

# Assignments in this Course

## Management Science

### Welcome

This course uses two practical assignments to help you learn how to think like an optimization-minded analyst. Your focus throughout should be: understand the problem, design a sensible algorithmic approach, interpret results, and translate findings into clear business recommendations. Perfect code is not the goal—insight is.



Tip

You're encouraged to use AI tools responsibly. Treat them like a smart pair-programmer that helps you learn and debug, but make sure you understand all code you submit and can explain it.

### The Two Assignments

Assignment 1: Risk Analysis & Forecasting (Due at the start of Lecture 7, weight 30%)

You'll simulate investment outcomes to understand risk and then forecast demand using simple time-series techniques. Expect to implement Monte Carlo simulations, moving averages, and exponential smoothing, then compare methods and justify a recommendation.

Assignment 2: Optimization in Practice (Due at the start of Lecture 10, weight 20%)

You'll build and improve delivery routes and construct valid staff schedules under constraints, then iterate to improve quality and satisfaction. Expect to implement heuristics, evaluate improvements, visualize routes/schedules, and discuss trade-offs.

Work is in groups for both assignments. Typical time commitment is 4–6 hours for Assignment 1 and 5–7 hours for Assignment 2.

### How You'll Be Graded

#### Technical Implementation (60% of assignment grade)

- Excellent (90-100%): Code runs correctly, handles edge cases, well-structured
- Good (80-89%): Code runs with minor issues, mostly well-structured
- Satisfactory (70-79%): Code runs but has some problems, basic structure
- Needs Improvement (60-69%): Code has significant issues but shows understanding
- Inadequate (<60%): Code doesn't run or shows fundamental misunderstanding

## Business Insight (20% of assignment grade)

- Excellent: Clear actionable recommendations with quantified impact
- Good: Solid insights with some quantification
- Satisfactory: Basic insights communicated clearly
- Needs Improvement: Insights unclear or not business-relevant
- Inadequate: No meaningful insights provided

## Presentation Quality (20% of assignment grade)

- Excellent: Compelling visualizations, clear narrative
- Good: Good visuals, mostly clear communication
- Satisfactory: Basic visualizations, understandable presentation
- Needs Improvement: Poor visuals or unclear communication
- Inadequate: No effective communication of results

### ! Important

Use AI to clarify concepts, explore alternatives, and debug. Please, do not submit code you don't understand. You're responsible for every line. If you draw from external sources, cite them briefly in your notebook.

## Tips for Success

- Start simple, then iterate. Get a working baseline first.
- Check your logic with small tests and intermediate prints.
- Keep business questions in view—your job is to inform a decision.

### 💡 Tip

Common pitfalls to watch for: forgetting return-to-depot in routing, violating constraints after a change, optimizing forever instead of delivering “good enough,” and failing to actually answer the business questions.

## Submission Requirements

Submit one Jupyter notebook per assignment that includes:

- Your commented code with outputs and clear visualizations.
- Short, direct answers to the business questions.

File naming:

- Assignment 1: Assignment\_01\_[GroupNames].ipynb
- Assignment 2: Assignment\_02\_[GroupNames].ipynb

## Late Policy

Submissions lose 10% per day late, up to a maximum of three days. Exceptions are granted only for documented emergencies.

## Bibliography