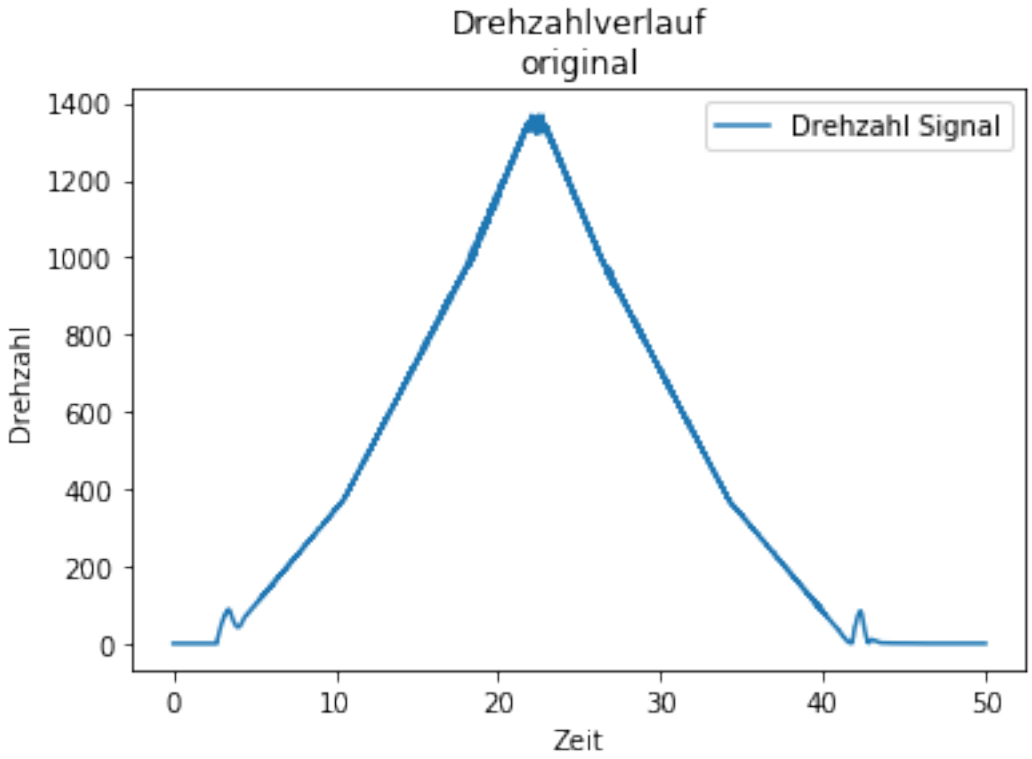
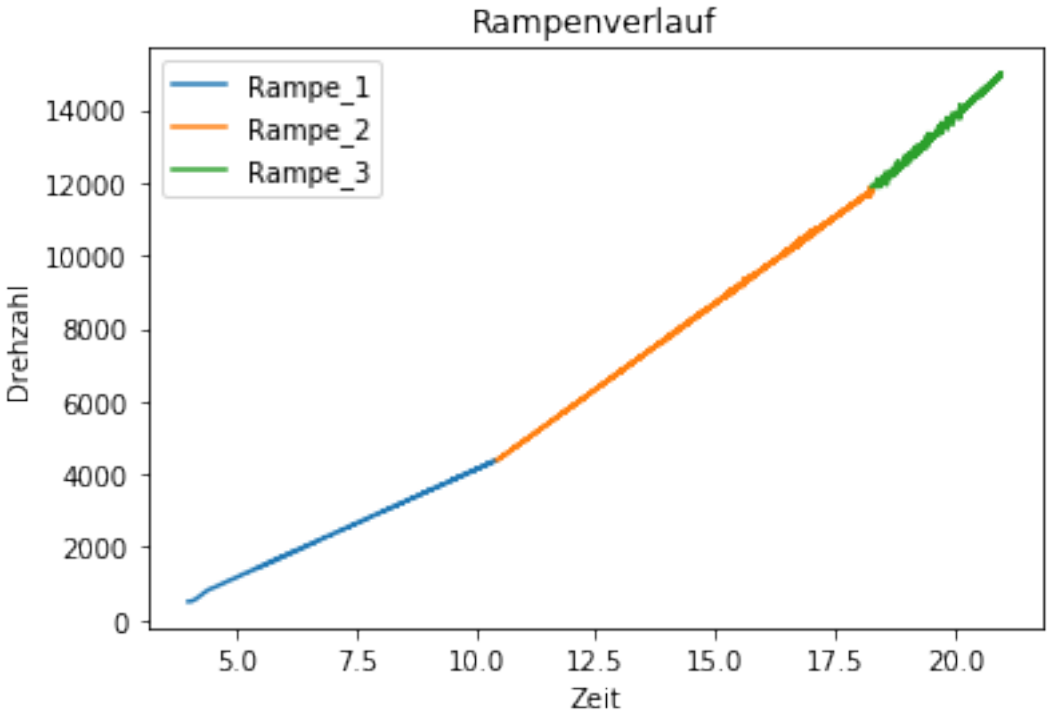
```
#!/usr/bin/env python
# coding: utf-8


import csv # import csv module
import matplotlib.pyplot as plt # import plotting module
# plt.rcParams['figure.figsize'] = [20, 10]
import numpy as np
import scipy as sp
from scipy import stats


# open CSV file
time = []
mic = []
vib = []
rpm = []
file_name = "C:\\Users\\XXXX\\Desktop\\test.csv"
with open(file_name,'rb') as f:
    reader = csv.reader(f, delimiter=';')
    for row in reader:
        # print row
        time.append(float(row[0].replace(',', '.')))
        mic.append(float(row[1].replace(',', '.')))
        vib.append(float(row[2].replace(',', '.')))
    f.close()



'''
Import the first ramp

'''
t1 = []
r1 = []
schw1 = []
for i, p in enumerate(rpm):
    if p*11.883 >= 100 and p*11.883 < 4396.8  and time[i]> 4:
        t1.append(time[i])
        r1.append(p*11.883)
        schw1.append(vib[i])
    if p*11.883 > 4396.8:
        break



'''
Import the second ramp

'''
```

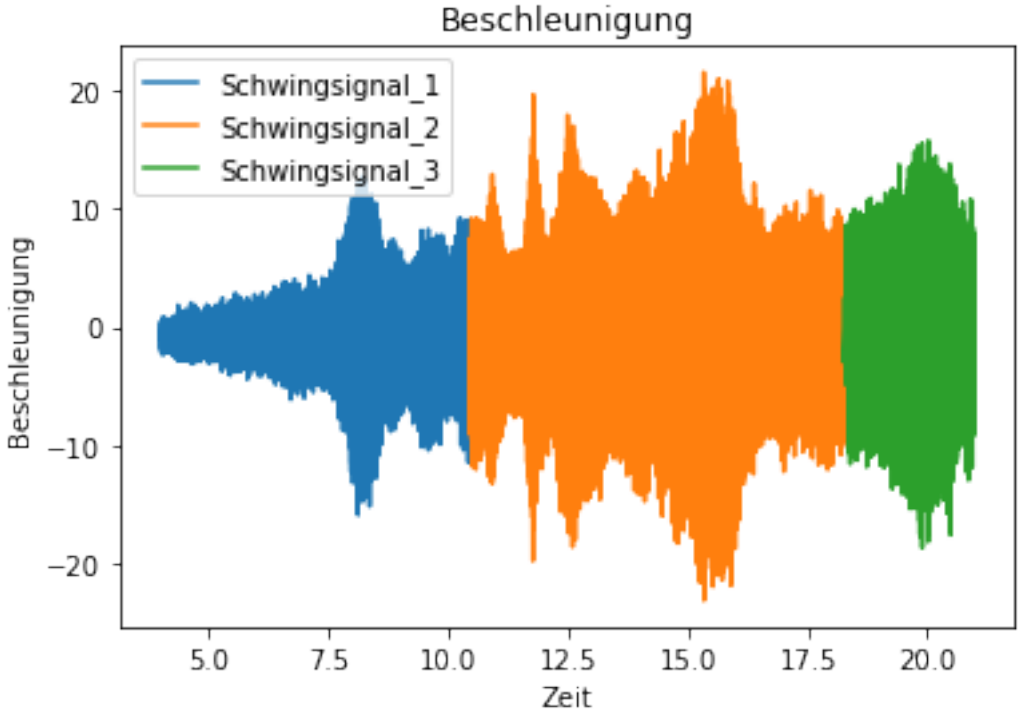
```
t2 = []
r2 = []
schw2 = []
for i, p in enumerate(rpm):
    if p*11.883 >= 4396.8 and p*11.883 < 11883.2 :
        t2.append(time[i])
        r2.append(p*11.883)
        schw2.append(vib[i])
    if p*11.883 > 11883.2:
        break




'''
Import the third ramp

'''

t3 = []
r3 = []
schw3 = []
for i, p in enumerate(rpm):
    if p*11.883 >= 11883.2  and p*11.883 < 15000 :
        t3.append(time[i])
        r3.append(p*11.883)
        schw3.append(vib[i])
    if p*11.883 > 15000:
        break



'''
Plot all sections
'''


plt.plot(time,rpm,label="Drehzahl Signal")
plt.xlabel('Zeit')
plt.ylabel('Drehzahl')
plt.title('Drehzahlverlauf\noriginal')
plt.legend()
plt.show()

plt.plot(t1,r1,label="Rampe_1")
plt.plot(t2,r2,label="Rampe_2")
plt.plot(t3,r3,label="Rampe_3")
plt.xlabel('Zeit')
plt.ylabel('Drehzahl')
plt.title('Rampenverlauf')
plt.legend()
```

```
plt.show()
```

```
plt.plot(t1,schw1,label="Schwingsignal_1")
plt.plot(t2,schw2,label="Schwingsignal_2")
plt.plot(t3,schw3,label="Schwingsignal_3")
plt.xlabel('Zeit')
plt.ylabel('Beschleunigung')
plt.title('Beschleunigung')
plt.legend()
plt.show()
```



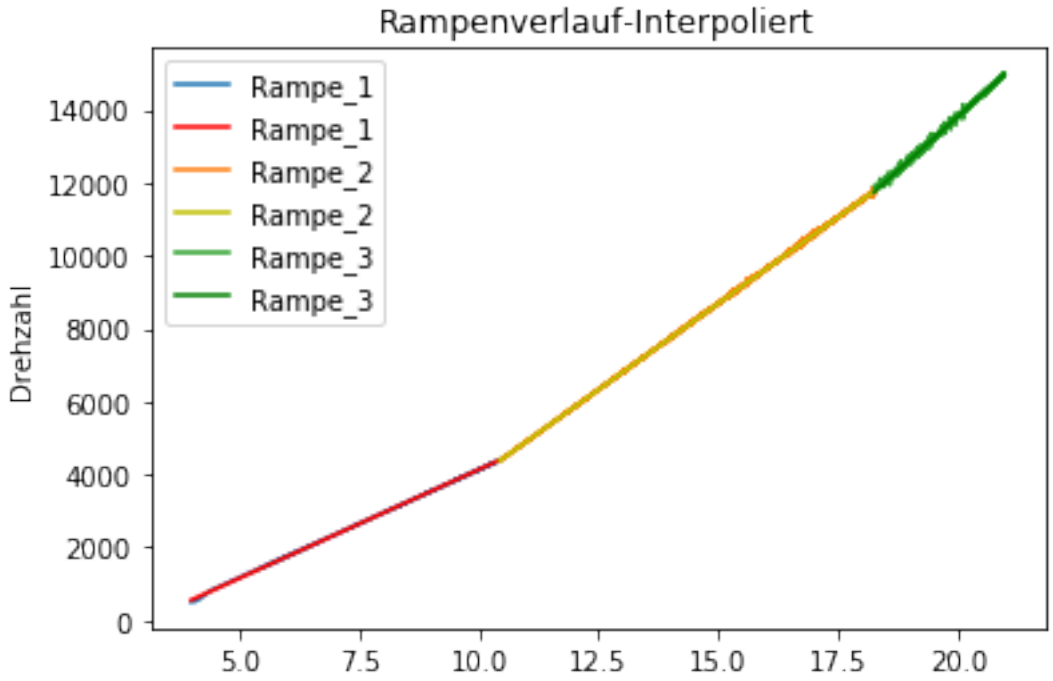




'''
linear regression for each ramp
'''

slope, intercept, r_value, p_value, std_err = stats.linregress(t1,r1)

```
line = slope*np.float32(t1)+intercept


slope, intercept, r_value, p_value, std_err = stats.linregress(t2,r2)
line2 = slope*np.float32(t2)+intercept

slope, intercept, r_value, p_value, std_err = stats.linregress(t3,r3)
line3 = slope*np.float32(t3)+intercept



'''
Plot linear regresion of all ramps
'''

plt.plot(t1,r1,t1,line,'r',label=("Rampe_1"))
plt.plot(t2,r2,t2,line2,'y',label="Rampe_2")
plt.plot(t3,r3,t3,line3,'g',label="Rampe_3")
plt.ylabel('Drehzahl')
plt.title('Rampenverlauf-Interpoliert')
plt.legend()
plt.show()
```

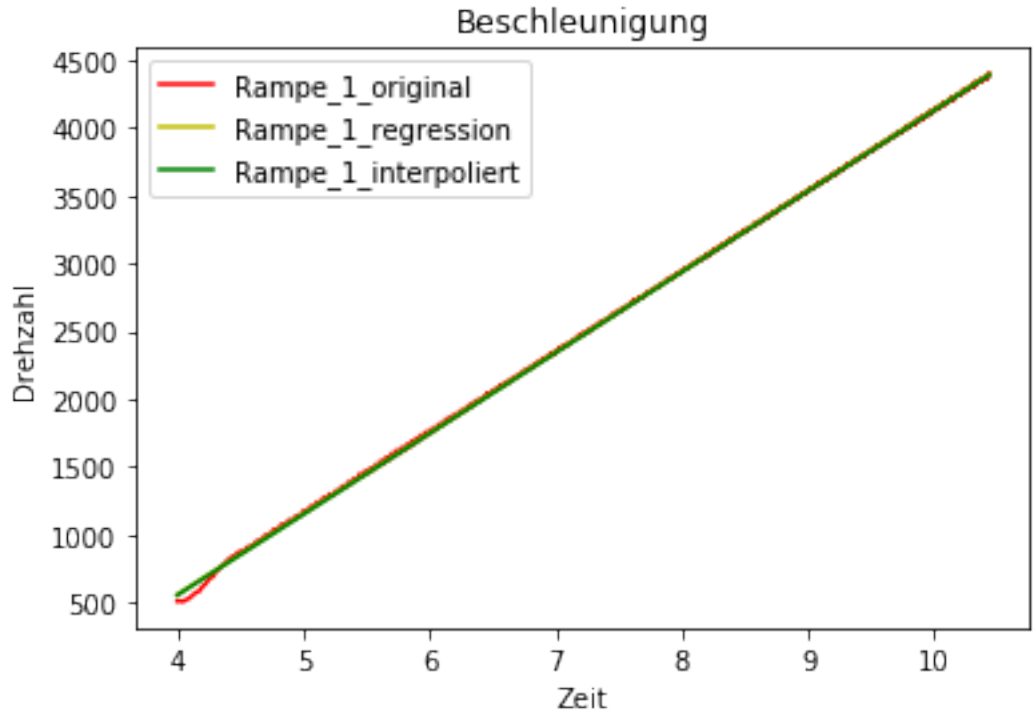


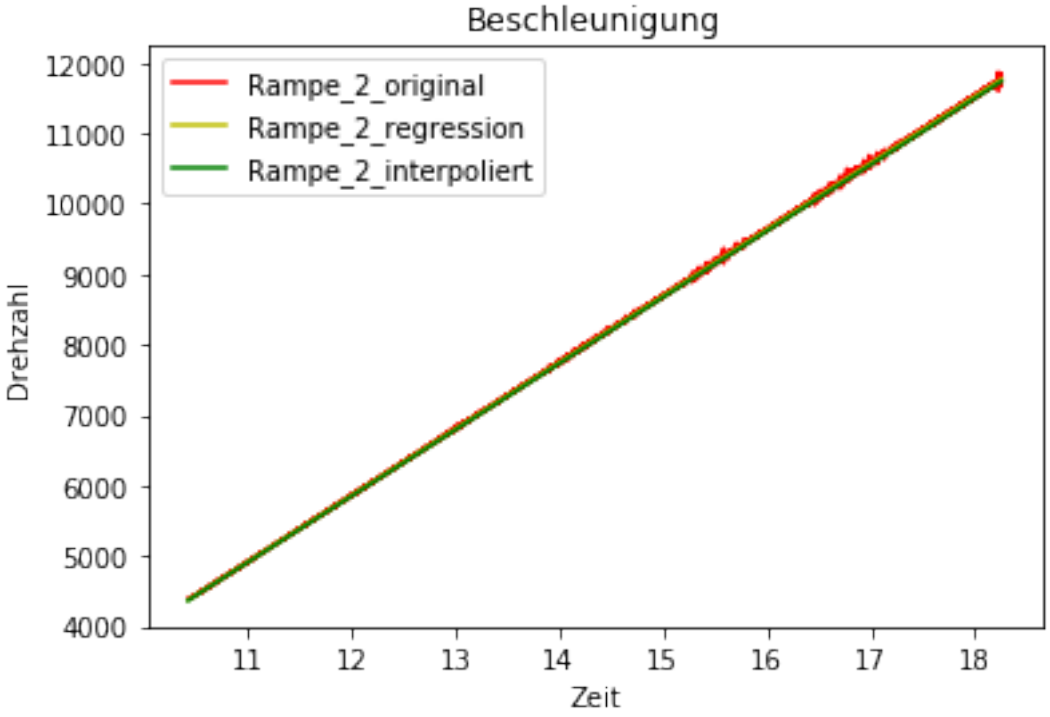

```
'''
calculate interpolation coefs for each ramp

'''

a1 = (line1[-1]-line1[-2000])/(t1[-1]-t1[-2000])
b1 = line1[0]-(a1*t1[0])
```

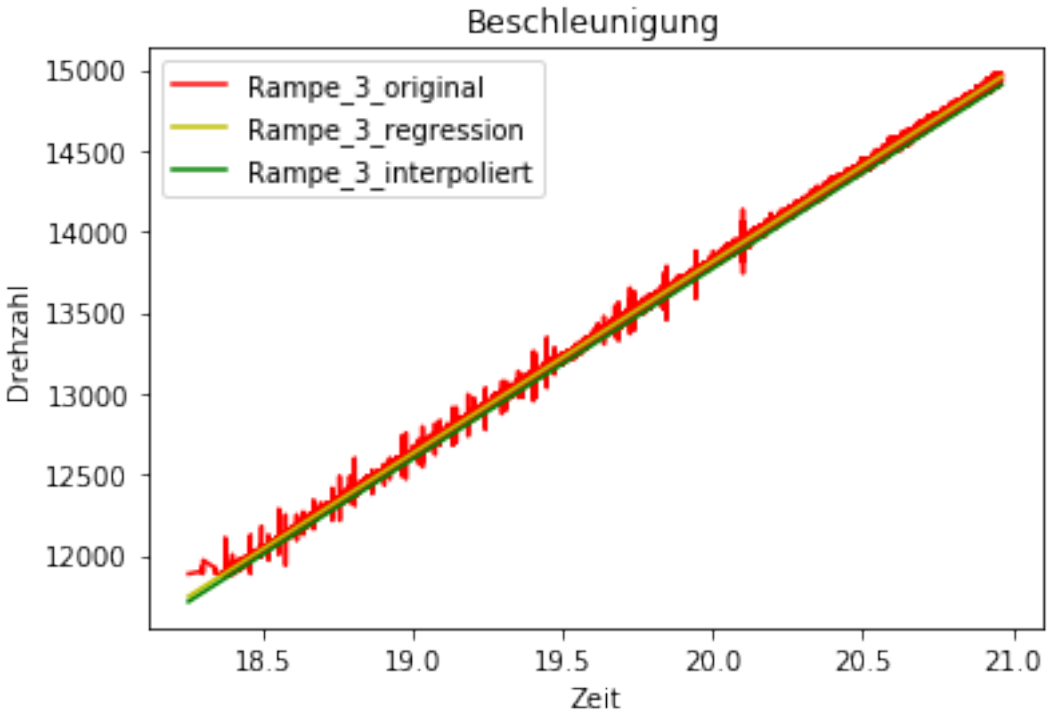
```
print ("a1= {} , b1= {}").format(a1,b1)

a2 = (line2[-1]-line2[-2000])/(t2[-1]-t2[-2000])
b2 = line2[0]-(a2*t2[0])
print ("a2= {} , b2= {}").format(a2,b2)

a3 = (line3[-1]-line3[-2000])/(t3[-1]-t3[-2000])
b3 = line3[0]-(a3*t3[0])
print ("a3= {} , b3= {}").format(a3,b3)



a4 = (line4[-1]-line4[-2000])/(t4[-1]-t4[-2000])
b4 = line4[0]-(a4*t4[0])
print ("a4= {} , b4= {}").format(a4,b4)

a5 = (line5[-1]-line5[-2000])/(t5[-1]-t5[-2000])
b5 = line5[0]-(a5*t5[0])
print ("a5= {} , b5= {}").format(a5,b5)

a6 = (line6[-1]-line6[-2000])/(t6[-1]-t6[-2000])
b6 = line6[0]-(a6*t6[0])
print ("a6= {} , b6= {}").format(a6,b6)



'''
interpolation coefficients
'''



a1= 595.71582571
b1= -1830.55869696

a2= 942.893322415
b2= -5474.28052147

a3= 1178.18283953
b3= -9791.40827982

r1_inter = []
for i,t in enumerate(t1):
    r1_inter.append(a1*t+b1)

r2_inter = []
for i,t in enumerate(t2):
    r2_inter.append(a2*t+b2)

r3_inter = []
for i,t in enumerate(t3):
    r3_inter.append(a3*t+b3)
```

'''
Plot results compare regression/interpolation
'''

```
plt.plot(t1,r1,'r',label="Rampe_1_original")
plt.plot(t1,line,'y',label="Rampe_1_regression")
plt.plot(t1,r1_inter,'g',label="Rampe_1_interpoliert")
plt.xlabel('Zeit')
plt.ylabel('Drehzahl')
plt.title('Beschleunigung')
plt.legend()
plt.show()

plt.plot(t2,r2,'r',label="Rampe_2_original")
plt.plot(t2,line2,'y',label="Rampe_2_regression")
plt.plot(t2,r2_inter,'g',label="Rampe_2_interpoliert")
plt.xlabel('Zeit')
plt.ylabel('Drehzahl')
plt.title('Beschleunigung')
plt.legend()


plt.show()

plt.plot(t3,r3,'r',label="Rampe_3_original")
plt.plot(t3,line3,'y',label="Rampe_3_regression")
plt.plot(t3,r3_inter,'g',label="Rampe_3_interpoliert")
plt.xlabel('Zeit')
plt.ylabel('Drehzahl')
plt.title('Beschleunigung')
plt.legend()
plt.show()
```


Beschleunigung
Rampe_1_original
Rampe_1_regression
Rampe_1_interpoliert
Drehzahl
Zeit
4500
4000
3500
3000
2500
2000
1500
1000
500
4
5
6
7
8
9
10


Beschleunigung
Rampe_2_original
Rampe_2_regression
Rampe_2_interpoliert
Drehzahl
Zeit
12000
11000
10000
9000
8000
7000
6000
5000
4000
11
12
13
14
15
16
17
18


Beschleunigung
Rampe_3_original
Rampe_3_regression
Rampe_3_interpoliert
Drehzahl
15000
14500
14000
13500
13000
12500
12000
18.5
19.0
19.5
20.0
20.5
21.0
Zeit