NAME: SHUBHAM TIWARI
ROLL NO: HBSU2330

Monte Carlo Simulation Report: Lexus Business

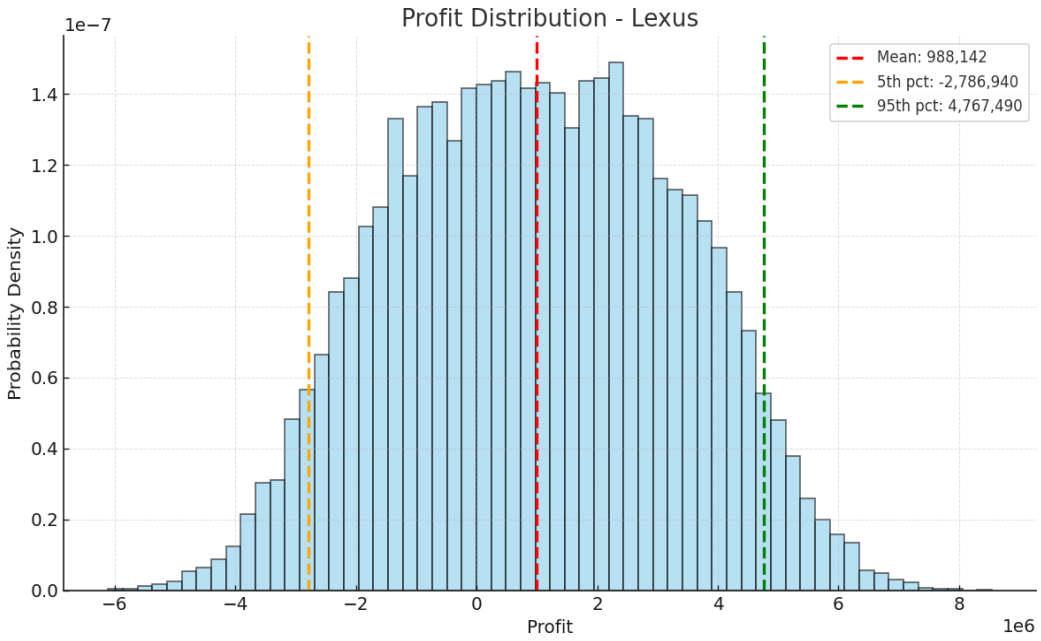
Assumptions:

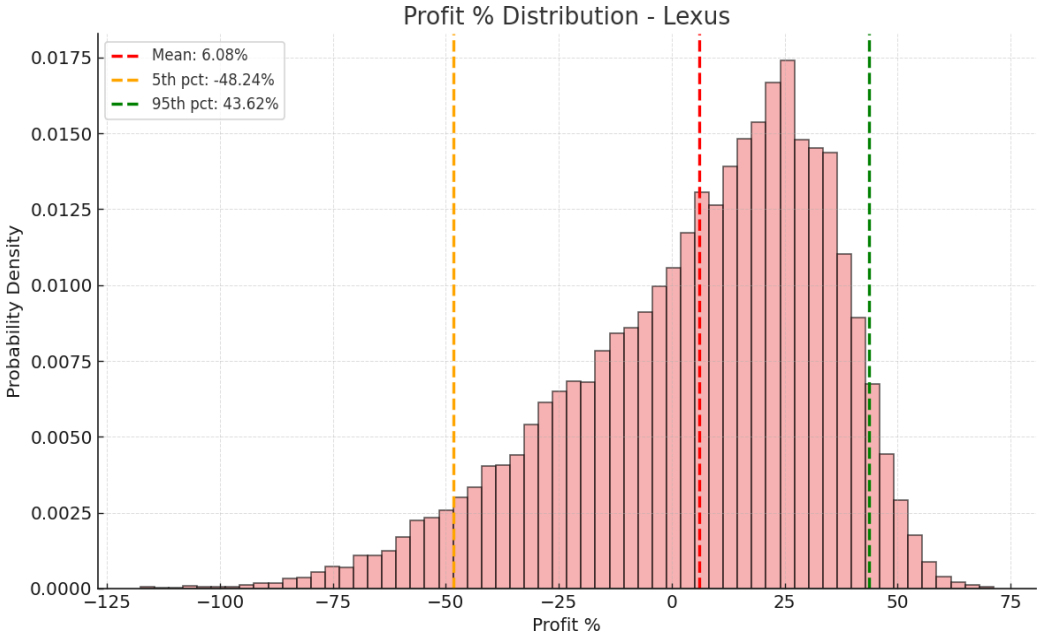
Revenue ~ Uniform(5,000,000, 12,000,000)
Costs ~ Normal(mean=7,500,000, std=1,200,000)
Target Profit % Range: 15.0% – 25.0%

Summary Statistics:

Metric	Value
Mean Profit	988,141.54
Std Dev Profit	2,348,146.60
5th Percentile Profit	-2,786,940.42
95th Percentile Profit	4,767,489.83
Mean Profit %	6.08%
5th Percentile Profit %	-48.24%
95th Percentile Profit %	43.62%
Probability Profit% in Range	15.74%
Probability of Loss	35.97%

Distributions:





Appendix: Python Script

```
#!/usr/bin/env python3
"""
Monte Carlo Simulation for a Car Company (LEXUS)

This script simulates profit and profit percentage under uncertainty for a Lexus business unit.
It mirrors the structure of a standard Monte Carlo setup:
- Random revenue
- Random costs
- Profit = Revenue - Costs
- Profit % = (Profit / Revenue) * 100
- Summary statistics and probability of meeting a target profit %

USAGE
--------
$ python lexus_monte_carlo.py

CONFIG
---------
Adjust the parameters in the CONFIG section below to match your case.
By default, values are illustrative (currency-agnostic).

NOTE
------
- Charts are optional; set SAVE_PLOTS = True to save PNGs.
- No external data required.
"""

import numpy as np
import matplotlib.pyplot as plt

# =====================
# CONFIG (Edit freely)
# =====================
SEED = 42
N_SIM = 50_000  # number of Monte Carlo trials

# Revenue assumptions (choose ONE approach and comment the other if desired)
# Approach A: Uniform revenue between downside and upside
REV_UNIFORM_LOW = 5_000_000     # downside revenue
REV_UNIFORM_HIGH = 12_000_000   # upside revenue

# Approach B: Triangular revenue if you have most-likely value (optional)
USE_TRIANGULAR_REVENUE = False
REV_TRI_LOW = 5_000_000
REV_TRI_MODE = 9_000_000
REV_TRI_HIGH = 12_000_000

# Cost assumptions: Normally distributed
COST_MEAN = 7_500_000
COST_STD = 1_200_000

# Business/Reporting targets
TARGET_PROFIT_PCT_RANGE = (15.0, 25.0)  # desired profit % band

# Plot options
SAVE_PLOTS = True
PLOT_DIR = "."  # directory to save plots

# =====================
# Simulation
# =====================
rng = np.random.default_rng(SEED)

if USE_TRIANGULAR_REVENUE:
    # numpy triangular uses left, mode, right
    revenue = rng.triangular(REV_TRI_LOW, REV_TRI_MODE, REV_TRI_HIGH, size=N_SIM)
else:
    revenue = rng.uniform(REV_UNIFORM_LOW, REV_UNIFORM_HIGH, size=N_SIM)

costs = rng.normal(COST_MEAN, COST_STD, size=N_SIM)

profit = revenue - costs
profit_pct = np.where(revenue != 0, (profit / revenue) * 100.0, np.nan)

# Summary stats
```

```
def pct(arr, q):
    return float(np.nanpercentile(arr, q))

summary = {
    "mean_profit": float(np.nanmean(profit)),
    "std_profit": float(np.nanstd(profit, ddof=1)),
    "p5_profit": pct(profit, 5),
    "p50_profit": pct(profit, 50),
    "p95_profit": pct(profit, 95),
    "mean_profit_pct": float(np.nanmean(profit_pct)),
    "p5_profit_pct": pct(profit_pct, 5),
    "p50_profit_pct": pct(profit_pct, 50),
    "p95_profit_pct": pct(profit_pct, 95),
}

low, high = TARGET_PROFIT_PCT_RANGE
meets_target = np.logical_and(profit_pct >= low, profit_pct <= high)
prob_meet_target = float(np.nanmean(meets_target))  # in 0..1

# Risk views
prob_loss = float(np.nanmean(profit < 0))
pctl_loss_5 = pct(profit, 5)  # 5th percentile -> downside
pctl_gain_95 = pct(profit, 95)  # 95th percentile -> upside

# Print results
print("=== Monte Carlo Results: Lexus Business ===")
print(f"Trials: {N_SIM:,}")
print("\n--- Assumptions ---")
if USE_TRIANGULAR_REVENUE:
    print(f"Revenue ~ Triangular(low={REV_TRI_LOW:,.0f}, mode={REV_TRI_MODE:,.0f}, high={REV_TRI_HIG
else:
    print(f"Revenue ~ Uniform({REV_UNIFORM_LOW:,.0f}, {REV_UNIFORM_HIGH:,.0f})")
print(f"Costs   ~ Normal(mean={COST_MEAN:,.0f}, std={COST_STD:,.0f})")
print(f"Target Profit % Range: {low:.1f}% to {high:.1f}%")

print("\n--- Summary Statistics ---")
print(f"Mean Profit: {summary['mean_profit']:,.2f}")
print(f"Std Dev Profit: {summary['std_profit']:,.2f}")
print(f"5th / 50th / 95th Profit: {summary['p5_profit']:,.2f} | {summary['p50_profit']:,.2f} |
print(f"Mean Profit %: {summary['mean_profit_pct']:.2f}%")
print(f"5th / 50th / 95th Profit %: {summary['p5_profit_pct']:.2f}% | {summary['p50_profit_pct']:.2f

print("\n--- Goal & Risk ---")
print(f"Probability Profit %% in [{low:.1f}%, {high:.1f}%]: {prob_meet_target*100:.2f}%")
print(f"Probability of Loss (Profit < 0): {prob_loss*100:.2f}%")
print(f"Downside (5th pct Profit): {pctl_loss_5:,.2f}")
print(f"Upside   (95th pct Profit): {pctl_gain_95:,.2f}")

# Optional plots
if SAVE_PLOTS:
    # Profit histogram
    plt.figure()
    plt.hist(profit, bins=60)
    plt.xlabel("Profit")
    plt.ylabel("Frequency")
    plt.title("Profit Distribution - Lexus")
    plt.tight_layout()
    plt.savefig(f"{PLOT_DIR}/lexus_profit_hist.png", dpi=150)
    plt.close()

    # Profit % histogram
    plt.figure()
    plt.hist(profit_pct[~np.isnan(profit_pct)], bins=60)
    plt.xlabel("Profit %")
    plt.ylabel("Frequency")
    plt.title("Profit % Distribution - Lexus")
    plt.tight_layout()
    plt.savefig(f"{PLOT_DIR}/lexus_profit_pct_hist.png", dpi=150)
    plt.close()

    print("\nCharts saved:")
    print(f" - {PLOT_DIR}/lexus_profit_hist.png")
    print(f" - {PLOT_DIR}/lexus_profit_pct_hist.png")
```