Statistical inference for OU process dX (t) = -mu X (t) dt + dB (t)

```
<< Statistics`DescriptiveStatistics`; << Statistics`ContinuousDistributions`;
```

The transition density function for the OU process pOU[mu, x, y, t] = P {X (t) ∈ dy | X (0) = x} dy can be found at various places in the literature :

```
Clear[pOU]; pOU[mu_, x_, y_, t_] := Exp[-(y - x * Exp[-mu * t]) ^ 2 / (2 *
      (1 - Exp[-2 * mu * t]) / (2 * mu))] / (Sqrt[2 * Pi] * Sqrt[1 - Exp[-2 * mu * t]] / Sqrt[2 * mu]
```

Check that pOU[mu, x, y, t] satisfies the Backward Equation (6.27) D_t p - (D_x) ^ 2 p / 2 + mu * x * D_x p = 0 :

```
Simplify[
 D[pOU[mu, x, y, t], t] - D[D[pOU[mu, x, y, t], x], x] / 2 + mu * x * D[pOU[mu, x, y, t], x]]
```

```
0
```

The stationary distribution of the OU process is given by (6.65) :

```
Clear[PDFStat]; PDFStat[mu_, x_] := PDF[NormalDistribution[0, 1 / Sqrt[2 * mu]], x];
```

Check that PDFStat[mu, x] satisfies the Forward Equation in Example 12 (D_x) ^ 2 p / 2 + D_x (mu * x * p) = 0 (Note : There is a misprint with a - sign instead of a + in Example 12) :

```
Simplify[D[D[PDFStat[mu, x], x], x] / 2 + D[(mu * x * PDFStat[mu, x]), x]]
```

```
0
```

Check that PDFStat[mu, x] satisfies the defining Equation (6.62) for a stationary density

$\int$ **pOU[mu, x, y, t] PDFStat[mu, x] dx = PDFStat[mu, y] :**

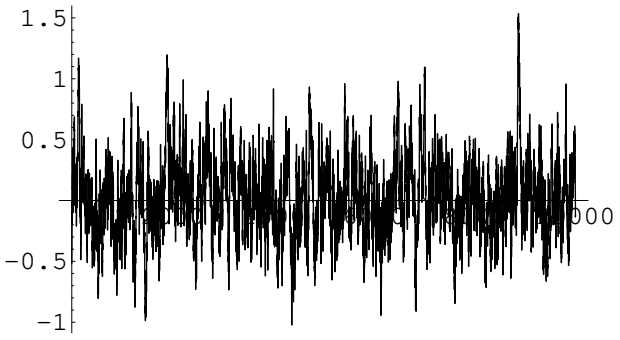
```
Simplify[Integrate[pOU[mu, x, y, t] * PDFStat[mu, x], {x, -Infinity, Infinity},
   Assumptions → mu > 0 && t > 0 && y ∈ Reals] - PDFStat[mu, y]]
```

```
0
```

Simulation of an OU process is straightforward by the Euler method :

```
mu0 = 5; StepsPerTimeUnit = 100; T = 100;
OUData = {Random[NormalDistribution[0, 1 / Sqrt[2 * mu0]]]};
For[i = 2, i ≤ StepsPerTimeUnit * T, i++,
 AppendTo[OUData, OUData[[i - 1]] - mu0 * OUData[[i - 1]] / StepsPerTimeUnit +
   Random[NormalDistribution[0, 1 / Sqrt[StepsPerTimeUnit]]]]];
```

```
ListPlot[OUData, PlotJoined → True];
```




```
Method 1 : Estimation of  mu  by Likelihood Ratio according to Example 10.6
```

```
mu1 =
 -Sum[OUData[[i - 1]] * (OUData[[i]] - OUData[[i - 1]]), {i, 2, StepsPerTimeUnit * T}] /
  Sum[OUData[[i]] ^ 2 / StepsPerTimeUnit, {i, 1, StepsPerTimeUnit * T}]
```

```
5.00198
```

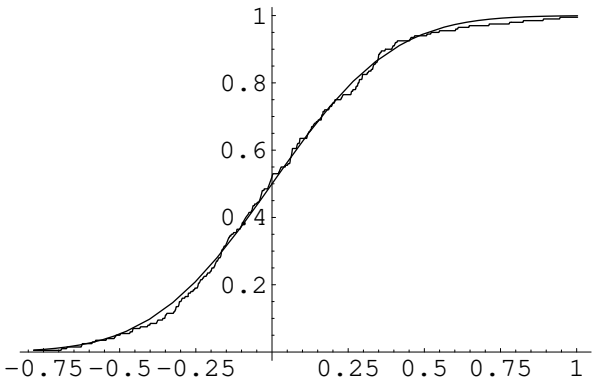
```
Method 2 : Estimation of  mu  by minimizing Kolmogorov -
  Smirnov Distance between empirical CDF and stationary CDF
```

```
Clear[FEmp]; FEmp[data_, x_] :=
 {lt = Length[data], Sum[If[data[[i]] ≤ x, 1, 0], {i, 1, lt}] / lt}[[2]];
Clear[CDFStat]; CDFStat[mu_, x_] := CDF[NormalDistribution[0, 1 / Sqrt[2 * mu]], x];

SampleDistance = 50; NData = StepsPerTimeUnit * T / SampleDistance;
NewOUData = Table[OUData[[i * SampleDistance]], {i, 1, NData}];
SortNewOUData = Sort[NewOUData]; mu2 =
 mu /. NMinimize[Max[Table[Max[{Abs[CDFStat[mu, SortNewOUData[[i]]] - (i - 1) / NData],
        Abs[CDFStat[mu, SortNewOUData[[i]]] - i / NData]}], {i, 1, NData}]], {mu}][[2]]
```

```
5.19097
```

```
Show[{Plot[FEmp[NewOUData, x], {x, SortNewOUData[[1]], SortNewOUData[[NData]]}],
   Plot[CDFStat[mu2, x], {x, SortNewOUData[[1]], SortNewOUData[[NData]]}]}];
```




```
Method 3 : Estimation of  mu  by Maximum Likelihood method based on  pOU[mu, x, y, t]
```

```
mu3 =
 mu /. NMaximize[{Sum[Log[pOU[mu, NewOUData[[i - 1]], NewOUData[[i]], T / NData]], {i,
       2, NData}], mu > 0}, {mu}][[2]]
```

```
4.91408
```