1 Random Variables

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [ ]:

```
n = np.random.randint(2, 10, 40)
print(n)
```

```
[8 7 3 6 3 3 9 6 9 9 8 6 6 9 2 8 5 7 6 6 7 3 3 2 3 9 6 7 3 7 4 4 7 3 9 3 8
 7 4 4]
```

1.1 Probability Mass Function

In [ ]:

```
#Convert list n to dataframe
df = pd.DataFrame(n)

#Count each variable how many times repeated
df = pd.DataFrame(df[0].value_counts())
df
```

Out[20]:

	0
3	9
7	7
6	7
9	6
8	4
4	4
2	2
5	1

In [ ]:

```
length = len(n)
length
```

Out[21]:

```
40
```

In [ ]:

```
df.columns = ["Counts"]
df
```

Out[22]:

	Counts
3	9
7	7
6	7
9	6
8	4
4	4
2	2
5	1

In [ ]:

```
df["Prob"] = df["Counts"] / length
df
```

Out[23]:

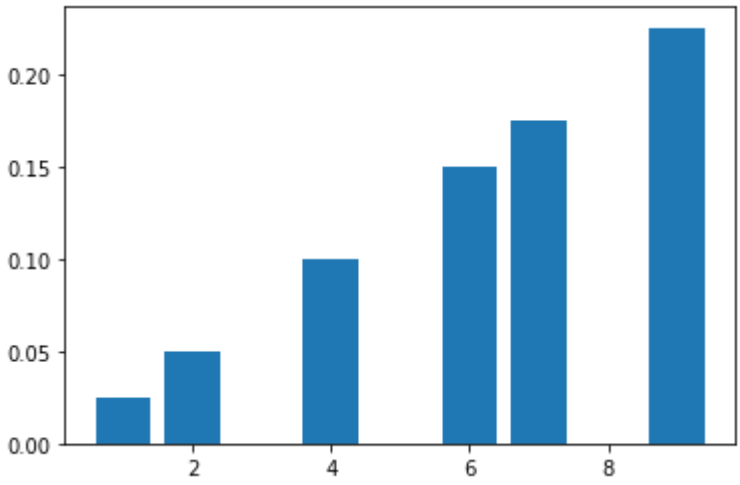
	Counts	Prob
3	9	0.225
7	7	0.175
6	7	0.175
9	6	0.150
8	4	0.100
4	4	0.100
2	2	0.050
5	1	0.025

In [ ]:

```
# Plot pmf
plt.bar(df["Counts"], df["Prob"])
```

Out[24]:

```
<BarContainer object of 8 artists>
```



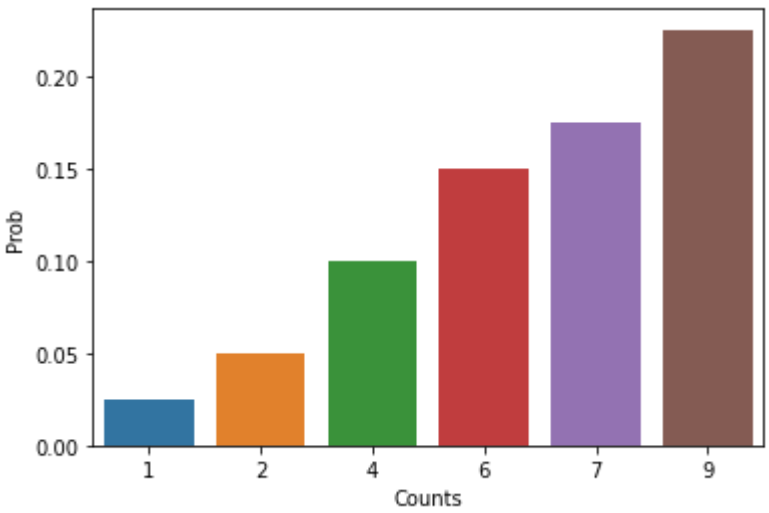
In [ ]:

```
sns.barplot(df["Counts"], df["Prob"])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarn
ing: Pass the following variables as keyword args: x, y. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc527a53e90>
```



In [ ]:

```
data = {"Candy" : ["Blue", "Orange", "Green", "Purple"],
        "Total" : [30000, 10000, 20000, 12000]}

df = pd.DataFrame(data)
df
```

Out[28]:

	Candy	Total
0	Blue	30000
1	Orange	10000
2	Green	20000
3	Purple	12000

In [ ]:

```
df["pmf"] = df["Total"]/df["Total"].sum()
df
```

Out[30]:

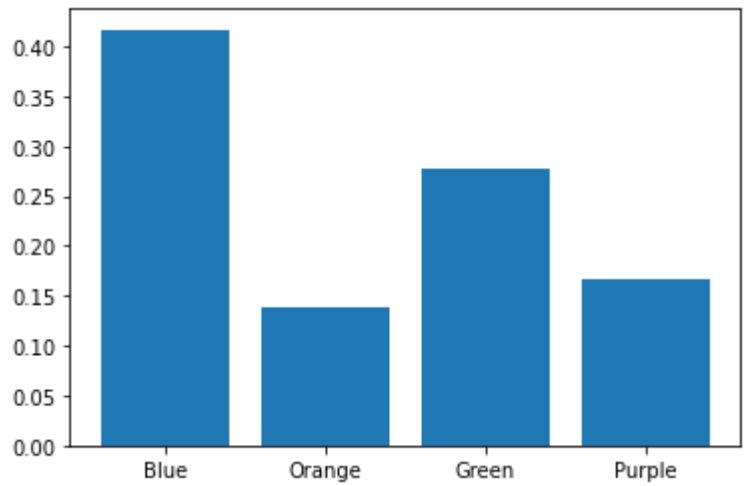
	Candy	Total	pmf
0	Blue	30000	0.416667
1	Orange	10000	0.138889
2	Green	20000	0.277778
3	Purple	12000	0.166667

In [ ]:

```
plt.bar(df["Candy"], df["pmf"])
```

Out[32]:

```
<BarContainer object of 4 artists>
```



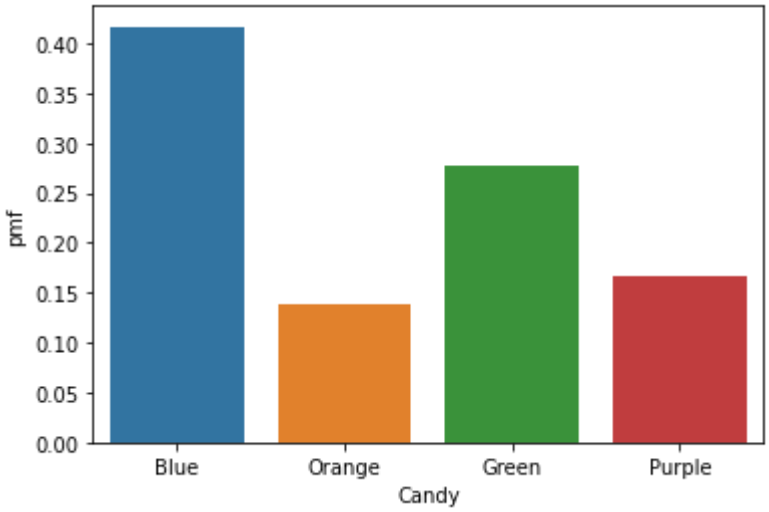
In [ ]:

```
#Plot pmf using seaborne
sns.barplot(df["Candy"], df["pmf"])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarn
ing: Pass the following variables as keyword args: x, y. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc525618fd0>
```



1.2 Probability Density Function

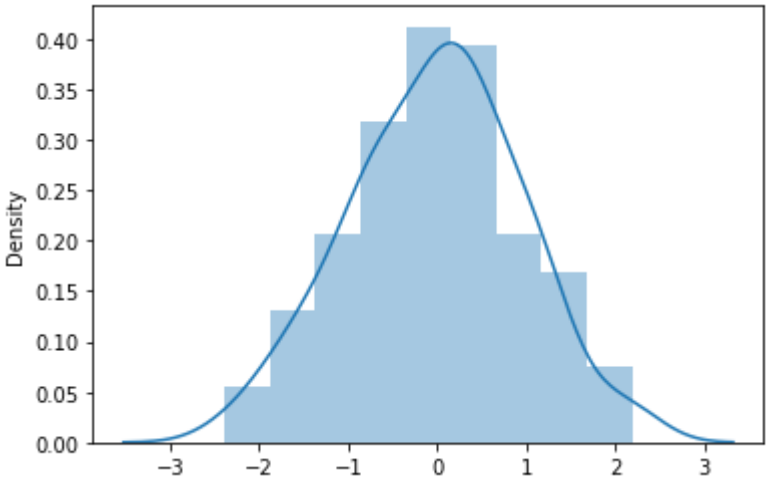
In [ ]:

```
data = np.random.normal(size = 100)
data = np.append(data, [1.2, 1.2, 1.2, 1.2, 1.2])
sns.distplot(data)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: Future
Warning: `distplot` is a deprecated function and will be removed in a future
version. Please adapt your code to use either `displot` (a figure-level func
tion with similar flexibility) or `histplot` (an axes-level function for his
tograms).
  warnings.warn(msg, FutureWarning)
```

Out[33]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc52539f750>
```



In [ ]:

```
dir(sns)
```

In [ ]:

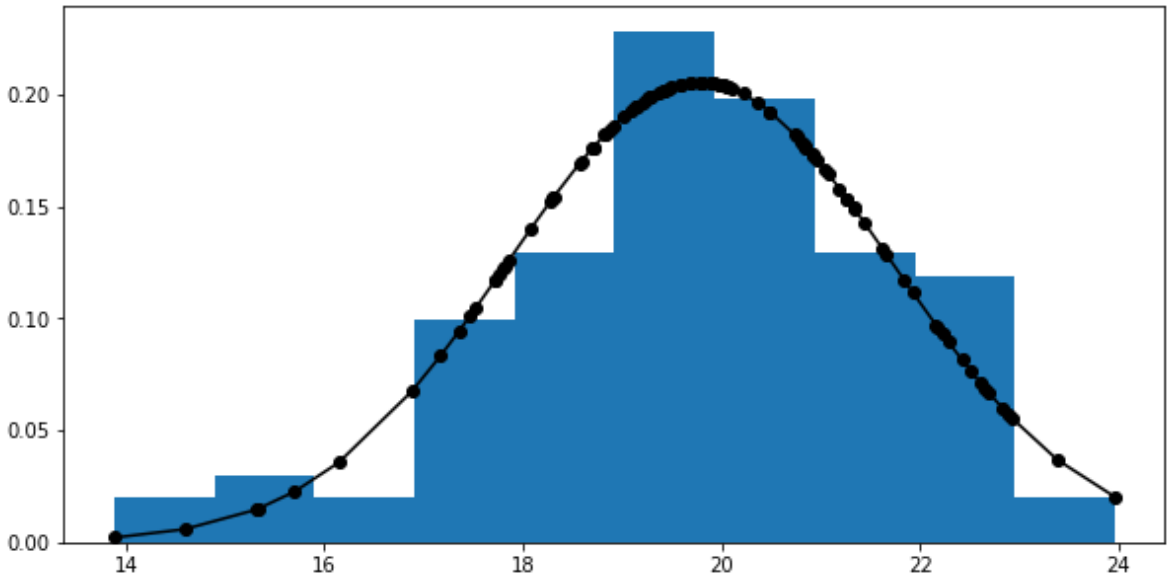
```
import scipy.stats as stats
mu = 20
sigma = 2

h = sorted(np.random.normal(mu, sigma, 100))
```

In [ ]:

```
from scipy.interpolate import fitpack2
plt.figure(figsize = (10, 5))

fit = stats.norm.pdf(h, np.mean(h), np.std(h))

plt.plot(h, fit, "-o", color = "Black")

plt.hist(h, density = True)
```

Out[49]:

```
(array([0.01983939, 0.02975908, 0.01983939, 0.09919693, 0.12895601,
        0.22815294, 0.19839386, 0.12895601, 0.11903632, 0.01983939]),
 array([13.88008162, 14.88817731, 15.89627301, 16.9043687 , 17.9124644 ,
        18.92056009, 19.92865579, 20.93675148, 21.94484718, 22.95294288,
        23.96103857]),
 <a list of 10 Patch objects>)
```



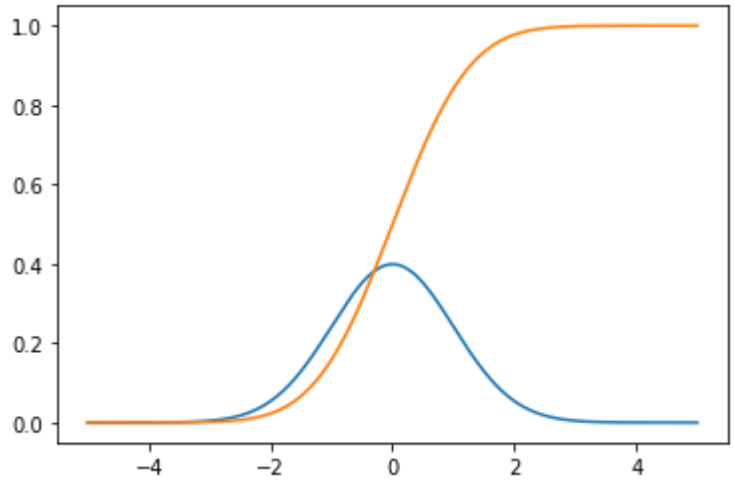
1.3 Cumulative Distribution Function

In [ ]:

```
import scipy.stats as ss

x  = np.linspace(-5, 5, 5000)
mu = 0
sigma = 1

y_pdf = ss.norm.pdf(x, mu, sigma)
y_cdf = ss.norm.cdf(x, mu, sigma)

plt.plot(x, y_pdf, label = "pdf")
plt.plot(x, y_cdf, label = "cdf")
```

Out[51]:

```
[<matplotlib.lines.Line2D at 0x7fc524a9ad50>]
```



In [ ]:

```
plt.figure(figsize = (10, 5))

fit = stats.norm.cdf(h, np.mean(h), np.std(h))

plt.plot(h, fit, "-o", color = "Black")

plt.hist(h, density = True)
```

Out[54]:

```
(array([0.01983939, 0.02975908, 0.01983939, 0.09919693, 0.12895601,
        0.22815294, 0.19839386, 0.12895601, 0.11903632, 0.01983939]),
 array([13.88008162, 14.88817731, 15.89627301, 16.9043687 , 17.9124644 ,
        18.92056009, 19.92865579, 20.93675148, 21.94484718, 22.95294288,
        23.96103857]),
 <a list of 10 Patch objects>)
```