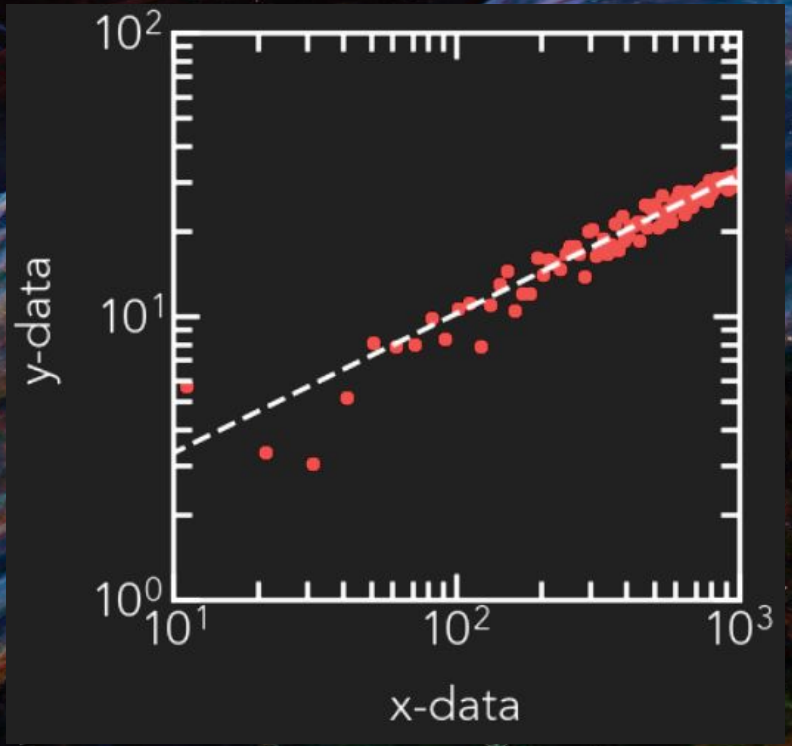

Lecture 11
Statistics and Plotting

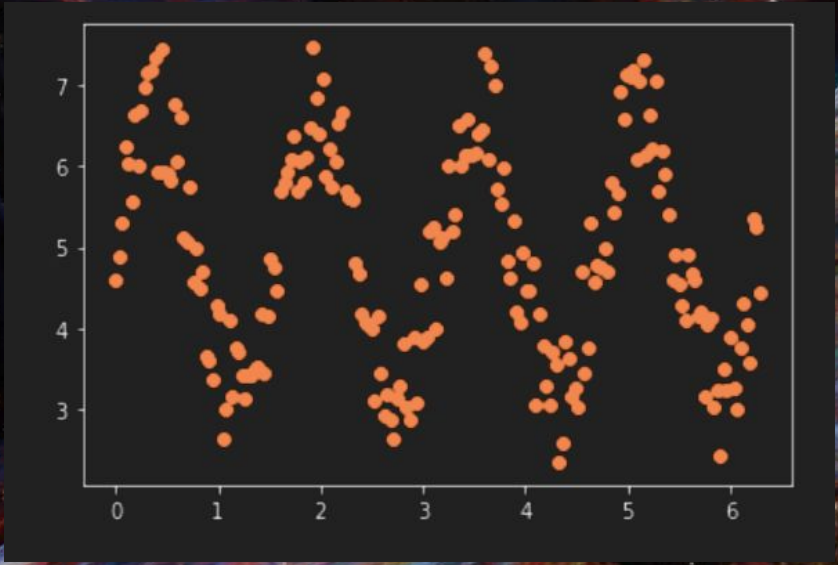
What's better than a plot?

- A plot with statistical significance
- We can do this by fitting our plots with a trend line.
- From this we can make predictions of the data with a known uncertainty



What do we need for fitting?

DATA

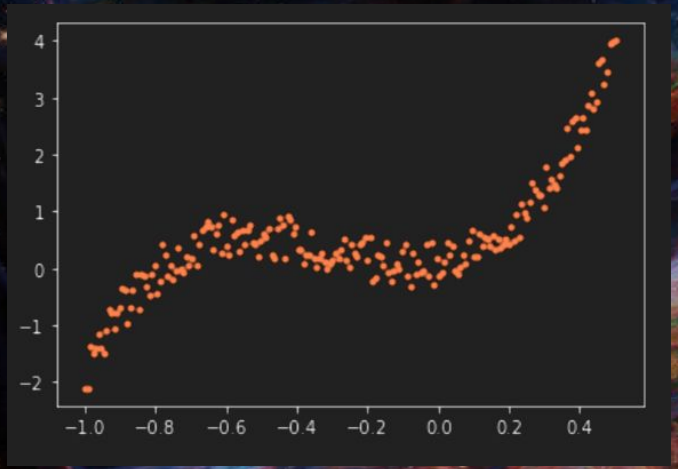


MODEL

$$y = f(x)$$

FIrst Step: Learning Polynomial Fits

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_P x^P = \sum_{i=0}^{P} a_i x^i$$

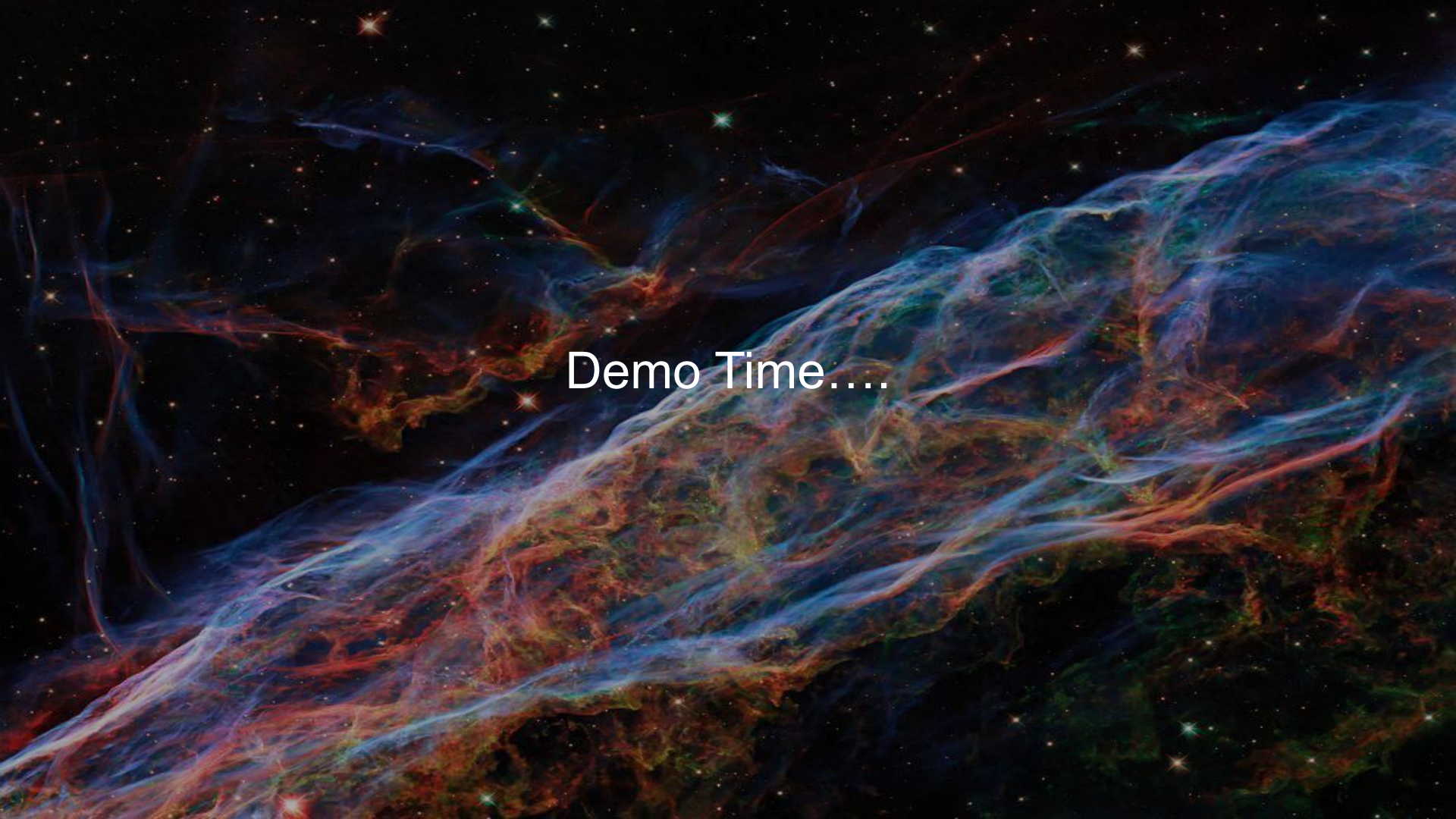


Degree of the polynomial
(the highest power P)

```
np.polyfit(x, y, deg)
```

It outputs an array of coefficients

```
[aP, … , a1, a0]
```


Demo Time….

Something slightly more useful…

```
from scipy.optimize import curve_fit
```

Importing the function this way will allow you to directly use the **curve_fit** function without typing everything else.

This command basically picks out the function from **scipy.optimize** package.

The `curve_fit` function

```
fit_params = curve_fit(model, xdata, ydata, \
                       p0=[...], \
                       sigma=[...], \
                       bounds=([...], [...]))
```

Initial guess, [...] is an array

Uncertainty on ydata, [...] is an array of length len(ydata)

Fixes range for your fitted parameter

A note on defining the model function:

```
def model(x, a1, a2, a3):
    return a1*np.sin(x)**a2 + a3
```

Fitted parameters can be obtained by calling `fit_params[0]`

What is a good fit?

Minimise chi-squared

Fitted model

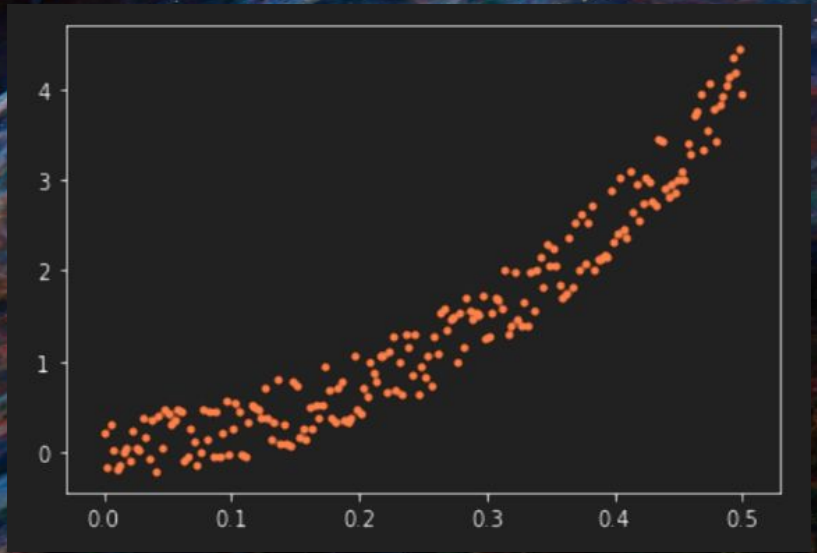
$$\chi^2 = \sum_i \left(\frac{y_i - f(x_i)}{\sigma_i} \right)^2$$

uncertainty

Be aware of overfitting

- Number of data points should be larger than the number of fitted parameters
- I can guarantee once you start taking lab classes they will ask for the chi-squared value

What function could be the model?




Demo Time Part 2…

Root Finding

$$\sqrt{1-x^2} = \sin(x)$$

- There are equations we simply cannot solve analytically by hand

```
from scipy.optimize import root

sol = root(func, x0)
```

```
def func(x):
    return np.sqrt(1-x**2)-np.sin(x)
```

The equation you want to find roots of

Initial guess (can be found by plotting)

Solution can be obtained by calling `sol.x`

Root Finding

- We can also use the function from scipy.optimize:

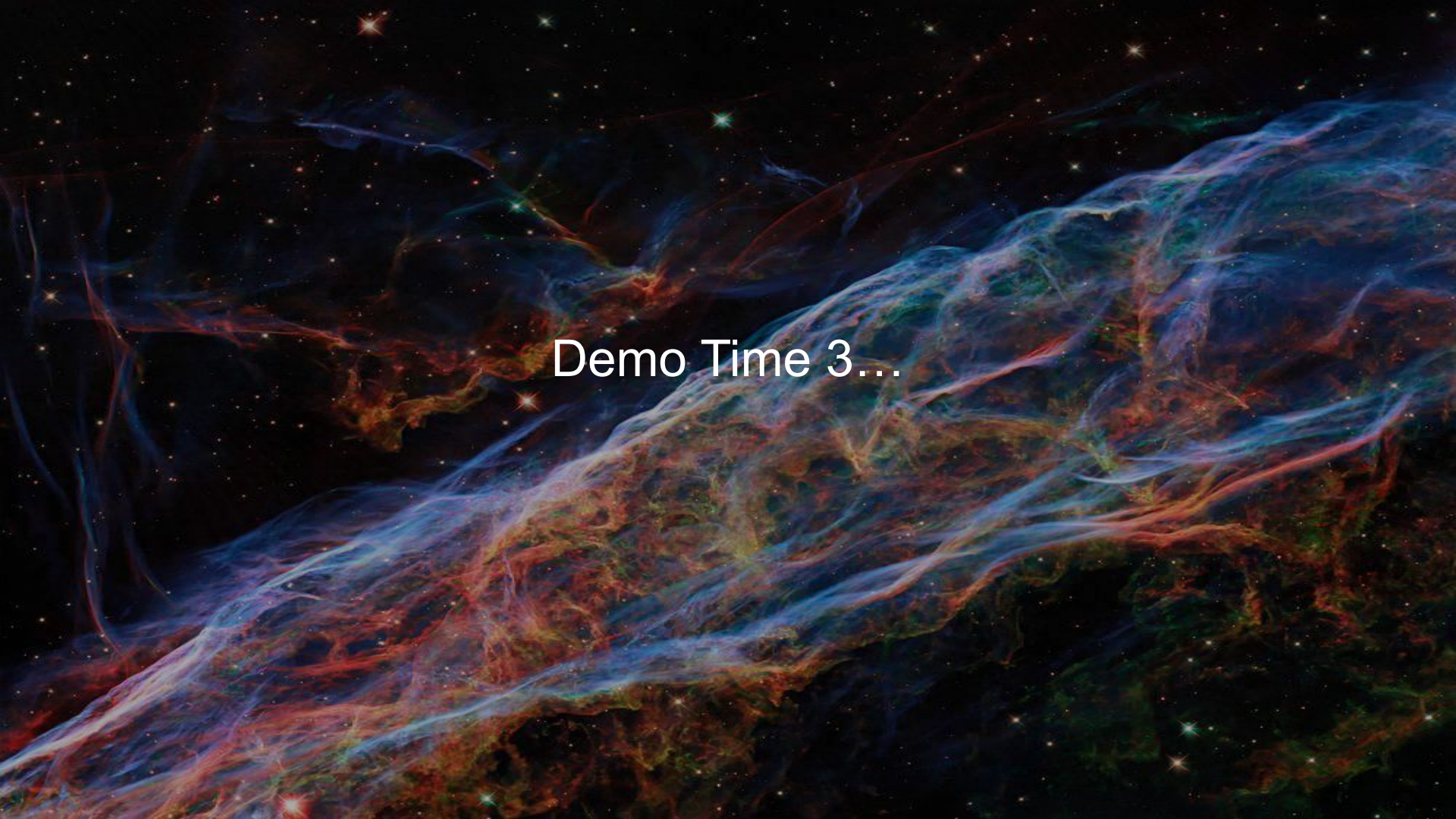
```
from scipy.optimize import fsolve

fsolve(func, x0, args=())
```

- What if you are given this function for the first time? Go to breakout rooms and discuss with others and come back with the explanation for the highlighted variables!

Root Finding

```
fsolve(func, x0, args=())
```

- func: a callable function that takes in at least one arguments and returns output of the same length.
- x0: The starting estimate of the roots of func(x)=0
- args: any other arguments that goes to func that you have defined.
- Outputs an ndarray that contains the solution.


Demo Time 3…