Optimization I Project III - Newsvendor Problem

Jack Feen
Mihir Jeshurun Gandham
Radha Pawar
Ishrak Wasif Udoy

Context

There are two key decisions to make in production while manufacturing our products: the *quantity* to produce and the *price* to sell at. Produce too few products, and you lose revenue from unsatisfied demand. Conversely, producing too many products causes higher costs with no additional revenue. When we change the pricing of our products, we also affect our consumer demand for them. Pricing too high causes demand to shrink, yet pricing too low creates a poor profit margin. This means pricing and production decisions cannot be treated independently.

After observing the actual production demand, we have the option to 'rush' the manufacturing of additional products at a higher cost to meet demand shortfalls. Additionally, if we overestimate demand, we are required to pay a disposal fee.

Currently, we are making production decisions using a model that does not factor in rush costs, disposal fees, or the relationship between price and demand. In this report, we outline the development of a new model that incorporates these considerations and examine the resulting gains in our net profits from utilizing the extended model. Because demand depends on price, we also estimate a linear price-demand relationship from historical data and use it to simulate demand and optimize both price and quantity in the extended model.

Modeling Assumptions

To optimize our quantity and price, we first make the following assumptions about consumer behavior and production conditions:

I. Demand follows a linear relationship with the price of our products.
II. The linear price–demand relationship is estimated once from historical data and is held fixed across all bootstrap resampling.
III. The residuals of our demand are random and follow the same distribution at every price.
IV. The cost to produce, rush, or dispose of our products is constant.

Current Model

The current version of the newsvendor problem maximizes the expected profit by changing the quantity produced only:

$$\max_{q} \frac{1}{n} \sum_{i=1}^{n} \left(p \min\left(q, D_i\right) - qc \right)$$

For example, if the print quantity is 500 and the demand is 450, the firm incurs 50 units of disposal at a cost of 0.15 per unit, resulting in a disposal cost of $7.50.

Due to the non-linearity of the minimization function, we will use dummy variables and constraints to model this in Gurobi:

$$\max_{q,h} \frac{1}{n} \sum_{i=1}^{n} (h_i)$$

s.t.

$$h_i \leq pD_i - qc$$

h_i constraint if the demand is less than the quantity produced

$$h_i \leq pq - qc$$

h_i constraint if the quantity produced is less than the demand

$$h_i \geq -\infty$$

h_i lower bound

This model has considerable room for improvement. It doesn't factor in the relationships between price and demand, or the rush and disposal costs. Because price and demand are 'unrelated', we lack a methodology for determining an optimal price.

Extended Model

There are two key extensions under our proposed model:

I. *Cost Allocation*

We incorporated the additional rush costs and disposal fees in our model.

$$\max_{q} \frac{1}{n} \sum_{i=1}^{n} \left(pD_i - qc - g(D_i - q)^+ - t(q - D_i)^+ \right)$$

pD_i: Net revenue
$-qc$: Less initial costs
$-g(D_i - q)^+$: Less rush costs when demand > quantity
$-t(q - D_i)^+$: Less disposal fees when quantity > demand

II. *Demand Modeling*

We added the linear relationship between price and demand to our model. Since we can assume that the demand is random and follows a given distribution at every price, we can use the demand residuals from the historic dataset to model the random demand at any price.

$$D = \beta_0 + \beta_1 p + \varepsilon$$

$$\therefore$$

$$D_{p,i} = \beta_0 + \beta_1 p + \varepsilon_i$$

$$\max_{q,p} \frac{1}{n} \sum_{i=1}^{n} \left(p(\beta_0 + \beta_1 p + \varepsilon_i) \; - qc - g(\beta_0 + \beta_1 p + \varepsilon_i - q)^+ - t(q - \beta_0 + \beta_1 p + \varepsilon_i)^+ \right)$$

To formulate the extended model as a Quadratically Constrained Programming (QCP) problem that can be solved by Gurobi, we transform it using dummy variables.

$$\max_{q,p,u,v} \frac{1}{n} \sum_{i=1}^{n} \left(\beta_0 p + \beta_1 p^2 + \varepsilon_i p - qc - gu_i - tv_i \right)$$

s.t.

$$u_i \geq \beta_0 + \beta_1 p + \varepsilon_i - q$$

Rush quantity is at least (demand - quantity)

$$v_i \geq q - \beta_0 - \beta_1 p - \varepsilon_i$$

Disposal quantity is at least (quantity - demand)

$$u_i \geq 0$$

Rush quantity cannot be less than 0

$$v_i \geq 0$$

Disposal quantity cannot be less than 0

$$p \geq 0$$

Price must be at least 0

$$q \geq 0$$

Quantity produced must be at least 0

In the complete model, price is also a decision variable that influences demand through the estimated regression model, and is optimized jointly with quantity in later analysis.

With these extensions defined, we will now show the results of implementing them.

III. *Calibration at Fixed Price and Quantity Optimization*

Part A: Regression to Estimate Demand:

To apply the extended model, we first estimated how demand responds to changes in price using the provided dataset. We fit a simple linear regression of demand on price. The estimated relationship showed an intercept of 1924.7175 and a slope of -1367.7125, with an R-squared value of 0.62. The negative slope confirms that higher prices reduce demand, and the residuals from this regression capture the remaining day-to-day randomness that is not explained by price.

The negative slope $\beta 1$ indicates that higher prices are associated with lower demand, which is consistent with intuition. An $R2$ of about 0.62 means price explains a substantial portion of the variation in demand, while the remaining variation is captured by the residuals.

The residuals are computed as;

$\varepsilon i = Di - (\beta 0 + \beta 1 pi)$, and have a mean essentially zero and a standard deviation of about 150. This residual distribution is used in the next step to generate simulated demand at a fixed price.

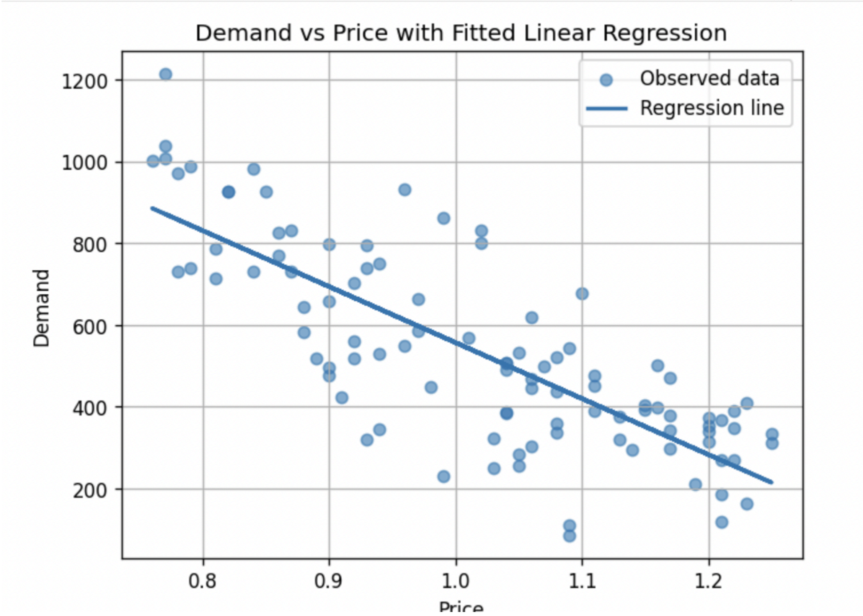
Statistic	Value
Intercept (β_0)	1924.7175
Slope (β_1)	-1367.7125
R^2	0.6215
Residual SD	≈150

```
# 2. Prepare data for regression: Demand = β0 + β1 * Price + ε
X = df[["price"]].values  # shape (n, 1)
y = df["demand"].values   # shape (n, )

# 3. Fit linear regression using sklearn
linreg = LinearRegression()
linreg.fit(X, y)

beta_0 = linreg.intercept_
beta_1 = linreg.coef_[0]
r2 = linreg.score(X, y)
```

This code fits the linear model of demand on price and stores the residuals for later use.



Part B: Simulating Demand at Price = 1:

Next, we fixed the selling price at p = 1 before jointly optimizing price and quantity. Using the regression equation and the residuals, we generated synthetic demand values for p = 1. The resulting demand distribution had a mean of about 557 units, a standard deviation of roughly 150 units, and ranged from approximately 210 to 900 units. The project specifies the following cost parameters:

- $c = 0.5$(regular printing cost per unit)
- g=0.75 (rush printing cost per unit)
- t=0.15 (disposal cost per unit)

With $\beta0$=1924.7175 and $\beta1$=−1367.7125, the simulated demand distribution at p=1 has the following statistics:

Statistic	Value
Mean	557.005
Std Dev	150.213
Min	210.094
Max	898.426

No negative values occurred, so there was no need to clip demands at zero. This distribution represents what demand would look like if the firm consistently charged p=1 while preserving the observed randomness in the data.

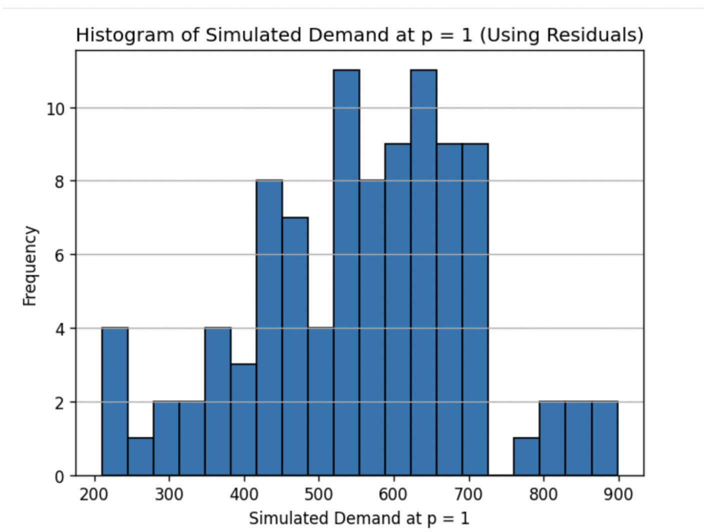
```
p_fixed = 1.0  # fixed selling price for Specifics 2 and 3

print(f"Using c = {c}, g = {g}, t = {t}, and fixed price p = {p_fixed}")

# For each residual ε_i, create synthetic demand at price p = 1:
# D_i(p=1) = β0 + β1 * 1 + ε_i
df["demand_p1"] = beta_0 + beta_1 * p_fixed + df["residual"]

# Demand cannot be negative in this context; clip at zero if necessary
num_negative = (df["demand_p1"] < 0).sum()
if num_negative > 0:
    print(f"\nWARNING: {num_negative} simulated demands were negative; clipping them to 0.")
df["demand_p1"] = df["demand_p1"].clip(lower=0)
```

This code constructs the simulated demand series at p = 1 using the estimated regression and residuals.



Part C: Solving for the Optimal Quantity at p = 1

With p fixed at 1, we solved the extended newsvendor problem as a linear program that includes both rush costs and disposal costs. For each day, we introduced two variables:

- a rush quantity (when demand exceeds q), and
- a disposal quantity (when q exceeds demand).

These were implemented using linear constraints that ensure each variable captures only the positive part of the shortage or leftover amount. The objective function maximizes the average daily profit using the costs provided in the project (printing cost 0.5, rush cost 0.75, disposal cost 0.15).

Solving this linear program using Gurobi produced an optimal print quantity of **471.87 units**. This number lies slightly below the mean simulated demand of 557 units, which makes sense because rushing is more expensive than disposing of extra inventory. Evaluating the profit at this optimal quantity gave an average expected profit of approximately **231.48**.

```python
# Build Gurobi model
m = gp.Model("LP_specific3_p1")

# Decision variable: initial print quantity
q = m.addVar(lb=0.0, name="q")

# Decision variables: u_i (rush quantity), v_i (disposal quantity) for each day
u = m.addVars(n, lb=0.0, name="u")  # shortage (rushed)
v = m.addVars(n, lb=0.0, name="v")  # leftover (disposed)

# Link u_i and v_i to the realized demand and q
for i in range(n):
    # u_i ≥ D_i - q
    m.addConstr(u[i] >= D_p1[i] - q, name=f"rush_link_{i}")
    # v_i ≥ q - D_i
    m.addConstr(v[i] >= q - D_p1[i], name=f"disp_link_{i}")

# Objective: maximize average profit across all days
# profit_i = p * D_i - c * q - g * u_i - t * v_i
profit_expr = gp.quicksum(
    p_fixed * D_p1[i] - c * q - g * u[i] - t * v[i]
    for i in range(n)
) / n

m.setObjective(profit_expr, GRB.MAXIMIZE)

# Optional: turn off verbose Gurobi output if desired
# m.setParam("OutputFlag", 0)

m.optimize()
```

This model captures rush and disposal behavior at p = 1 and returns the optimal fixed-price quantity q.*

```
Optimal objective  2.314836666e+02

Optimal q* (quantity to print at p = 1): 471.8654
```

This fixed-price analysis confirms that the extended model behaves sensibly before we move on to optimizing price and quantity jointly and performing the bootstrap analysis.

IV. *Joint Optimization of Price and Quantity*

In this step, we consider both the selling price p and the print quantity q as decision variables. Demand at any price is modeled by the linear regression.

$$D_i(p) = \beta_0 + \beta_1 p + \varepsilon_i,$$

where εi = residuals from the historical regression. For each demand scenario i, the extended model's profit is

$$\pi_i(p, q) = p \cdot D_i(p) - cq - g \max(D_i(p) - q, 0) - t \max(q - D_i(p), 0),$$

where c = regular printing cost, g = rush cost, and t = disposal cost.

Using the same slack-variable trick as before, we introduce shortage variables ui and leftover variables vi and enforce

$$u_i \geq D_i(p) - q, \quad v_i \geq q - D_i(p), \quad u_i \geq 0,\ v_i \geq 0.$$

The objective is to maximize the average profit across all demand scenarios.

$$\max_{p,q,u,v} \frac{1}{n} \sum_{i=1}^{n} \left[p \cdot D_i(p) - cq - gu_i - tv_i \right].$$

The resulting optimal price and quantity pair $(p*, q*)$ is the benchmark decision we use inside the bootstrap procedure in the next section.

Solving the QCP using Gurobi produced the following optimal decision:

Decision Variable	**Optimal Value**
Optimal price (p^*)	0.9536
Optimal quantity (q^*)	535.2910 units
Expected average profit	$234.42

The values above represent the optimal price and quantity pair, given the underlying demand–price relationship and cost structure of our dataset.

Final Results

Before running the bootstrap procedure, we first validated the behavior of the extended model at a fixed price of p = 1.

As an intermediate step before we ran the full bootstrap simulation, we estimated a single bootstrap resample to calculate how sensitive the optimal decisions are to variation in the underlying dataset. After resampling the historical (price, demand) pairs, refitting the regression, and re-solving the joint price–quantity QCP, we obtained:

Metric	**Value**
Optimal price (p*{boot})	0.9595
Optimal quantity (q*{boot})	513.3112 units
Expected profit	$226.50

Compared to the original solution (p* = 0.9536, q* = 535.29, profit ≈ $234.42), this bootstrap sample recommends a slightly higher price, a lower print quantity, and a lower expected profit. This confirms that optimal decisions change with the underlying sample, motivating the full bootstrap analysis that follows.

To perform the bootstrap analysis, we repeatedly generated new synthetic demand datasets by resampling the regression residuals and reconstructing demand using the estimated price–demand model. For each bootstrap sample, we re-solved the full joint optimization problem (price and quantity) using Gurobi. This produced 1,000 estimates of the optimal price (p*) and optimal print quantity (q*), giving us an empirical distribution that reflects how sensitive the optimal decisions are to demand variability.

Running a large number of bootstrap replications produced empirical distributions of the optimal price, optimal quantity, and expected profit under the extended model. The summary statistics are shown below:

Statistic	**(p*)**	**(q*)**	**Profit**
Mean	0.953	538.45	$234.94
Std Dev	0.014	32.489	8.766

Min	0.918	454.036	$211.80
Max	0.999	627.598	$258.28

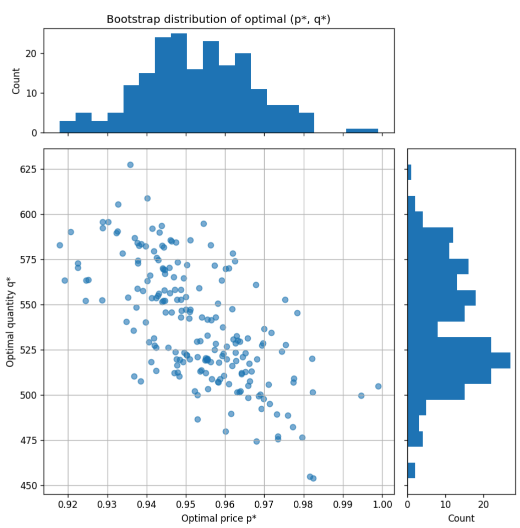
These results show:

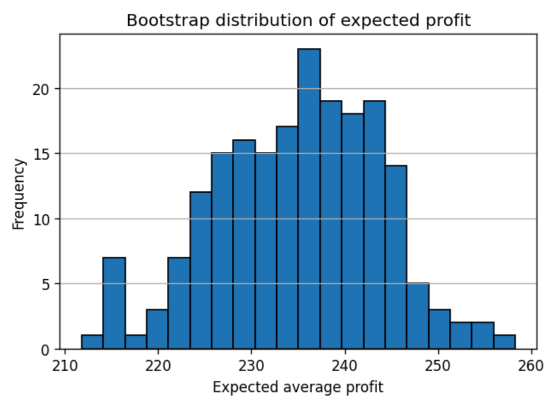
- The optimal price is extremely stable, varying within roughly ±1.5 cents.
- Optimal quantity varies more, reflecting uncertainty in the slope of the demand curve.
- Expected profit ranges from $212 to $258, with a strong center around $235, matching the original dataset.

The bootstrap distributions are visualized below:

- Scatterplot of optimal (p*, q*) with marginal histograms
- Histogram of optimal profit

The marginal histograms along the axes show the distributions of optimal prices and optimal quantities across the bootstrap samples.





To assess the performance of the extended model relative to the standard newsvendor model, we run a separate bootstrap experiment with 1,000 samples for the comparative analysis next.\

Comparison of the Standard NV Model and the Extended Model:
To compare the results of the two methodologies, we took 1000 bootstrap samples and optimized the original newsvendor solution and the extended solution. Since the original version does not optimize price, we used the average from the bootstrap sample as the input price.

The output from running the original NV model is the average profit without considering our real-world conditions. Therefore, the next step is to examine the true profits generated by modeling the additional costs of rushing and disposing, as well as the additional revenue brought about by being able to sell to 100% of the demand capacity.

'Real World' NV Profit = Average recognized profit + average unrecognized profit

Average recognized profit = Original NV model output

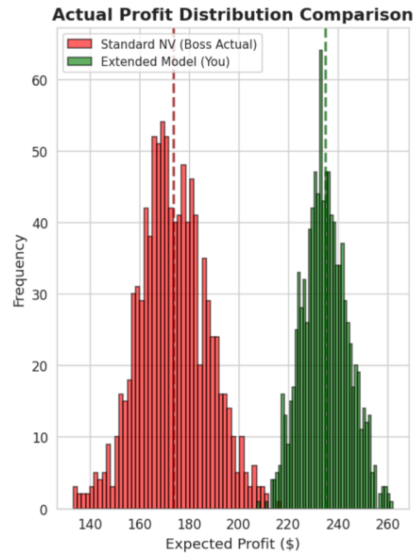
$$\text{Average unrecognized profit} = \frac{1}{n}\sum_{i=1}^{n}\left(p(D_i - q)^+ - g(D_i - q)^+ - t(q - D_i)^+\right)$$

$p(D_i - q)^+$
Additional profit from the ability to always meet demand

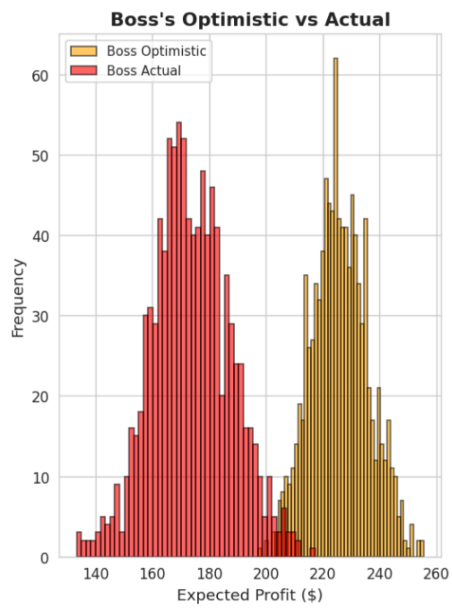
$-g(D_i - q)^+$
Additional rush costs

$-t(q - D_i)^+$
Additional disposal costs

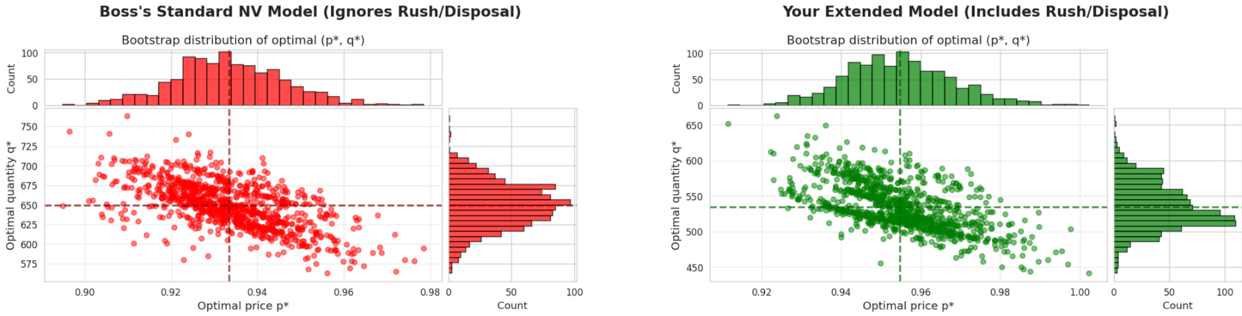
The average profits between the standard NV model and the extended model are **$173.62** and **$234.79**, respectively. This means that, on average, employing the extended model generates **$61.17** of additional profit (35.23% improvement). The extended model outperformed the standard model 100% of the time. A paired t-test across the 1,000 bootstrap replications reports t=246.45 with p< 10^{-6}, confirming that the extended model significantly outperforms the standard model. The distribution of average profits for the standard and extended models is shown below:



In addition to maximizing profitability, it is important to have a reliable estimate of your incoming cash flows. Under our current assumptions, there is an average overestimation of profits of **$52.29**.



The difference in performance between the standard and extended NV models appears to stem from a systematic underestimation of the ideal price and an overestimation of the ideal quantity.



Our fixed-price calibration confirmed that the extended model behaves consistently with the economic structure of underage and overage costs, which supports the validity of our full joint optimization and bootstrap analysis.

Summary of Key Outputs

Metric	Standard NV	Extended Model	Improvement
Mean actual profit	173.45	234.81	+61.36
Mean optimal price	0.932	0.954	+0.022
Mean optimal quantity	650.52	536.13	–114.39

Conclusion

In this project, we extended a basic newsvendor model by explicitly including rush and disposal costs and by incorporating a price–demand relationship estimated from historical data. After calibrating the model at a fixed price and verifying that it behaved sensibly, we solved the full joint optimization problem for price and quantity and evaluated its performance using bootstrap resampling.

The extended model recommends a price of about 0.95 dollars and a print quantity of roughly 535–540 units, compared with the boss's lower price of around 0.93 dollars and a much larger quantity of about 650 units. By charging slightly more and printing fewer units, the extended model reduces waste while still covering demand and the cost of rush production.

Across 1,000 bootstrap samples, the extended model improves mean actual profit by approximately 61 dollars per day, which is a 35% increase over the standard newsvendor approach, and it outperforms the boss's model in every sample. The improvement is statistically significant, and the profit distribution under the extended model is both higher and tighter, indicating more reliable financial performance. Overall, the analysis shows that modeling both price sensitivity and the operational frictions associated with rush and disposal costs leads to better and more robust production and pricing decisions.