Bracket_Number

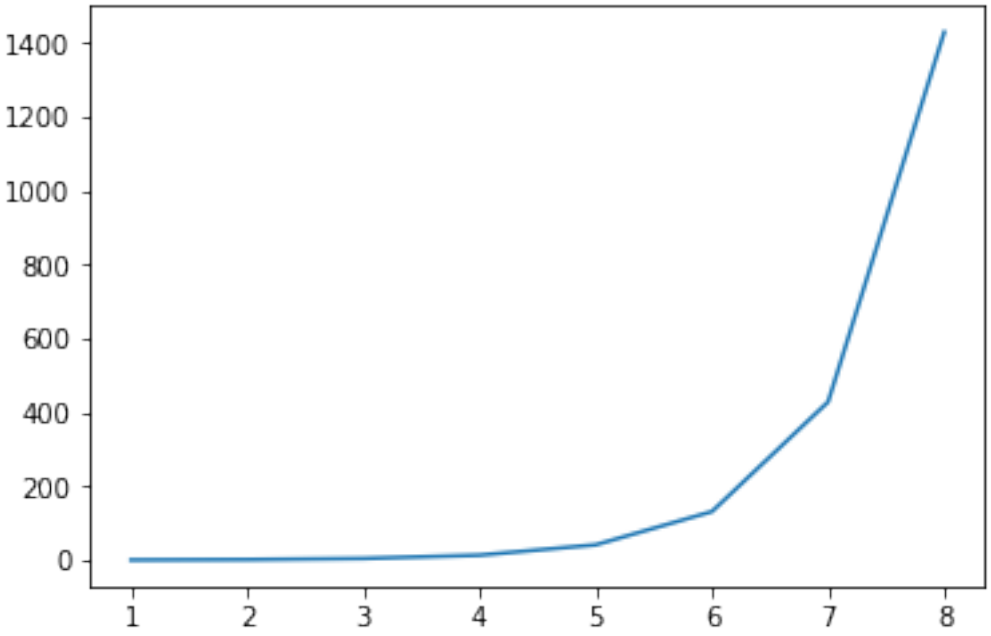
January 18, 2018

In [1]:
```
def dec_to_bin_string(n):
    bin_list=[]
    for i in range(2,len(list(bin(n)))):
        bin_list.append(list(bin(n))[i])
    return ''.join(bin_list)

def all_n_digit_bin(n):
    dec_eq=0
    all_bin=[]
    while (len(dec_to_bin_string(dec_eq))<=n):
        if (len(dec_to_bin_string(dec_eq))==n):
            all_bin.append(dec_to_bin_string(dec_eq))
        dec_eq = dec_eq+1
    return all_bin

def Q_valid_bracket(bin_string):
    if bin_string[0]!='1':
        return False
    #replace_0_by_neg1
    bin_list=list(bin_string)
    zero2n1_arr =[]
    for ticker in bin_list:
        if ticker == '1':
            zero2n1_arr.append(1)
        if ticker == '0':
            zero2n1_arr.append(-1)
    summation = 0
    for ticker in zero2n1_arr:
        summation = summation + ticker
        if summation<0: return False
    if summation != 0:
        return False
    else: return True

def nb_valid_in_size_n_pair(n):
    return sum(list(map(Q_valid_bracket,all_n_digit_bin(2*n))))

%matplotlib inline
```

```
import numpy as np
from matplotlib import pyplot as plt
x=np.arange(1,9)
f_subs = lambda t: nb_valid_in_size_n_pair(t)
vfunc = np.vectorize(f_subs)
y=vfunc(x)
print(np.polyfit(x,y,2))
plt.plot(x,y)
```

```
[  58.07738095 -373.32738095  455.875     ]
```

Out[1]: [<matplotlib.lines.Line2D at 0x20c9b5005f8>]



In [2]: x

Out[2]: array([1, 2, 3, 4, 5, 6, 7, 8])

In [3]: y

Out[3]: array([1,   2,   5,   14,   42, 132, 429, 1430])