Statistika Komputasi Bootstrap

Bootstrap mean dan resample

In [8]:
```
import random
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import mean_squared_error
import scipy.stats as st
```

In [74]:
```
# bootstrap mean dengan pengembalian
def btmean(data, r = 100, k = 2, alpha = 0.05):
    import random
    import numpy as np
    from sklearn.metrics import mean_squared_error
    import scipy.stats as st
    list_1 = []
    mean_true = np.repeat(np.mean(data), r)
    print("Start Bootstrapping")
    for i in range(r):
        list_2 = []
        for j in range(k):
            list_2.append(random.sample(data, 1))
        mean_sample = np.mean(list_2)
        list_1.append(mean_sample)
    print("End Bootstrapping")
    mean_resamp = np.mean(list_1)
    var_mean_resamp = np.var(list_1)
    mse_mean_resamp = mean_squared_error(list_1, mean_true)
    ci_mean_resamp = [np.mean(data) - st.norm.ppf(1-alpha/2)*var_mean_resamp**0.5, n
    return list_1, mean_resamp, var_mean_resamp, mse_mean_resamp, ci_mean_resamp
```

In [81]:
```
sample = [7, 11, 8, 30]
```

In [76]:
```
resample, mean_resample, var_mean_resample, mse_mean_resample, ci_mean_resample = bt
```

```
Start Bootstrapping
End Bootstrapping
```

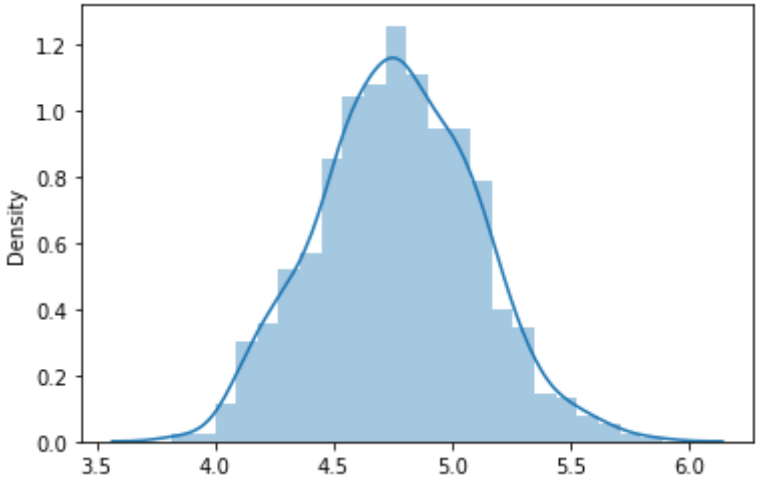
In [77]:
```
print("Mean Sesungguhnya: ", np.mean(sample))
print("Mean Bootstrap: ", mean_resample)
print("Var Mean Bootstrap: ", var_resample)
print("Error Bootstrap: ", mse_resample)
print("CI Mean Bootstrap: ", ci_mean_resample)
```

```
Mean Sesungguhnya:  14.0
Mean Bootstrap:  13.991657142857143
Var Mean Bootstrap:  0.018142862537758233
Error Bootstrap:  0.032047326892245154
CI Mean Bootstrap:  [11.852150188677895, 16.147849811322104]
```

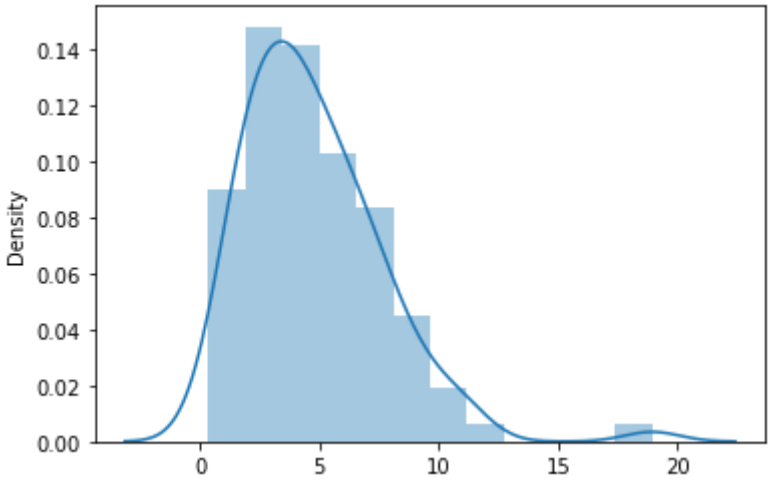
In [78]:
```
# Chisquare  - bootstrap - Normal
sample2 = np.random.chisquare(5, 100)
resample, mean_resample, _, _, _ = btmean(list(sample2), r=1000, k=70)
sns.distplot(resample)
plt.show()
print("Mean Bootstrap: ", mean_resample)
sns.distplot(sample2)
```

```
plt.show()
print("Mean True: ", np.mean(sample2))
```

```
Start Bootstrapping
End Bootstrapping
C:\Users\Amri\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Ple
ase adapt your code to use either `displot` (a figure-level function with similar fl
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```




```
Mean Bootstrap:  4.770027012795119
C:\Users\Amri\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Ple
ase adapt your code to use either `displot` (a figure-level function with similar fl
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



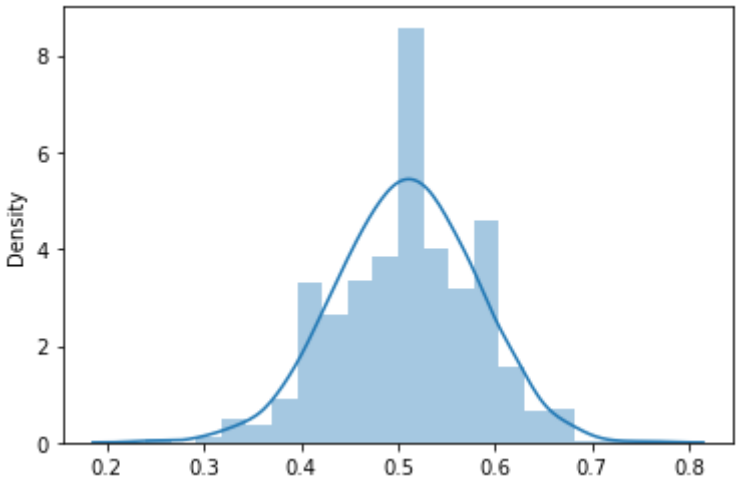

```
Mean True:  4.763617537210362
```

In [79]:

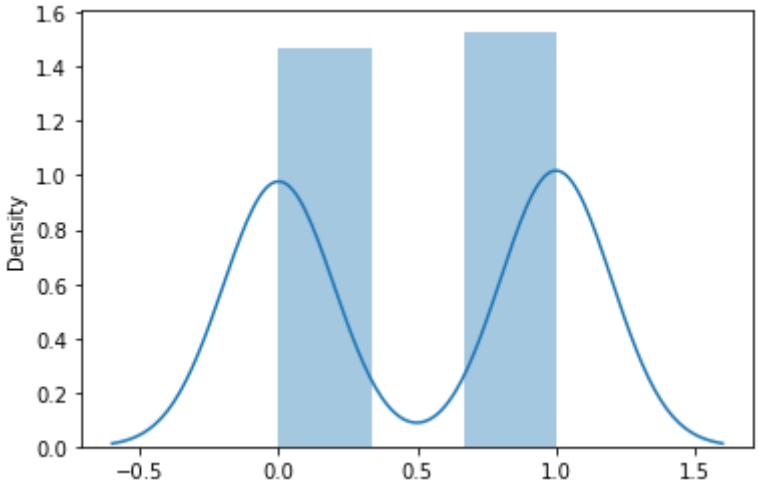
```
# Binomial to Normal
sample3 = np.random.binomial(1,0.5,100)
resample, mean_sample, _, _, _ = btmean(data = list(sample3), r = 1000, k = 50)
sns.distplot(resample, bins=20)
plt.show()
print(mean_sample)
sns.distplot(sample3)
plt.show()
print(np.mean(sample3))
```

```
Start Bootstrapping
End Bootstrapping
C:\Users\Amri\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Ple
ase adapt your code to use either `displot` (a figure-level function with similar fl
```

```
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```




```
0.50834
C:\Users\Amri\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Ple
ase adapt your code to use either `displot` (a figure-level function with similar fl
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

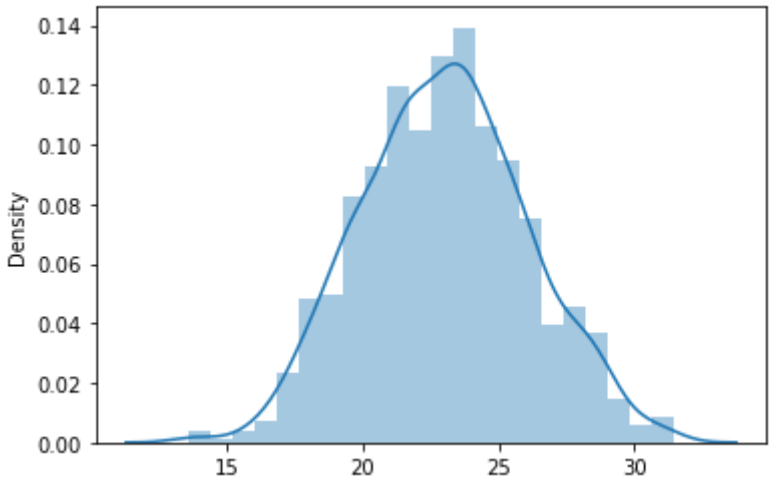



```
0.51
```

In [73]:

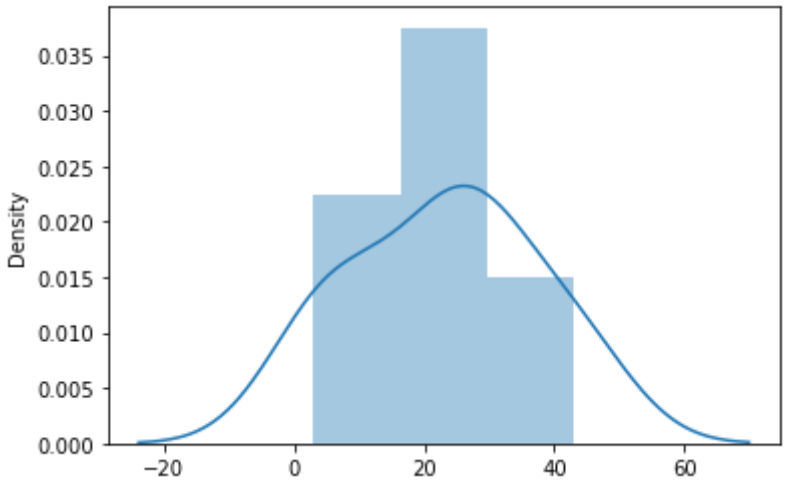
```
# Membangkitkan sample buatan
list_1 = list(np.random.randint(1, high=50, size=10))
resample, mean_sample, _, _, _ = btmean(data=list_1, r = 1000, k = 20)
sns.distplot(resample)
plt.show()
print(mean_sample)
sns.distplot(list_1)
plt.show()
print(np.mean(list_1))
```

```
Start Bootstrapping
End Bootstrapping
C:\Users\Amri\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Ple
ase adapt your code to use either `displot` (a figure-level function with similar fl
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



23.0246

```
C:\Users\Amri\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Ple
ase adapt your code to use either `displot` (a figure-level function with similar fl
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



23.1

Bootstrap variance

In [80]:
```
def btvar(data, r=100, k=2):
    import random
    import numpy as np
    list_1 = []
    for i in range(r):
        list_2 = []
        for j in range(k):
            list_2.append(random.sample(data, 1))
        var_sample = np.var(list_2)
        list_1.append(var_sample)
    var_resamp = np.mean(list_1)
    return list_1, var_resamp
```

In [82]:
```
# Bootstrap variance
_, var_resample = btvar(data=list(sample), r = 500, k = 50)
print(var_resample)
print(np.var(sample))
```

```
85.5192512
87.5
```

In [ ]: