Modeling Returns

Kannan Singaravelu, CQF

1 Normal Distribution

Normal Distribution, also known as Gaussian distribution is one of the most widely assumed distribution in Data Science. A normal distribution has a bell-shaped density curve described by its mean μ and standard deviation σ. The density curve is symmetrical, centered about its mean, with its spread determined by its standard deviation.

The probability distribution function of a normal density curve with mean μ and standard deviation σ at a given point x is given by:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

1.1 Import Libraries

We'll import the required libraries that we'll use in this example.

[ ]:
```
# Ignore Warnings
import warnings
warnings.filterwarnings('ignore')

# Import Pandas, Numpy and Scipy
import pandas as pd
import numpy as np
from scipy.stats import norm

# Import plotly express
import plotly.express as px
import plotly.graph_objects as go
px.defaults.width, px.defaults.height = 1000, 600
```

1.2 Load GBPUSD Data

[ ]:
```
# Load the CSV file
# df = pd.read_csv('CQF_Jan_24_M1L1_Excel-1.csv', index_col=0,␣
 ↪parse_dates=True)['2011':'2023']

df.tail()
```

[ ]:
```
# Visualize the plot to verify the data
px.line(df, y="Adj Close", title="SPY Historical Chart")
```

[ ]:
```
# Verify the datetime format
df.index
```

[ ]:
```
# Get last 300 index values
df.index[-300:]
```

1.3 Calculate return

[ ]:
```
# Calculate returns and add it to existing DataFrame as a column
df['Return'] = df['Adj Close'].pct_change().fillna(0)

# Get first 5 rows
df.head()
```

1.4 Calculate Mean, Sigma and Scaled Returns

[ ]:
```
# Calculate mean and sigma
mu = np.mean(df['Return'])
sigma = np.std(df['Return'])

# Calculate the scaled return : zscore
df['Scaled_Return'] = df['Return'].apply(lambda x: (x-mu)/sigma)

# Check the output
df.head()
```

1.5 Plot Histogram

[ ]:
```
# Generate some data (replace with your scaled returns data)
data = df['Scaled_Return']

# Create the normal distribution plot
x = np.linspace(data.min(), data.max(), 200)
y = norm.pdf(x, data.mean(), data.std())            # y = (1/np.sqrt(2*np.pi)*np.
 ↪exp(-0.5*x**2))

# Create the histogram
```

```
fig = px.histogram(data, x=data, histnorm='probability density', nbins=200,␣
↪labels={"x":""}, title='Empirical Vs Normal Distribution')

fig.add_trace(go.Scatter(x=x, y=y, mode='lines', name='Normal Distribution'))

# Show the plot
fig.show()
```

2 References

- Numpy Documentation
- Scipy Documentation
- Paul Wilmott (2007), Paul Wilmott introduces Quantitative Finance
- Python Resources

Python Labs by Kannan Singaravelu.