

Análise Estatística de Simuladores

Validação, UA/SA

Leo Bastos¹ Richard Wilkinson²

¹Departamento de Estatística



²Department of Statistics



The University of
Nottingham

19o SINAPE

- 1 Revisão
- 2 Diagnósticos e Validação
 - Numerical Diagnostics
 - Exemplo: Modelo de lixo nuclear
 - Exemplo: Nilson-Kuusk model
- 3 Análises de Incerteza e Sensibilidade

1 Revisão

2 Diagnósticos e Validação

- Numerical Diagnostics
- Exemplo: Modelo de lixo nuclear
- Exemplo: Nilson-Kuusk model

3 Análises de Incerteza e Sensibilidade

- **Simulator** é uma função determinística representando um sistema real.

$$\eta(\cdot) : \mathcal{X} \subset \mathbb{R}^p \rightarrow \mathbb{R}$$



- \mathbf{x} vetor de inputs $\longrightarrow y = \eta(\mathbf{x})$ é uma escalar
- Queremos aprender sobre o processo representado por $\eta(\cdot)$
- **Experimento computacional** consiste em um conjunto de rodadas do simulador para diferentes inputs.

- **Simulator** é uma função determinística representando um sistema real.

$$\eta(\cdot) : \mathcal{X} \subset \mathbb{R}^p \rightarrow \mathbb{R}$$



- \mathbf{x} vetor de inputs $\longrightarrow y = \eta(\mathbf{x})$ é uma escalar
- Queremos aprender sobre o processo representado por $\eta(\cdot)$
- **Experimento computacional** consiste em um conjunto de rodadas do simulador para diferentes inputs.

- **Simulator** é uma função determinística representando um sistema real.

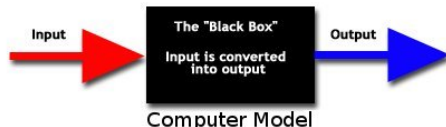
$$\eta(\cdot) : \mathcal{X} \subset \mathbb{R}^p \rightarrow \mathbb{R}$$



- \mathbf{x} vetor de inputs $\longrightarrow y = \eta(\mathbf{x})$ é uma escalar
- Queremos aprender sobre o processo representado por $\eta(\cdot)$
- **Experimento computacional** consiste em um conjunto de rodadas do simulador para diferentes inputs.

- **Simulator** é uma função determinística representando um sistema real.

$$\eta(\cdot) : \mathcal{X} \subset \mathbb{R}^p \rightarrow \mathbb{R}$$



- \mathbf{x} vetor de inputs $\longrightarrow y = \eta(\mathbf{x})$ é uma escalar
- Queremos aprender sobre o processo representado por $\eta(\cdot)$
- **Experimento computacional** consiste em um conjunto de rodadas do simulador para diferentes inputs.

Construindo um emulador gaussiano

- 1 obtenha os dados de treinamento,
 $D = \{y_1 = \eta(\mathbf{x}_1), \dots, y_n = \eta(\mathbf{x}_n)\}$. (**Planejamento**)
- 2 Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$ (**Emulação**)
 - Estime os parâmetro de correlação, $\delta = \tilde{\delta}$;
 - Escolha uma $\tilde{\eta}(\cdot)|D, \tilde{\delta}$ com uma aproximação rápida para $\eta(\cdot)$ e a distribuição de $\eta(\cdot)|D, \tilde{\delta}$ com uma aproximação rápida para $\eta(\cdot)$.
- 3 Valide seu emulador. (**Validação**)
- 4 Faça as análises de interesse. { Previsão, problema inverso, UA/SA, etc.}

Construindo um emulador gaussiano

- 1 obtenha os dados de treinamento,
 $D = \{y_1 = \eta(\mathbf{x}_1), \dots, y_n = \eta(\mathbf{x}_n)\}$. (**Planejamento**)
- 2 Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$ (**Emulação**)
 - Estime os parâmetro de correlação, $\delta = \tilde{\delta}$;
 - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
 - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.
- 3 Valide seu emulador. (**Validação**)
- 4 Faça as análises de interesse. { Previsão, problema inverso, UA/SA, etc.}

Construindo um emulador gaussiano

- 1 obtenha os dados de treinamento,
 $D = \{y_1 = \eta(\mathbf{x}_1), \dots, y_n = \eta(\mathbf{x}_n)\}$. (**Planejamento**)
- 2 Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$ (**Emulação**)
 - Estime os parâmetro de correlação, $\delta = \tilde{\delta}$;
 - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
 - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.
- 3 Valide seu emulador. (**Validação**)
- 4 Faça as análises de interesse. { Previsão, problema inverso, UA/SA, etc.}

Construindo um emulador gaussiano

- 1 obtenha os dados de treinamento,
 $D = \{y_1 = \eta(\mathbf{x}_1), \dots, y_n = \eta(\mathbf{x}_n)\}$. (**Planejamento**)
- 2 Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$ (**Emulação**)
 - Estime os parâmetro de correlação, $\delta = \tilde{\delta}$;
 - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
 - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.
- 3 Valide seu emulador. (**Validação**)
- 4 Faça as análises de interesse. { Previsão, problema inverso, UA/SA, etc.}

Construindo um emulador gaussiano

- 1 obtenha os dados de treinamento,
 $D = \{y_1 = \eta(\mathbf{x}_1), \dots, y_n = \eta(\mathbf{x}_n)\}$. (**Planejamento**)
- 2 Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$ (**Emulação**)
 - Estime os parâmetro de correlação, $\delta = \tilde{\delta}$;
 - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
 - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.
- 3 Valide seu emulador. (**Validação**)
- 4 Faça as análises de interesse. { Previsão, problema inverso, UA/SA, etc.}

Construindo um emulador gaussiano

- 1 obtenha os dados de treinamento,
 $D = \{y_1 = \eta(\mathbf{x}_1), \dots, y_n = \eta(\mathbf{x}_n)\}$. (**Planejamento**)
- 2 Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$ (**Emulação**)
 - Estime os parâmetro de correlação, $\delta = \tilde{\delta}$;
 - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
 - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.
- 3 Valide seu emulador. (**Validação**)
- 4 Faça as análises de interesse. { Previsão, problema inverso, UA/SA, etc.}

Construindo um emulador gaussiano

- 1 obtenha os dados de treinamento,
 $D = \{y_1 = \eta(\mathbf{x}_1), \dots, y_n = \eta(\mathbf{x}_n)\}$. (**Planejamento**)
- 2 Derive a posteriori para $\eta(\cdot)|D, \tilde{\delta}$ (**Emulação**)
 - Estime os parâmetro de correlação, $\delta = \tilde{\delta}$;
 - Podemos usar $\mathbb{E}[\eta(\cdot)|D, \tilde{\delta}]$ como uma aproximação rápida para $\eta(\cdot)$
 - A distribuição de $\eta(\cdot)|D, \tilde{\delta}$ quantifica a incerteza dessa aproximação.
- 3 Valide seu emulador. (**Validação**)
- 4 Faça as análises de interesse. { Previsão, problema inverso, UA/SA, etc.}

- **Emulador gaussiano:**

$$\eta(\cdot) | \beta, \sigma^2, \delta \sim GP(m_0(\cdot), V_0(\cdot, \cdot)),$$

onde

$$\begin{aligned} m_0(\mathbf{x}) &= h(\mathbf{x})^T \beta \\ V_0(\mathbf{x}, \mathbf{x}') &= \sigma^2 C_\delta(\mathbf{x}, \mathbf{x}') \end{aligned}$$

- $h(\cdot)$ e $C_\delta(\cdot, \cdot)$ são conhecidos
- $p(\beta, \sigma^2, \delta)$: priori
- **Dados de treinamento:** (\mathbf{y}, \mathbf{X}) onde $y_i = \eta(\mathbf{x}_i)$, $i = 1, \dots, n$.

- **Emulador gaussiano:**

$$\eta(\cdot) | \beta, \sigma^2, \delta \sim GP(m_0(\cdot), V_0(\cdot, \cdot)),$$

onde

$$\begin{aligned} m_0(\mathbf{x}) &= h(\mathbf{x})^T \beta \\ V_0(\mathbf{x}, \mathbf{x}') &= \sigma^2 C_\delta(\mathbf{x}, \mathbf{x}') \end{aligned}$$

- $h(\cdot)$ e $C_\delta(\cdot, \cdot)$ são conhecidos
- $p(\beta, \sigma^2, \delta)$: priori
- **Dados de treinamento:** (\mathbf{y}, \mathbf{X}) onde $y_i = \eta(\mathbf{x}_i)$, $i = 1, \dots, n$.

- **Emulador gaussiano:**

$$\eta(\cdot) | \beta, \sigma^2, \delta \sim GP(m_0(\cdot), V_0(\cdot, \cdot)),$$

onde

$$\begin{aligned} m_0(\mathbf{x}) &= h(\mathbf{x})^T \beta \\ V_0(\mathbf{x}, \mathbf{x}') &= \sigma^2 C_\delta(\mathbf{x}, \mathbf{x}') \end{aligned}$$

- $h(\cdot)$ e $C_\delta(\cdot, \cdot)$ são conhecidos
- $p(\beta, \sigma^2, \delta)$: priori
- **Dados de treinamento:** (\mathbf{y}, \mathbf{X}) onde $y_i = \eta(\mathbf{x}_i)$, $i = 1, \dots, n$.

- **Emulador gaussiano:**

$$\eta(\cdot) | \beta, \sigma^2, \delta \sim GP(m_0(\cdot), V_0(\cdot, \cdot)),$$

onde

$$\begin{aligned} m_0(\mathbf{x}) &= h(\mathbf{x})^T \beta \\ V_0(\mathbf{x}, \mathbf{x}') &= \sigma^2 C_\delta(\mathbf{x}, \mathbf{x}') \end{aligned}$$

- $h(\cdot)$ e $C_\delta(\cdot, \cdot)$ são conhecidos
- $p(\beta, \sigma^2, \delta)$: priori
- **Dados de treinamento:** (\mathbf{y}, \mathbf{X}) onde $y_i = \eta(\mathbf{x}_i)$, $i = 1, \dots, n$.

- Pode-se mostrar que

$$\eta(\cdot)|\mathbf{y}, \mathbf{X}, \delta \sim \text{Student-Process}(n - q, m_1(\cdot), V_1(\cdot, \cdot)),$$

onde

$$\begin{aligned} m_1(x) &= h(x)^T \hat{\beta} + t_\delta(x)^T \mathbf{A}^{-1}(\mathbf{y} - H\hat{\beta}), \\ V_1(x, x') &= \hat{\sigma}^2 [C_\delta(x, x') - t_\delta(x)^T \mathbf{A}^{-1} t_\delta(x') + (h(x) - t_\delta(x)^T \mathbf{A}^{-1} H) \\ &\quad \times (H^T \mathbf{A}^{-1} H)^{-1} (h(x') - t_\delta(x')^T \mathbf{A}^{-1} H)^T]. \end{aligned}$$

1 Revisão

2 Diagnósticos e Validação

- Numerical Diagnostics
- Exemplo: Modelo de lixo nuclear
- Exemplo: Nilson-Kuusk model

3 Análises de Incerteza e Sensibilidade

- **Validation** generally means: “*the emulator predictions are close enough to the simulator outputs*”.
- We want to take account all the uncertainty associated with the emulator.
- “Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?”
- Choices for the Gaussian process emulator:
 - Training strategy
 - Covariance function
 - Correlation of emulator coefficients

- **Validation** generally means: “*the emulator predictions are close enough to the simulator outputs*”.
- We want to take account all the uncertainty associated with the emulator.
- “Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?”
- Choices for the Gaussian process emulator:

- **Validation** generally means: “*the emulator predictions are close enough to the simulator outputs*”.
- We want to take account all the uncertainty associated with the emulator.
- “Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?”
- Choices for the Gaussian process emulator:
 - Normality
 - Stationarity
 - Constant mean

- **Validation** generally means: “*the emulator predictions are close enough to the simulator outputs*”.
- We want to take account all the uncertainty associated with the emulator.
- “Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?”
- Choices for the Gaussian process emulator:
 - Normality
 - Stationarity
 - Correlation parameters estimation

- **Validation** generally means: “*the emulator predictions are close enough to the simulator outputs*”.
- We want to take account all the uncertainty associated with the emulator.
- “Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?”
- Choices for the Gaussian process emulator:
 - Normality
 - Stationarity
 - Correlation parameters estimation

- **Validation** generally means: “*the emulator predictions are close enough to the simulator outputs*”.
- We want to take account all the uncertainty associated with the emulator.
- “Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?”
- Choices for the Gaussian process emulator:
 - Normality
 - Stationarity
 - Correlation parameters estimation

- **Validation** generally means: “*the emulator predictions are close enough to the simulator outputs*”.
- We want to take account all the uncertainty associated with the emulator.
- “Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?”
- Choices for the Gaussian process emulator:
 - Normality
 - Stationarity
 - Correlation parameters estimation

- 1 Since the simulator is deterministic, we need new computer experimental runs for validation.
- 2 Such data is called **validation data**
 - Validation inputs: $\mathbf{X}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_m^*)^T$
 - Validation outputs: $\mathbf{Y}^* = (y_1^*, \dots, y_m^*)^T$
- 3 The idea is to compare the validation outputs with the emulator outputs taking into account associated uncertainty.

- 1 Since the simulator is deterministic, we need new computer experimental runs for validation.
- 2 Such data is called **validation data**
 - Validation inputs: $\mathbf{X}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_m^*)^T$
 - Validation outputs: $(y_1^* = \eta(\mathbf{x}_1^*), \dots, y_m^* = \eta(\mathbf{x}_m^*))$
- 3 The idea is to compare the validation outputs with the emulator outputs taking into account associated uncertainty.

- 1 Since the simulator is deterministic, we need new computer experimental runs for validation.
- 2 Such data is called **validation data**
 - Validation inputs: $\mathbf{X}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_m^*)^T$
 - Validation outputs: $(y_1^* = \eta(\mathbf{x}_1^*), \dots, y_m^* = \eta(\mathbf{x}_m^*))$
- 3 The idea is to compare the validation outputs with the emulator outputs taking into account associated uncertainty.

- 1 Since the simulator is deterministic, we need new computer experimental runs for validation.
- 2 Such data is called **validation data**
 - Validation inputs: $\mathbf{X}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_m^*)^T$
 - Validation outputs: $(y_1^* = \eta(\mathbf{x}_1^*), \dots, y_m^* = \eta(\mathbf{x}_m^*))$
- 3 The idea is to compare the validation outputs with the emulator outputs taking into account associated uncertainty.

- 1 Since the simulator is deterministic, we need new computer experimental runs for validation.
- 2 Such data is called **validation data**
 - Validation inputs: $\mathbf{X}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_m^*)^T$
 - Validation outputs: $(y_1^* = \eta(\mathbf{x}_1^*), \dots, y_m^* = \eta(\mathbf{x}_m^*))$
- 3 The idea is to compare the validation outputs with the emulator outputs taking into account associated uncertainty.

1 Numerical Diagnostics (Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)

- (i) Individual prediction errors $D_i(\cdot)$
- (ii) Credible interval diagnostic $D_{CI}(\cdot)$
- (iii) Mahalanobis Distance $D_{MD}(\cdot)$

2 Graphical Diagnostics

- (i) Scatterplots and overlaid density
- (ii) Individual errors against simulator's output
- (iii) Individual errors against output
- (iv) Credible interval against the output
- (v) Mahalanobis distance against the output

- 1 Numerical Diagnostics
 - (Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)
 - (i) Individual prediction errors $D_i(\cdot)$
 - (ii) Credible interval diagnostic $D_{CI}(\cdot)$
 - (iii) Mahalanobis Distance $D_{MD}(\cdot)$
- 2 Graphical Diagnostics

- 1 Numerical Diagnostics
(Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)
 - (i) Individual prediction errors $D_i(\cdot)$
 - (ii) Credible interval diagnostic $D_{CI}(\cdot)$
 - (iii) Mahalanobis Distance $D_{MD}(\cdot)$
- 2 Graphical Diagnostics

1 Numerical Diagnostics

(Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)

- (i) Individual prediction errors $D_i(\cdot)$
- (ii) Credible interval diagnostic $D_{CI}(\cdot)$
- (iii) Mahalanobis Distance $D_{MD}(\cdot)$

2 Graphical Diagnostics

- (i) Correlated and uncorrelated errors

1 Numerical Diagnostics

(Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)

- (i) Individual prediction errors $D_i(\cdot)$
- (ii) Credible interval diagnostic $D_{CI}(\cdot)$
- (iii) Mahalanobis Distance $D_{MD}(\cdot)$

2 Graphical Diagnostics

- (i) Correlated and uncorrelated errors
- (ii) Individual errors against emulator's predictions
- (iii) Individual errors against inputs
- (iv) Uncorrelated errors against the index
- (v) QQ-plot of the uncorrelated errors

1 Numerical Diagnostics

(Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)

- (i) Individual prediction errors $D_i(\cdot)$
- (ii) Credible interval diagnostic $D_{CI}(\cdot)$
- (iii) Mahalanobis Distance $D_{MD}(\cdot)$

2 Graphical Diagnostics

- (i) Correlated and uncorrelated errors
- (ii) Individual errors against emulator's predictions
- (iii) Individual errors against inputs
- (iv) Uncorrelated errors against the index
- (v) QQ-plot of the uncorrelated errors

1 Numerical Diagnostics

(Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)

- (i) Individual prediction errors $D_i(\cdot)$
- (ii) Credible interval diagnostic $D_{CI}(\cdot)$
- (iii) Mahalanobis Distance $D_{MD}(\cdot)$

2 Graphical Diagnostics

- (i) Correlated and uncorrelated errors
- (ii) Individual errors against emulator's predictions
- (iii) Individual errors against inputs
- (iv) Uncorrelated errors against the index
- (v) QQ-plot of the uncorrelated errors

1 Numerical Diagnostics

(Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)

- (i) Individual prediction errors $D_i(\cdot)$
- (ii) Credible interval diagnostic $D_{CI}(\cdot)$
- (iii) Mahalanobis Distance $D_{MD}(\cdot)$

2 Graphical Diagnostics

- (i) Correlated and uncorrelated errors
- (ii) Individual errors against emulator's predictions
- (iii) Individual errors against inputs
- (iv) Uncorrelated errors against the index
- (v) QQ-plot of the uncorrelated errors

1 Numerical Diagnostics

(Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)

- (i) Individual prediction errors $D_i(\cdot)$
- (ii) Credible interval diagnostic $D_{CI}(\cdot)$
- (iii) Mahalanobis Distance $D_{MD}(\cdot)$

2 Graphical Diagnostics

- (i) Correlated and uncorrelated errors
- (ii) Individual errors against emulator's predictions
- (iii) Individual errors against inputs
- (iv) Uncorrelated errors against the index
- (v) QQ-plot of the uncorrelated errors

1 Numerical Diagnostics

(Compare $D(\mathbf{y}^*)$ with $\mathbb{P}(D(\eta(\mathbf{X}^*))|\mathbf{y})$)

- (i) Individual prediction errors $D_i(\cdot)$
- (ii) Credible interval diagnostic $D_{CI}(\cdot)$
- (iii) Mahalanobis Distance $D_{MD}(\cdot)$

2 Graphical Diagnostics

- (i) Correlated and uncorrelated errors
- (ii) Individual errors against emulator's predictions
- (iii) Individual errors against inputs
- (iv) Uncorrelated errors against the index
- (v) QQ-plot of the uncorrelated errors

- Individual prediction errors (standardised)

$$D_i(y^*) = \frac{y_i^* - m_1(\mathbf{x}_i^*)}{\sqrt{V_1(\mathbf{x}_i^*)}}$$

It can be shown that the joint distribution of the individual prediction errors is

$$D_1(\eta(\mathbf{X})^*) | D, \delta \sim \text{Multi Student-t}(n - q, 0, C(\mathbf{X}^*; \psi))$$

- Credible interval diagnostic

$$D_{CI}(\mathbf{y}^*) = \frac{1}{m} \sum_{i=1}^m \mathbf{1}(\mathbf{y}_i^* \in CI_i(\alpha))$$

The reference distribution can be obtained by simulation.

- Individual prediction errors (standardised)

$$D_i(y^*) = \frac{y_i^* - m_1(\mathbf{x}_i^*)}{\sqrt{V_1(\mathbf{x}_i^*)}}$$

It can be shown that the joint distribution of the individual prediction errors is

$$D_1(\eta(\mathbf{X})^*) | D, \delta \sim \text{Multi Student-t}(n - q, 0, C(\mathbf{X}^*; \psi))$$

- Credible interval diagnostic

$$D_{CI}(\mathbf{y}^*) = \frac{1}{m} \sum_{i=1}^m \mathbf{1}(\mathbf{y}_i^* \in CI_i(\alpha))$$

The reference distribution can be obtained by simulation.

- Mahalanobis Distance

$$D_{MD}(\mathbf{y}^*) = (\mathbf{y}^* - m_1(\mathbf{X}^*))^T (V_1(\mathbf{X}^*))^{-1} (\mathbf{y}^* - m_1(\mathbf{X}^*))$$

The distribution of the Mahalanobis distance is

$$\frac{(n - q)}{m(n - q - 2)} D_{MD}(\eta(\mathbf{X}^*)) | D, \delta \sim F_{m, n-q}$$

(Bastos, 2010)

- Extreme values for the observed Mahalanobis distance can indicate:

- Correlation hyperparameters (δ) were poorly estimated

- Mahalanobis Distance

$$D_{MD}(\mathbf{y}^*) = (\mathbf{y}^* - m_1(\mathbf{X}^*))^T (V_1(\mathbf{X}^*)^{-1} (\mathbf{y}^* - m_1(\mathbf{X}^*)))$$

The distribution of the Mahalanobis distance is

$$\frac{(n - q)}{m(n - q - 2)} D_{MD}(\eta(\mathbf{X}^*)) | D, \delta \sim F_{m, n-q}$$

(Bastos, 2010)

- Extreme values for the observed Mahalanobis distance can indicate:
 - Correlation hyperparameters (δ) were poorly estimated
 - Stationary GP assumption can be too for approximating the simulator

- Mahalanobis Distance

$$D_{MD}(\mathbf{y}^*) = (\mathbf{y}^* - m_1(\mathbf{X}^*))^T (V_1(\mathbf{X}^*)^{-1} (\mathbf{y}^* - m_1(\mathbf{X}^*)))$$

The distribution of the Mahalanobis distance is

$$\frac{(n - q)}{m(n - q - 2)} D_{MD}(\eta(\mathbf{X}^*)) | D, \delta \sim F_{m, n-q}$$

(Bastos, 2010)

- Extreme values for the observed Mahalanobis distance can indicate:
 - Correlation hyperparameters (δ) were poorly estimated
 - Stationary GP assumption can be too for approximating the simulator

- Mahalanobis Distance

$$D_{MD}(\mathbf{y}^*) = (\mathbf{y}^* - m_1(\mathbf{X}^*))^T (V_1(\mathbf{X}^*)^{-1} (\mathbf{y}^* - m_1(\mathbf{X}^*)))$$

The distribution of the Mahalanobis distance is

$$\frac{(n - q)}{m(n - q - 2)} D_{MD}(\eta(\mathbf{X}^*)) | D, \delta \sim F_{m, n-q}$$

(Bastos, 2010)

- Extreme values for the observed Mahalanobis distance can indicate:
 - Correlation hyperparameters (δ) were poorly estimated
 - Stationary GP assumption can be too for approximating the simulator

Individual errors

$$D_i(y^*) = \frac{y_i^* - E[\eta(\mathbf{x}_i^*)|D, \delta]}{\sqrt{V[\eta(\mathbf{x}_i^*)|D, \delta]}}$$

- $D_i(y^*)$ s against emulator's predictions
- $D_i(y^*)$ s against inputs
- These errors are correlated

Uncorrelated errors

$$D_G(\mathbf{y}^*) = \mathbf{G}^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

where $V[\eta(\mathbf{X}^*)|D, \delta] = \mathbf{G}\mathbf{G}^T$

- $D_G(\mathbf{y}^*)$ against the index*
- QQ-plot of $D_G(\mathbf{y}^*)$
- $D_{MD}(\mathbf{y}^*) = D_G(\mathbf{y}^*)^T D_G(\mathbf{y}^*)$

Individual errors

$$D_i(y^*) = \frac{y_i^* - E[\eta(\mathbf{x}_i^*)|D, \delta]}{\sqrt{V[\eta(\mathbf{x}_i^*)|D, \delta]}}$$

- $D_i(y^*)$ s against emulator's predictions
- $D_i(y^*)$ s against inputs
- These errors are correlated

Uncorrelated errors

$$D_G(\mathbf{y}^*) = \mathbf{G}^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

where $V[\eta(\mathbf{X}^*)|D, \delta] = \mathbf{G}\mathbf{G}^T$

- $D_G(\mathbf{y}^*)$ against the index*
- QQ-plot of $D_G(\mathbf{y}^*)$
- $D_{MD}(\mathbf{y}^*) = D_G(\mathbf{y}^*)^T D_G(\mathbf{y}^*)$

Individual errors

$$D_i(y^*) = \frac{y_i^* - E[\eta(\mathbf{x}_i^*)|D, \delta]}{\sqrt{V[\eta(\mathbf{x}_i^*)|D, \delta]}}$$

- $D_i(y^*)$ s against emulator's predictions
- $D_i(y^*)$ s against inputs
- These errors are correlated

Uncorrelated errors

$$D_G(\mathbf{y}^*) = \mathbf{G}^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

where $V[\eta(\mathbf{X}^*)|D, \delta] = \mathbf{G}\mathbf{G}^T$

- $D_G(\mathbf{y}^*)$ against the index*
- QQ-plot of $D_G(\mathbf{y}^*)$
- $D_{MD}(\mathbf{y}^*) = D_G(\mathbf{y}^*)^T D_G(\mathbf{y}^*)$

Individual errors

$$D_i(y^*) = \frac{y_i^* - E[\eta(\mathbf{x}_i^*)|D, \delta]}{\sqrt{V[\eta(\mathbf{x}_i^*)|D, \delta]}}$$

- $D_i(y^*)$ s against emulator's predictions
- $D_i(y^*)$ s against inputs
- **These errors are correlated**

Uncorrelated errors

$$D_G(\mathbf{y}^*) = \mathbf{G}^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

where $V[\eta(\mathbf{X}^*)|D, \delta] = \mathbf{G}\mathbf{G}^T$

- $D_G(\mathbf{y}^*)$ against the index*
- QQ-plot of $D_G(\mathbf{y}^*)$
- $D_{MD}(\mathbf{y}^*) = D_G(\mathbf{y}^*)^T D_G(\mathbf{y}^*)$

Individual errors

$$D_i(y^*) = \frac{y_i^* - E[\eta(\mathbf{x}_i^*)|D, \delta]}{\sqrt{V[\eta(\mathbf{x}_i^*)|D, \delta]}}$$

- $D_i(y^*)$ s against emulator's predictions
- $D_i(y^*)$ s against inputs
- These errors are correlated

Uncorrelated errors

$$D_G(\mathbf{y}^*) = \mathbf{G}^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

$$\text{where } V[\eta(\mathbf{X}^*)|D, \delta] = \mathbf{G}\mathbf{G}^T$$

- $D_G(\mathbf{y}^*)$ against the index*
- QQ-plot of $D_G(\mathbf{y}^*)$
- $D_{MD}(\mathbf{y}^*) = D_G(\mathbf{y}^*)^T D_G(\mathbf{y}^*)$

Individual errors

$$D_i(y^*) = \frac{y_i^* - E[\eta(\mathbf{x}_i^*)|D, \delta]}{\sqrt{V[\eta(\mathbf{x}_i^*)|D, \delta]}}$$

- $D_i(y^*)$ s against emulator's predictions
- $D_i(y^*)$ s against inputs
- These errors are correlated

Uncorrelated errors

$$D_G(\mathbf{y}^*) = \mathbf{G}^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

where $V[\eta(\mathbf{X}^*)|D, \delta] = \mathbf{G}\mathbf{G}^T$

- $D_G(\mathbf{y}^*)$ against the index*
- QQ-plot of $D_G(\mathbf{y}^*)$
- $D_{MD}(\mathbf{y}^*) = D_G(\mathbf{y}^*)^T D_G(\mathbf{y}^*)$

Individual errors

$$D_i(y^*) = \frac{y_i^* - E[\eta(\mathbf{x}_i^*)|D, \delta]}{\sqrt{V[\eta(\mathbf{x}_i^*)|D, \delta]}}$$

- $D_i(y^*)$ s against emulator's predictions
- $D_i(y^*)$ s against inputs
- These errors are correlated

Uncorrelated errors

$$D_G(\mathbf{y}^*) = \mathbf{G}^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

$$\text{where } V[\eta(\mathbf{X}^*)|D, \delta] = \mathbf{G}\mathbf{G}^T$$

- $D_G(\mathbf{y}^*)$ against the index*
- **QQ-plot of $D_G(\mathbf{y}^*)$**
- $D_{MD}(\mathbf{y}^*) = D_G(\mathbf{y}^*)^T D_G(\mathbf{y}^*)$

Individual errors

$$D_i(y^*) = \frac{y_i^* - E[\eta(\mathbf{x}_i^*)|D, \delta]}{\sqrt{V[\eta(\mathbf{x}_i^*)|D, \delta]}}$$

- $D_i(y^*)$ s against emulator's predictions
- $D_i(y^*)$ s against inputs
- These errors are correlated

Uncorrelated errors

$$D_G(\mathbf{y}^*) = \mathbf{G}^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

$$\text{where } V[\eta(\mathbf{X}^*)|D, \delta] = \mathbf{G}\mathbf{G}^T$$

- $D_G(\mathbf{y}^*)$ against the index*
- QQ-plot of $D_G(\mathbf{y}^*)$
- $D_{MD}(\mathbf{y}^*) = D_G(\mathbf{y}^*)^T D_G(\mathbf{y}^*)$

Decomposing the covariance matrix

- Cholesky decomposition (PCD)

$$V = \mathbf{R}^T \mathbf{R}$$

- Pivoting Cholesky decomposition (PCD)

$$\mathbf{P}^T \mathbf{V} \mathbf{P} = \mathbf{R}^T \mathbf{R}$$

- Eigen decomposition

$$V = \mathbf{E}^T \mathbf{\Lambda} \mathbf{E}$$

Decomposing the covariance matrix

- Cholesky decomposition (PCD)

$$V = \mathbf{R}^T \mathbf{R}$$

- Pivoting Cholesky decomposition (PCD)

$$\mathbf{P}^T V \mathbf{P} = \mathbf{R}^T \mathbf{R}$$

- Eigen decomposition

$$V = \mathbf{E}^T \mathbf{\Lambda} \mathbf{E}$$

Decomposing the covariance matrix

- Cholesky decomposition (PCD)

$$V = \mathbf{R}^T \mathbf{R}$$

- Pivoting Cholesky decomposition (PCD)

$$\mathbf{P}^T V \mathbf{P} = \mathbf{R}^T \mathbf{R}$$

- Eigen decomposition

$$V = \mathbf{E}^T \mathbf{\Lambda} \mathbf{E}$$

$$D^{PC}(\mathbf{y}^*) = (\mathbf{P}\mathbf{R}^T)^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

- $D^{PC}(\mathbf{y}^*)|D, \delta \sim \text{Multi Student-t}(n - q, 0, C_\delta(\mathbf{X}^*))$
- Extreme errors might indicate non-stationarity
- Large errors might indicate under-confident emulator
- Small errors might indicate over-confident emulator
- Pivoting order helps on the graphical interpretation

$$D^{PC}(\mathbf{y}^*) = (\mathbf{P}\mathbf{R}^T)^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

- $D^{PC}(\mathbf{y}^*)|D, \delta \sim \text{Multi Student-t}(n - q, 0, C_\delta(\mathbf{X}^*))$
- Extreme errors might indicate non-stationarity
- Large errors might indicate under-confident emulator
- Small errors might indicate over-confident emulator
- Pivoting order helps on the graphical interpretation

$$D^{PC}(\mathbf{y}^*) = (\mathbf{P}\mathbf{R}^T)^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

- $D^{PC}(\mathbf{y}^*)|D, \delta \sim \text{Multi Student-t}(n - q, 0, C_\delta(\mathbf{X}^*))$
- Extreme errors might indicate non-stationarity
- Large errors might indicate under-confident emulator
- Small errors might indicate over-confident emulator
- Pivoting order helps on the graphical interpretation

$$D^{PC}(\mathbf{y}^*) = (\mathbf{P}\mathbf{R}^T)^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

- $D^{PC}(\mathbf{y}^*)|D, \delta \sim \text{Multi Student-t}(n - q, 0, C_\delta(\mathbf{X}^*))$
- Extreme errors might indicate non-stationarity
- Large errors might indicate under-confident emulator
- Small errors might indicate over-confident emulator
- Pivoting order helps on the graphical interpretation

$$D^{PC}(\mathbf{y}^*) = (\mathbf{P}\mathbf{R}^T)^{-1}(\mathbf{y}^* - E[\eta(\mathbf{x}^*)|D, \delta])$$

- $D^{PC}(\mathbf{y}^*)|D, \delta \sim \text{Multi Student-t}(n - q, 0, C_\delta(\mathbf{X}^*))$
- Extreme errors might indicate non-stationarity
- Large errors might indicate under-confident emulator
- Small errors might indicate over-confident emulator
- Pivoting order helps on the graphical interpretation

Example: Nuclear Waste Repository



Source: <http://web.ead.anl.gov/resrad/>

- RESRAD is a computer model designed to estimate radiation doses and risks from RESidual RADioactive materials.
- Maximal release of contamination in drinking water over **10,000 years** (in millirems)

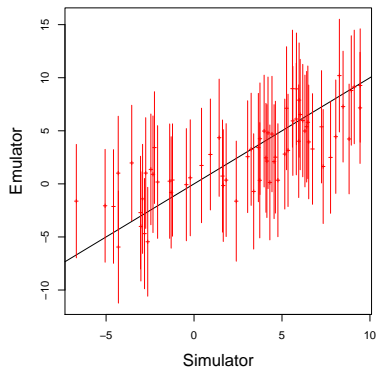
Example: Nuclear Waste Repository



Source: <http://web.ead.anl.gov/resrad/>

- RESRAD is a computer model designed to estimate radiation doses and risks from RESidual RADioactive materials.
- Maximal release of contamination in drinking water over **10,000 years** (in millirems)

Example: Nuclear Waste Repository

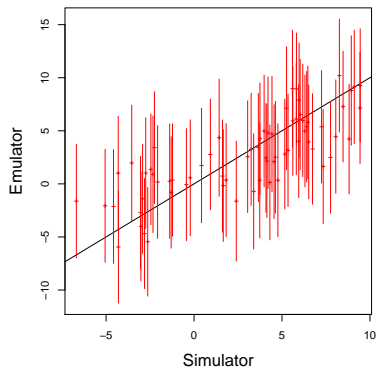


- Output - Log of maximal dose of radiation in drinking water
- 27 inputs
- Training data: $n = 190^*$
- Validation data: $m = 69^*$

• Numerical diagnostics

	Observed	Expected	95% CI
$D_{CI}(\cdot)$	0.943	0.950	(0.886; 1.000)
$D_{MD}(\cdot)$	58.96	70.00	(47.13; 104.70)

Example: Nuclear Waste Repository

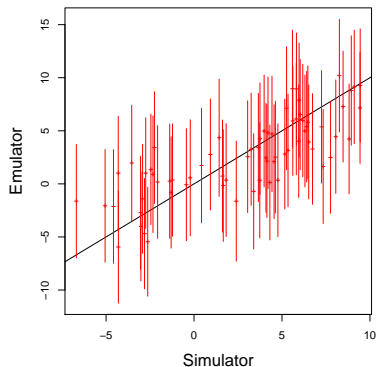


- Output - Log of maximal dose of radiation in drinking water
- 27 inputs
- Training data: $n = 190^*$
- Validation data: $m = 69^*$

• Numerical diagnostics

	Observed	Expected	95% CI
$D_{CI}(\cdot)$	0.943	0.950	(0.886; 1.000)
$D_{MD}(\cdot)$	58.96	70.00	(47.13; 104.70)

Example: Nuclear Waste Repository

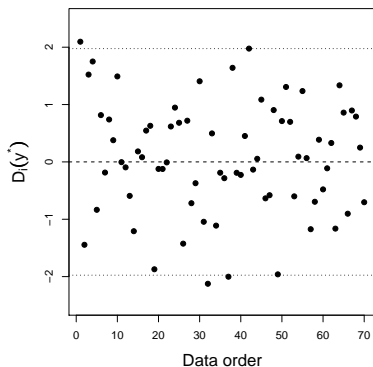


- Output - Log of maximal dose of radiation in drinking water
- 27 inputs
- Training data: $n = 190^*$
- Validation data: $m = 69^*$

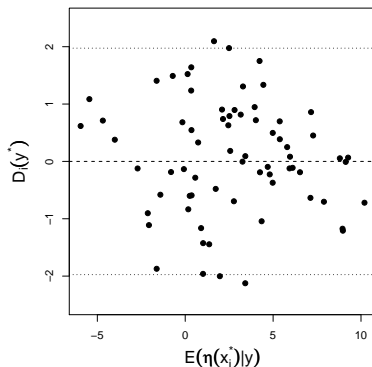
• Numerical diagnostics

	Observed	Expected	95% CI
$D_{CI}(\cdot)$	0.943	0.950	(0.886; 1.000)
$D_{MD}(\cdot)$	58.96	70.00	(47.13; 104.70)

Graphical Diagnostics: Individual errors

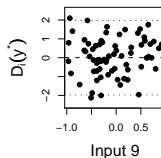
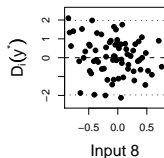
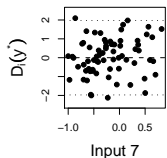
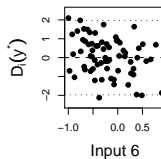
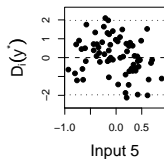
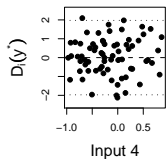
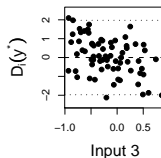
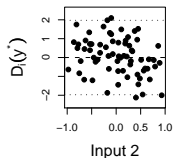
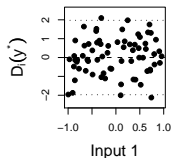


$D_i(\mathbf{y}^*)$ against data order

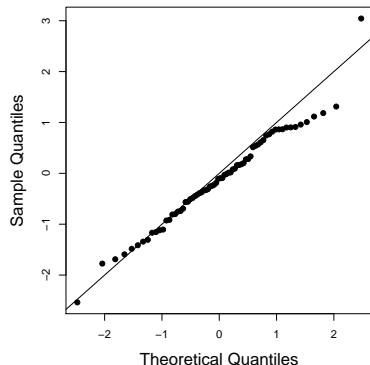


$D_i(\mathbf{y}^*)$ against emulator's predictions

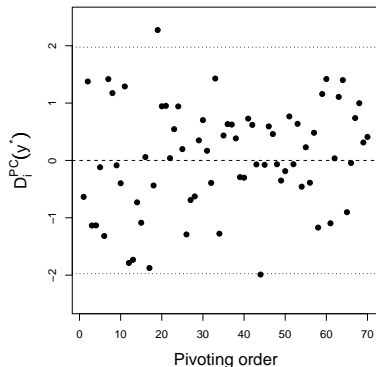
Graphical Diagnostics: Individual errors



Graphical Diagnostics: Correlated errors



QQ-plot of $D_i^{PC}(\mathbf{y}^*)$



$D_i^{PC}(\mathbf{y}^*)$ against the pivoting order

Example: Nilson-Kuusk model

- **Example:** Nilson-Kuusk model is a reflectance model for a homogeneous plant canopy.
 - The Nilson-Kuusk model is a single output model with 5 inputs
 - The training data contains 150 points
 - The validation data contains 100 points

Example: Nilson-Kuusk model

- **Example:** Nilson-Kuusk model is a reflectance model for a homogeneous plant canopy.
 - The Nilson-Kuusk model is a single output model with 5 inputs
 - The training data contains 150 points
 - The validation data contains 100 points

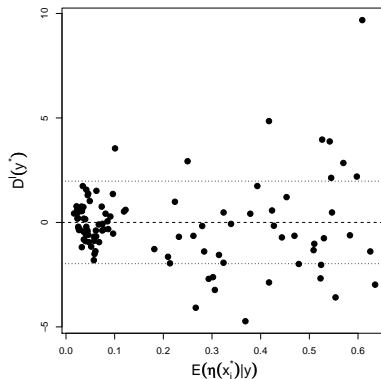
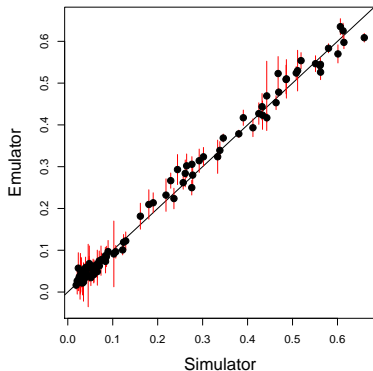
Example: Nilson-Kuusk model

- **Example:** Nilson-Kuusk model is a reflectance model for a homogeneous plant canopy.
 - The Nilson-Kuusk model is a single output model with 5 inputs
 - The training data contains 150 points
 - The validation data contains 100 points

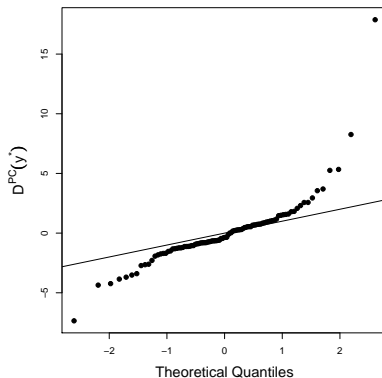
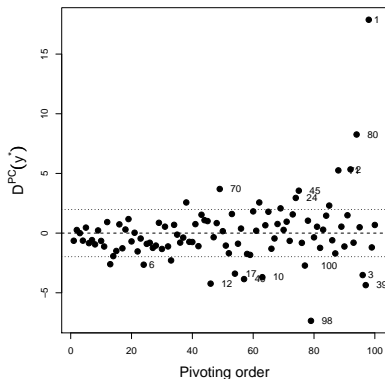
Example: Nilson-Kuusk model

- **Example:** Nilson-Kuusk model is a reflectance model for a homogeneous plant canopy.
 - The Nilson-Kuusk model is a single output model with 5 inputs
 - The training data contains 150 points
 - The validation data contains 100 points

Graphical Diagnostics - Individual Errors

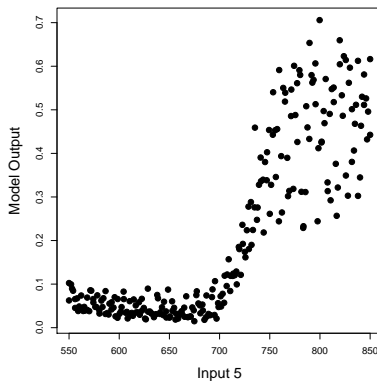
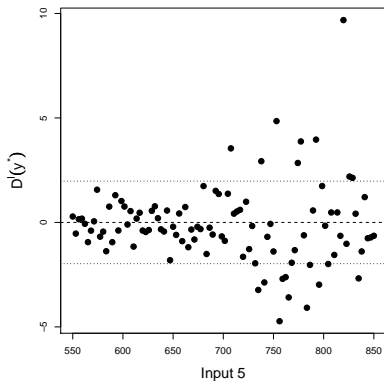


Graphical Diagnostics - Uncorrelated Errors



$D_{MD}(y^*) = 750.237$ and the 95% CI is (69.0, 142.6)
Indicating a conflict between emulator and simulator.

Graphical Diagnostics - Input 5

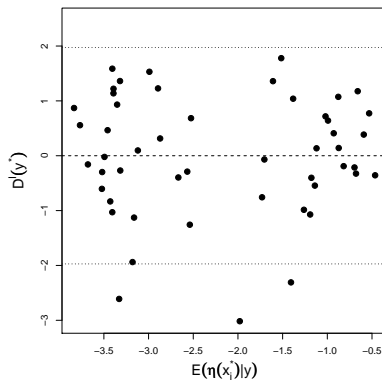
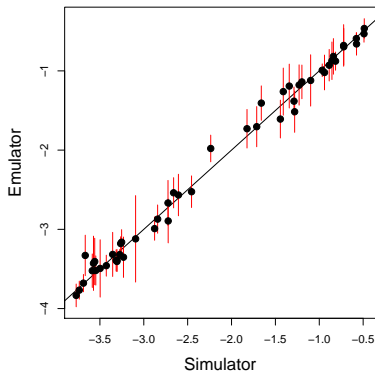


- The mean function $h(\cdot) = (1, \mathbf{x}, x_5^2, x_5^3, x_5^4)$
- Log transformation on outputs
- “new” dataset for validation

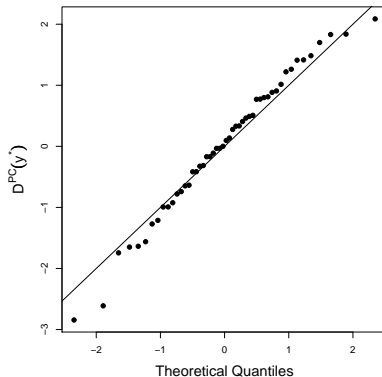
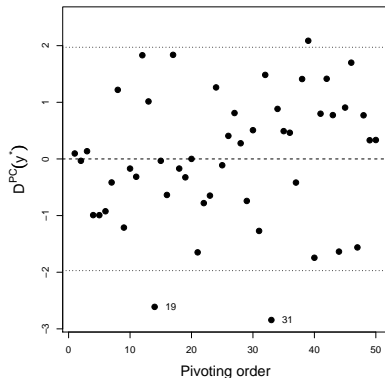
- The mean function $h(\cdot) = (1, \mathbf{x}, x_5^2, x_5^3, x_5^4)$
- Log transformation on outputs
- “new” dataset for validation

- The mean function $h(\cdot) = (1, \mathbf{x}, x_5^2, x_5^3, x_5^4)$
- Log transformation on outputs
- “new” dataset for validation

Individual errors



Uncorrelated Errors



$D_{MD}(\mathbf{y}^*) = 63.873$ and the 95% CI is (32.582, 79.508)

- 1 Revisão
- 2 Diagnósticos e Validação
 - Numerical Diagnostics
 - Exemplo: Modelo de lixo nuclear
 - Exemplo: Nilson-Kuusk model
- 3 Análises de Incerteza e Sensibilidade

Análises de incerteza e sensibilidade

- **Análise de Incerteza (UA)**

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

$$Y = \eta(X),$$

onde $X \sim \mathbb{G}$.

- **Análise de sensibilidade (SA)**

É usada para avaliar como cada input (ou grupo de inputs) afeta a distribuição dos outputs.

Normalmente, a SA local é baseada nos derivativos da função η em torno de um ponto de interesse.

A análise de sensibilidade global é baseada em grande número de amostras de inputs.

Os métodos clássicos de SA (Tornado (PSA) e o primeiro e segundo momentos induzidos nos outputs).

- UA/PSA podem ser feitas usando Monte Carlo.

Análises de incerteza e sensibilidade

- **Análise de Incerteza (UA)**

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

$$Y = \eta(X),$$

onde $X \sim \mathbb{G}$.

- **Análise de sensibilidade (SA)**

É usada para medir o impacto das incertezas (ou erros de input) sobre a distribuição da saída.

Existem dois tipos de SA: local e global. A análise local considera o impacto da incerteza de um único input sobre a incerteza da saída.

A análise de sensibilidade global é baseada em grande número de amostras produzidas.

● O método clássico é o de O'Neil (PSA) e produz a análise de sensibilidade induzida nos outputs.

- UA/PSA podem ser feitas usando Monte Carlo.

Análises de incerteza e sensibilidade

- Análise de Incerteza (UA)

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

Análises de incerteza e sensibilidade

- Análise de Incerteza (UA)

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

$$Y = \eta(X),$$

onde $X \sim \mathbb{G}$.

- Análise de sensibilidade (SA)

- É usada para entender como cada input (ou grupo de inputs) afeta a distribuição dos outputs.

• A análise de sensibilidade pode ser feita de duas maneiras: analisando a contribuição de cada input para a variância dos outputs (análise de primeira ordem) ou analisando a contribuição de cada input para a variância dos outputs (análise de segunda ordem).

• A análise de sensibilidade pode ser feita de duas maneiras: analisando a contribuição de cada input para a variância dos outputs (análise de primeira ordem) ou analisando a contribuição de cada input para a variância dos outputs (análise de segunda ordem).

- UA/PSA podem ser feitas usando Monte Carlo.

Análises de incerteza e sensibilidade

- **Análise de Incerteza (UA)**

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

$$Y = \eta(X),$$

onde $X \sim \mathbb{G}$.

- **Análise de sensibilidade (SA)**

- É usada para entender como cada input (ou grupo de inputs) afeta a distribuição dos outputs.
- Análise de sensibilidade local é baseada nos derivativos do simulador em torno de um ponto x_0 de interesse
- Análise de sensibilidade global é baseada em grande mudanças no input de interesse.
- Análise probabilística de sensibilidade (PSA) surge quando existe incerteza nos inputs.

- UA/PSA podem ser feitas usando Monte Carlo.

Análises de incerteza e sensibilidade

- **Análise de Incerteza (UA)**

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

$$Y = \eta(X),$$

onde $X \sim \mathbb{G}$.

- **Análise de sensibilidade (SA)**

- É usada para entender como cada input (ou grupo de inputs) afeta a distribuição dos outputs.
- Análise de sensibilidade local é baseada nos derivativos do simulador em torno de um ponto x_0 de interesse
- Análise de sensibilidade global é baseada em grande mudanças no input de interesse.
- Análise probabilística de sensibilidade (PSA) surge quando existe incerteza nos inputs.

- UA/PSA podem ser feitas usando Monte Carlo.

Análises de incerteza e sensibilidade

- **Análise de Incerteza (UA)**

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

$$Y = \eta(X),$$

onde $X \sim \mathbb{G}$.

- **Análise de sensibilidade (SA)**

- É usada para entender como cada input (ou grupo de inputs) afeta a distribuição dos outputs.
- Análise de sensibilidade local é baseada nos derivativos do simulador em torno de um ponto x_0 de interesse
- Análise de sensibilidade global é baseada em grande mudanças no input de interesse.
- Análise probabilística de sensibilidade (PSA) surge quando existe incerteza nos inputs.

- UA/PSA podem ser feitas usando Monte Carlo.

Análises de incerteza e sensibilidade

- **Análise de Incerteza (UA)**

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

$$Y = \eta(X),$$

onde $X \sim \mathbb{G}$.

- **Análise de sensibilidade (SA)**

- É usada para entender como cada input (ou grupo de inputs) afeta a distribuição dos outputs.
- Análise de sensibilidade local é baseada nos derivativos do simulador em torno de um ponto x_0 de interesse
- Análise de sensibilidade global é baseada em grandes mudanças no input de interesse.
- Análise probabilística de sensibilidade (PSA) surge quando existe incerteza nos inputs.

- UA/PSA podem ser feitas usando Monte Carlo.

Análises de incerteza e sensibilidade

- **Análise de Incerteza (UA)**

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

$$Y = \eta(X),$$

onde $X \sim \mathbb{G}$.

- **Análise de sensibilidade (SA)**

- É usada para entender como cada input (ou grupo de inputs) afeta a distribuição dos outputs.
- Análise de sensibilidade local é baseada nos derivativos do simulador em torno de um ponto x_0 de interesse
- Análise de sensibilidade global é baseada em grande mudanças no input de interesse.
- Análise probabilística de sensibilidade (PSA) surge quando existe incerteza nos inputs.

● UA/PSA podem ser feitas usando Monte Carlo.

Análises de incerteza e sensibilidade

- **Análise de Incerteza (UA)**

- Usada quando existe incerteza a respeito dos inputs
- A nossa incerteza pode ser descrita por uma distribuição de probabilidade \mathbb{G}
- o interesse então é conhecer a distribuição de probabilidade induzida nos outputs

$$Y = \eta(X),$$

onde $X \sim \mathbb{G}$.

- **Análise de sensibilidade (SA)**

- É usada para entender como cada input (ou grupo de inputs) afeta a distribuição dos outputs.
- Análise de sensibilidade local é baseada nos derivativos do simulador em torno de um ponto x_0 de interesse
- Análise de sensibilidade global é baseada em grande mudanças no input de interesse.
- Análise probabilística de sensibilidade (PSA) surge quando existe incerteza nos inputs.

- UA/PSA podem ser feitas usando Monte Carlo.

- Suponha que estamos interessados na média dos outputs

$$\mathbb{E}[Y] = \int_{\mathcal{X}} \eta(x) d\mathbb{G}(x)$$

- Um algoritmo de Monte Carlo para obter a distribuição de $\mathbb{E}[Y]$
 - Gere uma função aleatória $\eta(\eta)(\cdot)$ da distribuição de $\eta(\cdot)$
 - Gere amostras η_1, \dots, η_n independentes de $\eta(\cdot)$
 - Calcule as médias $\eta_1(Y), \dots, \eta_n(Y)$ da distribuição de $\eta(\cdot)$
 - $\{\eta_1(Y), \dots, \eta_n(Y)\}$ da distribuição de $\mathbb{E}[Y]$
- Esse procedimento é válido para obter qualquer estatística de Y
- Oakley and O'Hagan (2002) propoe um método alternativo ao Monte Carlo para obter a distribuição da função de distribuição e da função de densidade de Y .

- Suponha que estamos interessados na média dos outputs

$$\mathbb{E}[Y] = \int_{\mathcal{X}} \eta(x) d\mathbb{G}(x)$$

- Um algoritmo de Monte Carlo para obter a distribuição de $\mathbb{E}[Y]$

- 1 Gere uma função aleatória $\eta_{(i)}(\cdot)$ da distribuição de $\eta(\cdot)$
- 2 Obtenha $\mathbb{E}_{(i)}[Y] = \int_{\mathcal{X}} \eta_{(i)}(x) d\mathbb{G}(x)$
- 3 Repita os passos 1 e 2 até obter uma amostra $\{\mathbb{E}_{(1)}[Y], \dots, \mathbb{E}_{(M)}[Y]\}$ da distribuição de $\mathbb{E}[Y]$.

- Esse procedimento é válido para obter qualquer estatística de Y
- Oakley and O'Hagan (2002) propoe um método alternativo ao Monte Carlo para obter a distribuição da função de distribuição e da função de densidade de Y .

- Suponha que estamos interessados na média dos outputs

$$\mathbb{E}[Y] = \int_{\mathcal{X}} \eta(x) d\mathbb{G}(x)$$

- Um algoritmo de Monte Carlo para obter a distribuição de $\mathbb{E}[Y]$
 - 1 Gere uma função aleatória $\eta_{(i)}(\cdot)$ da distribuição de $\eta(\cdot)$
 - 2 Obtenha $\mathbb{E}_{(i)}[Y] = \int_{\mathcal{X}} \eta_{(i)}(x) d\mathbb{G}(x)$
 - 3 Repita os passos 1 e 2 até obter uma amostra $\{\mathbb{E}_{(1)}[Y], \dots, \mathbb{E}_{(N)}[Y]\}$ da distribuição de $\mathbb{E}[Y]$.
- Esse procedimento é válido para obter qualquer estatística de Y
- Oakley and O'Hagan (2002) propoe um método alternativo ao Monte Carlo para obter a distribuição da função de distribuição e da função de densidade de Y .

- Suponha que estamos interessados na média dos outputs

$$\mathbb{E}[Y] = \int_{\mathcal{X}} \eta(x) d\mathbb{G}(x)$$

- Um algoritmo de Monte Carlo para obter a distribuição de $\mathbb{E}[Y]$
 - 1 Gere uma função aleatória $\eta_{(i)}(\cdot)$ da distribuição de $\eta(\cdot)$
 - 2 Obtenha $\mathbb{E}_{(i)}[Y] = \int_{\mathcal{X}} \eta_{(i)}(x) d\mathbb{G}(x)$
 - 3 Repita os passos 1 e 2 até obter uma amostra $\{\mathbb{E}_{(1)}[Y], \dots, \mathbb{E}_{(N)}[Y]\}$ da distribuição de $\mathbb{E}[Y]$.
- Esse procedimento é válido para obter qualquer estatística de Y
- Oakley and O'Hagan (2002) propoe um método alternativo ao Monte Carlo para obter a distribuição da função de distribuição e da função de densidade de Y .

- Suponha que estamos interessados na média dos outputs

$$\mathbb{E}[Y] = \int_{\mathcal{X}} \eta(x) d\mathbb{G}(x)$$

- Um algoritmo de Monte Carlo para obter a distribuição de $\mathbb{E}[Y]$
 - 1 Gere uma função aleatória $\eta_{(i)}(\cdot)$ da distribuição de $\eta(\cdot)$
 - 2 Obtenha $\mathbb{E}_{(i)}[Y] = \int_{\mathcal{X}} \eta_{(i)}(x) d\mathbb{G}(x)$
 - 3 Repita os passos 1 e 2 até obter uma amostra $\{\mathbb{E}_{(1)}[Y], \dots, \mathbb{E}_{(N)}[Y]\}$ da distribuição de $\mathbb{E}[Y]$.
- Esse procedimento é válido para obter qualquer estatística de Y
- Oakley and O'Hagan (2002) propoe um método alternativo ao Monte Carlo para obter a distribuição da função de distribuição e da função de densidade de Y .

- Suponha que estamos interessados na média dos outputs

$$\mathbb{E}[Y] = \int_{\mathcal{X}} \eta(x) d\mathbb{G}(x)$$

- Um algoritmo de Monte Carlo para obter a distribuição de $\mathbb{E}[Y]$
 - 1 Gere uma função aleatória $\eta_{(i)}(\cdot)$ da distribuição de $\eta(\cdot)$
 - 2 Obtenha $\mathbb{E}_{(i)}[Y] = \int_{\mathcal{X}} \eta_{(i)}(x) d\mathbb{G}(x)$
 - 3 Repita os passos 1 e 2 até obter uma amostra $\{\mathbb{E}_{(1)}[Y], \dots, \mathbb{E}_{(N)}[Y]\}$ da distribuição de $\mathbb{E}[Y]$.
- Esse procedimento é válido para obter qualquer estatística de Y
- Oakley and O'Hagan (2002) propoe um método alternativo ao Monte Carlo para obter a distribuição da função de distribuição e da função de densidade de Y .

- Suponha que estamos interessados na média dos outputs

$$\mathbb{E}[Y] = \int_{\mathcal{X}} \eta(x) d\mathbb{G}(x)$$

- Um algoritmo de Monte Carlo para obter a distribuição de $\mathbb{E}[Y]$
 - 1 Gere uma função aleatória $\eta_{(i)}(\cdot)$ da distribuição de $\eta(\cdot)$
 - 2 Obtenha $\mathbb{E}_{(i)}[Y] = \int_{\mathcal{X}} \eta_{(i)}(x) d\mathbb{G}(x)$
 - 3 Repita os passos 1 e 2 até obter uma amostra $\{\mathbb{E}_{(1)}[Y], \dots, \mathbb{E}_{(N)}[Y]\}$ da distribuição de $\mathbb{E}[Y]$.
- Esse procedimento é válido para obter qualquer estatística de Y
- Oakley and O'Hagan (2002) propoe um método alternativo ao Monte Carlo para obter a distribuição da função de distribuição e da função de densidade de Y .

Análise probabilística de sensibilidade

- Suponha que estamos interessados nos efeitos dos inputs individualmente.
- Seja $z_i(x_i)$ o efeito principal do i -ésimo input

$$z_i(x_i) = \mathbb{E}[Y|x_i] - \mathbb{E}[Y]$$

onde $\mathbb{E}[Y|x_i] = \int_{\mathcal{X}_{-i}} \eta(x) dG(\mathbf{x}_{-i}|x_i)$.

- Outra quantidade de interesse é

$$V_i = \mathbb{V}ar\{\mathbb{E}[Y|x_i]\}$$

- Um gráfico de cada input x_i versus seu efeito principal $z_i(x_i)$, nos ajuda visualizar a influência de cada input.
- Para mais detalhes veja em Oakley e O'Hagan (2004)

Análise probabilística de sensibilidade

- Suponha que estamos interessados nos efeitos dos inputs individualmente.
- Seja $z_i(x_i)$ o efeito principal do i -ésimo input

$$z_i(x_i) = \mathbb{E}[Y|x_i] - \mathbb{E}[Y]$$

onde $\mathbb{E}[Y|x_i] = \int_{\mathcal{X}_{-i}} \eta(\mathbf{x}) d\mathbb{G}(\mathbf{x}_{-i}|x_i)$.

- Outra quantidade de interesse é

$$V_i = \text{Var}\{\mathbb{E}[Y|x_i]\}$$

- Um gráfico de cada input x_i versus seu efeito principal $z_i(x_i)$, nos ajuda visualizar a influência de cada input.
- Para mais detalhes veja em Oakley e O'Hagan (2004)

Análise probabilística de sensibilidade

- Suponha que estamos interessados nos efeitos dos inputs individualmente.
- Seja $z_i(x_i)$ o efeito principal do i -ésimo input

$$z_i(x_i) = \mathbb{E}[Y|x_i] - \mathbb{E}[Y]$$

onde $\mathbb{E}[Y|x_i] = \int_{\mathcal{X}_{-i}} \eta(x) d\mathbb{G}(\mathbf{x}_{-i}|x_i)$.

- Outra quantidade de interesse é

$$V_i = \mathbb{V}ar\{\mathbb{E}[Y|x_i]\}$$

- Um gráfico de cada input x_i versus seu efeito principal $z_i(x_i)$, nos ajuda visualizar a influência de cada input.
- Para mais detalhes veja em Oakley e O'Hagan (2004)

Análise probabilística de sensibilidade

- Suponha que estamos interessados nos efeitos dos inputs individualmente.
- Seja $z_i(x_i)$ o efeito principal do i -ésimo input

$$z_i(x_i) = \mathbb{E}[Y|x_i] - \mathbb{E}[Y]$$

onde $\mathbb{E}[Y|x_i] = \int_{\mathcal{X}_{-i}} \eta(x) d\mathbb{G}(\mathbf{x}_{-i}|x_i)$.

- Outra quantidade de interesse é

$$V_i = \mathbb{V}ar\{\mathbb{E}[Y|x_i]\}$$

- Um gráfico de cada input x_i versus seu efeito principal $z_i(x_i)$, nos ajuda visualizar a influência de cada input.
- Para mais detalhes veja em Oakley e O'Hagan (2004)

Exemplos

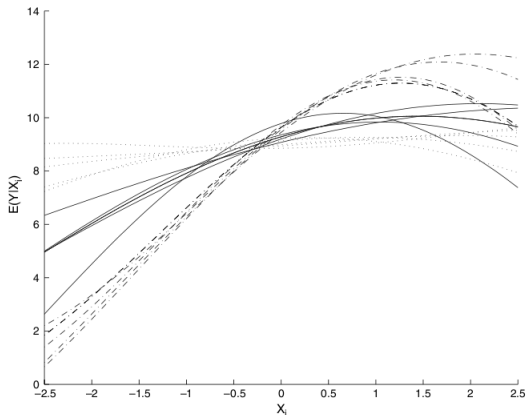


Fig. 1. Posterior expectation of $E(Y|X_i)$ against x_i for each input variable: synthetic example (....., X_1, X_2, X_3, X_4, X_5 ; —, $X_6, X_7, X_8, X_9, X_{10}$; ---, $X_{11}, X_{12}, X_{13}, X_{14}, X_{15}$)

Exemplos

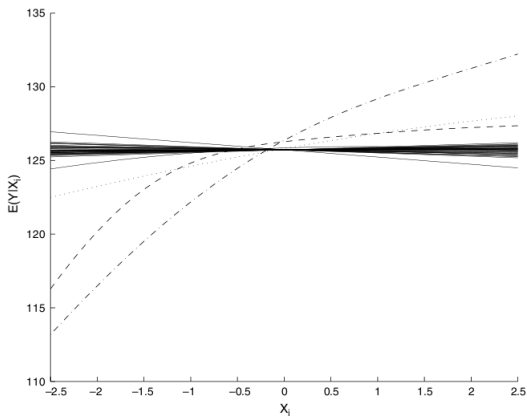


Fig. 3. Posterior expectation of $E(Y|x_i)$ against x_i for each input variable: reservoir example ($\cdots\cdots$, X_2 ; $\cdots\cdots$, X_3 ; $-\cdots-$, X_4 ; $---$, others)

- Suponha que estamos interessados nos efeitos dos inputs individualmente.
- Seja $z_i(x_i)$ o efeito principal do i -ésimo input

$$z_i(x_i) = \mathbb{E}[Y|x_i] - \mathbb{E}[Y]$$

onde $\mathbb{E}[Y|x_i] = \int_{\mathcal{X}_{-i}} \eta(x) d\mathbb{G}(\mathbf{x}_{-i}|x_i)$.

- Outra quantidade de interesse é

$$V_i = \mathbb{V}ar\{\mathbb{E}[Y|x_i]\}$$

- Um gráfico de cada input x_i versus seu efeito principal $z_i(x_i)$, nos ajuda visualizar a influência de cada input.
- Para mais detalhes veja em Oakley e O'Hagan (2004)

- Vimos alguns diagnósticos para validar um emulador gaussiano
- onde idéia principal é fazer uma análise de resíduos para um process gaussiano
 - Diagnósticos gráficos e numéricos
- Vimos brevemente as análises de incerteza e sensibilidade em experimentos computacionais

• Para essas análises, a simulação é fundamental, pois permite reduzir a variabilidade inerente à imitação com sucesso.

- Vimos alguns diagnósticos para validar um emulador gaussiano
- onde idéia principal é fazer uma análise de resíduos para um process gaussiano
 - Diagnósticos gráficos e numéricos
- Vimos brevemente as análises de incerteza e sensibilidade em experimentos computacionais

- Vimos alguns diagnósticos para validar um emulador gaussiano
- onde idéia principal é fazer uma análise de resíduos para um process gaussiano
 - Diagnósticos gráficos e numéricos
- Vimos brevemente as análises de incerteza e sensibilidade em experimentos computacionais
 - Para essas análises, a emulação é fundamental, pois pode reduzir consideravelmente o tempo computacional.

- Vimos alguns diagnósticos para validar um emulador gaussiano
- onde idéia principal é fazer uma análise de resíduos para um process gaussiano
 - Diagnósticos gráficos e numéricos
- Vimos brevemente as análises de incerteza e sensibilidade em experimentos computacionnais
 - Para essas análises, a emulação é fundamental, pois pode reduzir consideravelmente o tempo computacional.

- Vimos alguns diagnósticos para validar um emulador gaussiano
- onde idéia principal é fazer uma análise de resíduos para um process gaussiano
 - Diagnósticos gráficos e numéricos
- Vimos brevemente as análises de incerteza e sensibilidade em experimentos computacionnais
 - Para essas análises, a emulação é fundamental, pois pode reduzir consideravelmente o tempo computacional.

- **Approximate Bayesian Computation**

- É uma técnica de inferência quando não temos a função de verossimilhança, $\mathbb{P}(D|\theta)$, mas sabemos gerar dela.
- A idéia básica vem do seguinte algoritmo:

Generate $\theta_1, \dots, \theta_n$ i.i.d. from $q(\theta)$

Compute $\mathbb{P}(D|\theta_i)$

- Accept θ_i as $D \sim D$

- Obrigado!

- Approximate Bayesian Computation
- É uma técnica de inferência quando não temos a função de verossimilhança, $\mathbb{P}(D|\theta)$, mas sabemos gerar dela.
- A idéia básica vem do seguinte algoritmo:
 - Gere θ de uma distribuição $\pi(\cdot)$
 - Avalie $\mathbb{P}(D|\theta)$
 - Aceite θ com probabilidade α
- Obrigado!

- Approximate Bayesian Computation
- É uma técnica de inferência quando não temos a função de verossimilhança, $\mathbb{P}(D|\theta)$, mas sabemos gerar dela.
- A idéia básica vem do seguinte algoritmo:
 - 1 Gere θ de uma distribuição $\pi(\cdot)$
 - 2 Gere D' de $\mathbb{P}(\cdot|\theta)$
 - 3 Aceite θ se $D' \approx D$
- Obrigado!

- Approximate Bayesian Computation
- É uma técnica de inferência quando não temos a função de verossimilhança, $\mathbb{P}(D|\theta)$, mas sabemos gerar dela.
- A idéia básica vem do seguinte algoritmo:
 - 1 Gere θ de uma distribuição $\pi(\cdot)$
 - 2 Gere D' de $\mathbb{P}(\cdot|\theta)$
 - 3 Aceite θ se $D' \approx D$
- Obrigado!

- Approximate Bayesian Computation
- É uma técnica de inferência quando não temos a função de verossimilhança, $\mathbb{P}(D|\theta)$, mas sabemos gerar dela.
- A idéia básica vem do seguinte algoritmo:
 - 1 Gere θ de uma distribuição $\pi(\cdot)$
 - 2 Gere D' de $\mathbb{P}(\cdot|\theta)$
 - 3 Aceite θ se $D' \approx D$
- Obrigado!

- Approximate Bayesian Computation
- É uma técnica de inferência quando não temos a função de verossimilhança, $\mathbb{P}(D|\theta)$, mas sabemos gerar dela.
- A idéia básica vem do seguinte algoritmo:
 - 1 Gere θ de uma distribuição $\pi(\cdot)$
 - 2 Gere D' de $\mathbb{P}(\cdot|\theta)$
 - 3 Aceite θ se $D' \approx D$
- Obrigado!

- Approximate Bayesian Computation
- É uma técnica de inferência quando não temos a função de verossimilhança, $\mathbb{P}(D|\theta)$, mas sabemos gerar dela.
- A idéia básica vem do seguinte algoritmo:
 - 1 Gere θ de uma distribuição $\pi(\cdot)$
 - 2 Gere D' de $\mathbb{P}(\cdot|\theta)$
 - 3 Aceite θ se $D' \approx D$
- Obrigado!