```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as ss
import scipy.optimize as scp
import os
import csv

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

# fig, ax = plt.subplots(4,3,figsize = (20,12))
# ax = ax.ravel()

fig, ax = plt.subplots(figsize = (16,8))

names = ['0_681', '0_755', '0_805', '0_863', '1_036', '1_091', '1_126', '1_162',
         '1_232', '1_270', '1_319', '1_452']
freks = np.array([0.681, 0.755, 0.805, 0.863, 1.036, 1.091, 1.126, 1.162,
                  1.232, 1.270, 1.319, 1.452])*2*np.pi
driv_freks = freks / (2*np.pi)

A = []
err_A = []

for i in range(len(names)):
    print(i)
    sol1 = Data('Kalibrering/frek' + names[i])
    spÃ¦nding = sol1.points
    ts = sol1.t*1000

    vink = vinkel(spÃ¦nding, *kali)*(360/(2*np.pi))
    # ax[i].plot(ts, vink, 'ro', alpha = 0.4, markersize = 4)


    error = propagation_function(spÃ¦nding, vinkel, list(kali), pcov)

    def sinus(t, *p):
        A = p[0]
        w = p[1]
        k = p[2]
        d = p[3]
        return (A*np.cos(w*t+k)+d)

    guess = [(max(vink)-0.8)/2,freks[i],0.4,0.8]

    popt, pcov2 = scp.curve_fit(sinus, ts, vink, guess,
                                sigma = error, absolute_sigma = True,
                                bounds = ((0, -10, -10, -10),(3, 10, 10, 10)))
    A.append(popt[0])
    err_A.append(np.sqrt(np.diag(pcov2))[0])

def dreven(omega, *p):
    F = p[0]
    omega_0 = p[1]
    b = p[2]
    return F/(np.sqrt(0.585**2*(omega**2 - omega_0**2)**2+b**2*omega**2))

guess = [25,5,2]

popt1, pcov1 = scp.curve_fit(dreven, freks, A, guess,
                             sigma = err_A, absolute_sigma = True)
```

```
# chi_sq, p_value = chi_sq(A,
print(popt1)

frekses = np.linspace(0.681, 1.451, 100)*2*np.pi
ax.plot(frekses, dreven(frekses, *popt1), color = 'black')

plot_propagation(frekses, dreven, popt1, pcov1, ax)

ax.errorbar(freks, A, err_A, fmt = 'o', color = 'red')
ax.plot(freks, A, 'k--')

# chisq, pval = chi_sq(np.array(freks),
#                      np.array(A),
#                      np.array(err_A),
#                      dreven,
#                      popt1,
#                      len(A) - len(popt1))

om_0 = round(np.sqrt(np.diag(pcov1))[1],4)
ax.set_title('Dreven Oscillation'# + '$\chi$ = {}'.format(chisq)
             # + '  $p$ = '.format(pval),
             + '   $\omega_0$ = {} +- {}'.format(round(popt1[1],2), om_0),
             fontsize = 18)
ax.set_xlabel('frekvens', fontsize = 16)
ax.set_ylabel('Amplitude', fontsize = 16)
ax.legend()

fig.savefig('Plots/dreven')

fig, ax = plt.subplots(figsize = (16,8))

forsÃ¸g = [1, 2, 3, 4]
ww = [5.32, 5.30, 5.28, 5.25]

ax.plot(forsÃ¸g, ww, 'bo', label = 'mÃ¥lte $\omega_0$')
ax.plot(5, 6.03, 'ro', label = 'forudset $\omega_0$')

ax.set_title('Sammenligning af $\omega_0$', fontsize = 18)
ax.set_xlabel('forsÃ¸g', fontsize = 16)
ax.set_ylabel('\omega_0', fontsize = 16)
ax.legend()

fig.savefig('Plots/dreven_res')
plt.show()
```