ECE 462 - Homework #6

Collin Heist

March 5, 2020

1 Code

1.1 System Includes

```
In [1]: import pandas as pd
        import numpy as np
        from itertools import permutations
        import matplotlib.pyplot as plt
        from pylab import rcParams
```

1.2 Function to find all possible combinations for a given number of balls and energy total

```
In [2]: def combs(num_balls, required_sum, unique=False):
            valids = []
            if unique == True:
                possibilities = np.arange(num_balls + 1)
            elif unique == 'degenerate':
                possibilities = np.array([[val] * 2 for val in range(num_balls+1) if val <= requ
            else:
                possibilities = np.array([[val] * num_balls for val in range(num_balls+1) if val
            perms = list(permutations(possibilities, num_balls))
            valids = [np.sort(tupe) for tupe in perms if sum(tupe) == required_sum]
            if (len(valids) == 0):
                return pd.DataFrame(columns=["Ball {}".format(i+1) for i in range(num_balls)])

            return pd.DataFrame(np.unique(np.array(valids), axis=0), columns=["Ball {}".format(i
```

1.3 Function to find the probability distribution of a given combination set

```
In [3]: def prob_distribution(df):
            return (1.0 - (df.apply(pd.Series.value_counts, axis=1).isna().sum(axis='rows') / le
```

1.4 Function to plot a probability distribution

```
In [4]: def plot_probs(df):
            rcParams['figure.figsize'] = 14, 8
            display(prob_distribution(df))
```

```
plt.xlabel("Energy Level")
plt.ylabel("Probability (%)")
plt.ylim(0, 105)
plt.grid(True, axis='y', linewidth=0.5)
plt.plot(prob_distribution(df).index.values, prob_distribution(df).values, linewidth
```

2 Problems

2.1 Problem 5.2.1

With $E = 4$, and $n_{balls} = 4$, the following combinations are possible:

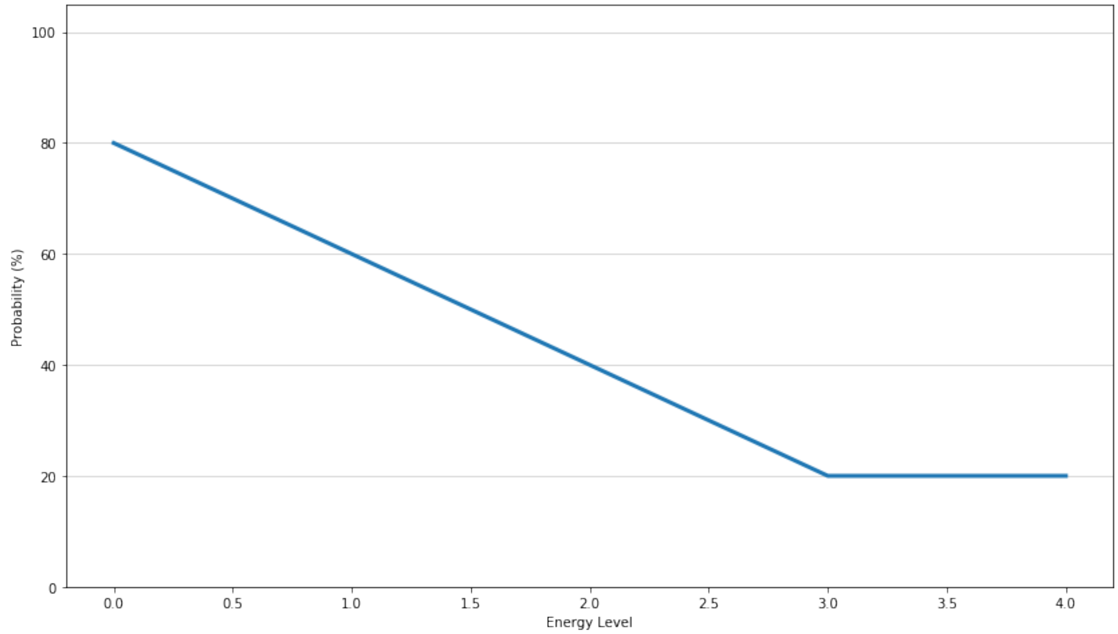
```
In [5]: df = combs(4, 4, unique=False)
        display(df)
```

```
   Ball 1  Ball 2  Ball 3  Ball 4
0       0       0       0       4
1       0       0       1       3
2       0       0       2       2
3       0       1       1       2
4       1       1       1       1
```

With the distribution of these being (with the probability being the chance a given state is occupied):

```
In [6]: plot_probs(df)
```

```
0    80.0
1    60.0
2    40.0
3    20.0
4    20.0
dtype: float64
```



2.2 Problem 5.2.2

With five balls and a total energy of 12, the only valid distribution is:

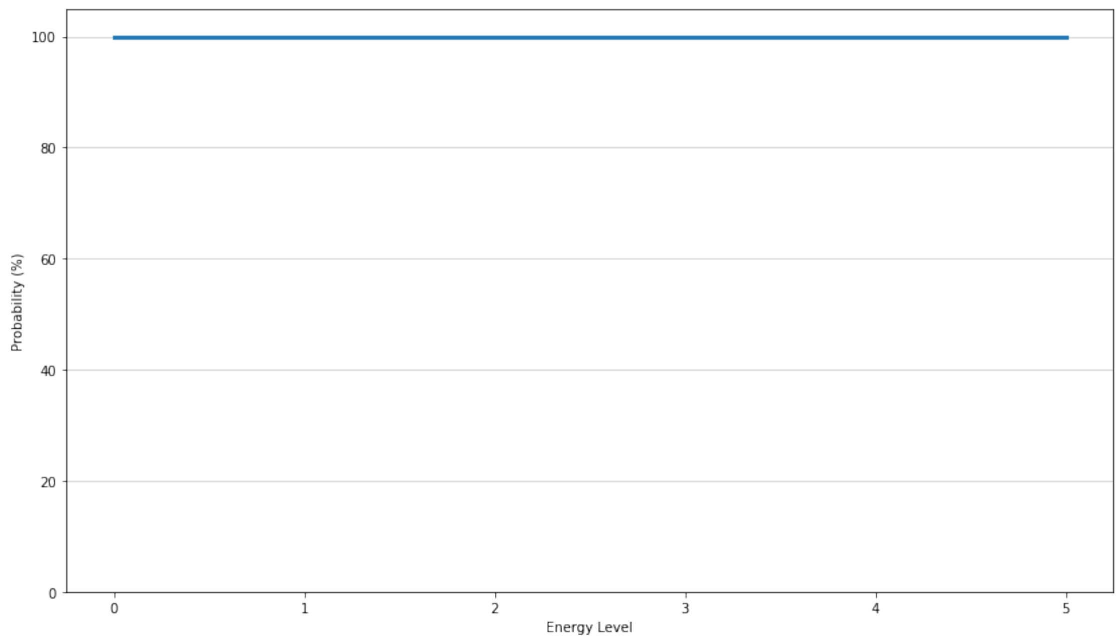
```
In [7]: df = combs(5, 12, unique=True)
        display(df)
```

	Ball 1	Ball 2	Ball 3	Ball 4	Ball 5
0	0	1	2	4	5

The distribution of this is (with the probability being the chance a given state is occupied):

```
In [8]: plot_probs(df)
```

```
5    100.0
4    100.0
2    100.0
1    100.0
0    100.0
dtype: float64
```



2.3 Problem 5.2.3

Below are the following valid states with five fermion particles totalling a combined energy of 8.

```
In [9]: df = combs(5, 8, 'degenerate')
        display(df)
```

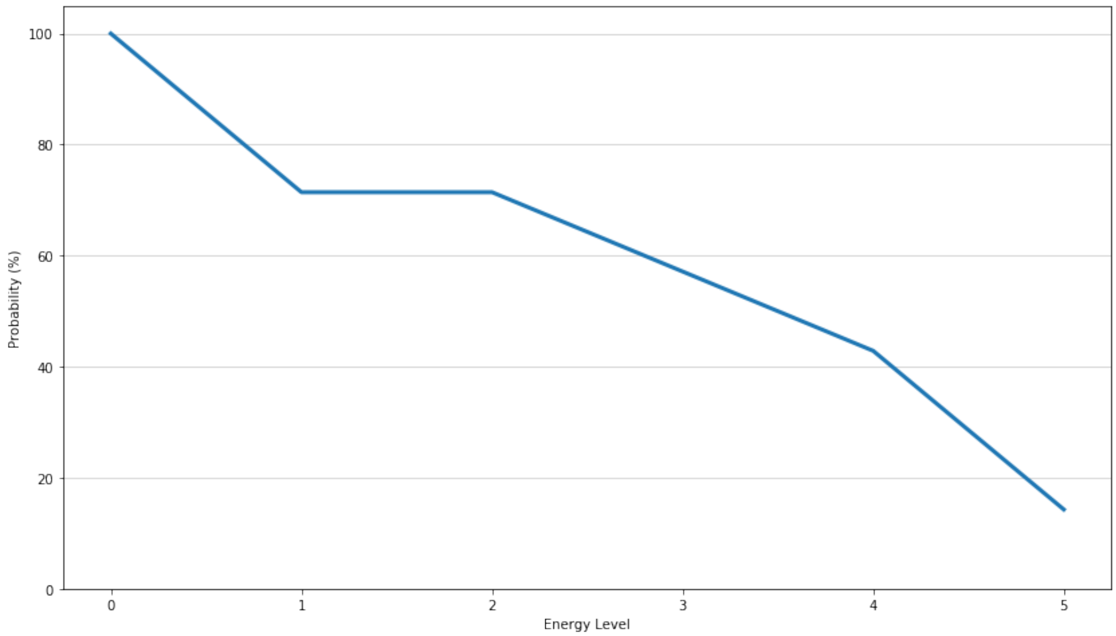
```
   Ball 1  Ball 2  Ball 3  Ball 4  Ball 5
0       0       0       1       2       5
1       0       0       1       3       4
2       0       0       2       2       4
3       0       0       2       3       3
4       0       1       1       2       4
5       0       1       1       3       3
6       0       1       2       2       3
```

The probability of occupation of each state is (with the probability being the chance a given state is occupied):

```
In [10]: plot_probs(df)
```

```
0    100.000000
1     71.428571
2     71.428571
3     57.142857
4     42.857143
```

```
5      14.285714
dtype: float64
```



2.4 Problem 5.2.4

The Fermi-Dirac distribution is:

$$f_F = \frac{1}{1 + e^{\frac{E-E_F}{k_B T}}}$$

The approximation is valid only if:

$$e^{\frac{E-E_f}{k_B T}} >> 1$$

This shows that the *Boltzman Approximation* is only valid if $E - E_f >> k_B T$.