# Title

## Table of contents

Step 1: Create the DataFrame in Python . . . . . 3

```
<IPython.core.display.HTML object>
```

```
//| echo: false
viewof aSlider = Inputs.range([-4, 0], { label: "a", step: 0.01, value: -1.7 });
viewof bSlider = Inputs.range([-2, 2], { label: "b", step: 0.01, value: 1.3 });
viewof cSlider = Inputs.range([-2, 2], { label: "c", step: 0.01, value: 1.0 });
```

```
//| echo: false
filtered = transpose(data);
// Create the plot

xValues = Array.from({ length: 100 }, (_, i) => i / 100);
parabolaData = xValues.map(x => ({ x, y: parabola(x, aSlider, bSlider, cSlider) }));


parabola = (x, a, b, c) => a * x**2 + b * x + c

calculateChiSquared = (data, a, b, c) => {
  let chisq = 0
  let x= data.map(d => d.x)
  let y= data.map(d => d.y)
  let err= data.map(d => d.error)
  for (let i = 0; i < x.length; i++) {
    let y_model = parabola(x[i], a, b, c)
    chisq += ((y[i] - y_model) / err[i])**2
  }
```

```
  return chisq
}

chisq = calculateChiSquared(filtered, aSlider, bSlider, cSlider)

Plot.plot({
  marks: [
    Plot.dot(filtered, { x: "x", y: "y" }),
    Plot.ruleY(filtered, { x: "x", y1: d => d.y - d.error, y2: d => d.y + d.error }),
    Plot.line(parabolaData, { x: "x", y: "y" }),
    Plot.text([{ x: 0.8, y: 1.5, label: `χ²: ${chisq.toFixed(2)}` }], {
          x: "x",
          y: "y",
          text: "label",
          dy: -10, // Adjust vertical position if needed
          fill: "black", // Set text color
          fontSize: 16
        }),
    Plot.frame()
  ],
  x: {
    label: "X Axis",
    labelAnchor: "center",
    labelOffset: 35,
    grid: true,
    tickFormat: ".2f", // Format ticks to 2 decimal places
    domain: [0, 1]
  },
  y: {
    label: "Y Axis",
    grid: true,
    tickFormat: ".2f", // Format ticks to 2 decimal places
    labelAnchor: "center",  // Center the label on its axis
    labelAngle: -90,
    labelOffset: 60,
    domain: [0, 2],
  },
  width: 400,
  height: 400,
  marginLeft: 100,
  marginBottom: 40,
  style: {
```

```
    fontSize: "14px",          // This sets the base font size
    "axis.label": {
      fontSize: "18px",        // This sets the font size for axis labels
      fontWeight: "bold"       // Optionally make it bold
    },
    "axis.tick": {
      fontSize: "14px"         // This sets the font size for tick labels
    }
  },
})
```

Hello Here's how you can do it:

**Step 1: Create the DataFrame in Python**

```
# Here is some data of the height measurements including untertainties
#x_data,y_data,err=np.loadtxt('data.txt',unpack=True)
x_data, y_data, err = np.loadtxt(io.StringIO(data_str), unpack=True)
```

```
import numpy as np
import io
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
plt.rcParams.update({'font.size': 18})
from ipywidgets import interact, interactive, fixed, interact_manual
import ipywidgets as widgets

# default values for plotting
plt.rcParams.update({'font.size': 12,
                     'lines.linewidth': 1,
                     'lines.markersize': 10,
                     'axes.labelsize': 11,
                     'xtick.labelsize' : 10,
                     'ytick.labelsize' : 10,
                     'xtick.top' : True,
                     'xtick.direction' : 'in',
                     'ytick.right' : True,
                     'ytick.direction' : 'in',})

data_str= """
```

```
0.00000000000000000e+00 9.91683920405706742 5.33281879919848197e-02
1.11111111111111049e-01 1.18366784044016171 5.33955964674283842e-02
2.22222222222222099e-01 1.31086214896105702 5.36678332638267225e-02
3.33333333333333148e-01 1.19317486726599964 5.43509749388307109e-02
4.44444444444444198e-01 1.26535413082458015 5.57699550148873235e-02
5.55555555555555802e-01 1.23427763410080615 5.83263657369805927e-02
6.66666666666666297e-01 1.06720479956899639 6.24270672925735376e-02
7.77777777777776791e-01 7.11370689472352047 6.84033407062092508e-02
8.88888888888888395e-01 5.11162542953954691 7.64587225708259766e-02
1.00000000000000000e+00 1.36083820999623839 4.66681116534375329e-01
"""
x_data, y_data, err = np.loadtxt(io.StringIO(data_str), unpack=True)

def parabola(x,a,b,c):
    return(a*x**2+b*x+c)

def plot(a,b,c):
    y=parabola(x,a,b,c)
    plt.figure(figsize=(8,6))
    chisq=(((y_data-parabola(x_data,a,b,c))/err)**2).sum()
    plt.plot(x,y,label=r'$\chi^2$={0:6.3f}'.format(chisq))
    plt.errorbar(x_data,y_data,yerr=err,marker='o',fmt="none",color='k')

    plt.scatter(x_data,y_data,marker='o',color='k')
    plt.legend()
    plt.xlabel('x- position')
    plt.ylabel('y- position')
    plt.show()

x=np.linspace(0,1,100)
interact(plot,a=-1.7,b=1.3,c=1.0);
```

```
interactive(children=(FloatSlider(value=-1.7, description='a', max=1.7, min=-5.1), FloatSlide
```