SzCzGiE-labs

Part I

In [73]:

```
# imports
import pandas as pd
import matplotlib.pyplot as plt
from math import log10, log2, log
import numpy as np
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
```

In [3]:

```
# loading data from csv
data = pd.read_csv('IBM.csv', parse_dates=['Date'])
N = len(data.index)
close = pd.Series(data['Close'], name='Close')
dates = pd.Series(data['Date'], name='Date')
```

In [4]:

```
data.head()
```

Out[4]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	1962-01-02	7.713333	7.713333	7.626667	7.626667	1.858243	390000
1	1962-01-03	7.626667	7.693333	7.626667	7.693333	1.874485	292500
2	1962-01-04	7.693333	7.693333	7.613333	7.616667	1.855805	262500
3	1962-01-05	7.606667	7.606667	7.453333	7.466667	1.819257	367500
4	1962-01-08	7.460000	7.460000	7.266667	7.326667	1.785148	547500

In [5]:

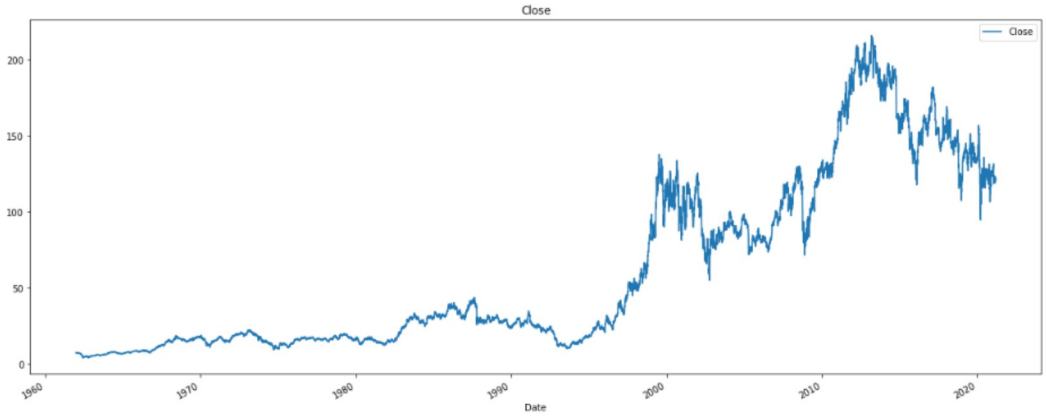
```
# separation of close price data
close = pd.DataFrame({'Date':data['Date'], 'Close':data['Close']})
close.set_index('Date')
pass
```

In [6]:

```
close.plot(figsize=(20,8), x='Date', y='Close', title='Close')
```

Out[6]:

```
<AxesSubplot:title={'center':'Close'}, xlabel='Date'>
```

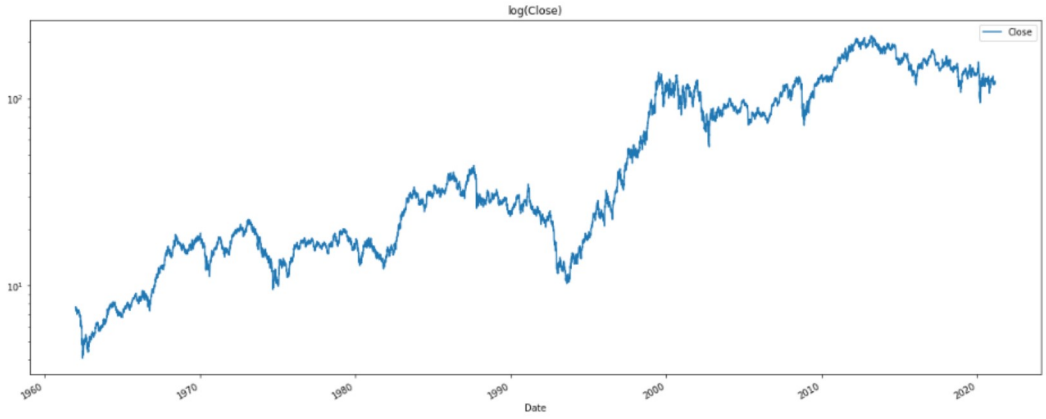


In [7]:

```
close.plot(figsize=(20,8), x='Date', y='Close', logy=True, title='log(Close)')
```

Out[7]:

```
<AxesSubplot:title={'center':'log(Close)'}, xlabel='Date'>
```



In [8]:

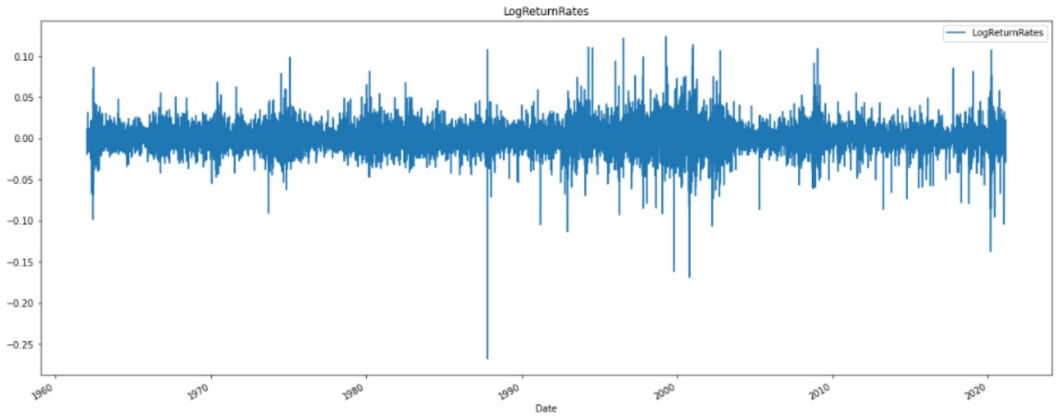
```
# logarithmic rates of return for close prices
close['LogReturnRates'] = pd.Series(np.array([0.0] + [log(close['Close'][i]/close['Close'][i-1]) for i in range(1, len(close['Close']))]))
```

In [9]:

```
close.plot(figsize=(20,8), x='Date', y='LogReturnRates', title='LogReturnRates')
```

Out[9]:

```
<AxesSubplot:title={'center':'LogReturnRates'}, xlabel='Date'>
```



In [10]:

```
# standarized logarithimc rates of return for close prices (using z-score: std=1, mean=0)
close['Z-score']= (close['LogReturnRates'] - close['LogReturnRates'].mean()) / close['LogReturnRates'].std()
close.head()
```

Out[10]:

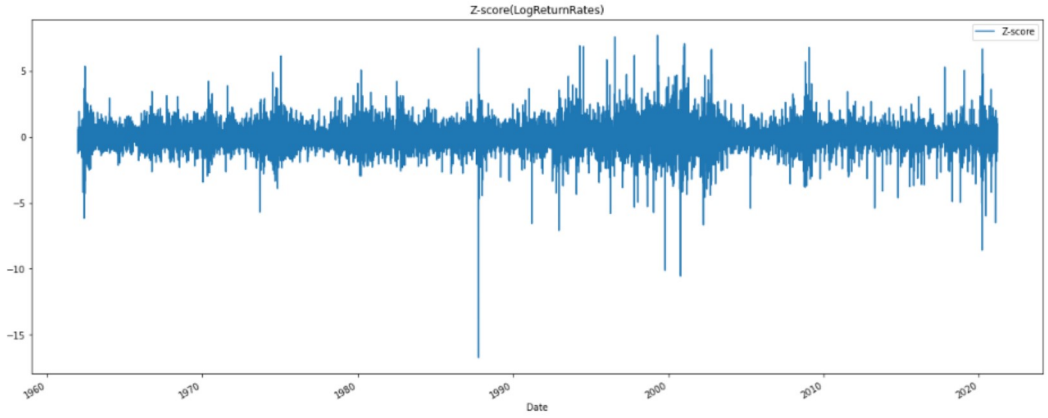
	Date	Close	LogReturnRates	Z-score
0	1962-01-02	7.626667	0.000000	-0.011651
1	1962-01-03	7.693333	0.008703	0.531862
2	1962-01-04	7.616667	-0.010015	-0.637102
3	1962-01-05	7.466667	-0.019890	-1.253790
4	1962-01-08	7.326667	-0.018928	-1.193704

In [11]:

```
close.plot(figsize=(20,8), x='Date', y='Z-score', title='Z-score(LogReturnRates)')
```

Out[11]:

```
<AxesSubplot:title={'center':'Z-score(LogReturnRates)'}, xlabel='Date'>
```



Part II

In [12]:

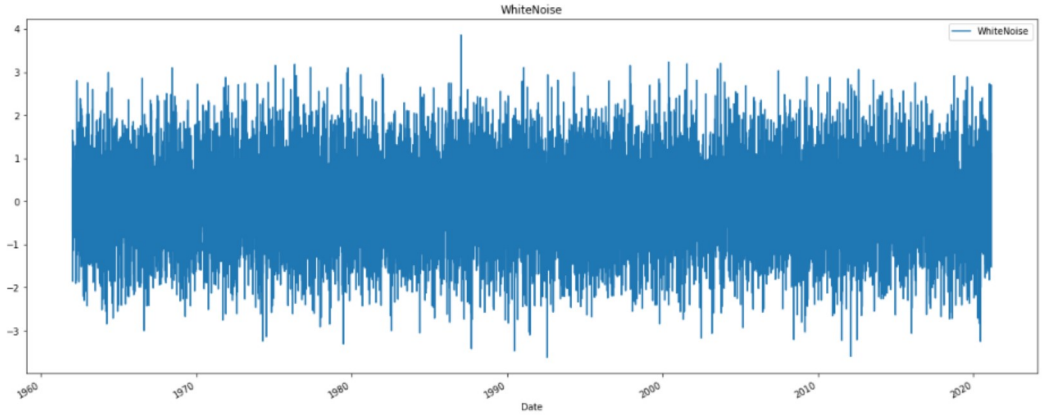
```
# white noise
mean = 0
std = 1
N = len(data['Close'])
white_noise = pd.DataFrame(data=np.random.normal(mean, std, size=N), columns=['WhiteNoise'], index=data['Date'])
```

In [13]:

```
white_noise.plot(figsize=(20,8), y='WhiteNoise', title='WhiteNoise')
```

Out[13]:

```
<AxesSubplot:title={'center':'WhiteNoise'}, xlabel='Date'>
```



In [14]:

```
# white noise cumulative sume
white_noise['CumulativeSum'] = white_noise.cumsum()
white_noise.head()
```

Out[14]:

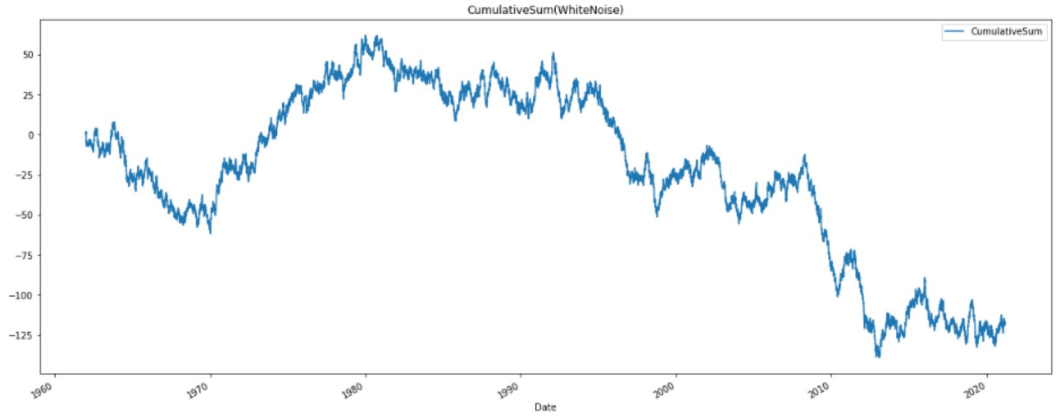
Date	WhiteNoise	CumulativeSum
1962-01-02	0.764974	0.764974
1962-01-03	-0.683988	0.080986
1962-01-04	1.656510	1.737495
1962-01-05	-0.301831	1.435664
1962-01-08	0.439684	1.875348

In [15]:

```
white_noise.plot(figsize=(20,8), y='CumulativeSum', title='CumulativeSum(WhiteNoise)')
```

Out[15]:

```
<AxesSubplot:title={'center':'CumulativeSum(WhiteNoise)'}, xlabel='Date'>
```

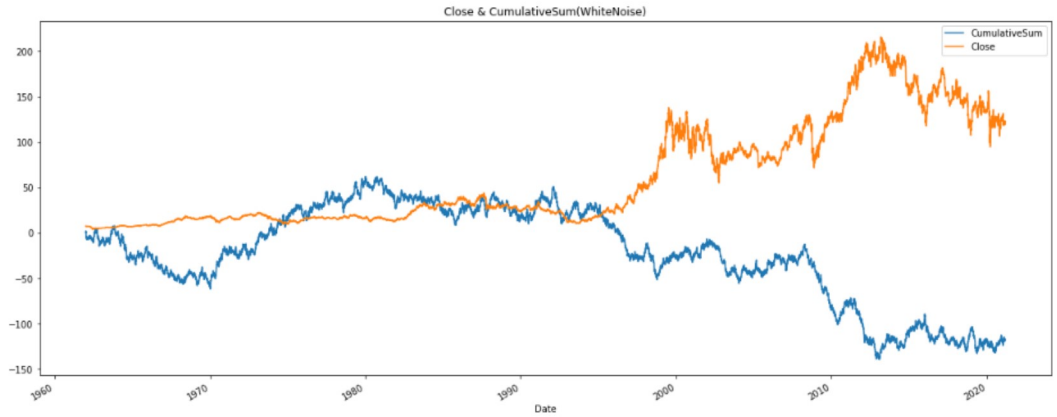


In [16]:

```
ax = white_noise.plot(y='CumulativeSum')
close.plot(figsize=(20,8), x='Date', y='Close', ax=ax, title='Close & CumulativeSum(WhiteNoise)', legend=True)
```

Out[16]:

```
<AxesSubplot:title={'center':'Close & CumulativeSum(WhiteNoise)'}, xlabel='Date'>
```

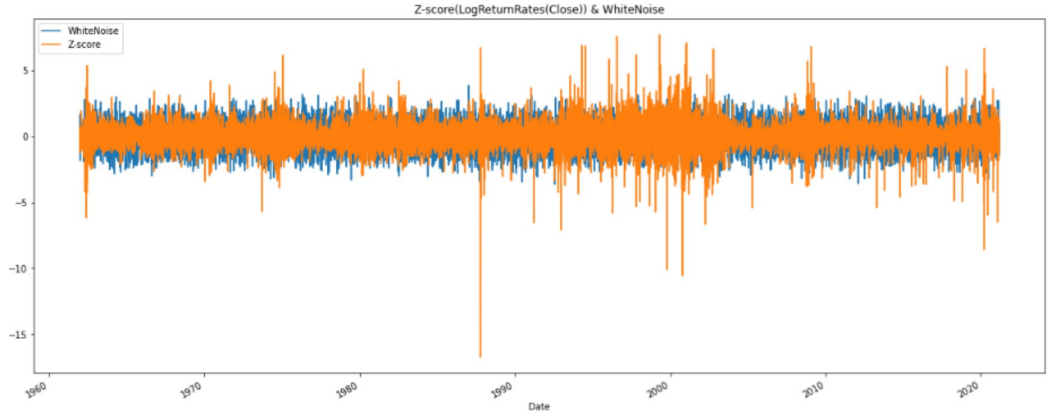


In [17]:

```
ax = white_noise.plot(y='WhiteNoise')
close.plot(figsize=(20,8), x='Date', y='Z-score', ax=ax, title='Z-score(LogReturnRates(Close)) & WhiteNoise', legend=True)
```

Out[17]:

```
<AxesSubplot:title={'center':'Z-score(LogReturnRates(Close)) & WhiteNoise'}, xlabel='Date'>
```



Part III

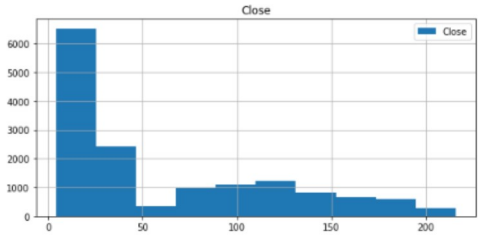
Histograms

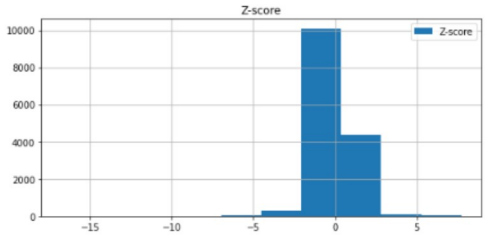
In [18]:

```
# Close prices histograms
close.hist(figsize=(20,4), column=['Close', 'Z-score'], legend=True)
```

Out[18]:

```
array([[<AxesSubplot:title={'center':'Close'}>,
        <AxesSubplot:title={'center':'Z-score'}>]], dtype=object)
```



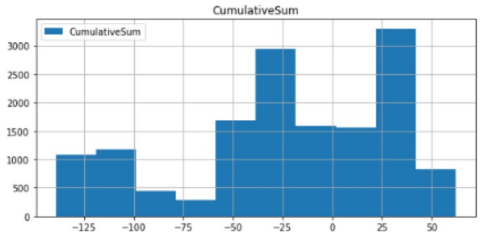


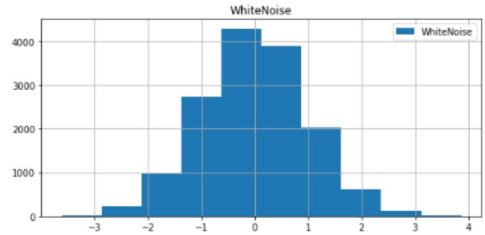
In [19]:

```
# White noise histograms
white_noise.hist(figsize=(20,4), column=['CumulativeSum', 'WhiteNoise'], legend=True)
```

Out[19]:

```
array([[<AxesSubplot:title={'center':'CumulativeSum'}>,
        <AxesSubplot:title={'center':'WhiteNoise'}>]], dtype=object)
```



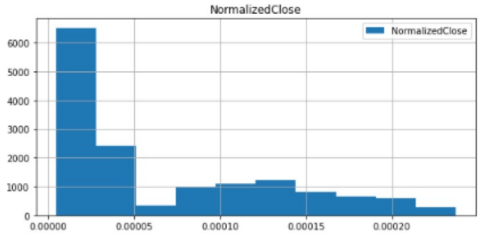


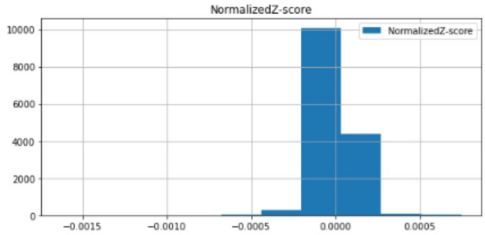
In [20]:

```
close['NormalizedClose'] = close['Close']/abs(close['Close']).sum()
close['NormalizedZ-score'] = close['Z-score']/abs(close['Z-score']).sum()
# Normalized Close prices histograms
close.hist(figsize=(20,4), column=['NormalizedClose', 'NormalizedZ-score'], legend=True)
```

Out[20]:

```
array([[<AxesSubplot:title={'center':'NormalizedClose'}>,
        <AxesSubplot:title={'center':'NormalizedZ-score'}>]], dtype=object)
```



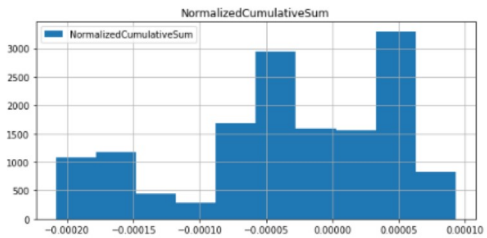


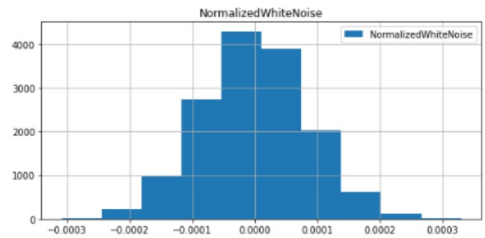
In [21]:

```
white_noise['NormalizedCumulativeSum'] = white_noise['CumulativeSum']/abs(white_noise['CumulativeSum']).sum()
white_noise['NormalizedWhiteNoise'] = white_noise['WhiteNoise']/abs(white_noise['WhiteNoise']).sum()
# Normalized White noise histograms
white_noise.hist(figsize=(20,4), column=['NormalizedCumulativeSum', 'NormalizedWhiteNoise'], legend=True)
```

Out[21]:

```
array([[<AxesSubplot:title={'center':'NormalizedCumulativeSum'}>,
        <AxesSubplot:title={'center':'NormalizedWhiteNoise'}>]],
      dtype=object)
```



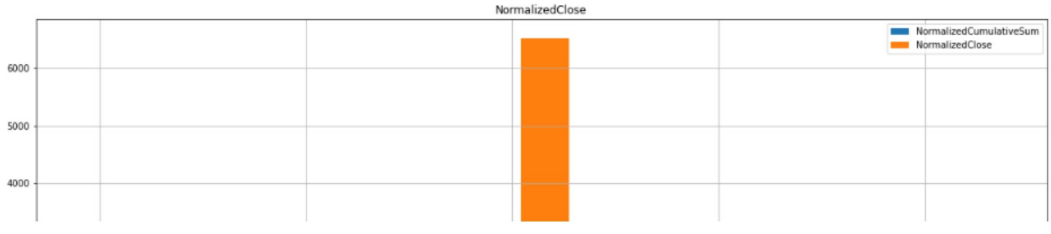


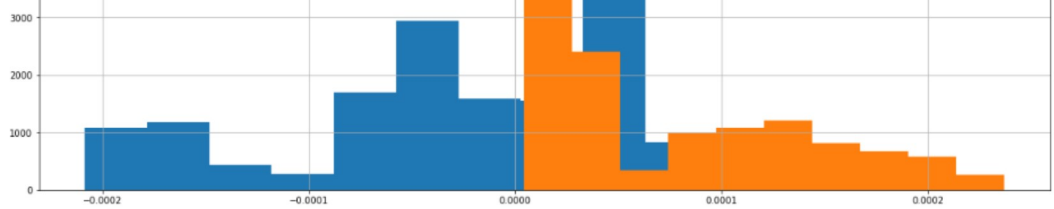
In [22]:

```
ax = white_noise.hist(figsize=(20,8), column=['NormalizedCumulativeSum'], legend=True)
close.hist(figsize=(20,8), column=['NormalizedClose'], ax=ax, legend=True)
```

Out[22]:

```
array([<AxesSubplot:title={'center':'NormalizedClose'}>], dtype=object)
```



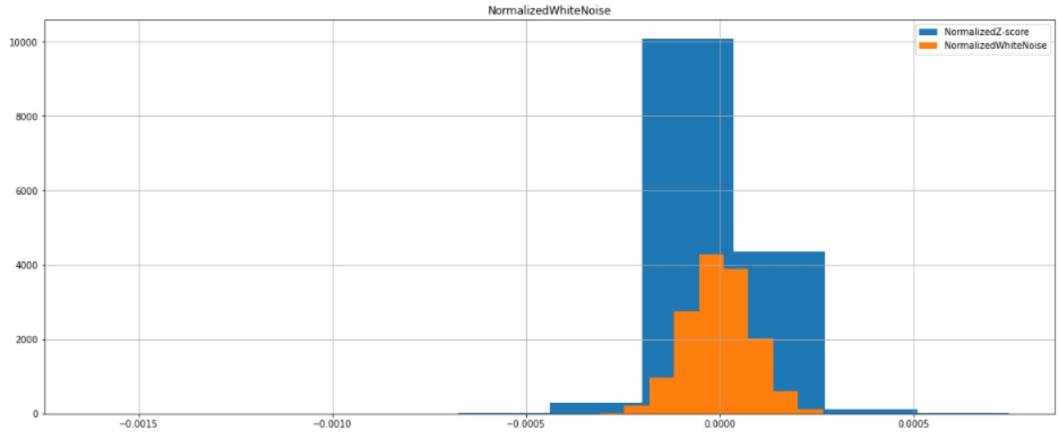


In [23]:

```
ax = close.hist(figsize=(20,8), column=['Normalized2-score'], legend=True)
white_noise.hist(figsize=(20,8), column=['NormalizedWhiteNoise'], ax=ax, legend=True)
```

Out[23]:

```
array([<AxesSubplot:title={'center':'NormalizedWhiteNoise'}>],
      dtype=object)
```



Distribution parameters

In [24]:

```
# close prices kurtosis
close.kurt(axis=0)
```

Out[24]:

```
Close                -0.485366
LogReturnRates       12.239468
Z-score              12.239468
NormalizedClose      -0.485366
Normalized2-score    12.239468
dtype: float64
```

In [25]:

```
# close prices skeweness
close.skew(axis=0)
```

Out[25]:

```
Close                 0.924353
LogReturnRates       -0.342694
Z-score              -0.342694
NormalizedClose       0.924353
Normalized2-score    -0.342694
dtype: float64
```

In [26]:

```
# white noise kurtosis
white_noise.kurt(axis=0)
```

Out[26]:

```
WhiteNoise               -0.008483
CumulativeSum            -0.663026
NormalizedCumulativeSum  -0.663026
NormalizedWhiteNoise     -0.008483
dtype: float64
```

In [27]:

```
# white noise skeweness
white_noise.skew(axis=0)
```

Out[27]:

```
WhiteNoise                0.004915
CumulativeSum            -0.596867
NormalizedCumulativeSum  -0.596867
NormalizedWhiteNoise      0.004915
dtype: float64
```

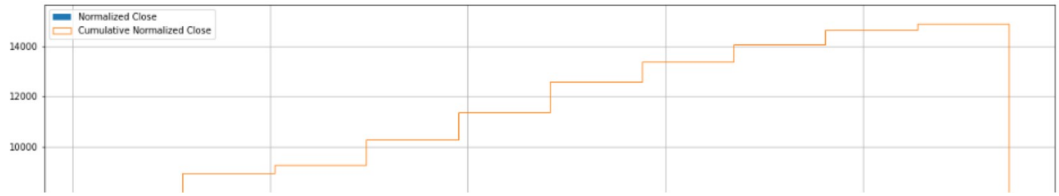
Cumulative Distribution Function for both series (close data and white noises)

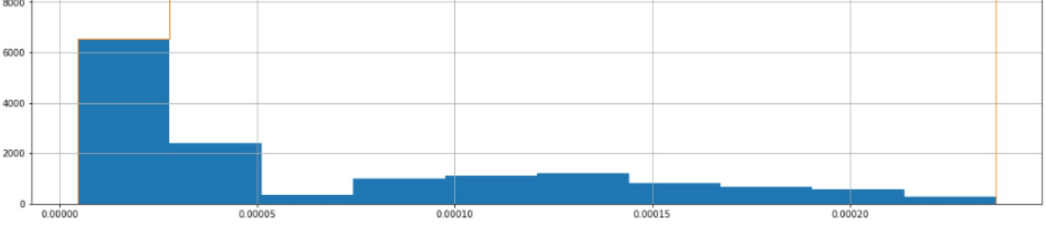
In [84]:

```
plt.figure(figsize=(20,8))
plt.grid(True)
plt.hist(close['NormalizedClose'], label="Normalized Close")
plt.hist(close['NormalizedClose'], cumulative=1, histtype='step', label="Cumulative Normalized Close")
plt.legend(loc='upper left')
```

Out[84]:

```
<matplotlib.legend.Legend at 0x28996f16430>
```



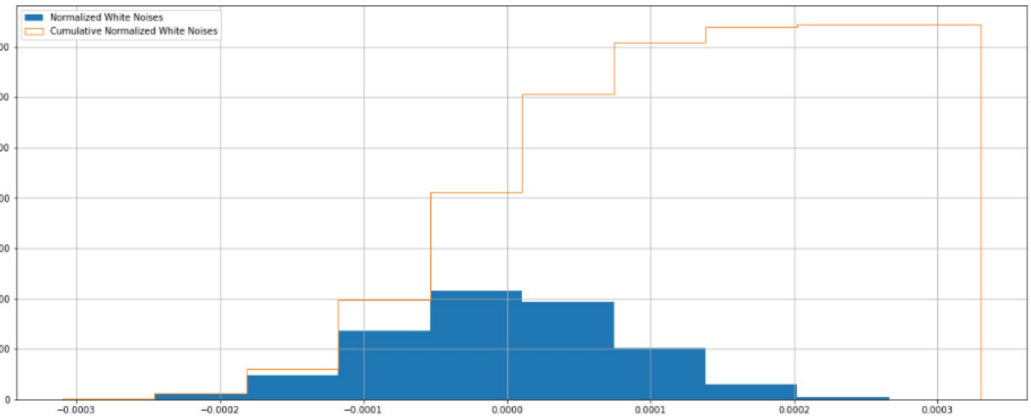


In [85]:

```
plt.figure(figsize=(20,8))
plt.grid(True)
plt.hist(white_noise['NormalizedWhiteNoise'], label="Normalized White Noises")
plt.hist(white_noise['NormalizedWhiteNoise'], cumulative=1, histtype='step', label="Cumulative Normalized White Noises")
plt.legend(loc='upper left')
```

Out[85]:

```
<matplotlib.legend.Legend at 0x2899681a310>
```



In [87]:

```
plt.figure(figsize=(20,8))
plt.grid(True)
plt.hist(close['NormalizedClose'], cumulative=1, histtype='step', label="Cumulative Normalized Close", log=True)
plt.hist(white_noise['NormalizedWhiteNoise'], cumulative=1, histtype='step', label="Cumulative Normalized White Noises", log=True)
plt.legend(loc='upper left')
```

Out[87]:

```
<matplotlib.legend.Legend at 0x289968d7d60>
```