1.) Pull in Data and Convert ot Monthly

In [37]:

```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [38]:

```
apple_data = yf.download('AAPL')
df = apple_data.resample("M").last()[["Adj Close"]]
```

```
[*********************100%***********************]  1 of 1 completed
```

2.) Create columns.

- Current Stock Price, Difference in stock price, Whether it went up or down over the next month, option premium

In [39]:

```
df.head()
```

Out[39]:

	Adj Close
Date	
1980-12-31	0.117887
1981-01-31	0.097591
1981-02-28	0.091546
1981-03-31	0.084637
1981-04-30	0.098023

In [40]:

```
# Difference in stock price
df["Diff"] = df["Adj Close"].diff().shift(-1)

#Target
df["Target"] = np.sign(df["Diff"])

df["Premium"] = .08 * df["Adj Close"]
```

3.) Pull in X data, normalize and build a LogReg on column 2

In [41]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
```

In [42]:

```
X = pd.read_csv("Xdata.csv", index_col="Date", parse_dates=["Date"])
```

In [43]:

```
y = df.loc[:"2023-09-30","Target"].copy()
df = df.loc[:"2023-09-30",:].copy()
```

In [44]:

```
# fit a log reg
logreg = LogisticRegression()
logreg.fit(X,y)
```

Out[44]:

```
▾ LogisticRegression

LogisticRegression()
```

4.) Add columns, prediction and profits.

In [45]:

```
y_pred = logreg.predict(X)
```

In [46]:

```
df["predictions"] = y_pred
```

In [47]:

```
df["profits"] = 0
```

In [48]:

```
#True positive
df.loc[(df["Target"] ==1) & (df["predictions"] ==1),"profits" ] = df["Premium"]

#False positive
df.loc[(df["Target"] == -1) & (df["predictions"] ==1),"profits" ] = 100 * df["Diff"] + df["Premium"]
```

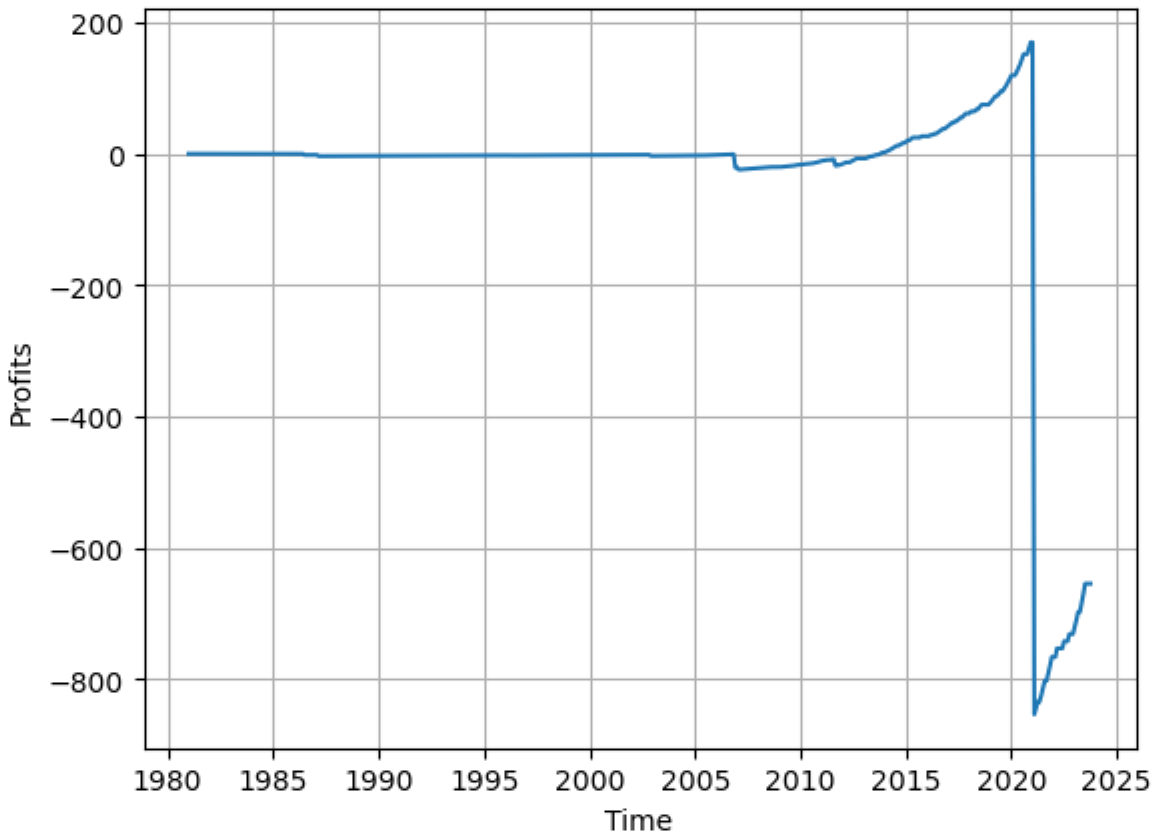
5.) Plot profits over time

In [49]:

```
plt.plot(np.cumsum(df["profits"]))
plt.grid()
plt.xlabel("Time")
plt.ylabel("Profits")
plt.plot()
```

Out[49]:

```
[]
```



5.5) Short write up about how you see your skills valuable to PJ and /or Philip Liu

I am more interested in Philip Liu's speech and NVIDIA's DGX cloud. With my Python and machine learning skills, I can develop and train models using frameworks like TensorFlow, or Keras. These models can be scaled up efficiently on DGX systems due to their high computational power. I can work on optimizing existing machine learning algorithms to run more efficiently on the DGX architecture. I may also help to contribute to test the performance of various AI models on DGX system.

6.) Create a loop that stores total profits over time

In [ ]:

7.) What is the optimal threshold and plot the total profits for this model.

In [ ]: