07_distribution_shifts

June 21, 2025

1 Distribution Shifts

- Consider our stock data.
- We are interested in testing changes in return distribution for our sample data around the time of the onset of the COVID 19 pandemic.

[1]:
```
%load_ext dotenv
%dotenv ../05_src/.env
import sys
sys.path.append("../05_src")
from logger import get_logger
_logs = get_logger(__name__)
```

[2]:
```
import dask
dask.config.set({'dataframe.query-planning': True})
import dask.dataframe as dd
import pandas as pd
import numpy as np
import os
from glob import glob
```

[3]:
```
ft_dir = os.getenv("FEATURES_DATA")
ft_glob = glob(ft_dir+'/*.parquet')
df = dd.read_parquet(ft_glob).compute().reset_index()
```

1.1 Data Preparation

- First, prepare four datasets, each with returns between March of a given year and March of the following year.
- For each data set, we can compute some descriptive statistics.
- We observe that there may be some distribution changes.

[4]:
```
df_2018 = df[(df['Date'] >= '2018-03-01') & (df['Date']  < '2019-03-01')]
df_2019 = df[(df['Date'] >= '2019-03-01') & (df['Date']  < '2020-03-01')]
df_2020 = df[(df['Date'] >= '2020-03-01') & (df['Date']  < '2021-03-01')]
df_2021 = df[(df['Date'] >= '2021-03-01') & (df['Date']  < '2022-03-01')]
df_2022 = df[(df['Date'] >= '2022-03-01') & (df['Date']  < '2023-03-01')]
```

[5]:
```
df_2018['returns'].describe()
```

[5]:
```
count    2257.000000
mean        0.008123
std         0.185545
min        -0.340990
25%        -0.007949
50%         0.001295
75%         0.009771
max         6.441369
Name: returns, dtype: float64
```

[6]:
```
df_2019['returns'].describe()
```

[6]:
```
count    2267.000000
mean        0.007072
std         0.216974
min        -0.303547
25%        -0.007273
50%         0.001091
75%         0.008462
max         9.660822
Name: returns, dtype: float64
```

[7]:
```
df_2020['returns'].describe()
```

[7]:
```
count    2259.000000
mean        0.009681
std         0.177753
min        -0.345949
25%        -0.010707
50%         0.002011
75%         0.014895
max         5.675929
Name: returns, dtype: float64
```

[8]:
```
df_2021['returns'].describe()
```

[8]:
```
count    2277.000000
mean        0.034039
std         1.116057
min        -0.101915
25%        -0.007290
50%         0.001064
75%         0.009198
max        51.348436
Name: returns, dtype: float64
```

[9]:
```
df_2022['returns'].describe()
```

[9]:
```
count    2259.000000
mean        0.016357
std         0.510301
min        -0.167932
25%        -0.012159
50%        -0.000541
75%         0.011908
max        22.977526
Name: returns, dtype: float64
```

2 Komogorov-Smirnov Test

- The KS test can be accessed via the scipy library: `scipy.stats.kstest`
- This function can be used to perform two sample tests.
- The null hypothesis is that the two distributions are identical.

[10]:
```
from scipy.stats import kstest

kstest(df_2018['returns'].dropna(),
       df_2019['returns'].dropna())
```

[10]:
```
KstestResult(statistic=0.034314065596832595, pvalue=0.13480604903839485,
statistic_location=0.013485812569593802, statistic_sign=-1)
```

[11]:
```
kstest(df_2019['returns'].dropna(),
       df_2020['returns'].dropna())
```

[11]:
```
KstestResult(statistic=0.13064753191322345, pvalue=2.670261822755509e-17,
statistic_location=0.013644776357206068, statistic_sign=1)
```

[12]:
```
kstest(df_2020['returns'].dropna(),
       df_2021['returns'].dropna())
```

[12]:
```
KstestResult(statistic=0.1100472943535476, pvalue=2.0196636826634023e-12,
statistic_location=0.01143078433526279, statistic_sign=-1)
```

[13]:
```
kstest(df_2021['returns'].dropna(),
       df_2022['returns'].dropna())
```

[13]:
```
KstestResult(statistic=0.0940567987164211, pvalue=3.4449944295519247e-09,
statistic_location=-0.007372480262691217, statistic_sign=-1)
```