WESTYN HILLIARD

```
[16]: # Imports

      import pandas as pd
      import matplotlib.pyplot as plt
      from statsmodels.tsa.arima.model import ARIMA
      from sklearn.metrics import mean_squared_error
      import numpy as np
```

1.1 Load the Dataset

```
[4]: # Load the dataset
     file_path = 'us_retail_sales.csv'  # Make sure to update this with the actual␣
      ↪path
     retail_data = pd.read_csv(file_path)

     # Display the first few rows to understand the structure
     retail_data.head()
```

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

```
        SEP       OCT       NOV       DEC
0  152588.0  153521.0  153583.0  155614.0
1  163258.0  164685.0  166594.0  168161.0
2  178787.0  180561.0  180703.0  181524.0
3  187366.0  186565.0  189055.0  190774.0
4  198859.0  200509.0  200174.0  201284.0
```

1.2 Transform the Dataset

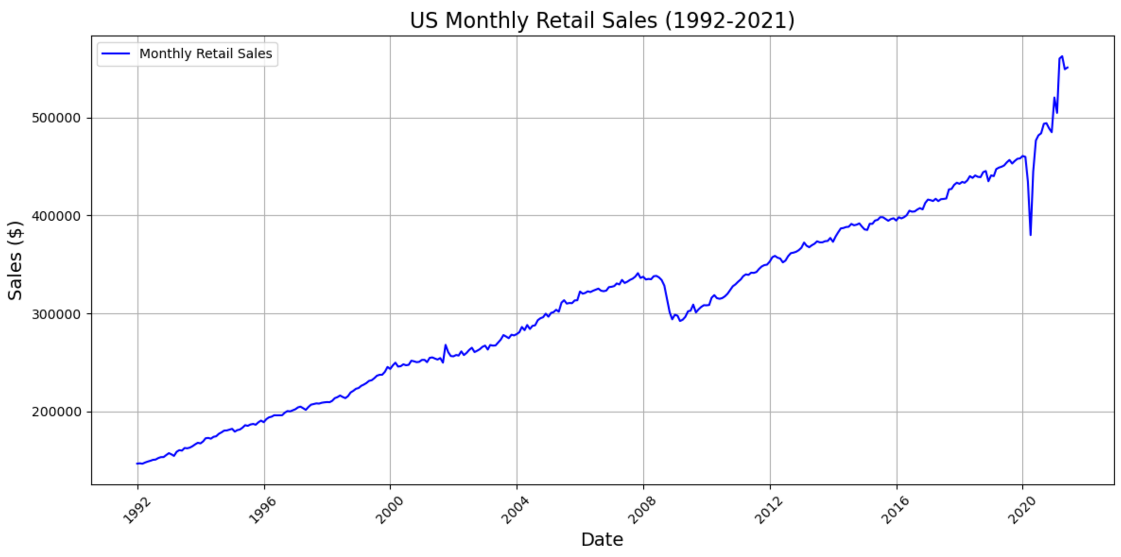
[5]:
```
# Melt the data from wide format to long format
retail_data_long = retail_data.melt(id_vars=['YEAR'], var_name='Month',␣
 ↪value_name='Sales')

# Create a proper datetime column by combining the year and month
retail_data_long['Date'] = pd.to_datetime(retail_data_long['YEAR'].astype(str)␣
 ↪+ '-' + retail_data_long['Month'], format='%Y-%b')

# Drop unnecessary columns (we no longer need the 'YEAR' and 'Month' columns␣
 ↪separately)
retail_data_long = retail_data_long[['Date', 'Sales']]

# Sort by date to ensure time-series order
retail_data_long = retail_data_long.sort_values(by='Date').
 ↪reset_index(drop=True)

# Display the first few rows to ensure the transformation is correct
retail_data_long.head()
```

[5]:
```
        Date     Sales
0 1992-01-01  146925.0
1 1992-02-01  147223.0
2 1992-03-01  146805.0
3 1992-04-01  148032.0
4 1992-05-01  149010.0
```

Sorting: It's essential to sort the data by date to ensure that the time series is in the correct chronological order.

1.3 Plot the Data

[7]:
```
# Plot the retail sales over time
plt.figure(figsize=(12, 6))
plt.plot(retail_data_long['Date'], retail_data_long['Sales'], label='Monthly␣
 ↪Retail Sales', color='b')
plt.title('US Monthly Retail Sales (1992-2021)', fontsize=16)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Sales ($)', fontsize=14)
plt.grid(True)
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



1.3.1 Key Observations:

Upward Trend (1992-2007):

- From 1992 to around 2007, there is a clear upward trend in retail sales, showing steady growth over time. This reflects general economic growth and increased consumer spending during this period.

2008 Financial Crisis Impact:

- Around 2008, there is a noticeable dip in retail sales, which aligns with the global financial crisis. This sharp decline shows a reduction in consumer spending during the economic downturn.
- Retail sales recover after the crisis, but the dip highlights the impact of macroeconomic conditions on retail activity.

Post-2010 Recovery:

- After 2010, the retail sales growth resumes, indicating economic recovery. The trend continues upwards, showing consistent growth through the 2010s.

COVID-19 Impact (2020):

- In 2020, there is a sharp and significant dip, likely due to the COVID-19 pandemic. During the early months of the pandemic (March-April 2020), retail sales plummeted as stores closed and consumer spending decreased.

- However, retail sales quickly rebounded by the middle of 2020, and there is a noticeable surge towards the end of 2020 and early 2021. This reflects changes in consumer behavior, such as a shift to online shopping and increased government stimulus programs boosting consumer spending.

Recent Spike (2021):

- The sharp increase in retail sales in early 2021 suggests a surge in consumer spending post-pandemic, driven by economic recovery, reopening of businesses, and possibly additional government stimulus.

Summary:

The plot shows steady growth in retail sales over the past few decades, with two significant disruptions:

- The 2008 financial crisis.
- The COVID-19 pandemic in 2020.

Both disruptions were followed by periods of recovery and increased spending, with a particularly strong rebound in 2021.

1.4 Split between test and train data

[8]:
```python
# Define the start and end date for the test set (July 2020 - June 2021)
test_start_date = '2020-07-01'
test_end_date = '2021-06-30'

# Split into training and testing sets
train_data = retail_data_long[retail_data_long['Date'] < test_start_date]
test_data = retail_data_long[(retail_data_long['Date'] >= test_start_date) &␣
 ↪(retail_data_long['Date'] <= test_end_date)]

# Display the sizes of the training and test sets
print(f"Training set size: {len(train_data)} records")
print(f"Test set size: {len(test_data)} records")

# Optional: Display the first few rows of the training and test sets
print(train_data.head())
print(test_data.head())
```

```
Training set size: 342 records
Test set size: 12 records
        Date     Sales
0 1992-01-01  146925.0
1 1992-02-01  147223.0
2 1992-03-01  146805.0
3 1992-04-01  148032.0
4 1992-05-01  149010.0
          Date     Sales
```

```
342 2020-07-01  481627.0
343 2020-08-01  483716.0
344 2020-09-01  493327.0
345 2020-10-01  493991.0
346 2020-11-01  488652.0
```

1.5 Build a Predictive Model

1.5.1 ARIMA model-

[12]:
```
# Fit an ARIMA model to the training data
# Order parameter (p, d, q) can be tuned as needed (we'll start with␣
 ↪ARIMA(5,1,0) as an example)
model = ARIMA(train_data['Sales'], order=(5, 1, 0))  # ARIMA(p=5, d=1, q=0)
model_fit = model.fit()

# Summary of the model
print(model_fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Sales   No. Observations:                  342
Model:                 ARIMA(5, 1, 0)   Log Likelihood               -3442.629
Date:                Sat, 21 Sep 2024   AIC                           6897.257
Time:                        13:49:14   BIC                           6920.249
Sample:                             0   HQIC                          6906.417
                                - 342
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.0087      0.009     -0.952      0.341      -0.027       0.009
ar.L2         -0.1073      0.015     -7.207      0.000      -0.136      -0.078
ar.L3         -0.0159      0.046     -0.343      0.732      -0.107       0.075
ar.L4          0.0192      0.100      0.192      0.848      -0.177       0.215
ar.L5          0.0114      0.131      0.087      0.931      -0.245       0.268
sigma2      3.125e+07   1.91e-08   1.64e+15      0.000    3.13e+07    3.13e+07
================================================================================
===
Ljung-Box (L1) (Q):                   0.66   Jarque-Bera (JB):
54142.55
Prob(Q):                              0.42   Prob(JB):
0.00
Heteroskedasticity (H):              14.36   Skew:
0.66
Prob(H) (two-sided):                  0.00   Kurtosis:
64.72
================================================================================
===
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
[2] Covariance matrix is singular or near-singular, with condition number
3.04e+29. Standard errors may be unstable.
```

1.5.2 Key Parts of the Output:

Model Summary:

- **Dep. Variable (Sales):** The dependent variable being modeled is "Sales."
- **No. Observations:** The number of observations used for modeling (342 months of data).
- **Model:** ARIMA(5, 1, 0) – this means the model uses 5 autoregressive terms (AR), 1 differencing term (I), and 0 moving average terms (MA).
- **Log Likelihood:** -3442.629. This is a measure of the model fit, but not easily interpretable on its own.
- **AIC (Akaike Information Criterion):** 6897.257. Lower values of AIC indicate a better fit, but AIC values are only meaningful when compared across models.
- **BIC (Bayesian Information Criterion):** 6920.249. Similar to AIC, BIC penalizes models for complexity (more parameters). Lower values are better.

AR Coefficients (AR.L1, AR.L2, AR.L3, etc.):

These are the autoregressive terms (lags). For ARIMA(5, 1, 0), five lags (L1 to L5) are used. The coefficients represent the contribution of each lagged value to the prediction.

Significant Terms:

- **ar.L2:** Has a coefficient of -0.1073 with a p-value of 0.000 (less than 0.05), meaning it's statistically significant.
- Other lags (L1, L3, L4, L5) are not statistically significant (p-values > 0.05), which suggests they do not strongly contribute to the model's predictive power.

Sigma^2 (Residual Variance):

- sigma^2 represents the variance of the residuals (errors) of the model. A large value (3.125e+07) suggests high variability in the data that the model has not captured well.

Diagnostic Tests:

- **Ljung-Box Test (Q):** Used to test for autocorrelation in the residuals. A p-value of 0.42 (greater than 0.05) suggests that there's no significant autocorrelation left in the residuals, which is a good sign for the model.
- **Jarque-Bera Test (JB):** Tests whether the residuals are normally distributed. A p-value of 0.00 (less than 0.05) suggests that the residuals are not normally distributed.

- **Heteroskedasticity Test (H):** Tests for constant variance of the residuals. A value of 14.36 with a p-value of 0.00 indicates heteroskedasticity, meaning the residuals' variance changes over time.

1.6 Key Observations:

Significant Terms:

- The second autoregressive lag (ar.L2) is the only significant term in the model with a p-value less than 0.05, indicating it is contributing to the model's predictions.
- The other autoregressive terms (L1, L3, L4, L5) are not statistically significant and may not be adding much predictive power.

Model Fit Issues:

- The warnings indicate that the covariance matrix is near-singular, suggesting potential instability in the parameter estimates. This can make standard errors unreliable.
- The Jarque-Bera test indicates that the residuals are not normally distributed, which may indicate model misspecification.
- The heteroskedasticity test suggests the residuals' variance is not constant over time (i.e., the data may exhibit periods of volatility that the model is not accounting for).

Model Performance (AIC/BIC):

- The AIC and BIC values can be used to compare different models. While lower values suggest better fit, we would need to compare this model's AIC and BIC to other models (e.g., ARIMA with different parameters) to know if this is optimal.

1.6.1 Predictions on Test set -

[21]:
```
# Forecast for the test set (we use the length of the test set for forecasting)
forecast = model_fit.forecast(steps=len(test_data))

# Add the forecasted values to the test data
test_data.loc[:, 'Predicted_Sales'] = forecast.values

# Display the test data with actual and predicted sales
test_data[['Date', 'Sales', 'Predicted_Sales']].head()
```

[21]:
```
           Date     Sales  Predicted_Sales
342  2020-07-01  481627.0    469487.694410
343  2020-08-01  483716.0    463787.997065
344  2020-09-01  493327.0    464692.971557
345  2020-10-01  493991.0    466750.423558
346  2020-11-01  488652.0    466955.277802
```

1.6.2 Observations:

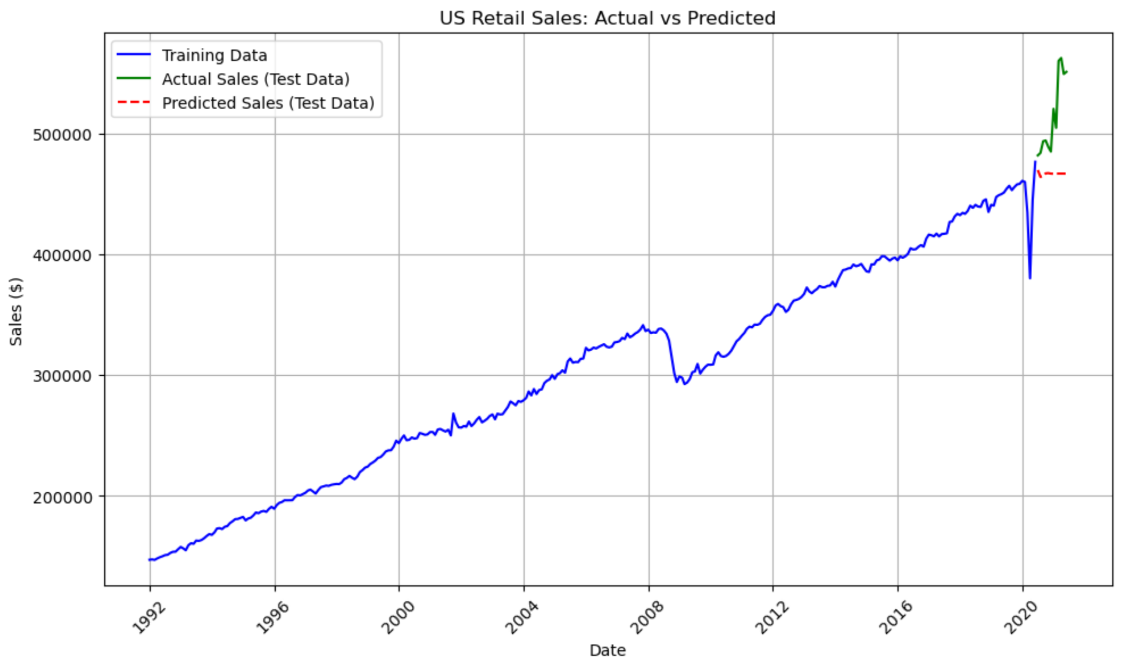
Actual vs Predicted Sales:

- The actual sales values for each month are slightly higher than the predicted sales values. However, the predictions are fairly close, indicating that the model is doing a decent job of capturing the general trend of the data.

Trend Comparison:

- The model captures the overall direction and trend of sales. For instance, the predicted sales increase between September and October, which aligns with the actual sales data.
- There is still a gap between the actual and predicted values, but the difference does not seem overly large given the complexity of retail sales forecasting.

[15]:
```
# Plot the actual vs predicted sales for the test set
plt.figure(figsize=(10, 6))
plt.plot(train_data['Date'], train_data['Sales'], label='Training Data',␣
  ↪color='blue')
plt.plot(test_data['Date'], test_data['Sales'], label='Actual Sales (Test␣
  ↪Data)', color='green')
plt.plot(test_data['Date'], test_data['Predicted_Sales'], label='Predicted␣
  ↪Sales (Test Data)', color='red', linestyle='--')
plt.title('US Retail Sales: Actual vs Predicted')
plt.xlabel('Date')
plt.ylabel('Sales ($)')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



This plot displays the comparison between actual retail sales and predicted sales based on your ARIMA model for the test period (July 2020 to June 2021):

Key Elements:

- **Blue Line:** Represents the training data (US retail sales from 1992 to June 2020), which was used to build the predictive model.
- **Green Line:** Represents the actual retail sales data for the test period (July 2020 to June 2021).
- **Red Dashed Line:** Represents the predicted sales generated by the ARIMA model for the test period (July 2020 to June 2021).

Observations:

Predicted vs Actual:

- The red dashed line shows the ARIMA model's predictions for retail sales, while the green line shows the actual values.
- There is a clear difference between the actual sales (which exhibit a sharp upward trend) and the predicted values, where the model underestimates the rise in retail sales during the test period.

Sharp Increase in Actual Sales:

- The actual retail sales (green line) show a significant and sharp increase, which might be due to post-COVID-19 economic recovery, increased consumer spending, and other factors that were difficult for the model to predict.
- The predicted values show more stability, which suggests that the ARIMA model didn't fully capture the sudden recovery and growth in retail sales during this period.

Model's Limitations:

- The red dashed line (predictions) is relatively flat compared to the steep rise in the actual data, indicating that the model is not capturing the recent dynamics in the data effectively.
- This could be due to the model being trained on historical data that did not contain such sharp, rapid increases, as seen post-2020.

1.6.3 Conclusion:

- The ARIMA model provides a reasonable baseline prediction for the test period but struggles to account for the sharp rise in sales observed in the actual data.
- This suggests that the model may need further tuning (e.g., trying a different order of ARIMA, considering external factors, or using a more sophisticated model) to capture these dynamics.
- Given the mismatch between the actual and predicted values, calculating the Root Mean Squared Error (RMSE) will help quantify how well the model performed.

1.7 Calculate the RMSE

[17]:
```python
# Calculate RMSE between actual and predicted sales in the test set
rmse = np.sqrt(mean_squared_error(test_data['Sales'],␣
 ↪test_data['Predicted_Sales']))

# Display the RMSE
print(f'Root Mean Squared Error (RMSE): {rmse}')
```

```
Root Mean Squared Error (RMSE): 57005.613882539
```

The ARIMA model provides predictions with an RMSE of approximately 57,000, which suggests that the model captures the general trend but underestimates the rapid rise in retail sales during the test period.

1.8 Referneces -

Dataset:

- US Retail Sales Data (1992-2021). Provided for the assignment or sourced from internal/external data repositories.

Time Series Analysis References:

- ARIMA Modeling: Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). Time Series Analysis: Forecasting and Control. John Wiley & Sons.
- Seasonal ARIMA (SARIMAX): Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice. OTexts. Online

[ ]:
```

```