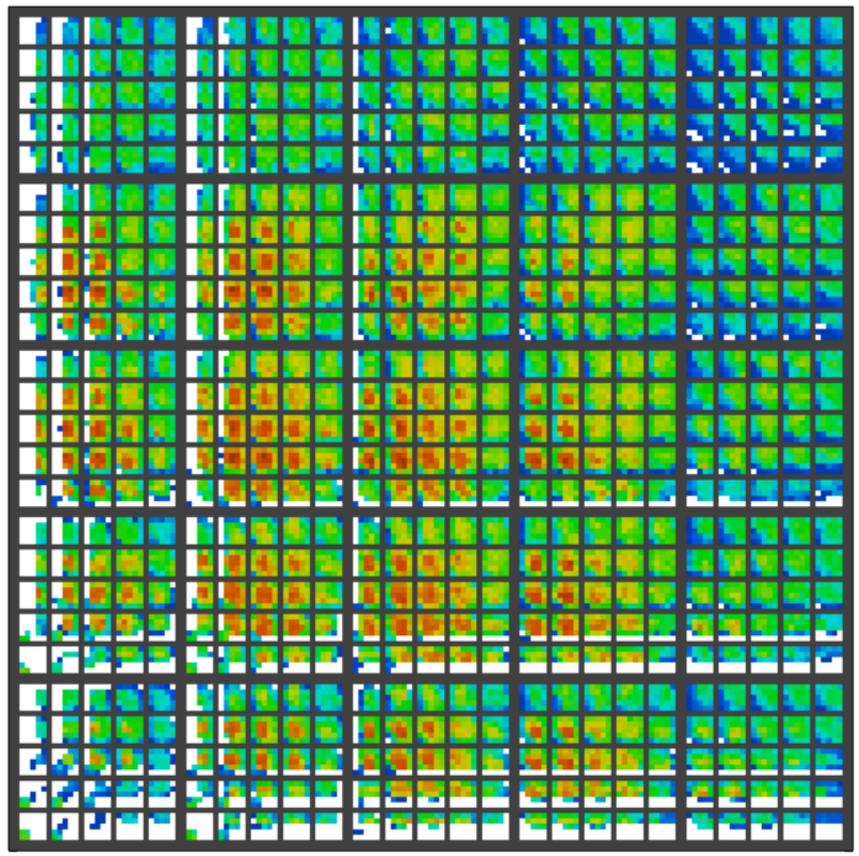
3.Test time: We want to pick the best performing policy in a new environment/ in a new robot configuration.



intuitions about
different ways to move





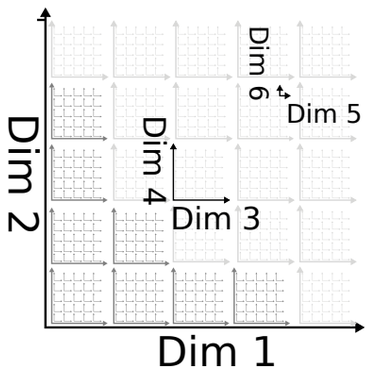


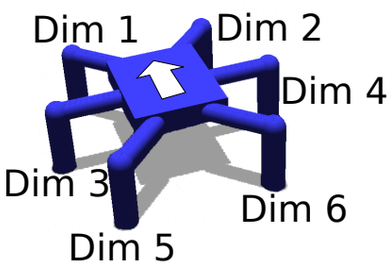
Initial Map

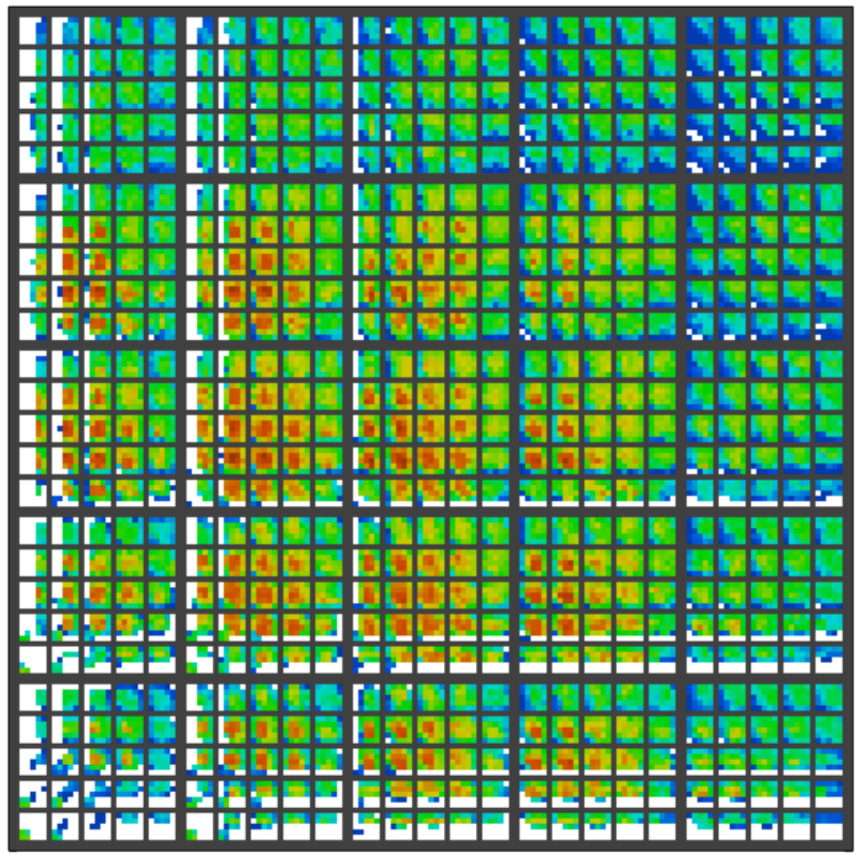
On the undamaged,
simulated robot

Robots that can adapt like animals, Cully et al.

intuitions about different ways to move







Initial Map

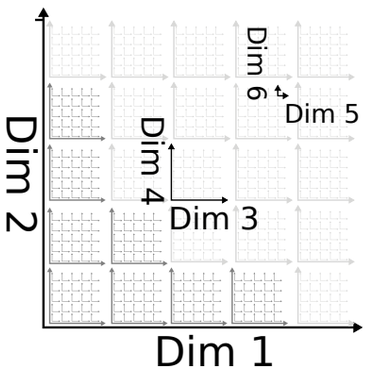
few, intelligent tests

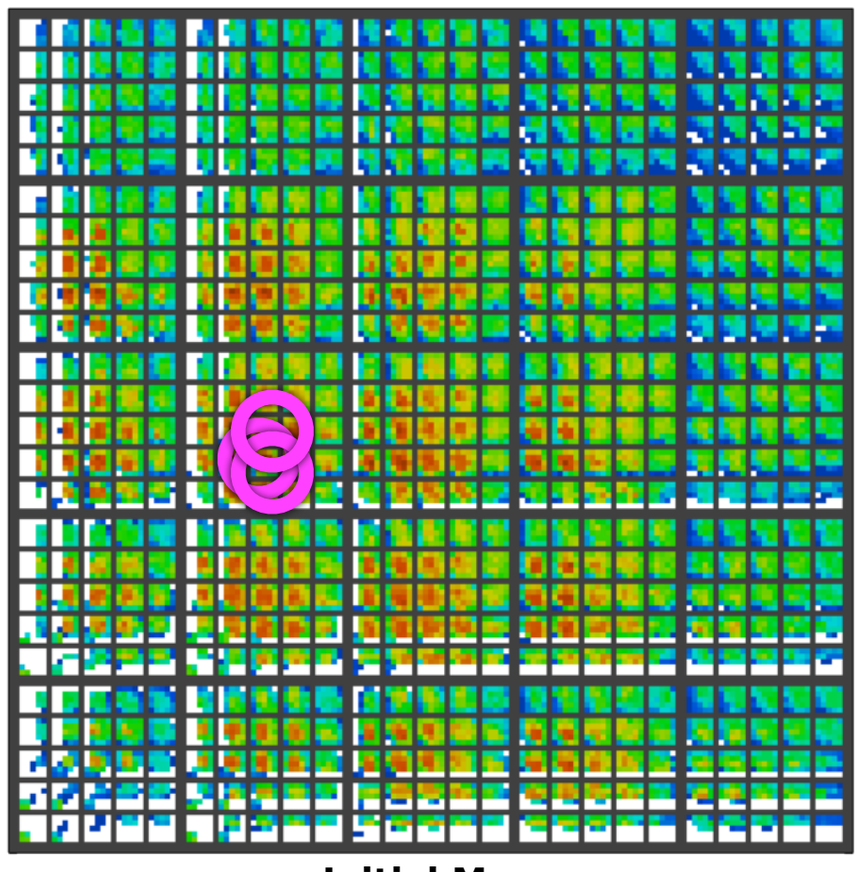
Which behaviors should we test?

Robots that can adapt like animals, Cully et al.

intuitions about different ways to move

few, intelligent tests





Initial Map

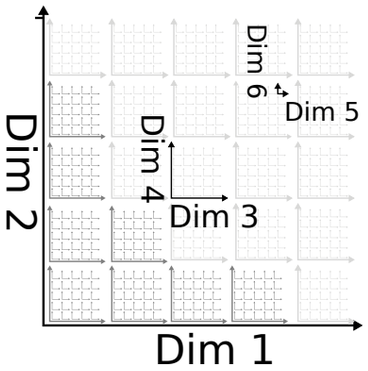
Could try top N:

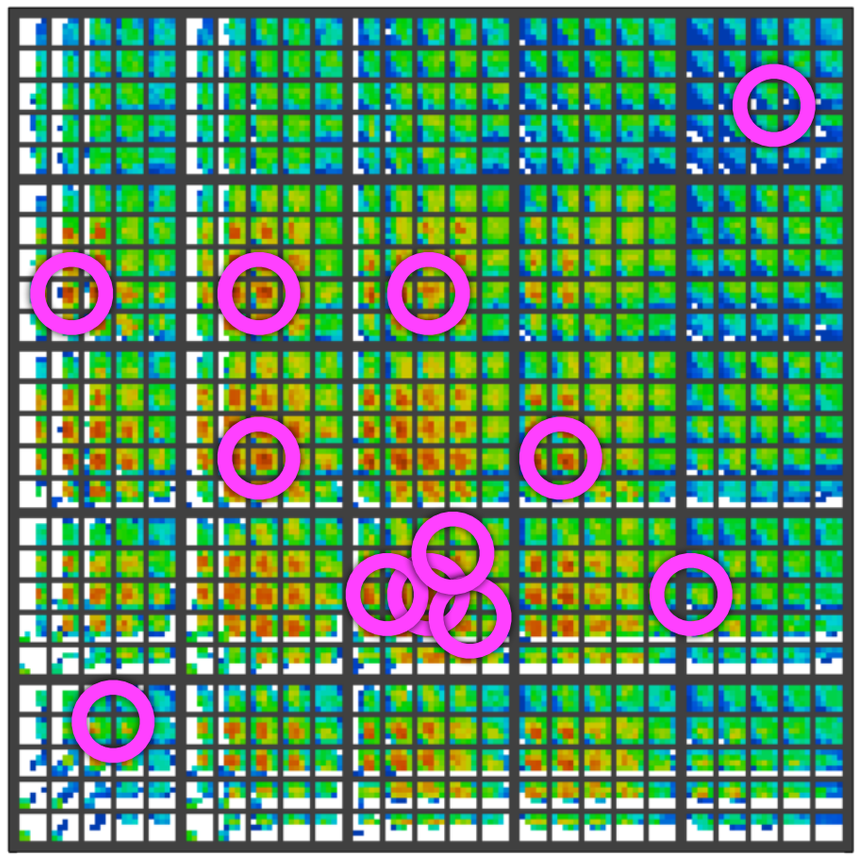
But they are likely very similar.

Robots that can adapt like animals, Cully et al.

intuitions about different ways to move

few, intelligent tests





Initial Map

Bayesian Optimization:

Tries different types solutions

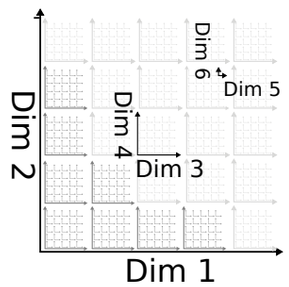
Robots that can adapt like animals, Cully et al.

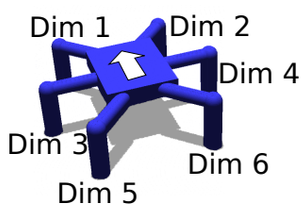
Bayesian Optimization

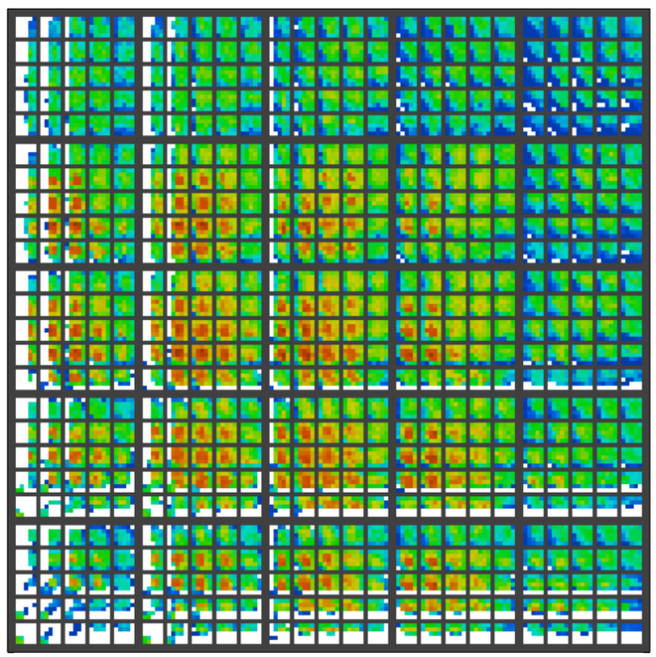
Prior:
MAP-Elites Map

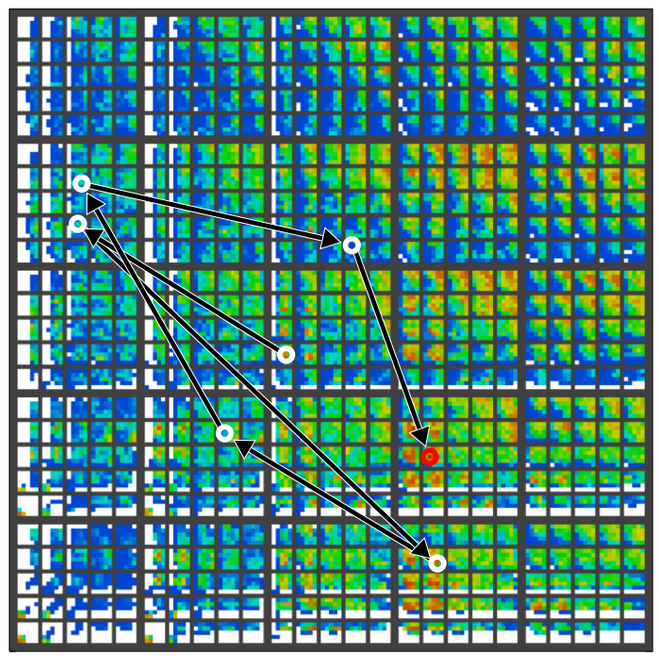
Posterior:
Map updated after real-world tests

Stop when:
A real-world behavior is >90% of best untested point





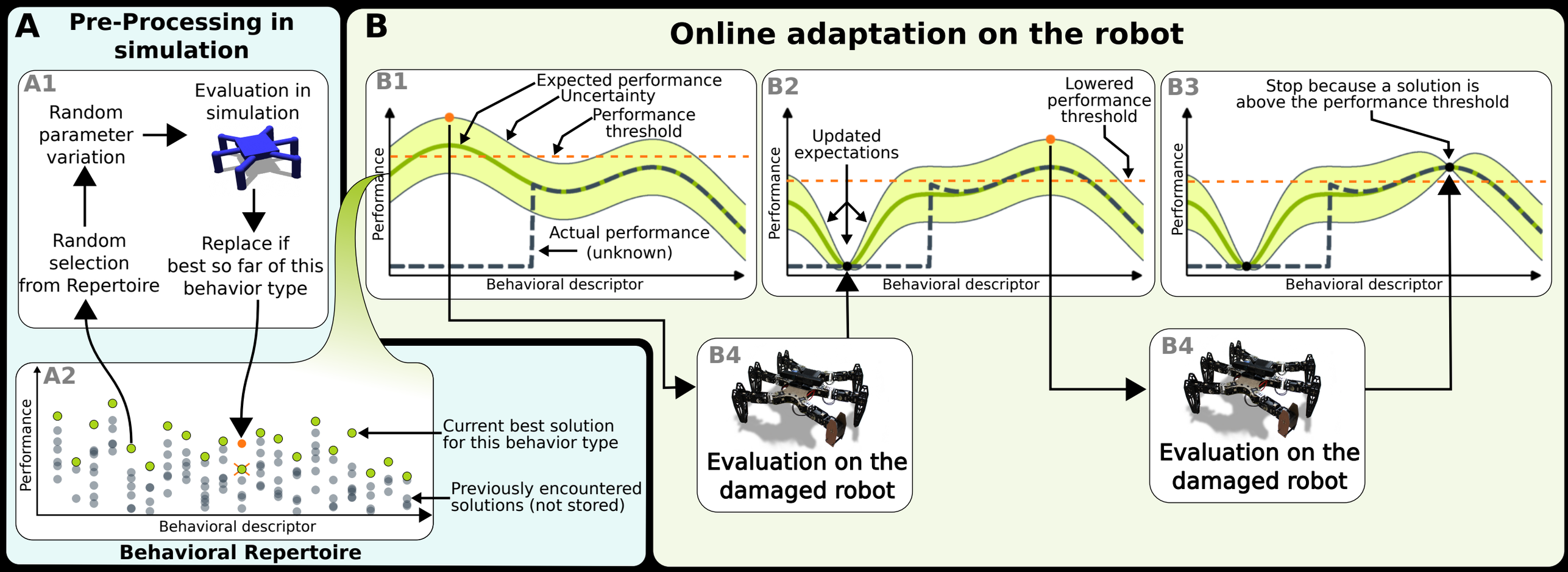




Robots that can adapt like animals, Cully et al.

Few shot adaptation through Bayesian Optimization

One-dimensional Example



Robots that can adapt like animals, Cully et al.

Robots that can adapt like animals

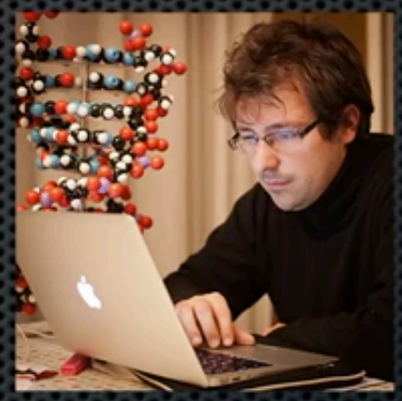
Nature, 2015

which describes damage recovery via Intelligent Trial and Error

Antoine Cully
UPMC/CNRS
(France)

Jeff Clune
University of Wyoming
(USA)

Danesh Tarapore
UPMC/CNRS
(France)



Jean-Baptiste Mouret
UPMC/CNRS/Inria/UL
(France)

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR
DGA
erc European Research Council Established by the European Commission
EVOLVING ARTIFICIAL INTELLIGENCE LABORATORY
UNIVERSITY OF WYOMING
Inria informatics mathematics
UNIVERSITÉ DE LORRAINE
Loria
ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE
cnrs
UPMC SORBONNE UNIVERSITÉS

Representing Uncertainty

In later lectures we will explore representing uncertainty in regression and classification using neural networks.

We will look into:

- Bayesian neural networks, where we are estimating a distribution for each weight parameter as opposed to a point estimate,
- neural network ensembles: sets of neural networks trained on different subsets of the data and with different initializations, the entropy of their predictions quantify the uncertainty of their estimates.