coffee-sales-analysis

October 14, 2024

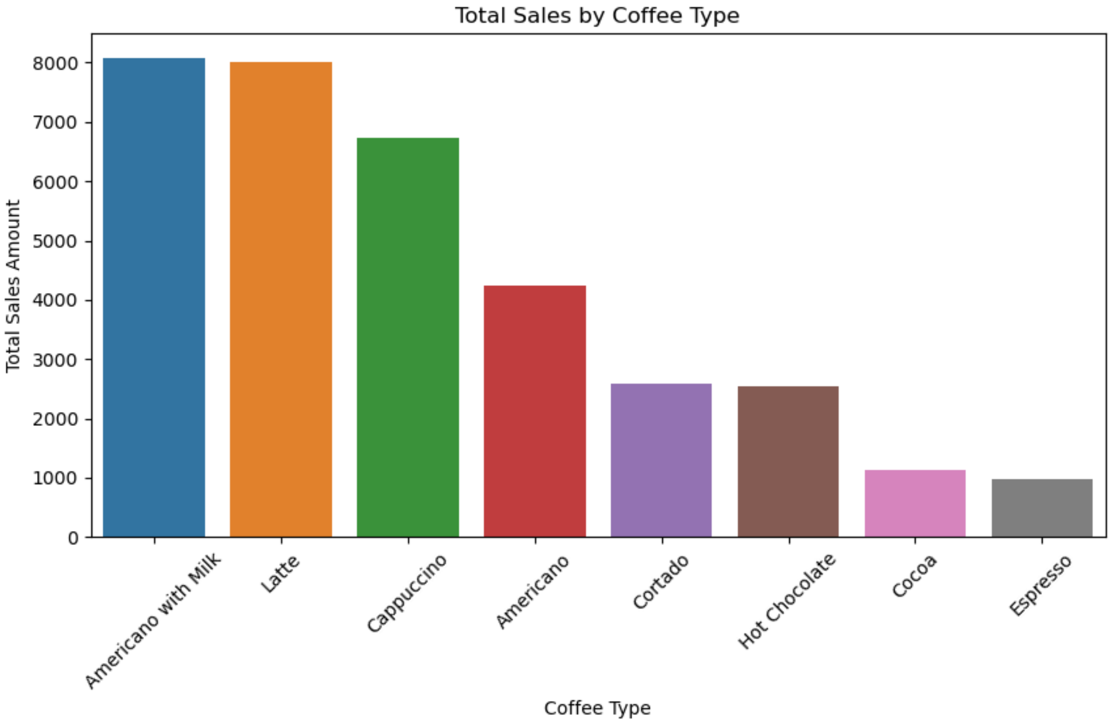
[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

[2]:
```
# Step 1: Load and clean the data
data = pd.read_csv('index.csv')
```

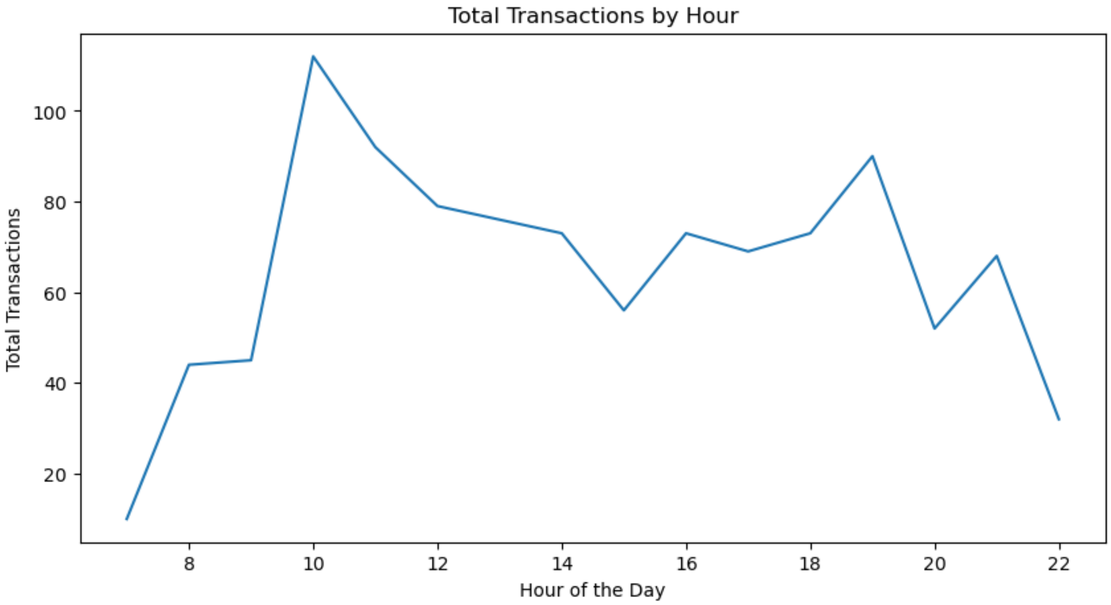
[3]:
```
# Convert 'date' to datetime (correct format)
data['date'] = pd.to_datetime(data['date'], format='%Y-%m-%d')
data['datetime'] = pd.to_datetime(data['datetime'])  # Ensure datetime is also␣
 ↪in the right format

# Drop duplicates
data.drop_duplicates(inplace=True)

# Check for missing values
print("Missing values in each column:")
print(data.isnull().sum())
data.dropna(inplace=True)  # Drop rows with missing values if any

# Step 2: Conduct Exploratory Data Analysis (EDA)
# Total sales by coffee type
sales_by_coffee = data.groupby('coffee_name')['money'].sum().
 ↪sort_values(ascending=False)
plt.figure(figsize=(10, 5))
sns.barplot(x=sales_by_coffee.index, y=sales_by_coffee.values)
plt.title('Total Sales by Coffee Type')
plt.xticks(rotation=45)
plt.ylabel('Total Sales Amount')
plt.xlabel('Coffee Type')
plt.show()

# Peak hour analysis
data['hour'] = data['datetime'].dt.hour
```

```
peak_hours = data.groupby('hour').size()
plt.figure(figsize=(10, 5))
sns.lineplot(x=peak_hours.index, y=peak_hours.values)
plt.title('Total Transactions by Hour')
plt.ylabel('Total Transactions')
plt.xlabel('Hour of the Day')
plt.show()

# Step 3: Prepare data for machine learning
X = data[['hour', 'coffee_name']]  # Features
y = data['money']  # Target

# One-hot encode categorical variables
X = pd.get_dummies(X, columns=['coffee_name'], drop_first=True)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Step 4: Train a simple linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Step 5: Evaluate the model's performance
y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse:.2f}')
print(f'R² Score: {r2:.2f}')
```

```
Missing values in each column:
date            0
datetime        0
cash_type       0
card           89
money           0
coffee_name     0
dtype: int64
```



```
C:\Users\RITESH PATIL\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\RITESH PATIL\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



```
Mean Squared Error: 4.74
R² Score: 0.80
```

[ ]: