Staff Answer Key for Resource Price Fluctuation Simulation

ISYS2001 Introduction to Business Programming

Table of contents

Introduction

This marking guide is designed to assist you in understanding how to evaluate submissions for the Resource Price Fluctuation Simulation assignment. As a tutor in this unit, you likely have the ability to recognize poor, good, and excellent submissions based on your experience and knowledge. However, the detailed guidelines provided below will help you gain a deeper understanding and offer a structured approach to applying the rubric effectively.

Workflow for Marking Submissions

1. **Initial Impressions:**
    - Open the student's notebook and take note of your first impressions of the report.
    - Check if the code is hidden or displayed, the presence of section headings and titles, and whether the plots look appropriate (remember that Colab may remove output cells when saving).
2. **Code Execution:**

- Run the code cells to ensure they execute correctly without errors.
- Review the text cells to see if they accurately describe the plots and the simulation results.

3. **Content Structure:**
   - Determine if the student has used a setup, conflict, and resolution framework to tell the story of their simulation.
   - Ensure that the report follows a logical structure and is easy to follow.
4. **Supporting Files:**
   - Inspect the repository folder for any support files and additional notebooks.
   - Look for evidence of testing, exploration, and AI conversations that might support the main notebook.
5. **Applying the Rubric:**
   - Use the provided rubric in Blackboard to mark the submission.
   - Consult the detailed guidelines below if you need further clarification or if specific aspects of the submission require more in-depth evaluation.

Detailed Guidelines

These guidelines provide further details on what to look for in each section of the student's submission. They help ensure that your marking is consistent and fair, and provide a reference for evaluating the various components of the assignment. Use these guidelines to deepen your understanding and to support your application of the rubric.

By following this structured approach, you will be able to assess each submission thoroughly and provide constructive feedback to help students improve their understanding and skills.

Guidelines for Providing Feedback on Resource Price Fluctuations Simulation Assignments

Be Specific:

- **Code:** Pinpoint specific lines with errors or areas for optimisation.
- **Visualisations:** Identify if they are unclear, inaccurate, or lacking detail. Suggest improvements (clearer labels, better chart types).
- **Analysis:** Specify which aspects lack depth (missing comparisons, insufficient interpretation).

Be Constructive:

- **Code:** Suggest alternative approaches, libraries, and relevant resources.

- **Visualisations:** Recommend specific tools/functions to enhance clarity and impact.
- **Analysis:** Guide students to ask more probing questions or consider additional metrics for deeper insights.

Balance Feedback:

- **Positive:** Acknowledge successes, such as correctly implementing simulation parameters or generating clear visualisations.
- **Negative:** Frame as learning opportunities, focusing on how they can improve.

Encourage Self-Reflection:

- **Ask questions like:**
  - "How would a longer supply disruption affect prices?"
  - "What if the disruption caused a price decrease instead of an increase?"
  - "Which hedging strategy seems most effective in this scenario, and why?"

Use Clear Language:

- Avoid jargon. Explain technical terms or provide links to definitions.
- Maintain a supportive and encouraging tone throughout the feedback.

Offer Examples and Resources:

- Share examples of well-done work (anonymised) for reference.
- Recommend Simulacra documentation, economic modeling tutorials, or relevant resources on commodity markets.

Follow Up:

- Encourage students to discuss feedback in office hours or meetings.

Example Feedback (Tailored to Resource Price Fluctuations)

Poor:

> Your analysis of resource price fluctuations is insufficient. You haven't explored hedging strategies, and your explanations lack depth.

Good:

> You've done a good job simulating resource prices with varying volatility and supply disruptions. Your visualisations are clear. To improve, consider exploring hedging strategies and calculate metrics like standard deviation to quantify the impact of volatility.

Best:

> Excellent analysis! You've thoroughly explored volatility, supply disruptions, and hedging strategies, with clear visualisations and insightful interpretations. Your cost-benefit analysis of hedging is particularly strong. To take it further, you could research and model more complex real-world hedging strategies.

Resource Price Fluctuations Simulation Analysis – Staff Answer Key

Purpose: This key provides guidance for analysing the resource price fluctuations simulation, specifically exploring the impact of volatility, supply disruptions, and potential hedging strategies. It is meant to be used in conjunction with the assignment specification.

Important Considerations for Tutors:

- **Clarity:** Emphasise the importance of clear, well-labeled visualisations.
- **Explanation:** Encourage students to provide detailed explanations and interpretations of their observations, linking them to economic and business concepts.
- **Limitations:** Remind students to discuss the limitations of the simulation model (e.g., simplified market dynamics, lack of external factors).
- **Creativity:** Reward students who demonstrate creativity and initiative in exploring additional scenarios or strategies.

Task 1: Assess the Impact of Volatility on Price Stability

Objective: Investigate how varying levels of volatility influence the stability of resource prices.

Approach:

1. **Vary Volatility:** Run simulations with different volatility values (e.g., 0.01, 0.05, 0.10) while keeping other parameters constant (e.g., start price, drift).
2. **Visualise:** Plot the resource prices for each simulation on the same graph.
3. **Analyse:**
    - **Qualitative:** Describe the visual differences in price patterns as volatility increases. How does higher volatility affect the frequency and magnitude of price swings?
    - **Quantitative:** Calculate and compare metrics such as:
        - Standard deviation of prices (a measure of volatility).
        - Range of prices (the difference between the highest and lowest prices).
        - Number of days with significant price changes (e.g., exceeding a certain percentage threshold).
    - **Interpretation:** Explain how different volatility levels impact a business's ability to predict and manage costs. What are the implications for procurement strategies and financial planning?

Pseudo Code

```
volatility_levels = [0.01, 0.02, 0.05]  # Different levels of volatility to test

for volatility in volatility_levels:
    sim = ResourceSimulation(start_price=100, days=250, volatility=volatility, drift=0.0003)
    prices = sim.run_simulation()
    plot_prices(prices, volatility)  # Visualise the price fluctuations for each volatility l
    print_statistics(prices)  # Function to print or calculate statistical metrics (mean, st
```

Implementation

1. **Setup Simulation with Different Volatility Levels**

   ```
   from simulacra import ResourceSimulation
   import matplotlib.pyplot as plt

   def plot_prices(prices, volatility):
       plt.figure(figsize=(10, 6))
       plt.plot(prices, label=f'Volatility: {volatility}')
       plt.xlabel('Days')
       plt.ylabel('Price')
       plt.title(f'Resource Price Simulation with Volatility {volatility}')
       plt.legend()
       plt.show()

   def print_statistics(prices):
       mean_price = prices.mean()
       std_dev = prices.std()
       max_drawdown = (prices.max() - prices.min()) / prices.max()
       print(f'Mean Price: {mean_price:.2f}')
       print(f'Standard Deviation: {std_dev:.2f}')
       print(f'Max Drawdown: {max_drawdown:.2f}')

   volatility_levels = [0.01, 0.02, 0.05]
   for volatility in volatility_levels:
       sim = ResourceSimulation(start_price=100, days=250, volatility=volatility, drift=0.0
       prices = sim.run_simulation()
       plot_prices(prices, volatility)
       print_statistics(prices)
   ```

2. **Example Outputs and Analysis**

- **Volatility 0.01:**
  - Mean Price: $100.30
  - Standard Deviation: $1.50
  - Max Drawdown: 0.01

  Analysis: Low volatility results in stable prices with minor fluctuations, indicating low risk but also limited potential for high returns.

- **Volatility 0.02:**
  - Mean Price: $100.45
  - Standard Deviation: $3.10
  - Max Drawdown: 0.02

  Analysis: Moderate volatility introduces more fluctuations, increasing both risk and potential returns. Prices show more significant swings compared to low volatility.

- **Volatility 0.05:**
  - Mean Price: $100.75
  - Standard Deviation: $7.80
  - Max Drawdown: 0.05

  Analysis: High volatility leads to substantial price swings, resulting in higher risk and potential for greater returns. Price variability is significant, making the market more unpredictable.

Task 2: Model a Supply Disruption Event and Analyse Its Impact

Objective: Examine how a supply disruption affects resource prices, considering both the timing and severity of the disruption.

Approach:

1. **Introduce Disruptions:** Choose specific days to simulate supply disruptions, varying the severity (e.g., 10% decrease, 30% increase in price). Consider both positive (shortage) and negative (surplus) disruptions.
2. **Visualise:** Plot resource prices, clearly marking the disruption day on the graph.
3. **Analyse:**
   - **Immediate Impact:** Quantify the immediate price change after the disruption. Did the simulation accurately reflect the severity parameter?
   - **Recovery/Adjustment:** Analyse how prices evolve in the days/weeks following the disruption. Does the market return to pre-disruption levels, or are there lasting effects?

- **Interpretation:** Discuss the factors that seem to influence the market's response to disruptions (timing, severity, underlying market conditions). What are the implications for supply chain risk management?

Pseudo Code

```
supply_disruption_days = [50, 150, 200]  # Days to simulate a supply disruption event
disruption_severities = [0.1, 0.3, 0.5]  # Different severities of the disruption

for day in supply_disruption_days:
    for severity in disruption_severities:
        sim = ResourceSimulation(start_price=100, days=250, volatility=0.02, drift=0.0003, su
        prices = sim.run_simulation()
        plot_prices(prices, day, severity)  # Visualise with the disruption day marked
        analyse_disruption_impact(prices, day)  # Analyse and compare before and after disru
```

Implementation

1. **Setup Simulation with Supply Disruption Events**

```
def plot_prices(prices, supply_disruption_day, disruption_severity):
    plt.figure(figsize=(10, 6))
    plt.plot(prices, label=f'Disruption Severity: {disruption_severity} on Day {supply_d
    plt.axvline(x=supply_disruption_day, color='red', linestyle='--', label='Supply Disr
    plt.xlabel('Days')
    plt.ylabel('Price')
    plt.title(f'Resource Price Simulation with Disruption Severity {disruption_severity}
    plt.legend()
    plt.show()

def analyse_disruption_impact(prices, supply_disruption_day):
    pre_disruption_price = prices[:supply_disruption_day].mean()
    post_disruption_price = prices[supply_disruption_day:].mean()
    print(f'Average Price Before Disruption: {pre_disruption_price:.2f}')
    print(f'Average Price After Disruption: {post_disruption_price:.2f}')

supply_disruption_days = [50, 150, 200]
disruption_severities = [0.1, 0.3, 0.5]
for day in supply_disruption_days:
    for severity in disruption_severities:
        sim = ResourceSimulation(start_price=100, days=250, volatility=0.02, drift=0.000
        prices = sim.run_simulation()
        plot_prices(prices, day, severity)
        analyse_disruption_impact(prices, day)
```

2. **Example Outputs and Analysis**

   - **Supply Disruption Day 50, Severity 0.1:**

     – Average Price Before Disruption: $100.25
     – Average Price After Disruption: $110.50

     Analysis: A mild disruption early in the simulation causes a noticeable but moderate increase in prices, indicating a temporary market reaction.

   - **Supply Disruption Day 150, Severity 0.3:**

     – Average Price Before Disruption: $100.40
     – Average Price After Disruption: $130.75

     Analysis: A significant disruption mid-way through the simulation results in a sharp price increase, highlighting the market's sensitivity to supply changes.

   - **Supply Disruption Day 200, Severity 0.5:**

     – Average Price Before Disruption: $100.60
     – Average Price After Disruption: $160.30

     Analysis: A severe disruption later in the simulation causes a dramatic rise in prices, emphasising the substantial impact of high-severity supply disruptions on the market.

Task 3: (Optional) Explore Simple Hedging Strategies

Objective: Explore simple strategies a business might use to hedge against price fluctuations.

Approach:

1. **Choose Strategies:** Research and select common hedging strategies like futures contracts, options, or inventory stockpiling.
2. **Model Strategies:** Modify the simulation to incorporate these strategies, making assumptions about their costs and effectiveness.
3. **Evaluate:**

   - **Price Stability:** Compare the volatility of resource prices with and without the hedging strategies.
   - **Cost-Benefit Analysis:** Calculate the overall cost of each strategy and compare it to the potential savings from reduced price volatility.

- **Interpretation:** Discuss the pros and cons of each strategy. Which one(s) seem most suitable for different business scenarios?

Pseudo Code

```
hedging_strategies = ['future_contracts', 'options', 'diversification']
hedging_costs = {'future_contracts': 500, 'options': 300, 'diversification': 700}

for strategy in hedging_strategies:
    sim = ResourceSimulation(start_price=100, days=250, volatility=0.02, drift=0.0003)
    prices = sim.run_simulation()
    hedged_prices = apply_hedging(prices, strategy)  # Apply and analyse hedging strategy
    plot_hedging_results(hedged_prices, strategy)  # Visualise effectiveness of strategies
    total_cost = hedging_costs[strategy]
    evaluate_hedging_effectiveness(prices, hedged_prices, total_cost)  # Calculate and displa
```

Implementation

1. **Setup Simulation with Hedging Strategies**

```
def apply_hedging(prices, strategy):
    if strategy == 'future_contracts':
        hedged_prices = prices * 0.95  # Simple reduction in price variability
    elif strategy == 'options':
        hedged_prices = prices * 0.97  # Slight reduction in price variability
    elif strategy == 'diversification':
        hedged_prices = prices * 0.93  # Significant reduction in price variability
    return hedged_prices

def evaluate_hedging_effectiveness(prices, hedged_prices, cost):
    price_variability = prices.std()
    hedged_variability = hedged_prices.std()
    reduction_in_variability = price_variability - hedged_variability
    roi = reduction_in_variability / cost
    print(f'Reduction in Variability: {reduction_in_variability:.2f}')
    print(f'ROI: {roi:.2f}')

def plot_hedging_results(hedged_prices, strategy):
    plt.figure(figsize=(10, 6))
    plt.plot(hedged_prices, label=f'Strategy: {strategy}')
    plt.xlabel('Days')
    plt.ylabel('Price')
    plt.title(f'Hedging Strategy: {strategy}')
    plt.legend()
    plt.show()

hedging_strategies = ['future_contracts', 'options', 'diversification']
hedging_costs = {'future_contracts': 500, 'options': 300, 'diversification': 700}
for strategy in hedging_strategies:
    sim = ResourceSimulation(start_price=100, days=250, volatility=0.02, drift=0.0003)
    prices = sim.run_simulation()
    hedged_prices = apply_hedging(prices, strategy
```