Notebook

March 19, 2019

Local date & time is : 03/19/2019 00:55:11 PDT

In [6]:
```
#Answer to 1a

# Construct the distribution of X
dist_X = Table().values(np.arange(6)).probability(make_array(0.45, 0.25, 0, 0.2, 0.05, 0.05))

# Extract the array of probabilities
probs_X = dist_X[1]

# Get the coefficients of the pgf in the appropriate order
coeffs_X = np.flipud(probs_X)

# Construct the pgf
pgf_X =  np.poly1d(coeffs_X)

# Display the pgf
print(pgf_X)
```

```
     5        4       3
0.05 x + 0.05 x + 0.2 x + 0.25 x + 0.45
```

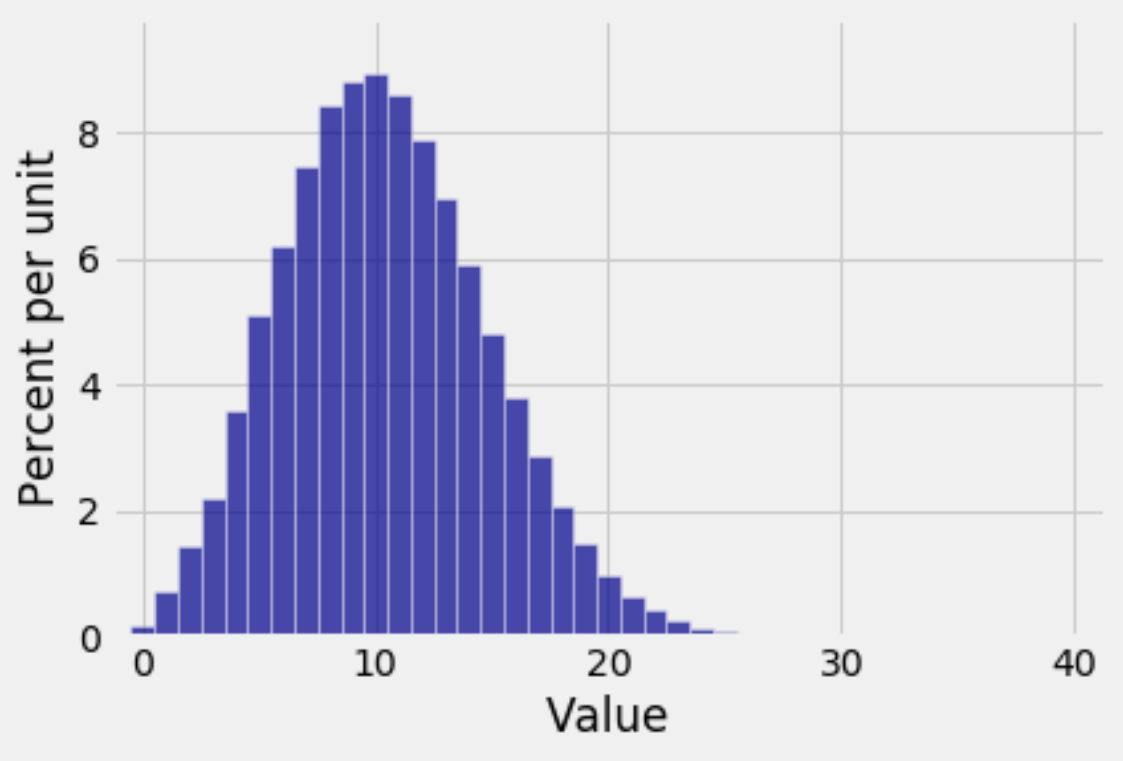
In [7]:
```
#Answer to 1b

pgf_SX = pgf_X**8 # pgf of S_X
coeffs_SX = pgf_SX.c # coefficients of pgf of S_X

# Distribution object for S_X
# Careful ...
# Think how you will extract the possible values and corresponding chances.
# Use extra lines if you need them.

probs_SX = np.flipud(coeffs_SX)
dist_SX = Table().values(np.arange(41)).probability(probs_SX)

Plot(dist_SX)
```




```
In [9]: #Answer to 1c
        probs_SX[13]
```

```
Out[9]: 0.06964776875000002
```

```
In [10]: np.arange(4, 9)
```

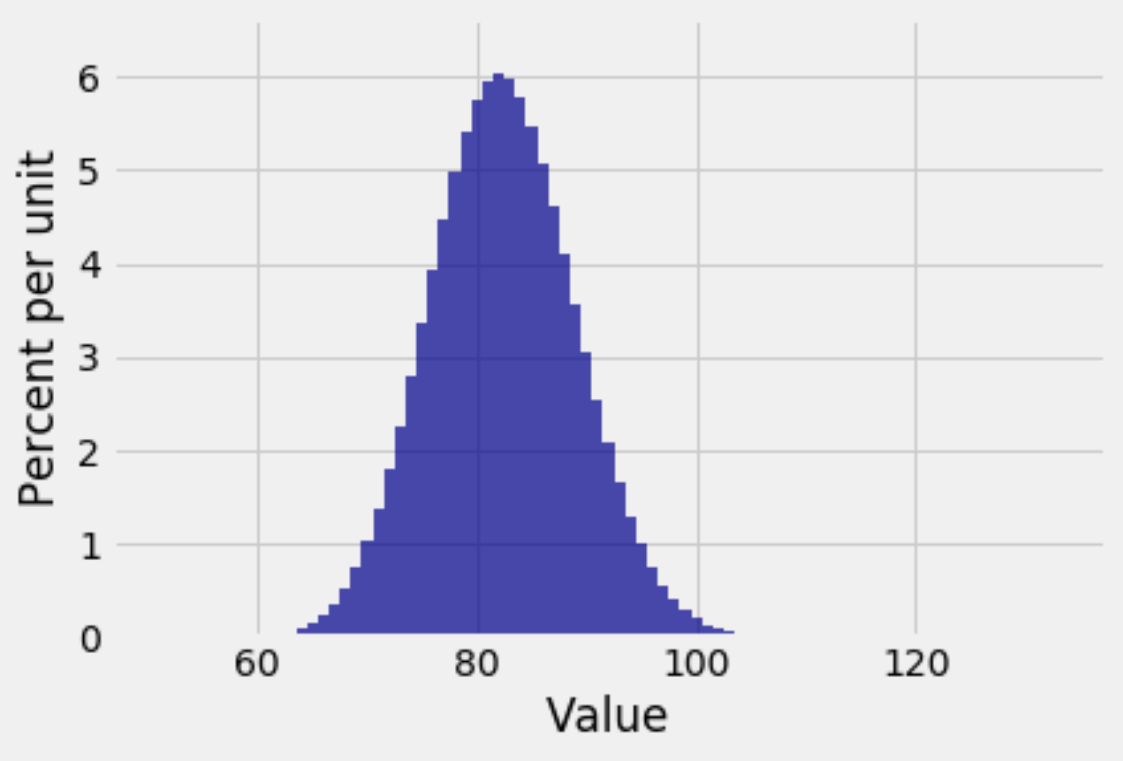
```
Out[10]: array([4, 5, 6, 7, 8])
```

```
In [15]: #Answer to 1d

         # The following is provided as a very brief skeleton
         # The solution uses more lines than are provided in the skeleton
         # Be sure to refer to your setup for (1a) and (1b)

         dist_Y = Table().values(np.arange(9)).probability(make_array(0, 0, 0, 0, 0.2, 0.2, 0.2, 0.2, 0
         probs_Y = dist_Y[1]
         coeffs_Y = np.flipud(probs_Y)
         pgf_Y =  np.poly1d(coeffs_Y)# pgf of Y
         pgf_SY =  pgf_Y**12 # pgf of S_Y
         pgf_W = pgf_SY*pgf_SX # pgf of W = S_X + S_Y
         probs_W = np.flipud(pgf_W.c)
         dist_W =  Table().values(np.arange(137)).probability(probs_W)# distribution object for W

         Plot(dist_W)
```



In [17]:
```
#Answer to 1e

        # Use dist_W here
        print("E(W) =", dist_W.ev())
        print("SD(W) =", dist_W.sd())

        # Use dist_X and dist_Y here

        print("E(W) =", dist_X.ev()*8 + dist_Y.ev()*12)
        print("SD(W) =", ((dist_X.sd()**2)*8 + (dist_Y.sd()**2)*12)**0.5)
```

```
E(W) = 82.40000000000006
SD(W) = 6.578753681359413
E(W) = 82.4
SD(W) = 6.578753681359411
```

1 newpage

2 newpage

3 newpage

4 newpage

5 newpage

```
In [4]: 4/12*2*100**2 + 8/12*2*200**2
```

```
Out[4]: 59999.99999999999
```

```
In [6]: (60000 - (166.67)**2)**0.5
```

```
Out[6]: 179.5023985912166
```