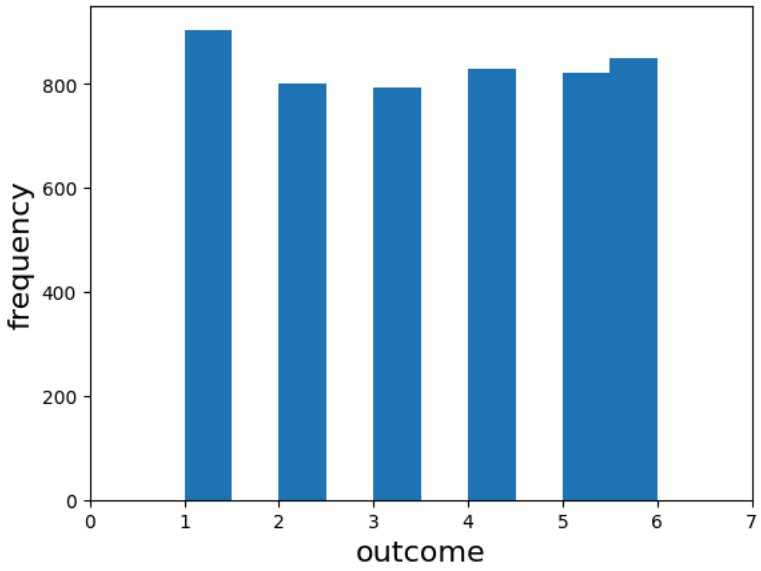
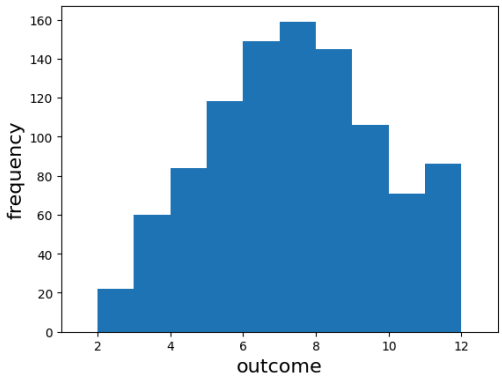
```
#
import numpy as np
import matplotlib.pyplot as plt
#
d1 = np.random.randint(1,7,5000) #random values 1,2,3,4,5,6
plt.hist(d1)
plt.xlabel('outcome', fontsize=16)
plt.ylabel('frequency', fontsize=16)
plt.xlim(0, 7)
plt.show()
#
# See Figure 3-8(a)
```




Start coding or generate with AI.

```
#
import numpy as np
import matplotlib.pyplot as plt
#
d1 =np. random.randint(1,7,1000)
d2 = np.random.randint(1,7,1000)
plt.hist(d1+d2)
plt.xlabel('outcome', fontsize=16)
plt.ylabel('frequency', fontsize=16)
plt.xlim(1, 13)
plt.show()
#
# See Figure 3-8(b)
```




File "<ipython-input-22-520e8e7ece16>", line 13 S

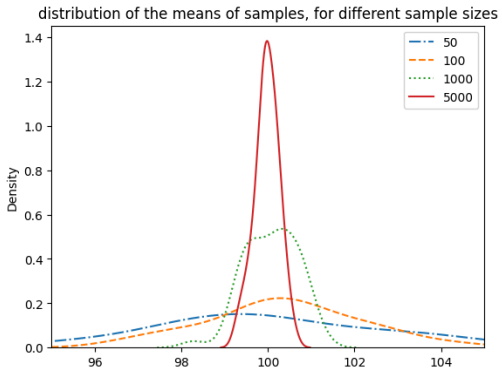
bin
boot
content
datalab
dev
etc
home
kaggle
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
tools
usr
var
NGC-DL-CONTAINER-LICENSE
cuda-keyring_1.0-1_all.deb

Double-click (or enter) to edit

Double-click (or enter) to edit

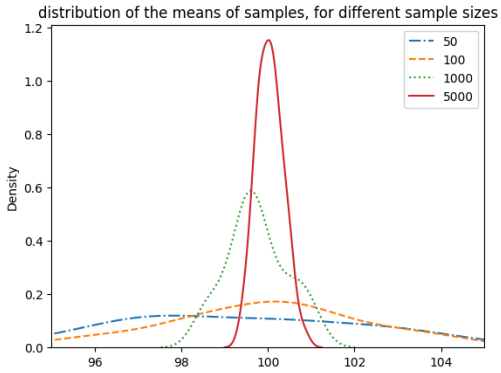
Start coding or generate with AI.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

mu, sigma, nPoultry = 100, 20, 50
samples = [50, 100, 1000, 5000]

ls = ['dashdot', 'dashed', 'dotted', 'solid']
l = 0

for nChicks in samples:
    S = 100
    for i in range(1, nPoultry):
        s = np.random.normal(mu, sigma, nChicks)
        S = np.append(S, s.mean())
    plt.xlim(95, 105)
    sns.kdeplot(S, linestyle=ls[l])
    l += 1
plt.legend(samples, loc='best', prop={"size": 10})  # font size 10
t = 'distribution of the means of samples, for different sample sizes'
plt.title(t, fontweight=10)
plt.show()
```




Double-click (or enter) to edit

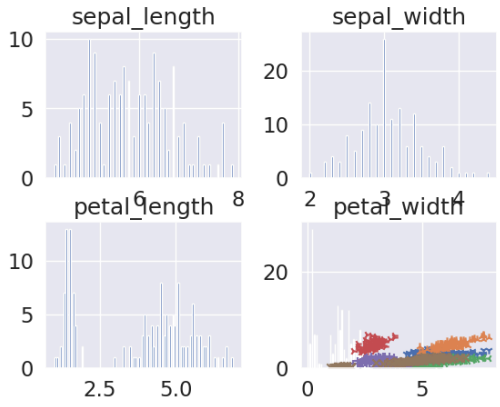
```
#
import numpy as np
import matplotlib.pyplot as plt
#
mu, sigma, nPoultry = 100, 20, 50
samples = [50,100,1000,5000]
#
import seaborn as sns
mu, sigma, nPoultry = 100, 20, 50
ls=['dashdot','dashed','dotted','solid']
l=0
for nChicks in samples:
    S = 100
    for i in range(1, nPoultry):
        s = np.random.normal(mu, sigma, nChicks)
        S = np.append(S,s.mean())
    plt.xlim(95,105)
    sns.kdeplot(S, linestyle=ls[l])
    l+=1
plt.legend(samples,loc='best',prop={"size":10}) #font size 10
t = 'distribution of the means of samples, for different sample sizes'
plt.title(t, fontweight=10)
plt.show()
# See Figure 3-16
#
```



```
# matplotlib is a Python library for plots.
import matplotlib.pyplot as plt
# seaborn gives rich graphics, and it uses matplotlib functions
# refer https://seaborn.pydata.org/
import seaborn as sb
sb.set(font_scale=1.5) #set font size to 150%
#
d=sb.load_dataset('penguins')
d.dropna(inplace=True) # drop rows having null valued cells
d.describe() # descriptive statistics of all numeric variables
#
# Figure 3-1(a): boxplot of mass by species
sb.boxplot (data=d, x='body_mass_g', y='species')
# Figure 3-1(b): boxplot of mass by species and island
```

```
sb.boxplot (data=d, x='body_mass_g', y='species', hue='island')
# Figure 3-2(a): histogram of mass by species
sb.histplot(data=d, x='body_mass_g', hue='species',multiple='stack')
# Figure 3-2(b): histogram of mass by island
sb.histplot(data=d, x='body_mass_g', hue='island',element='step' )
# Figure 3-3(a) Scatter Plot
plt.scatter(data=d, x='bill_length_mm', y ='flipper_length_mm')
plt.xlabel('bill_length_mm')
plt.ylabel('flipper_length_mm')
# Figure 3-3(b) KDE Plot
sb.kdeplot (data=d, x='bill_length_mm', y ='flipper_length_mm',

hue='species', shade=True)

#sb.stripplot(hue='sex', y='body_mass_g', x='island', data=d)
#

#
# See Figure 3-4
import seaborn as sb
pdf = sb.load_dataset('iris')
# Draw the histograms of all numeric variables
pdf.hist(bins=100)
#pdf.hist(bins=100, by='species')
# See Figure 3-5
import seaborn as sb # for loading the iris dataset
import matplotlib.pyplot as plt # for graphics
#
pdf = sb.load_dataset('iris')
leg = ['sl.sw', 'sl.pl', 'sl.pw', 'sw.pl', 'sw.pw', 'pl.pw' ]
#
plt.scatter(pdf['sepal_length'], pdf['sepal_width'], marker='1')
plt.scatter(pdf['sepal_length'], pdf['petal_length'], marker='2')
plt.scatter(pdf['sepal_length'], pdf['petal_width'], marker='3')
plt.scatter(pdf['sepal_width'], pdf['petal_length'], marker='4')
plt.scatter(pdf['sepal_width'], pdf['petal_width'], marker='+')
plt.scatter(pdf['petal_length'], pdf['petal_width'], marker='x')
#
```

```
<matplotlib.collections.PathCollection at
0x7c8c748d11e0>
```




Disk 83.30 GB available