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

1 symbol created by 14 timeslots

Jonas

We have 4 photons
14 timeslots
1 symbol created by 14 timeslots

Number of ways to order the photons

+ Code + Text

```
def ways(n):
    return math.factorial(n)
```

Number of Bits per Symbol

$$log_2 n!$$

```
def bps(W):
    return math.log2(W)
```

Number of Bits per Photon

```
def bpph(b,n):
    return b/n
```

Number of Bits per Timeslot

```
def bpt(B, n, T):
    return B*n/T
```

```
bpph(bps(24), 4)*4/14
```

```
0.3274973214800826
```

Putting all the functions together

```
def Jonasbpt(n):
  def Jonasbpph(n):
      def Jonasbps(n):
          def Jonasways(n):
              return math.factorial(n)
          return math.log2(Jonasways(n))
      return math.log2(Jonasways(n)) / n
  return print(
        #[n],
        #"Number of wyas:", ways(n),
        #"Number of Bits per Symbol:", math.log2(Jonasways(n)),
        #"Number of Bits per Photon: ", math.log2(Jonasways(n)) / n,
        #"Number of Bits per Timeslot",
        round(math.log2(Jonasways(n)) / n*(n/14), 3)
        )
```

```
Jonasbpt(4)
```

```
0.327
```

```
Jonasnumber = []
JonasPermutation = []
JonasBPS = []
JonasBPP = []
JonasBPT = []

for n in range(1,21):
    #Jonasbpt(n)
    Jonasnumber.append(n)
    JonasPermutation.append(Jonasways(n))
    JonasBPS.append(Jonasbps(n))
    JonasBPP.append(Jonasbpph(n))
    JonasBPT.append(Jonasbpt(4))

print(Jonasnumber)
print(JonasPermutation)
print(JonasBPS)
print(JonasBPP)
print(JonasBPT)
```

```
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
0.327
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
[1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, 39916800, 479001600, 6227020800, 87178291200, 1307674368000, 209227
[0.0, 1.0, 2.584962500721156, 4.584962500721156, 6.906890595608519, 9.491853096329674, 12.29920801838728, 15.29920801838
[0.0, 0.5, 0.861654166907052, 1.146240625180289, 1.3813781191217038, 1.5819755160549456, 1.7570297169124685, 1.912401002
[None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None]
```

Plot Graph

```
dictJonas = {
        'Number of Photon':
        [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20],
        'Number of Permutation':
        [1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, 39916800, 479001600, 6227020800, 87178291200, 130767
        'Number of Bits per Symbol':
        [0.0, 1.0, 2.584962500721156, 4.584962500721156, 6.906890595608519, 9.491853096329674, 12.29920801838728,
        'Number of Bits per Photon ':
        [0.0, 0.5, 0.861654166907052, 1.146240625180289, 1.3813781191217038, 1.5819755160549456, 1.757029716912468
        'Number of Bits per Time Slots':
        [0.0, 0.327, 0.071, 0.327, 0.185, 0.327, 0.327, 0.327, 0.493, 0.327, 0.678, 0.327, 0.879, 0.327, 1.093, 0.
        }
```

```
dfJonas[dfJonas.columns[0]]
```

```
0      1
1      2
2      3
3      4
4      5
5      6
6      7
7      8
8      9
9     10
10    11
11    12
```

```
12    13
13    14
14    15
15    16
16    17
17    18
18    19
19    20
Name: Number of Photon, dtype: int64
```

```
1 dfJonas[dfJonas.columns[1]]
```

```
0                        1
1                        2
2                        6
3                       24
4                      120
5                      720
6                     5040
7                    40320
8                   362880
9                  3628800
10                39916800
11               479001600
12              6227020800
13             87178291200
14           1307674368000
15          20922789888000
16         355687428096000
17        6402373705728000
18      121645100408832000
19     2432902008176640000
Name: Number of Permutation, dtype: int64
```

```
1 dfJonas[dfJonas.columns[2]]
```

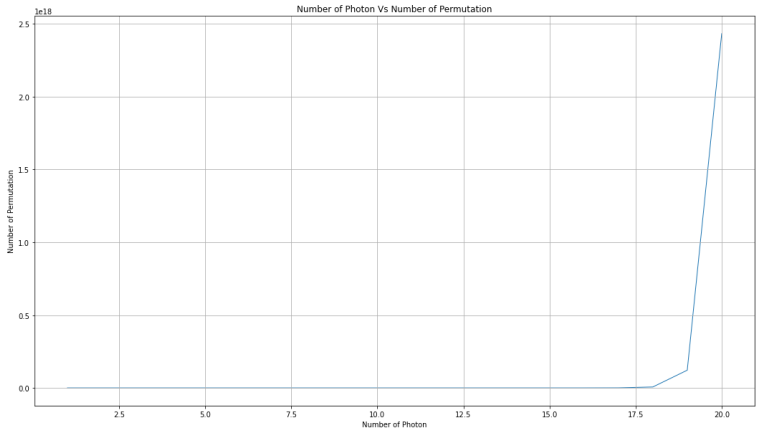
```
0      0.000000
1      1.000000
2      2.584963
3      4.584963
4      6.906891
5      9.491853
6     12.299208
7     15.299208
8     18.469133
9     21.791061
10    25.250493
11    28.835455
12    32.535895
13    36.343250
14    40.250140
15    44.250140
16    48.337603
17    52.507528
18    56.755456
19    61.077384
Name: Number of Bits per Symbol, dtype: float64
```

```
1 dfJonas[dfJonas.columns[3]]
```

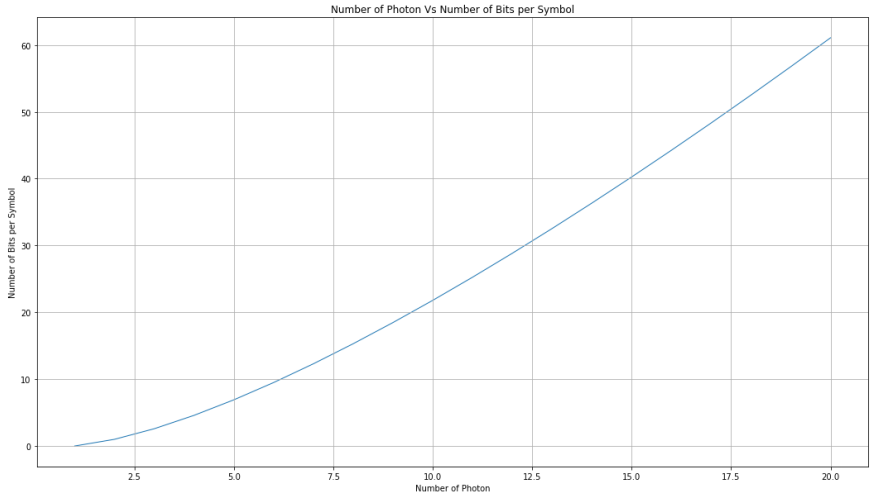
```
0     0.000000
1     0.500000
2     0.861654
3     1.146241
4     1.381378
5     1.581976
6     1.757030
7     1.912401
8     2.052126
9     2.179106
10    2.295499
11    2.402955
12    2.502761
13    2.595946
14    2.683343
15    2.765634
16    2.843388
17    2.917085
18    2.987129
19    3.053869
Name: Number of Bits per Photon , dtype: float64
```

```
1 dfJonas = pd.DataFrame(dictJonas)
```

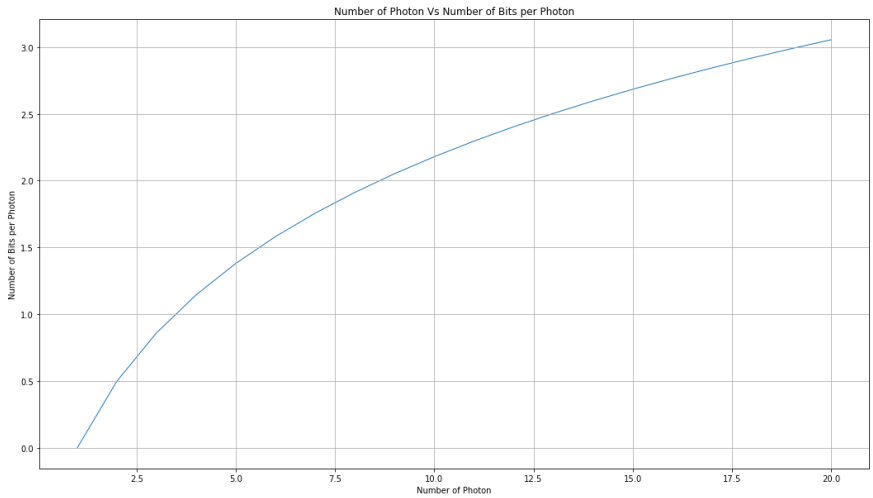
```
figure, ax1 = plt.subplots(figsize=(18,10))


#ax = df1.plot(, xticks=range(0, 61), title = 'Detection Rate in Data 1')
ax1.plot(dfJonas[dfJonas.columns[0]],dfJOnas[dfJonas.columns[1]],linewidth=1,zorder=1, label = "bits")


ax1.set_title('Number of Photon Vs Number of Permutation', fontsize = 12)

ax1.set_xlabel('Number of Photon')
ax1.set_ylabel('Number of Permutation')

#ax1.set_ylim([6000,7000])
#ax2.set_ylim([6000,7000])

ax1.grid(True)

#figure.set_facecolor("white")
#plt.savefig('channel2data1.png', dpi=300, bbox_inches='tight')
```




```
figure, ax1 = plt.subplots(figsize=(18,10))


#ax = df1.plot(, xticks=range(0, 61), title = 'Detection Rate in Data 1')
ax1.plot(dfJonas[dfJonas.columns[0]],dfJOnas[dfJonas.columns[2]],linewidth=1,zorder=1, label = "bits")


ax1.set_title('Number of Photon Vs Number of Bits per Symbol', fontsize = 12)

ax1.set_xlabel('Number of Photon')
ax1.set_ylabel('Number of Bits per Symbol')

#ax1.set_ylim([6000,7000])
#ax2.set_ylim([6000,7000])

ax1.grid(True)

#figure.set_facecolor("white")
#plt.savefig('channel2data1.png', dpi=300, bbox_inches='tight')
```

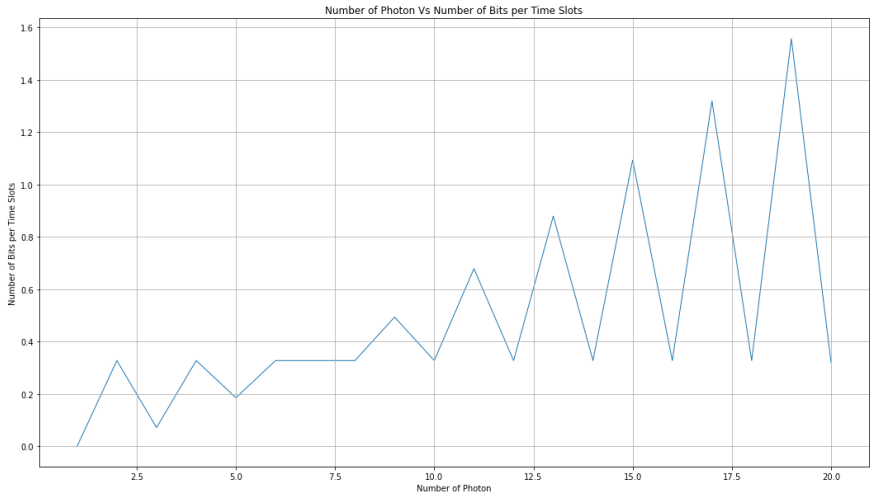



```
figure, ax1 = plt.subplots(figsize=(18,10))


#ax = df1.plot(, xticks=range(0, 61), title = 'Detection Rate in Data 1')
ax1.plot(dfJonas[dfJonas.columns[0]],dfJOnas[dfJonas.columns[3]],linewidth=1,zorder=1, label = "bits")


ax1.set_title('Number of Photon Vs Number of Bits per Photon', fontsize = 12)

ax1.set_xlabel('Number of Photon')
ax1.set_ylabel('Number of Bits per Photon')

#ax1.set_ylim([6000,7000])
#ax2.set_ylim([6000,7000])

ax1.grid(True)

#figure.set_facecolor("white")
#plt.savefig('channel2data1.png', dpi=300, bbox_inches='tight')
```




```
figure, ax1 = plt.subplots(figsize=(18,10))

```

```
#ax = df1.plot(, xticks=range(0, 61), title = 'Detection Rate in Data 1')
ax1.plot(dfJonas[dfJonas.columns[0]],dfJOnas[dfJonas.columns[4]],linewidth=1,zorder=1, label = "bits")


ax1.set_title('Number of Photon Vs Number of Bits per Time Slots', fontsize = 12)

ax1.set_xlabel('Number of Photon')
ax1.set_ylabel('Number of Bits per Time Slots')

#ax1.set_ylim([6000,7000])
#ax2.set_ylim([6000,7000])

ax1.grid(True)

#figure.set_facecolor("white")
#plt.savefig('channel2data1.png', dpi=300, bbox_inches='tight')
```




```
1
```

General

We have 4 photons 14 timeslots 1,0001 ways to order them

```
2**14
```

```
16384
```

PPM

We have 1 photon
14 timeslots
14 ways to order them

```
def PPMbpt(T, n):
    def PPMbpph(n):
        def PPMbps(n):
            def PPMways(T):
                return T
            return math.log2(T)
        return math.log2(T) / n
```

```
    return print([n], "Number of wyas:", ways(n),
                 "Number of Bits per Symbol:", math.log2(ways(n)),
                 "Number of Bits per Photon: ", math.log2(ways(n)) / n,
                 "Number of Bits per Timeslot", math.log2(ways(n)) / n*(n/T)
                )
```

```
for n in range(1,21):
    PPMbpt(n, 14)
```

```
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
[14] Number of wyas: 87178291200 Number of Bits per Symbol: 36.3432498742741 Number of Bits per Photon:  2.5959464195910(
```

OOK

```
def OOKbpt(T, n):
  def OOKbpph(n, T):
    def OOKbps(T):
      def OOKways(T):
        return (2)**T
      return math.log2((2)**T)
    return math.log2((2)**T) / n
  return print(
                 [n],
                 "Number of Permutation:", OOKways(T),
                 "Number of Bits per Symbol:", math.log2(OOKways(T)),
                 "Number of Bits per Photon: ", math.log2(OOKways(T)) / n,
                 "Number of Bits per Timeslot", math.log2(OOKways(T)) / n*(n/T)
                 )
```

```
OOKbpt(14, 7)
```

```
[7] Number of Permutation: 16384 Number of Bits per Symbol: 14.0 Number of Bits per Photon:  2.0 Number of Bits per Times
```

```
OOKnumber = []

OOKPermutation = []

OOKBPS = []

OOKBPP = []

T = 14
for n in range(1,21):
    OOKnumber.append(n)
    OOKPermutation.append(OOKways(T))
    OOKBPS.append(OOKbps(T))
```

```
print(OOKnumber)
```

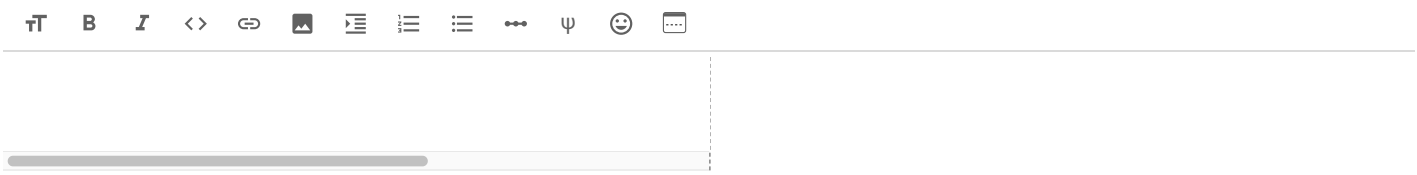
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

```
print(OOKways(T))
```

```
16384
```

```
print(OOKBPS)
```

```
[14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0, 14.0]
```



✓ 0s completed at 16:10