Note: This is going to be more of a “follow along” sort of class. So the notebook will be more sparse than usual, and mostly serves as an outline of the lecture.

Visualizing Data

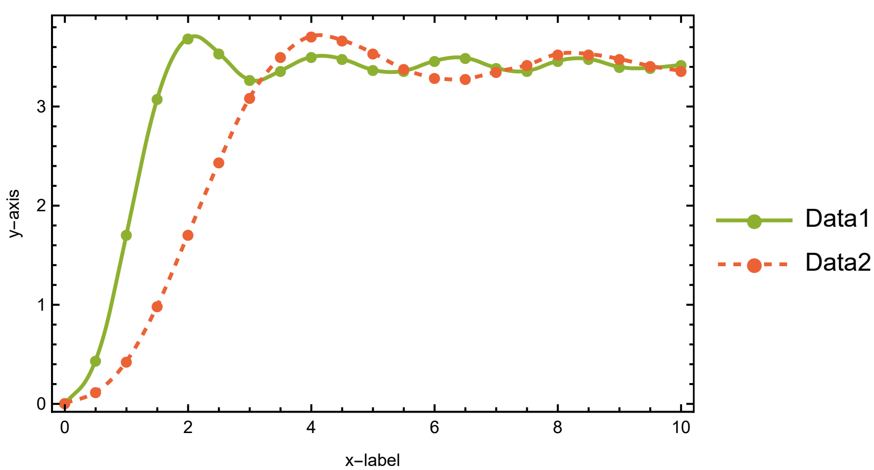
ListPlot and ListLinePlot

In[∘]:=
```
data1 = {{0., 0.01`}, {0.5, 0.44}, {1., 1.71}, {1.5, 3.08},
    {2., 3.69}, {2.5, 3.54}, {3., 3.27}, {3.5, 3.36}, {4., 3.5}, {4.5, 3.48},
    {5., 3.37}, {5.5, 3.36}, {6., 3.46}, {6.5, 3.49}, {7., 3.39}, {7.5, 3.36},
    {8., 3.46}, {8.5, 3.48}, {9., 3.4}, {9.5, 3.39}, {10., 3.42}};

data2 = {{0., 0.01`}, {0.5, 0.12}, {1., 0.43}, {1.5, 0.99},
    {2., 1.71}, {2.5, 2.44}, {3., 3.09}, {3.5, 3.5}, {4., 3.71}, {4.5, 3.67},
    {5., 3.54}, {5.5, 3.38}, {6., 3.29}, {6.5, 3.28}, {7., 3.35}, {7.5, 3.42},
    {8., 3.53}, {8.5, 3.53}, {9., 3.48}, {9.5, 3.41}, {10., 3.36}};
```

In[∘]:=
```
ListPlot[{data1, data2}, PlotRange → All, Frame → True,
 FrameLabel → {"x-label", "y-axis"}, Joined → True, InterpolationOrder → Automatic,
 PlotMarkers → "●", PlotLegends → {"Data1", "Data2"},
 PlotStyle → {ColorData[97, 3], Directive[{Dashed, ColorData[97, 4]}]}]
```

Out[∘]=



In[∘]:=
```
ColorData[97]
```

Out[∘]=

ColorDataFunction[Index: 97 Colors: ∞ Palette:]

```
\[ FilledCircle ] → "●"
```

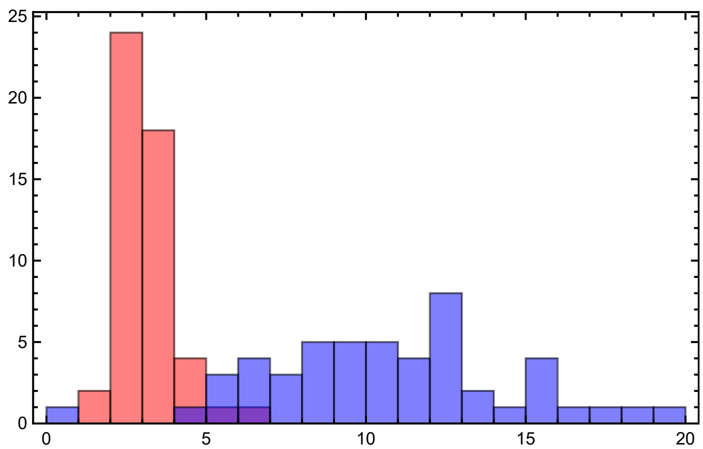
●

Histogram and SmoothHistogram

In[ ]:=
```
data3 = {2.89175, 2.61613, 2.95601, 2.71121, 3.61643, 2.61884, 2.84275, 2.98486,
   2.87965, 4.6046, 3.00923, 5.42387, 2.09306, 3.00936, 3.79174, 2.95188, 1.64658,
   3.44998, 2.47346, 2.50094, 1.58891, 6.38088, 3.91435, 2.58256, 3.73725,
   2.53284, 3.15859, 3.19481, 2.16886, 2.57557, 3.55481, 4.67243, 2.88272,
   3.12072, 2.95147, 2.17699, 3.62933, 2.94488, 3.01143, 4.10778, 2.54096,
   2.49918, 3.93841, 3.085, 3.71711, 2.46394, 2.64561, 3.52655, 4.33454, 3.37427};
data4 = {0.711584, 9.20076, 10.6202, 12.5684, 15.208, 12.8391, 9.64308,
   9.7832, 17.2989, 5.22502, 19.1653, 8.93812, 13.1611, 4.17281, 10.956, 12.975,
   11.1901, 12.343, 15.0823, 10.1314, 12.0914, 6.17371, 12.1185, 6.88504, 8.54537,
   7.8976, 8.48243, 16.0958, 15.7727, 14.316, 6.26089, 11.1846, 10.2433, 12.5578,
   8.45365, 7.89192, 6.51205, 9.21282, 5.19501, 8.93034, 9.62611, 18.1242,
   15.1844, 12.1187, 11.7128, 13.975, 10.3568, 11.024, 7.35753, 5.94573};
```

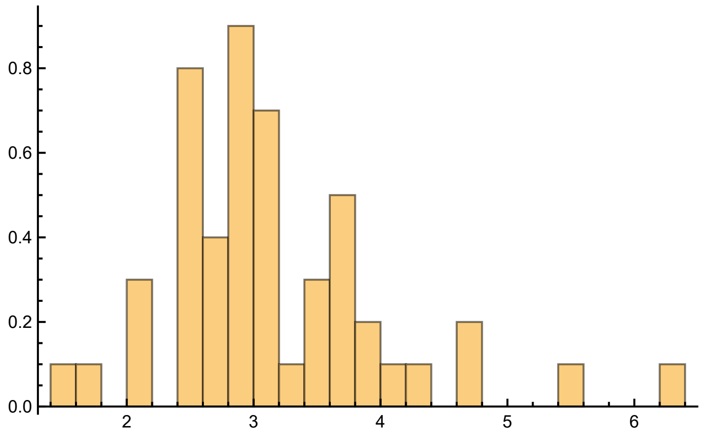
In[ ]:=
```
Histogram[{data3, data4}, 15, Frame → True, ChartStyle → {Red, Blue}]
```

Out[ ]=

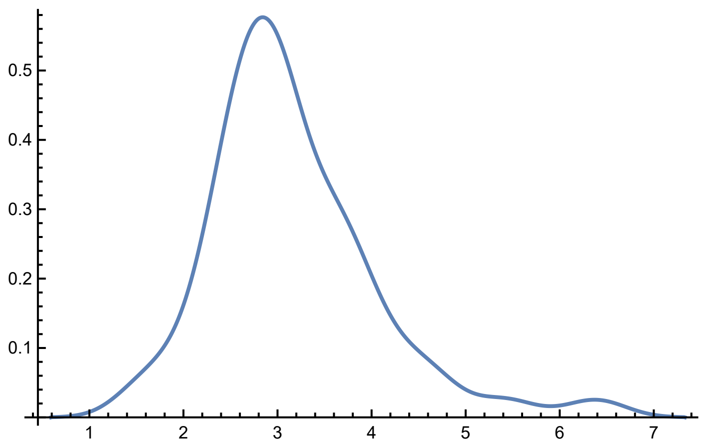


In[ ]:=
```
Histogram[data3, 15, "PDF"]
SmoothHistogram[data3]
```

Out[ ]=



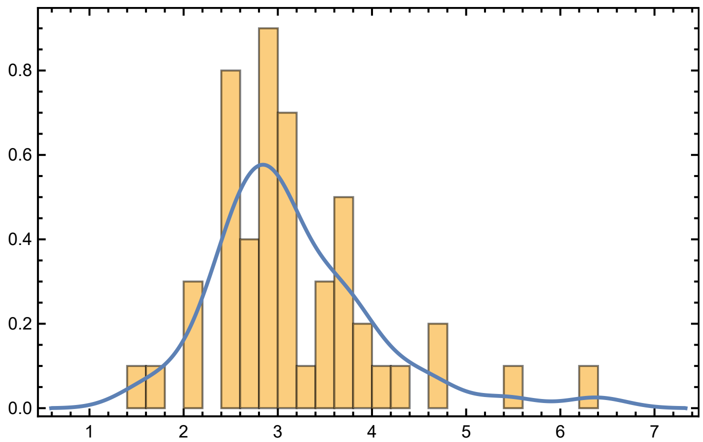
Out[ ]=



Overlaying Plots with Show

In[ ]:=
```
Show[{
  Histogram[data3, 15, "PDF"],
  SmoothHistogram[data3]
 }, Frame → True, PlotRange → All, Axes → False]
```

Out[ ]=



Many More! https://reference.wolfram.com/language/guide/DataVisualization.html

Raw visualization with Grid

In[ ]:=
```
data1 // Grid
```

Out[ ]=

0.	0.01
0.5	0.44
1.	1.71
1.5	3.08
2.	3.69
2.5	3.54
3.	3.27
3.5	3.36
4.	3.5
4.5	3.48
5.	3.37
5.5	3.36
6.	3.46
6.5	3.49
7.	3.39
7.5	3.36
8.	3.46
8.5	3.48
9.	3.4
9.5	3.39
10.	3.42

Importing Data

The Import Function Import[]

Download example dataset here: https://drive.google.com/file/d/1bp-HtrjpPpEdmM43eQPla3DAXUa-jr4ku/view?usp=drive_link

In[ ]:=
```
Import["C:\\Users\\Collin\\Downloads\\Week5Dataset (2).csv"]
```

Out[ ]=
```
{{Time (s), Channel 1 (V), Channel 2 (V)}, {-0.00511926, -1.01111, -2.57694}, {-0.00511801, -1.01478, -2.57325},
 {-0.00511676, -1.01844, -2.57325}, {-0.00511551, -1.01478, -2.57694}, {-0.00511426, -1.01111, -2.57325},
 {-0.00511301, -1.01478, -2.57325}, {-0.00511176, -1.01111, -2.57694}, {-0.00511051, -1.01111, -2.57325},
 … 8175 …, {0.00510949, 1.00128, 2.54519}, {0.00511074, 0.997619, 2.54888}, {0.00511199, 0.993953, 2.5415},
 {0.00511324, 0.997619, 2.55257}, {0.00511449, 0.993953, 2.54888}, {0.00511574, 0.997619, 2.54519},
 {0.00511699, 0.997619, 2.54519}, {0.00511824, 0.997619, 2.55257}, {0.00511949, 0.997619, 2.54888}}
```

Full expression not available (original memory size: 1 MB)

```
In[ ]:= data = {⋯} ;
```

How to import table data from a pdf?

View Example PDF here: https://drive.google.com/file/d/1OUziyDsIM4d1Sq4m_z_3MoQvxEgdEj-JA/view?usp=drive_link

```
In[ ]:= data2 = Partition[ToExpression /@ StringSplit["0 0 120
0.1 0.739 415
0.2 0.849 649
0.3 0.892 828
0.4 0.915 959
0.5 0.929 1050
0.6 0.938 1109
0.7 0.944 1150
0.8 0.95 1176
0.9 0.962 1210
1 1 1250"], 3(*Number of rows in your table*)]

Out[ ]=
{{0, 0, 120}, {0.1, 0.739, 415}, {0.2, 0.849, 649},
 {0.3, 0.892, 828}, {0.4, 0.915, 959}, {0.5, 0.929, 1050}, {0.6, 0.938, 1109},
 {0.7, 0.944, 1150}, {0.8, 0.95, 1176}, {0.9, 0.962, 1210}, {1, 1, 1250}}
```

```
In[ ]:= data2 // Grid

Out[ ]=
 0    0    120
0.1 0.739  415
0.2 0.849  649
0.3 0.892  828
0.4 0.915  959
0.5 0.929 1050
0.6 0.938 1109
0.7 0.944 1150
0.8  0.95 1176
0.9 0.962 1210
 1    1   1250
```

How can I import data from a scanned pdf or an image?

x_1	x_2	P [mmHg]
0	0	120
0.1	0.739	415
0.2	0.849	649
0.3	0.892	828
0.4	0.915	959
0.5	0.929	1050
0.6	0.938	1109
0.7	0.944	1150
0.8	0.95	1176
0.9	0.962	1210
1	1	1250

Example Image:

x_1	x_2	P [mmHg]
0	0	120
0.1	0.739	415
0.2	0.849	649
0.3	0.892	828
0.4	0.915	959
0.5	0.929	1050
0.6	0.938	1109
0.7	0.944	1150
0.8	0.95	1176
0.9	0.962	1210
1	1	1250

In[∘]:=

```
Partition[ToExpression /@ StringSplit[TextRecognize[
```

0	0	120
0.1	0.739	415
0.2	0.849	649
0.3	0.892	828
0.4	0.915	959
0.5	0.929	1050
0.6	0.938	1109
0.7	0.944	1150
0.8	0.95	1176
0.9	0.962	1210
1	1	1250

```
], {" ", "\n"}], 3]
```

Out[∘]=

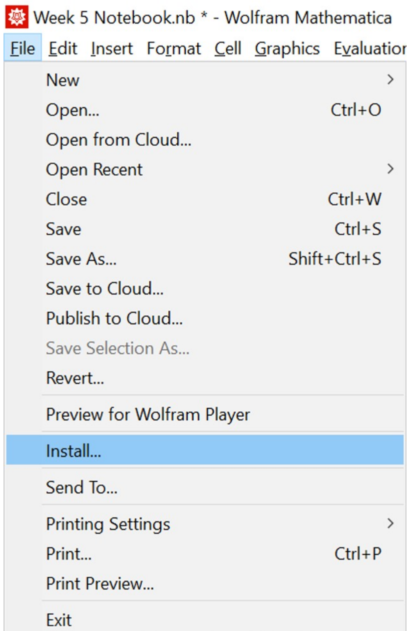
```
{{0, 0, 120}, {0.1, 0.739, 415}, {0.2, 0.849, 649},
 {0.3, 0.892, 828}, {0.4, 0.915, 959}, {0, 0.929, 1050}, {0.6, 0.938, 1109},
 {0.7, 0.944, 1150}, {0.8, 0.95, 1176}, {0.9, 0.962, 1210}, {Null, 1, 1}}
```

My preferred way of importing data

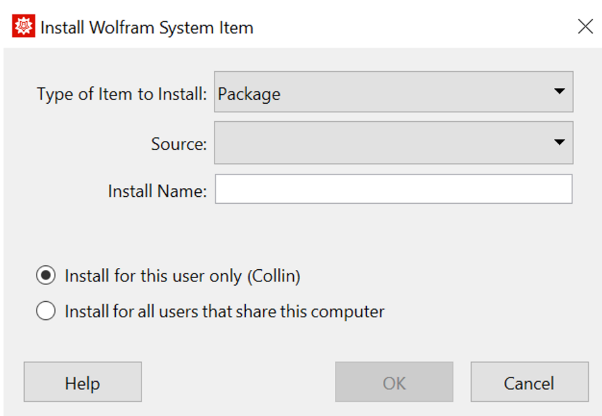
This works using a custom function I wrote which can be installed as follows

Installing my SpreadsheetEdit Package

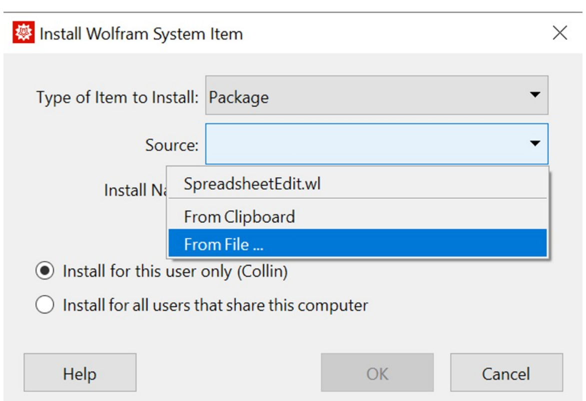
- Download the Mathematica Package file here: https://drive.google.com/file/d/1jv-mnb2ryPVfUQP2Ui6Me50_b224hll5/view?usp=drive_link
- Inside of Mathematica click on File > Install

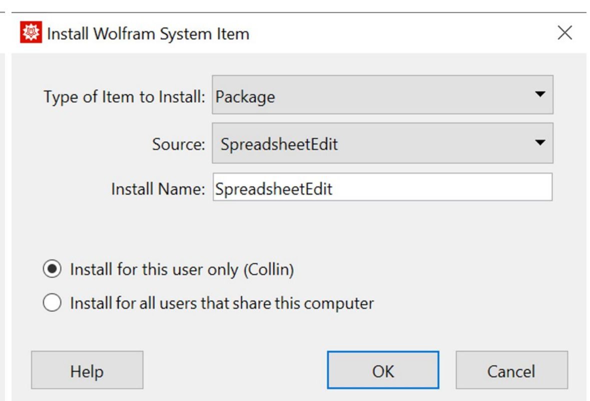


- Set the type of install to “Package”



- The set the source to file and select the SpreadsheetEdit.wl file that you downloaded





- Leave the install name as “SpreadsheetEdit” and click “OK”

Using SpreadsheetEdit

Now that SpreadsheetEdit has been installed you can load the package in any notebook on your computer using the following

```
In[]:= Needs["SpreadsheetEdit`"]
```

Note that if you ever plan on sharing a notebook with someone else, they won't be able to use this package unless they also install it themselves.

```
In[]:= SpreadsheetEdit[{…}] // Iconize
Out[]=
{…}
```

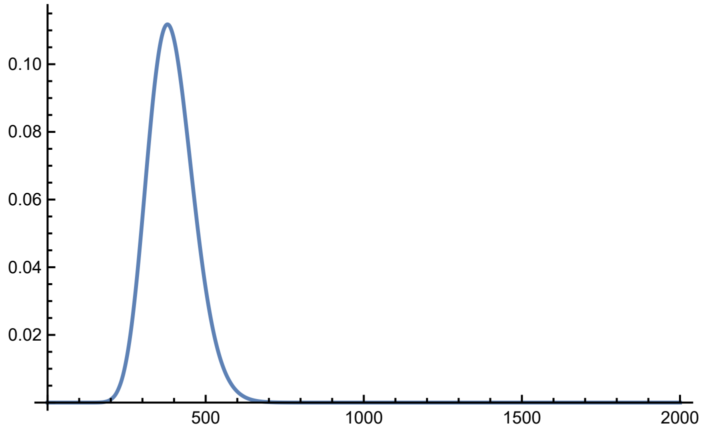
Analyzing Data

Yet another example data set: https://docs.google.com/spreadsheets/d/152ZTudb8YgkI00XGE7vlAx7gG-bKBv_DN/edit?usp=drive_link&ouid=102495462541733164891&rtpof=true&sd=true

Interpolation

```
In[]:= data = {{0, 0}, {20, 0}, {40, 0}, {60, 0}, {80, 0}, {100, 1.16029`*^-10}, {120, 4.36176`*^-8},
    {140, 1.76819`*^-6}, {160, 0.0000252811`}, {180, 0.000190681`}, {200, 0.000923138`},
    {220, 0.003212027`}, {240, 0.008632948`}, {260, 0.018828078`}, {280, 0.0345222`},
    {300, 0.054641447`}, {320, 0.076193932`}, {340, 0.095119986`}, {360, 0.107695436`},
    {380, 0.111764194`}, {400, 0.107255004`}, {420, 0.095888541`}, {440, 0.080370327`},
    {460, 0.063498734`}, {480, 0.047513989`}, {500, 0.033810651`}, {520, 0.022963329`},
    {540, 0.014933398`}, {560, 0.009325356`}, {580, 0.005606159`}, {600, 0.003252047`},
    {620, 0.001824069`}, {640, 0.000991143`}, {660, 0.000522619`}, {680, 0.000267836`},
    {700, 0.0001336`}, {720, 0.0000649492`}, {740, 0.0000308102`}, {760, 0.0000142777`},
    {780, 6.47011`*^-6}, {800, 2.86996`*^-6}, {820, 1.24721`*^-6}, {840, 5.31455`*^-7},
    {860, 2.22225`*^-7}, {880, 9.12508`*^-8}, {900, 3.68209`*^-8}, {920, 1.46099`*^-8},
    {940, 5.70364`*^-9}, {960, 2.1921`*^-9}, {980, 8.29855`*^-10}, {1000, 3.09598`*^-10},
    {1020, 1.13883`*^-10}, {1040, 0}, {1060, 0}, {1080, 0}, {1100, 0}, {1120, 0},
    {1140, 0}, {1160, 0}, {1180, 0}, {1200, 0}, {1220, 0}, {1240, 0}, {1260, 0}, {1280, 0},
    {1300, 0}, {1320, 0}, {1340, 0}, {1360, 0}, {1380, 0}, {1400, 0}, {1420, 0}, {1440, 0},
    {1460, 0}, {1480, 0}, {1500, 0}, {1520, 0}, {1540, 0}, {1560, 0}, {1580, 0},
    {1600, 0}, {1620, 0}, {1640, 0}, {1660, 0}, {1680, 0}, {1700, 0}, {1720, 0},
    {1740, 0}, {1760, 0}, {1780, 0}, {1800, 0}, {1820, 0}, {1840, 0}, {1860, 0},
    {1880, 0}, {1900, 0}, {1920, 0}, {1940, 0}, {1960, 0}, {1980, 0}, {2000, 0}};
```

In[ ]:=
```
ListPlot[data, PlotRange → All, Joined → True, InterpolationOrder → Automatic]
```
Out[ ]=



In[ ]:=
```
ifun = Interpolation[data]
```
Out[ ]=

InterpolatingFunction[Domain: {{0., 2.00×10^3}} Output: scalar]

In[ ]:=
```
ifun[310]
```
Out[ ]=

0.0654923

In[ ]:=
∂_x ifun[x] /. x → 310

Out[ ]=

0.00108608

In[ ]:=
```
A = Integrate[ifun[x], {x, 0, 2000}]
```

In[ ]:=
μ = Integrate$\left[\frac{\text{ifun[x]}}{\text{A}}\text{ x, \{x, 0, 2000\}}\right]$ // N

$\mu 2$ = Integrate$\left[\frac{\text{ifun[x]}}{\text{A}}\text{ x}^2\text{, \{x, 0, 2000\}}\right]$ // N;

$\sigma = \sqrt{\mu 2 - \mu^2}$

Out[ ]=

392.273

Out[ ]=

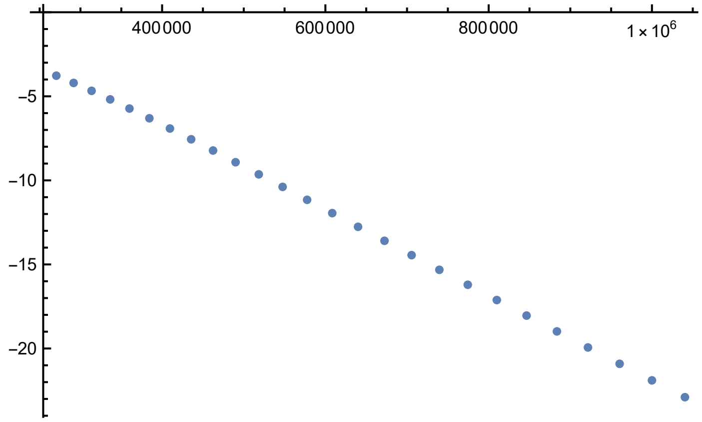
72.1924

Linear Regression

In[ ]:=
linData = Cases[data, {x_ /; 270 400 ≤ x^2 ≤ 1 040 400, y_} :→ {x^2, Log[y]}];

In[∘]:=
```
ListPlot[linData]
```

Out[∘]=



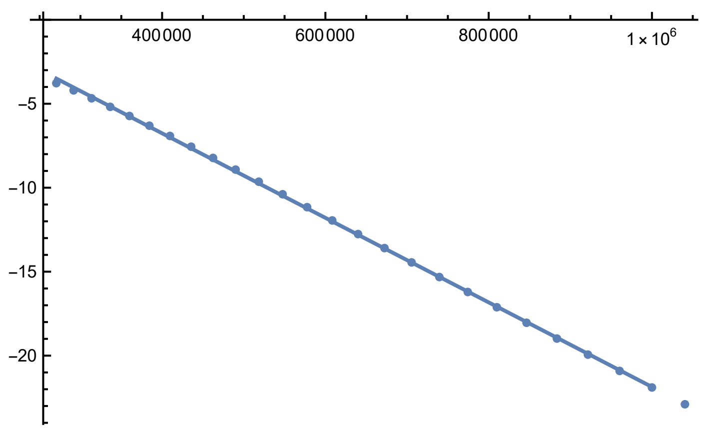
In[∘]:=
```
model1 = LinearModelFit[linData, x, x]
```

Out[∘]=

```
FittedModel[ 3.30891 - 0.0000251726 x ]
```

In[∘]:=
```
Show[{
  ListPlot[linData],
  Plot[model1[x], {x, 270400, 10^6}]
 }]
```

Out[∘]=



In[ ]:=
```
model1["Properties"]
```
Out[ ]=
```
{AdjustedRSquared, AIC, AICc, ANOVATable, ANOVATableDegreesOfFreedom, ANOVATableEntries,
 ANOVATableFStatistics, ANOVATableMeanSquares, ANOVATablePValues, ANOVATableSumsOfSquares,
 BasisFunctions, BetaDifferences, BestFit, BestFitParameters, BIC, CatcherMatrix,
 CoefficientOfVariation, CookDistances, CorrelationMatrix, CovarianceMatrix,
 CovarianceRatios, Data, DesignMatrix, DurbinWatsonD, EigenstructureTable,
 EigenstructureTableEigenvalues, EigenstructureTableEntries, EigenstructureTableIndexes,
 EigenstructureTablePartitions, EstimatedVariance, FitDifferences, FitResiduals, Function,
 FVarianceRatios, HatDiagonal, MeanPredictionBands, MeanPredictionConfidenceIntervals,
 MeanPredictionConfidenceIntervalTable, MeanPredictionConfidenceIntervalTableEntries,
 MeanPredictionErrors, ParameterConfidenceIntervals, ParameterConfidenceIntervalTable,
 ParameterConfidenceIntervalTableEntries, ParameterConfidenceRegion, ParameterErrors,
 ParameterPValues, ParameterTable, ParameterTableEntries, ParameterTStatistics,
 PartialSumOfSquares, PredictedResponse, Properties, Response, RSquared,
 SequentialSumOfSquares, SingleDeletionVariances, SinglePredictionBands,
 SinglePredictionConfidenceIntervals, SinglePredictionConfidenceIntervalTable,
 SinglePredictionConfidenceIntervalTableEntries, SinglePredictionErrors,
 StandardizedResiduals, StudentizedResiduals, VarianceInflationFactors}
```

In[ ]:=
```
model1["RSquared"]
```
Out[ ]=
```
0.999776
```

In[ ]:=
```
model1["ParameterConfidenceIntervalTable"]
```
Out[ ]=

	Estimate	Standard Error	Confidence Interval
1	3.30891	0.0506086	{3.20446, 3.41336}
x	−0.0000251726	7.69557×10^{-8}	{−0.0000253314, −0.0000250137}

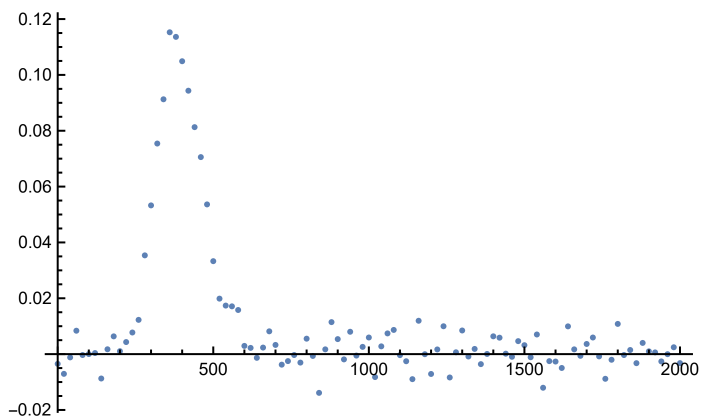
In[ ]:=
```
model1["ParameterTable"]
```
Out[ ]=

	Estimate	Standard Error	t-Statistic	P-Value
1	3.30891	0.0506086	65.3824	1.48159×10^{-28}
x	−0.0000251726	7.69557×10^{-8}	−327.105	2.60761×10^{-45}

Non-Linear Regression

In[∘]:=
```
(*Add artificial noise to the data*)
SeedRandom[3875];
NoisyData = data /. {x_, y_} ↦ {x, y + RandomVariate[NormalDistribution[0, 0.005]]};
ListPlot[NoisyData, PlotRange → All]
```

Out[∘]=



In[∘]:=
```
NormalDistribution[μ0, σ0] // PDF
```

Out[∘]=

$$\text{Function}\left[x, \frac{e^{-\frac{(x-\mu 0)^2}{2\,\sigma 0^2}}}{\sqrt{2\pi}\,\sigma 0}\right]$$

In[∘]:=

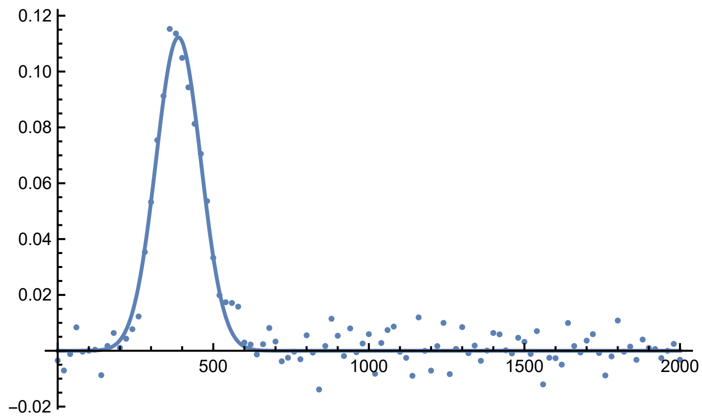
$$\text{model2} = \text{NonlinearModelFit}\left[\text{NoisyData}, \frac{A0\, e^{-\frac{(x-\mu 0)^2}{2\,\sigma 0^2}}}{\sqrt{2\pi}\,\sigma 0}, \{\{A0, 1\}, \{\mu 0, 360\}, \{\sigma 0, 100\}\}, x\right]$$

Out[∘]=

$$\text{FittedModel}\left[\boxed{0.112229\, e^{-0.0000960871\,(-387.924+x)^2}}\right]$$

In[ ]:=
```
Show[{
  ListPlot[NoisyData, PlotRange → All],
  Plot[model2[x], {x, 0, 2000}, PlotRange → All]
 }]
```

Out[ ]=



In[ ]:=
```
model2["Properties"]
```

Out[ ]=

{AdjustedRSquared, AIC, AICc, ANOVATable, ANOVATableDegreesOfFreedom, ANOVATableEntries, ANOVATableMeanSquares, ANOVATableSumsOfSquares, BestFit, BestFitParameters, BIC, CorrelationMatrix, CovarianceMatrix, CurvatureConfidenceRegion, Data, EstimatedVariance, FitCurvatureTable, FitCurvatureTableEntries, FitResiduals, Function, HatDiagonal, MaxIntrinsicCurvature, MaxParameterEffectsCurvature, MeanPredictionBands, MeanPredictionConfidenceIntervals, MeanPredictionConfidenceIntervalTable, MeanPredictionConfidenceIntervalTableEntries, MeanPredictionErrors, ParameterBias, ParameterConfidenceIntervals, ParameterConfidenceIntervalTable, ParameterConfidenceIntervalTableEntries, ParameterConfidenceRegion, ParameterErrors, ParameterPValues, ParameterTable, ParameterTableEntries, ParameterTStatistics, PredictedResponse, Properties, Response, RSquared, SingleDeletionVariances, SinglePredictionBands, SinglePredictionConfidenceIntervals, SinglePredictionConfidenceIntervalTable, SinglePredictionConfidenceIntervalTableEntries, SinglePredictionErrors, StandardizedResiduals, StudentizedResiduals}

In[ ]:=
```
model2["ParameterConfidenceIntervalTable"]
```

Out[ ]=

	Estimate	Standard Error	Confidence Interval
A0	20.2931	0.492436	{19.3159, 21.2703}
μ0	387.924	2.02127	{383.912, 391.935}
σ0	72.1361	2.02127	{68.1249, 76.1472}