1) Let us start by exploring the data

```
import pandas as pd
# 1. Load the data
df = pd.read_csv("fulton-1.csv")

# 2. Check data structure and head
print(df.head())
```

```
   Mon  Tue  Wed  Thu  Date  Stormy         p         q
0    1    0    0    0     1       1  1.569217  8.994421
1    0    1    0    0     2       1  2.000000  7.707063
2    0    0    1    0     3       0  2.072321  8.350194
3    0    0    0    1     4       1  2.247139  8.656955
4    0    0    0    0     5       1  2.664327  7.844241
```

a) First, create two scatterplots – First one with putting prices (p) on the y axis and the Date on x-axis, Second with the quantity (q) on the y axis and the Date on x-axis. What do you conclude regarding variation in prices?

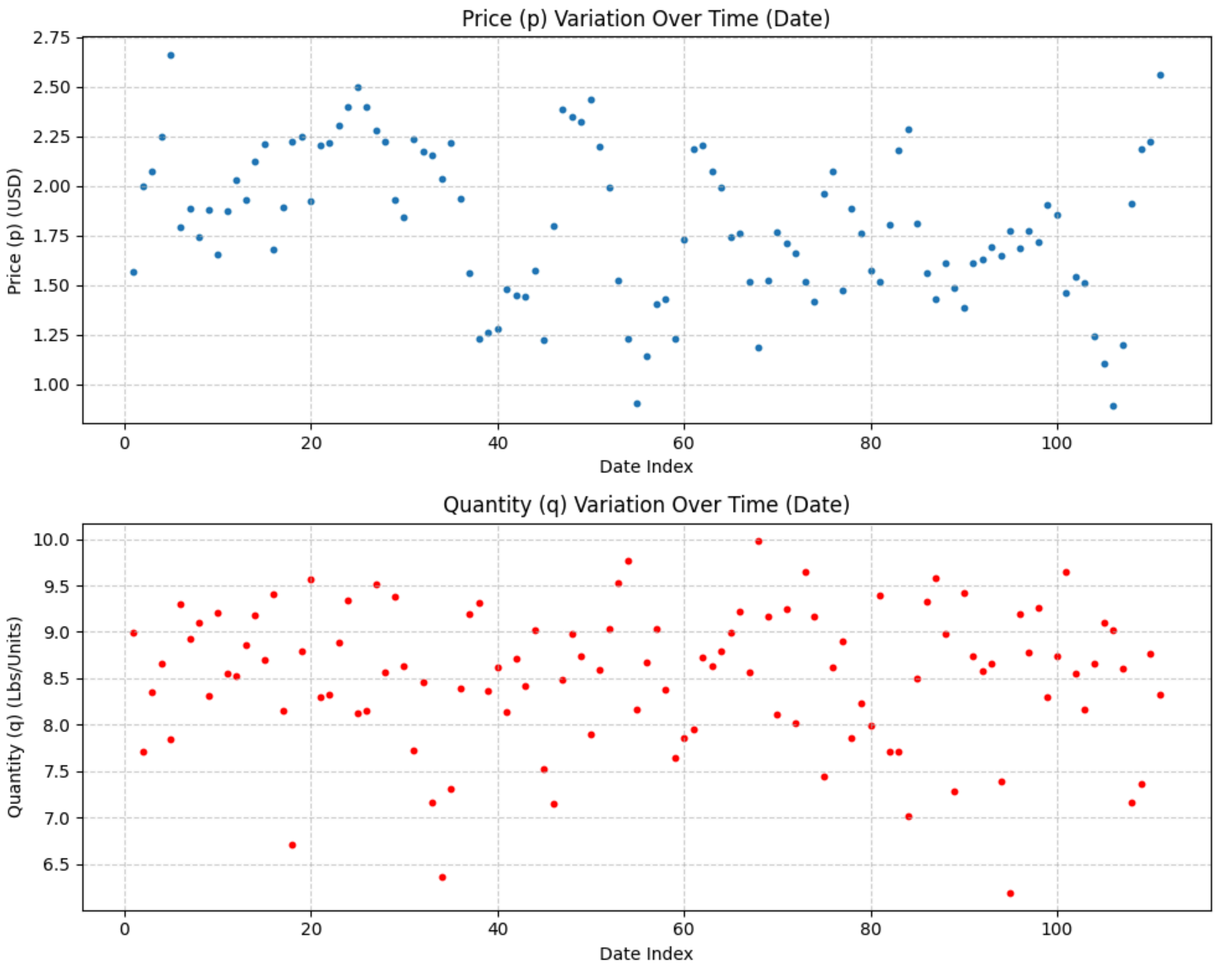
```
import matplotlib.pyplot as plt

# 3. Create the two scatterplots
fig, axes = plt.subplots(2, 1, figsize=(10, 8))

# Plot 1: Price (p) vs. Date
axes[0].scatter(df['Date'], df['p'], s=10)
axes[0].set_title('Price (p) Variation Over Time (Date)')
axes[0].set_xlabel('Date Index')
axes[0].set_ylabel('Price (p) (USD)')
axes[0].grid(True, linestyle='--', alpha=0.6)
# Plot 2: Quantity (q) vs. Date
axes[1].scatter(df['Date'], df['q'], s=10, color='red')
axes[1].set_title('Quantity (q) Variation Over Time (Date)')
axes[1].set_xlabel('Date Index')
axes[1].set_ylabel('Quantity (q) (Lbs/Units)')
axes[1].grid(True, linestyle='--', alpha=0.6)

plt.tight_layout()

# Save the combined figure
plt.savefig('price_quantity_variation_over_time.png')
plt.show()
```



ANS:

Conclusion Regarding Variation in Prices:

1. Prices exhibit a high degree of day-to-day volatility. There is no obvious long-term trend, but rather a wide spread of prices observed throughout the data collection period (Date index 1 to 111).

2. Prices fluctuate significantly, ranging roughly from $1.00 to over 3.50 per unit.
3. The price shows no stable seasonal or long-term trend. Instead, it moves up and down quickly in a way that looks random

b) What is the average price (p) and quantity (q) of fish sold on each weekday (Mon-Thu)? On which day are fish sold for highest price? Can you provide any potential explanation for that.

```
day_cols = ['Mon', 'Tue', 'Wed', 'Thu']

# 2. Create the categorical 'Day' column using idxmax(axis=1)
# This finds the column name (Mon, Tue, etc.) where the value is 1 for each row
df['Day'] = df[day_cols].idxmax(axis=1)

# 3. Use groupby() to calculate the average price (p) and quantity (q) for each
results_df = df.groupby('Day')[['p', 'q']].mean().rename(
    columns={'p': 'Avg_Price_p', 'q': 'Avg_Quantity_q'}
).reindex(day_cols) # Reindex ensures the output order is Mon, Tue, Wed, Thu

results_df = pd.DataFrame(results_df).T

print("--- Average Price and Quantity by Weekday ---")
print(results_df.round(4).to_markdown())


# 4. Determine the day with the highest average price
highest_price_day = results_df.loc['Avg_Price_p'].idxmax()
highest_price = results_df.loc['Avg_Price_p'].max()



print(f"\n The day with the highest average price is {highest_price_day}, with
```

```
--- Average Price and Quantity by Weekday ---
|                |    Mon |    Tue |    Wed |    Thu |
|:---------------|-------:|-------:|-------:|-------:|
| Avg_Price_p    | 1.7846 | 1.7926 | 1.7989 | 1.8683 |
| Avg_Quantity_q | 8.7349 | 8.2073 | 8.1646 | 8.7626 |

 The day with the highest average price is Thu, with an average price of $1.8683
```

Higest average price on Thursdays can be explained by a typical market mechanism where demand increases at the end of the week, while supply remains constrained (or is even reduced due to the weekly cycle):

Demand-Side Pressure (Weekend Rush): Demand for fish surges on Thursday as buyers (restaurants and retailers) make their final weekly purchase to stock up for anticipated higher consumer consumption over the upcoming weekend. This "last-chance" inelastic demand allows sellers to charge a premium.

Supply-Side Constraint (Inelasticity): The market's supply of fresh fish is determined by catches made days earlier, making the quantity available on Thursday fixed (inelastic). Since supply cannot easily increase to meet the sudden surge in demand, the equilibrium price is pushed to its highest point of the weekemains constrained (or is even reduced due to the weekly cycle).

2) Build a linear regression model to predict quantity (q), using all of the other variables (except the “Stormy” and “Date” variable).

```
import statsmodels.formula.api as smf

# Define the model formula
formula = 'q ~ p + Mon + Tue + Wed + Thu'

# Fit the OLS model using smf.ols()
ols_model_smf = smf.ols(formula=formula, data=df).fit()

# 4. Print the regression summary
print(ols_model_smf.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      q   R-squared:                       0.220
Model:                            OLS   Adj. R-squared:                  0.183
Method:                 Least Squares   F-statistic:                     5.940
Date:                Sun, 07 Dec 2025   Prob (F-statistic):           7.08e-05
Time:                        00:51:45   Log-Likelihood:                -110.00
No. Observations:                 111   AIC:                             232.0
Df Residuals:                     105   BIC:                             248.3
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      9.7320      0.336     28.936      0.000       9.065      10.399
p             -0.5625      0.168     -3.344      0.001      -0.896      -0.229
Mon            0.0143      0.203      0.071      0.944      -0.387       0.416
Tue           -0.5162      0.198     -2.611      0.010      -0.908      -0.124
Wed           -0.5554      0.202     -2.745      0.007      -0.957      -0.154
Thu            0.0816      0.198      0.413      0.681      -0.311       0.474
```

```
==============================================================================
Omnibus:                       14.325   Durbin-Watson:                   1.487
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               15.876
Skew:                          -0.804   Prob(JB):                     0.000357
Kurtosis:                       3.920   Cond. No.                         13.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

a) What is the linear regression equation produced by your model?

ANS:

q = 9.7320 - 0.5625 * p + 0.0143 * Mon - 0.5162 * Tue - 0.5554 * Wed + 0.0816 * Thu

b) Evaluate the quality of your model. What is the R^2 value? Which independent variables are significant?

ANS:

R^2 value : 0.220 (22% of R^2 suggests the model has a relatively poor fit. A large portion (approx 78%) of the variation in quantity remains unexplained by the model.)

Significant independent variables are: 'price', 'Tue' & 'Wed'

c) What is the coefficient of price (p)? Can we use this coefficient to estimate the effect of change in prices on quantity (q)? Explain your reasoning?

ANS:

Coffifiant of price is: - 0.5625

Eventhough, the regression equation shows the relation between quantity and price (a significant variable), but it has limitations:

1. This estimate is valid only if all other factors included in the model (Mon, Tue, Wed, Thu) remain unchanged. If other factors also change, the effect on quantity will be different.
2. While the coefficient itself might be significant, the overall explanatory power of the model (R-squared = 0.220) is low. This suggests that price, along with the day-of-week variables, only explains 22% of the variation in quantity. A large portion of the quantity's variation is still unexplained by this model. Therefore, while the coefficient provides an estimate, the model might not be robust enough for precise predictions in all scenarios.

3. The model assumes a linear relationship. If the true relationship is non-linear, this linear approximation might not be accurate, especially for large price changes.
4. To claim that a price change causes a quantity change, one needs to consider potential issues like endogeneity (where price and quantity might influence each other simultaneously, or an unobserved variable affects both).

3) Let us now look at estimating the causal effect of price (p) on quantity (q)

a) Graddy (1995) proposed the use of two-day lagged weather as an instrument for prices (p). Justify why this could be a good instrument. (Note an instrument needs to satisfy two properties.)

ANS:

The use of two-day lagged weather ("Stormy") as an instrumental variable (IV) for the price of whiting fish is justified because it satisfies the two core properties required of a valid instrument: Relevance and the Exclusion Restriction (Exogeneity).

1. Relevance (Correlation with the Endogenous Variable):

The weather two days ago directly impacts the supply of fish available today. A storm two days prior would have prevented fishing boats from going out, resulting in a significantly smaller catch. Since fish supply is a critical determinant of market price in a competitive setting, the reduced catch shifts the supply curve inward (upward) today, leading to an increase in today's equilibrium price (P).

2. Exclusion Restriction / Exogeneity (No Direct Correlation with the Error Term):

The purchasing decisions of fish buyers today are driven by factors like today's price, current consumer demand, and day-of-week preferences. The weather conditions from two days ago should have no direct, independent influence on a consumer's desire or need to buy fish today, other than the price set by the resulting supply shock.

By isolating a variable that acts as a pure supply shifter (affecting P) without directly affecting the demand curve (or Q), the IV ensures that the variation in price used for estimation is exogenous, thus correcting the simultaneity bias found in OLS

b) Is there any way in which this instrument could correlate directly with the quantity of fish sold on a day. If yes explain how?

ANS: The instrument could correlate directly with Q if the fish quality is affected by the storm, independent of the price:

1. Delayed Catch/Processing: A storm two days ago could also complicate or delay the landing, storage, or processing of the fish that were caught immediately before or after the storm.
2. Perceived Lower Quality: If the fish available for sale on Day-3 that originated from a storm-affected period Day-1 are noticeably less fresh or lower in quality than usual, can result into decrease in buyers' demand (Q) regardless of the price.

c) Now run an iv regression in python using IV2SLS function with the variable “Stormy” as an instrumental variable for price (p). Note, “Stormy” is set to 1 if it was stormy two days ago and is 0 otherwise.

```
!pip install linearmodels # install a new package
```

```
Requirement already satisfied: linearmodels in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: numpy<3,>=1.22.3 in /usr/local/lib/python3.12/dis
Requirement already satisfied: pandas>=1.4.0 in /usr/local/lib/python3.12/dist-p
Requirement already satisfied: scipy>=1.8.0 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: statsmodels>=0.13.0 in /usr/local/lib/python3.12/
Requirement already satisfied: mypy_extensions>=0.4 in /usr/local/lib/python3.12
Requirement already satisfied: pyhdfe>=0.1 in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: formulaic>=1.2.1 in /usr/local/lib/python3.12/dis
Requirement already satisfied: interface-meta>=1.2.0 in /usr/local/lib/python3.1
Requirement already satisfied: narwhals>=1.17 in /usr/local/lib/python3.12/dist-
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python
Requirement already satisfied: wrapt>=1.0 in /usr/local/lib/python3.12/dist-pack
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.12/dist
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packag
```

```
from linearmodels.iv import IV2SLS # instrumental variable 2 stage least square
#linear regression => ols (ordinary least squares)
formula = "q ~ 1 + Mon + Tue + Wed + Thu + [p ~ Stormy]"

iv_regression = IV2SLS.from_formula(formula = formula, data = df)
iv_regression_result = iv_regression.fit()
print(iv_regression_result.summary)
```

```
                          IV-2SLS Estimation Summary
==============================================================================
Dep. Variable:                      q   R-squared:                      0.1391
Estimator:                    IV-2SLS   Adj. R-squared:                 0.0981
No. Observations:                 111   F-statistic:                    24.946
```

```
Date:                  Sun, Dec 07 2025   P-value (F-stat)                0.0001
Time:                          00:54:35   Distribution:                  chi2(5)
Cov. Estimator:                  robust

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept      10.745     0.7860     13.671     0.0000      9.2043      12.285
Mon           -0.0254     0.2154    -0.1179     0.9061     -0.4475      0.3967
Tue           -0.5308     0.1966    -2.7004     0.0069     -0.9160     -0.1455
Wed           -0.5664     0.2013    -2.8141     0.0049     -0.9608     -0.1719
Thu            0.1093     0.1735     0.6296     0.5290     -0.2309      0.4494
p             -1.1194     0.4310    -2.5970     0.0094     -1.9643     -0.2746
==============================================================================

Endogenous: p
Instruments: Stormy
Robust Covariance (Heteroskedastic)
Debiased: False
```

d) In which direction did the coefficient of price (p) change? Is this expected? Explain

ANS:

The coefficient of price changed from -0.5625 to -1.1194. The coefficient became more negative (steeper/more elastic).

Yes, this direction of change is expected due to the presence of Simultaneity Bias (or endogeneity) in the OLS model.

Explaination:

1. Source of Bias: In the Fulton Fish Market, price (P) and quantity (Q) are set simultaneously. The OLS model suffers from reverse causality because the variation in observed prices is influenced by unobserved Supply Shocks (e.g., fishing success) and unobserved Demand Shocks (e.g., unexpected consumer behavior).
2. The Role of Supply: Supply shocks cause the supply curve to shift.1 When supply is high, P falls and Q rises. When supply is low, P rises and Q falls. These are the expected movements along the true, negatively sloped demand curve.
3. The Bias: However, unobserved Demand Shocks can shift the demand curve outward (upward), leading to higher observed P and higher observed Q. When the OLS regression averages all these points, the positive correlation induced by the shifting demand curve pulls the overall slope toward zero (less negative)($\mathbf{-0.5625}$), making the price effect appear artificially small (inelastic).

4. Correction by IV: The Instrumental Variable (Stormy) works by only isolating the price variation caused by supply shocks. This allows the IV regression to successfully "trace out" the true demand curve, revealing the true, steeper negative slope of $-\mathbf{1.1194}$.

Double-click (or enter) to edit

e) Write down the updated linear regression equation (as in 2 a) using the coefficients from the instrumental variable regression.

ANS:

q = 10.7447 - 1.1194 * p - 0.0254 * Mon - 0.5308 * Tue - 0.5664 * Wed + 0.1093 * Thu

4) Let us now look at calculating optimal fish price for each day of the week.

a) Assign the value “Mon” = 1 and 0 for the rest of the days. Set the value of p = 0.1. Calculate the quantity using the linear regression model from question 3e. Then report the revenue a seller would make at this price point on Monday.

```
p = 0.1
Mon = 1
Tue = 0
Wed = 0
Thu = 0
q = 10.7447 - 1.1194 * p - 0.0254 * Mon - 0.5308 * Tue - 0.5664 * Wed + 0.1093

revenue = p * q

print("Revenue a seller would make at price (0.1) point on Monday: ")
print(revenue)
```

```
Revenue a seller would make at price (0.1) point on Monday:
1.0607360000000001
```

b) Using np.arange, create a list of prices starting from 0, ending at 6, with an increment of 0.01. Calculate the revenue for each price value on the list

```
import numpy as np
# 1. Create the list of prices using np.arange
```

```
# Use 6.01 as the stop value to ensure 6.00 is included due to floating-point a
prices = np.arange(0, 6.01, 0.01)

# 2. Calculate the quantity (q) for each price value
q = 10.7447 - 1.1194 * prices - 0.0254 * Mon - 0.5308 * Tue - 0.5664 * Wed + 0.

# 3. Calculate the revenue for each price value
revenue = prices * q

results_df = pd.DataFrame({
    'Price (p)': prices,
    'Quantity (q)': q,
    'Revenue': revenue
})
```

c) Plot these prices and revenues. Create a scatter plot with revenue on the y-axis and the corresponding price on the x-axis. What price do you think sellers should set on a Monday based on this plot?

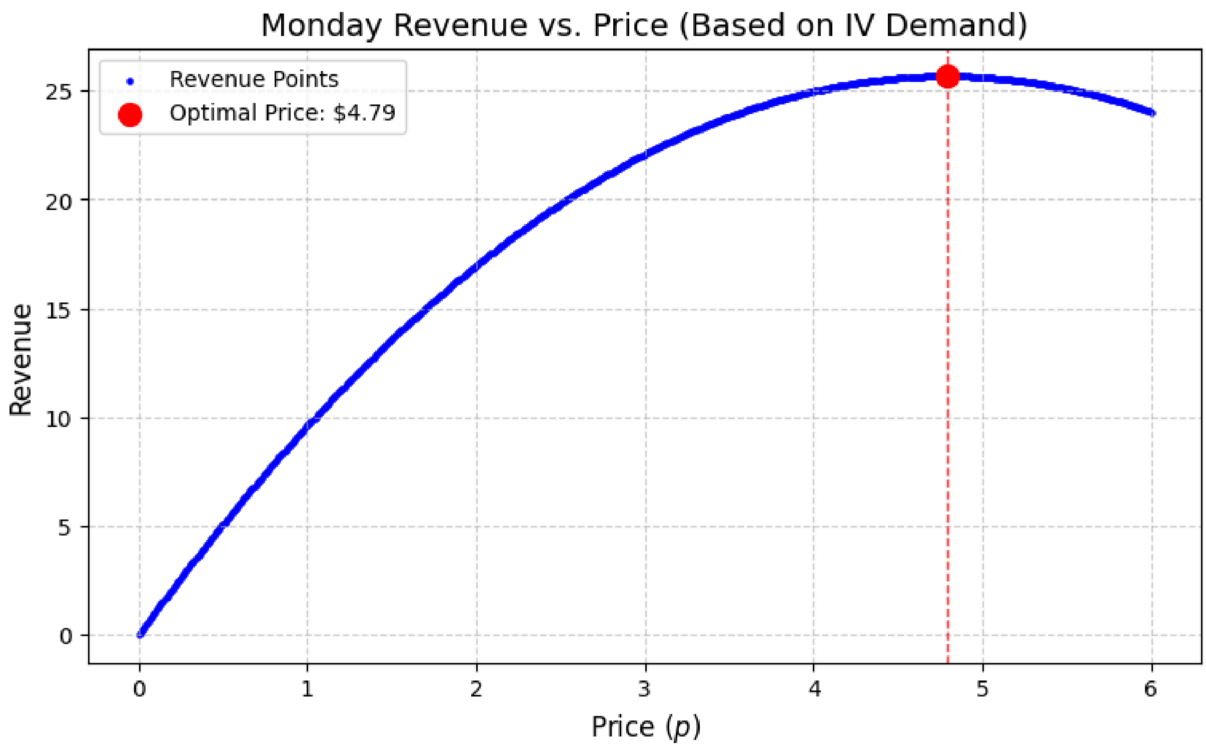
```
# 2. Find the optimal price (price that maximizes revenue)
max_revenue_row = results_df.loc[results_df['Revenue'].idxmax()]
optimal_price = max_revenue_row['Price (p)']
max_revenue = max_revenue_row['Revenue']


plt.figure(figsize=(9, 5))
plt.scatter(results_df['Price (p)'], results_df['Revenue'], s=5, color='blue', 

# Highlight the optimal price point
plt.scatter(optimal_price, max_revenue, color='red', s=100, zorder=5, label=f'C
plt.axvline(x=optimal_price, color='red', linestyle='--', linewidth=1, alpha=0.

# Add annotations
plt.title('Monday Revenue vs. Price (Based on IV Demand)', fontsize=14)
plt.xlabel('Price ($p$)', fontsize=12)
plt.ylabel('Revenue', fontsize=12)
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)

plt.savefig('monday_revenue_plot.png')
```



ANS:

Setting the price at $4.79 is expected to maximize revenue on Monday.

d) Repeat this exercise for other days (Tue, Wed, and Thu).

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# --- IV MODEL COEFFICIENTS (from question 3e) ---
CONST = 10.7447
P_COEF = -1.1194

# Day-specific coefficients
DAY_COEFS = {
    'Mon': -0.0254,
    'Tue': -0.5308,
    'Wed': -0.5664,
    'Thu': 0.1093
}
```

```
# --- 1. Generate Price Points ---
prices = np.arange(0, 6.01, 0.01)

# --- 2. Calculate Revenue and Find Maxima for All Days ---
optimal_results = []
all_rev_data = []

plt.figure(figsize=(10, 6))

for day, coef in DAY_COEFS.items():
    # Calculate daily intercept (where all other day dummies are 0)
    daily_intercept = CONST + coef

    # Calculate Quantity: q = Intercept_day + P_COEF * p
    quantities = daily_intercept + (P_COEF * prices)

    # Calculate Revenue: Revenue = p * q
    revenue = prices * quantities

    # Store for combined plot
    day_df = pd.DataFrame({'Price (p)': prices, 'Revenue': revenue, 'Day': da
    all_rev_data.append(day_df)

    # Find the optimal price and maximum revenue
    max_revenue_index = revenue.argmax()
    optimal_price = prices[max_revenue_index]
    max_revenue = revenue[max_revenue_index]

    optimal_results.append({
        'Day': day,
        'Optimal Price (p*)': optimal_price,
        'Max Revenue ($)': max_revenue
    })

    # --- 3. Plot the Revenue Curve for the Day ---
    plt.plot(prices, revenue, label=f'{day} Revenue', linewidth=2)
    plt.scatter(optimal_price, max_revenue, marker='o', s=50, color='black', 
    plt.annotate(f' ${optimal_price:.2f}', (optimal_price, max_revenue),
                 textcoords="offset points", xytext=(5, 5), fontsize=9)

# --- 4. Final Plot Formatting ---
plt.title('Daily Revenue Maximization (Based on IV Demand)', fontsize=14)
plt.xlabel('Price ($p$)', fontsize=12)
plt.ylabel('Revenue', fontsize=12)
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.xlim(0, 6.0)
plt.ylim(0, revenue.max() * 1.1)


# --- 5. Report Results ---
```

```
optimal_df = pd.DataFrame(optimal_results)
```

```
--- Daily Optimal Pricing Strategy ---
| Day   |   Optimal Price (p*) |   Max Revenue ($) |
|:------|---------------------:|------------------:|
| Mon   |               4.7900 |           25.6618 |
| Tue   |               4.5600 |           23.2990 |
| Wed   |               4.5500 |           23.1369 |
| Thu   |               4.8500 |           26.3108 |
```