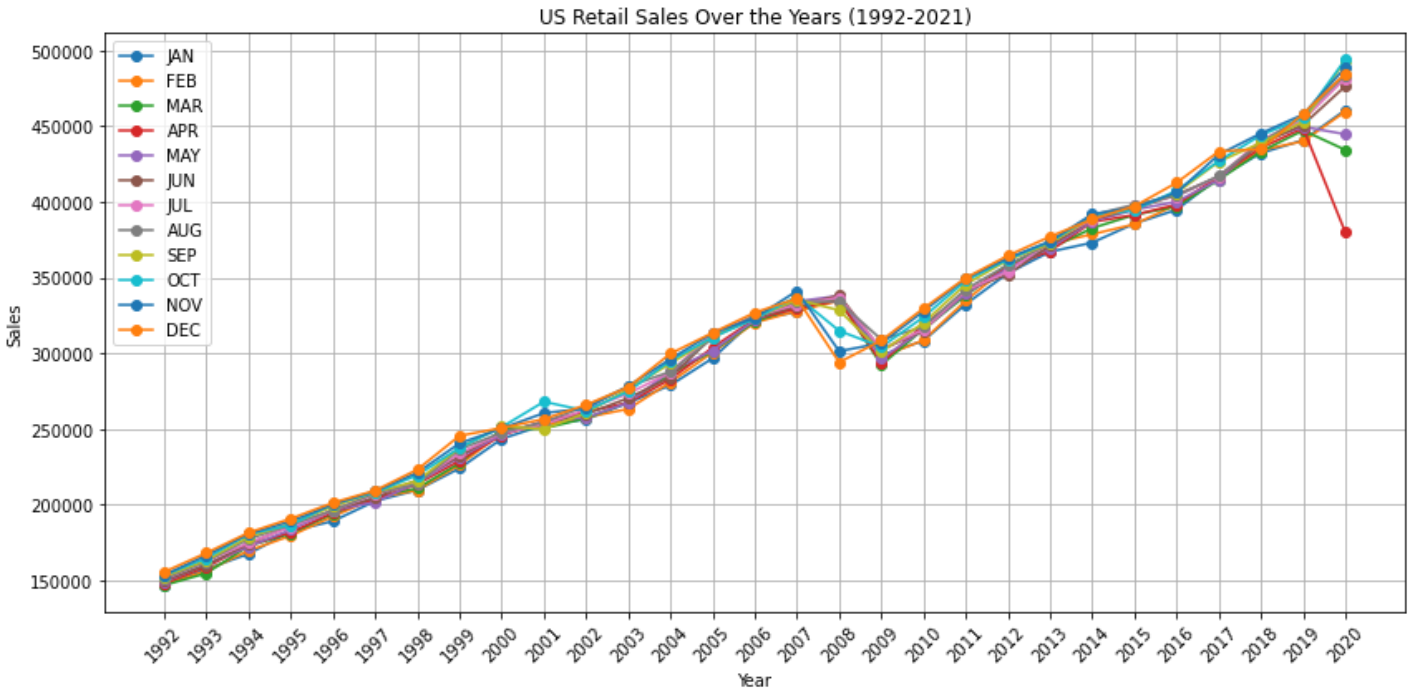
In [1]:

```
# William Barker
# DSC630
# Week 8

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import numpy as np

# Load the dataset
data = pd.read_csv('us_retail_sales.csv')

# Drop rows with NaN values
data = data.dropna()

# Create a copy of the data for plotting
data_copy = data.copy()
# Set the 'YEAR' column as the index
data_copy.set_index('YEAR', inplace=True)

# Plot the data
plt.figure(figsize=(12, 6))

# Loop through the columns (months) and plot each one
for column in data_copy.columns:
    plt.plot(data_copy.index, data_copy[column], label=column, marker='o')

plt.title('US Retail Sales Over the Years (1992-2021)')
plt.xlabel('Year')
plt.ylabel('Sales')
plt.legend()
plt.grid(True)
plt.xticks(data_copy.index, rotation=45)  # Set x-axis ticks to the years
plt.tight_layout()

# Show the plot
plt.show()
```



In [4]:

```
# Extract features (months) and target (sales values)
X_train = data.drop(['YEAR'], axis=1).values  # Use .values to get a NumPy array
```

```
y_train = data.iloc[:, 1:].values  # No need to flatten
```

In [5]:

```
# Initialize and train the model (linear regression)
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[5]:

▾ LinearRegression

LinearRegression()

In [7]:

```
# Make predictions for 2020 and 2021
X_test = data.drop(['YEAR'], axis=1).values
predictions = model.predict(X_test)
```

In [8]:

```
from sklearn.metrics import mean_squared_error
import numpy as np
# Calculate RMSE for 2020
actual_2020 = data[data['YEAR'] == 2020].iloc[:, 1:].values.flatten()
predicted_2020 = predictions[0]
rmse_2020 = np.sqrt(mean_squared_error(actual_2020, predicted_2020))
print(f"RMSE for 2020: {rmse_2020}")
```

```
RMSE for 2020: 316117.83648388146
```

In [ ]: