In class exercise for Tutorial 13: Looping back and forward a little

URL of my tutorial13Exercise on GitHub: https://github.com/kzapalac/FDS-2022-Exercises/blob/main/tutorial13Exercise_kz.ipynb

Simulating Data

In [2]:
```
# import what you need for numerical arrays and pretty plots
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [7]:
```
# Create arrays for your means and SDs values
mymeans = [5, 7, 9, 11, 13]
mysds = [2, 4, 6, 8, 10]
nrows = 1000
ncols = 5
```

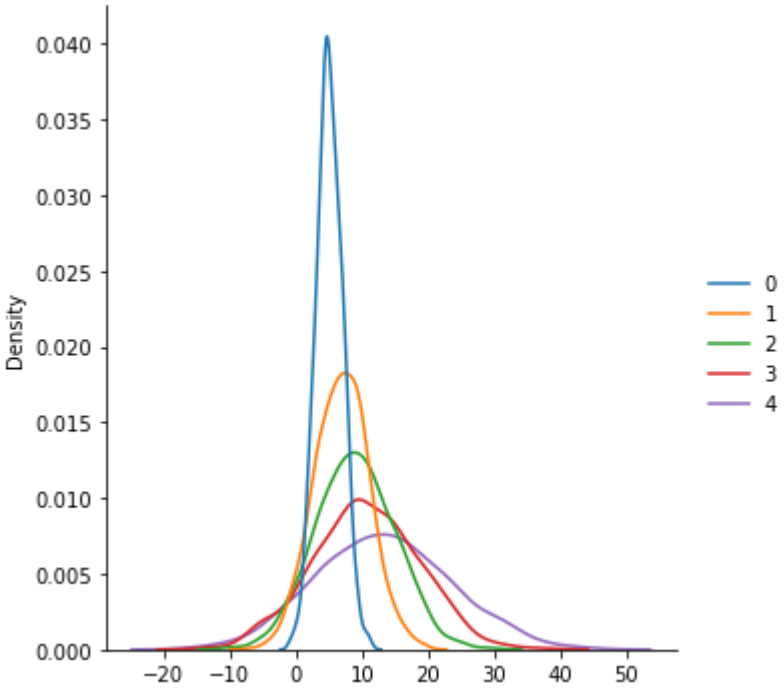
In [10]:
```
# create the distributions and store them
# into a single numpy array 'dist'
dist = np.zeros((nrows, ncols))

for j in range(ncols) :
    for i in range(nrows) :
        dist[i,j] = mymeans[j] + mysds[j]*np.random.randn()
```

In [13]:
```
#checking the means
np.mean(dist, axis=0)
```

Out[13]:
```
array([ 5.04435335,  6.89736163,  8.96946578, 10.55750966, 13.26759491])
```

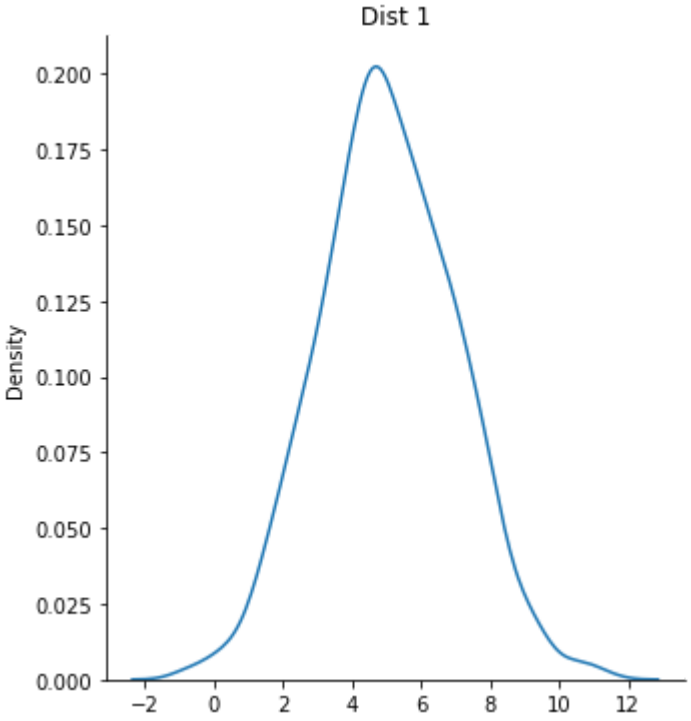
In [14]:
```
# Plot the KDE plots of the distributions in a single figure
sns.displot(dist, kind='kde')
```

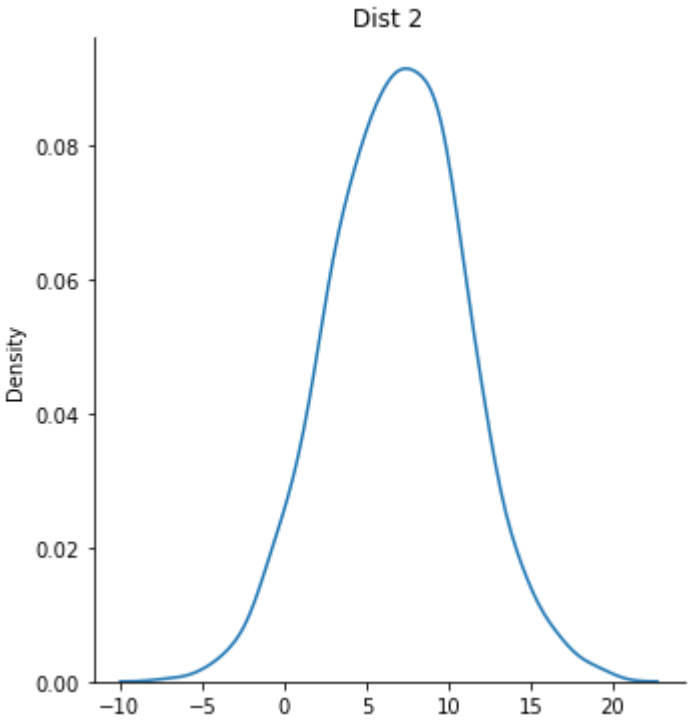
Out[14]:
```
<seaborn.axisgrid.FacetGrid at 0x1a9a9876280>
```

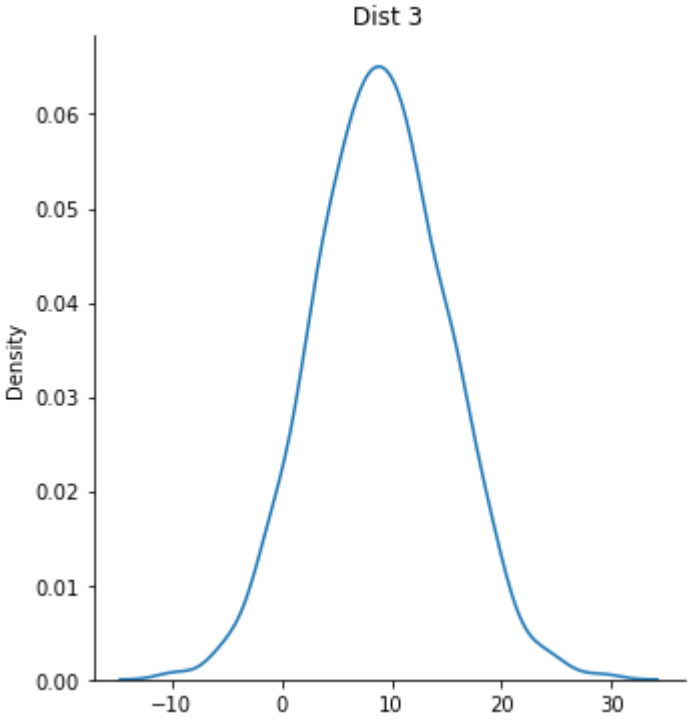


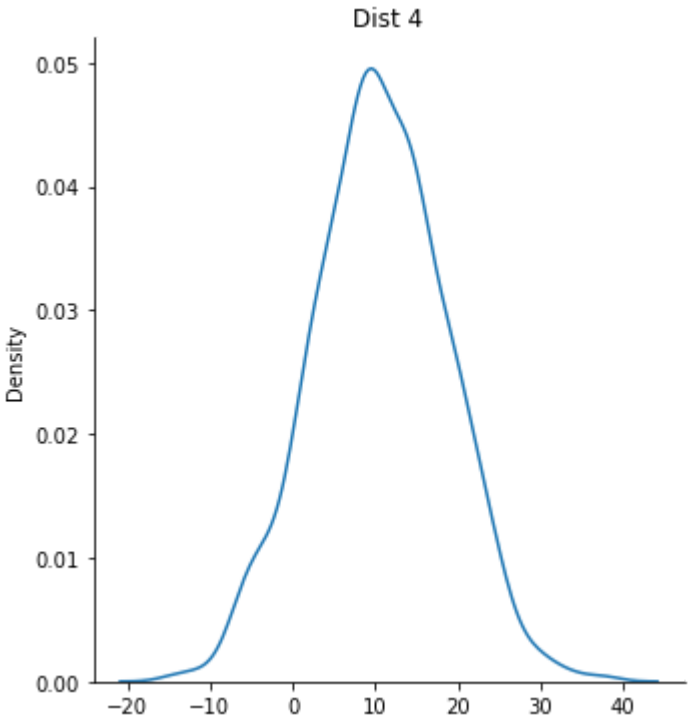
In [21]:

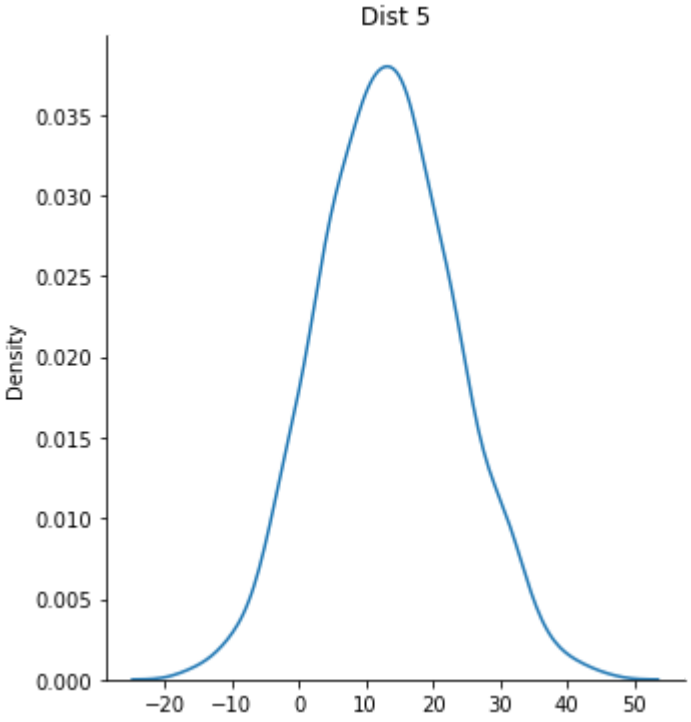
```
# Plot the KDE plots of the distributions in multiple subplots
#
# hint: you might need to use zip() to return two counters as a tuple
#       zip(my_index_vals, my_axis_vals)
#       https://docs.python.org/3/library/functions.html#zip
titles= ['Dist 1', 'Dist 2','Dist 3', 'Dist 4', 'Dist 5']
for j in range(ncols):
    sns.displot(dist[:,j], kind='kde')
    plt.title(titles[j])
```




Dist 2
Density
0.08
0.06
0.04
0.02
0.00
−10
−5
0
5
10
15
20


Dist 3
Density
0.06
0.05
0.04
0.03
0.02
0.01
0.00
−10
0
10
20
30


Dist 4
Density
0.05
0.04
0.03
0.02
0.01
0.00
−20
−10
0
10
20
30
40


Dist 5
Density
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
−20
−10
0
10
20
30
40
50