제 15회 KRUG Meetup

*2014/10/30*

# Anomaly detection using R

MULTIVARIATE GAUSSIAN DISTRIBUTION BASED ANOMALY DETECT

**유 충 현**

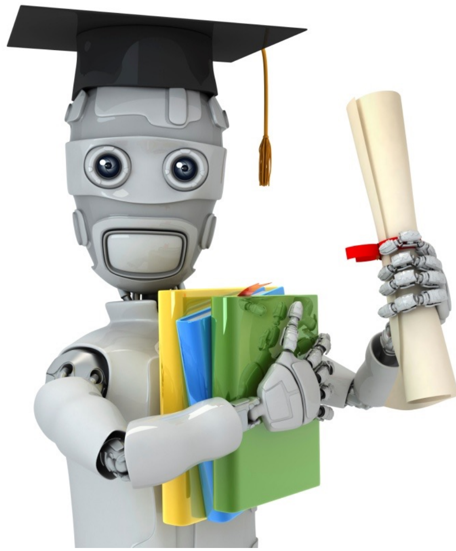
**KRUG**

# AGENDA

- Problem motivation
- Gaussian distribution
- Algorithm
- Anomaly detection using the multivariate Gaussian distribution
- Developing and evaluating an anomaly detection system
- anomalyDensityEastimation class

# 일러두기

- 스라이드 및 이론 설명
  - Coursera 강좌 중 Machine Learning
- 구현 및 예제
  - S4를 적용하여 구현함
  - Detection function / Visualization function



Machine Learning

# Anomaly detection example

Aircraft engine features:

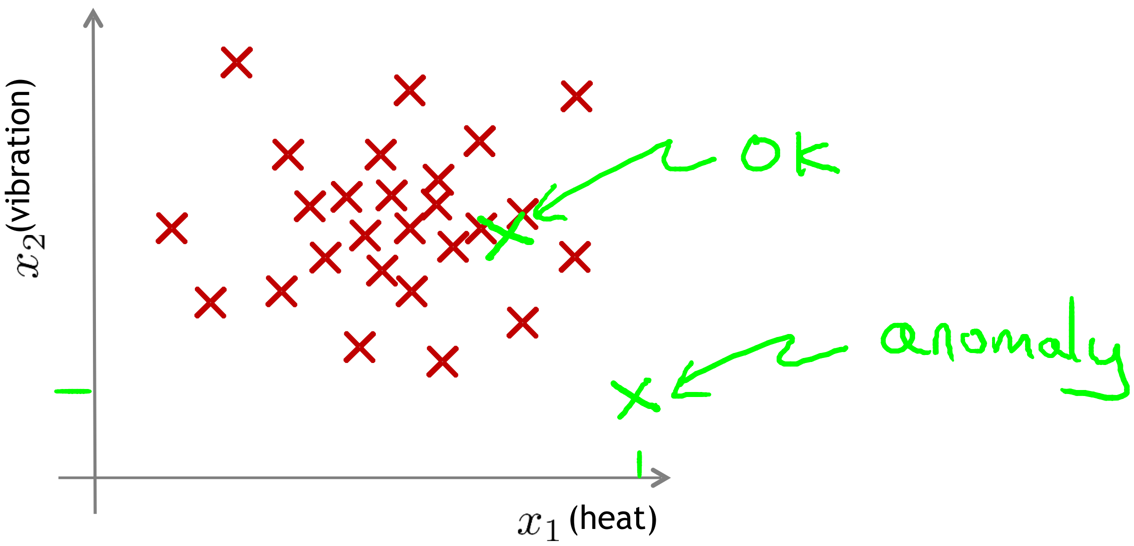
$x_1$ = heat generated

$x_2$ = vibration intensity

...

Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

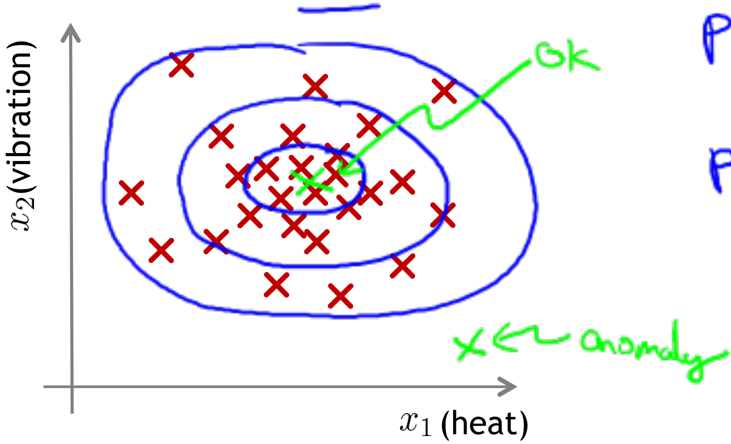
New engine: $x_{test}$



# Density estimation

- Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
- Is $x_{test}$ anomalous?

Model p(x).




$p(x_{test}) < \varepsilon \rightarrow$ flag anomaly

$p(x_{test}) \geq \varepsilon \rightarrow$ OK

# Anomaly detection example

Fraud detection:

$x^{(i)}$ = features of user $i$'s activities

Model $p(x)$ from data.

Identify unusual users by checking which have $p(x) < \varepsilon$

Manufacturing

Monitoring computers in a data center.

$x^{(i)}$ = features of machine $i$

$x_1$ = memory use, $x_2$ = number of disk accesses/sec,

$x_3$ = CPU load, $x_4$ = CPU load/network traffic.

…

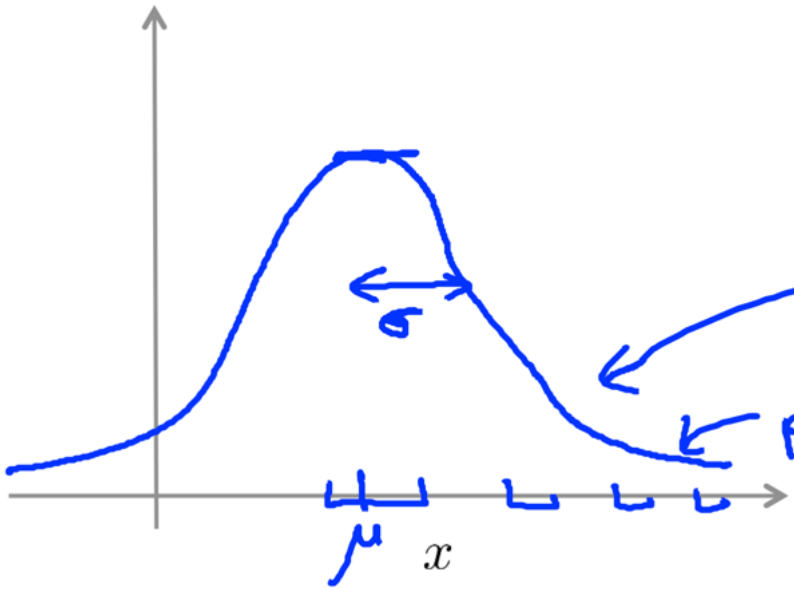
# Gaussian(Normal) distribution

## Gaussian (Normal) distribution

Say $x \in \mathbb{R}$. If $x$ is a distributed Gaussian with mean $\mu$ , variance $\sigma^2$ .

$x \sim \mathcal{N}(\mu, \sigma^2)$ "distributed as"

$\sigma$ standard deviation

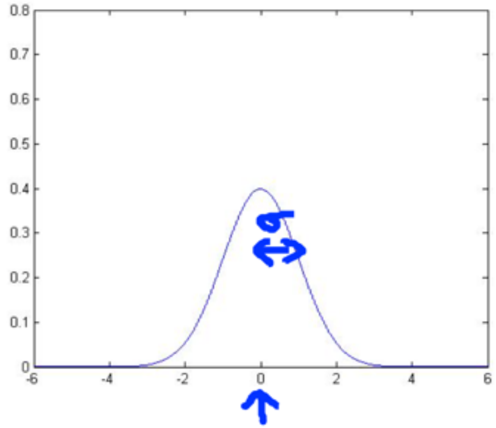



$p(x; \mu, \sigma^2)$

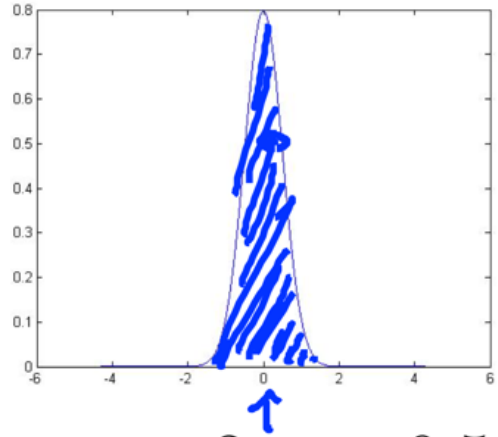
$$p(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

# Gaussian distribution example

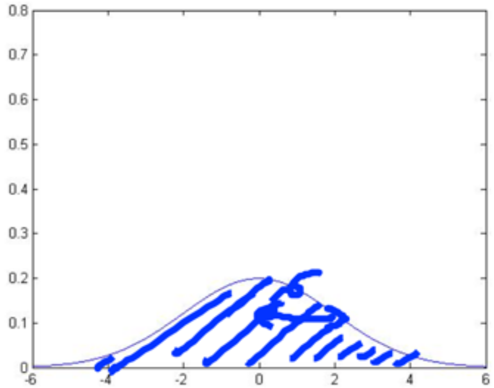
$\mu = 0, \sigma = 1$



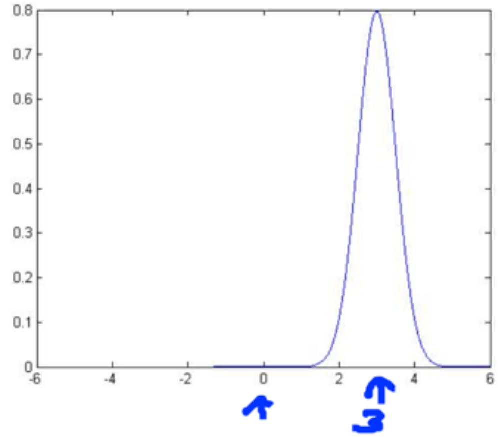

$\mu = 0, \sigma = \underline{0.5}$




$\mu = 0, \sigma = 2$



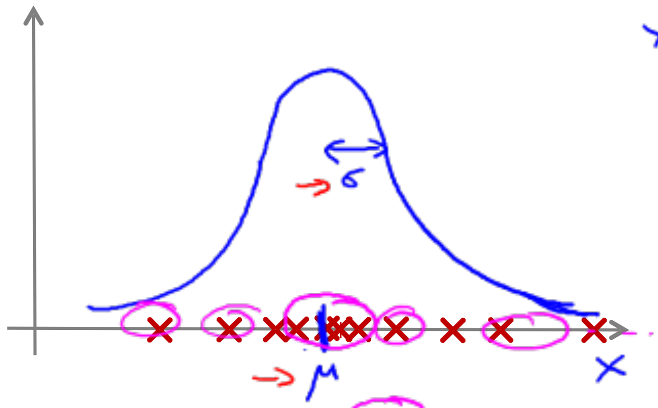

$\mu = 3, \sigma = 0.5$



# Parameter estimation

Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ $\quad x^{(i)} \in \mathbb{R}$

$x^{(i)} \sim \mathcal{N}(\mu, \sigma^2)$




$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} \left(x^{(i)} - \mu\right)^2$$

$m-1$ $\quad \frac{1}{m-1}$

# Density estimation

$\rightarrow$

$\rightarrow$ Training set: $\{x^{(1)}, \ldots, x^{(m)}\}$
Each example is $\underline{x} \in \mathbb{R}^n$

$x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$
$x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$
$x_3 \sim \mathcal{N}(\mu_3, \sigma_3^2)$

$$\rightarrow p(x)$$
$$= \boxed{p(x_1; \mu_1, \sigma_1^2)\, p(x_2; \mu_2, \sigma_2^2)\, p(x_3; \mu_3, \sigma_3^2) \cdots p(x_n; \mu_n, \sigma_n^2)} \leftarrow$$
$$= \boxed{\prod_{j=1}^{n}} \underline{p(x_j; \mu_j, \sigma_j^2)}$$

$$\sum_{i=1}^{n} i = 1+2+3+\cdots+n$$

$$\prod_{i=1}^{n} i = 1 \times 2 \times 3 \times \cdots \times n$$

# Anomaly detection algorithm

1. Choose features $x_i$ that you think might be indicative of anomalous examples.
2. Fit parameters $\mu_1, \ldots, \mu_n, \sigma_1^2, \ldots, \sigma_n^2$

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)}$$

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^{m} (x_j^{(i)} - \mu_j)^2$$

3. Given new example $x$, compute $p(x)$ :

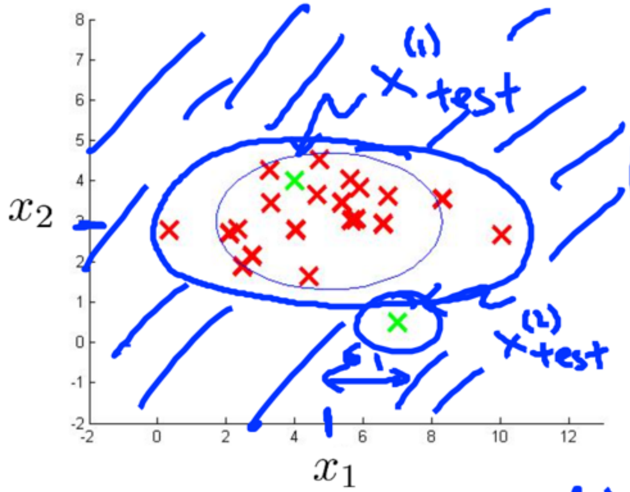
$$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

Anomaly if $p(x) < \varepsilon$

# Anomaly detection example

$\rightarrow p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2; \mu_2, \sigma_2^2)$

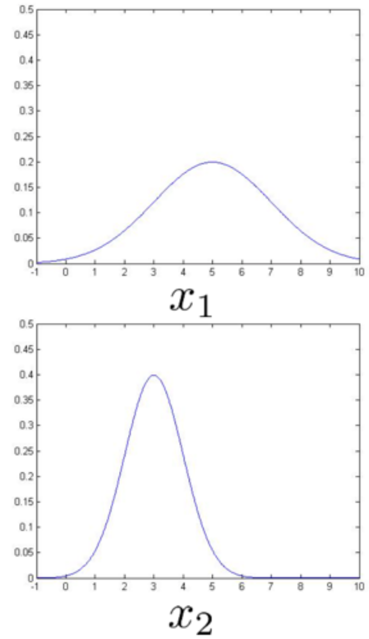
## Anomaly detection example




$\sigma_1^2, \sigma_2^2$

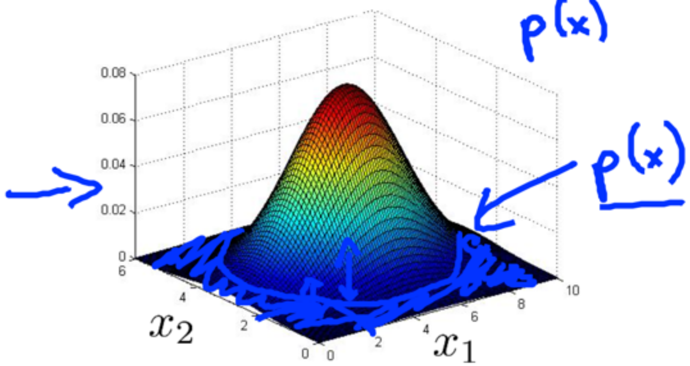
$= 4$

$$\boxed{\begin{aligned} \mu_1 &= 5, \underline{\sigma_1} = 2 \\ \mu_2 &= 3, \underline{\sigma_2} = 1 \end{aligned}}$$




$\underline{p(x_1; \mu_1, \sigma_1^2)}$

$\underline{p(x_2; \mu_2, \sigma_2^2)}$




$\underline{\varepsilon = 0.02}$

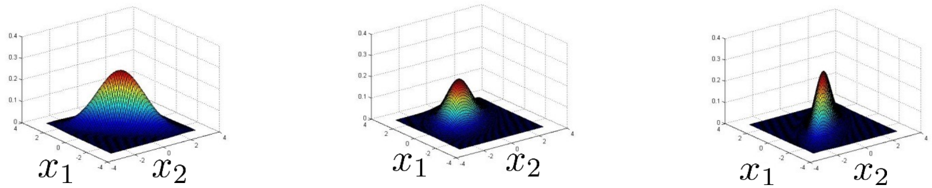
$p(x_{test}^{(1)}) = \underline{0.0426} \geq \varepsilon$

$p(x_{test}^{(2)}) = \underline{0.0021} < \varepsilon$

# Multivariate Gaussian distribution

Parameters $\mu, \Sigma$

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$




Parameter fitting:

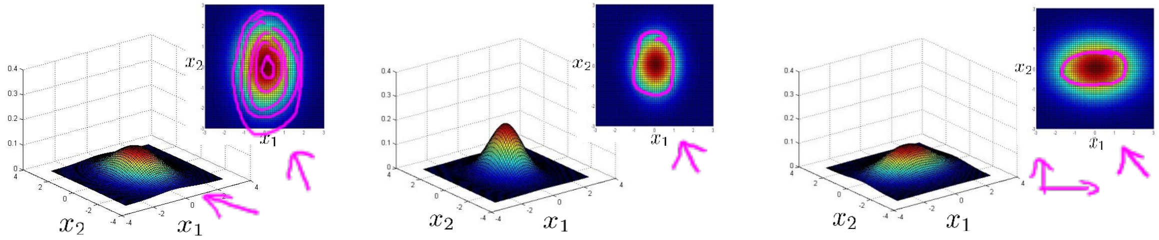
Given training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \qquad \Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\mu)(x^{(i)}-\mu)^T$$

# Relationship to original model

Original model: $p(x) = p(x_1;\mu_1,\sigma_1^2) \times p(x_2;\mu_2,\sigma_2^2) \times \cdots \times p(x_n;\mu_n,\sigma_n^2)$

$p(x)$



$\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$

Corresponds to multivariate Gaussian

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

where $\Sigma = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \sigma_2^2 & \\ & & \ddots & \\ 0 & & & \sigma_n^2 \end{bmatrix}$

# The importance of real-number evalution

When developing a learning algorithm (choosing features, etc.), making decisions is much easier if we have a way of evaluating our learning algorithm.

Assume we have some labeled data, of anomalous and non-anomalous examples. ( $y = 0$ if normal, $y = 1$ if anomalous).

Training set: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ (assume normal examples/ not anomalous)

Cross validation set: $(x_{cv}^{(1)}, y_{cv}^{(1)}), \ldots, (x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$
Test set: $(x_{test}^{(1)}, y_{test}^{(1)}), \ldots, (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

# Aircraft engines example

**Aircraft engines motivating example**

10000 good (normal) engines
20 flawed engines (anomalous)

Training set: 6000 good engines
CV: 2000 good engines ( $y = 0$ ), 10 anomalous ( $y = 1$ )
Test: 2000 good engines ( $y = 0$ ), 10 anomalous ( $y = 1$ )

Alternative:
Training set: 6000 good engines
CV: 4000 good engines ( $y = 0$ ), 10 anomalous ( $y = 1$ )
Test: 4000 good engines ( $y = 0$ ), 10 anomalous ( $y = 1$ )

# Algorithm evalution

- Fit model $p(x)$ on training set $\{x^{(1)}, \ldots, x^{(m)}\}$
- On a cross validation/test example $x$ , predict

$(x^{(i)}_{test}, y^{(i)}_{test})$

$$y = \begin{cases} 1 & \text{if } p(x) < \varepsilon \text{ (anomaly)} \\ 0 & \text{if } p(x) \geq \varepsilon \text{ (normal)} \end{cases}$$

$y=0$

Possible evaluation metrics:

- - True positive, false positive, false negative, true negative
- - Precision/Recall
- - $F_1$-score

CV

Test set

Can also use cross validation set to choose parameter $\varepsilon$

# Algorithm evalution

$$prec = \frac{tp}{tp + fp} \tag{4}$$

$$rec = \frac{tp}{tp + fn}, \tag{5}$$

where

- $tp$ is the number of true positives: the ground truth label says it's an anomaly and our algorithm correctly classified it as an anomaly.
- $fp$ is the number of false positives: the ground truth label says it's not an anomaly, but our algorithm incorrectly classified it as an anomaly.
- $fn$ is the number of false negatives: the ground truth label says it's an anomaly, but our algorithm incorrectly classified it as not being anomalous.

$$F_1 = \frac{2 \cdot prec \cdot rec}{prec + rec},$$

# class

- anomalyDensityEastimation
  - S4 class
  - detection & visualization
- slots
  - X : data. matrix object
  - param : density parameter. list object (mu, sigma2)
  - p : probability
  - y : training set's anomaly flag.  vector (0=normality, 1=anomaly)
  - threshold : epsilon. list object
  - anomaly : detected observations

# methods

- estimateGaussian
  - 확률밀도함수의 모수추정
- multivariateGaussian
  - multivariate Gaussian 확률 계산
- select Threshold
  - best epsilon, best F1 계산
- findAnomaly
  - anomaly detection
- plot
  - 2D multivariate Gaussian plotting

# paramter estimation

```
> anomaly.trian <- new("anomalyDensityEastimation", X = X)
> anomaly.trian@param <- estimateGaussian(anomaly.trian)
> anomaly.trian@param
$mu
throughput    latency
  14.11223   14.99771


$sigma2
         [,1]     [,2]
[1,] 1.832631 0.000000
[2,] 0.000000 1.709745
```

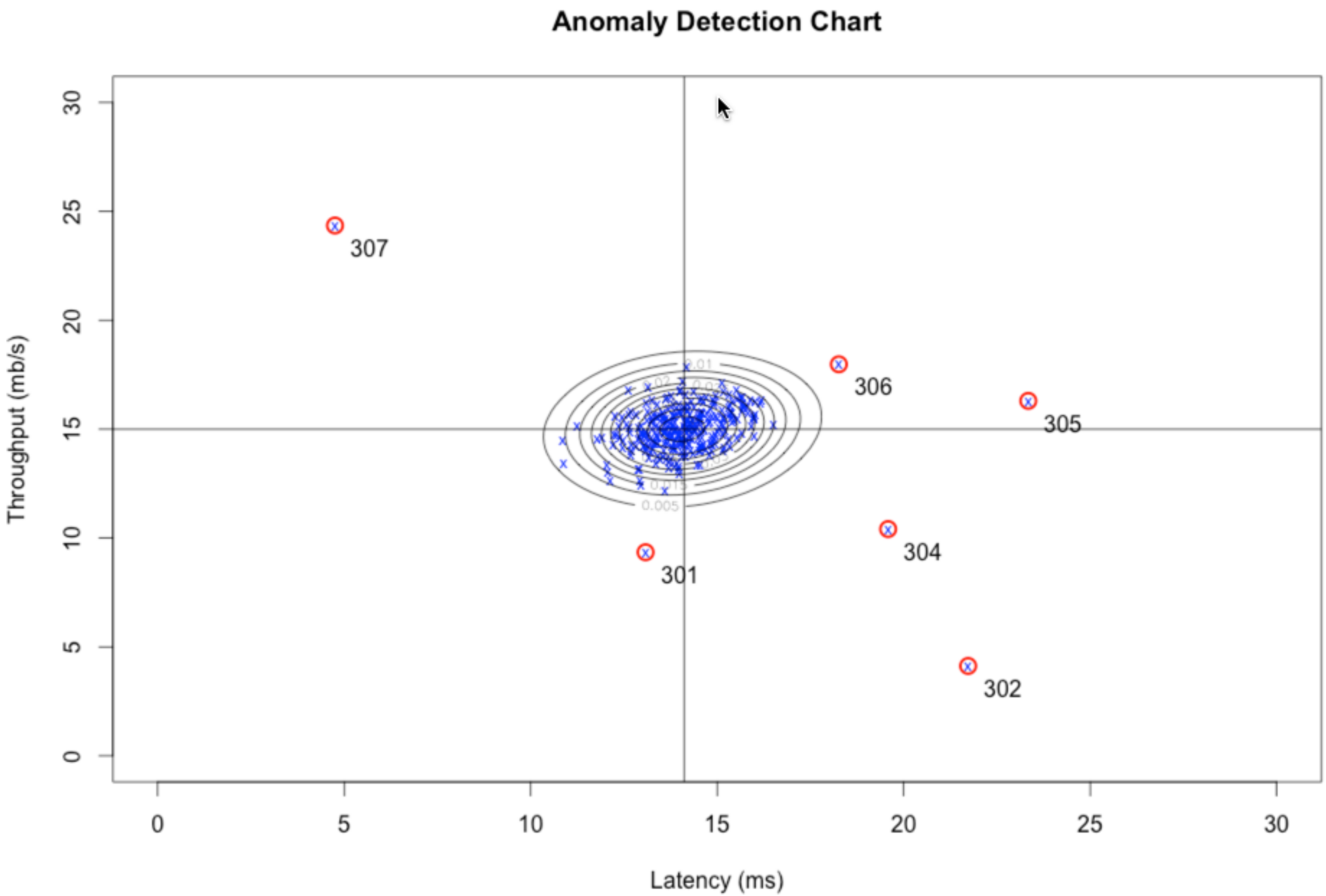
# select threshold

```
> anomaly.cv@threshold <- selectThreshold(anomaly.cv)
> anomaly.cv@threshold
$eps
[1] 8.990853e-05

$F1
[1] 0.875
```

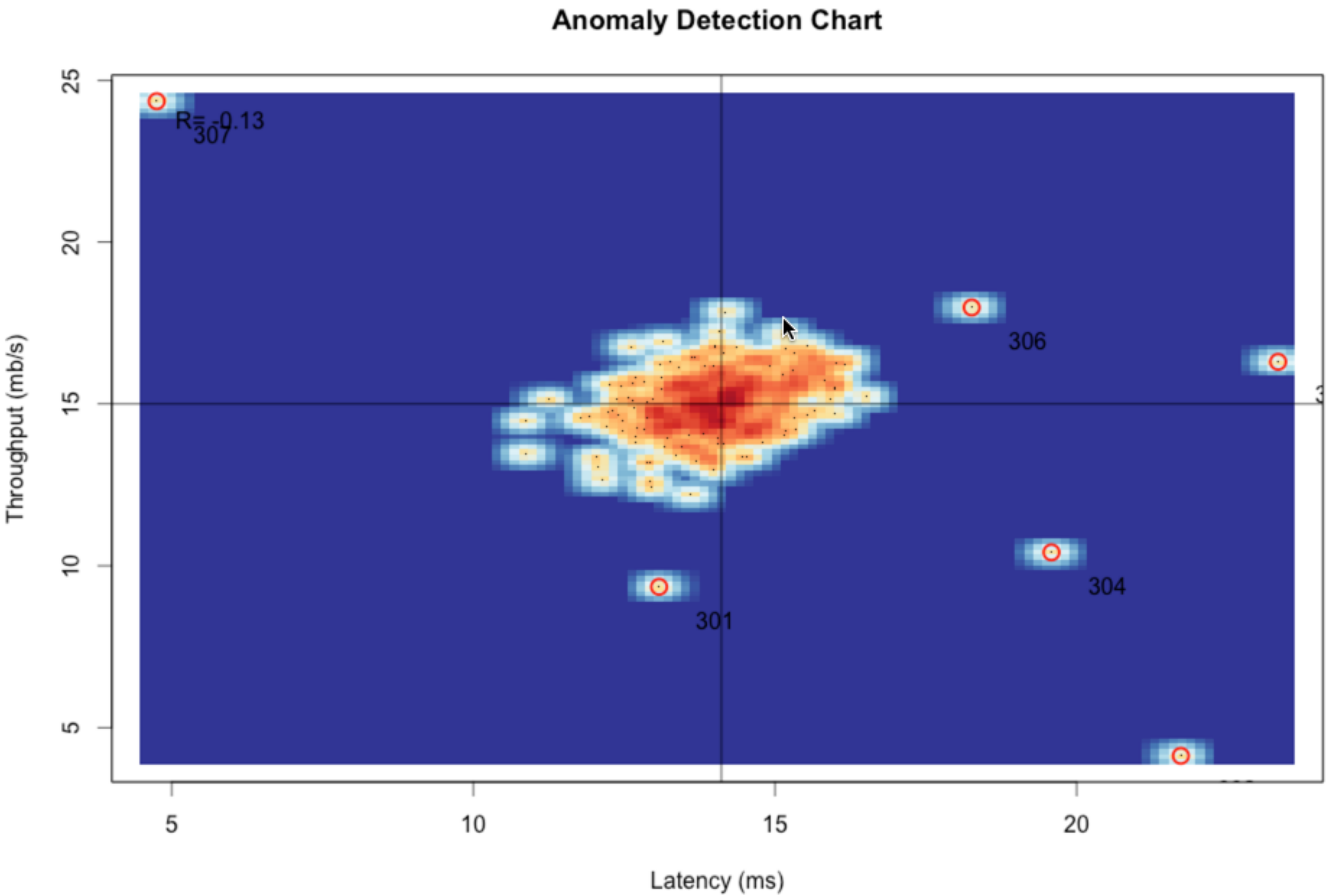
# anomaly detect

```
> anomaly.trian@p <- multivariateGaussian(anomaly.trian)
> anomaly.trian@anomaly <- findAnomaly(anomaly.trian, anomaly.cv@threshold)
> anomaly.trian@anomaly
$idx
[1] 301 302 304 305 306 307

$value
     throughput   latency
[1,]  13.079310  9.347878
[2,]  21.727134  4.126232
[3,]  19.582573 10.411619
[4,]  23.339868 16.298874
[5,]  18.261188 17.978309
[6,]   4.752613 24.350407
```

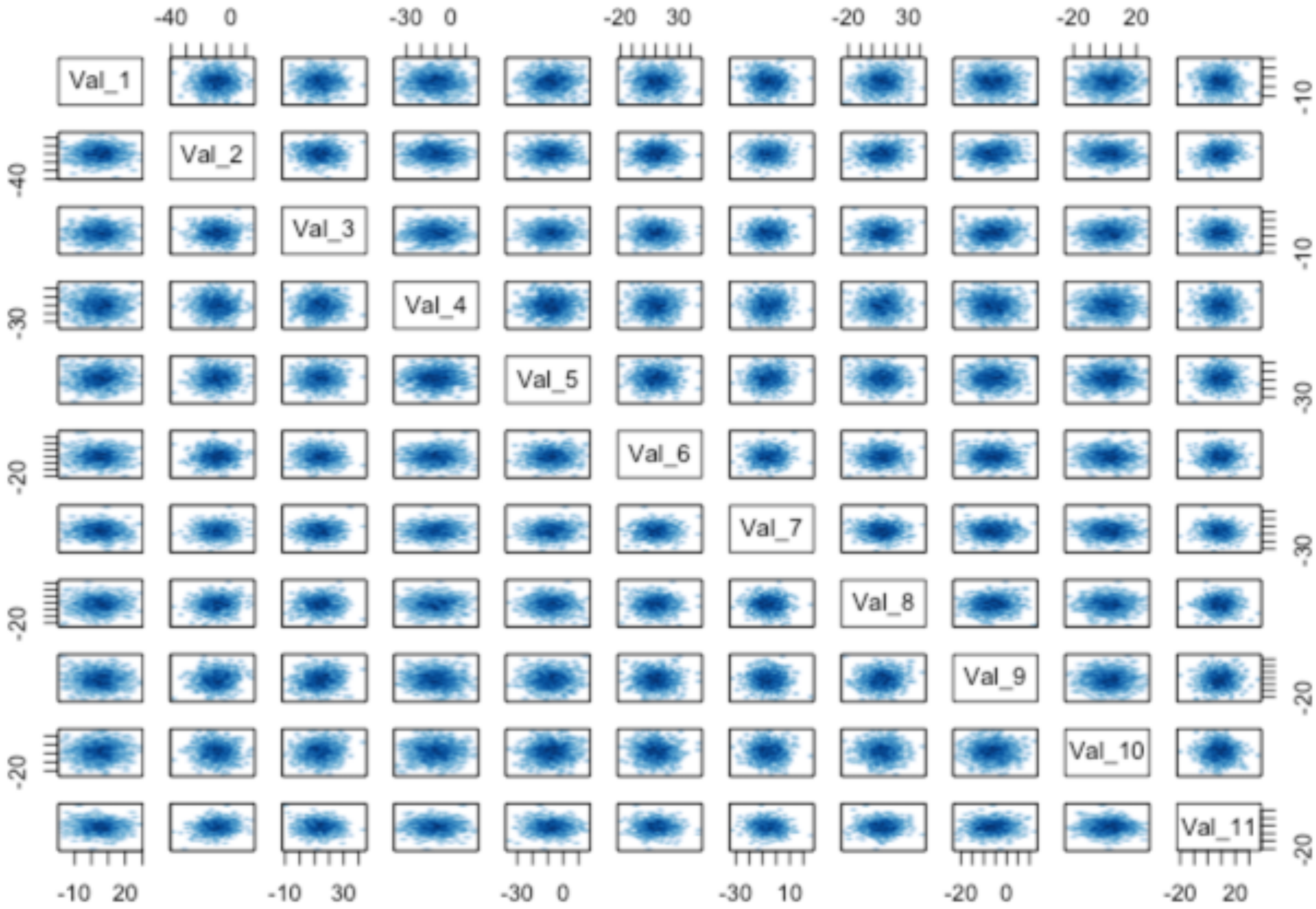
# plotting visualization



# plotting visualization



# plotting visualization



# Demos

감사합니다