Time Series Modeling

1. Plot the data with proper labeling and make some observations on the graph.

In [65]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from datetime import datetime
from sklearn import metrics
from statsmodels.tsa.ar_model import AutoReg
from numpy import sqrt
from sklearn.metrics import mean_squared_error
```

In ...

```
time_df = pd.read_csv("/Users/theranmeadows/Desktop/Bellevue University/dsc630pred
time_df.head()
```

Out[...

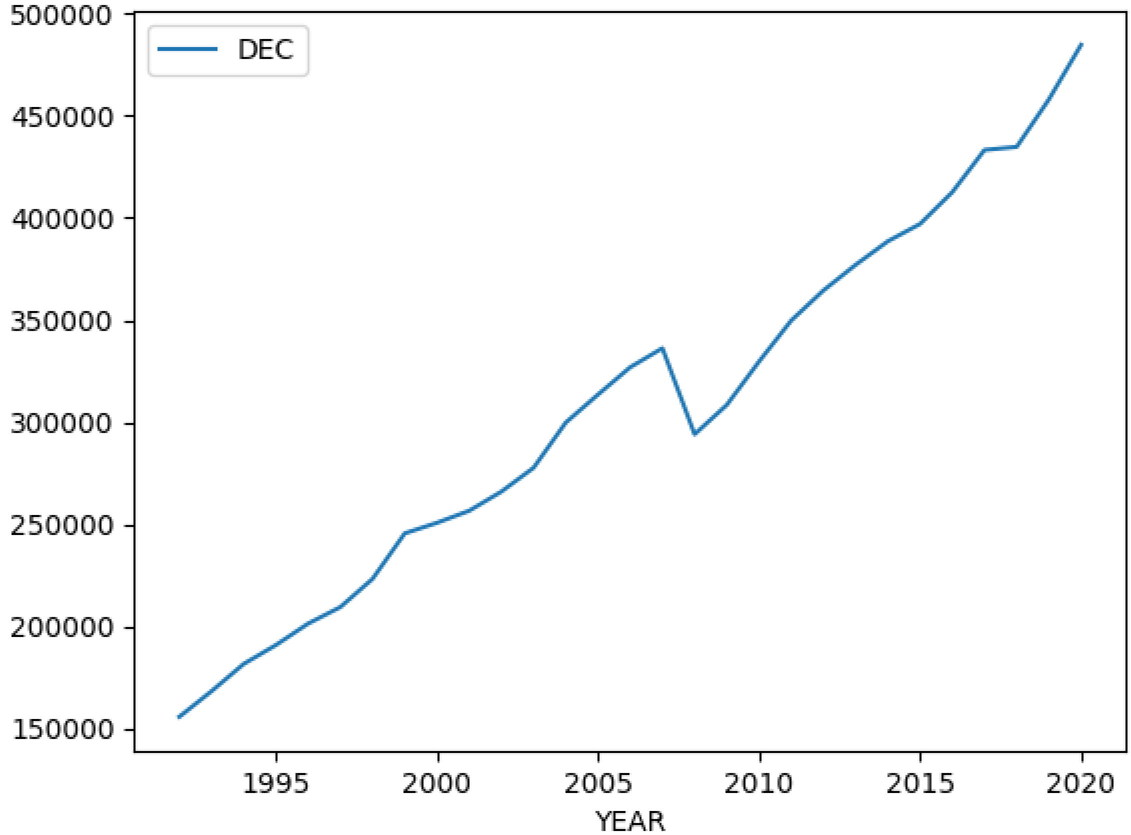
	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
0	1992	146925	147223	146805	148032	149010	149800	150761.0	151067.0	152588.0
1	1993	157555	156266	154752	158979	160605	160127	162816.0	162506.0	163258.0
2	1994	167518	169649	172766	173106	172329	174241	174781.0	177295.0	178787.0
3	1995	182413	179488	181013	181686	183536	186081	185431.0	186806.0	187366.0
4	1996	189135	192266	194029	194744	196205	196136	196187.0	196218.0	198859.0

In [46]:

```
time_df.plot('YEAR', 'DEC')
```

Out[46]:

```
<Axes: xlabel='YEAR'>
```



Although not pictured to save room, plotting all months by year showed a very similar line shape across all months. An increase from 1995 to 2006-2007, a sharp decline from 2008-2009 and then back to an increase all th way till 2021. We can confrim this with the MELT function below.

In [4...

```
#transpose the data with melt
time_df_short = pd.melt(time_df, id_vars=['YEAR'], var_name="MONTH", value_name='
```

In...

```
#assign number to the month names
time_df_short['Month'] = time_df_short['MONTH'].map({'JAN': 1, 'FEB': 2, 'MAR': 3,
```

In [4...

```
time_df_short['DATE']=pd.to_datetime(time_df_short[['YEAR','Month']].assign(DAY=
```

In [50]:

```
time_df_short.head()
```

Out[50]:

	YEAR	MONTH	SALES	Month	DATE
0	1992	JAN	146925.0	1	1992-01-01
1	1993	JAN	157555.0	1	1993-01-01
2	1994	JAN	167518.0	1	1994-01-01
3	1995	JAN	182413.0	1	1995-01-01
4	1996	JAN	189135.0	1	1996-01-01

In [51]:

```
#clean the data up
time_df_clean = time_df_short[['DATE','SALES']].sort_values('DATE')
time_df_clean = time_df_clean.dropna()
```

```
time_datetime_index = pd.DatetimeIndex(time_df_clean['DATE'].values)
time_df_clean = time_df_clean.set_index(time_datetime_index)
time_df_clean.drop('DATE', axis=1, inplace=True)
time_df_clean
```

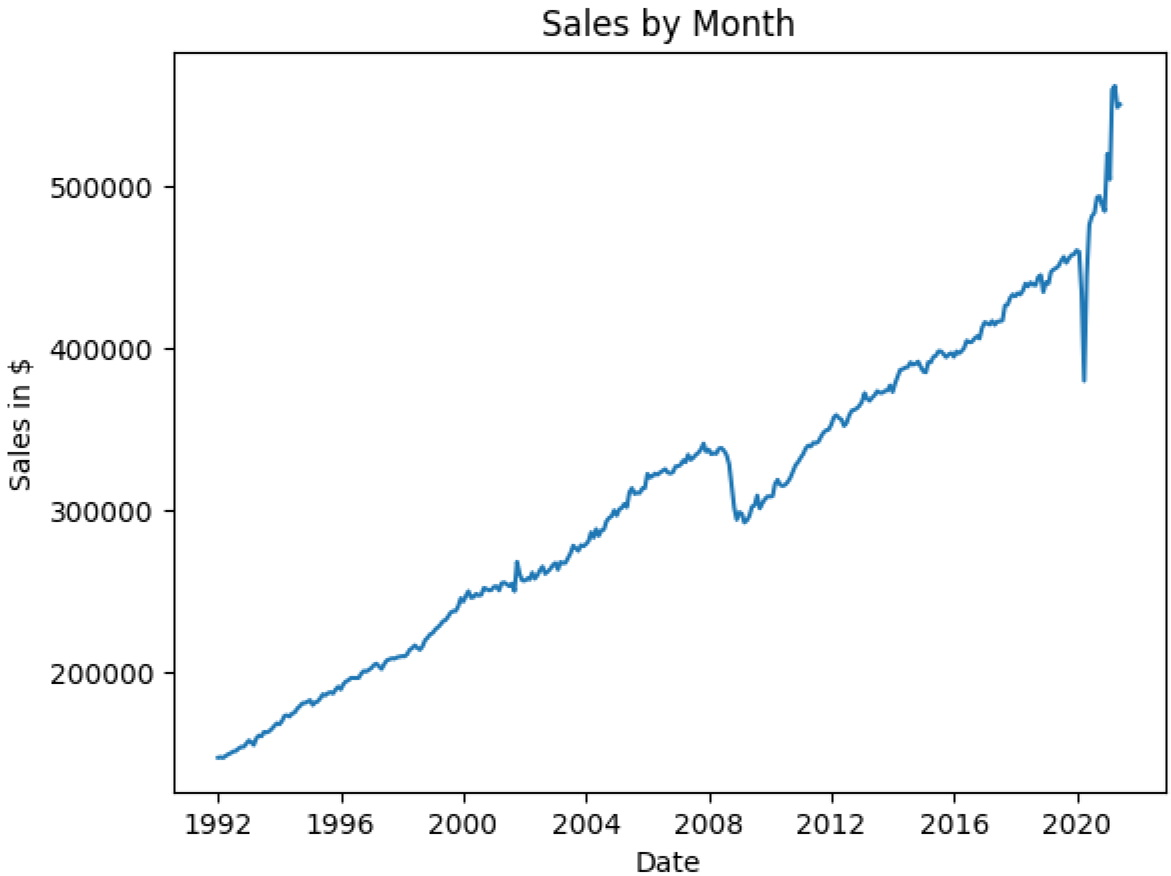
Out[51]:

	SALES
1992-01-01	146925.0
1992-02-01	147223.0
1992-03-01	146805.0
1992-04-01	148032.0
1992-05-01	149010.0
...	...
2021-02-01	504458.0
2021-03-01	559871.0
2021-04-01	562269.0
2021-05-01	548987.0
2021-06-01	550782.0

354 rows × 1 columns

In [52]:

```
#plot the new frame to confrim the line trends of the first graph
plt.plot(time_df_clean["SALES"])
plt.title('Sales by Month')
plt.xlabel('Date')
plt.ylabel('Sales in $')
plt.show()
```



2. Split this data into a training and test set. Use the last year of data (July 2020 – June 2021) of data as your test set and the rest as your training set.

In [53]:
```
train_time_df = time_df_clean[time_df_clean.index < '2020-07-01']
test_time_df = time_df_clean[time_df_clean.index >= '2020-07-01']
```

In [54]:
```
train_time_df
```

Out[54]:

	SALES
1992-01-01	146925.0
1992-02-01	147223.0
1992-03-01	146805.0
1992-04-01	148032.0
1992-05-01	149010.0
...	...
2020-02-01	459610.0
2020-03-01	434281.0
2020-04-01	379892.0
2020-05-01	444631.0

	SALES
2020-06-01	476343.0

342 rows × 1 columns

In [55]:
```
test_time_df
```

Out[55]:

	SALES
2020-07-01	481627.0
2020-08-01	483716.0
2020-09-01	493327.0
2020-10-01	493991.0
2020-11-01	488652.0
2020-12-01	484782.0
2021-01-01	520162.0
2021-02-01	504458.0
2021-03-01	559871.0
2021-04-01	562269.0
2021-05-01	548987.0
2021-06-01	550782.0

3. Use the training set to build a predictive model for the monthly retail sales.

In [58]:
```
time_model = AutoReg(train_time_df,lags=5).fit()
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/statsm
odels/tsa/base/tsa_model.py:473: ValueWarning: No frequency information was provided,
so inferred frequency MS will be used.
  self._init_dates(dates, freq)
```

In [59]:
```
print(time_model.summary())
```

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:                  SALES   No. Observations:                  342
Model:                     AutoReg(5)   Log Likelihood               -3375.942
Method:               Conditional MLE   S.D. of innovations           5424.567
Date:                Sat, 03 Feb 2024   AIC                           6765.884
Time:                        20:36:22   BIC                           6792.624
Sample:                    06-01-1992   HQIC                          6776.542
                         - 06-01-2020
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       2180.0628   1089.933      2.000      0.045      43.833    4316.293
```

```
SALES.L1       0.8523      0.055     15.449      0.000       0.744       0.960
SALES.L2      -0.3524      0.080     -4.407      0.000      -0.509      -0.196
SALES.L3       0.3427      0.130      2.629      0.009       0.087       0.598
SALES.L4       0.2459      0.134      1.839      0.066      -0.016       0.508
SALES.L5      -0.0908      0.100     -0.904      0.366      -0.288       0.106
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0013           -0.0000j            1.0013           -0.0000
AR.2            0.0573           -1.2351j            1.2364           -0.2426
AR.3            0.0573           +1.2351j            1.2364            0.2426
AR.4           -2.0016           -0.0000j            2.0016           -0.5000
AR.5            3.5941           -0.0000j            3.5941           -0.0000
-----------------------------------------------------------------------------
```

4. Use the model to predict the monthly retail sales on the last year of data.

In [...

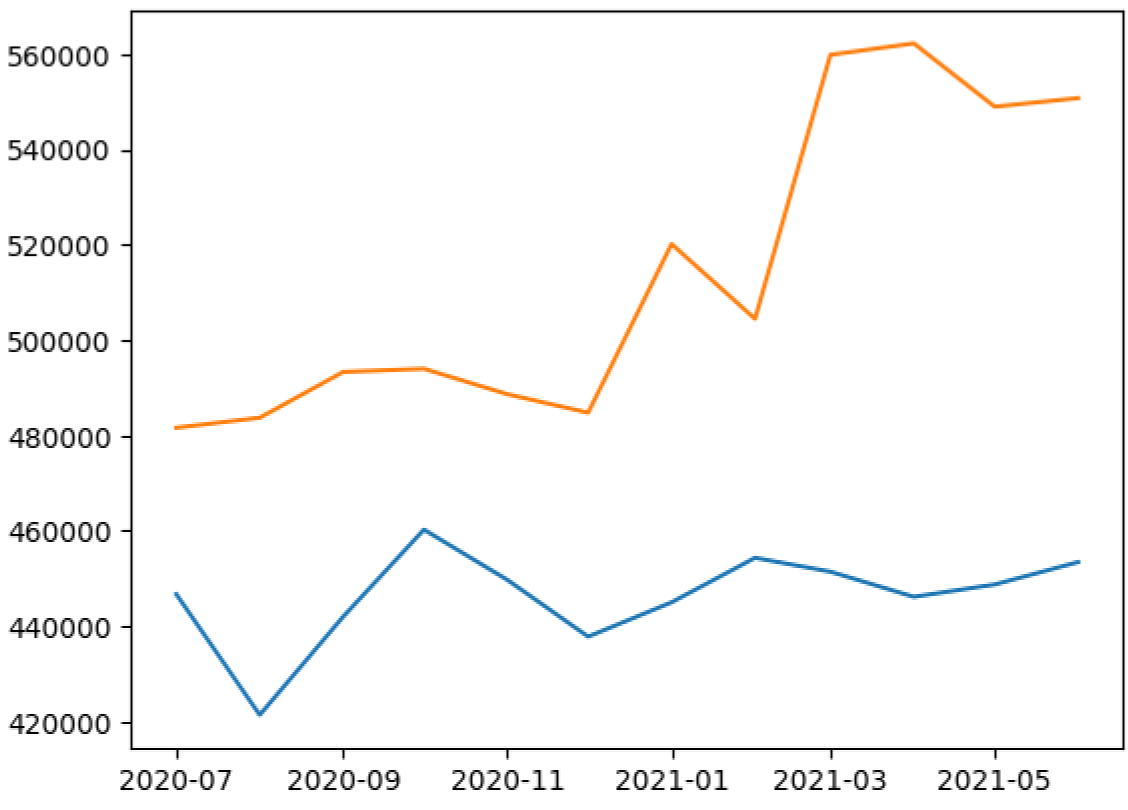
```
time_pred = time_model.predict(start = len(train_time_df), end = (len(time_df_cle
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/statsm
odels/tsa/deterministic.py:302: UserWarning: Only PeriodIndexes, DatetimeIndexes with
a frequency set, RangesIndexes, and Index with a unit increment support extending. The
index is set will contain the position relative to the data length.
  fcast_index = self._extend_index(index, steps, forecast_index)
```

In [61]:

```
plt.plot(time_pred)
plt.plot(test_time_df)
```

Out[61]: [<matplotlib.lines.Line2D at 0x1200c6690>]



5. Report the RMSE of the model predictions on the test set.

In [67]:

```
rmse = sqrt(mean_squared_error(test_time_df, time_pred))
print('Test RMSE: %.3f' % rmse)
```

```
Test RMSE: 73875.521
```