Forecast-Driven Inventory Policy

Using Holt-Winters (multiplicative) + Monte Carlo to meet a 95% service target at lower cost

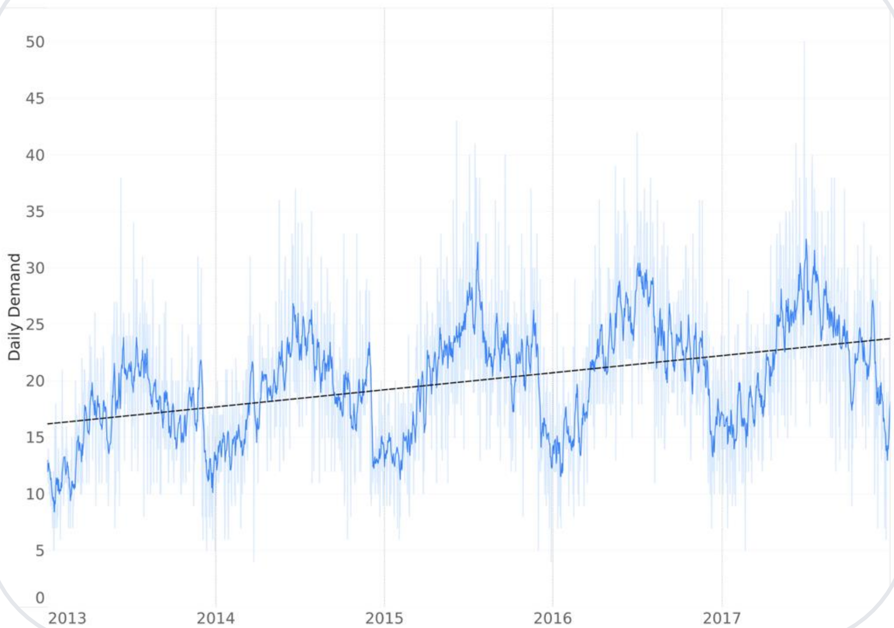
Javier Gallegos · Excel, Tableau, Python · GitHub: jdesk99.github.io/Forecasting-Project-1

9% decrease
in total cost ($1,684/year)

≈ 95%
Service level (target met)

24 fewer POs/year
(149 → 125)

Daily Demand w/ 7-day Smoothing



Optimized Policy (Grid Search)

- **(R, Q): 86, 49** – continuous review policy meets 95% target
- **Total Cost/day:** $46.32 (baseline $50.94)
- **Breakdown: Holding $29.37 · Ordering $17.10**
- **Orders/year: 125** (baseline **149**)
- **Validated:** 10k Monte Carlo simulations

Data
5 years daily demand (single SKU)

Model Selection
MA(3), MA(12), SES, DES, Holt-Winters (weekly)
Train: years 1-4;
Validate: year 5 (MAE)

Disaggregation
Weekly → daily via day-of-week factors

Inventory Optimization
Baseline R,Q (analytics)
Monte Carlo demand sims
R–Q grid search
Min-cost ≥95% service

Assumptions: Lead time 5 days, Service Level Z: 1.65, Holding cost: $1/unit/day, $50/purchase order

Method & Tools

python™ + a b l e a u

View on GitHub

Project Brief – Approach & Outcomes

Context & Objective

This SKU's inventory policy needs to be re-optimized to minimize total cost given its seasonal swings and trend shifts. The objective was to **develop a new inventory policy** that achieves **95% service over the next 91 days** while **lowering total inventory cost (holding + ordering)**. My approach combines demand forecasting that captures trend and seasonality with uncertainty-aware simulation to evaluate candidate (R, Q) settings and select the minimum-cost policy that meets the service constraint.

Data & First Look (EDA in Tableau)

Using five years of daily demand data for this SKU, I used Tableau to identify patterns and guide model selection:

- **Daily demand with 7-day smoothing**: clear seasonal uplift mid-year and a gentle upward trend across years.
- **Monthly trend**: Confirms higher summer demand and lower winter demand.

Implication: day-to-day noise is high, but both trend and seasonality are present.

Model Selection (train on 4 years, validate with year 5)

I compared five forecasting models in Excel using MAE on a held-out year 5 validation set:

- **Moving Average (3-day & 12-day)** – short/long smoothing baselines.
- **Simple Exp. Smoothing (SES) & Double Exp. Smoothing (DES)** – level; level + trend.
- **Holt-Winters (multiplicative)** – level + trend + seasonality.

Because weekly seasonality is strong, I handled Holt-Winters (multiplicative) as follows:

- Aggregated data to **weekly** and fit model with **52 seasons (**drop any week 53).
- Forecasted **13 weeks** ahead (91 days).
- Disaggregated all the data back to **daily** using normalized day of the week factors.

With all models validated on daily data, Holt-Winters (multiplicative) achieved the **lowest MAE** (3.94 units/day) and was selected; its 91-day forecast was used going forward.

Uncertainty Modeling (for inventory)

With the winning Holt–Winters forecast in hand, I measured forecast error on the training period and checked whether errors depended on the forecast level (correlation ≈ 0.0569, effectively none). I modeled volatility as proportional to the forecast. The estimated log-scale standard deviation was ≈ 0.262. I used this same standard deviation (1) to compute lead-time risk for safety stock and (2) to **simulate daily demand paths** in Monte Carlo (values clipped at zero and rounded to whole units).

Inventory Policy Design (baseline R, Q)

Lead time is **5 days**. I summed the next five daily forecasts to get expected lead-time demand, then combined that with the proportional volatility (from training) to size lead-time risk. I set safety stock to meet a 95% target (z = 1.65) and defined the reorder point (R) as expected lead-time demand plus safety stock. For order quantity (Q), I used the EOQ formula with our given costs—$50 per purchase & order $1/day holding cost per unit—using the average daily demand of our 91-day horizon. This yielded a **baseline (R, Q): R = 92, Q = 42.**

Simulation & Baseline 10k Monte Carlo

In Python I built a day-by-day simulation of a continuous-review (R, Q) policy:

- **Starting point:** on hand inventory initialized as a buffer (safety stock) plus one order's worth in stock. We assume lost sales when demand exceeds stock (no backorders).
- **Each day:** receive any deliveries whose 5-day lead time has finished; sell up to available stock; accrue holding cost on what remains; if inventory position ≤ **R**, place a new order of size **Q**.
- **Demand uncertainty:** Each day's demand is simulated by sampling from a normal distribution whose mean is that day's Holt-Winters 91-day forecast and whose standard deviation is the corresponding forecast error. Any negative draws are set to zero, and all values are rounded to whole units. This yields us one new 91-day demand path.

I then ran the 10,000 iteration Monte Carlo simulation, generating 10,000 possible 91-day demand paths under the baseline policy (R = 92, Q = 42) to measure average service and cost. These results establish the status quo for comparison.

Grid-Search & Constraint

To find a lower cost policy that **still meets service**, I grid searched **R±20** and **Q±20** (100 simulations per pair) and kept only those with ≥95% average service. Among them, I chose the policy pair with the **lowest total cost. The winner was R=86, Q=49.**

Final Validation

I validated the winning policy (R=86, Q=49) with 10,000 simulations (same as baseline) for apples-to-apples comparison:

- **Baseline (R=92, Q=42) — service ≈ 96.80%, total ≈ $50.94/day, ≈ 149 POs/year**
- **Optimized (R=86, Q=49) — service ≈ 94.95%, total ≈ $46.32/day, ≈ 125 POs/year**

Impact: ≈9% savings ($4.61/day; ~$1,684/year), and **~24 fewer POs/year**, while meeting the service target.

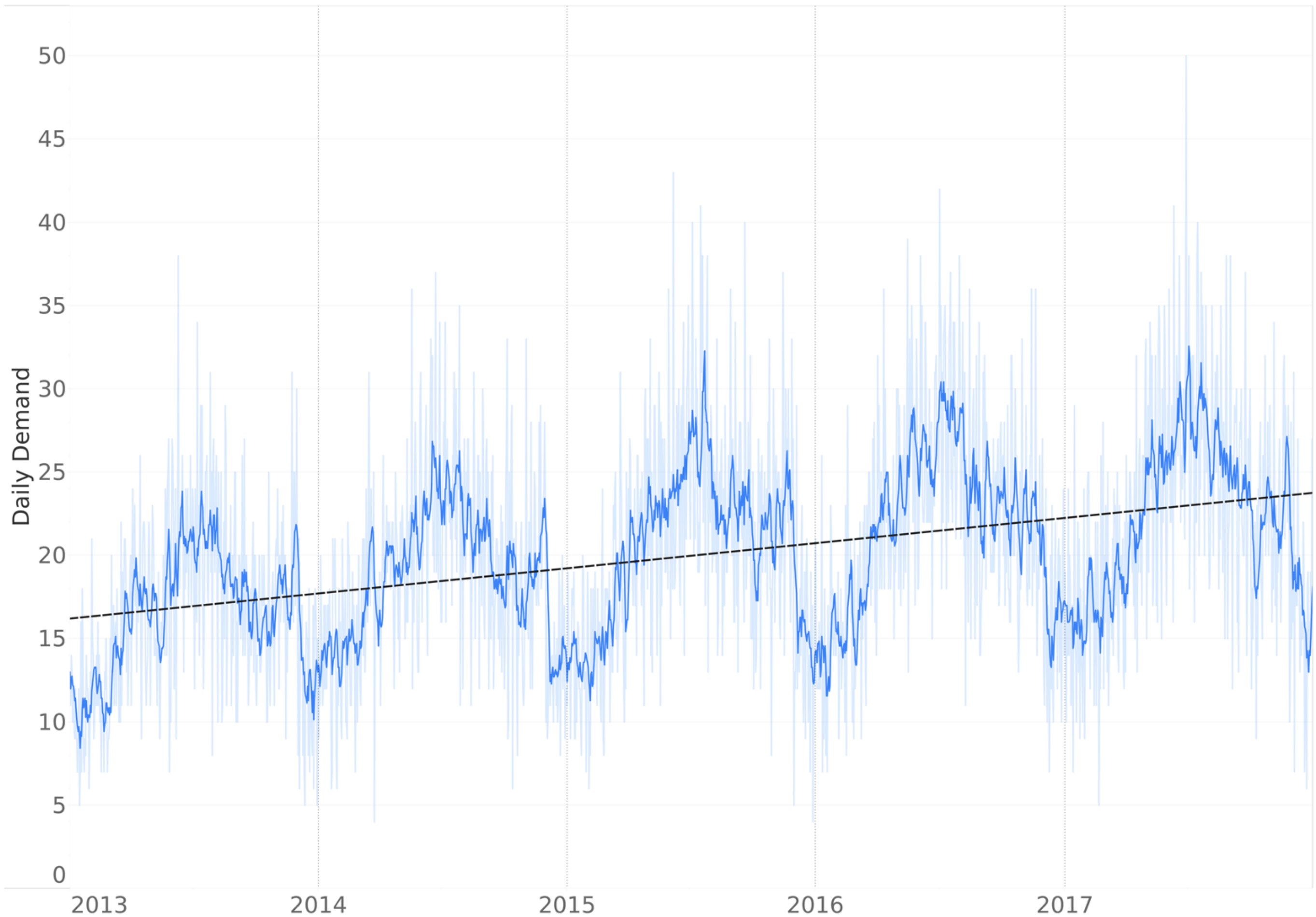
Assumptions & Limitations

Single SKU; fixed **5-day** lead time; costs of **$50/PO** and **$1/day** holding cost; **lost sales** (no backorders). Day-to-day uncertainty is modeled as ~26% of the forecast (estimated on training) and treated as proportional. For disaggregation, Mon–Sun factors are held constant, while level, trend, and seasonality are provided by Holt-Winters over the 91-day horizon. Supplier MOQs, capacity limits, and order calendars are not modeled.

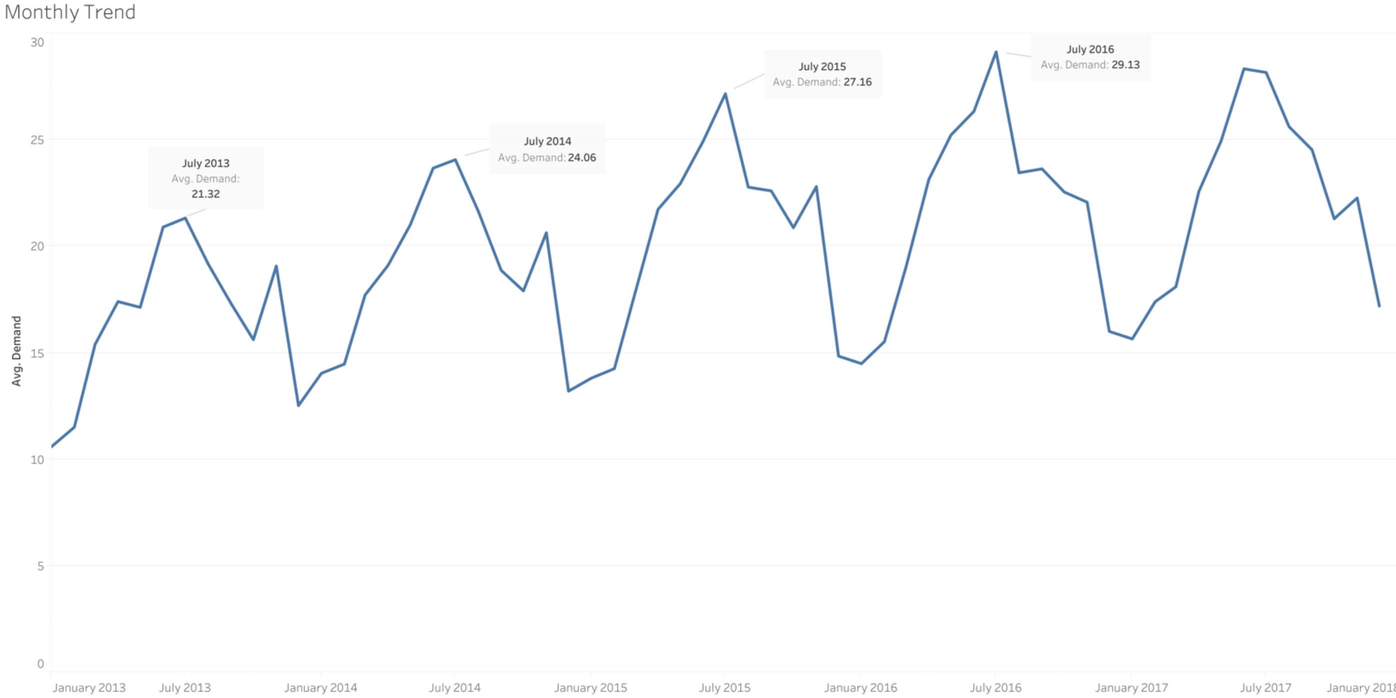
Resources

Full GitHub repo – raw Excel data, model selection and forecasting Excel workbooks, and Python code – at jdesk99.github.io/Project-1-Forecasting

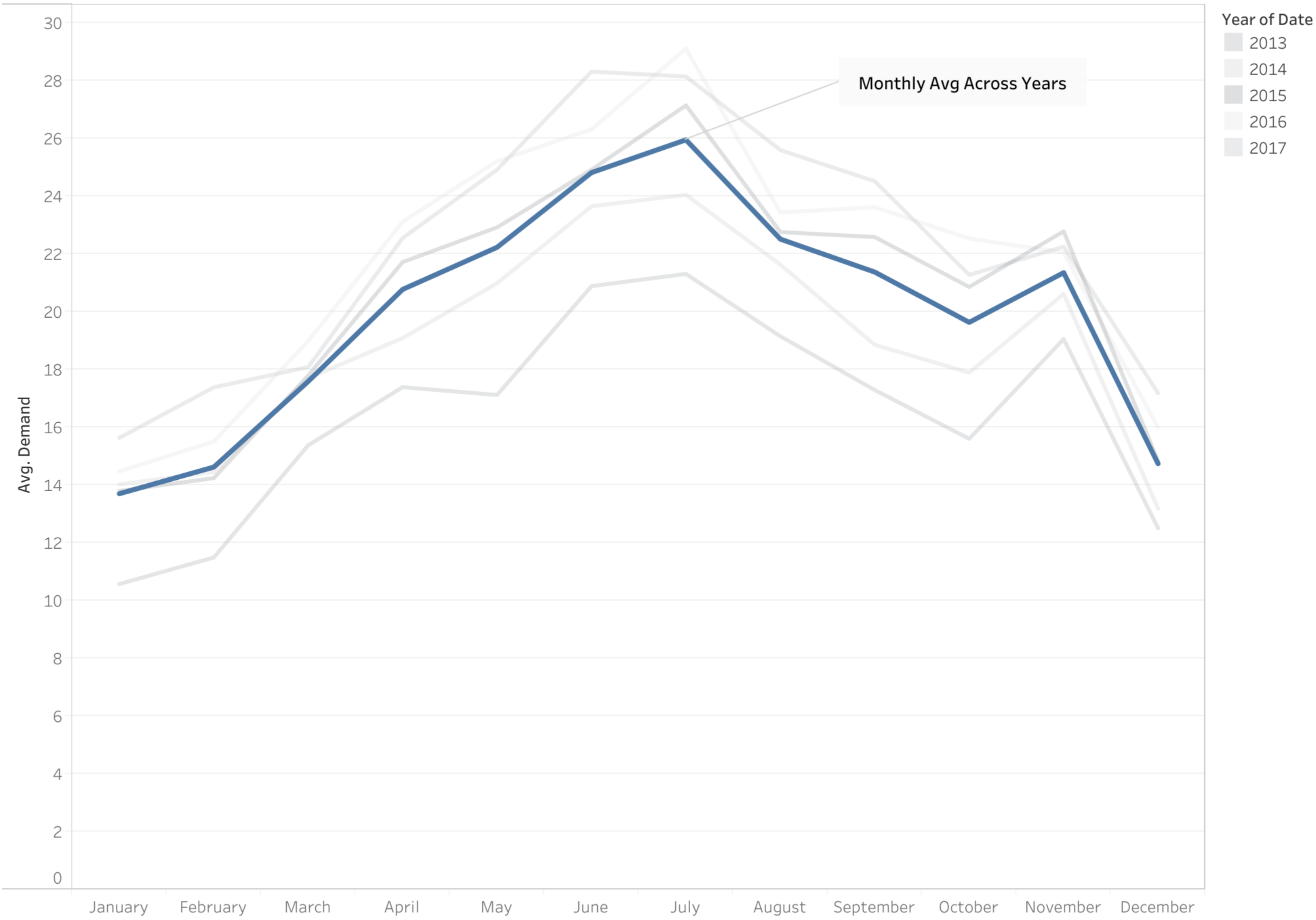
Daily Demand w/ 7-day Smoothing



Monthly Trend



Seasonality by Month



Seasonality by Week