# Linear Model Fit

## Simple simulation to check properties of the linear model

### Preparations for the matrix method

In[55]:=
```
A = Table[{1, 0.025 + i * 0.05}, {i, 0, 19}]
```

Out[55]=
```
{{1, 0.025}, {1, 0.075}, {1, 0.125}, {1, 0.175}, {1, 0.225},
 {1, 0.275}, {1, 0.325}, {1, 0.375}, {1, 0.425}, {1, 0.475},
 {1, 0.525}, {1, 0.575}, {1, 0.625}, {1, 0.675}, {1, 0.725},
 {1, 0.775}, {1, 0.825}, {1, 0.875}, {1, 0.925}, {1, 0.975}}
```

In[56]:=
```
xvec = #2 & @@@ A
```

Out[56]=
```
{0.025, 0.075, 0.125, 0.175, 0.225, 0.275, 0.325, 0.375, 0.425, 0.475,
 0.525, 0.575, 0.625, 0.675, 0.725, 0.775, 0.825, 0.875, 0.925, 0.975}
```

In[57]:=
```
σ = 0.2 * (1.0 - 2.0 # + 2.0 * #^2) & /@ xvec
```

Out[57]=
```
{0.19025, 0.17225, 0.15625, 0.14225, 0.13025, 0.12025,
 0.11225, 0.10625, 0.10225, 0.10025, 0.10025, 0.10225, 0.10625,
 0.11225, 0.12025, 0.13025, 0.14225, 0.15625, 0.17225, 0.19025}
```

### Create covariance matrix (the data) and weight matrix

In[58]:=
```
V = DiagonalMatrix[σ^2];
W = Inverse[V];
```

### Calculate the covariance matrix of the parameters and their uncertainties

Note: The calculation only needs the errors in the data and the x values, not the actual data

In[60]:=
```
Va = Inverse[Transpose[A].W.A]
√Diagonal[Va]
TableForm[{%}, TableHeadings → {None, {"σ0", "σ1"}}]
```

Out[60]=
```
{{0.00428159, -0.00700774}, {-0.00700774, 0.0140155}}
```

Out[61]=
```
{0.0654339, 0.118387}
```

Out[62]//TableForm=

| $\sigma_0$ | $\sigma_1$ |
|---|---|
| 0.0654339 | 0.118387 |

# Simulation of the data and determination of the fit parameters and the $\chi^2$

$y_i = a_0 + a_1 x_i + \epsilon_i$

$a_0 = 2$

$a_1 = -3$

$\epsilon_i$ will be taken from a normal distribution $\mathcal{N}(0, \sigma_i)$

In[63]:=
```
yvec = Transpose[
  {RandomReal[NormalDistribution[2 - 3 #1, #2]] & @@@ Transpose[{xvec, σ}]}]
Flatten[{a = Va.Transpose[A].W.yvec,
  Transpose[yvec].W.yvec - Transpose[a].Transpose[A].W.yvec}]
```

Out[63]=
```
{{1.66979}, {1.64086}, {1.84468}, {1.22649}, {1.43502},
 {1.22932}, {0.880414}, {0.934108}, {0.786346}, {0.577267},
 {0.447829}, {0.257084}, {0.111838}, {0.0409262}, {-0.134252},
 {-0.0179125}, {-0.485101}, {-0.939034}, {-0.798955}, {-0.728479}}
```

Out[64]=
```
{1.95522, -2.8917, 20.9787}
```

The whole thing is done 100,000 times

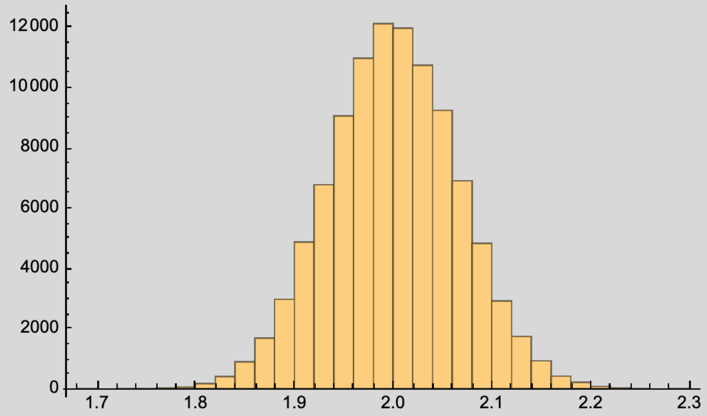
In[65]:=
```
n = 100 000;
data = Table[yvec = Transpose[
     {RandomReal[NormalDistribution[2 - 3 #1, #2]] & @@@ Transpose[{xvec, σ}]}];
  Flatten[{a = Va.Transpose[A].W.yvec,
    Transpose[yvec].W.yvec - Transpose[a].Transpose[A].W.yvec}], {n}];
```

## Split the fiter results into separate tables

In[66]:=
```
data0 = #1 & @@@ data;
data1 = #2 & @@@ data;
data2 = #3 & @@@ data;
```

## Parameter $a_0$ (y - intercept)

In[69]:=
```
Histogram[data0]
Mean[data0]
StandardDeviation[data0]
```

Out[69]=


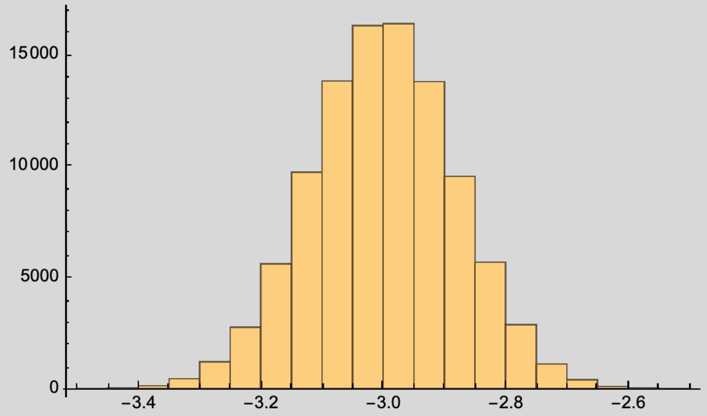

Out[70]= 2.00026

Out[71]= 0.0655114

Good agreement with the value determined above: $\sigma_0 = 0.0654339$

## Parameter $a_1$ (slope)

In[72]:=
```
Histogram[data1]
Mean[data1]
StandardDeviation[data1]
```

Out[72]=


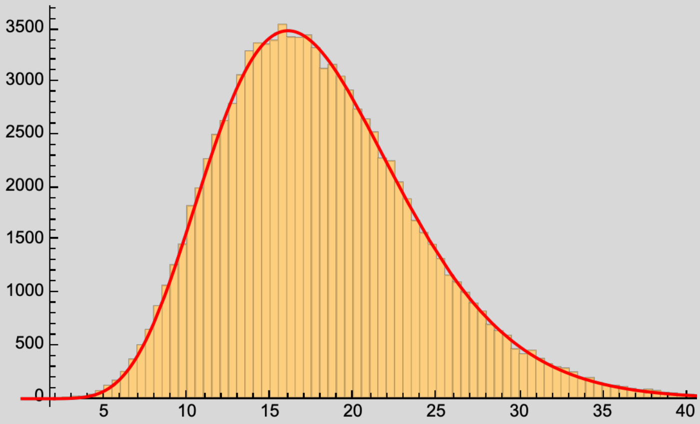

Out[73]= -3.00033

Out[74]= 0.118395

Good agreement with the value determined above: $\sigma_1 = 0.118387$

## SumOfSquares

In[75]:=
```
Histogram[data2, {0.5}, Epilog → First@
   Plot[0.5 n PDF[ChiSquareDistribution[18], x], {x, 0, 60}, PlotStyle → Red]]
Mean[data2]
Variance[data2]
```

Out[75]=



Out[76]= 18.0089

Out[77]= 36.0476

Good agreement with a $\chi^2$ distribution with 18 d.o.f

## Median

In[78]:=
$$\text{median} = 18\left(1 - \frac{2}{9 \times 18}\right)^3 \text{ // N}$$

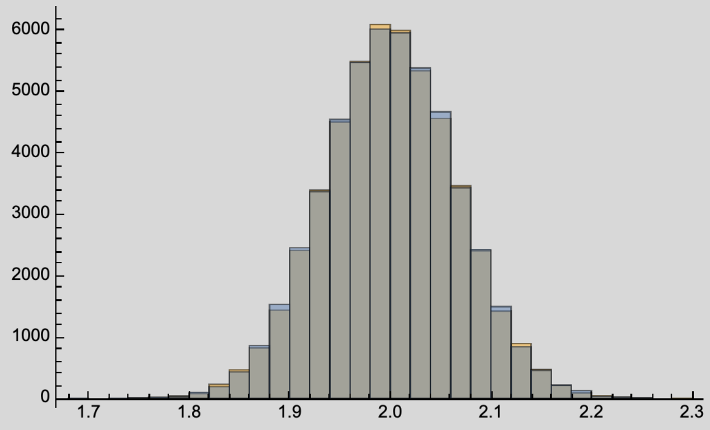
Out[78]= 17.3415

## Split into 2 lists : small and large $\chi^2$

In[79]:=
```
dataL = Select[data, #[[3]] < median &];
dataH = Select[data, #[[3]] > median &];
```

In[81]:=
```
Histogram[{dataL[[All, 1]], dataH[[All, 1]]}]
```

Out[81]=




Note: The $\chi^2$ does not say anything about how close the parameters are to the “true” value.

## Covariance ellipse

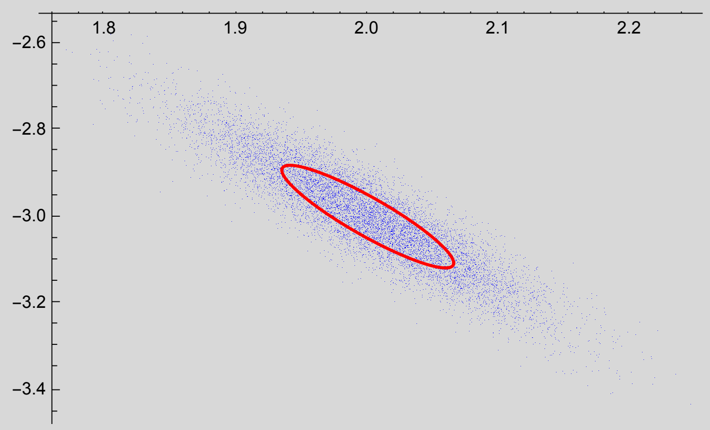
In[82]:=
```
{u, s, v} = SingularValueDecomposition[Va]
ellipse = u.(Sqrt[Diagonal[s]] {Cos[t], Sin[t]})
```

Out[82]=
```
{{{-0.463448, 0.886124}, {0.886124, 0.463448}},
 {{0.0176806, 0.}, {0., 0.000616501}},
 {{-0.463448, 0.886124}, {0.886124, 0.463448}}}
```

Out[83]=
```
{-0.0616239 Cos[t] + 0.022002 Sin[t], 0.117826 Cos[t] + 0.0115071 Sin[t]}
```

In[84]:=
```
Show[ListPlot[data[[1 ;; 10000, 1 ;; 2]], PlotStyle → {Blue, PointSize[0.002]}],
 ParametricPlot[ellipse + {2, -3}, {t, 0, 2 π}, PlotStyle → Red]]
```

Out[84]=



## Data points within the ellipse

In[85]:=
```
dataE = Select[data, Total[
      ((1 / Sqrt[Diagonal[s]]) (Transpose[u].(#[[1 ;; 2]] - {2, -3}))) ^2] < 1 &];
Dimensions[dataE]
CDF[ChiSquareDistribution[2], 1] // N
```

Out[86]=
```
{39 346, 3}
```

Out[87]=
```
0.393469
```

It is expected that 39.3% of the points are within the ellipse

In[88]:=
```
Show[ListPlot[data[[1 ;; 10 000, 1 ;; 2]], PlotStyle → {Blue, PointSize[0.002]}],
 ListPlot[dataE[[1 ;; 3935, 1 ;; 2]], PlotStyle → {Yellow, PointSize[0.002]}],
 ParametricPlot[ellipse + {2, -3}, {t, 0, 2 π}, PlotStyle → Red]]
```

Out[88]=