NMR Analysis

March 9, 2019

```
In [2]: #import stuff
        import numpy as np
        import pandas as pd
        import matplotlib as mpl
        import matplotlib.pyplot as plt
        from scipy.optimize import curve_fit
```

```
In [3]: #pull and define data
        raw = pd.read_csv('T1 Data.csv')
        t = raw.iloc[:,0].values
        V = raw.iloc[:,1].values

        raw2 = pd.read_csv('T2 Data.csv')
        t2 = raw2.iloc[:,0].values
        V2 = raw2.iloc[:,1].values
```

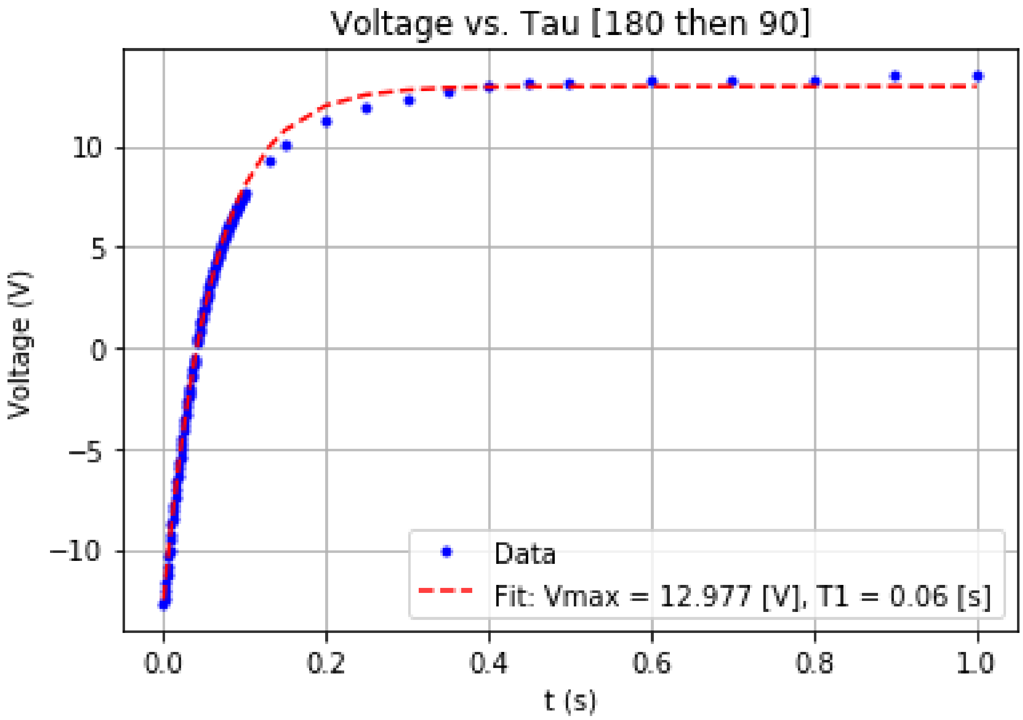
```
In [4]: #store data as an array
        MagData=[t,V]
        #print(MagData)

        #then flip some stuff
        ind=np.argmin(V)
        for i in range(ind):
            V[i]=-V[i]
```

```
In [6]: #now plot with fit...
        plt.close('all')
        plt.plot(t,V,"b.",label="Data")
        plt.title("Voltage vs. Tau [180 then 90]")
        plt.xlabel("t (s)")
        plt.ylabel("Voltage (V)")
        plt.legend()
        def func(t,Vmax,T1):
            return Vmax*(1.0-2.0*np.exp(-t/T1))
        popt,pcov=curve_fit(func,t,V,p0=(13,0.6))
        print("Vmax,T1",popt)
        plt.plot(t,func(t,*popt),'r--',label='Fit: Vmax = %3.3f [V],\
         T1 = %4.2f [s]' % tuple(popt))
        plt.grid()
        plt.legend()

Vmax,T1 [12.97701552  0.06075578]


Out[6]: <matplotlib.legend.Legend at 0x1af3d356630>
```



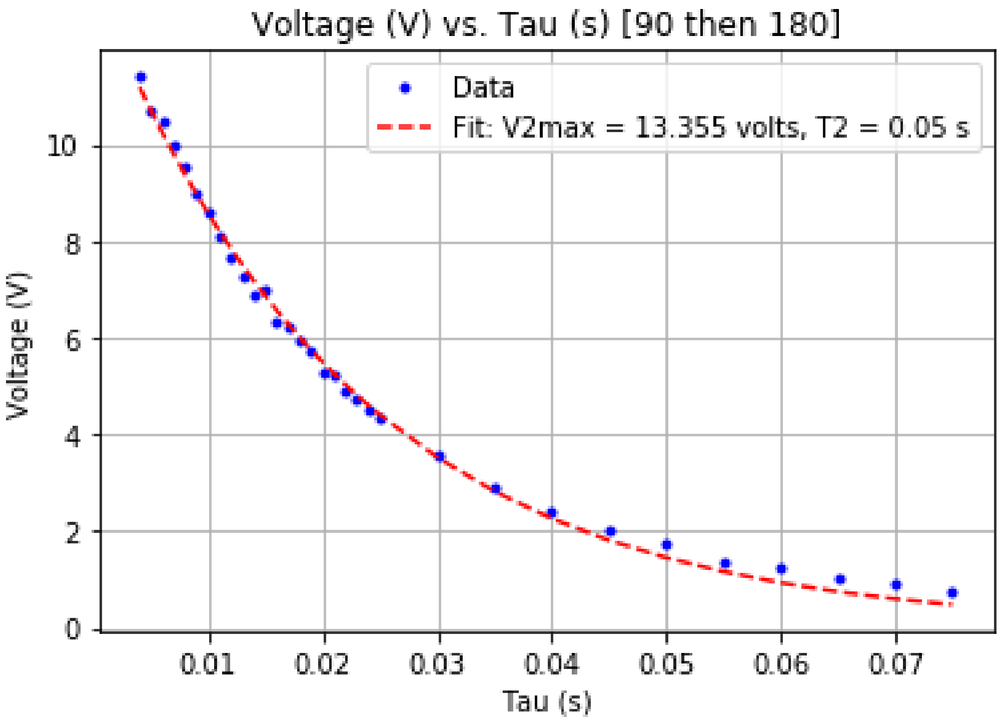
```
In [47]: #Standard Deviations
         perr = np.sqrt(np.diag(pcov))
         print (perr)

[0.04540425 0.00019555]


In [7]: #now do the above, but for T2...
        plt.close('all')
        plt.plot(t2,V2,"b.",label="Data")
        plt.title("Voltage (V) vs. Tau (s) [90 then 180]")
        plt.xlabel("Tau (s)")
        plt.ylabel("Voltage (V)")
        plt.legend()
        #
        #Define Function to Fit
        #
        def func(t2,V2max,T2):
            return V2max*np.exp(-2*t2/T2)
        #
        #Set Initial Guess of Fit Parameters and Curve Fit
        #
        popt,pcov=curve_fit(func,t2,V2,p0=(10,0.04))
        print("V2max,T2",popt)
        plt.plot(t2,func(t2,*popt),'r--',label='Fit: V2max = %3.3f volts,\
         T2 = %4.2f s' % tuple(popt))
        plt.grid()
        plt.legend()
```

```
V2max,T2 [13.35461813  0.0450339 ]
```

```
Out[7]: <matplotlib.legend.Legend at 0x1af3d3c1e48>
```



```
In [8]: #Standard Deviation T2
        perr = np.sqrt(np.diag(pcov))
        print (perr)
```

```
[0.13451543 0.00073203]
```

```
In [ ]:
```