Exercise 4

In a retail experiment, we want to understand how advertising expenditure, store location, and competition affect sales revenue. Using synthetic data, implement multiple linear regression in Python to analyse these factors. Interpret the coefficients, perform an F-test to assess overall model significance, and conduct t-tests to evaluate the significance of individual coefficients.

In [31]:
```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

In [32]:
```
df = pd.read_csv("sales.csv")
df.head()
```

Out[32]:

	AdvertisingExpenditure	StoreLocation	Competition	SalesRevenue
0	4269	1	1.509	16259
1	4441	1	1.285	18432
2	1866	0	1.018	9630
3	3871	0	1.116	14029
4	4760	1	1.015	18392

In [33]:
```
X = df[["AdvertisingExpenditure", "Competition", "StoreLocation"]]
Y = df["SalesRevenue"]
```

In [34]:
```
model = LinearRegression()

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25,
random_state=42)

model.fit(X_train, Y_train)
```

Out[34]:

```
▾ LinearRegression  i ?

LinearRegression()
```

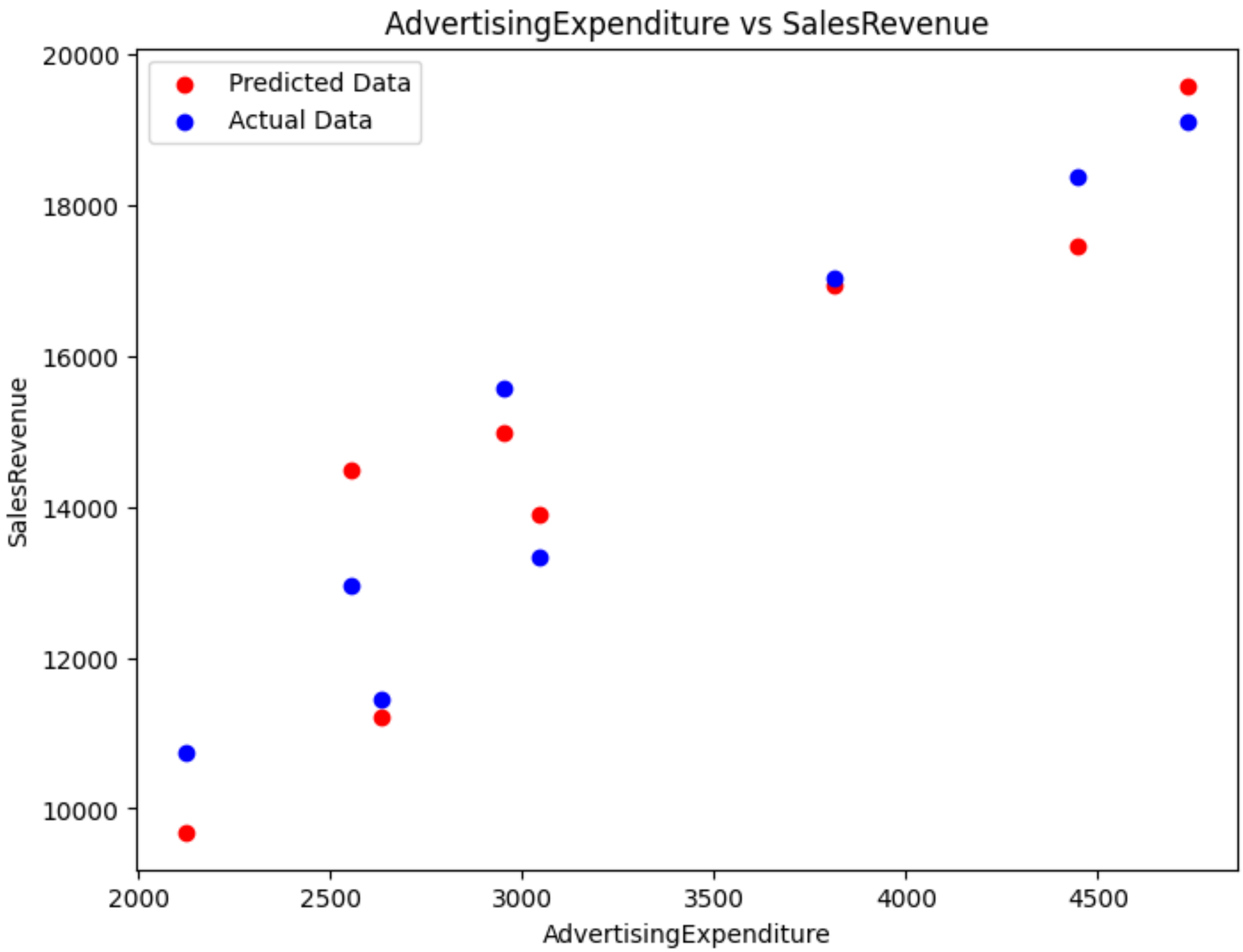
In [35]:
```
coefficients = model.coef_
intercept = model.intercept_
print(f"Coefficient = {coefficients}\nIntercept = {intercept}")
```

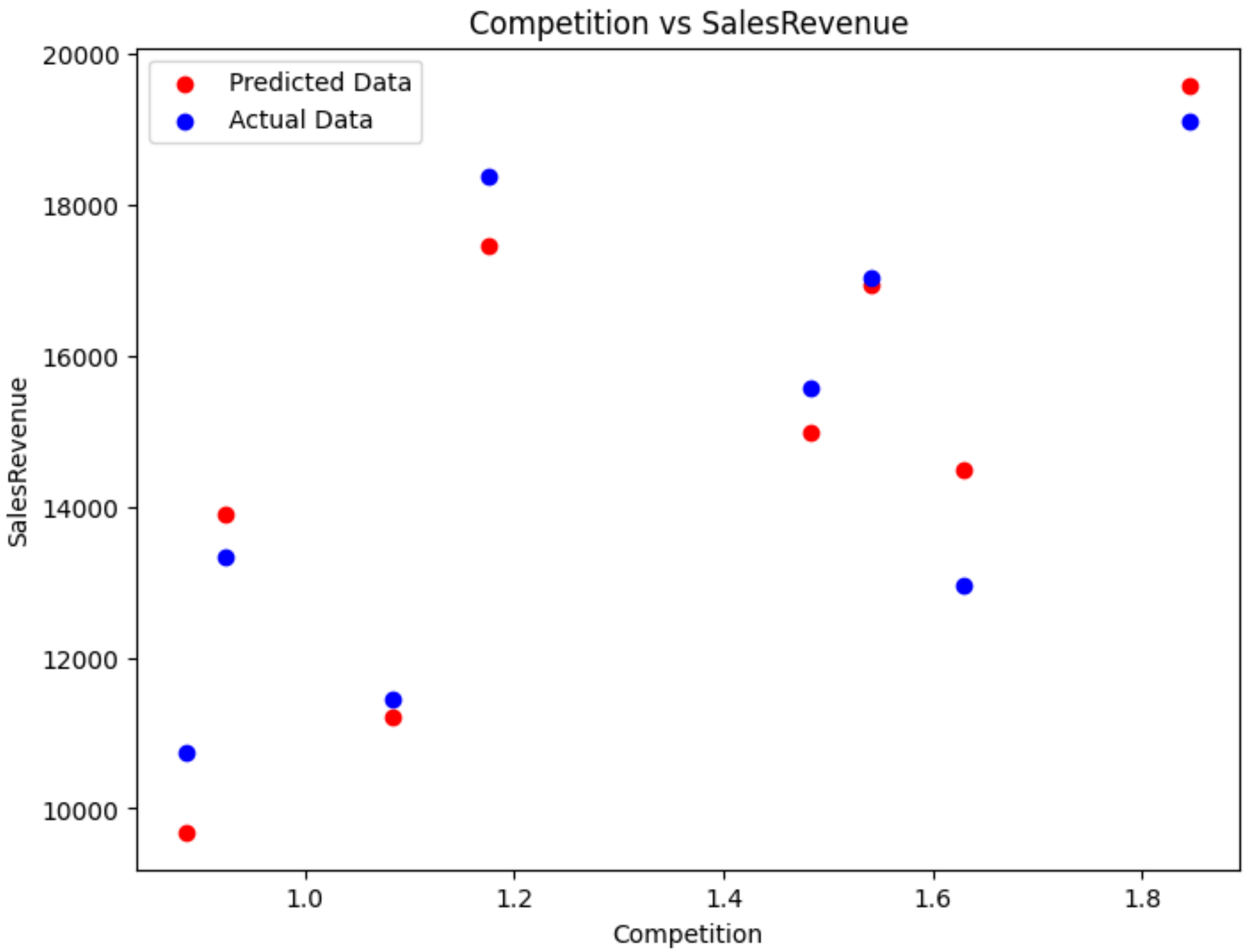
```
Coefficient = [2.11493691e+00 2.27274333e+03 2.19396228e+03]
Intercept = 3176.7913667305384
```

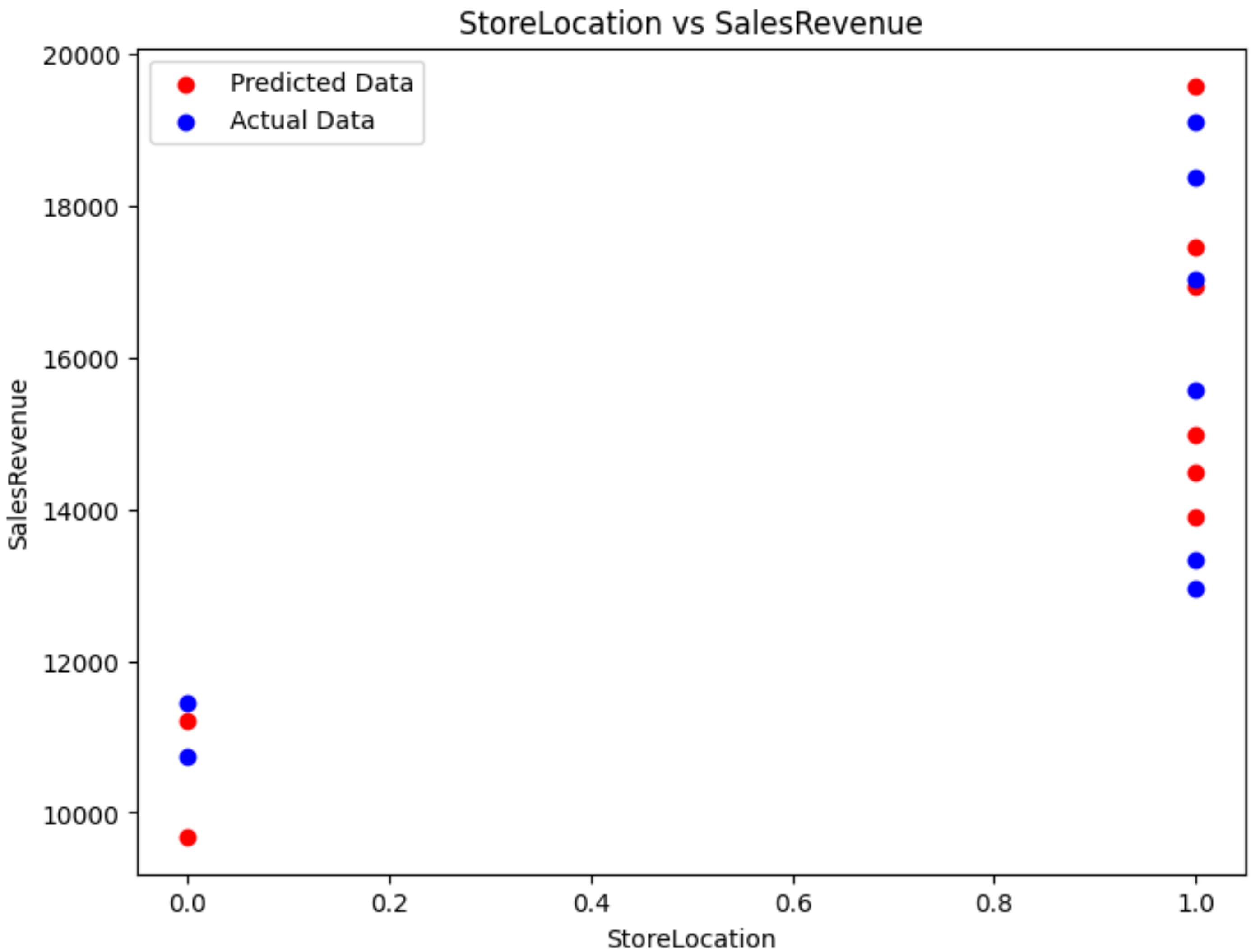
In [36]:
```
Y_pred = model.predict(X_test)
```

In [42]:
```
predictors = ["AdvertisingExpenditure", "Competition", "StoreLocation"]

for predictor in predictors:
    plt.figure(figsize=(8, 6))
    plt.title(f"{predictor} vs SalesRevenue")
    plt.xlabel(predictor)
    plt.ylabel("SalesRevenue")
    plt.scatter(X_test[predictor], Y_pred, color="r", label="Predicted Data")
    plt.scatter(X_test[predictor], Y_test, color="b", label="Actual Data")
    plt.legend()
```




Competition vs SalesRevenue
Predicted Data
Actual Data
SalesRevenue
Competition
20000
18000
16000
14000
12000
10000
1.0
1.2
1.4
1.6
1.8



In [38]:
```
import statsmodels.api as sm
```

In [39]:
```
X_with_const = sm.add_constant(X)
```

In [40]:
```
model = sm.OLS(Y, X_with_const).fit()

for predictor in predictors:
    t_statistic = model.tvalues[predictor]
    p_value_t = model.pvalues[predictor]

    print(f"t-statistic for {predictor} = {t_statistic}")

    if p_value_t < 0.05:
        print(f"{predictor} is a statistically significant predictor of SalesRevenue.")
    else:
        print(f"{predictor} is NOT a statistically significant predictor of
SalesRevenue.")
```

```
t-statistic for AdvertisingExpenditure = 12.738460146150278
AdvertisingExpenditure is a statistically significant predictor of SalesRevenue.
t-statistic for Competition = 5.350557857468894
Competition is a statistically significant predictor of SalesRevenue.
t-statistic for StoreLocation = 4.899145856634402
StoreLocation is a statistically significant predictor of SalesRevenue.
```

In [41]:

```
X_with_const = sm.add_constant(X[predictor])
model = sm.OLS(Y, X_with_const).fit()

f_statistic = model.fvalue
p_value_f = model.f_pvalue

print(f"F-statistic for {predictor} = {f_statistic}")

if p_value_f < 0.05:
    print(f"{predictor} is a statistically significant predictor of SalesRevenue.")
else:
    print(f"{predictor} is NOT a statistically significant predictor of SalesRevenue.")
```

```
F-statistic for StoreLocation = 44.458964675049536
StoreLocation is a statistically significant predictor of SalesRevenue.
```