

# Graph

July 12, 2018

```
In [1]: import numpy as np
        from matplotlib import pyplot as plt
        from time import time
        from math import factorial as fact
        from numpy import sqrt
        import commpy
        from random import shuffle
        from scipy.stats import norm
        import pandas as pd
        pi = np.pi
        tol = 1e-14
        num_trials = 100
        num_points = 500
        max_iters = 1000
```

```
In [2]: df = pd.read_csv('/Users/darren/IRES_DATA/Trial Results',index_col=0)
        df.describe()
```

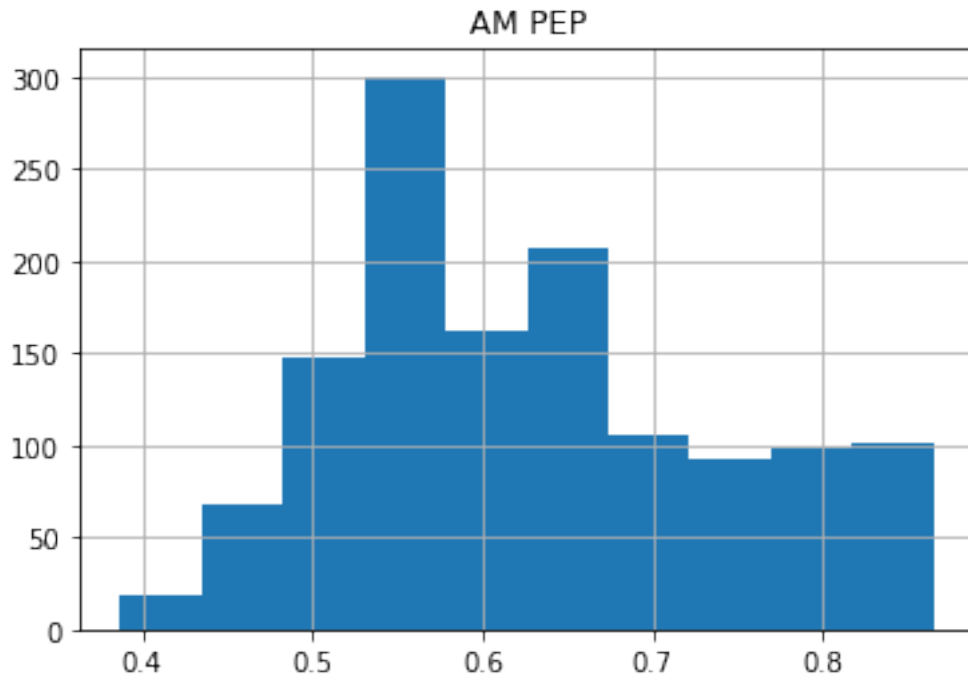
```
Out[2]:
```

	M.T.R.	Time	N.R.A.	M.N.P.R	M.W.P.	\
count	1300.000000	1300.000000	1300.0	1300.000000	1300.000000	
mean	15.644615	93.450313	1.0	91.575385	1.769231	
std	9.658520	158.276275	0.0	3.794329	1.367941	
min	5.000000	41.600934	1.0	79.000000	0.000000	
25%	10.000000	53.843167	1.0	89.000000	1.000000	
50%	13.000000	62.014103	1.0	92.000000	2.000000	
75%	17.000000	85.403123	1.0	94.000000	3.000000	
max	68.000000	3214.500514	1.0	99.000000	4.000000	

	AM PEP	PL	N.Packets
count	1300.000000	1300.000000	1300.0
mean	0.629062	19.461538	100.0
std	0.110973	7.695268	0.0
min	0.387251	11.000000	100.0
25%	0.544431	11.000000	100.0
50%	0.607168	21.000000	100.0
75%	0.691762	21.000000	100.0
max	0.864831	31.000000	100.0

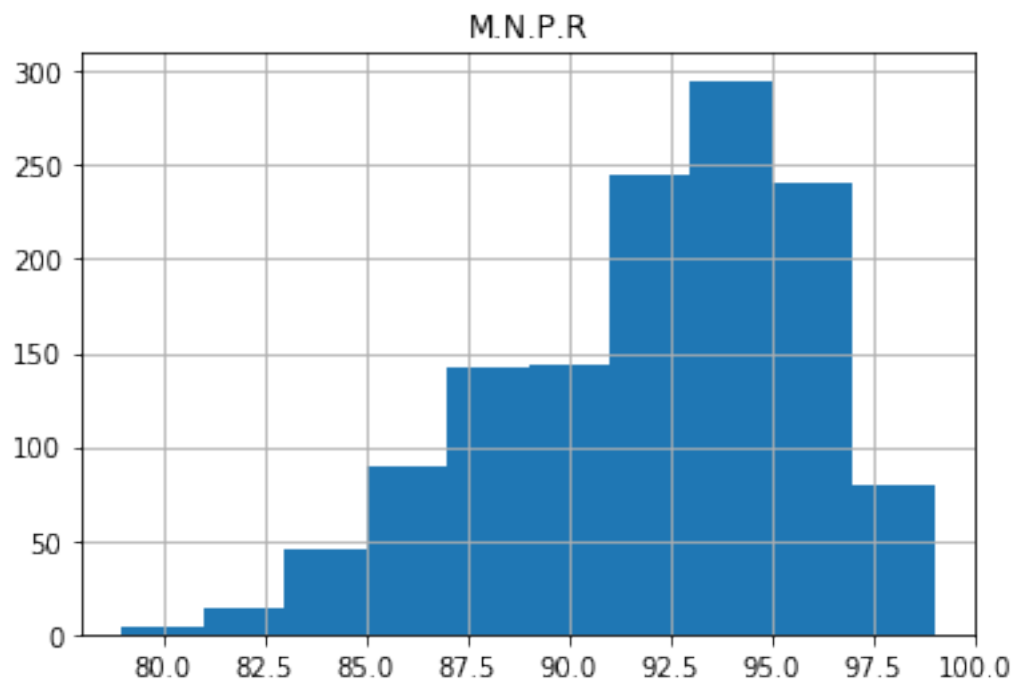
```
In [3]: df.hist(column='AM PEP')
```

```
Out[3]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x108b17198>]],  
             dtype=object)
```



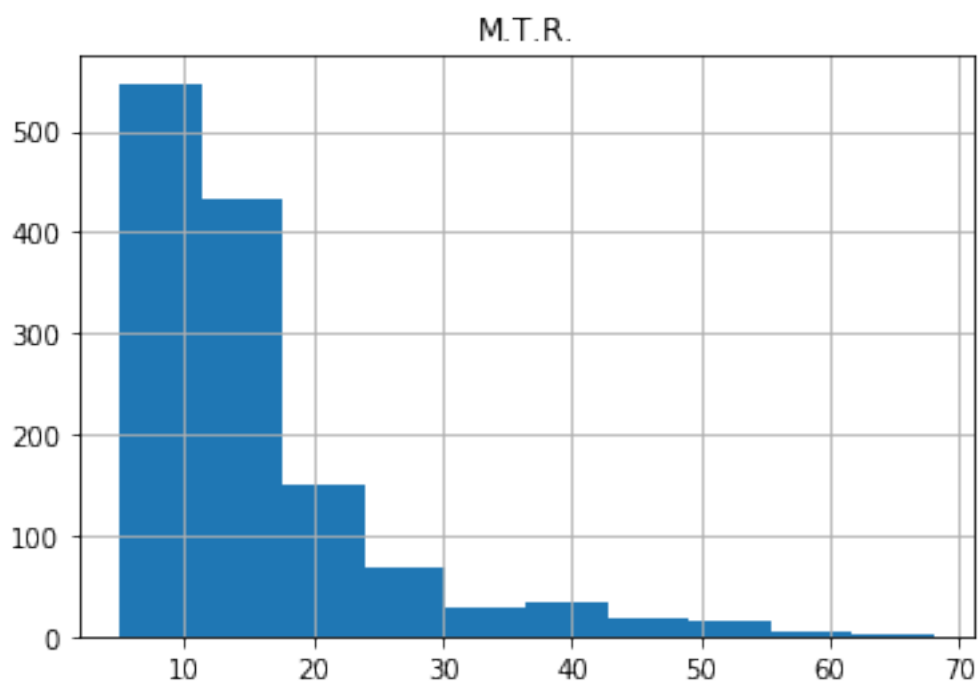
```
In [5]: df.hist(column='M.N.P.R')
```

```
Out[5]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x10b30e160>]],  
             dtype=object)
```



```
In [6]: df.hist(column='M.T.R.')
```

```
Out[6]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x10b33c940>]],  
             dtype=object)
```



```

In [24]: val1 = np.average(df.loc[df['AM PEP'] <= 0.5]['M.T.R.'])
val2 = df.loc[df['AM PEP'] <= 0.6]
val2 = np.average(val2.loc[val2['AM PEP'] > 0.5]['M.T.R.'])
val3 = df.loc[df['AM PEP'] <= 0.7]
val3 = np.average(val3.loc[val3['AM PEP'] > 0.6]['M.T.R.'])
val4 = df.loc[df['AM PEP'] <= 0.8]
val4 = np.average(val4.loc[val4['AM PEP'] > 0.7]['M.T.R.'])
val5 = df.loc[df['AM PEP'] <= 0.9]
val5 = np.average(val5.loc[val5['AM PEP'] > 0.8]['M.T.R.'])
val6 = df.loc[df['AM PEP'] <= 1.0]
val6 = np.average(val6.loc[val6['AM PEP'] > 0.9]['M.T.R.'])
values = [val1, val2, val3, val4, val5, val6]
plt.plot(range(len(values)), values)
plt.title('Max Times Requested')
plt.xlabel('Packet Error Probability (0 is <50%, 1 is 50-60%, ...)')
plt.show()

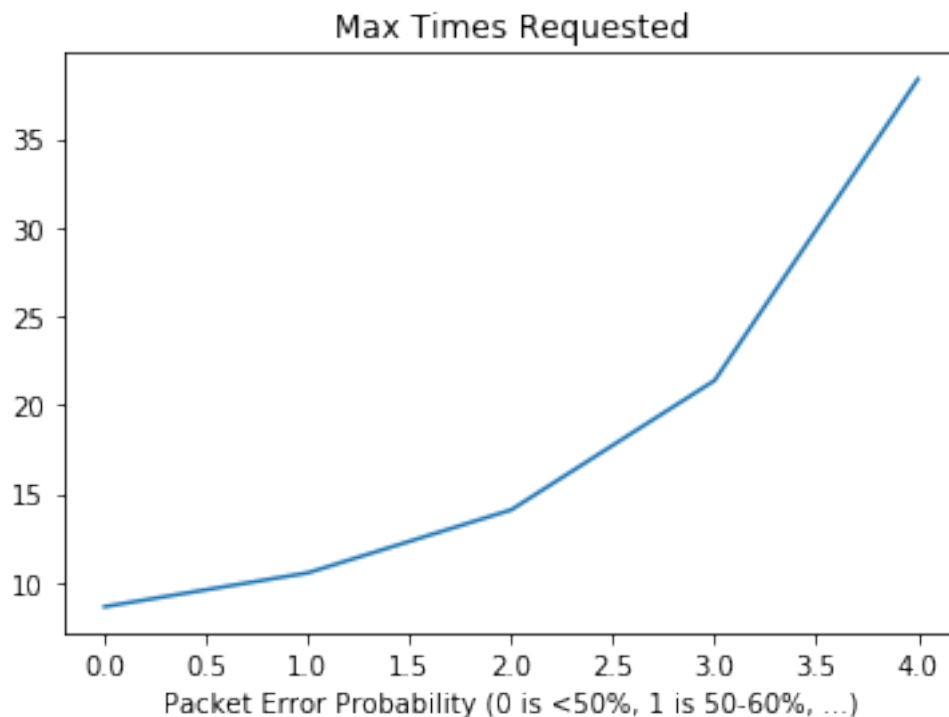
```

/anaconda3/lib/python3.6/site-packages/numpy/lib/function\_base.py:1128: RuntimeWarning: Mean of empty slice

avg = a.mean(axis)

/anaconda3/lib/python3.6/site-packages/numpy/core/\_methods.py:80: RuntimeWarning: invalid value encountered in divide

ret = ret.dtype.type(ret / rcount)



```

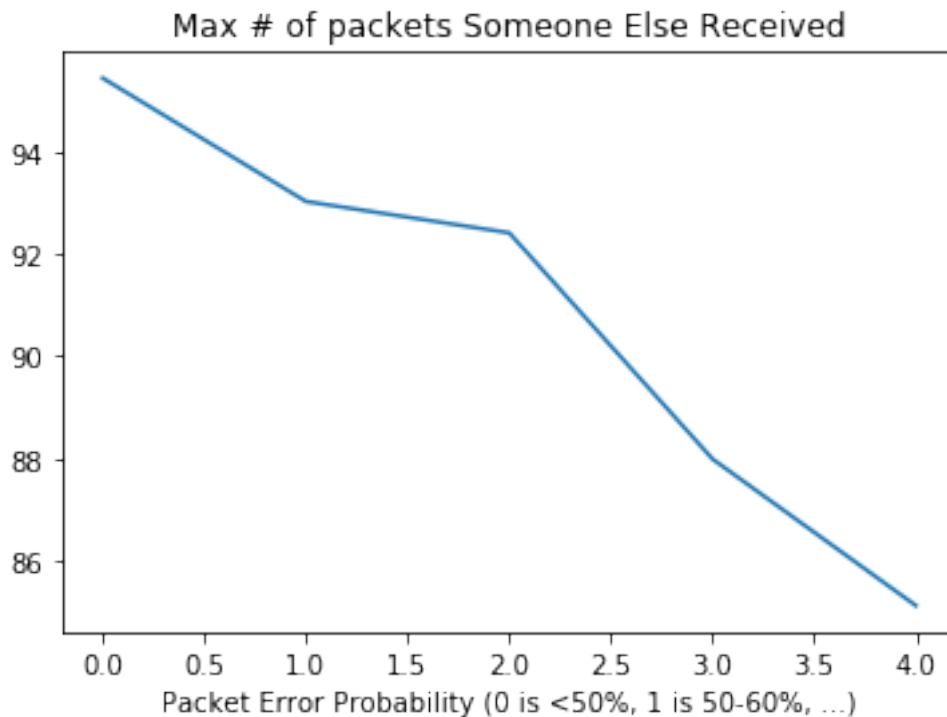
In [25]: val1 = np.average(df.loc[df['AM PEP'] <= 0.5]['M.N.P.R'])
val2 = df.loc[df['AM PEP'] <= 0.6]
val2 = np.average(val2.loc[val2['AM PEP'] > 0.5]['M.N.P.R'])
val3 = df.loc[df['AM PEP'] <= 0.7]
val3 = np.average(val3.loc[val3['AM PEP'] > 0.6]['M.N.P.R'])
val4 = df.loc[df['AM PEP'] <= 0.8]
val4 = np.average(val4.loc[val4['AM PEP'] > 0.7]['M.N.P.R'])
val5 = df.loc[df['AM PEP'] <= 0.9]
val5 = np.average(val5.loc[val5['AM PEP'] > 0.8]['M.N.P.R'])
val6 = df.loc[df['AM PEP'] <= 1.0]
val6 = np.average(val6.loc[val6['AM PEP'] > 0.9]['M.N.P.R'])
values = [val1, val2, val3, val4, val5, val6]
plt.plot(range(len(values)), values)
plt.title('Max # of packets Someone Else Received')
plt.xlabel('Packet Error Probability (0 is <50%, 1 is 50-60%, ...)')
plt.show()

```

```

/anaconda3/lib/python3.6/site-packages/numpy/lib/function_base.py:1128: RuntimeWarning: Mean of empty slice
  avg = a.mean(axis)
/anaconda3/lib/python3.6/site-packages/numpy/core/_methods.py:80: RuntimeWarning: invalid value encountered in divide
  ret = ret.dtype.type(ret / rcount)

```



```

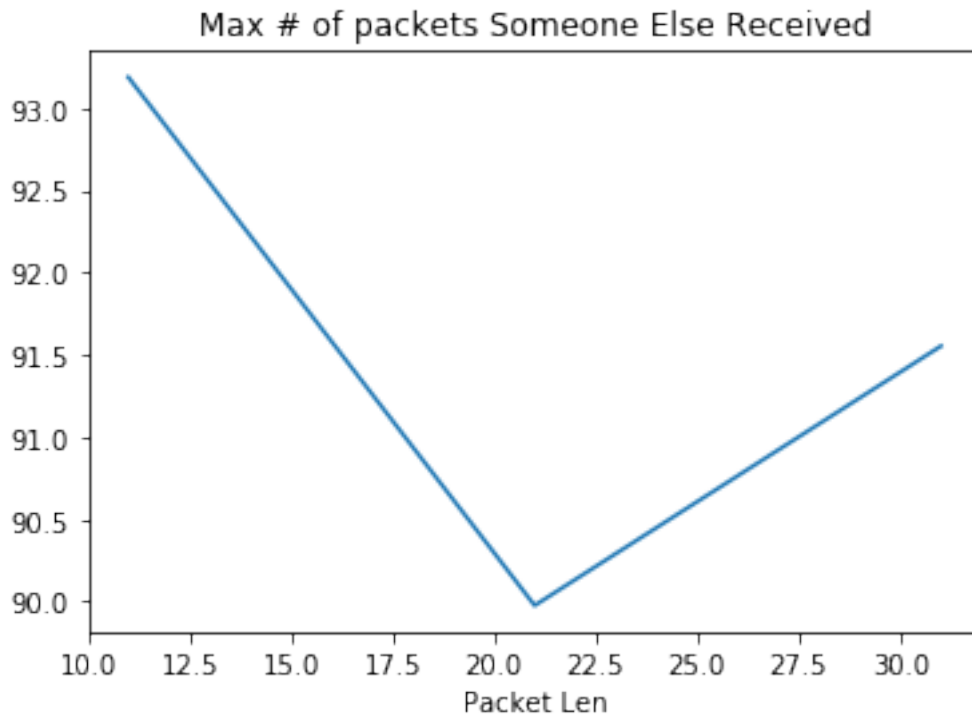
In [33]: val1 = np.average(df.loc[df['PL'] == 11]['M.N.P.R'])
val2 = np.average(df.loc[df['PL'] == 21]['M.N.P.R'])

```

```

val3 = np.average(df.loc[df['PL'] == 31]['M.N.P.R'])
values = [val1, val2, val3]
plt.plot([11,21,31],values)
plt.title('Max # of packets Someone Else Received')
plt.xlabel('Packet Len')
plt.show()

```



```

In [32]: val1 = np.average(df.loc[df['M.W.P.'] == 0]['M.N.P.R'])
val2 = np.average(df.loc[df['M.W.P.'] == 1]['M.N.P.R'])
val3 = np.average(df.loc[df['M.W.P.'] == 2]['M.N.P.R'])
val4 = np.average(df.loc[df['M.W.P.'] == 3]['M.N.P.R'])
val5 = np.average(df.loc[df['M.W.P.'] == 4]['M.N.P.R'])
values = [val1, val2, val3, val4, val5]
plt.plot(range(len(values)),values)
plt.title('Max # of packets Someone Else Received')
plt.xlabel('Max Num of Weird Patterns')
plt.show()

```

