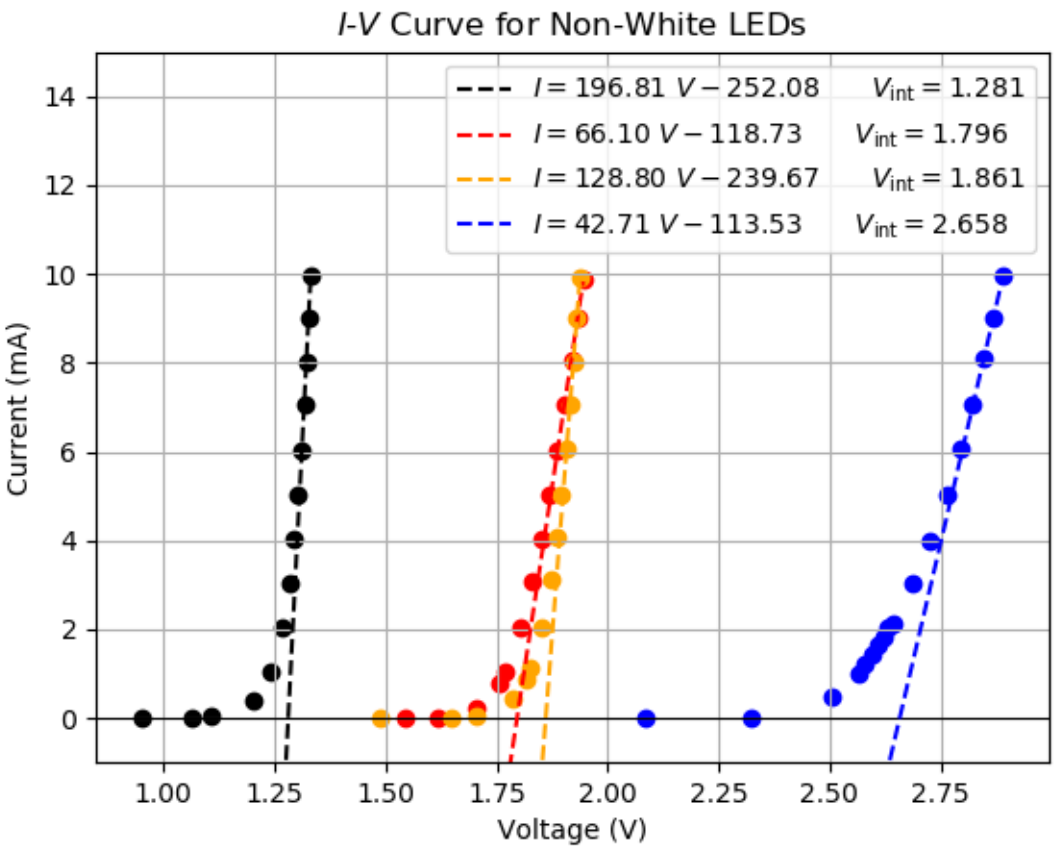
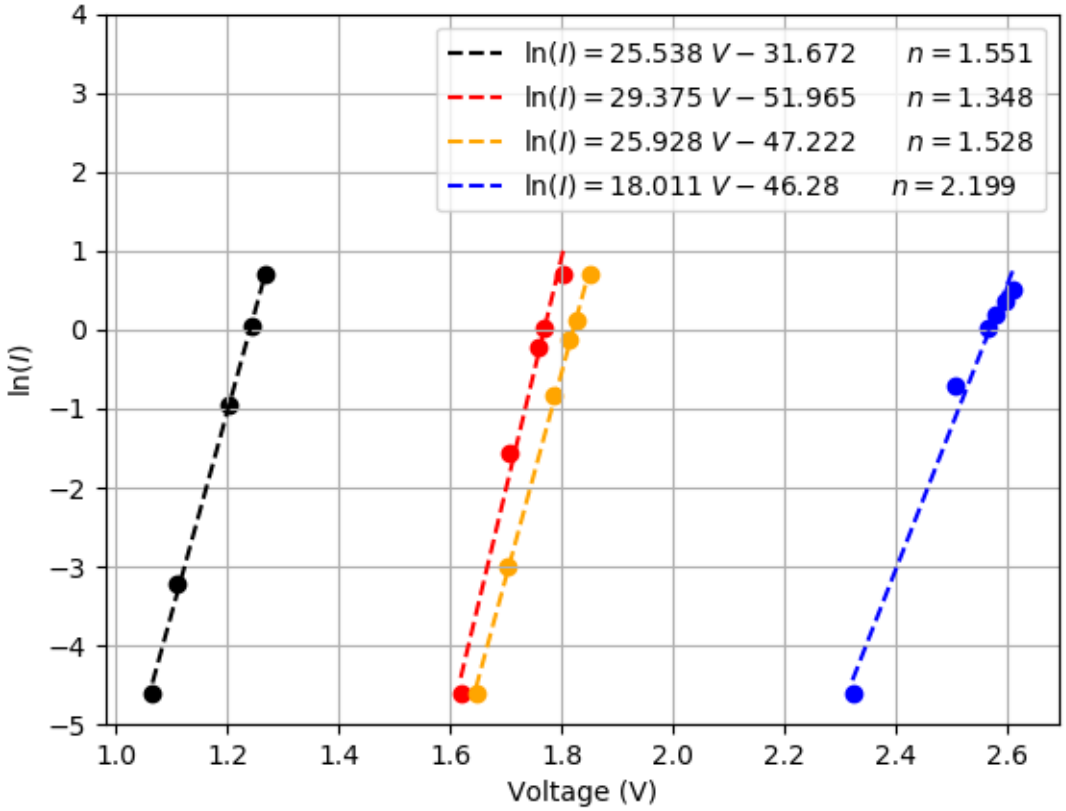
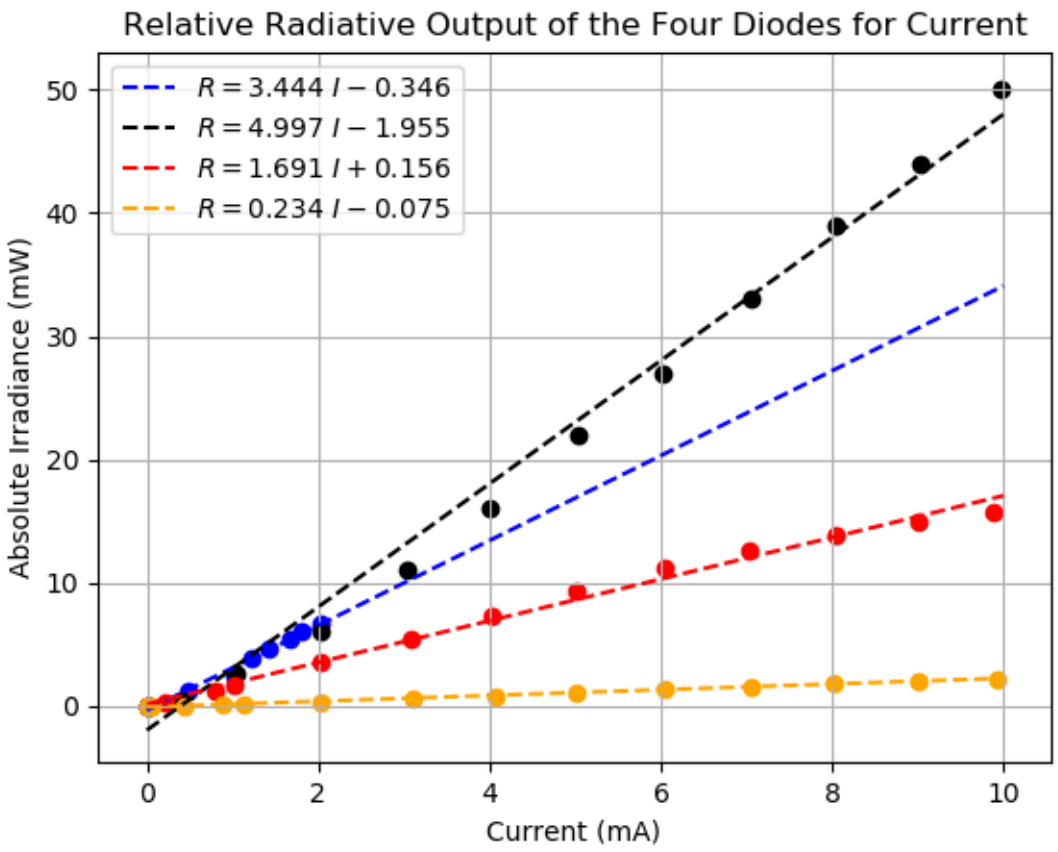


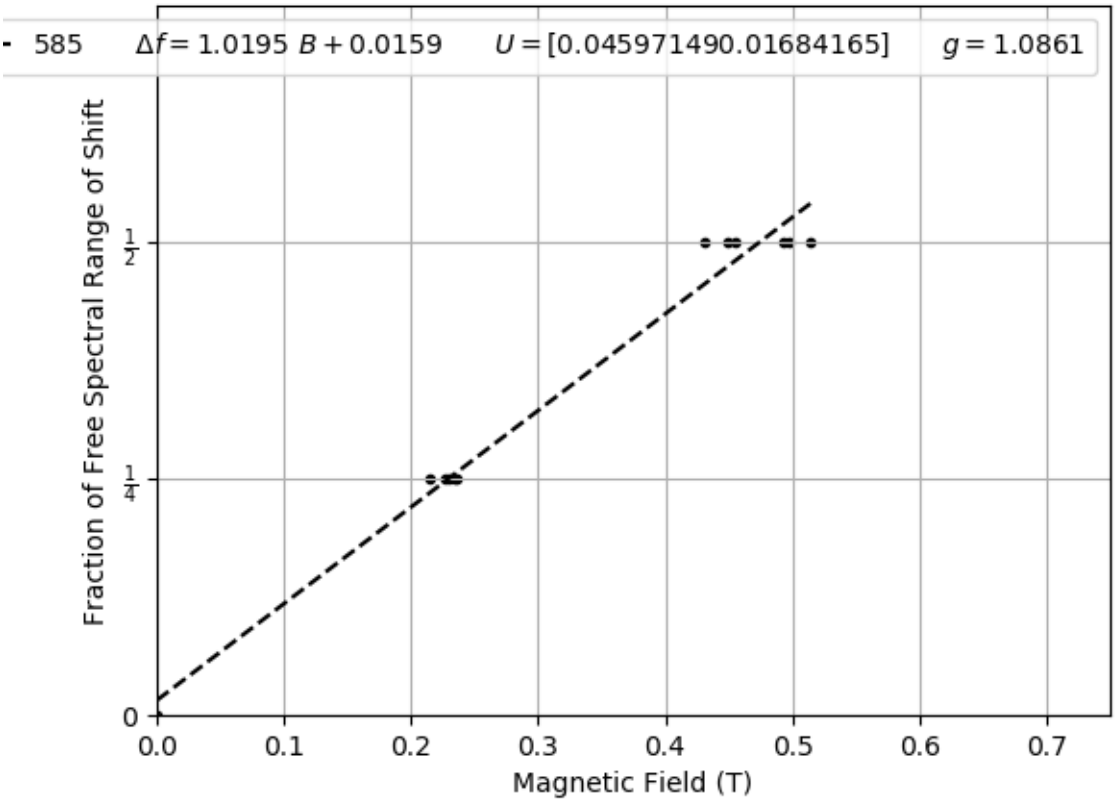

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

boltz = 1.380649*10**(-23)
room_temp = 293
elem = 1.60217662*10**(-19)

class IV:
    def __init__(self, filename):
        self.df = pd.read_csv(filename, sep='\t').sort_values(by='V')
        self.v = self.df['V']
        self.i = self.df['I']

    def ivPlot(self, ax, color, points):
        ax.scatter(self.v, self.i, color=color, label='_nolabel_')
        #self.ax.errorbar(self.i, self.v, yerr=self.df['Ve'], fmt='none')

        lx = self.v[-(points+2):].min()
        mx = self.v.max()
        m, b = np.polyfit(self.v[-points:], self.i[-points:], 1)
        x = np.linspace(lx*0.95, mx, 10)
        vint = -b/m
        label = r'$I=%1.2f \ V%1.2f\qquad V_\mathrm{int}=%1.3f$' % (m, b, vint)
        ax.plot(x, m*x + b, linestyle='--', color=color, label=label)

    def logIvPlot(self, ax, color, points):
        ax.grid(True)
        x = self.v[1:points]
        y = np.log(self.i[1:points])
        ax.scatter(x, y, color=color, label='_no_label_')

        lx = np.min(x)
        mx = np.max(x)
        m, b = np.polyfit(x, y, 1)
        n = elem/(m*boltz*room_temp)
        xval = np.linspace(lx, mx, 10)
        label = r'$\ln (I)={}\ V{}\qquad n = {}$'.format(round(m,3), round(b,3), round(n, 3))
        ax.plot(xval, m*xval + b, linestyle='--', color=color, label=label)
```



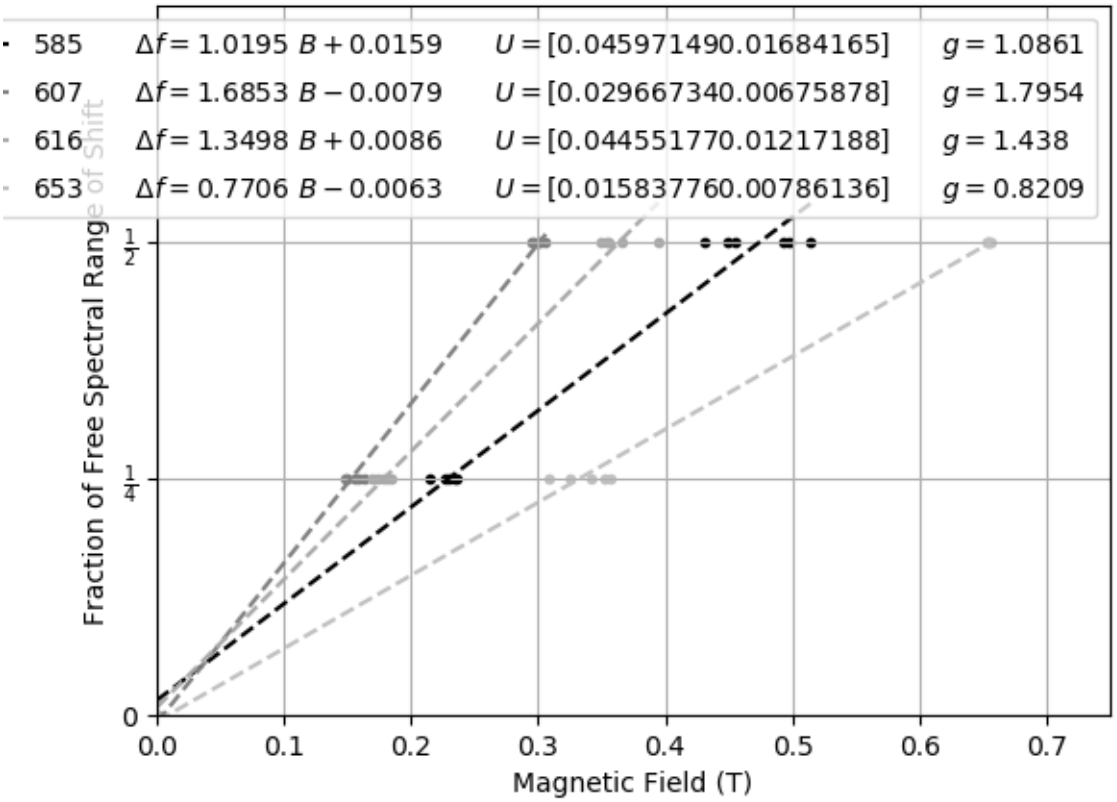

```
blue = IV("bIVR.tsv")
orange = IV("oIVR.tsv")
red = IV("rIVR.tsv")
ir = IV("irIVR.tsv")

fig, ax = plt.subplots()
ax.grid(True)
ax.set_title(r'$I$-$V$ Curve for Non-White LEDs')
ax.set_xlabel('Voltage (V)')
ax.set_ylabel('Current (mA)')
ax.set_ylim(-1, 15)
ax.axhline(0, color='black', linewidth=0.75)

ir.ivPlot(ax, "black", 5)
red.ivPlot(ax, "red", 5)
orange.ivPlot(ax, "orange", 5)
blue.ivPlot(ax, "blue", 4)
plt.legend()
plt.savefig('./images/linear_IV.png')

fig2, ax2 = plt.subplots()
ax2.set_xlabel('Voltage (V)')
ax2.set_ylabel(r'$\ln(I)$')
ax2.set_ylim(-5,4)
ir.logIvPlot(ax2, "black", 6)
red.logIvPlot(ax2, "red", 6)
orange.logIvPlot(ax2, "orange", 7)
blue.logIvPlot(ax2, "blue", 7)
plt.legend(loc='upper right')
plt.savefig('./images/log_IV.png')
```




```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

fig, ax = plt.subplots()

class IR:
    def __init__(self, filename, intTime):
        self.df = pd.read_csv(filename, sep='\t').dropna().sort_values(by='I')
        self.i = self.df['I']
        self.r = self.df['R']/(1000*intTime)

    def irPlot(self, color):
        ax.grid(True)
        ax.set_xlabel('Current (mA)')
        ax.set_ylabel(r'Absolute Irradiance (mW)')
        #ax.errorbar(self.i, self.r, yerr=self.df['Re'], fmt='none')

        m, b = np.polyfit(self.i, self.r, 1)
        x = np.linspace(0,10,5)
        label = r'$R={}\ I{}{}$'.format(round(m,3),'+' if b > 0 else '', round(b,3))
        ax.plot(x, m*x+b, linestyle='--', color=color, label=label)

        ax.scatter(self.i, self.r, color=color, marker='o', label='_nolabel_')

blue = IR("bIVR.tsv", 1)
orange = IR("oIVR.tsv", 1)
red = IR("rIVR.tsv", 1)
ir = IR("irIVR.tsv", 1)

ax.set_title('Relative Radiative Output of the Four Diodes for Current')
blue.irPlot("blue")
ir.irPlot("black")
red.irPlot("red")
orange.irPlot("orange")
plt.legend()
plt.savefig('./images/ri.png')
```




```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

bohrmag = 9.274*10**(-24)
planck = 6.626*10**(-34)

def magForI(currents):
    B = 249+8.2999*currents+3.9277*10**(-3)*currents**2 - 5.5726*10**(-6)*currents**3
    B /=10000
    return B

class Zeem:
    def __init__(self, filename):
        df = pd.read_csv(filename, sep='\t')
        self.b = magForI(df['I'])
        self.b = sorted(np.append(self.b, 0))
        self.df = df['df']
        self.df = sorted(np.append(self.df, 0))

    def plot(self, axes, label, color):
        axes.scatter(self.b, self.df, marker='.', color=color, label='_nolabel_')

        fit, cov = np.polyfit(self.b, self.df, 1, cov=True)
        u = np.sqrt(np.diag(cov))
        print(u)
        lx = np.min(self.b)
        mx = np.max(self.b)

        m, b = fit
        x = np.linspace(0, mx, 10)
        label = str(label) + r'$\qquad$'
        label += r'$\Delta f={}\ B{}{}\qquad U={}$'.format(round(m, 4), '+' if b>0 else '', round(b, 4), u)

        g = planck*3*10**8*m/(bohrmag*1*2*10.06*10**(-3))
        label += r'$\qquad g = {}$'.format(round(g, 4))

        axes.plot(x, m*x+b, color=color, linestyle='--', label=label)
```




```
fig, ax = plt.subplots()
ax.grid(True)
ax.set_ylim(0, 0.75)
ax.set_xlim(0, 0.75)
ax.set_yticks([0,0.25,0.5])
#ax.set_yticklabels(['0', r'$\frac{\Delta f_{fsr}}{4}$', r'$\frac{\Delta f_{fsr}}{2}$'])
ax.set_yticklabels(['0', r'$\frac{1}{4}$', r'$\frac{1}{2}$'])
ax.set_xlabel('Magnetic Field (T)')
ax.set_ylabel('Fraction of Free Spectral Range of Shift')

w585 = Zeem('585.tsv')
w585.plot(ax, 585, 'black')
plt.legend()
plt.savefig('./images/zeem1.png')

w607 = Zeem('607.tsv')
w607.plot(ax, 607, 'gray')

w616 = Zeem('616.tsv')
w616.plot(ax, 616, 'darkgray')

w653 = Zeem('653.tsv')
w653.plot(ax, 653, 'silver')

ax.legend(loc='upper right')
plt.savefig('./images/zeem.png')
#plt.show()
```