```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.optimize as so
import xlwt
from tempfile import TemporaryFile




###############################################################################
### Definieren der Fit-Funktionen ###
###############################################################################
def exp(x, b, k, d):
    return(np.exp(k*(x-d))+b)

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

def linear(x,m,b):
    return(m*x+b)


###############################################################################
### Abkuerzende Funktionen ###
###############################################################################
def untermenge_daten(x,y,x0,x1,ery=[0]):
    xn=[]
    yn=[]
    eryn=[]
    for i,v in enumerate(x):
        if x0<=v<=x1:
            xn.append(x[i])
            yn.append(y[i])
            eryn.append(ery[i])
    return (np.array(xn),np.array(yn),np.array(eryn))

def fit_plot(model, x_arr, year_mean, year_e, model_type, years ):
    popt, pcov = so.curve_fit(model, x_arr, year_mean,
                              sigma = year_e, absolute_sigma = True)
    res = year_mean - model(x_arr, *popt)
    year_e_arr = np.array(year_e)
    chiq = (res**2 / year_e_arr**2).sum() / (len(year)-3)
    fig1 , (ax1,ax2) = plt.subplots(2,sharex=True,figsize=(10,8))
    fig1.subplots_adjust(hspace=0.05)
    ax1.grid(True, alpha=0.6)
    ax1.set_ylabel('$CO_2$ Konzentration in ppm',fontsize=14)
    ax1.errorbar(x_arr, year_mean ,yerr=year_e,fmt='.', label='$CO_2$-Daten',
                 zorder=1)
    x_gen = np.linspace(0,60,1000)
    ax1.plot(x_gen,model(x_gen, *popt), label='Ausgleichskurve',color='r'
             ,zorder=2)
    ax1.set_xlim(-2,62)
    if model_type == "Gerade":
        ax1.set_title(
            '$\mathrm{CO}_2$-Konzentration Ausgleichsgerade '+years
            ,fontsize=16)
    if model_type == "quadr":
        ax1.set_title(
                '$\mathrm{CO}_2$-Konzentration Ausgleichskurve (quadratisch) '
```

```
                +years
                ,fontsize=16)
    if model_type == "exp":
        ax1.set_title(
                '$\mathrm{CO}_2$-Konzentration Ausgleichskurve (exponentiell) '
                +years
                ,fontsize=16)
    ax1.legend()
    ax2.errorbar(x_arr, res, year_e, fmt = 'o', zorder=1)
    ax2.axhline(0, color = 'k',zorder = 2)
    ax2.grid()
    ax2.set_xlabel('Jahre seit 1958',fontsize=14)
    if years == '1958-2018':
        ax2.set_xticks(np.arange(0,70,10))
        ax2.set_xticklabels(np.arange(1958, 1958+70, 10), fontsize=12)
        ax2.set_ylabel('"Daten minus Ausgleichskurve"',fontsize=14)
        plt.savefig('CO2_ingesamt_'+model_type+'.png',
                    bbox_inches = 'tight', dpi=400)
    if years == '1958-1990':
        ax2.set_xticks(np.arange(0,70,10))
        ax2.set_xticklabels(np.arange(1958,  1958+70, 10), fontsize=12)
        ax2.set_ylabel('"Daten minus Ausgleichskurve"',fontsize=14)
        plt.savefig('CO2_bis1990_'+model_type+'.png',
                    bbox_inches = 'tight', dpi=400)
    plt.show()
    plt.close()
    return(popt, chiq)


################################################################################
### Datenbearbeitung ###
################################################################################

data = np.genfromtxt('Hawaii_2.txt') #Einlesen

#Listen zum Abspreichern der Fitparameter
Lin_params = []
Exp_params = []
Quadr_params = []
Lin_chiq = []
Exp_chiq = []
Quadr_chiq = []

year = np.arange(1958,2019,1)
year_mean = []
year_e = []

#Erstellen von Datenreihen mit den Jahresmittelwerten
for j in range(len(year)):
    temp = []
    for i in range(len(data)):
        if data[i][0] == year[j]:
            temp.append(data[i][1])
    year_mean.append(np.mean(temp))
    #Unsicherheit als Mittelwert-Standardabweichung
    year_e.append( np.std(temp) / np.sqrt( float(len(temp)) ) )



################################################################################
```

```python
### Daten exportieren fuer Tabelle zum Laden in GeoGebra ###
################################################################################
book = xlwt.Workbook()
sheet1 = book.add_sheet('sheet1')

sheet1.write(0, 0, 'Jahreszahlen')
sheet1.write(0, 1, 'CO2 Mittelwerte')
sheet1.write(0, 2, 'CO2 Std')

for i,e in enumerate(year):
    sheet1.write(i+1,0,e)
for i,e in enumerate(year_mean):
    sheet1.write(i+1,1,e)
for i,e in enumerate(year_e):
    sheet1.write(i+1,2,e)

name = "random.xls"
book.save(name)
book.save(TemporaryFile())




''' Gesamte Daten '''

################################################################################
### Rohdarstellung der Jahresdurchschnittswerte ###
################################################################################

x_arr = np.arange(0, len(year),1)
fig1 , ax1 = plt.subplots(1,figsize=(10,6))
ax1.grid(True, alpha=0.6)
ax1.set_ylabel('$CO_2$ Konzentration in ppm',fontsize=14)
ax1.errorbar(year, year_mean ,yerr=year_e,fmt='.', label='$CO_2$-Daten',
             zorder=1)
ax1.set_title(
        'Durchschnittliche (jaehlich) $\mathrm{CO}_2$-Konzentration 1958-2018',
        fontsize=16)
ylim = ax1.get_ylim()
ax1.legend()
plt.savefig('CO2_ingesamt_ohneFit.png', bbox_inches = 'tight', dpi=400)
plt.show()
plt.close()


################################################################################
### Linearer Fit an alle Daten ###
################################################################################

temp = fit_plot(linear, x_arr, year_mean, year_e, 'Gerade', '1958-2018' )
Lin_params.append(temp[0])
Lin_chiq.append(temp[1])

################################################################################
### Quadratischer Fit an alle Daten ###
################################################################################

temp = fit_plot(sq, x_arr, year_mean, year_e, 'quadr', '1958-2018' )
Quadr_params.append(temp[0])
Quadr_chiq.append(temp[1])
```

```python
##############################################################################
### Exponentieller Fit an alle Daten ###
##############################################################################

temp = fit_plot(exp, x_arr, year_mean, year_e, 'exp', '1958-2018' )
Exp_params.append(temp[0])
Exp_chiq.append(temp[1])



''' BIS 1990 '''

##############################################################################
### Datenauswahl bis 1990 ###
##############################################################################
end_year = 1990
year, year_mean, year_e = untermenge_daten(year, year_mean, 1958, end_year,
                                              ery = year_e)
x_arr = np.arange(0, len(year),1)


##############################################################################
### Rohdarstellung der Jahresdurchschnittswerte ###
##############################################################################
fig1 , ax1 = plt.subplots(1,figsize=(10,6))
ax1.grid(True, alpha=0.6)
ax1.set_ylabel('$CO_2$ Konzentration in ppm',fontsize=14)
ax1.errorbar(year, year_mean ,yerr=year_e,fmt='.', label='$CO_2$-Daten',
             zorder=1)
ax1.set_title(
        'Durchschnittliche (jaehlich) $\mathrm{CO}_2$-Konzentration 1958-1990',
        fontsize=16)
ax1.legend()
plt.savefig('CO2_bisxy_ohneFit.png', bbox_inches = 'tight', dpi=400)
plt.show()
plt.close()

##############################################################################
### Linearer Fit bis 1990 ###
##############################################################################
temp = fit_plot(linear, x_arr, year_mean, year_e, 'Gerade', '1958-1990' )
Lin_params.append(temp[0])
Lin_chiq.append(temp[1])

##############################################################################
### Quadratischer Fit bis 1990 ###
##############################################################################

temp = fit_plot(sq, x_arr, year_mean, year_e, 'quadr', '1958-1990' )
Quadr_params.append(temp[0])
Quadr_chiq.append(temp[1])

##############################################################################
### Exponentieller Fit bis 1990 ###
##############################################################################

temp = fit_plot(exp, x_arr, year_mean, year_e, 'exp', '1958-1990' )
Exp_params.append(temp[0])
Exp_chiq.append(temp[1])
```