```
## Se rullende_legeme.py for dokumentation, dette script fungerer pÃ¥
## prÃ¦cis samme mÃ¥de.

import matplotlib.pyplot as plt
import numpy as np
import scipy.optimize as scp
import scipy.special as ss
import os

exec(open('Kalibrering/kalibrering.py').read())
exec(open('../Scripts/Statistik.py').read())
exec(open('../Scripts/data_renser.py').read())

fig, ax = plt.subplots(2, 3, figsize = (20, 10))

ax = ax.ravel()
# Importer data

def fit(t,*p):
    a = p[0]
    t0 = p[1]
    c = p[2]
    return np.heaviside(t-t0,1)*(1/2*a*(t-t0)**2)+c

## Plot_data tager nu ogsÃ¥ en vinkel.

def plot_data(data, ax, labels, title, kali, grader):

    theta = grader*(2*np.pi/360)

    sol1 = Data(data)
    x = func(sol1.points, *kali)
    t = sol1.t

    mask = sol1.rinse([[-1, 0.1], [0.4, 0.3], [0.6, 0.45]])

    ax.scatter(t[~mask], x[~mask], color = 'blue', label = 'outliers')
    ax.scatter(t[mask], x[mask], color = 'red', label = 'data points')

    guess_params = [1,-0.17,0.05]

###
    popt,pcov = scp.curve_fit(fit, t[mask], x[mask],
                              guess_params, bounds = ((-10, -0.3, -10),(10, -0.1, 10)))
###


    t_fit = np.linspace(-0.4,1.0,1000)
    ax.plot(t_fit, fit(t_fit, *popt), color = 'k', linewidth = 2,
            label = 'fitted function')

    var_a = round(np.sqrt(np.diag(pcov)[0]), 2)
    eksp_a = round(popt[0], 3)

    error = propagation_function(t_fit, fit, list(popt), pcov)
    ax.fill_between(t_fit, fit(t_fit, *popt) + error,
                    fit(t_fit, *popt) - error, alpha = 0.3)

    ax.set_ylim(-0.2,0.7)
    ax.set_ylabel('x/m')
    ax.set_xlabel('t/s')
    ax.set_title(title)
    ax.legend()
```

```
    print( "Teoretisk a = {}, ".format(round(np.sin(theta)*9.82*0.66, 3))+
          "Eksperimentel a = {} $\pm$ {}".format(eksp_a, var_a))


plot_data("Sol1_11grader", ax[0], labels = None, title = '11 grader',
          kali = kali, grader = 11)

plot_data("Sol1_13grader", ax[1], labels = None, title = '13 grader',
          kali = kali, grader = 13)

plot_data("Sol1", ax[2], labels = None, title = '15 grader',
          kali = kali, grader = 15)

plot_data("Sol1_19grader", ax[3], labels = None, title = '19 grader',
          kali = kali, grader = 19)

plot_data("Sol1_21grader", ax[4], labels = None, title = '21 grader',
          kali = kali, grader = 21)

ax[5].remove()
plt.show()

    ###############

fig.savefig('Plots/vinkel_ruller.png')
```