In [36]:
```
from scipy import stats
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [51]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
gla_df=pd.read_csv(r"D:\NIT\DATASCIENCE\ARNAK TASK\1 to 100\day 1\stat\inf stats
```

In [3]:
```
bem_df=pd.read_csv(r"D:\NIT\DATASCIENCE\ARNAK TASK\1 to 100\day 1\stat\inf stats
```

In [4]:
```
gla_df
```

Out[4]:

	Date	Date.1	Close	gain
0	2010-01-05	2010-01-05	1616.80	-0.005444
1	2010-01-06	2010-01-06	1638.50	0.013422
2	2010-01-07	2010-01-07	1648.70	0.006225
3	2010-01-08	2010-01-08	1639.80	-0.005398
4	2010-01-11	2010-01-11	1629.45	-0.006312
...	...	...	...	...
1733	2016-12-26	2016-12-26	2723.50	-0.001283
1734	2016-12-27	2016-12-27	2701.75	-0.007986
1735	2016-12-28	2016-12-28	2702.15	0.000148
1736	2016-12-29	2016-12-29	2727.90	0.009529
1737	2016-12-30	2016-12-30	2729.80	0.000697

1738 rows × 4 columns

In [5]:
```
bem_df
```

Out[5]:

	Date	Date.1	Close	gain
0	2010-01-05	2010-01-05	1134.60	-0.000881
1	2010-01-06	2010-01-06	1139.60	0.004407
2	2010-01-07	2010-01-07	1144.15	0.003993
3	2010-01-08	2010-01-08	1144.05	-0.000087
4	2010-01-11	2010-01-11	1137.00	-0.006162
...	...	...	...	...
1733	2016-12-26	2016-12-26	950.25	-0.021924
1734	2016-12-27	2016-12-27	975.70	0.026782
1735	2016-12-28	2016-12-28	974.40	-0.001332
1736	2016-12-29	2016-12-29	986.05	0.011956
1737	2016-12-30	2016-12-30	1000.60	0.014756

1738 rows × 4 columns

In [7]:
```
gla_df.gain.mean()
```

Out[7]: 0.00038604108259229303

In [8]:
```
gla_df.gain.std()
```

Out[8]: 0.013360538552253302

In [9]:
```
bem_df.gain.mean()
```

Out[9]: 0.00027074807905723414

In [12]:
```
np.round(bem_df.gain.std(),4)
```

Out[12]: 0.0264

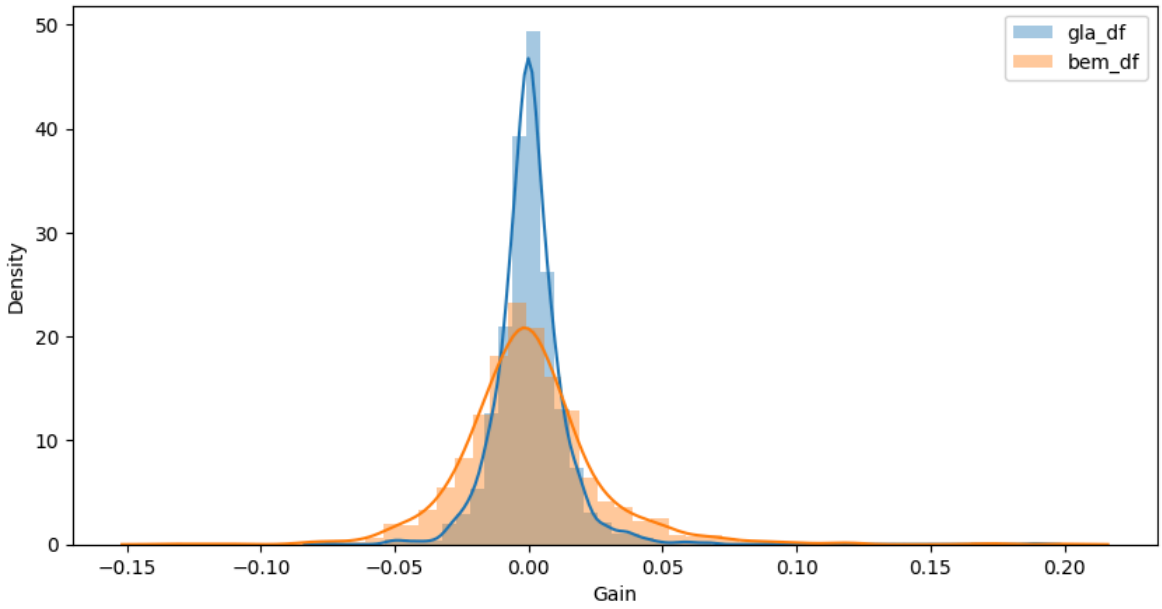
In [21]:
```
bem_df.describe()
```

Out[21]:

	Close	gain
count	1738.000000	1738.000000
mean	698.183688	0.000271
std	357.378754	0.026431
min	129.150000	-0.133940
25%	370.650000	-0.013736
50%	682.100000	-0.001541
75%	1010.350000	0.011985
max	1558.500000	0.198329

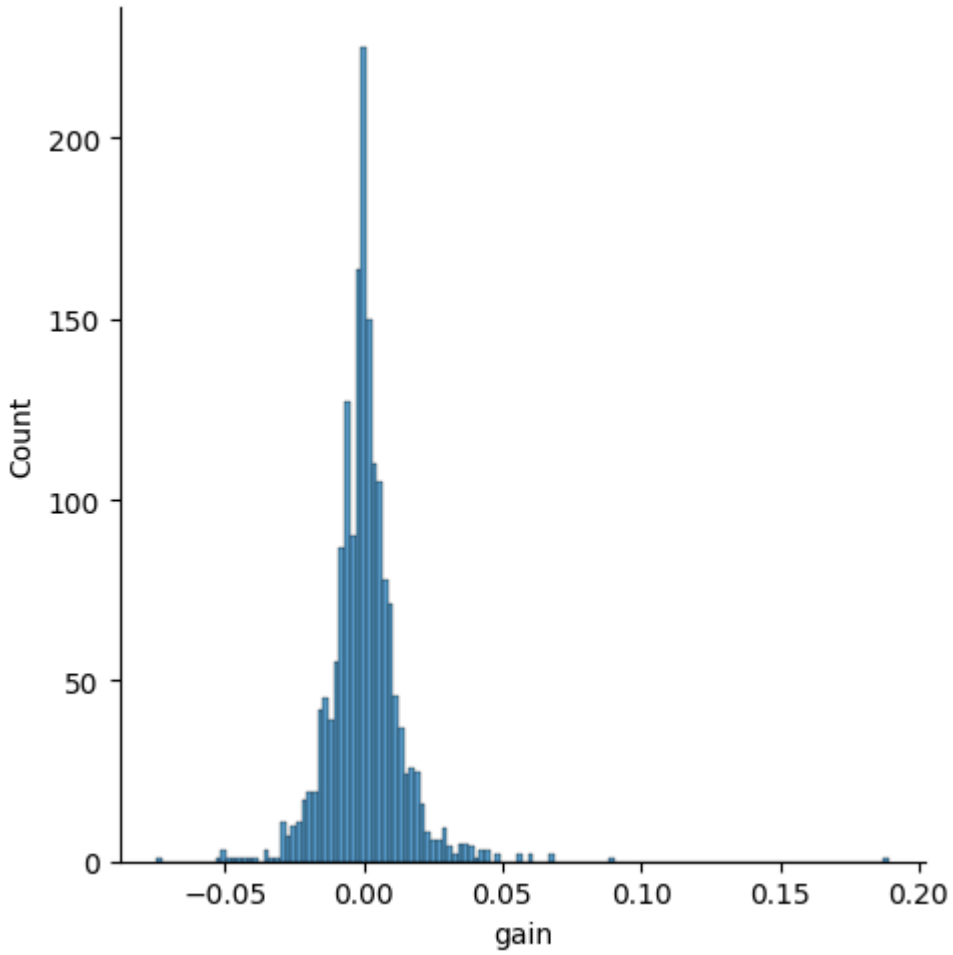
In [52]:
```
plt.figure(figsize=(10,5))
sns.distplot(gla_df.gain, label = 'gla_df')
sns.distplot(bem_df.gain, label = 'bem_df')
plt.xlabel('Gain')
plt.ylabel('Density')
plt.legend()
```

Out[52]:
```
<matplotlib.legend.Legend at 0x204384a64d0>
```



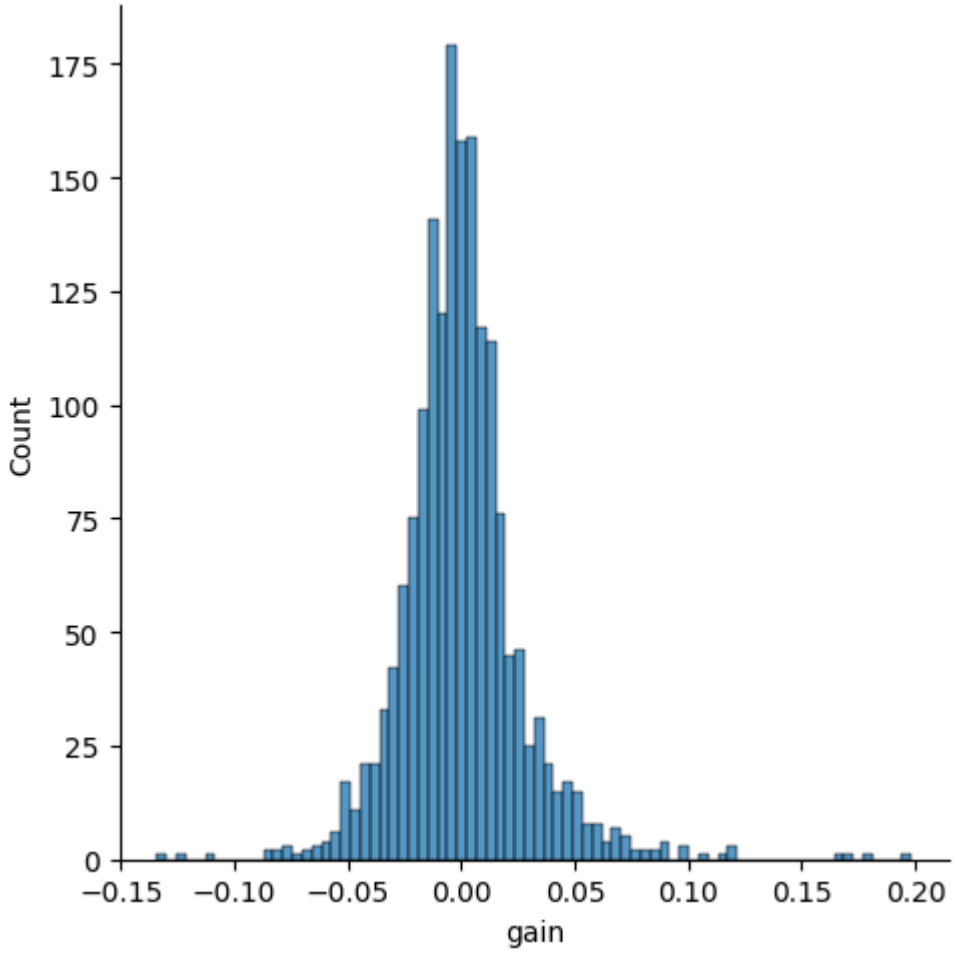
In [49]:
```
sns.displot(gla_df.gain)
```

Out[49]:
```
<seaborn.axisgrid.FacetGrid at 0x204386d7310>
```



In [17]:
```
sns.displot(bem_df.gain)
```

Out[17]:
```
<seaborn.axisgrid.FacetGrid at 0x2042baaa690>
```



confidence interval using the Normal-distribution

In [20]:
```
# Calculate the 95% confidence interval
glo_ci = stats.norm.interval(0.95, loc=0.000386, scale=0.013361)
# Round the interval to four decimal places
rounded_interval = np.round(glo_ci, 4)
print(rounded_interval)
```

```
[-0.0258  0.0266]
```

In [23]:
```
# Calculate the 95% confidence interval
bem_ci = stats.norm.interval(0.95, loc=0.00027074807905723414, scale=0.026431)
# Round the interval to four decimal places
rounded_interval = np.round(bem_ci, 4)
print(rounded_interval)
```

```
[-0.0515  0.0521]
```

confidence interval using the t-distribution

In [26]:
```
# Calculate the 95% confidence interval
bem_ci = stats.t.interval(0.95,1731, loc=0.00027074807905723414, scale=0.026431)
# Round the interval to four decimal places
```

```
rounded_interval = np.round(bem_ci, 4)
print(rounded_interval)
```

```
[-0.0516  0.0521]
```

In [27]:
```
# Calculate the 95% confidence interval
glo_ci = stats.t.interval(0.95,1731, loc=0.000386, scale=0.013361)
# Round the interval to four decimal places
rounded_interval = np.round(glo_ci, 4)
print(rounded_interval)
```

```
[-0.0258  0.0266]
```

In [ ]: