# 1 Generating Test Data

When we implement or test various methods for machine learning (e.g. neural networks) we need the ability to generate data with prescribed properties and also to generate "random" data or randomly select data from a given set of data.

For generating random data, we need to know probability distribution fo the data:

(a) for a random variable $V$ with the *uniform probability distribution* on an interval $\langle A, B\rangle$ it holds:

$$\text{the probability that } V = x \text{ is } P(V = x) = \begin{cases} \frac{1}{B-A} & \text{for } x \in \langle A, B\rangle \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

An $m\times n$ matrix of data from the uniform distribution on the interval$\langle A, B\rangle$ can be generated as

```
>> A=3; B=10;
>> m=3; n=4;
>> (B-A)*rand(m,n)+A
ans =

  3.31198  6.28767  7.37211  8.75329
  5.75145  9.82846  8.29843  3.21501
  7.42435  4.56690  5.99315  6.79303
```

(b) For a random variable $V$ with the *normal (or Gauss) distribution* with mean $\mu$ and variance $\sigma^2$ it holds:

$$\text{the probability that } V = x \text{ is } P(V = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

For the random variable $V$ it holds that with the probability 95% its value is from the interval $\langle \mu - 2\sigma, \mu + 2\sigma\rangle$.

The function `randn` generates random numbers from the normal distribution with mean 0 and variance 1 (i.e. standard deviation $\sqrt{1} = 1$). Matrix of size $m \times n$ with numbers from the normal distribution with mean `S` and variance $R$ (i.e. standard deviation $\sqrt{\mathtt{R}}$) can be generated as follows

```
>> S=2; R=3;
>> m=3; n=4;
>> sqrt(R)*randn(m,n)+S
ans =

   1.76953   1.02155   2.90496   3.54724
   1.81023   9.41794  -1.39530   3.38295
   4.53440  -2.64642   3.07368   0.76086
```

## 1.1 Generating Vector with Two Clusters

Implement a function `randv2n(n1,S1,R1,n2,S2,R2)`, which generates one row vector containing `n1` numbers from the normal distribution with mean `S1` and variance `R1` and another `n2` numbers from the normal distribution with mean `S2` and variance `R2`.

Example:

```
>> randv2n(3,-10,1,4,10,1)
ans =

  1.0e+01  *

   0.86684   0.97688  -0.94214  -0.85999   1.09027   1.09225  -1.19626
```

Note that in the returned vector the numbers from the two clusters are randomly mixed. This can be achieved using the function `randperm` – see `help`. Demonstrate your implementation by constructing histogram (function `hist`) of the returned vector. How can we set-up the number of bins in the histogram?

## 1.2 Generating Clusters in 2D

Implement a function `randn2d(p)`, with parameter `p` being a matrix of size $5 \times n$. The function should generate a matrix of size $2 \times \sum_{i=1}^{n} \texttt{p(1,}i\texttt{)}$ containing random points in 2D. Each column of the resulting matrix is interpreted as the coordinates of a point on a plane. In the resulting matrix, for each $i = 1, \ldots, n$, there is `p(1,`$i$`)` columns from a cluster with the normal distribution with mean `p(2,`$i$`)` and variance `p(4,`$i$`)` in the first coordinate, and with mean `p(3,`$i$`)` and variance `p(5,`$i$`)` in the second coordinate.

Example:

```
>> rm=randn2d([6,8; -10,10; -10,10; 1,2; 1,2])
rm =

  Columns 1 through 5

   -9.3723  -10.8649    9.9903   -9.6808  -10.1649
  -11.1135  -10.8637   10.7814   -8.9067  -11.2141

  Columns 6 through 10

   10.5252    8.9115    8.4598   11.5802   12.1675
   10.1215   12.1838    8.4987    8.9502   11.5565

  Columns 11 through 14

   -9.6871    9.6810   10.0460  -10.0301
   -8.8907    7.8906   13.3240   -9.9226
```

Extend the function by adding an optional second parameter. If the function is called with two parameters, then before returning the resulting matrix, it plots a graph with graphically distinguished clusters, e.g. using the function `scatter`.

### 1.3 Generating a Sample from Data

Write a function `selectk(x,k)`, which randomly selects `k` columns from the input matrix `x`.

Example:

```
>> rm=rand(2,8)
rm =

  0.59672  0.25610  0.40015  0.22067  0.08944  0.20306  0.96964  0.80791
  0.48115  0.60608  0.13022  0.57005  0.33733  0.21467  0.57249  0.33636

>> srm=selectk(rm,4)
srm =

  0.22067  0.80791  0.25610  0.08944
  0.57005  0.33636  0.60608  0.33733
```

Demonstrate your implementation on a randomly generated matrix with two rows. Plot the original and selected column vectors as point on a plane. Distinguish not selected and selected data with different colors or marks.