

# Percolation

Noah Green

October 19, 2017

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

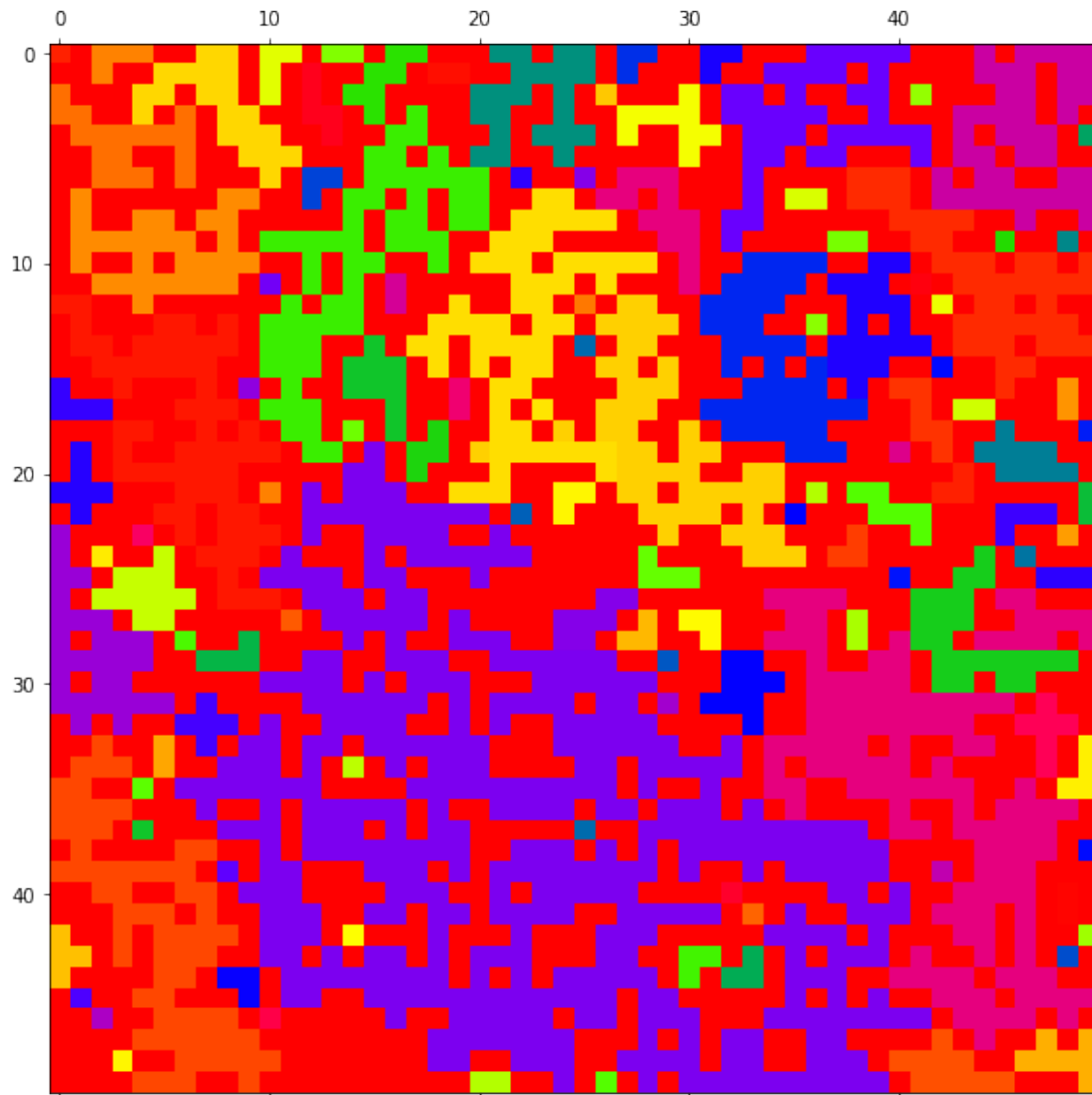
## 1 Part 1

Below, I take the grid from my percolation code and relabel the clusters so that the cluster labels only go up to the number of clusters. This reduces the size of the color scale needed to represent all clusters.

```
In [2]: grid = pd.read_csv("Grid.tsv", sep='\s+', header=None).as_matrix()
```

```
relabel = 0
labels = dict()
i = 0
j = 0
for i in range(grid.shape[0]):
    for j in range(grid.shape[1]):
        if grid[i][j] != 0:
            if grid[i][j] not in labels:
                relabel += 1
                labels[grid[i][j]] = relabel
                grid[i][j] = relabel
            else:
                grid[i][j] = labels[grid[i][j]]
```

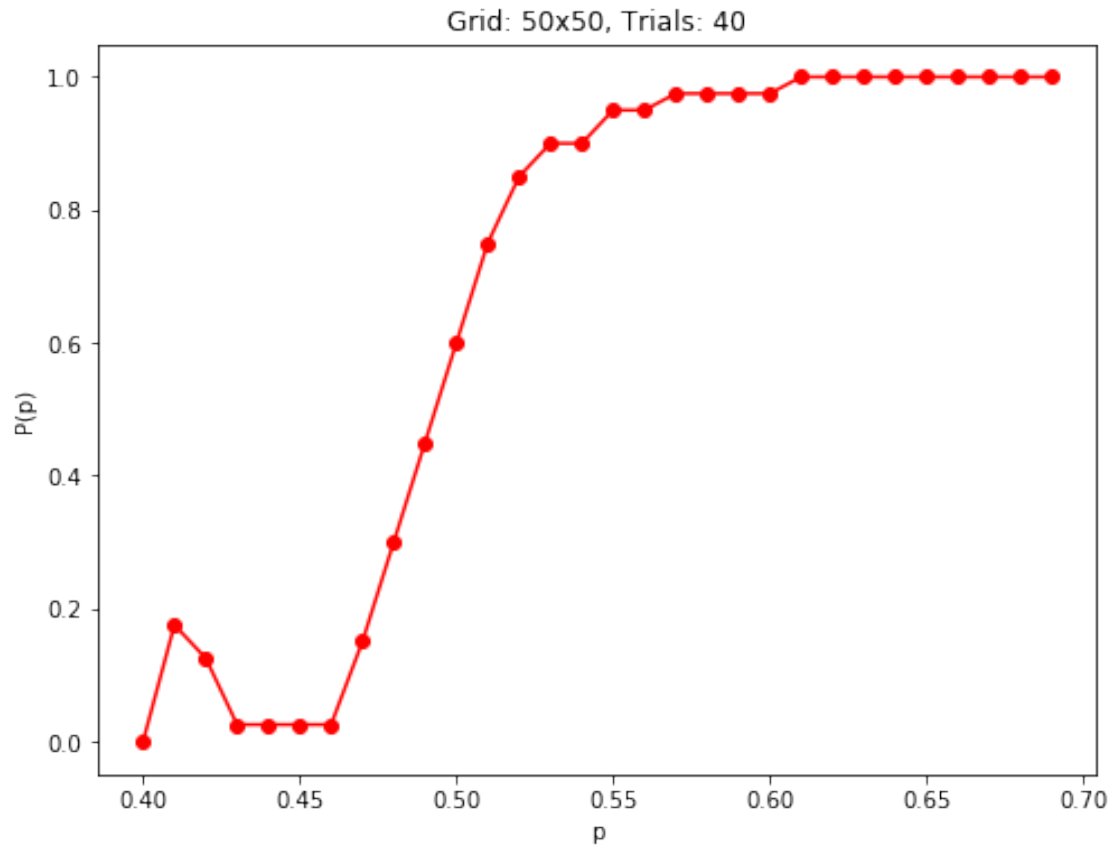
```
In [3]: fig, axes = plt.subplots(figsize = (10,10))
axes.matshow(grid, cmap="prism")
plt.show()
```



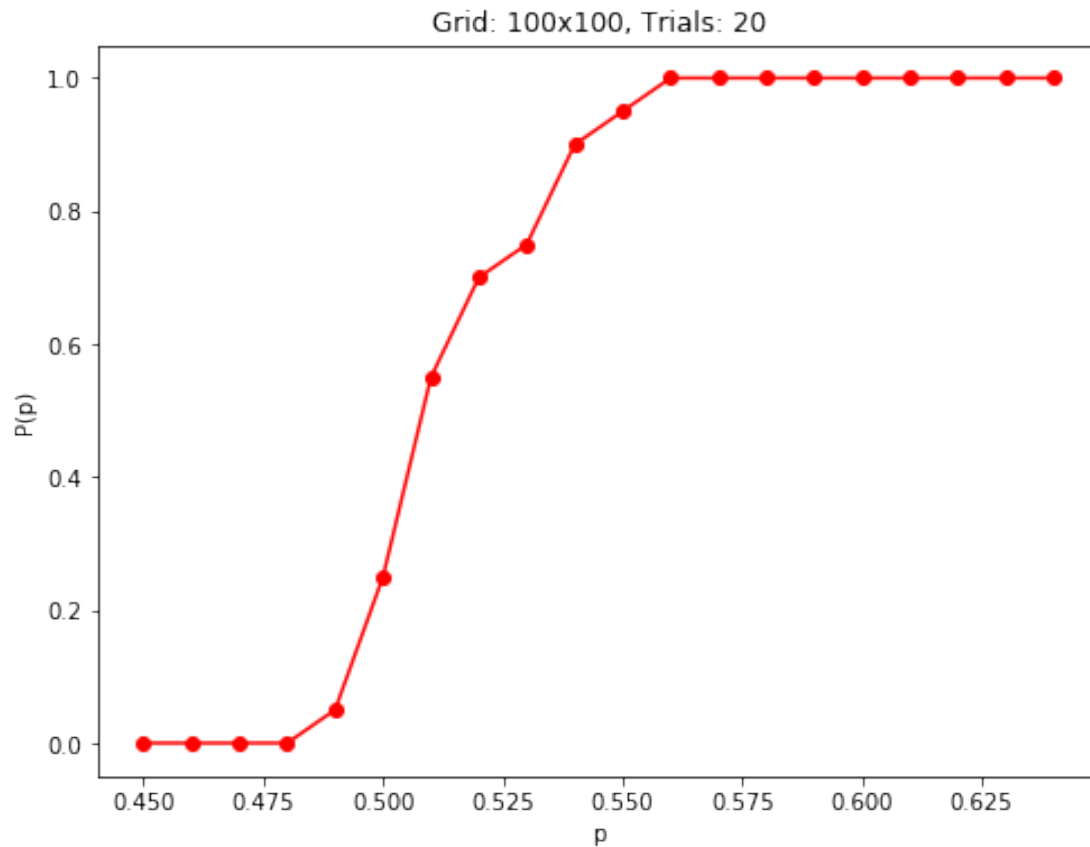
## 2 Part 2

We can see below that the probability of having a percolating cluster rapidly increases when  $p \approx 0.58$ . Note that the increase becomes more rapid for larger grids.

```
In [4]: Px = pd.read_csv("Px50.tsv",sep="\s+",header=None).as_matrix()[0]
        Py = pd.read_csv("Py50.tsv",sep="\s+",header=None).as_matrix()[0]
        fg,ax = plt.subplots(figsize=(8,6))
        ax.plot(Px,Py,'ro-')
        ax.set_xlabel("p")
        ax.set_ylabel("P(p)")
        ax.set_title("Grid: 50x50, Trials: 40")
        plt.show()
```



```
In [5]: Px = pd.read_csv("Px100.tsv",sep="\s+",header=None).as_matrix()[0]
Py = pd.read_csv("Py100.tsv",sep="\s+",header=None).as_matrix()[0]
fg,ax = plt.subplots(figsize=(8,6))
ax.plot(Px,Py, 'ro-')
ax.set_xlabel("p")
ax.set_ylabel("P(p)")
ax.set_title("Grid: 100x100, Trials: 20")
plt.show()
```



```
In [6]: Px = pd.read_csv("Px200.tsv",sep="\s+",header=None).as_matrix()[0]
Py = pd.read_csv("Py200.tsv",sep="\s+",header=None).as_matrix()[0]
fg,ax = plt.subplots(figsize=(8,6))
ax.plot(Px,Py, 'ro-')
ax.set_xlabel("p")
ax.set_ylabel("P(p)")
ax.set_title("Grid: 200x200, Trials: 10")
plt.show()
```

