In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

1.) Clean the Apple Data to get a quarterly series of EPS.

In [2]:
```
y = pd.read_csv("AAPL_quarterly_financials.csv")
```

In [3]:
```
y.index = y.name
```

In [4]:
```
y = pd.DataFrame(y.loc["BasicEPS", :]).iloc[2:,:]
```

In [5]:
```
y.index = pd.to_datetime(y.index)
```

In [6]:
```
# CHECK IF NAS ARE NO DIVIDEND PERIOD
y = y.sort_index().fillna(0.)
```

In [7]:
```
y.isnull().sum()
```

Out[7]:
```
BasicEPS    0
dtype: int64
```

In [8]:
```
y.head()
```

Out[8]:

	BasicEPS
1985-09-30	0.0
1985-12-31	0.004
1986-03-31	0.002
1986-06-30	0.002
1986-09-30	0.0

In [9]:
```
y.tail()
```

Out[9]:

	BasicEPS
2022-09-30	1.29
2022-12-31	1.89
2023-03-31	1.53
2023-06-30	1.27
2023-09-30	1.47

2.) Come up with 6 search terms you think could nowcast earnings. (Different than the ones I used) Add in 3 terms that that you think will not Nowcast earnings. Pull in the gtrends data

In [10]:
```
from pytrends.request import TrendReq
```

In [56]:
```
# Create pytrends object
pytrends = TrendReq(hl='en-US', tz=360)

# Set up the keywords and the timeframe
keywords = ['Microsoft', "inflation", 'Apple Watch', 'layoffs', 'ebt', 'apple tv', 'wa
start_date = '2004-01-01'
end_date = '2024-01-01'

# Create an empty DataFrame to store the results
df = pd.DataFrame()

# Iterate through keywords and fetch data
for keyword in keywords:
    pytrends.build_payload([keyword], cat=0, timeframe=f'{start_date} {end_date}', geo
    interest_over_time_df = pytrends.interest_over_time()
    df[keyword] = interest_over_time_df[keyword]
```

In [57]:
```
df.resample("Q").mean()
```

Out[57]:

date	Microsoft	inflation	Apple Watch	layoffs	ebt	apple tv	waffle fries	valentines day	cemet
2004-03-31	93.000000	47.000000	0.000000	11.333333	5.333333	2.000000	4.333333	35.333333	71.333
2004-06-30	91.666667	43.333333	0.000000	9.333333	5.333333	1.333333	0.000000	1.000000	97.000
2004-09-30	86.333333	36.000000	0.000000	9.333333	5.666667	2.000000	12.000000	0.666667	81.000
2004-12-31	81.000000	39.000000	0.000000	10.000000	5.666667	2.000000	7.333333	1.666667	74.666
2005-03-31	83.000000	39.000000	0.000000	9.666667	5.000000	2.000000	3.000000	35.000000	70.333
...	...	...	...	...	...	...	...	...	
2023-03-31	27.333333	73.666667	65.333333	68.666667	67.000000	59.000000	75.666667	20.333333	11.666
2023-06-30	25.333333	61.000000	63.000000	38.333333	57.333333	62.000000	65.666667	1.000000	14.333
2023-09-30	25.333333	59.333333	71.333333	27.333333	70.666667	71.666667	66.000000	1.000000	13.000
2023-12-31	26.000000	60.333333	81.000000	29.666667	54.000000	66.333333	62.000000	1.333333	13.333
2024-03-31	28.000000	62.000000	72.000000	44.000000	59.000000	74.000000	67.000000	10.000000	12.000

81 rows × 9 columns

In [58]:
```
df = df.resample("Q").mean()
```

In [ ]:

In [59]:
```
# ALIGN DATA
temp = pd.concat([y, df],axis = 1).dropna()
y = temp[["BasicEPS"]].copy()
df = temp.iloc[:,1:].copy()
```

3.) Normalize all the X data

In [60]:
```
from sklearn.preprocessing import StandardScaler
```

In [61]:
```
scaler = StandardScaler()
```

In [62]:
```
X_scaler = scaler.fit_transform(df)
```

4.) Run a Lasso with lambda of .5. Plot a bar chart.

In [83]:
```
from sklearn.linear_model import Lasso
```

In [84]:
```
lasso = Lasso(alpha = .5)
```

In [90]:
```
lasso.fit(X_scaler,y)
```

Out[90]:
```
▾ Lasso
Lasso(alpha=0.5)
```

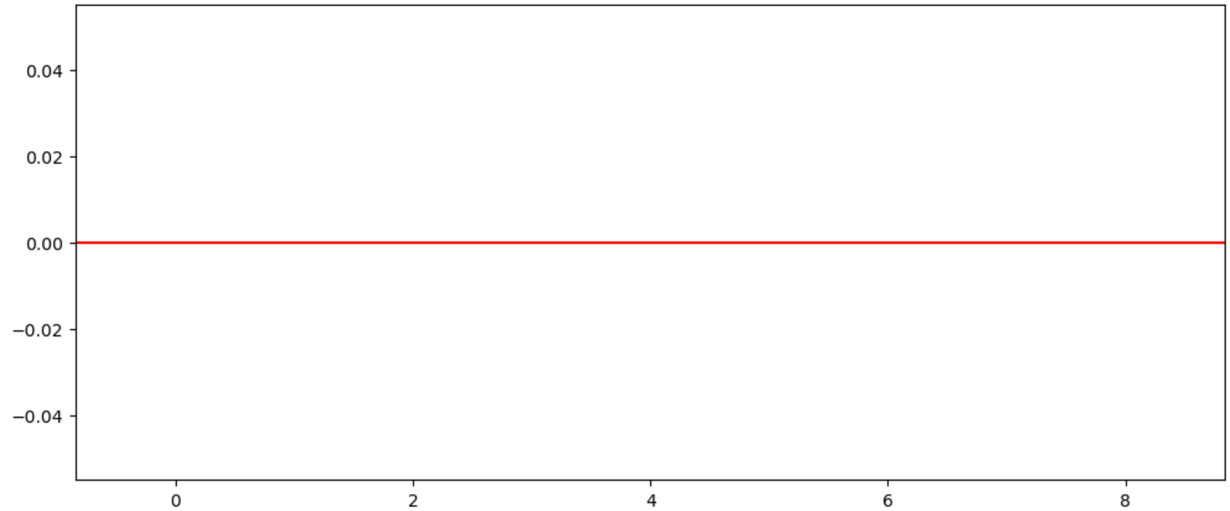
In [91]:
```
coefficients = lasso.coef_
```

In [92]:
```
df.head()
```

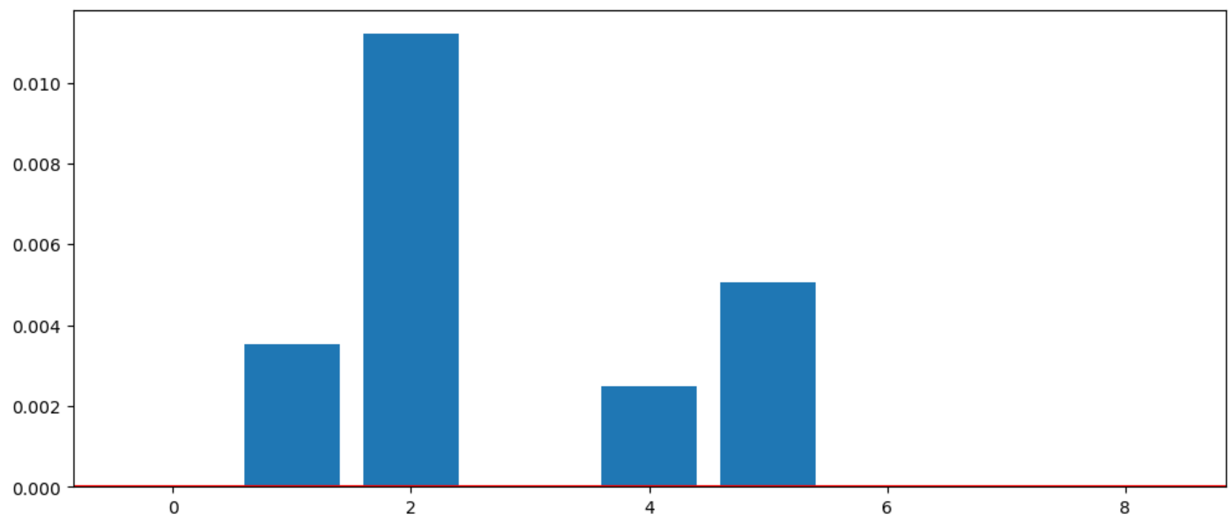
Out[92]:

	Microsoft	inflation	Apple Watch	layoffs	ebt	apple tv	waffle fries	valentines day	cemetary
2004-03-31	93.000000	47.000000	0.0	11.333333	5.333333	2.000000	4.333333	35.333333	71.333333
2004-06-30	91.666667	43.333333	0.0	9.333333	5.333333	1.333333	0.000000	1.000000	97.000000
2004-09-30	86.333333	36.000000	0.0	9.333333	5.666667	2.000000	12.000000	0.666667	81.000000
2004-12-31	81.000000	39.000000	0.0	10.000000	5.666667	2.000000	7.333333	1.666667	74.666667
2005-03-31	83.000000	39.000000	0.0	9.666667	5.000000	2.000000	3.000000	35.000000	70.333333

In [93]:
```
plt.figure(figsize = (12,5))
plt.bar(range(len(coefficients)), coefficients)
plt.axhline(0, color = "red")
plt.show()
#model under scaler transformations shows no models
```



In [89]:
```
plt.figure(figsize = (12,5))
plt.bar(range(len(coefficients)), coefficients)
plt.axhline(0, color = "red")
plt.show()
#this is the model under non scaler X
```



In [ ]:

5.) Do these coefficient magnitudes make sense?

magnitudes seem to make sense as the two of the strongest coeffieicnets remaining are the apple watch and apple tv which we would expect to be most strongly related, and the unrelated terms are dropped