**Title of the Document**

*Author Name*

**Table of Contents**

## Introduction to the NESA Stage 4 Data Science Program

Program Title: Data Science: Unlocking Insights in the Real World

Target Audience: Year 8 Students (Ages 13-14) - Selective School Edition

Program Overview:

Welcome to the Year 8 Data Science program, an exciting and innovative course designed to equip students with essential skills for the 21st century. In today's data-rich world, the ability to understand, analyse, and interpret data is no longer a niche skill but a fundamental competency. This program provides a foundational understanding of data science principles and practices, empowering students to become data-literate citizens and future innovators.

Alignment with NSW NESA Stage 4 Science Syllabus and Australian Curriculum: Science (Version 8.4):

This program is meticulously designed to align with the NSW Stage 4 Science Syllabus, specifically addressing the "Data Science 1" focus area (SC4-DA1) and the broader Working Scientifically skills (SC4-WS). It also strongly integrates with the Australian Curriculum: Science (Version 8.4), particularly the Science Inquiry Skills strand.

Key Curriculum Links:

- NSW Stage 4 Science Syllabus - Data Science 1:  
  
SC4-DA1-01: Explains how data is used by scientists to model and predict scientific phenomena.  
SC4-WS-06: Uses data to identify trends, patterns and relationships, and draw conclusions.  
SC4-WS-07: Identifies problem-solving strategies and proposes solutions.  
Throughout the program, students will develop skills across all Working Scientifically outcomes (SC4-WS-01 to SC4-WS-08), including observing, questioning, planning investigations, processing and analysing data, problem-solving, communicating, and reflecting on scientific methods.

- SC4-DA1-01: Explains how data is used by scientists to model and predict scientific phenomena.

- SC4-WS-06: Uses data to identify trends, patterns and relationships, and draw conclusions.

- SC4-WS-07: Identifies problem-solving strategies and proposes solutions.

- Throughout the program, students will develop skills across all Working Scientifically outcomes (SC4-WS-01 to SC4-WS-08), including observing, questioning, planning investigations, processing and analysing data, problem-solving, communicating, and reflecting on scientific methods.

- Australian Curriculum: Science (Version 8.4) - Science Inquiry Skills (Year 7-8):  
  
ACSIS124: Formulating questions or hypotheses that can be investigated scientifically.  
ACSIS125: Planning, selecting and using appropriate investigation types, including fair tests and controlled experiments, to collect reliable data; assessing risk and addressing ethical considerations.  
ACSIS126: Processing, analysing and evaluating data; identifying patterns and relationships, and summarising data using a range of representations.  
ACSIS127: Communicating ideas, findings and evidence-based conclusions using scientific language, and appropriate representations.  
ACSIS131: Evaluating investigation methods and conclusions, including identifying limitations and suggesting improvements.

- ACSIS124: Formulating questions or hypotheses that can be investigated scientifically.

- ACSIS125: Planning, selecting and using appropriate investigation types, including fair tests and controlled experiments, to collect reliable data; assessing risk and addressing ethical considerations.

- ACSIS126: Processing, analysing and evaluating data; identifying patterns and relationships, and summarising data using a range of representations.

- ACSIS127: Communicating ideas, findings and evidence-based conclusions using scientific language, and appropriate representations.

- ACSIS131: Evaluating investigation methods and conclusions, including identifying limitations and suggesting improvements.

- SC4-DA1-01: Explains how data is used by scientists to model and predict scientific phenomena.

- SC4-WS-06: Uses data to identify trends, patterns and relationships, and draw conclusions.

- SC4-WS-07: Identifies problem-solving strategies and proposes solutions.

- Throughout the program, students will develop skills across all Working Scientifically outcomes (SC4-WS-01 to SC4-WS-08), including observing, questioning, planning investigations, processing and analysing data, problem-solving, communicating, and reflecting on scientific methods.

- ACSIS124: Formulating questions or hypotheses that can be investigated scientifically.

- ACSIS125: Planning, selecting and using appropriate investigation types, including fair tests and controlled experiments, to collect reliable data; assessing risk and addressing ethical considerations.

- ACSIS126: Processing, analysing and evaluating data; identifying patterns and relationships, and summarising data using a range of representations.

- ACSIS127: Communicating ideas, findings and evidence-based conclusions using scientific language, and appropriate representations.

- ACSIS131: Evaluating investigation methods and conclusions, including identifying limitations and suggesting improvements.

Pedagogical Logic: Real-World Analysis with Modern Tooling

This program is built upon a pedagogical approach that emphasizes:

- Inquiry-Based Learning: Students are active learners, encouraged to ask questions, investigate phenomena, and discover insights through data exploration and analysis. The program moves beyond rote memorization to foster genuine understanding and critical thinking.

- Hands-On Activities: Learning is grounded in practical experience. Students engage in hands-on data collection, manipulation, visualization, and analysis activities, reinforcing theoretical concepts through direct application.

- Real-World Relevance: The program connects data science concepts to real-world problems and applications across diverse scientific and societal domains. This approach enhances student engagement by demonstrating the practical value and impact of data science in addressing contemporary challenges and making discoveries.

- Development of 21st-Century Skills: The program explicitly cultivates essential 21st-century skills, including:  
  
Digital Literacy: Students become proficient in using modern digital tools for data science, including Observable notebooks and Python programming.  
Computational Thinking: Students develop computational thinking skills through data analysis, algorithmic thinking (implicitly through AI interaction and potentially simplified coding), and problem-solving using computational methods.  
Critical Thinking and Data-Driven Reasoning: Students learn to evaluate data critically, identify biases, draw evidence-based conclusions, and understand the limitations of data and models.  
Communication and Collaboration: Students develop communication skills through data visualization and presentation, and collaboration skills through group projects and peer feedback activities.

- Digital Literacy: Students become proficient in using modern digital tools for data science, including Observable notebooks and Python programming.

- Computational Thinking: Students develop computational thinking skills through data analysis, algorithmic thinking (implicitly through AI interaction and potentially simplified coding), and problem-solving using computational methods.

- Critical Thinking and Data-Driven Reasoning: Students learn to evaluate data critically, identify biases, draw evidence-based conclusions, and understand the limitations of data and models.

- Communication and Collaboration: Students develop communication skills through data visualization and presentation, and collaboration skills through group projects and peer feedback activities.

- Gradual Skill Development: The curriculum follows a structured progression, starting with foundational concepts of data and data science in Week 1 and gradually building towards more advanced topics like data wrangling, statistical analysis, and an introduction to AI in data analysis by Week 7. This scaffolded approach ensures students develop a solid understanding and skillset incrementally.

- Integration of Modern Tools: The program leverages cutting-edge, industry-standard tools to provide students with authentic data science experiences:  
  
Observable Framework: Provides an interactive, web-based environment for data exploration, visualization, and computational notebooks. Its real-time interactivity and collaborative features enhance student engagement and learning.  
Python Programming: Introduces Python as a powerful yet accessible language for data manipulation, analysis, and visualization. Students gain practical coding skills that are highly valuable in data science and STEM fields.  
AI Tutor Integration: Incorporates an AI Tutor as a personalized learning assistant, providing students with on-demand support, feedback, and guidance. This AI integration aims to enhance personalized learning and cater to diverse student needs.

- Observable Framework: Provides an interactive, web-based environment for data exploration, visualization, and computational notebooks. Its real-time interactivity and collaborative features enhance student engagement and learning.

- Python Programming: Introduces Python as a powerful yet accessible language for data manipulation, analysis, and visualization. Students gain practical coding skills that are highly valuable in data science and STEM fields.

- AI Tutor Integration: Incorporates an AI Tutor as a personalized learning assistant, providing students with on-demand support, feedback, and guidance. This AI integration aims to enhance personalized learning and cater to diverse student needs.

- Digital Literacy: Students become proficient in using modern digital tools for data science, including Observable notebooks and Python programming.

- Computational Thinking: Students develop computational thinking skills through data analysis, algorithmic thinking (implicitly through AI interaction and potentially simplified coding), and problem-solving using computational methods.

- Critical Thinking and Data-Driven Reasoning: Students learn to evaluate data critically, identify biases, draw evidence-based conclusions, and understand the limitations of data and models.

- Communication and Collaboration: Students develop communication skills through data visualization and presentation, and collaboration skills through group projects and peer feedback activities.

- Observable Framework: Provides an interactive, web-based environment for data exploration, visualization, and computational notebooks. Its real-time interactivity and collaborative features enhance student engagement and learning.

- Python Programming: Introduces Python as a powerful yet accessible language for data manipulation, analysis, and visualization. Students gain practical coding skills that are highly valuable in data science and STEM fields.

- AI Tutor Integration: Incorporates an AI Tutor as a personalized learning assistant, providing students with on-demand support, feedback, and guidance. This AI integration aims to enhance personalized learning and cater to diverse student needs.

Program Goal:

The overarching goal of this Year 8 Data Science program is to empower students to become confident, data-literate individuals who can:

- Understand the role of data in scientific inquiry and real-world problem-solving.

- Apply data science principles and techniques to investigate scientific questions.

- Effectively communicate data-driven insights using visualizations and evidence-based reasoning.

- Critically evaluate data and models, understanding their strengths and limitations.

- Appreciate the ethical considerations of data collection and use in a digital society.

By the end of this program, students will be well-prepared to engage with data critically and creatively, fostering a lifelong curiosity for scientific inquiry and a strong foundation for future STEM studies and careers in a data-driven world.

## 2. 10 Week Learning Sequence

Overall Theme: Data Science: Unlocking Insights in the Real World

Week 1: Data Science Foundations & Digital Immersion

- Focus: Rapid Introduction to Data Science, Digital Tool Immersion (Observable, Python, AI Tutor).

- Content:  
  
What is Data Science? Applications in cutting-edge science.  
Data Sources: Diverse scientific data sources (real-time, simulations, large datasets).  
Observable Notebooks: Intensive introduction - navigation, markdown/code cells, Python basics, sharing.  
AI Tutor: Introduction and practical exercises for Q&A and code assistance.

- What is Data Science? Applications in cutting-edge science.

- Data Sources: Diverse scientific data sources (real-time, simulations, large datasets).

- Observable Notebooks: Intensive introduction - navigation, markdown/code cells, Python basics, sharing.

- AI Tutor: Introduction and practical exercises for Q&A and code assistance.

- What is Data Science? Applications in cutting-edge science.

- Data Sources: Diverse scientific data sources (real-time, simulations, large datasets).

- Observable Notebooks: Intensive introduction - navigation, markdown/code cells, Python basics, sharing.

- AI Tutor: Introduction and practical exercises for Q&A and code assistance.

Week 2: Data Collection Mastery & Digital Responsibility

- Focus: Data Collection Techniques (rigorous methods), Digital Footprint & Ethics, Accuracy, Precision, Validity (data quality).

- Content:  
  
Lesson 1: Digital Footprint & Data Ethics: Online safety, privacy, responsible data use, ethical considerations.  
Lessons 2 & 3: Data Collection & Quality:  
  
Primary and Secondary Data, data collection methods design.  
Accuracy, Precision, Validity: Practical exercises, error analysis, experimental design for data quality.  
Digital Data Capture in Observable.

- Lesson 1: Digital Footprint & Data Ethics: Online safety, privacy, responsible data use, ethical considerations.

- Lessons 2 & 3: Data Collection & Quality:  
  
Primary and Secondary Data, data collection methods design.  
Accuracy, Precision, Validity: Practical exercises, error analysis, experimental design for data quality.  
Digital Data Capture in Observable.

- Primary and Secondary Data, data collection methods design.

- Accuracy, Precision, Validity: Practical exercises, error analysis, experimental design for data quality.

- Digital Data Capture in Observable.

- Lesson 1: Digital Footprint & Data Ethics: Online safety, privacy, responsible data use, ethical considerations.

- Lessons 2 & 3: Data Collection & Quality:  
  
Primary and Secondary Data, data collection methods design.  
Accuracy, Precision, Validity: Practical exercises, error analysis, experimental design for data quality.  
Digital Data Capture in Observable.

- Primary and Secondary Data, data collection methods design.

- Accuracy, Precision, Validity: Practical exercises, error analysis, experimental design for data quality.

- Digital Data Capture in Observable.

- Primary and Secondary Data, data collection methods design.

- Accuracy, Precision, Validity: Practical exercises, error analysis, experimental design for data quality.

- Digital Data Capture in Observable.

Week 3: Data Visualization & Interactive Storytelling

- Focus: Data Visualization Techniques (graph types), Interactive Dashboards in Observable, Visual Clarity & Data Storytelling.

- Content:  
  
Types of Graphs: Comprehensive exploration, strengths/weaknesses for data types.  
Python Plotting Libraries (matplotlib, Plotly, Seaborn): Hands-on visualization creation in Observable.  
Interactive Elements: Dashboards with sliders, dropdowns, tooltips in Observable.  
Peer Review & Critique of data visualizations.

- Types of Graphs: Comprehensive exploration, strengths/weaknesses for data types.

- Python Plotting Libraries (matplotlib, Plotly, Seaborn): Hands-on visualization creation in Observable.

- Interactive Elements: Dashboards with sliders, dropdowns, tooltips in Observable.

- Peer Review & Critique of data visualizations.

- Types of Graphs: Comprehensive exploration, strengths/weaknesses for data types.

- Python Plotting Libraries (matplotlib, Plotly, Seaborn): Hands-on visualization creation in Observable.

- Interactive Elements: Dashboards with sliders, dropdowns, tooltips in Observable.

- Peer Review & Critique of data visualizations.

Week 4: Descriptive Statistics & Data Interpretation

- Focus: Descriptive Statistics Mastery (mean, median, mode, standard deviation, variance, percentiles), Data Distribution & Outliers, Statistical Interpretation.

- Content:  
  
Central Tendency & Dispersion: In-depth exploration, formulas, appropriate use.  
Data Distribution: Histograms, frequency distributions, normal distribution, skewness, kurtosis (conceptual).  
Outlier Detection & Analysis: Methods for identification, causes, handling outliers.  
Python for Statistical Calculation (NumPy, SciPy) in Observable.

- Central Tendency & Dispersion: In-depth exploration, formulas, appropriate use.

- Data Distribution: Histograms, frequency distributions, normal distribution, skewness, kurtosis (conceptual).

- Outlier Detection & Analysis: Methods for identification, causes, handling outliers.

- Python for Statistical Calculation (NumPy, SciPy) in Observable.

- Central Tendency & Dispersion: In-depth exploration, formulas, appropriate use.

- Data Distribution: Histograms, frequency distributions, normal distribution, skewness, kurtosis (conceptual).

- Outlier Detection & Analysis: Methods for identification, causes, handling outliers.

- Python for Statistical Calculation (NumPy, SciPy) in Observable.

Week 5: Scientific Question Formulation & Experimental Design - Advanced

- Focus: Formulating Complex Scientific Questions, Advanced Experimental Design, Hypothesis Testing & Prediction.

- Content:  
  
Question Types: Descriptive, comparative, correlational, causal – design for each.  
Experimental Design Principles: Control groups, variables, randomization, replication, blinding.  
Virtual Experiments & Simulations in Observable for hypothesis testing.  
AI Scenario Generator for complex experimental design challenges.

- Question Types: Descriptive, comparative, correlational, causal – design for each.

- Experimental Design Principles: Control groups, variables, randomization, replication, blinding.

- Virtual Experiments & Simulations in Observable for hypothesis testing.

- AI Scenario Generator for complex experimental design challenges.

- Question Types: Descriptive, comparative, correlational, causal – design for each.

- Experimental Design Principles: Control groups, variables, randomization, replication, blinding.

- Virtual Experiments & Simulations in Observable for hypothesis testing.

- AI Scenario Generator for complex experimental design challenges.

Week 6: Advanced Data Wrangling & Real-World Datasets

- Focus: Handling messy real-world datasets, advanced data cleaning (Python/Pandas), data transformation, feature engineering (conceptual).

- Content:  
  
Working with real-world datasets (provided and student-sourced).  
Data Cleaning Techniques in Python (Pandas): Handling missing values, inconsistent formats, data validation.  
Data Transformation: Reshaping, merging, aggregating data.  
Feature Engineering (Conceptual): Creating new variables from existing data for better analysis.

- Working with real-world datasets (provided and student-sourced).

- Data Cleaning Techniques in Python (Pandas): Handling missing values, inconsistent formats, data validation.

- Data Transformation: Reshaping, merging, aggregating data.

- Feature Engineering (Conceptual): Creating new variables from existing data for better analysis.

- Working with real-world datasets (provided and student-sourced).

- Data Cleaning Techniques in Python (Pandas): Handling missing values, inconsistent formats, data validation.

- Data Transformation: Reshaping, merging, aggregating data.

- Feature Engineering (Conceptual): Creating new variables from existing data for better analysis.

Week 7: AI-Powered Data Analysis & Predictive Modelling (Introduction)

- Focus: Introduction to AI in Data Analysis (pattern recognition, clustering, classification - conceptual), AI tools for data summarization and trend analysis, Basic Predictive Modelling (simplified introduction).

- Content:  
  
AI Concepts in Data Analysis: Pattern recognition, clustering, classification (conceptual overview).  
AI Tools for Data Analysis in Observable (if available/suitable - or explore external simplified AI tools).  
Simplified Introduction to Predictive Modelling: Basic concepts, visual tools (if feasible), understanding model limitations.  
Ethical considerations of AI in data analysis and prediction.

- AI Concepts in Data Analysis: Pattern recognition, clustering, classification (conceptual overview).

- AI Tools for Data Analysis in Observable (if available/suitable - or explore external simplified AI tools).

- Simplified Introduction to Predictive Modelling: Basic concepts, visual tools (if feasible), understanding model limitations.

- Ethical considerations of AI in data analysis and prediction.

- AI Concepts in Data Analysis: Pattern recognition, clustering, classification (conceptual overview).

- AI Tools for Data Analysis in Observable (if available/suitable - or explore external simplified AI tools).

- Simplified Introduction to Predictive Modelling: Basic concepts, visual tools (if feasible), understanding model limitations.

- Ethical considerations of AI in data analysis and prediction.

Week 8: Group Project - Real-World Data Science Challenge

- Focus: Collaboration, problem-solving, applying data science to real-world scientific/societal issues, scientific communication.

- Content:  
  
Team-based projects: Students select a real-world problem and relevant dataset.  
Project Stages: Question formulation, data acquisition, data analysis, visualization, model building (if applicable), presentation.  
Emphasis on collaborative work, project management, and effective communication of findings.  
Regular AI-driven feedback on project progress and data analysis.

- Team-based projects: Students select a real-world problem and relevant dataset.

- Project Stages: Question formulation, data acquisition, data analysis, visualization, model building (if applicable), presentation.

- Emphasis on collaborative work, project management, and effective communication of findings.

- Regular AI-driven feedback on project progress and data analysis.

- Team-based projects: Students select a real-world problem and relevant dataset.

- Project Stages: Question formulation, data acquisition, data analysis, visualization, model building (if applicable), presentation.

- Emphasis on collaborative work, project management, and effective communication of findings.

- Regular AI-driven feedback on project progress and data analysis.

Week 9: Project Refinement & Advanced Review

- Focus: Refining project analysis, visuals, presentations, Comprehensive knowledge checks, Exam preparation.

- Content:  
  
Peer and AI Review of group projects: Iterative improvement based on feedback.  
Advanced Practice Exam Questions: Challenging data analysis problems mimicking exam format.  
Focus on consolidating data science skills and preparing for summative assessment.  
Refinement of Observable notebooks and project presentations.

- Peer and AI Review of group projects: Iterative improvement based on feedback.

- Advanced Practice Exam Questions: Challenging data analysis problems mimicking exam format.

- Focus on consolidating data science skills and preparing for summative assessment.

- Refinement of Observable notebooks and project presentations.

- Peer and AI Review of group projects: Iterative improvement based on feedback.

- Advanced Practice Exam Questions: Challenging data analysis problems mimicking exam format.

- Focus on consolidating data science skills and preparing for summative assessment.

- Refinement of Observable notebooks and project presentations.

Week 10: End-of-Semester Exam & Future Pathways

- Focus: Formal assessment of data analysis, visualization, coding, scientific reasoning, Self-evaluation and future pathways.

- Content:  
  
End-of-Semester Exam:  
  
Section 1: Data Interpretation (graphs, statistics, short-answer questions).  
Section 2: Coding and Notebook Tasks (Python in Observable).  
Section 3: Real-World Problem Scenarios (evidence-based problem-solving).  
  
Exam review and feedback.  
Discussion of future pathways in data science and related STEM fields.  
Student self-reflection on learning and skill development.

- End-of-Semester Exam:  
  
Section 1: Data Interpretation (graphs, statistics, short-answer questions).  
Section 2: Coding and Notebook Tasks (Python in Observable).  
Section 3: Real-World Problem Scenarios (evidence-based problem-solving).

- Section 1: Data Interpretation (graphs, statistics, short-answer questions).

- Section 2: Coding and Notebook Tasks (Python in Observable).

- Section 3: Real-World Problem Scenarios (evidence-based problem-solving).

- Exam review and feedback.

- Discussion of future pathways in data science and related STEM fields.

- Student self-reflection on learning and skill development.

- End-of-Semester Exam:  
  
Section 1: Data Interpretation (graphs, statistics, short-answer questions).  
Section 2: Coding and Notebook Tasks (Python in Observable).  
Section 3: Real-World Problem Scenarios (evidence-based problem-solving).

- Section 1: Data Interpretation (graphs, statistics, short-answer questions).

- Section 2: Coding and Notebook Tasks (Python in Observable).

- Section 3: Real-World Problem Scenarios (evidence-based problem-solving).

- Exam review and feedback.

- Discussion of future pathways in data science and related STEM fields.

- Student self-reflection on learning and skill development.

- Section 1: Data Interpretation (graphs, statistics, short-answer questions).

- Section 2: Coding and Notebook Tasks (Python in Observable).

- Section 3: Real-World Problem Scenarios (evidence-based problem-solving).

## 3. 3x10 - One-Hour Data Science Lesson Plans

### Week 1: Data Science Foundations & Digital Immersion

### Lesson 1.1: Introduction to Data Science: What, Why, and Where?

- Title: Data Science: Unveiling the Power of Data in the 21st Century

- Learning Outcomes:  
  
Students will be able to define data science and explain its interdisciplinary nature.  
Students will be able to identify key applications of data science in various scientific fields and real-world scenarios.  
Students will understand the increasing importance of data science in modern society and scientific discovery.  
NSW Curriculum Outcomes: SC4-DA1-01 (explains how data is used), SC4-WS-07 (identifies problem-solving strategies)

- Students will be able to define data science and explain its interdisciplinary nature.

- Students will be able to identify key applications of data science in various scientific fields and real-world scenarios.

- Students will understand the increasing importance of data science in modern society and scientific discovery.

- NSW Curriculum Outcomes: SC4-DA1-01 (explains how data is used), SC4-WS-07 (identifies problem-solving strategies)

- Overview: This introductory lesson provides a compelling overview of data science, its definition, interdisciplinary nature, and significant applications, emphasizing its relevance in today's world and scientific advancements.

- Activities:  
  
Warm-Up (10 mins): "Data Science Headlines" - Show a series of news headlines or short summaries related to data science breakthroughs in different fields (e.g., AI in medicine, climate change modelling, astronomical discoveries, personalized learning). Ask students: "What do these seemingly different stories have in common?" Lead to the idea of data being central.  
Main Activity 1: "Defining Data Science - The Interdisciplinary Web" (25 mins):  
  
Introduce the definition of Data Science: "an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from data in various forms."  
Visually represent Data Science as a hub with connections to different disciplines: Statistics, Computer Science, Domain Expertise (e.g., Biology, Physics, Social Sciences), Communication, Ethics.  
Discuss each connection: How does Statistics contribute? Computer Science? Why is domain expertise crucial? Why is communication and ethics essential in data science?  
  
Main Activity 2: "Data Science in Action - Real-World Impact" (25 mins):  
  
Present a series of short case studies showcasing the impact of data science in diverse fields (tailored to be engaging for selective students):  
  
Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).  
Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).  
Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).  
Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).  
  
For each case study, guide students to identify:  
  
What is the scientific problem being addressed?  
What types of data are being used?  
How is data science contributing to solving the problem or making new discoveries?  
  
Brief group discussions on each case study, followed by class sharing.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why is data science considered a "game-changer" in science and many other fields?  
What are some of the most exciting potential future applications of data science you can imagine?  
What skills and qualities do you think are important to succeed in data science?  
  
Assessments:  
  
Formative Quiz (5 mins): Short multiple-choice quiz on the definition of data science, its interdisciplinary nature, and key application areas. AI-generated quiz questions.  
Observation of student participation in discussions and group activities.  
  
Student Workbook:  
  
Definition of Data Science and its interdisciplinary components (diagram).  
Summary table for case studies, identifying the problem, data types, and data science impact.  
Reflection questions on the importance and future of data science.  
  
Extension Activities:  
  
Students research a specific data scientist working in a field that interests them and create a short presentation or profile.  
Challenge: Debate: "Is data science just a new name for statistics, or is it a fundamentally different field?"

- Warm-Up (10 mins): "Data Science Headlines" - Show a series of news headlines or short summaries related to data science breakthroughs in different fields (e.g., AI in medicine, climate change modelling, astronomical discoveries, personalized learning). Ask students: "What do these seemingly different stories have in common?" Lead to the idea of data being central.

- Main Activity 1: "Defining Data Science - The Interdisciplinary Web" (25 mins):  
  
Introduce the definition of Data Science: "an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from data in various forms."  
Visually represent Data Science as a hub with connections to different disciplines: Statistics, Computer Science, Domain Expertise (e.g., Biology, Physics, Social Sciences), Communication, Ethics.  
Discuss each connection: How does Statistics contribute? Computer Science? Why is domain expertise crucial? Why is communication and ethics essential in data science?

- Introduce the definition of Data Science: "an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from data in various forms."

- Visually represent Data Science as a hub with connections to different disciplines: Statistics, Computer Science, Domain Expertise (e.g., Biology, Physics, Social Sciences), Communication, Ethics.

- Discuss each connection: How does Statistics contribute? Computer Science? Why is domain expertise crucial? Why is communication and ethics essential in data science?

- Main Activity 2: "Data Science in Action - Real-World Impact" (25 mins):  
  
Present a series of short case studies showcasing the impact of data science in diverse fields (tailored to be engaging for selective students):  
  
Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).  
Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).  
Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).  
Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).  
  
For each case study, guide students to identify:  
  
What is the scientific problem being addressed?  
What types of data are being used?  
How is data science contributing to solving the problem or making new discoveries?  
  
Brief group discussions on each case study, followed by class sharing.

- Present a series of short case studies showcasing the impact of data science in diverse fields (tailored to be engaging for selective students):  
  
Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).  
Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).  
Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).  
Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).

- Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).

- Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).

- Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).

- Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).

- For each case study, guide students to identify:  
  
What is the scientific problem being addressed?  
What types of data are being used?  
How is data science contributing to solving the problem or making new discoveries?

- What is the scientific problem being addressed?

- What types of data are being used?

- How is data science contributing to solving the problem or making new discoveries?

- Brief group discussions on each case study, followed by class sharing.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is data science considered a "game-changer" in science and many other fields?  
What are some of the most exciting potential future applications of data science you can imagine?  
What skills and qualities do you think are important to succeed in data science?

- Why is data science considered a "game-changer" in science and many other fields?

- What are some of the most exciting potential future applications of data science you can imagine?

- What skills and qualities do you think are important to succeed in data science?

- Assessments:  
  
Formative Quiz (5 mins): Short multiple-choice quiz on the definition of data science, its interdisciplinary nature, and key application areas. AI-generated quiz questions.  
Observation of student participation in discussions and group activities.

- Formative Quiz (5 mins): Short multiple-choice quiz on the definition of data science, its interdisciplinary nature, and key application areas. AI-generated quiz questions.

- Observation of student participation in discussions and group activities.

- Student Workbook:  
  
Definition of Data Science and its interdisciplinary components (diagram).  
Summary table for case studies, identifying the problem, data types, and data science impact.  
Reflection questions on the importance and future of data science.

- Definition of Data Science and its interdisciplinary components (diagram).

- Summary table for case studies, identifying the problem, data types, and data science impact.

- Reflection questions on the importance and future of data science.

- Extension Activities:  
  
Students research a specific data scientist working in a field that interests them and create a short presentation or profile.  
Challenge: Debate: "Is data science just a new name for statistics, or is it a fundamentally different field?"

- Students research a specific data scientist working in a field that interests them and create a short presentation or profile.

- Challenge: Debate: "Is data science just a new name for statistics, or is it a fundamentally different field?"

- Lesson Materials:  
  
Slide Deck: Introduction to Data Science, definition, interdisciplinary web diagram, case study examples (with visuals).  
Handout: Case study descriptions, summary table for case study analysis, definition of data science.  
Formative Quiz (printable or digital).  
(Optional) Short video clips showcasing data science applications.

- Slide Deck: Introduction to Data Science, definition, interdisciplinary web diagram, case study examples (with visuals).

- Handout: Case study descriptions, summary table for case study analysis, definition of data science.

- Formative Quiz (printable or digital).

- (Optional) Short video clips showcasing data science applications.

- Students will be able to define data science and explain its interdisciplinary nature.

- Students will be able to identify key applications of data science in various scientific fields and real-world scenarios.

- Students will understand the increasing importance of data science in modern society and scientific discovery.

- NSW Curriculum Outcomes: SC4-DA1-01 (explains how data is used), SC4-WS-07 (identifies problem-solving strategies)

- Warm-Up (10 mins): "Data Science Headlines" - Show a series of news headlines or short summaries related to data science breakthroughs in different fields (e.g., AI in medicine, climate change modelling, astronomical discoveries, personalized learning). Ask students: "What do these seemingly different stories have in common?" Lead to the idea of data being central.

- Main Activity 1: "Defining Data Science - The Interdisciplinary Web" (25 mins):  
  
Introduce the definition of Data Science: "an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from data in various forms."  
Visually represent Data Science as a hub with connections to different disciplines: Statistics, Computer Science, Domain Expertise (e.g., Biology, Physics, Social Sciences), Communication, Ethics.  
Discuss each connection: How does Statistics contribute? Computer Science? Why is domain expertise crucial? Why is communication and ethics essential in data science?

- Introduce the definition of Data Science: "an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from data in various forms."

- Visually represent Data Science as a hub with connections to different disciplines: Statistics, Computer Science, Domain Expertise (e.g., Biology, Physics, Social Sciences), Communication, Ethics.

- Discuss each connection: How does Statistics contribute? Computer Science? Why is domain expertise crucial? Why is communication and ethics essential in data science?

- Main Activity 2: "Data Science in Action - Real-World Impact" (25 mins):  
  
Present a series of short case studies showcasing the impact of data science in diverse fields (tailored to be engaging for selective students):  
  
Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).  
Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).  
Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).  
Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).  
  
For each case study, guide students to identify:  
  
What is the scientific problem being addressed?  
What types of data are being used?  
How is data science contributing to solving the problem or making new discoveries?  
  
Brief group discussions on each case study, followed by class sharing.

- Present a series of short case studies showcasing the impact of data science in diverse fields (tailored to be engaging for selective students):  
  
Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).  
Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).  
Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).  
Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).

- Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).

- Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).

- Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).

- Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).

- For each case study, guide students to identify:  
  
What is the scientific problem being addressed?  
What types of data are being used?  
How is data science contributing to solving the problem or making new discoveries?

- What is the scientific problem being addressed?

- What types of data are being used?

- How is data science contributing to solving the problem or making new discoveries?

- Brief group discussions on each case study, followed by class sharing.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is data science considered a "game-changer" in science and many other fields?  
What are some of the most exciting potential future applications of data science you can imagine?  
What skills and qualities do you think are important to succeed in data science?

- Why is data science considered a "game-changer" in science and many other fields?

- What are some of the most exciting potential future applications of data science you can imagine?

- What skills and qualities do you think are important to succeed in data science?

- Assessments:  
  
Formative Quiz (5 mins): Short multiple-choice quiz on the definition of data science, its interdisciplinary nature, and key application areas. AI-generated quiz questions.  
Observation of student participation in discussions and group activities.

- Formative Quiz (5 mins): Short multiple-choice quiz on the definition of data science, its interdisciplinary nature, and key application areas. AI-generated quiz questions.

- Observation of student participation in discussions and group activities.

- Student Workbook:  
  
Definition of Data Science and its interdisciplinary components (diagram).  
Summary table for case studies, identifying the problem, data types, and data science impact.  
Reflection questions on the importance and future of data science.

- Definition of Data Science and its interdisciplinary components (diagram).

- Summary table for case studies, identifying the problem, data types, and data science impact.

- Reflection questions on the importance and future of data science.

- Extension Activities:  
  
Students research a specific data scientist working in a field that interests them and create a short presentation or profile.  
Challenge: Debate: "Is data science just a new name for statistics, or is it a fundamentally different field?"

- Students research a specific data scientist working in a field that interests them and create a short presentation or profile.

- Challenge: Debate: "Is data science just a new name for statistics, or is it a fundamentally different field?"

- Introduce the definition of Data Science: "an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from data in various forms."

- Visually represent Data Science as a hub with connections to different disciplines: Statistics, Computer Science, Domain Expertise (e.g., Biology, Physics, Social Sciences), Communication, Ethics.

- Discuss each connection: How does Statistics contribute? Computer Science? Why is domain expertise crucial? Why is communication and ethics essential in data science?

- Present a series of short case studies showcasing the impact of data science in diverse fields (tailored to be engaging for selective students):  
  
Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).  
Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).  
Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).  
Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).

- Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).

- Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).

- Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).

- Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).

- For each case study, guide students to identify:  
  
What is the scientific problem being addressed?  
What types of data are being used?  
How is data science contributing to solving the problem or making new discoveries?

- What is the scientific problem being addressed?

- What types of data are being used?

- How is data science contributing to solving the problem or making new discoveries?

- Brief group discussions on each case study, followed by class sharing.

- Precision Medicine: How data science and AI are revolutionizing disease diagnosis, personalized treatment plans, and drug discovery (e.g., genomics, AI-powered diagnostics).

- Climate Change Modelling & Prediction: How data science is used to analyse vast climate datasets, build complex models, and predict future climate scenarios (e.g., climate simulations, extreme weather forecasting).

- Astrophysics & Big Data Astronomy: How data science enables astronomers to analyse massive datasets from telescopes, discover new celestial objects, and understand the universe (e.g., exoplanet discovery, galaxy evolution).

- Materials Science & Accelerated Discovery: How data science and machine learning are speeding up the discovery and design of new materials with specific properties (e.g., new battery materials, high-performance alloys).

- What is the scientific problem being addressed?

- What types of data are being used?

- How is data science contributing to solving the problem or making new discoveries?

- Why is data science considered a "game-changer" in science and many other fields?

- What are some of the most exciting potential future applications of data science you can imagine?

- What skills and qualities do you think are important to succeed in data science?

- Formative Quiz (5 mins): Short multiple-choice quiz on the definition of data science, its interdisciplinary nature, and key application areas. AI-generated quiz questions.

- Observation of student participation in discussions and group activities.

- Definition of Data Science and its interdisciplinary components (diagram).

- Summary table for case studies, identifying the problem, data types, and data science impact.

- Reflection questions on the importance and future of data science.

- Students research a specific data scientist working in a field that interests them and create a short presentation or profile.

- Challenge: Debate: "Is data science just a new name for statistics, or is it a fundamentally different field?"

- Slide Deck: Introduction to Data Science, definition, interdisciplinary web diagram, case study examples (with visuals).

- Handout: Case study descriptions, summary table for case study analysis, definition of data science.

- Formative Quiz (printable or digital).

- (Optional) Short video clips showcasing data science applications.

### Lesson 1.2: Digital Toolkit: Getting Started with Observable & AI Tutor

- Title: Your Data Science Lab: Navigating Observable and Meeting Your AI Assistant

- Learning Outcomes:  
  
Students will be able to create and navigate an Observable notebook environment.  
Students will be able to create and use markdown and code cells within Observable.  
Students will be able to execute basic Python code in Observable and view output.  
Students will be able to interact with the integrated AI Tutor for questions and assistance within Observable.  
NSW Curriculum Outcomes: SC4-WS-03 (Use digital technologies), SC4-WS-06 (Process data and information)

- Students will be able to create and navigate an Observable notebook environment.

- Students will be able to create and use markdown and code cells within Observable.

- Students will be able to execute basic Python code in Observable and view output.

- Students will be able to interact with the integrated AI Tutor for questions and assistance within Observable.

- NSW Curriculum Outcomes: SC4-WS-03 (Use digital technologies), SC4-WS-06 (Process data and information)

- Overview: This hands-on lesson introduces students to their primary digital tools: the Observable notebook and the AI Tutor. Students will learn to navigate Observable, create notebooks, use markdown and code cells, and start interacting with the AI Tutor for support.

- Activities:  
  
Warm-Up (10 mins): "Digital Tools We Use" - Quick brainstorm: "What digital tools do you already use for learning, creating, or exploring information?" (e.g., Google Docs, online simulations, coding platforms, online encyclopedias). Discuss: What makes these tools useful? What are the benefits of digital tools in learning? Lead to the idea of Observable as a powerful digital tool for data science.  
Main Activity 1: "Observable Interface Expedition" (25 mins):  
  
Guided, step-by-step walkthrough of the Observable interface:  
  
Account login and homepage overview.  
Navigating to "Notebooks" and creating a new notebook ("+ New").  
Introduction to the notebook interface: Title, cells (markdown and code), toolbar.  
Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).  
Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.  
Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).  
  
Encourage students to actively follow along and experiment within their own Observable notebooks as you demonstrate.  
  
Main Activity 2: "Meet Your AI Tutor - Ask Me Anything (Data Science Edition)" (20 mins):  
  
Introduce the integrated AI Tutor (specific implementation will depend on the chosen AI tool).  
Demonstrate how to access and interact with the AI Tutor (voice or chat interface).  
Guided practice using the AI Tutor:  
  
Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".  
Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".  
Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".  
  
Encourage students to ask their own questions and explore the AI Tutor's capabilities. Emphasize it's a tool for help and learning, not just getting answers.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are your first impressions of using Observable? What do you find interesting or useful?  
How do you think the AI Tutor can support your learning in data science? What are its potential benefits?  
What are you most looking forward to doing with Observable and data science in the coming weeks?  
  
Assessments:  
  
Formative Check (5 mins): Quickly review student Observable notebooks to ensure they have created a notebook, added markdown and code cells, and run basic code.  
Observe student interaction with the AI Tutor.  
  
Student Workbook:  
  
Step-by-step guide to creating an Observable account (if needed) and creating a basic notebook.  
Key features of the Observable interface (screenshot and labels).  
Basic markdown formatting guide.  
Instructions for accessing and using the AI Tutor with example questions.  
Practice exercises for creating markdown and code cells and running Python code.  
  
Extension Activities:  
  
Explore Observable's example notebooks in the "Explore" section to see more advanced uses of the platform.  
Challenge: Experiment with more advanced markdown formatting (links, images, tables). Try slightly more complex Python code in code cells (basic arithmetic, variable assignment).

- Warm-Up (10 mins): "Digital Tools We Use" - Quick brainstorm: "What digital tools do you already use for learning, creating, or exploring information?" (e.g., Google Docs, online simulations, coding platforms, online encyclopedias). Discuss: What makes these tools useful? What are the benefits of digital tools in learning? Lead to the idea of Observable as a powerful digital tool for data science.

- Main Activity 1: "Observable Interface Expedition" (25 mins):  
  
Guided, step-by-step walkthrough of the Observable interface:  
  
Account login and homepage overview.  
Navigating to "Notebooks" and creating a new notebook ("+ New").  
Introduction to the notebook interface: Title, cells (markdown and code), toolbar.  
Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).  
Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.  
Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).  
  
Encourage students to actively follow along and experiment within their own Observable notebooks as you demonstrate.

- Guided, step-by-step walkthrough of the Observable interface:  
  
Account login and homepage overview.  
Navigating to "Notebooks" and creating a new notebook ("+ New").  
Introduction to the notebook interface: Title, cells (markdown and code), toolbar.  
Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).  
Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.  
Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).

- Account login and homepage overview.

- Navigating to "Notebooks" and creating a new notebook ("+ New").

- Introduction to the notebook interface: Title, cells (markdown and code), toolbar.

- Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).

- Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.

- Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).

- Encourage students to actively follow along and experiment within their own Observable notebooks as you demonstrate.

- Main Activity 2: "Meet Your AI Tutor - Ask Me Anything (Data Science Edition)" (20 mins):  
  
Introduce the integrated AI Tutor (specific implementation will depend on the chosen AI tool).  
Demonstrate how to access and interact with the AI Tutor (voice or chat interface).  
Guided practice using the AI Tutor:  
  
Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".  
Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".  
Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".  
  
Encourage students to ask their own questions and explore the AI Tutor's capabilities. Emphasize it's a tool for help and learning, not just getting answers.

- Introduce the integrated AI Tutor (specific implementation will depend on the chosen AI tool).

- Demonstrate how to access and interact with the AI Tutor (voice or chat interface).

- Guided practice using the AI Tutor:  
  
Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".  
Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".  
Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".

- Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".

- Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".

- Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".

- Encourage students to ask their own questions and explore the AI Tutor's capabilities. Emphasize it's a tool for help and learning, not just getting answers.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are your first impressions of using Observable? What do you find interesting or useful?  
How do you think the AI Tutor can support your learning in data science? What are its potential benefits?  
What are you most looking forward to doing with Observable and data science in the coming weeks?

- What are your first impressions of using Observable? What do you find interesting or useful?

- How do you think the AI Tutor can support your learning in data science? What are its potential benefits?

- What are you most looking forward to doing with Observable and data science in the coming weeks?

- Assessments:  
  
Formative Check (5 mins): Quickly review student Observable notebooks to ensure they have created a notebook, added markdown and code cells, and run basic code.  
Observe student interaction with the AI Tutor.

- Formative Check (5 mins): Quickly review student Observable notebooks to ensure they have created a notebook, added markdown and code cells, and run basic code.

- Observe student interaction with the AI Tutor.

- Student Workbook:  
  
Step-by-step guide to creating an Observable account (if needed) and creating a basic notebook.  
Key features of the Observable interface (screenshot and labels).  
Basic markdown formatting guide.  
Instructions for accessing and using the AI Tutor with example questions.  
Practice exercises for creating markdown and code cells and running Python code.

- Step-by-step guide to creating an Observable account (if needed) and creating a basic notebook.

- Key features of the Observable interface (screenshot and labels).

- Basic markdown formatting guide.

- Instructions for accessing and using the AI Tutor with example questions.

- Practice exercises for creating markdown and code cells and running Python code.

- Extension Activities:  
  
Explore Observable's example notebooks in the "Explore" section to see more advanced uses of the platform.  
Challenge: Experiment with more advanced markdown formatting (links, images, tables). Try slightly more complex Python code in code cells (basic arithmetic, variable assignment).

- Explore Observable's example notebooks in the "Explore" section to see more advanced uses of the platform.

- Challenge: Experiment with more advanced markdown formatting (links, images, tables). Try slightly more complex Python code in code cells (basic arithmetic, variable assignment).

- Lesson Materials:  
  
Slide Deck: Introduction to Observable, interface tour, step-by-step guide for notebook creation, examples of markdown and code cells, introduction to AI Tutor.  
Handout: Observable guide, AI Tutor usage guide, practice exercises for Observable.  
Computers/Tablets with internet access for Observable.  
(Optional) Pre-lesson instructions for Observable account setup if needed.

- Slide Deck: Introduction to Observable, interface tour, step-by-step guide for notebook creation, examples of markdown and code cells, introduction to AI Tutor.

- Handout: Observable guide, AI Tutor usage guide, practice exercises for Observable.

- Computers/Tablets with internet access for Observable.

- (Optional) Pre-lesson instructions for Observable account setup if needed.

- Students will be able to create and navigate an Observable notebook environment.

- Students will be able to create and use markdown and code cells within Observable.

- Students will be able to execute basic Python code in Observable and view output.

- Students will be able to interact with the integrated AI Tutor for questions and assistance within Observable.

- NSW Curriculum Outcomes: SC4-WS-03 (Use digital technologies), SC4-WS-06 (Process data and information)

- Warm-Up (10 mins): "Digital Tools We Use" - Quick brainstorm: "What digital tools do you already use for learning, creating, or exploring information?" (e.g., Google Docs, online simulations, coding platforms, online encyclopedias). Discuss: What makes these tools useful? What are the benefits of digital tools in learning? Lead to the idea of Observable as a powerful digital tool for data science.

- Main Activity 1: "Observable Interface Expedition" (25 mins):  
  
Guided, step-by-step walkthrough of the Observable interface:  
  
Account login and homepage overview.  
Navigating to "Notebooks" and creating a new notebook ("+ New").  
Introduction to the notebook interface: Title, cells (markdown and code), toolbar.  
Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).  
Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.  
Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).  
  
Encourage students to actively follow along and experiment within their own Observable notebooks as you demonstrate.

- Guided, step-by-step walkthrough of the Observable interface:  
  
Account login and homepage overview.  
Navigating to "Notebooks" and creating a new notebook ("+ New").  
Introduction to the notebook interface: Title, cells (markdown and code), toolbar.  
Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).  
Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.  
Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).

- Account login and homepage overview.

- Navigating to "Notebooks" and creating a new notebook ("+ New").

- Introduction to the notebook interface: Title, cells (markdown and code), toolbar.

- Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).

- Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.

- Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).

- Encourage students to actively follow along and experiment within their own Observable notebooks as you demonstrate.

- Main Activity 2: "Meet Your AI Tutor - Ask Me Anything (Data Science Edition)" (20 mins):  
  
Introduce the integrated AI Tutor (specific implementation will depend on the chosen AI tool).  
Demonstrate how to access and interact with the AI Tutor (voice or chat interface).  
Guided practice using the AI Tutor:  
  
Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".  
Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".  
Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".  
  
Encourage students to ask their own questions and explore the AI Tutor's capabilities. Emphasize it's a tool for help and learning, not just getting answers.

- Introduce the integrated AI Tutor (specific implementation will depend on the chosen AI tool).

- Demonstrate how to access and interact with the AI Tutor (voice or chat interface).

- Guided practice using the AI Tutor:  
  
Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".  
Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".  
Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".

- Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".

- Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".

- Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".

- Encourage students to ask their own questions and explore the AI Tutor's capabilities. Emphasize it's a tool for help and learning, not just getting answers.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are your first impressions of using Observable? What do you find interesting or useful?  
How do you think the AI Tutor can support your learning in data science? What are its potential benefits?  
What are you most looking forward to doing with Observable and data science in the coming weeks?

- What are your first impressions of using Observable? What do you find interesting or useful?

- How do you think the AI Tutor can support your learning in data science? What are its potential benefits?

- What are you most looking forward to doing with Observable and data science in the coming weeks?

- Assessments:  
  
Formative Check (5 mins): Quickly review student Observable notebooks to ensure they have created a notebook, added markdown and code cells, and run basic code.  
Observe student interaction with the AI Tutor.

- Formative Check (5 mins): Quickly review student Observable notebooks to ensure they have created a notebook, added markdown and code cells, and run basic code.

- Observe student interaction with the AI Tutor.

- Student Workbook:  
  
Step-by-step guide to creating an Observable account (if needed) and creating a basic notebook.  
Key features of the Observable interface (screenshot and labels).  
Basic markdown formatting guide.  
Instructions for accessing and using the AI Tutor with example questions.  
Practice exercises for creating markdown and code cells and running Python code.

- Step-by-step guide to creating an Observable account (if needed) and creating a basic notebook.

- Key features of the Observable interface (screenshot and labels).

- Basic markdown formatting guide.

- Instructions for accessing and using the AI Tutor with example questions.

- Practice exercises for creating markdown and code cells and running Python code.

- Extension Activities:  
  
Explore Observable's example notebooks in the "Explore" section to see more advanced uses of the platform.  
Challenge: Experiment with more advanced markdown formatting (links, images, tables). Try slightly more complex Python code in code cells (basic arithmetic, variable assignment).

- Explore Observable's example notebooks in the "Explore" section to see more advanced uses of the platform.

- Challenge: Experiment with more advanced markdown formatting (links, images, tables). Try slightly more complex Python code in code cells (basic arithmetic, variable assignment).

- Guided, step-by-step walkthrough of the Observable interface:  
  
Account login and homepage overview.  
Navigating to "Notebooks" and creating a new notebook ("+ New").  
Introduction to the notebook interface: Title, cells (markdown and code), toolbar.  
Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).  
Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.  
Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).

- Account login and homepage overview.

- Navigating to "Notebooks" and creating a new notebook ("+ New").

- Introduction to the notebook interface: Title, cells (markdown and code), toolbar.

- Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).

- Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.

- Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).

- Encourage students to actively follow along and experiment within their own Observable notebooks as you demonstrate.

- Account login and homepage overview.

- Navigating to "Notebooks" and creating a new notebook ("+ New").

- Introduction to the notebook interface: Title, cells (markdown and code), toolbar.

- Creating a Markdown Cell: Click "+", choose "Markdown". Type a title (e.g., "My First Observable Notebook"). Demonstrate markdown formatting basics (headings, bold, italics). Run the cell (Shift+Enter).

- Creating a Code Cell: Click "+", choose "Code". Type simple Python code: print("Hello Data Science!"). Run the cell. Explain output display.

- Adding more markdown and code cells. Rearranging cells (drag and drop). Saving notebooks (automatic, but emphasize). Sharing notebooks (briefly introduce "Share" button).

- Introduce the integrated AI Tutor (specific implementation will depend on the chosen AI tool).

- Demonstrate how to access and interact with the AI Tutor (voice or chat interface).

- Guided practice using the AI Tutor:  
  
Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".  
Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".  
Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".

- Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".

- Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".

- Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".

- Encourage students to ask their own questions and explore the AI Tutor's capabilities. Emphasize it's a tool for help and learning, not just getting answers.

- Ask basic questions about Observable: "How do I create a code cell?", "What is markdown used for?", "How do I save my notebook?".

- Ask basic data science concept questions: "What is data?", "What is an algorithm?", "What is statistics?".

- Ask for help with simple Python code: "How do I print something in Python?", "How do I create a list in Python?".

- What are your first impressions of using Observable? What do you find interesting or useful?

- How do you think the AI Tutor can support your learning in data science? What are its potential benefits?

- What are you most looking forward to doing with Observable and data science in the coming weeks?

- Formative Check (5 mins): Quickly review student Observable notebooks to ensure they have created a notebook, added markdown and code cells, and run basic code.

- Observe student interaction with the AI Tutor.

- Step-by-step guide to creating an Observable account (if needed) and creating a basic notebook.

- Key features of the Observable interface (screenshot and labels).

- Basic markdown formatting guide.

- Instructions for accessing and using the AI Tutor with example questions.

- Practice exercises for creating markdown and code cells and running Python code.

- Explore Observable's example notebooks in the "Explore" section to see more advanced uses of the platform.

- Challenge: Experiment with more advanced markdown formatting (links, images, tables). Try slightly more complex Python code in code cells (basic arithmetic, variable assignment).

- Slide Deck: Introduction to Observable, interface tour, step-by-step guide for notebook creation, examples of markdown and code cells, introduction to AI Tutor.

- Handout: Observable guide, AI Tutor usage guide, practice exercises for Observable.

- Computers/Tablets with internet access for Observable.

- (Optional) Pre-lesson instructions for Observable account setup if needed.

### Lesson 1.3: Data Exploration: Types of Data in Science

- Title: Data Under the Microscope: Exploring the Variety of Data in Science

- Learning Outcomes:  
  
Students will be able to define "data" in a scientific context.  
Students will be able to differentiate between qualitative and quantitative data, and provide scientific examples of each.  
Students will be able to further classify quantitative data as discrete or continuous, and provide scientific examples.  
Students will be able to identify examples of different data types in real-world scientific datasets.  
NSW Curriculum Outcomes: SC4-DA1-01 (examines a range of sources of data), SC4-WS-06 (analyses data and information)

- Students will be able to define "data" in a scientific context.

- Students will be able to differentiate between qualitative and quantitative data, and provide scientific examples of each.

- Students will be able to further classify quantitative data as discrete or continuous, and provide scientific examples.

- Students will be able to identify examples of different data types in real-world scientific datasets.

- NSW Curriculum Outcomes: SC4-DA1-01 (examines a range of sources of data), SC4-WS-06 (analyses data and information)

- Overview: This lesson delves into the fundamental concept of data, focusing on the different types of data encountered in scientific investigations. Students will learn to classify data as qualitative or quantitative, and further refine quantitative data into discrete and continuous types, using scientific examples throughout.

- Activities:  
  
Warm-Up (10 mins): "Data Around the Room" - Quick observation activity. Ask students to look around the classroom and identify things that could be considered "data" related to the classroom environment. Examples: colours of objects, number of desks, temperature reading, types of materials, student opinions (if surveyed). Briefly list these on the board.  
Main Activity 1: "Qualitative vs. Quantitative Data - The Great Divide" (25 mins):  
  
Introduce the fundamental distinction: Qualitative Data (descriptive, categorical, non-numerical) vs. Quantitative Data (numerical, measurable).  
Use examples from the "Data Around the Room" brainstorm to classify each item as Qualitative or Quantitative.  
Provide more scientific examples of each data type across different scientific disciplines:  
  
Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.  
Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).  
  
Interactive sorting activity: Provide cards or a digital drag-and-drop activity with various scientific data examples. Students classify them as Qualitative or Quantitative. Discuss and clarify classifications as a class.  
  
Main Activity 2: "Discrete vs. Continuous Data - Refining the Numbers" (20 mins):  
  
Focus on Quantitative Data. Introduce the sub-classification: Discrete Data (countable, whole numbers) vs. Continuous Data (measurable, can take any value within a range, often includes decimals/fractions).  
Use previous Quantitative Data examples to classify them as Discrete or Continuous.  
Provide more examples and scenarios to solidify understanding:  
  
Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.  
Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.  
  
"Data Type Challenge": Present scientific scenarios or research questions. Students identify the type of quantitative data that would be collected (Discrete or Continuous) and justify their choice.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to distinguish between different types of data in science?  
How does the type of data you collect influence how you can analyse it and what kinds of conclusions you can draw?  
Can some data be considered both qualitative and quantitative in certain situations? (e.g., colour intensity can be described qualitatively or measured quantitatively with instruments).  
  
Assessments:  
  
Formative Quiz (5 mins): Short multiple-choice quiz on defining data, differentiating between qualitative and quantitative data, and classifying data as discrete or continuous (scientific examples). AI-generated quiz.  
Observation of student participation in sorting activities and class discussions.  
  
Student Workbook:  
  
Definitions of data, qualitative data, quantitative data, discrete data, continuous data (with examples).  
Sorting table for classifying scientific data examples (Qualitative/Quantitative, Discrete/Continuous).  
"Data Type Challenge" exercises with scientific scenarios.  
  
Extension Activities:  
  
Students find real-world scientific datasets online and identify examples of qualitative, quantitative, discrete, and continuous data within those datasets.  
Challenge: Research measurement scales (nominal, ordinal, interval, ratio) and try to classify the data types (qualitative/quantitative, discrete/continuous) according to these scales (optional – more advanced).

- Warm-Up (10 mins): "Data Around the Room" - Quick observation activity. Ask students to look around the classroom and identify things that could be considered "data" related to the classroom environment. Examples: colours of objects, number of desks, temperature reading, types of materials, student opinions (if surveyed). Briefly list these on the board.

- Main Activity 1: "Qualitative vs. Quantitative Data - The Great Divide" (25 mins):  
  
Introduce the fundamental distinction: Qualitative Data (descriptive, categorical, non-numerical) vs. Quantitative Data (numerical, measurable).  
Use examples from the "Data Around the Room" brainstorm to classify each item as Qualitative or Quantitative.  
Provide more scientific examples of each data type across different scientific disciplines:  
  
Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.  
Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).  
  
Interactive sorting activity: Provide cards or a digital drag-and-drop activity with various scientific data examples. Students classify them as Qualitative or Quantitative. Discuss and clarify classifications as a class.

- Introduce the fundamental distinction: Qualitative Data (descriptive, categorical, non-numerical) vs. Quantitative Data (numerical, measurable).

- Use examples from the "Data Around the Room" brainstorm to classify each item as Qualitative or Quantitative.

- Provide more scientific examples of each data type across different scientific disciplines:  
  
Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.  
Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).

- Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.

- Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).

- Interactive sorting activity: Provide cards or a digital drag-and-drop activity with various scientific data examples. Students classify them as Qualitative or Quantitative. Discuss and clarify classifications as a class.

- Main Activity 2: "Discrete vs. Continuous Data - Refining the Numbers" (20 mins):  
  
Focus on Quantitative Data. Introduce the sub-classification: Discrete Data (countable, whole numbers) vs. Continuous Data (measurable, can take any value within a range, often includes decimals/fractions).  
Use previous Quantitative Data examples to classify them as Discrete or Continuous.  
Provide more examples and scenarios to solidify understanding:  
  
Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.  
Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.  
  
"Data Type Challenge": Present scientific scenarios or research questions. Students identify the type of quantitative data that would be collected (Discrete or Continuous) and justify their choice.

- Focus on Quantitative Data. Introduce the sub-classification: Discrete Data (countable, whole numbers) vs. Continuous Data (measurable, can take any value within a range, often includes decimals/fractions).

- Use previous Quantitative Data examples to classify them as Discrete or Continuous.

- Provide more examples and scenarios to solidify understanding:  
  
Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.  
Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.

- Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.

- Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.

- "Data Type Challenge": Present scientific scenarios or research questions. Students identify the type of quantitative data that would be collected (Discrete or Continuous) and justify their choice.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to distinguish between different types of data in science?  
How does the type of data you collect influence how you can analyse it and what kinds of conclusions you can draw?  
Can some data be considered both qualitative and quantitative in certain situations? (e.g., colour intensity can be described qualitatively or measured quantitatively with instruments).

- Why is it important to distinguish between different types of data in science?

- How does the type of data you collect influence how you can analyse it and what kinds of conclusions you can draw?

- Can some data be considered both qualitative and quantitative in certain situations? (e.g., colour intensity can be described qualitatively or measured quantitatively with instruments).

- Assessments:  
  
Formative Quiz (5 mins): Short multiple-choice quiz on defining data, differentiating between qualitative and quantitative data, and classifying data as discrete or continuous (scientific examples). AI-generated quiz.  
Observation of student participation in sorting activities and class discussions.

- Formative Quiz (5 mins): Short multiple-choice quiz on defining data, differentiating between qualitative and quantitative data, and classifying data as discrete or continuous (scientific examples). AI-generated quiz.

- Observation of student participation in sorting activities and class discussions.

- Student Workbook:  
  
Definitions of data, qualitative data, quantitative data, discrete data, continuous data (with examples).  
Sorting table for classifying scientific data examples (Qualitative/Quantitative, Discrete/Continuous).  
"Data Type Challenge" exercises with scientific scenarios.

- Definitions of data, qualitative data, quantitative data, discrete data, continuous data (with examples).

- Sorting table for classifying scientific data examples (Qualitative/Quantitative, Discrete/Continuous).

- "Data Type Challenge" exercises with scientific scenarios.

- Extension Activities:  
  
Students find real-world scientific datasets online and identify examples of qualitative, quantitative, discrete, and continuous data within those datasets.  
Challenge: Research measurement scales (nominal, ordinal, interval, ratio) and try to classify the data types (qualitative/quantitative, discrete/continuous) according to these scales (optional – more advanced).

- Students find real-world scientific datasets online and identify examples of qualitative, quantitative, discrete, and continuous data within those datasets.

- Challenge: Research measurement scales (nominal, ordinal, interval, ratio) and try to classify the data types (qualitative/quantitative, discrete/continuous) according to these scales (optional – more advanced).

- Lesson Materials:  
  
Slide Deck: Data definition, qualitative vs. quantitative data (definitions, examples), discrete vs. continuous data (definitions, examples), real-world data examples.  
Handout: Data sorting cards or digital drag-and-drop activity, "Data Type Challenge" scenarios worksheet, definitions of data types.  
Formative Quiz (printable or digital).

- Slide Deck: Data definition, qualitative vs. quantitative data (definitions, examples), discrete vs. continuous data (definitions, examples), real-world data examples.

- Handout: Data sorting cards or digital drag-and-drop activity, "Data Type Challenge" scenarios worksheet, definitions of data types.

- Formative Quiz (printable or digital).

- Students will be able to define "data" in a scientific context.

- Students will be able to differentiate between qualitative and quantitative data, and provide scientific examples of each.

- Students will be able to further classify quantitative data as discrete or continuous, and provide scientific examples.

- Students will be able to identify examples of different data types in real-world scientific datasets.

- NSW Curriculum Outcomes: SC4-DA1-01 (examines a range of sources of data), SC4-WS-06 (analyses data and information)

- Warm-Up (10 mins): "Data Around the Room" - Quick observation activity. Ask students to look around the classroom and identify things that could be considered "data" related to the classroom environment. Examples: colours of objects, number of desks, temperature reading, types of materials, student opinions (if surveyed). Briefly list these on the board.

- Main Activity 1: "Qualitative vs. Quantitative Data - The Great Divide" (25 mins):  
  
Introduce the fundamental distinction: Qualitative Data (descriptive, categorical, non-numerical) vs. Quantitative Data (numerical, measurable).  
Use examples from the "Data Around the Room" brainstorm to classify each item as Qualitative or Quantitative.  
Provide more scientific examples of each data type across different scientific disciplines:  
  
Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.  
Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).  
  
Interactive sorting activity: Provide cards or a digital drag-and-drop activity with various scientific data examples. Students classify them as Qualitative or Quantitative. Discuss and clarify classifications as a class.

- Introduce the fundamental distinction: Qualitative Data (descriptive, categorical, non-numerical) vs. Quantitative Data (numerical, measurable).

- Use examples from the "Data Around the Room" brainstorm to classify each item as Qualitative or Quantitative.

- Provide more scientific examples of each data type across different scientific disciplines:  
  
Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.  
Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).

- Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.

- Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).

- Interactive sorting activity: Provide cards or a digital drag-and-drop activity with various scientific data examples. Students classify them as Qualitative or Quantitative. Discuss and clarify classifications as a class.

- Main Activity 2: "Discrete vs. Continuous Data - Refining the Numbers" (20 mins):  
  
Focus on Quantitative Data. Introduce the sub-classification: Discrete Data (countable, whole numbers) vs. Continuous Data (measurable, can take any value within a range, often includes decimals/fractions).  
Use previous Quantitative Data examples to classify them as Discrete or Continuous.  
Provide more examples and scenarios to solidify understanding:  
  
Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.  
Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.  
  
"Data Type Challenge": Present scientific scenarios or research questions. Students identify the type of quantitative data that would be collected (Discrete or Continuous) and justify their choice.

- Focus on Quantitative Data. Introduce the sub-classification: Discrete Data (countable, whole numbers) vs. Continuous Data (measurable, can take any value within a range, often includes decimals/fractions).

- Use previous Quantitative Data examples to classify them as Discrete or Continuous.

- Provide more examples and scenarios to solidify understanding:  
  
Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.  
Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.

- Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.

- Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.

- "Data Type Challenge": Present scientific scenarios or research questions. Students identify the type of quantitative data that would be collected (Discrete or Continuous) and justify their choice.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to distinguish between different types of data in science?  
How does the type of data you collect influence how you can analyse it and what kinds of conclusions you can draw?  
Can some data be considered both qualitative and quantitative in certain situations? (e.g., colour intensity can be described qualitatively or measured quantitatively with instruments).

- Why is it important to distinguish between different types of data in science?

- How does the type of data you collect influence how you can analyse it and what kinds of conclusions you can draw?

- Can some data be considered both qualitative and quantitative in certain situations? (e.g., colour intensity can be described qualitatively or measured quantitatively with instruments).

- Assessments:  
  
Formative Quiz (5 mins): Short multiple-choice quiz on defining data, differentiating between qualitative and quantitative data, and classifying data as discrete or continuous (scientific examples). AI-generated quiz.  
Observation of student participation in sorting activities and class discussions.

- Formative Quiz (5 mins): Short multiple-choice quiz on defining data, differentiating between qualitative and quantitative data, and classifying data as discrete or continuous (scientific examples). AI-generated quiz.

- Observation of student participation in sorting activities and class discussions.

- Student Workbook:  
  
Definitions of data, qualitative data, quantitative data, discrete data, continuous data (with examples).  
Sorting table for classifying scientific data examples (Qualitative/Quantitative, Discrete/Continuous).  
"Data Type Challenge" exercises with scientific scenarios.

- Definitions of data, qualitative data, quantitative data, discrete data, continuous data (with examples).

- Sorting table for classifying scientific data examples (Qualitative/Quantitative, Discrete/Continuous).

- "Data Type Challenge" exercises with scientific scenarios.

- Extension Activities:  
  
Students find real-world scientific datasets online and identify examples of qualitative, quantitative, discrete, and continuous data within those datasets.  
Challenge: Research measurement scales (nominal, ordinal, interval, ratio) and try to classify the data types (qualitative/quantitative, discrete/continuous) according to these scales (optional – more advanced).

- Students find real-world scientific datasets online and identify examples of qualitative, quantitative, discrete, and continuous data within those datasets.

- Challenge: Research measurement scales (nominal, ordinal, interval, ratio) and try to classify the data types (qualitative/quantitative, discrete/continuous) according to these scales (optional – more advanced).

- Introduce the fundamental distinction: Qualitative Data (descriptive, categorical, non-numerical) vs. Quantitative Data (numerical, measurable).

- Use examples from the "Data Around the Room" brainstorm to classify each item as Qualitative or Quantitative.

- Provide more scientific examples of each data type across different scientific disciplines:  
  
Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.  
Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).

- Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.

- Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).

- Interactive sorting activity: Provide cards or a digital drag-and-drop activity with various scientific data examples. Students classify them as Qualitative or Quantitative. Discuss and clarify classifications as a class.

- Qualitative: Colour of chemical precipitates, descriptions of animal behaviour, types of rock formations, patient symptoms descriptions, survey responses with open-ended questions.

- Quantitative: Temperature readings, height of plants, reaction rates, number of species, population size, experimental measurements (mass, volume, time).

- Focus on Quantitative Data. Introduce the sub-classification: Discrete Data (countable, whole numbers) vs. Continuous Data (measurable, can take any value within a range, often includes decimals/fractions).

- Use previous Quantitative Data examples to classify them as Discrete or Continuous.

- Provide more examples and scenarios to solidify understanding:  
  
Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.  
Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.

- Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.

- Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.

- "Data Type Challenge": Present scientific scenarios or research questions. Students identify the type of quantitative data that would be collected (Discrete or Continuous) and justify their choice.

- Discrete: Number of leaves on a plant, number of bacteria colonies, number of trials in an experiment, number of students in a class.

- Continuous: Height of a tree, temperature of a solution, time taken for a reaction, mass of a sample, pH value.

- Why is it important to distinguish between different types of data in science?

- How does the type of data you collect influence how you can analyse it and what kinds of conclusions you can draw?

- Can some data be considered both qualitative and quantitative in certain situations? (e.g., colour intensity can be described qualitatively or measured quantitatively with instruments).

- Formative Quiz (5 mins): Short multiple-choice quiz on defining data, differentiating between qualitative and quantitative data, and classifying data as discrete or continuous (scientific examples). AI-generated quiz.

- Observation of student participation in sorting activities and class discussions.

- Definitions of data, qualitative data, quantitative data, discrete data, continuous data (with examples).

- Sorting table for classifying scientific data examples (Qualitative/Quantitative, Discrete/Continuous).

- "Data Type Challenge" exercises with scientific scenarios.

- Students find real-world scientific datasets online and identify examples of qualitative, quantitative, discrete, and continuous data within those datasets.

- Challenge: Research measurement scales (nominal, ordinal, interval, ratio) and try to classify the data types (qualitative/quantitative, discrete/continuous) according to these scales (optional – more advanced).

- Slide Deck: Data definition, qualitative vs. quantitative data (definitions, examples), discrete vs. continuous data (definitions, examples), real-world data examples.

- Handout: Data sorting cards or digital drag-and-drop activity, "Data Type Challenge" scenarios worksheet, definitions of data types.

- Formative Quiz (printable or digital).

### Week 2: Data Collection Mastery & Digital Responsibility

### Lesson 2.1: Digital Footprint & Data Ethics

- Title: Navigating the Digital World: Your Footprint and Ethical Data Use

- Learning Outcomes:  
  
Students will be able to define and explain the concept of a digital footprint.  
Students will be able to identify various online activities that contribute to their digital footprint.  
Students will be able to analyse the potential implications of their digital footprint for privacy and security.  
Students will understand ethical considerations related to data collection, use, and privacy in the digital age, particularly in scientific contexts.  
NSW Curriculum Outcomes: (Implicitly linked to broader digital literacy, ethical understanding, and responsible data use within SC4-DA1-01 and SC4-WS-07 – focus on responsible digital citizenship).

- Students will be able to define and explain the concept of a digital footprint.

- Students will be able to identify various online activities that contribute to their digital footprint.

- Students will be able to analyse the potential implications of their digital footprint for privacy and security.

- Students will understand ethical considerations related to data collection, use, and privacy in the digital age, particularly in scientific contexts.

- NSW Curriculum Outcomes: (Implicitly linked to broader digital literacy, ethical understanding, and responsible data use within SC4-DA1-01 and SC4-WS-07 – focus on responsible digital citizenship).

- Overview: This consolidated lesson focuses on digital footprints and data ethics. Students will explore what constitutes a digital footprint, its implications, and the ethical responsibilities associated with data in the digital world, including scientific data. This lesson is designed to be more engaging and thought-provoking for selective school students, encouraging critical reflection on their online presence and ethical data practices.

- Activities:  
  
Warm-Up (10 mins): "Digital Footprint Brainstorm & Scenario" -  
  
Quick brainstorm: "What are some things you do online that leave a trace or record behind?" (e.g., social media posts, searches, online shopping, website visits, app usage). Write these on the board.  
Present a short, thought-provoking scenario: "Imagine a future employer or university admissions officer is looking at your online activity. What kind of picture would your digital footprint paint of you? Are there things you would want to change or be more mindful of?" Brief class discussion.  
  
Main Activity 1: "Deconstructing the Digital Footprint" (25 mins):  
  
Introduce the definition of a digital footprint: passive and active footprints.  
Categorize brainstormed examples from warm-up into passive and active footprints.  
Discuss different types of data collected online: personal information, browsing history, location data, social media activity, etc.  
Explore the potential implications of digital footprints:  
  
Privacy concerns and data security risks.  
Online reputation and long-term consequences.  
Targeted advertising and filter bubbles.  
Use of digital footprint data by companies, governments, and other entities.  
  
  
Main Activity 2: "Ethical Data Dilemmas in Science" (20 mins):  
  
Shift focus to data ethics, particularly in scientific contexts.  
Present ethical dilemma scenarios related to data science (designed to be more complex and nuanced for selective students):  
  
Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?  
Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?  
Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?  
  
Small group discussions on each scenario, followed by class sharing and debate. Encourage students to consider different perspectives and justify their ethical reasoning.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are the key takeaways about managing your digital footprint responsibly?  
Why are ethical considerations particularly important in data science and scientific research?  
How can we ensure data is used ethically and responsibly in science and technology?  
  
Assessments:  
  
Formative Task (5 mins): "Digital Footprint Reflection" - Students write a short paragraph reflecting on their own digital footprint and one action they will take to manage it more responsibly.  
Observation of student participation in discussions and group work, focusing on the depth of their ethical reasoning.  
  
Student Workbook:  
  
Definitions of digital footprint (active/passive).  
Table to categorize online activities and their footprint types.  
Summary of ethical considerations in data science.  
Space for reflection on personal digital footprint and action plan.  
  
Extension Activities:  
  
Research current events or news articles related to data privacy breaches, ethical AI dilemmas, or digital surveillance. Present findings to the class and lead a discussion.  
Challenge: Design a "Digital Footprint Awareness Campaign" for the school, aimed at educating students about responsible online behaviour and data ethics.

- Warm-Up (10 mins): "Digital Footprint Brainstorm & Scenario" -  
  
Quick brainstorm: "What are some things you do online that leave a trace or record behind?" (e.g., social media posts, searches, online shopping, website visits, app usage). Write these on the board.  
Present a short, thought-provoking scenario: "Imagine a future employer or university admissions officer is looking at your online activity. What kind of picture would your digital footprint paint of you? Are there things you would want to change or be more mindful of?" Brief class discussion.

- Quick brainstorm: "What are some things you do online that leave a trace or record behind?" (e.g., social media posts, searches, online shopping, website visits, app usage). Write these on the board.

- Present a short, thought-provoking scenario: "Imagine a future employer or university admissions officer is looking at your online activity. What kind of picture would your digital footprint paint of you? Are there things you would want to change or be more mindful of?" Brief class discussion.

- Main Activity 1: "Deconstructing the Digital Footprint" (25 mins):  
  
Introduce the definition of a digital footprint: passive and active footprints.  
Categorize brainstormed examples from warm-up into passive and active footprints.  
Discuss different types of data collected online: personal information, browsing history, location data, social media activity, etc.  
Explore the potential implications of digital footprints:  
  
Privacy concerns and data security risks.  
Online reputation and long-term consequences.  
Targeted advertising and filter bubbles.  
Use of digital footprint data by companies, governments, and other entities.

- Introduce the definition of a digital footprint: passive and active footprints.

- Categorize brainstormed examples from warm-up into passive and active footprints.

- Discuss different types of data collected online: personal information, browsing history, location data, social media activity, etc.

- Explore the potential implications of digital footprints:  
  
Privacy concerns and data security risks.  
Online reputation and long-term consequences.  
Targeted advertising and filter bubbles.  
Use of digital footprint data by companies, governments, and other entities.

- Privacy concerns and data security risks.

- Online reputation and long-term consequences.

- Targeted advertising and filter bubbles.

- Use of digital footprint data by companies, governments, and other entities.

- Main Activity 2: "Ethical Data Dilemmas in Science" (20 mins):  
  
Shift focus to data ethics, particularly in scientific contexts.  
Present ethical dilemma scenarios related to data science (designed to be more complex and nuanced for selective students):  
  
Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?  
Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?  
Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?  
  
Small group discussions on each scenario, followed by class sharing and debate. Encourage students to consider different perspectives and justify their ethical reasoning.

- Shift focus to data ethics, particularly in scientific contexts.

- Present ethical dilemma scenarios related to data science (designed to be more complex and nuanced for selective students):  
  
Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?  
Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?  
Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?

- Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?

- Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?

- Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?

- Small group discussions on each scenario, followed by class sharing and debate. Encourage students to consider different perspectives and justify their ethical reasoning.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the key takeaways about managing your digital footprint responsibly?  
Why are ethical considerations particularly important in data science and scientific research?  
How can we ensure data is used ethically and responsibly in science and technology?

- What are the key takeaways about managing your digital footprint responsibly?

- Why are ethical considerations particularly important in data science and scientific research?

- How can we ensure data is used ethically and responsibly in science and technology?

- Assessments:  
  
Formative Task (5 mins): "Digital Footprint Reflection" - Students write a short paragraph reflecting on their own digital footprint and one action they will take to manage it more responsibly.  
Observation of student participation in discussions and group work, focusing on the depth of their ethical reasoning.

- Formative Task (5 mins): "Digital Footprint Reflection" - Students write a short paragraph reflecting on their own digital footprint and one action they will take to manage it more responsibly.

- Observation of student participation in discussions and group work, focusing on the depth of their ethical reasoning.

- Student Workbook:  
  
Definitions of digital footprint (active/passive).  
Table to categorize online activities and their footprint types.  
Summary of ethical considerations in data science.  
Space for reflection on personal digital footprint and action plan.

- Definitions of digital footprint (active/passive).

- Table to categorize online activities and their footprint types.

- Summary of ethical considerations in data science.

- Space for reflection on personal digital footprint and action plan.

- Extension Activities:  
  
Research current events or news articles related to data privacy breaches, ethical AI dilemmas, or digital surveillance. Present findings to the class and lead a discussion.  
Challenge: Design a "Digital Footprint Awareness Campaign" for the school, aimed at educating students about responsible online behaviour and data ethics.

- Research current events or news articles related to data privacy breaches, ethical AI dilemmas, or digital surveillance. Present findings to the class and lead a discussion.

- Challenge: Design a "Digital Footprint Awareness Campaign" for the school, aimed at educating students about responsible online behaviour and data ethics.

- Lesson Materials:  
  
Slide Deck: Digital footprint definition, types, implications, ethical data dilemmas scenarios, key ethical principles.  
Handout: Digital Footprint Brainstorm worksheet, Ethical Dilemma scenarios descriptions, Digital Footprint Reflection prompt.  
(Optional) Short video clip or news report related to digital privacy or data ethics to spark discussion.

- Slide Deck: Digital footprint definition, types, implications, ethical data dilemmas scenarios, key ethical principles.

- Handout: Digital Footprint Brainstorm worksheet, Ethical Dilemma scenarios descriptions, Digital Footprint Reflection prompt.

- (Optional) Short video clip or news report related to digital privacy or data ethics to spark discussion.

- Students will be able to define and explain the concept of a digital footprint.

- Students will be able to identify various online activities that contribute to their digital footprint.

- Students will be able to analyse the potential implications of their digital footprint for privacy and security.

- Students will understand ethical considerations related to data collection, use, and privacy in the digital age, particularly in scientific contexts.

- NSW Curriculum Outcomes: (Implicitly linked to broader digital literacy, ethical understanding, and responsible data use within SC4-DA1-01 and SC4-WS-07 – focus on responsible digital citizenship).

- Warm-Up (10 mins): "Digital Footprint Brainstorm & Scenario" -  
  
Quick brainstorm: "What are some things you do online that leave a trace or record behind?" (e.g., social media posts, searches, online shopping, website visits, app usage). Write these on the board.  
Present a short, thought-provoking scenario: "Imagine a future employer or university admissions officer is looking at your online activity. What kind of picture would your digital footprint paint of you? Are there things you would want to change or be more mindful of?" Brief class discussion.

- Quick brainstorm: "What are some things you do online that leave a trace or record behind?" (e.g., social media posts, searches, online shopping, website visits, app usage). Write these on the board.

- Present a short, thought-provoking scenario: "Imagine a future employer or university admissions officer is looking at your online activity. What kind of picture would your digital footprint paint of you? Are there things you would want to change or be more mindful of?" Brief class discussion.

- Main Activity 1: "Deconstructing the Digital Footprint" (25 mins):  
  
Introduce the definition of a digital footprint: passive and active footprints.  
Categorize brainstormed examples from warm-up into passive and active footprints.  
Discuss different types of data collected online: personal information, browsing history, location data, social media activity, etc.  
Explore the potential implications of digital footprints:  
  
Privacy concerns and data security risks.  
Online reputation and long-term consequences.  
Targeted advertising and filter bubbles.  
Use of digital footprint data by companies, governments, and other entities.

- Introduce the definition of a digital footprint: passive and active footprints.

- Categorize brainstormed examples from warm-up into passive and active footprints.

- Discuss different types of data collected online: personal information, browsing history, location data, social media activity, etc.

- Explore the potential implications of digital footprints:  
  
Privacy concerns and data security risks.  
Online reputation and long-term consequences.  
Targeted advertising and filter bubbles.  
Use of digital footprint data by companies, governments, and other entities.

- Privacy concerns and data security risks.

- Online reputation and long-term consequences.

- Targeted advertising and filter bubbles.

- Use of digital footprint data by companies, governments, and other entities.

- Main Activity 2: "Ethical Data Dilemmas in Science" (20 mins):  
  
Shift focus to data ethics, particularly in scientific contexts.  
Present ethical dilemma scenarios related to data science (designed to be more complex and nuanced for selective students):  
  
Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?  
Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?  
Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?  
  
Small group discussions on each scenario, followed by class sharing and debate. Encourage students to consider different perspectives and justify their ethical reasoning.

- Shift focus to data ethics, particularly in scientific contexts.

- Present ethical dilemma scenarios related to data science (designed to be more complex and nuanced for selective students):  
  
Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?  
Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?  
Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?

- Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?

- Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?

- Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?

- Small group discussions on each scenario, followed by class sharing and debate. Encourage students to consider different perspectives and justify their ethical reasoning.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the key takeaways about managing your digital footprint responsibly?  
Why are ethical considerations particularly important in data science and scientific research?  
How can we ensure data is used ethically and responsibly in science and technology?

- What are the key takeaways about managing your digital footprint responsibly?

- Why are ethical considerations particularly important in data science and scientific research?

- How can we ensure data is used ethically and responsibly in science and technology?

- Assessments:  
  
Formative Task (5 mins): "Digital Footprint Reflection" - Students write a short paragraph reflecting on their own digital footprint and one action they will take to manage it more responsibly.  
Observation of student participation in discussions and group work, focusing on the depth of their ethical reasoning.

- Formative Task (5 mins): "Digital Footprint Reflection" - Students write a short paragraph reflecting on their own digital footprint and one action they will take to manage it more responsibly.

- Observation of student participation in discussions and group work, focusing on the depth of their ethical reasoning.

- Student Workbook:  
  
Definitions of digital footprint (active/passive).  
Table to categorize online activities and their footprint types.  
Summary of ethical considerations in data science.  
Space for reflection on personal digital footprint and action plan.

- Definitions of digital footprint (active/passive).

- Table to categorize online activities and their footprint types.

- Summary of ethical considerations in data science.

- Space for reflection on personal digital footprint and action plan.

- Extension Activities:  
  
Research current events or news articles related to data privacy breaches, ethical AI dilemmas, or digital surveillance. Present findings to the class and lead a discussion.  
Challenge: Design a "Digital Footprint Awareness Campaign" for the school, aimed at educating students about responsible online behaviour and data ethics.

- Research current events or news articles related to data privacy breaches, ethical AI dilemmas, or digital surveillance. Present findings to the class and lead a discussion.

- Challenge: Design a "Digital Footprint Awareness Campaign" for the school, aimed at educating students about responsible online behaviour and data ethics.

- Quick brainstorm: "What are some things you do online that leave a trace or record behind?" (e.g., social media posts, searches, online shopping, website visits, app usage). Write these on the board.

- Present a short, thought-provoking scenario: "Imagine a future employer or university admissions officer is looking at your online activity. What kind of picture would your digital footprint paint of you? Are there things you would want to change or be more mindful of?" Brief class discussion.

- Introduce the definition of a digital footprint: passive and active footprints.

- Categorize brainstormed examples from warm-up into passive and active footprints.

- Discuss different types of data collected online: personal information, browsing history, location data, social media activity, etc.

- Explore the potential implications of digital footprints:  
  
Privacy concerns and data security risks.  
Online reputation and long-term consequences.  
Targeted advertising and filter bubbles.  
Use of digital footprint data by companies, governments, and other entities.

- Privacy concerns and data security risks.

- Online reputation and long-term consequences.

- Targeted advertising and filter bubbles.

- Use of digital footprint data by companies, governments, and other entities.

- Privacy concerns and data security risks.

- Online reputation and long-term consequences.

- Targeted advertising and filter bubbles.

- Use of digital footprint data by companies, governments, and other entities.

- Shift focus to data ethics, particularly in scientific contexts.

- Present ethical dilemma scenarios related to data science (designed to be more complex and nuanced for selective students):  
  
Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?  
Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?  
Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?

- Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?

- Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?

- Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?

- Small group discussions on each scenario, followed by class sharing and debate. Encourage students to consider different perspectives and justify their ethical reasoning.

- Scenario 1: Anonymized Medical Data: Researchers have a large dataset of anonymized patient medical records for studying disease patterns. Is it ethical to use this data without explicit consent, even if anonymized? What are the potential benefits and risks?

- Scenario 2: Environmental Monitoring & Surveillance: A city wants to use AI-powered surveillance cameras to monitor pollution levels in real-time. This data could also be used for general public surveillance. What are the ethical trade-offs between environmental protection and privacy?

- Scenario 3: AI Bias in Data Analysis: An AI algorithm used to predict student success in science is found to be biased against certain demographic groups due to biases in the training data. How should this be addressed? What are the ethical responsibilities of data scientists in mitigating bias?

- What are the key takeaways about managing your digital footprint responsibly?

- Why are ethical considerations particularly important in data science and scientific research?

- How can we ensure data is used ethically and responsibly in science and technology?

- Formative Task (5 mins): "Digital Footprint Reflection" - Students write a short paragraph reflecting on their own digital footprint and one action they will take to manage it more responsibly.

- Observation of student participation in discussions and group work, focusing on the depth of their ethical reasoning.

- Definitions of digital footprint (active/passive).

- Table to categorize online activities and their footprint types.

- Summary of ethical considerations in data science.

- Space for reflection on personal digital footprint and action plan.

- Research current events or news articles related to data privacy breaches, ethical AI dilemmas, or digital surveillance. Present findings to the class and lead a discussion.

- Challenge: Design a "Digital Footprint Awareness Campaign" for the school, aimed at educating students about responsible online behaviour and data ethics.

- Slide Deck: Digital footprint definition, types, implications, ethical data dilemmas scenarios, key ethical principles.

- Handout: Digital Footprint Brainstorm worksheet, Ethical Dilemma scenarios descriptions, Digital Footprint Reflection prompt.

- (Optional) Short video clip or news report related to digital privacy or data ethics to spark discussion.

### Lesson 2.2: Data Collection Techniques: Primary vs. Secondary Data (Hands-On Design)

- Title: Becoming Data Collectors: Designing Scientific Investigations

- Learning Outcomes:  
  
Students will be able to differentiate between primary and secondary data sources in scientific research.  
Students will be able to describe various methods of primary data collection (experiments, surveys, observations, simulations).  
Students will be able to design basic data collection plans for different scientific questions, specifying data types and collection methods.  
NSW Curriculum Outcomes: SC4-WS-02 (Planning investigations), ACSIS125 (Conducting investigations safely), SC4-DA1-01 (Examining data sources).

- Students will be able to differentiate between primary and secondary data sources in scientific research.

- Students will be able to describe various methods of primary data collection (experiments, surveys, observations, simulations).

- Students will be able to design basic data collection plans for different scientific questions, specifying data types and collection methods.

- NSW Curriculum Outcomes: SC4-WS-02 (Planning investigations), ACSIS125 (Conducting investigations safely), SC4-DA1-01 (Examining data sources).

- Overview: This lesson shifts to practical data collection. Students will learn about primary and secondary data and focus on designing their own primary data collection methods. The lesson emphasizes active learning and experimental design skills, tailored for a selective student group.

- Activities:  
  
Warm-Up (10 mins): "Data Source Challenge" - Present a series of scientific questions (designed to be engaging and potentially complex for selective students):  
  
"How does caffeine affect reaction time in teenagers?"  
"Is there a correlation between air pollution levels and respiratory illness in our city?"  
"What is the biodiversity of insect species in our schoolyard?"  
"How do different types of exercise affect heart rate recovery time?"  
For each question, ask students: "Where could scientists get data to answer this question? Would it be primary or secondary data? What are some potential sources?" Class discussion to elicit different data source ideas.  
  
Main Activity 1: "Primary vs. Secondary Data Deep Dive" (20 mins):  
  
Explicitly define Primary Data (collected firsthand for a specific purpose) and Secondary Data (collected by others, for other purposes).  
Discuss the advantages and disadvantages of each data type in scientific research (e.g., primary - control, relevance, but time-consuming; secondary - readily available, large datasets, but potentially less relevant, quality concerns).  
Examples of Primary Data Collection Methods: Experiments, Surveys, Observations, Simulations. Briefly describe each method and when it is most appropriate.  
  
Main Activity 2: "Design Your Data Collection Plan" (30 mins):  
  
Divide students into small groups.  
Assign each group one or two scientific questions from the warm-up or provide new, more complex scientific questions suitable for Year 8 selective students (e.g., related to local environmental issues, human behaviour, simple physics phenomena).  
Each group task: Design a primary data collection plan to investigate their assigned question. The plan must include:  
  
Scientific Question (clearly stated).  
Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.  
Specific Data to be Collected (what variables will you measure or observe? Be precise).  
Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).  
Data Types (will you collect qualitative, quantitative, or both?).  
  
Groups work on their plans. Teacher circulates to provide guidance and feedback, encouraging more rigorous design and consideration of potential challenges.  
Groups briefly present their data collection plans to the class. Class feedback and discussion – peer review of plans.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are the key considerations when designing a good data collection plan for scientific research?  
What are some potential challenges or limitations you anticipate in carrying out your data collection plans?  
How does the choice of data collection method influence the type of data you get and the conclusions you can draw?  
  
Assessments:  
  
Formative Task (5 mins): "Data Collection Method Matching" - Quick matching exercise (worksheet or digital) where students match scientific questions to appropriate primary data collection methods.  
Review of group data collection plans – assess clarity, completeness, and feasibility.  
  
Student Workbook:  
  
Definitions of primary and secondary data with advantages/disadvantages.  
Descriptions of primary data collection methods (experiments, surveys, observations, simulations).  
Template/worksheet for designing a data collection plan (with prompts for question, method, data, procedure, data types).  
  
Extension Activities:  
  
For a chosen scientific question, design multiple data collection plans using different primary data collection methods and compare the potential data and insights each method could provide.  
Challenge: Research examples of real-world scientific studies that used specific primary data collection methods (e.g., a famous experiment, a large-scale survey). Analyze why that method was chosen and its strengths and limitations in that context.

- Warm-Up (10 mins): "Data Source Challenge" - Present a series of scientific questions (designed to be engaging and potentially complex for selective students):  
  
"How does caffeine affect reaction time in teenagers?"  
"Is there a correlation between air pollution levels and respiratory illness in our city?"  
"What is the biodiversity of insect species in our schoolyard?"  
"How do different types of exercise affect heart rate recovery time?"  
For each question, ask students: "Where could scientists get data to answer this question? Would it be primary or secondary data? What are some potential sources?" Class discussion to elicit different data source ideas.

- "How does caffeine affect reaction time in teenagers?"

- "Is there a correlation between air pollution levels and respiratory illness in our city?"

- "What is the biodiversity of insect species in our schoolyard?"

- "How do different types of exercise affect heart rate recovery time?"  
For each question, ask students: "Where could scientists get data to answer this question? Would it be primary or secondary data? What are some potential sources?" Class discussion to elicit different data source ideas.

- Main Activity 1: "Primary vs. Secondary Data Deep Dive" (20 mins):  
  
Explicitly define Primary Data (collected firsthand for a specific purpose) and Secondary Data (collected by others, for other purposes).  
Discuss the advantages and disadvantages of each data type in scientific research (e.g., primary - control, relevance, but time-consuming; secondary - readily available, large datasets, but potentially less relevant, quality concerns).  
Examples of Primary Data Collection Methods: Experiments, Surveys, Observations, Simulations. Briefly describe each method and when it is most appropriate.

- Explicitly define Primary Data (collected firsthand for a specific purpose) and Secondary Data (collected by others, for other purposes).

- Discuss the advantages and disadvantages of each data type in scientific research (e.g., primary - control, relevance, but time-consuming; secondary - readily available, large datasets, but potentially less relevant, quality concerns).

- Examples of Primary Data Collection Methods: Experiments, Surveys, Observations, Simulations. Briefly describe each method and when it is most appropriate.

- Main Activity 2: "Design Your Data Collection Plan" (30 mins):  
  
Divide students into small groups.  
Assign each group one or two scientific questions from the warm-up or provide new, more complex scientific questions suitable for Year 8 selective students (e.g., related to local environmental issues, human behaviour, simple physics phenomena).  
Each group task: Design a primary data collection plan to investigate their assigned question. The plan must include:  
  
Scientific Question (clearly stated).  
Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.  
Specific Data to be Collected (what variables will you measure or observe? Be precise).  
Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).  
Data Types (will you collect qualitative, quantitative, or both?).  
  
Groups work on their plans. Teacher circulates to provide guidance and feedback, encouraging more rigorous design and consideration of potential challenges.  
Groups briefly present their data collection plans to the class. Class feedback and discussion – peer review of plans.

- Divide students into small groups.

- Assign each group one or two scientific questions from the warm-up or provide new, more complex scientific questions suitable for Year 8 selective students (e.g., related to local environmental issues, human behaviour, simple physics phenomena).

- Each group task: Design a primary data collection plan to investigate their assigned question. The plan must include:  
  
Scientific Question (clearly stated).  
Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.  
Specific Data to be Collected (what variables will you measure or observe? Be precise).  
Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).  
Data Types (will you collect qualitative, quantitative, or both?).

- Scientific Question (clearly stated).

- Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.

- Specific Data to be Collected (what variables will you measure or observe? Be precise).

- Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).

- Data Types (will you collect qualitative, quantitative, or both?).

- Groups work on their plans. Teacher circulates to provide guidance and feedback, encouraging more rigorous design and consideration of potential challenges.

- Groups briefly present their data collection plans to the class. Class feedback and discussion – peer review of plans.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the key considerations when designing a good data collection plan for scientific research?  
What are some potential challenges or limitations you anticipate in carrying out your data collection plans?  
How does the choice of data collection method influence the type of data you get and the conclusions you can draw?

- What are the key considerations when designing a good data collection plan for scientific research?

- What are some potential challenges or limitations you anticipate in carrying out your data collection plans?

- How does the choice of data collection method influence the type of data you get and the conclusions you can draw?

- Assessments:  
  
Formative Task (5 mins): "Data Collection Method Matching" - Quick matching exercise (worksheet or digital) where students match scientific questions to appropriate primary data collection methods.  
Review of group data collection plans – assess clarity, completeness, and feasibility.

- Formative Task (5 mins): "Data Collection Method Matching" - Quick matching exercise (worksheet or digital) where students match scientific questions to appropriate primary data collection methods.

- Review of group data collection plans – assess clarity, completeness, and feasibility.

- Student Workbook:  
  
Definitions of primary and secondary data with advantages/disadvantages.  
Descriptions of primary data collection methods (experiments, surveys, observations, simulations).  
Template/worksheet for designing a data collection plan (with prompts for question, method, data, procedure, data types).

- Definitions of primary and secondary data with advantages/disadvantages.

- Descriptions of primary data collection methods (experiments, surveys, observations, simulations).

- Template/worksheet for designing a data collection plan (with prompts for question, method, data, procedure, data types).

- Extension Activities:  
  
For a chosen scientific question, design multiple data collection plans using different primary data collection methods and compare the potential data and insights each method could provide.  
Challenge: Research examples of real-world scientific studies that used specific primary data collection methods (e.g., a famous experiment, a large-scale survey). Analyze why that method was chosen and its strengths and limitations in that context.

- For a chosen scientific question, design multiple data collection plans using different primary data collection methods and compare the potential data and insights each method could provide.

- Challenge: Research examples of real-world scientific studies that used specific primary data collection methods (e.g., a famous experiment, a large-scale survey). Analyze why that method was chosen and its strengths and limitations in that context.

- Lesson Materials:  
  
Slide Deck: Primary vs. secondary data definitions, examples of primary data collection methods, data collection plan design process.  
Handout: Data Source Challenge questions, Data Collection Plan template/worksheet, Data Collection Method Matching exercise.

- Slide Deck: Primary vs. secondary data definitions, examples of primary data collection methods, data collection plan design process.

- Handout: Data Source Challenge questions, Data Collection Plan template/worksheet, Data Collection Method Matching exercise.

- Students will be able to differentiate between primary and secondary data sources in scientific research.

- Students will be able to describe various methods of primary data collection (experiments, surveys, observations, simulations).

- Students will be able to design basic data collection plans for different scientific questions, specifying data types and collection methods.

- NSW Curriculum Outcomes: SC4-WS-02 (Planning investigations), ACSIS125 (Conducting investigations safely), SC4-DA1-01 (Examining data sources).

- Warm-Up (10 mins): "Data Source Challenge" - Present a series of scientific questions (designed to be engaging and potentially complex for selective students):  
  
"How does caffeine affect reaction time in teenagers?"  
"Is there a correlation between air pollution levels and respiratory illness in our city?"  
"What is the biodiversity of insect species in our schoolyard?"  
"How do different types of exercise affect heart rate recovery time?"  
For each question, ask students: "Where could scientists get data to answer this question? Would it be primary or secondary data? What are some potential sources?" Class discussion to elicit different data source ideas.

- "How does caffeine affect reaction time in teenagers?"

- "Is there a correlation between air pollution levels and respiratory illness in our city?"

- "What is the biodiversity of insect species in our schoolyard?"

- "How do different types of exercise affect heart rate recovery time?"  
For each question, ask students: "Where could scientists get data to answer this question? Would it be primary or secondary data? What are some potential sources?" Class discussion to elicit different data source ideas.

- Main Activity 1: "Primary vs. Secondary Data Deep Dive" (20 mins):  
  
Explicitly define Primary Data (collected firsthand for a specific purpose) and Secondary Data (collected by others, for other purposes).  
Discuss the advantages and disadvantages of each data type in scientific research (e.g., primary - control, relevance, but time-consuming; secondary - readily available, large datasets, but potentially less relevant, quality concerns).  
Examples of Primary Data Collection Methods: Experiments, Surveys, Observations, Simulations. Briefly describe each method and when it is most appropriate.

- Explicitly define Primary Data (collected firsthand for a specific purpose) and Secondary Data (collected by others, for other purposes).

- Discuss the advantages and disadvantages of each data type in scientific research (e.g., primary - control, relevance, but time-consuming; secondary - readily available, large datasets, but potentially less relevant, quality concerns).

- Examples of Primary Data Collection Methods: Experiments, Surveys, Observations, Simulations. Briefly describe each method and when it is most appropriate.

- Main Activity 2: "Design Your Data Collection Plan" (30 mins):  
  
Divide students into small groups.  
Assign each group one or two scientific questions from the warm-up or provide new, more complex scientific questions suitable for Year 8 selective students (e.g., related to local environmental issues, human behaviour, simple physics phenomena).  
Each group task: Design a primary data collection plan to investigate their assigned question. The plan must include:  
  
Scientific Question (clearly stated).  
Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.  
Specific Data to be Collected (what variables will you measure or observe? Be precise).  
Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).  
Data Types (will you collect qualitative, quantitative, or both?).  
  
Groups work on their plans. Teacher circulates to provide guidance and feedback, encouraging more rigorous design and consideration of potential challenges.  
Groups briefly present their data collection plans to the class. Class feedback and discussion – peer review of plans.

- Divide students into small groups.

- Assign each group one or two scientific questions from the warm-up or provide new, more complex scientific questions suitable for Year 8 selective students (e.g., related to local environmental issues, human behaviour, simple physics phenomena).

- Each group task: Design a primary data collection plan to investigate their assigned question. The plan must include:  
  
Scientific Question (clearly stated).  
Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.  
Specific Data to be Collected (what variables will you measure or observe? Be precise).  
Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).  
Data Types (will you collect qualitative, quantitative, or both?).

- Scientific Question (clearly stated).

- Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.

- Specific Data to be Collected (what variables will you measure or observe? Be precise).

- Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).

- Data Types (will you collect qualitative, quantitative, or both?).

- Groups work on their plans. Teacher circulates to provide guidance and feedback, encouraging more rigorous design and consideration of potential challenges.

- Groups briefly present their data collection plans to the class. Class feedback and discussion – peer review of plans.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the key considerations when designing a good data collection plan for scientific research?  
What are some potential challenges or limitations you anticipate in carrying out your data collection plans?  
How does the choice of data collection method influence the type of data you get and the conclusions you can draw?

- What are the key considerations when designing a good data collection plan for scientific research?

- What are some potential challenges or limitations you anticipate in carrying out your data collection plans?

- How does the choice of data collection method influence the type of data you get and the conclusions you can draw?

- Assessments:  
  
Formative Task (5 mins): "Data Collection Method Matching" - Quick matching exercise (worksheet or digital) where students match scientific questions to appropriate primary data collection methods.  
Review of group data collection plans – assess clarity, completeness, and feasibility.

- Formative Task (5 mins): "Data Collection Method Matching" - Quick matching exercise (worksheet or digital) where students match scientific questions to appropriate primary data collection methods.

- Review of group data collection plans – assess clarity, completeness, and feasibility.

- Student Workbook:  
  
Definitions of primary and secondary data with advantages/disadvantages.  
Descriptions of primary data collection methods (experiments, surveys, observations, simulations).  
Template/worksheet for designing a data collection plan (with prompts for question, method, data, procedure, data types).

- Definitions of primary and secondary data with advantages/disadvantages.

- Descriptions of primary data collection methods (experiments, surveys, observations, simulations).

- Template/worksheet for designing a data collection plan (with prompts for question, method, data, procedure, data types).

- Extension Activities:  
  
For a chosen scientific question, design multiple data collection plans using different primary data collection methods and compare the potential data and insights each method could provide.  
Challenge: Research examples of real-world scientific studies that used specific primary data collection methods (e.g., a famous experiment, a large-scale survey). Analyze why that method was chosen and its strengths and limitations in that context.

- For a chosen scientific question, design multiple data collection plans using different primary data collection methods and compare the potential data and insights each method could provide.

- Challenge: Research examples of real-world scientific studies that used specific primary data collection methods (e.g., a famous experiment, a large-scale survey). Analyze why that method was chosen and its strengths and limitations in that context.

- "How does caffeine affect reaction time in teenagers?"

- "Is there a correlation between air pollution levels and respiratory illness in our city?"

- "What is the biodiversity of insect species in our schoolyard?"

- "How do different types of exercise affect heart rate recovery time?"  
For each question, ask students: "Where could scientists get data to answer this question? Would it be primary or secondary data? What are some potential sources?" Class discussion to elicit different data source ideas.

- Explicitly define Primary Data (collected firsthand for a specific purpose) and Secondary Data (collected by others, for other purposes).

- Discuss the advantages and disadvantages of each data type in scientific research (e.g., primary - control, relevance, but time-consuming; secondary - readily available, large datasets, but potentially less relevant, quality concerns).

- Examples of Primary Data Collection Methods: Experiments, Surveys, Observations, Simulations. Briefly describe each method and when it is most appropriate.

- Divide students into small groups.

- Assign each group one or two scientific questions from the warm-up or provide new, more complex scientific questions suitable for Year 8 selective students (e.g., related to local environmental issues, human behaviour, simple physics phenomena).

- Each group task: Design a primary data collection plan to investigate their assigned question. The plan must include:  
  
Scientific Question (clearly stated).  
Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.  
Specific Data to be Collected (what variables will you measure or observe? Be precise).  
Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).  
Data Types (will you collect qualitative, quantitative, or both?).

- Scientific Question (clearly stated).

- Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.

- Specific Data to be Collected (what variables will you measure or observe? Be precise).

- Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).

- Data Types (will you collect qualitative, quantitative, or both?).

- Groups work on their plans. Teacher circulates to provide guidance and feedback, encouraging more rigorous design and consideration of potential challenges.

- Groups briefly present their data collection plans to the class. Class feedback and discussion – peer review of plans.

- Scientific Question (clearly stated).

- Type of Primary Data Collection Method(s) to be used (Experiment, Survey, Observation, Simulation – or combination). Justify your choice.

- Specific Data to be Collected (what variables will you measure or observe? Be precise).

- Brief outline of the Procedure (how will you collect the data? What steps will you follow? Consider safety and ethical aspects).

- Data Types (will you collect qualitative, quantitative, or both?).

- What are the key considerations when designing a good data collection plan for scientific research?

- What are some potential challenges or limitations you anticipate in carrying out your data collection plans?

- How does the choice of data collection method influence the type of data you get and the conclusions you can draw?

- Formative Task (5 mins): "Data Collection Method Matching" - Quick matching exercise (worksheet or digital) where students match scientific questions to appropriate primary data collection methods.

- Review of group data collection plans – assess clarity, completeness, and feasibility.

- Definitions of primary and secondary data with advantages/disadvantages.

- Descriptions of primary data collection methods (experiments, surveys, observations, simulations).

- Template/worksheet for designing a data collection plan (with prompts for question, method, data, procedure, data types).

- For a chosen scientific question, design multiple data collection plans using different primary data collection methods and compare the potential data and insights each method could provide.

- Challenge: Research examples of real-world scientific studies that used specific primary data collection methods (e.g., a famous experiment, a large-scale survey). Analyze why that method was chosen and its strengths and limitations in that context.

- Slide Deck: Primary vs. secondary data definitions, examples of primary data collection methods, data collection plan design process.

- Handout: Data Source Challenge questions, Data Collection Plan template/worksheet, Data Collection Method Matching exercise.

### Lesson 2.3: Accuracy, Precision, and Validity: Ensuring Data Quality (Practical Activities)

- Title: Data Quality Control: Accuracy, Precision, and Validity in Scientific Measurement

- Learning Outcomes:  
  
Students will be able to define and differentiate between accuracy and precision in scientific measurements.  
Students will understand the concept of validity in data and experimental design and identify factors that affect validity.  
Students will be able to apply strategies to improve accuracy, precision, and validity in data collection.  
NSW Curriculum Outcomes: SC4-WS-02 (Planning investigations), SC4-WS-05 (Using data to develop conclusions), ACSIS125 (Conducting investigations safely), ACSIS131 (Evaluating methods and data).

- Students will be able to define and differentiate between accuracy and precision in scientific measurements.

- Students will understand the concept of validity in data and experimental design and identify factors that affect validity.

- Students will be able to apply strategies to improve accuracy, precision, and validity in data collection.

- NSW Curriculum Outcomes: SC4-WS-02 (Planning investigations), SC4-WS-05 (Using data to develop conclusions), ACSIS125 (Conducting investigations safely), ACSIS131 (Evaluating methods and data).

- Overview: This highly practical lesson focuses on data quality. Students will engage in hands-on activities to understand and improve accuracy, precision, and validity in scientific measurements. The lesson is designed to be interactive and challenging for selective students, emphasizing critical thinking and practical application.

- Activities:  
  
Warm-Up (10 mins): "Accuracy vs. Precision Challenge" - Use physical examples or visual aids (targets, diagrams) to quickly review the difference between accuracy and precision. Pose quick questions: "Which is more important in measuring the length of a critical component for a rocket: accuracy or precision?" "If you are repeatedly weighing samples of a chemical in an experiment, and all your measurements are very close together, but consistently slightly off the known mass, is your measurement precise or accurate or both or neither?" Class discussion to reinforce definitions.  
Main Activity 1: "Measurement Olympics" (30 mins): (Hands-on, stations-based activity)  
  
Set up different measurement stations around the classroom. Each station focuses on measuring a different physical quantity using different tools and methods. Examples:  
  
Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).  
Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.  
Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.  
Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).  
  
Students rotate through stations in small groups. At each station, they:  
  
Take multiple measurements of the same quantity using the assigned tools/methods.  
Record all measurements.  
Calculate the average and range of their measurements at each station.  
  
Class discussion after "Olympics":  
  
At which stations were measurements most precise? Least precise? Why?  
How could you improve precision at each station?  
Were there any measurements that you suspect were inaccurate? How would you check for accuracy?  
  
  
Main Activity 2: "Validity Challenge: Designing a Valid Experiment" (20 mins):  
  
Introduce the concept of Validity: Are we measuring what we intend to measure? Is our experiment designed to answer our scientific question without bias or confounding factors?  
Present a flawed experimental design scenario (related to a topic students are familiar with or from a previous lesson, e.g., investigating plant growth and sunlight, or reaction time and caffeine). The scenario should have clear flaws that affect validity (e.g., uncontrolled variables, biased sampling, lack of control group, unclear procedure).  
In groups, students analyse the flawed experimental design and:  
  
Identify specific threats to validity in the design.  
Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).  
  
Groups share their analysis and proposed improvements. Class discussion on strategies for ensuring validity in experimental design.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why are accuracy, precision, and validity all essential for high-quality scientific data?  
What are some practical steps scientists can take to maximize accuracy, precision, and validity in their research?  
How does understanding these concepts help you become a better data scientist and scientific investigator?  
  
Assessments:  
  
Formative Task (5 mins): "Accuracy, Precision, Validity Definitions" - Students write brief definitions of accuracy, precision, and validity in their own words.  
Observation of student engagement in "Measurement Olympics" and "Validity Challenge" activities, assessing their understanding of the concepts through their actions and discussions.  
  
Student Workbook:  
  
Definitions of accuracy, precision, and validity with examples.  
Tables to record measurements from "Measurement Olympics" and calculate averages/ranges.  
Worksheet for "Validity Challenge" with the flawed experimental design scenario and prompts for analysis and improvement suggestions.  
  
Extension Activities:  
  
Design an experiment specifically focused on comparing the accuracy and precision of different measurement tools or methods for the same quantity. Conduct the experiment and analyse the results.  
Research examples of scientific studies where issues of accuracy, precision, or validity have been critical (e.g., in medical trials, environmental monitoring, forensic science). Present findings and discuss the consequences of data quality problems in those cases.

- Warm-Up (10 mins): "Accuracy vs. Precision Challenge" - Use physical examples or visual aids (targets, diagrams) to quickly review the difference between accuracy and precision. Pose quick questions: "Which is more important in measuring the length of a critical component for a rocket: accuracy or precision?" "If you are repeatedly weighing samples of a chemical in an experiment, and all your measurements are very close together, but consistently slightly off the known mass, is your measurement precise or accurate or both or neither?" Class discussion to reinforce definitions.

- Main Activity 1: "Measurement Olympics" (30 mins): (Hands-on, stations-based activity)  
  
Set up different measurement stations around the classroom. Each station focuses on measuring a different physical quantity using different tools and methods. Examples:  
  
Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).  
Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.  
Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.  
Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).  
  
Students rotate through stations in small groups. At each station, they:  
  
Take multiple measurements of the same quantity using the assigned tools/methods.  
Record all measurements.  
Calculate the average and range of their measurements at each station.  
  
Class discussion after "Olympics":  
  
At which stations were measurements most precise? Least precise? Why?  
How could you improve precision at each station?  
Were there any measurements that you suspect were inaccurate? How would you check for accuracy?

- Set up different measurement stations around the classroom. Each station focuses on measuring a different physical quantity using different tools and methods. Examples:  
  
Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).  
Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.  
Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.  
Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).

- Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).

- Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.

- Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.

- Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).

- Students rotate through stations in small groups. At each station, they:  
  
Take multiple measurements of the same quantity using the assigned tools/methods.  
Record all measurements.  
Calculate the average and range of their measurements at each station.

- Take multiple measurements of the same quantity using the assigned tools/methods.

- Record all measurements.

- Calculate the average and range of their measurements at each station.

- Class discussion after "Olympics":  
  
At which stations were measurements most precise? Least precise? Why?  
How could you improve precision at each station?  
Were there any measurements that you suspect were inaccurate? How would you check for accuracy?

- At which stations were measurements most precise? Least precise? Why?

- How could you improve precision at each station?

- Were there any measurements that you suspect were inaccurate? How would you check for accuracy?

- Main Activity 2: "Validity Challenge: Designing a Valid Experiment" (20 mins):  
  
Introduce the concept of Validity: Are we measuring what we intend to measure? Is our experiment designed to answer our scientific question without bias or confounding factors?  
Present a flawed experimental design scenario (related to a topic students are familiar with or from a previous lesson, e.g., investigating plant growth and sunlight, or reaction time and caffeine). The scenario should have clear flaws that affect validity (e.g., uncontrolled variables, biased sampling, lack of control group, unclear procedure).  
In groups, students analyse the flawed experimental design and:  
  
Identify specific threats to validity in the design.  
Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).  
  
Groups share their analysis and proposed improvements. Class discussion on strategies for ensuring validity in experimental design.

- Introduce the concept of Validity: Are we measuring what we intend to measure? Is our experiment designed to answer our scientific question without bias or confounding factors?

- Present a flawed experimental design scenario (related to a topic students are familiar with or from a previous lesson, e.g., investigating plant growth and sunlight, or reaction time and caffeine). The scenario should have clear flaws that affect validity (e.g., uncontrolled variables, biased sampling, lack of control group, unclear procedure).

- In groups, students analyse the flawed experimental design and:  
  
Identify specific threats to validity in the design.  
Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).

- Identify specific threats to validity in the design.

- Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).

- Groups share their analysis and proposed improvements. Class discussion on strategies for ensuring validity in experimental design.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are accuracy, precision, and validity all essential for high-quality scientific data?  
What are some practical steps scientists can take to maximize accuracy, precision, and validity in their research?  
How does understanding these concepts help you become a better data scientist and scientific investigator?

- Why are accuracy, precision, and validity all essential for high-quality scientific data?

- What are some practical steps scientists can take to maximize accuracy, precision, and validity in their research?

- How does understanding these concepts help you become a better data scientist and scientific investigator?

- Assessments:  
  
Formative Task (5 mins): "Accuracy, Precision, Validity Definitions" - Students write brief definitions of accuracy, precision, and validity in their own words.  
Observation of student engagement in "Measurement Olympics" and "Validity Challenge" activities, assessing their understanding of the concepts through their actions and discussions.

- Formative Task (5 mins): "Accuracy, Precision, Validity Definitions" - Students write brief definitions of accuracy, precision, and validity in their own words.

- Observation of student engagement in "Measurement Olympics" and "Validity Challenge" activities, assessing their understanding of the concepts through their actions and discussions.

- Student Workbook:  
  
Definitions of accuracy, precision, and validity with examples.  
Tables to record measurements from "Measurement Olympics" and calculate averages/ranges.  
Worksheet for "Validity Challenge" with the flawed experimental design scenario and prompts for analysis and improvement suggestions.

- Definitions of accuracy, precision, and validity with examples.

- Tables to record measurements from "Measurement Olympics" and calculate averages/ranges.

- Worksheet for "Validity Challenge" with the flawed experimental design scenario and prompts for analysis and improvement suggestions.

- Extension Activities:  
  
Design an experiment specifically focused on comparing the accuracy and precision of different measurement tools or methods for the same quantity. Conduct the experiment and analyse the results.  
Research examples of scientific studies where issues of accuracy, precision, or validity have been critical (e.g., in medical trials, environmental monitoring, forensic science). Present findings and discuss the consequences of data quality problems in those cases.

- Design an experiment specifically focused on comparing the accuracy and precision of different measurement tools or methods for the same quantity. Conduct the experiment and analyse the results.

- Research examples of scientific studies where issues of accuracy, precision, or validity have been critical (e.g., in medical trials, environmental monitoring, forensic science). Present findings and discuss the consequences of data quality problems in those cases.

- Lesson Materials:  
  
Slide Deck: Accuracy, precision, validity definitions, examples, strategies for improvement, experimental validity concepts.  
Handout: "Measurement Olympics" station instructions and recording sheets, "Validity Challenge" flawed experiment scenario worksheet, definitions of accuracy, precision, validity.  
Materials for "Measurement Olympics" stations: Rulers, measuring tapes, digital calipers (optional), various objects for length measurement; electronic balances, triple beam balances, objects for mass measurement; graduated cylinders, beakers, pipettes, liquids for volume measurement; stopwatches (digital preferable), materials for time measurement events (pendulums, ramps, balls).

- Slide Deck: Accuracy, precision, validity definitions, examples, strategies for improvement, experimental validity concepts.

- Handout: "Measurement Olympics" station instructions and recording sheets, "Validity Challenge" flawed experiment scenario worksheet, definitions of accuracy, precision, validity.

- Materials for "Measurement Olympics" stations: Rulers, measuring tapes, digital calipers (optional), various objects for length measurement; electronic balances, triple beam balances, objects for mass measurement; graduated cylinders, beakers, pipettes, liquids for volume measurement; stopwatches (digital preferable), materials for time measurement events (pendulums, ramps, balls).

- Students will be able to define and differentiate between accuracy and precision in scientific measurements.

- Students will understand the concept of validity in data and experimental design and identify factors that affect validity.

- Students will be able to apply strategies to improve accuracy, precision, and validity in data collection.

- NSW Curriculum Outcomes: SC4-WS-02 (Planning investigations), SC4-WS-05 (Using data to develop conclusions), ACSIS125 (Conducting investigations safely), ACSIS131 (Evaluating methods and data).

- Warm-Up (10 mins): "Accuracy vs. Precision Challenge" - Use physical examples or visual aids (targets, diagrams) to quickly review the difference between accuracy and precision. Pose quick questions: "Which is more important in measuring the length of a critical component for a rocket: accuracy or precision?" "If you are repeatedly weighing samples of a chemical in an experiment, and all your measurements are very close together, but consistently slightly off the known mass, is your measurement precise or accurate or both or neither?" Class discussion to reinforce definitions.

- Main Activity 1: "Measurement Olympics" (30 mins): (Hands-on, stations-based activity)  
  
Set up different measurement stations around the classroom. Each station focuses on measuring a different physical quantity using different tools and methods. Examples:  
  
Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).  
Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.  
Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.  
Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).  
  
Students rotate through stations in small groups. At each station, they:  
  
Take multiple measurements of the same quantity using the assigned tools/methods.  
Record all measurements.  
Calculate the average and range of their measurements at each station.  
  
Class discussion after "Olympics":  
  
At which stations were measurements most precise? Least precise? Why?  
How could you improve precision at each station?  
Were there any measurements that you suspect were inaccurate? How would you check for accuracy?

- Set up different measurement stations around the classroom. Each station focuses on measuring a different physical quantity using different tools and methods. Examples:  
  
Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).  
Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.  
Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.  
Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).

- Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).

- Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.

- Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.

- Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).

- Students rotate through stations in small groups. At each station, they:  
  
Take multiple measurements of the same quantity using the assigned tools/methods.  
Record all measurements.  
Calculate the average and range of their measurements at each station.

- Take multiple measurements of the same quantity using the assigned tools/methods.

- Record all measurements.

- Calculate the average and range of their measurements at each station.

- Class discussion after "Olympics":  
  
At which stations were measurements most precise? Least precise? Why?  
How could you improve precision at each station?  
Were there any measurements that you suspect were inaccurate? How would you check for accuracy?

- At which stations were measurements most precise? Least precise? Why?

- How could you improve precision at each station?

- Were there any measurements that you suspect were inaccurate? How would you check for accuracy?

- Main Activity 2: "Validity Challenge: Designing a Valid Experiment" (20 mins):  
  
Introduce the concept of Validity: Are we measuring what we intend to measure? Is our experiment designed to answer our scientific question without bias or confounding factors?  
Present a flawed experimental design scenario (related to a topic students are familiar with or from a previous lesson, e.g., investigating plant growth and sunlight, or reaction time and caffeine). The scenario should have clear flaws that affect validity (e.g., uncontrolled variables, biased sampling, lack of control group, unclear procedure).  
In groups, students analyse the flawed experimental design and:  
  
Identify specific threats to validity in the design.  
Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).  
  
Groups share their analysis and proposed improvements. Class discussion on strategies for ensuring validity in experimental design.

- Introduce the concept of Validity: Are we measuring what we intend to measure? Is our experiment designed to answer our scientific question without bias or confounding factors?

- Present a flawed experimental design scenario (related to a topic students are familiar with or from a previous lesson, e.g., investigating plant growth and sunlight, or reaction time and caffeine). The scenario should have clear flaws that affect validity (e.g., uncontrolled variables, biased sampling, lack of control group, unclear procedure).

- In groups, students analyse the flawed experimental design and:  
  
Identify specific threats to validity in the design.  
Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).

- Identify specific threats to validity in the design.

- Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).

- Groups share their analysis and proposed improvements. Class discussion on strategies for ensuring validity in experimental design.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are accuracy, precision, and validity all essential for high-quality scientific data?  
What are some practical steps scientists can take to maximize accuracy, precision, and validity in their research?  
How does understanding these concepts help you become a better data scientist and scientific investigator?

- Why are accuracy, precision, and validity all essential for high-quality scientific data?

- What are some practical steps scientists can take to maximize accuracy, precision, and validity in their research?

- How does understanding these concepts help you become a better data scientist and scientific investigator?

- Assessments:  
  
Formative Task (5 mins): "Accuracy, Precision, Validity Definitions" - Students write brief definitions of accuracy, precision, and validity in their own words.  
Observation of student engagement in "Measurement Olympics" and "Validity Challenge" activities, assessing their understanding of the concepts through their actions and discussions.

- Formative Task (5 mins): "Accuracy, Precision, Validity Definitions" - Students write brief definitions of accuracy, precision, and validity in their own words.

- Observation of student engagement in "Measurement Olympics" and "Validity Challenge" activities, assessing their understanding of the concepts through their actions and discussions.

- Student Workbook:  
  
Definitions of accuracy, precision, and validity with examples.  
Tables to record measurements from "Measurement Olympics" and calculate averages/ranges.  
Worksheet for "Validity Challenge" with the flawed experimental design scenario and prompts for analysis and improvement suggestions.

- Definitions of accuracy, precision, and validity with examples.

- Tables to record measurements from "Measurement Olympics" and calculate averages/ranges.

- Worksheet for "Validity Challenge" with the flawed experimental design scenario and prompts for analysis and improvement suggestions.

- Extension Activities:  
  
Design an experiment specifically focused on comparing the accuracy and precision of different measurement tools or methods for the same quantity. Conduct the experiment and analyse the results.  
Research examples of scientific studies where issues of accuracy, precision, or validity have been critical (e.g., in medical trials, environmental monitoring, forensic science). Present findings and discuss the consequences of data quality problems in those cases.

- Design an experiment specifically focused on comparing the accuracy and precision of different measurement tools or methods for the same quantity. Conduct the experiment and analyse the results.

- Research examples of scientific studies where issues of accuracy, precision, or validity have been critical (e.g., in medical trials, environmental monitoring, forensic science). Present findings and discuss the consequences of data quality problems in those cases.

- Set up different measurement stations around the classroom. Each station focuses on measuring a different physical quantity using different tools and methods. Examples:  
  
Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).  
Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.  
Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.  
Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).

- Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).

- Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.

- Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.

- Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).

- Students rotate through stations in small groups. At each station, they:  
  
Take multiple measurements of the same quantity using the assigned tools/methods.  
Record all measurements.  
Calculate the average and range of their measurements at each station.

- Take multiple measurements of the same quantity using the assigned tools/methods.

- Record all measurements.

- Calculate the average and range of their measurements at each station.

- Class discussion after "Olympics":  
  
At which stations were measurements most precise? Least precise? Why?  
How could you improve precision at each station?  
Were there any measurements that you suspect were inaccurate? How would you check for accuracy?

- At which stations were measurements most precise? Least precise? Why?

- How could you improve precision at each station?

- Were there any measurements that you suspect were inaccurate? How would you check for accuracy?

- Station 1: Length Measurement: Measure the length of a complex object (e.g., a curved object, irregular shape) using rulers, measuring tapes, and potentially digital calipers (if available).

- Station 2: Mass Measurement: Measure the mass of several objects using different types of balances (e.g., electronic balance, triple beam balance). Include objects with slightly varying masses.

- Station 3: Volume Measurement: Measure the volume of liquids using graduated cylinders, beakers, and pipettes. Emphasize reading meniscus correctly.

- Station 4: Time Measurement: Measure the time for a simple event to occur repeatedly (e.g., pendulum swing, ball rolling down a ramp) using stopwatches (potentially digital stopwatches).

- Take multiple measurements of the same quantity using the assigned tools/methods.

- Record all measurements.

- Calculate the average and range of their measurements at each station.

- At which stations were measurements most precise? Least precise? Why?

- How could you improve precision at each station?

- Were there any measurements that you suspect were inaccurate? How would you check for accuracy?

- Introduce the concept of Validity: Are we measuring what we intend to measure? Is our experiment designed to answer our scientific question without bias or confounding factors?

- Present a flawed experimental design scenario (related to a topic students are familiar with or from a previous lesson, e.g., investigating plant growth and sunlight, or reaction time and caffeine). The scenario should have clear flaws that affect validity (e.g., uncontrolled variables, biased sampling, lack of control group, unclear procedure).

- In groups, students analyse the flawed experimental design and:  
  
Identify specific threats to validity in the design.  
Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).

- Identify specific threats to validity in the design.

- Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).

- Groups share their analysis and proposed improvements. Class discussion on strategies for ensuring validity in experimental design.

- Identify specific threats to validity in the design.

- Propose specific improvements to the experimental design to increase its validity (e.g., how to control variables, improve randomization, add a control group, refine procedure).

- Why are accuracy, precision, and validity all essential for high-quality scientific data?

- What are some practical steps scientists can take to maximize accuracy, precision, and validity in their research?

- How does understanding these concepts help you become a better data scientist and scientific investigator?

- Formative Task (5 mins): "Accuracy, Precision, Validity Definitions" - Students write brief definitions of accuracy, precision, and validity in their own words.

- Observation of student engagement in "Measurement Olympics" and "Validity Challenge" activities, assessing their understanding of the concepts through their actions and discussions.

- Definitions of accuracy, precision, and validity with examples.

- Tables to record measurements from "Measurement Olympics" and calculate averages/ranges.

- Worksheet for "Validity Challenge" with the flawed experimental design scenario and prompts for analysis and improvement suggestions.

- Design an experiment specifically focused on comparing the accuracy and precision of different measurement tools or methods for the same quantity. Conduct the experiment and analyse the results.

- Research examples of scientific studies where issues of accuracy, precision, or validity have been critical (e.g., in medical trials, environmental monitoring, forensic science). Present findings and discuss the consequences of data quality problems in those cases.

- Slide Deck: Accuracy, precision, validity definitions, examples, strategies for improvement, experimental validity concepts.

- Handout: "Measurement Olympics" station instructions and recording sheets, "Validity Challenge" flawed experiment scenario worksheet, definitions of accuracy, precision, validity.

- Materials for "Measurement Olympics" stations: Rulers, measuring tapes, digital calipers (optional), various objects for length measurement; electronic balances, triple beam balances, objects for mass measurement; graduated cylinders, beakers, pipettes, liquids for volume measurement; stopwatches (digital preferable), materials for time measurement events (pendulums, ramps, balls).

### Week 3: Data Visualization & Interactive Storytelling

### Lesson 3.1: Data Display: Tables - Organising Information

- Title: Data in Order: Mastering Tables for Scientific Clarity

- Learning Outcomes:  
  
Students will be able to construct tables to effectively organise and present scientific data.  
Students will be able to identify the key components of a well-designed table (title, headings, data cells, units).  
Students will be able to choose appropriate table formats for different types of scientific data.  
Students will be able to interpret data presented in tables and draw basic conclusions.  
NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information)

- Students will be able to construct tables to effectively organise and present scientific data.

- Students will be able to identify the key components of a well-designed table (title, headings, data cells, units).

- Students will be able to choose appropriate table formats for different types of scientific data.

- Students will be able to interpret data presented in tables and draw basic conclusions.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information)

- Overview: This lesson focuses on tables as a fundamental method for organising and presenting scientific data. Students will learn the principles of table design, practice constructing tables, and develop skills in interpreting tabular data to extract information and draw conclusions.

- Activities:  
  
Warm-Up (10 mins): "Information Overload Challenge" - Present students with a paragraph of raw scientific data (e.g., a list of plant heights, temperatures, reaction times in text format). Ask them to quickly find specific pieces of information (e.g., "What is the maximum temperature?", "What is the height of plant number 3?"). Discuss how difficult it is to extract information from unstructured text data. Lead to the need for organised data presentation methods like tables.  
Main Activity 1: "Anatomy of a Table - Deconstructing Good Design" (25 mins):  
  
Introduce the key components of a scientific table:  
  
Table Number and Title: Clear, concise, informative title describing the table's content.  
Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).  
Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.  
Data Cells: Cells containing the actual data values.  
Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.  
  
Show examples of well-designed and poorly designed scientific tables. For the poorly designed examples, ask students to identify what is wrong and how it could be improved. Focus on clarity, organization, and completeness of information.  
"Table Critique": Provide students with poorly designed tables (pre-prepared or from scientific publications - simplified if necessary). In groups, students critique the tables, identifying weaknesses and suggesting specific improvements to enhance clarity and readability.  
  
Main Activity 2: "Table Construction Challenge - From Raw Data to Organised Table" (20 mins):  
  
Provide students with sets of raw scientific data (different datasets for different groups - e.g., plant growth data, reaction time data, survey results). Data can be provided in text lists or simple spreadsheets.  
Each group task: Construct a clear and well-designed table to present their assigned dataset. They need to decide on appropriate column and row headings, units, and table title. Encourage them to consider the most effective way to organise the data for clarity and easy interpretation.  
Groups share their constructed tables with the class. Class feedback and discussion – peer review of table design. Discuss different approaches and best practices for table construction.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why are tables such a fundamental and important tool for presenting scientific data?  
What are the key principles of good table design for scientific communication?  
When is it most appropriate to use a table to present data, compared to other visualization methods like graphs?  
  
Assessments:  
  
Formative Task (5 mins): "Table Component Identification" - Provide a well-designed table. Ask students to label and identify the key components (title, headings, data cells, units).  
Review of group-constructed tables – assess clarity, completeness, and adherence to table design principles.  
  
Student Workbook:  
  
Diagram of a well-designed scientific table with labels for key components and descriptions.  
Checklist of criteria for good table design.  
Raw data sets for table construction exercises.  
"Table Critique" worksheet with poorly designed tables and prompts for analysis and improvement suggestions.  
  
Extension Activities:  
  
Students find scientific tables in research papers or reports and analyse their design. Critique their effectiveness and suggest any improvements.  
Challenge: Explore more advanced table formatting options (e.g., using borders, shading, conditional formatting – in software like spreadsheets or online table generators). Create tables with more complex data structures (e.g., hierarchical tables, tables with multiple levels of headings).

- Warm-Up (10 mins): "Information Overload Challenge" - Present students with a paragraph of raw scientific data (e.g., a list of plant heights, temperatures, reaction times in text format). Ask them to quickly find specific pieces of information (e.g., "What is the maximum temperature?", "What is the height of plant number 3?"). Discuss how difficult it is to extract information from unstructured text data. Lead to the need for organised data presentation methods like tables.

- Main Activity 1: "Anatomy of a Table - Deconstructing Good Design" (25 mins):  
  
Introduce the key components of a scientific table:  
  
Table Number and Title: Clear, concise, informative title describing the table's content.  
Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).  
Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.  
Data Cells: Cells containing the actual data values.  
Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.  
  
Show examples of well-designed and poorly designed scientific tables. For the poorly designed examples, ask students to identify what is wrong and how it could be improved. Focus on clarity, organization, and completeness of information.  
"Table Critique": Provide students with poorly designed tables (pre-prepared or from scientific publications - simplified if necessary). In groups, students critique the tables, identifying weaknesses and suggesting specific improvements to enhance clarity and readability.

- Introduce the key components of a scientific table:  
  
Table Number and Title: Clear, concise, informative title describing the table's content.  
Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).  
Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.  
Data Cells: Cells containing the actual data values.  
Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.

- Table Number and Title: Clear, concise, informative title describing the table's content.

- Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).

- Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.

- Data Cells: Cells containing the actual data values.

- Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.

- Show examples of well-designed and poorly designed scientific tables. For the poorly designed examples, ask students to identify what is wrong and how it could be improved. Focus on clarity, organization, and completeness of information.

- "Table Critique": Provide students with poorly designed tables (pre-prepared or from scientific publications - simplified if necessary). In groups, students critique the tables, identifying weaknesses and suggesting specific improvements to enhance clarity and readability.

- Main Activity 2: "Table Construction Challenge - From Raw Data to Organised Table" (20 mins):  
  
Provide students with sets of raw scientific data (different datasets for different groups - e.g., plant growth data, reaction time data, survey results). Data can be provided in text lists or simple spreadsheets.  
Each group task: Construct a clear and well-designed table to present their assigned dataset. They need to decide on appropriate column and row headings, units, and table title. Encourage them to consider the most effective way to organise the data for clarity and easy interpretation.  
Groups share their constructed tables with the class. Class feedback and discussion – peer review of table design. Discuss different approaches and best practices for table construction.

- Provide students with sets of raw scientific data (different datasets for different groups - e.g., plant growth data, reaction time data, survey results). Data can be provided in text lists or simple spreadsheets.

- Each group task: Construct a clear and well-designed table to present their assigned dataset. They need to decide on appropriate column and row headings, units, and table title. Encourage them to consider the most effective way to organise the data for clarity and easy interpretation.

- Groups share their constructed tables with the class. Class feedback and discussion – peer review of table design. Discuss different approaches and best practices for table construction.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are tables such a fundamental and important tool for presenting scientific data?  
What are the key principles of good table design for scientific communication?  
When is it most appropriate to use a table to present data, compared to other visualization methods like graphs?

- Why are tables such a fundamental and important tool for presenting scientific data?

- What are the key principles of good table design for scientific communication?

- When is it most appropriate to use a table to present data, compared to other visualization methods like graphs?

- Assessments:  
  
Formative Task (5 mins): "Table Component Identification" - Provide a well-designed table. Ask students to label and identify the key components (title, headings, data cells, units).  
Review of group-constructed tables – assess clarity, completeness, and adherence to table design principles.

- Formative Task (5 mins): "Table Component Identification" - Provide a well-designed table. Ask students to label and identify the key components (title, headings, data cells, units).

- Review of group-constructed tables – assess clarity, completeness, and adherence to table design principles.

- Student Workbook:  
  
Diagram of a well-designed scientific table with labels for key components and descriptions.  
Checklist of criteria for good table design.  
Raw data sets for table construction exercises.  
"Table Critique" worksheet with poorly designed tables and prompts for analysis and improvement suggestions.

- Diagram of a well-designed scientific table with labels for key components and descriptions.

- Checklist of criteria for good table design.

- Raw data sets for table construction exercises.

- "Table Critique" worksheet with poorly designed tables and prompts for analysis and improvement suggestions.

- Extension Activities:  
  
Students find scientific tables in research papers or reports and analyse their design. Critique their effectiveness and suggest any improvements.  
Challenge: Explore more advanced table formatting options (e.g., using borders, shading, conditional formatting – in software like spreadsheets or online table generators). Create tables with more complex data structures (e.g., hierarchical tables, tables with multiple levels of headings).

- Students find scientific tables in research papers or reports and analyse their design. Critique their effectiveness and suggest any improvements.

- Challenge: Explore more advanced table formatting options (e.g., using borders, shading, conditional formatting – in software like spreadsheets or online table generators). Create tables with more complex data structures (e.g., hierarchical tables, tables with multiple levels of headings).

- Lesson Materials:  
  
Slide Deck: Introduction to tables, components of a well-designed table, examples of good and bad tables, table design principles.  
Handout: Raw data sets for table construction exercises, "Table Critique" worksheet with poorly designed tables, checklist for good table design.  
Formative Task (printable or digital).

- Slide Deck: Introduction to tables, components of a well-designed table, examples of good and bad tables, table design principles.

- Handout: Raw data sets for table construction exercises, "Table Critique" worksheet with poorly designed tables, checklist for good table design.

- Formative Task (printable or digital).

- Students will be able to construct tables to effectively organise and present scientific data.

- Students will be able to identify the key components of a well-designed table (title, headings, data cells, units).

- Students will be able to choose appropriate table formats for different types of scientific data.

- Students will be able to interpret data presented in tables and draw basic conclusions.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information)

- Warm-Up (10 mins): "Information Overload Challenge" - Present students with a paragraph of raw scientific data (e.g., a list of plant heights, temperatures, reaction times in text format). Ask them to quickly find specific pieces of information (e.g., "What is the maximum temperature?", "What is the height of plant number 3?"). Discuss how difficult it is to extract information from unstructured text data. Lead to the need for organised data presentation methods like tables.

- Main Activity 1: "Anatomy of a Table - Deconstructing Good Design" (25 mins):  
  
Introduce the key components of a scientific table:  
  
Table Number and Title: Clear, concise, informative title describing the table's content.  
Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).  
Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.  
Data Cells: Cells containing the actual data values.  
Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.  
  
Show examples of well-designed and poorly designed scientific tables. For the poorly designed examples, ask students to identify what is wrong and how it could be improved. Focus on clarity, organization, and completeness of information.  
"Table Critique": Provide students with poorly designed tables (pre-prepared or from scientific publications - simplified if necessary). In groups, students critique the tables, identifying weaknesses and suggesting specific improvements to enhance clarity and readability.

- Introduce the key components of a scientific table:  
  
Table Number and Title: Clear, concise, informative title describing the table's content.  
Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).  
Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.  
Data Cells: Cells containing the actual data values.  
Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.

- Table Number and Title: Clear, concise, informative title describing the table's content.

- Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).

- Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.

- Data Cells: Cells containing the actual data values.

- Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.

- Show examples of well-designed and poorly designed scientific tables. For the poorly designed examples, ask students to identify what is wrong and how it could be improved. Focus on clarity, organization, and completeness of information.

- "Table Critique": Provide students with poorly designed tables (pre-prepared or from scientific publications - simplified if necessary). In groups, students critique the tables, identifying weaknesses and suggesting specific improvements to enhance clarity and readability.

- Main Activity 2: "Table Construction Challenge - From Raw Data to Organised Table" (20 mins):  
  
Provide students with sets of raw scientific data (different datasets for different groups - e.g., plant growth data, reaction time data, survey results). Data can be provided in text lists or simple spreadsheets.  
Each group task: Construct a clear and well-designed table to present their assigned dataset. They need to decide on appropriate column and row headings, units, and table title. Encourage them to consider the most effective way to organise the data for clarity and easy interpretation.  
Groups share their constructed tables with the class. Class feedback and discussion – peer review of table design. Discuss different approaches and best practices for table construction.

- Provide students with sets of raw scientific data (different datasets for different groups - e.g., plant growth data, reaction time data, survey results). Data can be provided in text lists or simple spreadsheets.

- Each group task: Construct a clear and well-designed table to present their assigned dataset. They need to decide on appropriate column and row headings, units, and table title. Encourage them to consider the most effective way to organise the data for clarity and easy interpretation.

- Groups share their constructed tables with the class. Class feedback and discussion – peer review of table design. Discuss different approaches and best practices for table construction.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are tables such a fundamental and important tool for presenting scientific data?  
What are the key principles of good table design for scientific communication?  
When is it most appropriate to use a table to present data, compared to other visualization methods like graphs?

- Why are tables such a fundamental and important tool for presenting scientific data?

- What are the key principles of good table design for scientific communication?

- When is it most appropriate to use a table to present data, compared to other visualization methods like graphs?

- Assessments:  
  
Formative Task (5 mins): "Table Component Identification" - Provide a well-designed table. Ask students to label and identify the key components (title, headings, data cells, units).  
Review of group-constructed tables – assess clarity, completeness, and adherence to table design principles.

- Formative Task (5 mins): "Table Component Identification" - Provide a well-designed table. Ask students to label and identify the key components (title, headings, data cells, units).

- Review of group-constructed tables – assess clarity, completeness, and adherence to table design principles.

- Student Workbook:  
  
Diagram of a well-designed scientific table with labels for key components and descriptions.  
Checklist of criteria for good table design.  
Raw data sets for table construction exercises.  
"Table Critique" worksheet with poorly designed tables and prompts for analysis and improvement suggestions.

- Diagram of a well-designed scientific table with labels for key components and descriptions.

- Checklist of criteria for good table design.

- Raw data sets for table construction exercises.

- "Table Critique" worksheet with poorly designed tables and prompts for analysis and improvement suggestions.

- Extension Activities:  
  
Students find scientific tables in research papers or reports and analyse their design. Critique their effectiveness and suggest any improvements.  
Challenge: Explore more advanced table formatting options (e.g., using borders, shading, conditional formatting – in software like spreadsheets or online table generators). Create tables with more complex data structures (e.g., hierarchical tables, tables with multiple levels of headings).

- Students find scientific tables in research papers or reports and analyse their design. Critique their effectiveness and suggest any improvements.

- Challenge: Explore more advanced table formatting options (e.g., using borders, shading, conditional formatting – in software like spreadsheets or online table generators). Create tables with more complex data structures (e.g., hierarchical tables, tables with multiple levels of headings).

- Introduce the key components of a scientific table:  
  
Table Number and Title: Clear, concise, informative title describing the table's content.  
Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).  
Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.  
Data Cells: Cells containing the actual data values.  
Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.

- Table Number and Title: Clear, concise, informative title describing the table's content.

- Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).

- Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.

- Data Cells: Cells containing the actual data values.

- Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.

- Show examples of well-designed and poorly designed scientific tables. For the poorly designed examples, ask students to identify what is wrong and how it could be improved. Focus on clarity, organization, and completeness of information.

- "Table Critique": Provide students with poorly designed tables (pre-prepared or from scientific publications - simplified if necessary). In groups, students critique the tables, identifying weaknesses and suggesting specific improvements to enhance clarity and readability.

- Table Number and Title: Clear, concise, informative title describing the table's content.

- Column Headings: Descriptive headings for each column, indicating the variable being measured and units of measurement (if applicable).

- Row Headings (or first column): Labels for each row, indicating categories or experimental conditions.

- Data Cells: Cells containing the actual data values.

- Units of Measurement: Clearly stated units (e.g., cm, °C, seconds) in column headings or below table if consistent for all values.

- Provide students with sets of raw scientific data (different datasets for different groups - e.g., plant growth data, reaction time data, survey results). Data can be provided in text lists or simple spreadsheets.

- Each group task: Construct a clear and well-designed table to present their assigned dataset. They need to decide on appropriate column and row headings, units, and table title. Encourage them to consider the most effective way to organise the data for clarity and easy interpretation.

- Groups share their constructed tables with the class. Class feedback and discussion – peer review of table design. Discuss different approaches and best practices for table construction.

- Why are tables such a fundamental and important tool for presenting scientific data?

- What are the key principles of good table design for scientific communication?

- When is it most appropriate to use a table to present data, compared to other visualization methods like graphs?

- Formative Task (5 mins): "Table Component Identification" - Provide a well-designed table. Ask students to label and identify the key components (title, headings, data cells, units).

- Review of group-constructed tables – assess clarity, completeness, and adherence to table design principles.

- Diagram of a well-designed scientific table with labels for key components and descriptions.

- Checklist of criteria for good table design.

- Raw data sets for table construction exercises.

- "Table Critique" worksheet with poorly designed tables and prompts for analysis and improvement suggestions.

- Students find scientific tables in research papers or reports and analyse their design. Critique their effectiveness and suggest any improvements.

- Challenge: Explore more advanced table formatting options (e.g., using borders, shading, conditional formatting – in software like spreadsheets or online table generators). Create tables with more complex data structures (e.g., hierarchical tables, tables with multiple levels of headings).

- Slide Deck: Introduction to tables, components of a well-designed table, examples of good and bad tables, table design principles.

- Handout: Raw data sets for table construction exercises, "Table Critique" worksheet with poorly designed tables, checklist for good table design.

- Formative Task (printable or digital).

### Lesson 3.2: Visualizing Relationships: Bar Graphs

- Title: Bar Graphs: Comparing Categories and Showing Differences

- Learning Outcomes:  
  
Students will be able to construct bar graphs to visually represent categorical data and compare quantities across different categories.  
Students will be able to identify the key components of a bar graph (axes, bars, labels, title, scale).  
Students will be able to choose appropriate bar graph types (vertical, horizontal, grouped) for different datasets and analytical goals.  
Students will be able to interpret bar graphs and extract meaningful comparisons and conclusions from them.  
NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Students will be able to construct bar graphs to visually represent categorical data and compare quantities across different categories.

- Students will be able to identify the key components of a bar graph (axes, bars, labels, title, scale).

- Students will be able to choose appropriate bar graph types (vertical, horizontal, grouped) for different datasets and analytical goals.

- Students will be able to interpret bar graphs and extract meaningful comparisons and conclusions from them.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Overview: This lesson introduces bar graphs as a powerful visualization tool for comparing categorical data. Students will learn the components of bar graphs, practice constructing them, and develop skills in interpreting bar graphs to identify comparisons and draw conclusions about categorical relationships in scientific data.

- Activities:  
  
Warm-Up (10 mins): "Categorical Data Brainstorm" - Ask students to brainstorm examples of categorical data in science and everyday life (e.g., types of animals, colours of flowers, types of rocks, favourite subjects, survey responses with categories). List these on the board. Discuss: What kind of questions can we answer with categorical data? How can we visually compare categories?  
Main Activity 1: "Anatomy of a Bar Graph - Building Blocks of Comparison" (25 mins):  
  
Introduce the key components of a bar graph:  
  
Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).  
Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.  
Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.  
Title: Clear, concise title describing what the bar graph represents.  
Labels: Category labels along the x-axis, and y-axis labels with units.  
  
Show examples of well-designed and poorly designed bar graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of data, and effective visual comparison of categories.  
Introduce different types of bar graphs:  
  
Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.  
Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.  
Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.  
  
"Bar Graph Type Match": Present scenarios or research questions with categorical data. Students identify which bar graph type (vertical, horizontal, grouped) would be most appropriate and justify their choice.  
  
Main Activity 2: "Bar Graph Construction Challenge - From Data to Visual Comparison" (20 mins):  
  
Provide students with datasets suitable for bar graphs (different datasets for different groups - e.g., species counts in different habitats, survey results on favourite science topics, experimental results comparing different treatments). Data can be in tables or lists.  
Each group task: Construct a bar graph (using paper and pencil initially, or digital tools if available - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose the appropriate bar graph type, label axes, set scale, and add a title. Encourage them to focus on creating a clear and visually effective bar graph for comparison.  
Groups share their constructed bar graphs with the class. Class feedback and discussion – peer review of bar graph design and effectiveness in communicating comparisons.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why are bar graphs effective for visualizing and comparing categorical data?  
What are the key elements of a well-designed bar graph for scientific communication?  
When is it most appropriate to use a bar graph compared to tables or other graph types? What types of questions are best answered with bar graphs?  
  
Assessments:  
  
Formative Task (5 mins): "Bar Graph Component Identification" - Provide a bar graph. Ask students to label and identify key components (axes, bars, labels, title, scale).  
Review of group-constructed bar graphs – assess clarity, accuracy, and appropriate choice of graph type and design.  
  
Student Workbook:  
  
Diagram of a well-designed bar graph with labels for key components and descriptions.  
Checklist of criteria for good bar graph design.  
Data sets for bar graph construction exercises.  
"Bar Graph Type Match" worksheet with scenarios and prompts for choosing appropriate bar graph types.  
  
Extension Activities:  
  
Students find bar graphs in scientific publications or news articles and analyse their design and effectiveness in communicating data. Critique their strengths and weaknesses.  
Challenge: Explore variations of bar graphs (e.g., stacked bar graphs, 100% stacked bar graphs) and investigate when these types are most useful. Create examples of these more complex bar graph types using digital tools.

- Warm-Up (10 mins): "Categorical Data Brainstorm" - Ask students to brainstorm examples of categorical data in science and everyday life (e.g., types of animals, colours of flowers, types of rocks, favourite subjects, survey responses with categories). List these on the board. Discuss: What kind of questions can we answer with categorical data? How can we visually compare categories?

- Main Activity 1: "Anatomy of a Bar Graph - Building Blocks of Comparison" (25 mins):  
  
Introduce the key components of a bar graph:  
  
Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).  
Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.  
Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.  
Title: Clear, concise title describing what the bar graph represents.  
Labels: Category labels along the x-axis, and y-axis labels with units.  
  
Show examples of well-designed and poorly designed bar graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of data, and effective visual comparison of categories.  
Introduce different types of bar graphs:  
  
Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.  
Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.  
Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.  
  
"Bar Graph Type Match": Present scenarios or research questions with categorical data. Students identify which bar graph type (vertical, horizontal, grouped) would be most appropriate and justify their choice.

- Introduce the key components of a bar graph:  
  
Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).  
Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.  
Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.  
Title: Clear, concise title describing what the bar graph represents.  
Labels: Category labels along the x-axis, and y-axis labels with units.

- Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).

- Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.

- Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.

- Title: Clear, concise title describing what the bar graph represents.

- Labels: Category labels along the x-axis, and y-axis labels with units.

- Show examples of well-designed and poorly designed bar graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of data, and effective visual comparison of categories.

- Introduce different types of bar graphs:  
  
Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.  
Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.  
Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.

- Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.

- Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.

- Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.

- "Bar Graph Type Match": Present scenarios or research questions with categorical data. Students identify which bar graph type (vertical, horizontal, grouped) would be most appropriate and justify their choice.

- Main Activity 2: "Bar Graph Construction Challenge - From Data to Visual Comparison" (20 mins):  
  
Provide students with datasets suitable for bar graphs (different datasets for different groups - e.g., species counts in different habitats, survey results on favourite science topics, experimental results comparing different treatments). Data can be in tables or lists.  
Each group task: Construct a bar graph (using paper and pencil initially, or digital tools if available - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose the appropriate bar graph type, label axes, set scale, and add a title. Encourage them to focus on creating a clear and visually effective bar graph for comparison.  
Groups share their constructed bar graphs with the class. Class feedback and discussion – peer review of bar graph design and effectiveness in communicating comparisons.

- Provide students with datasets suitable for bar graphs (different datasets for different groups - e.g., species counts in different habitats, survey results on favourite science topics, experimental results comparing different treatments). Data can be in tables or lists.

- Each group task: Construct a bar graph (using paper and pencil initially, or digital tools if available - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose the appropriate bar graph type, label axes, set scale, and add a title. Encourage them to focus on creating a clear and visually effective bar graph for comparison.

- Groups share their constructed bar graphs with the class. Class feedback and discussion – peer review of bar graph design and effectiveness in communicating comparisons.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are bar graphs effective for visualizing and comparing categorical data?  
What are the key elements of a well-designed bar graph for scientific communication?  
When is it most appropriate to use a bar graph compared to tables or other graph types? What types of questions are best answered with bar graphs?

- Why are bar graphs effective for visualizing and comparing categorical data?

- What are the key elements of a well-designed bar graph for scientific communication?

- When is it most appropriate to use a bar graph compared to tables or other graph types? What types of questions are best answered with bar graphs?

- Assessments:  
  
Formative Task (5 mins): "Bar Graph Component Identification" - Provide a bar graph. Ask students to label and identify key components (axes, bars, labels, title, scale).  
Review of group-constructed bar graphs – assess clarity, accuracy, and appropriate choice of graph type and design.

- Formative Task (5 mins): "Bar Graph Component Identification" - Provide a bar graph. Ask students to label and identify key components (axes, bars, labels, title, scale).

- Review of group-constructed bar graphs – assess clarity, accuracy, and appropriate choice of graph type and design.

- Student Workbook:  
  
Diagram of a well-designed bar graph with labels for key components and descriptions.  
Checklist of criteria for good bar graph design.  
Data sets for bar graph construction exercises.  
"Bar Graph Type Match" worksheet with scenarios and prompts for choosing appropriate bar graph types.

- Diagram of a well-designed bar graph with labels for key components and descriptions.

- Checklist of criteria for good bar graph design.

- Data sets for bar graph construction exercises.

- "Bar Graph Type Match" worksheet with scenarios and prompts for choosing appropriate bar graph types.

- Extension Activities:  
  
Students find bar graphs in scientific publications or news articles and analyse their design and effectiveness in communicating data. Critique their strengths and weaknesses.  
Challenge: Explore variations of bar graphs (e.g., stacked bar graphs, 100% stacked bar graphs) and investigate when these types are most useful. Create examples of these more complex bar graph types using digital tools.

- Students find bar graphs in scientific publications or news articles and analyse their design and effectiveness in communicating data. Critique their strengths and weaknesses.

- Challenge: Explore variations of bar graphs (e.g., stacked bar graphs, 100% stacked bar graphs) and investigate when these types are most useful. Create examples of these more complex bar graph types using digital tools.

- Lesson Materials:  
  
Slide Deck: Introduction to bar graphs, components of a bar graph, types of bar graphs (vertical, horizontal, grouped), examples of good and bad bar graphs, bar graph design principles.  
Handout: Data sets for bar graph construction exercises, "Bar Graph Type Match" worksheet with scenarios, checklist for good bar graph design.  
Formative Task (printable or digital).  
Graph paper, rulers, coloured pencils (for paper-based construction) OR access to spreadsheet software, Observable, or online graph makers (for digital construction).

- Slide Deck: Introduction to bar graphs, components of a bar graph, types of bar graphs (vertical, horizontal, grouped), examples of good and bad bar graphs, bar graph design principles.

- Handout: Data sets for bar graph construction exercises, "Bar Graph Type Match" worksheet with scenarios, checklist for good bar graph design.

- Formative Task (printable or digital).

- Graph paper, rulers, coloured pencils (for paper-based construction) OR access to spreadsheet software, Observable, or online graph makers (for digital construction).

- Students will be able to construct bar graphs to visually represent categorical data and compare quantities across different categories.

- Students will be able to identify the key components of a bar graph (axes, bars, labels, title, scale).

- Students will be able to choose appropriate bar graph types (vertical, horizontal, grouped) for different datasets and analytical goals.

- Students will be able to interpret bar graphs and extract meaningful comparisons and conclusions from them.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Warm-Up (10 mins): "Categorical Data Brainstorm" - Ask students to brainstorm examples of categorical data in science and everyday life (e.g., types of animals, colours of flowers, types of rocks, favourite subjects, survey responses with categories). List these on the board. Discuss: What kind of questions can we answer with categorical data? How can we visually compare categories?

- Main Activity 1: "Anatomy of a Bar Graph - Building Blocks of Comparison" (25 mins):  
  
Introduce the key components of a bar graph:  
  
Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).  
Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.  
Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.  
Title: Clear, concise title describing what the bar graph represents.  
Labels: Category labels along the x-axis, and y-axis labels with units.  
  
Show examples of well-designed and poorly designed bar graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of data, and effective visual comparison of categories.  
Introduce different types of bar graphs:  
  
Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.  
Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.  
Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.  
  
"Bar Graph Type Match": Present scenarios or research questions with categorical data. Students identify which bar graph type (vertical, horizontal, grouped) would be most appropriate and justify their choice.

- Introduce the key components of a bar graph:  
  
Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).  
Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.  
Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.  
Title: Clear, concise title describing what the bar graph represents.  
Labels: Category labels along the x-axis, and y-axis labels with units.

- Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).

- Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.

- Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.

- Title: Clear, concise title describing what the bar graph represents.

- Labels: Category labels along the x-axis, and y-axis labels with units.

- Show examples of well-designed and poorly designed bar graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of data, and effective visual comparison of categories.

- Introduce different types of bar graphs:  
  
Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.  
Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.  
Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.

- Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.

- Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.

- Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.

- "Bar Graph Type Match": Present scenarios or research questions with categorical data. Students identify which bar graph type (vertical, horizontal, grouped) would be most appropriate and justify their choice.

- Main Activity 2: "Bar Graph Construction Challenge - From Data to Visual Comparison" (20 mins):  
  
Provide students with datasets suitable for bar graphs (different datasets for different groups - e.g., species counts in different habitats, survey results on favourite science topics, experimental results comparing different treatments). Data can be in tables or lists.  
Each group task: Construct a bar graph (using paper and pencil initially, or digital tools if available - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose the appropriate bar graph type, label axes, set scale, and add a title. Encourage them to focus on creating a clear and visually effective bar graph for comparison.  
Groups share their constructed bar graphs with the class. Class feedback and discussion – peer review of bar graph design and effectiveness in communicating comparisons.

- Provide students with datasets suitable for bar graphs (different datasets for different groups - e.g., species counts in different habitats, survey results on favourite science topics, experimental results comparing different treatments). Data can be in tables or lists.

- Each group task: Construct a bar graph (using paper and pencil initially, or digital tools if available - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose the appropriate bar graph type, label axes, set scale, and add a title. Encourage them to focus on creating a clear and visually effective bar graph for comparison.

- Groups share their constructed bar graphs with the class. Class feedback and discussion – peer review of bar graph design and effectiveness in communicating comparisons.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are bar graphs effective for visualizing and comparing categorical data?  
What are the key elements of a well-designed bar graph for scientific communication?  
When is it most appropriate to use a bar graph compared to tables or other graph types? What types of questions are best answered with bar graphs?

- Why are bar graphs effective for visualizing and comparing categorical data?

- What are the key elements of a well-designed bar graph for scientific communication?

- When is it most appropriate to use a bar graph compared to tables or other graph types? What types of questions are best answered with bar graphs?

- Assessments:  
  
Formative Task (5 mins): "Bar Graph Component Identification" - Provide a bar graph. Ask students to label and identify key components (axes, bars, labels, title, scale).  
Review of group-constructed bar graphs – assess clarity, accuracy, and appropriate choice of graph type and design.

- Formative Task (5 mins): "Bar Graph Component Identification" - Provide a bar graph. Ask students to label and identify key components (axes, bars, labels, title, scale).

- Review of group-constructed bar graphs – assess clarity, accuracy, and appropriate choice of graph type and design.

- Student Workbook:  
  
Diagram of a well-designed bar graph with labels for key components and descriptions.  
Checklist of criteria for good bar graph design.  
Data sets for bar graph construction exercises.  
"Bar Graph Type Match" worksheet with scenarios and prompts for choosing appropriate bar graph types.

- Diagram of a well-designed bar graph with labels for key components and descriptions.

- Checklist of criteria for good bar graph design.

- Data sets for bar graph construction exercises.

- "Bar Graph Type Match" worksheet with scenarios and prompts for choosing appropriate bar graph types.

- Extension Activities:  
  
Students find bar graphs in scientific publications or news articles and analyse their design and effectiveness in communicating data. Critique their strengths and weaknesses.  
Challenge: Explore variations of bar graphs (e.g., stacked bar graphs, 100% stacked bar graphs) and investigate when these types are most useful. Create examples of these more complex bar graph types using digital tools.

- Students find bar graphs in scientific publications or news articles and analyse their design and effectiveness in communicating data. Critique their strengths and weaknesses.

- Challenge: Explore variations of bar graphs (e.g., stacked bar graphs, 100% stacked bar graphs) and investigate when these types are most useful. Create examples of these more complex bar graph types using digital tools.

- Introduce the key components of a bar graph:  
  
Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).  
Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.  
Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.  
Title: Clear, concise title describing what the bar graph represents.  
Labels: Category labels along the x-axis, and y-axis labels with units.

- Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).

- Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.

- Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.

- Title: Clear, concise title describing what the bar graph represents.

- Labels: Category labels along the x-axis, and y-axis labels with units.

- Show examples of well-designed and poorly designed bar graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of data, and effective visual comparison of categories.

- Introduce different types of bar graphs:  
  
Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.  
Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.  
Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.

- Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.

- Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.

- Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.

- "Bar Graph Type Match": Present scenarios or research questions with categorical data. Students identify which bar graph type (vertical, horizontal, grouped) would be most appropriate and justify their choice.

- Axes: Horizontal (x-axis) for categories, Vertical (y-axis) for frequency or quantity. Label axes clearly (category name, quantity name, units if applicable).

- Bars: Rectangular bars representing each category. Bar height or length corresponds to the frequency or quantity for that category. Bars should be visually distinct and have appropriate spacing.

- Scale: Appropriate scale on the y-axis to accommodate the data range, with clear tick marks and labels.

- Title: Clear, concise title describing what the bar graph represents.

- Labels: Category labels along the x-axis, and y-axis labels with units.

- Vertical Bar Graph: Bars extend vertically from the x-axis. Common for comparing categories along the x-axis.

- Horizontal Bar Graph: Bars extend horizontally from the y-axis. Useful when category labels are long or when comparing many categories.

- Grouped Bar Graph (Clustered Bar Graph): Used to compare multiple categories for different groups or conditions. Bars for each group are clustered together for each category.

- Provide students with datasets suitable for bar graphs (different datasets for different groups - e.g., species counts in different habitats, survey results on favourite science topics, experimental results comparing different treatments). Data can be in tables or lists.

- Each group task: Construct a bar graph (using paper and pencil initially, or digital tools if available - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose the appropriate bar graph type, label axes, set scale, and add a title. Encourage them to focus on creating a clear and visually effective bar graph for comparison.

- Groups share their constructed bar graphs with the class. Class feedback and discussion – peer review of bar graph design and effectiveness in communicating comparisons.

- Why are bar graphs effective for visualizing and comparing categorical data?

- What are the key elements of a well-designed bar graph for scientific communication?

- When is it most appropriate to use a bar graph compared to tables or other graph types? What types of questions are best answered with bar graphs?

- Formative Task (5 mins): "Bar Graph Component Identification" - Provide a bar graph. Ask students to label and identify key components (axes, bars, labels, title, scale).

- Review of group-constructed bar graphs – assess clarity, accuracy, and appropriate choice of graph type and design.

- Diagram of a well-designed bar graph with labels for key components and descriptions.

- Checklist of criteria for good bar graph design.

- Data sets for bar graph construction exercises.

- "Bar Graph Type Match" worksheet with scenarios and prompts for choosing appropriate bar graph types.

- Students find bar graphs in scientific publications or news articles and analyse their design and effectiveness in communicating data. Critique their strengths and weaknesses.

- Challenge: Explore variations of bar graphs (e.g., stacked bar graphs, 100% stacked bar graphs) and investigate when these types are most useful. Create examples of these more complex bar graph types using digital tools.

- Slide Deck: Introduction to bar graphs, components of a bar graph, types of bar graphs (vertical, horizontal, grouped), examples of good and bad bar graphs, bar graph design principles.

- Handout: Data sets for bar graph construction exercises, "Bar Graph Type Match" worksheet with scenarios, checklist for good bar graph design.

- Formative Task (printable or digital).

- Graph paper, rulers, coloured pencils (for paper-based construction) OR access to spreadsheet software, Observable, or online graph makers (for digital construction).

### Lesson 3.3: Showing Trends Over Time: Line Graphs

- Title: Line Graphs: Revealing Trends and Changes Over Time

- Learning Outcomes:  
  
Students will be able to construct line graphs to visually represent continuous data and show trends or changes over time or a continuous variable.  
Students will be able to identify the key components of a line graph (axes, data points, lines, labels, title, scale).  
Students will be able to choose appropriate scales and line styles for effective line graph construction.  
Students will be able to interpret line graphs to identify trends, patterns, and relationships in data over time or a continuous variable.  
NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Students will be able to construct line graphs to visually represent continuous data and show trends or changes over time or a continuous variable.

- Students will be able to identify the key components of a line graph (axes, data points, lines, labels, title, scale).

- Students will be able to choose appropriate scales and line styles for effective line graph construction.

- Students will be able to interpret line graphs to identify trends, patterns, and relationships in data over time or a continuous variable.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Overview: This lesson introduces line graphs as the primary tool for visualizing trends and changes in continuous data, particularly over time. Students will learn the components of line graphs, practice constructing them, and develop skills in interpreting line graphs to identify trends, patterns, and relationships in scientific data.

- Activities:  
  
Warm-Up (10 mins): "Trend Spotting Challenge" - Show students several simple line graphs depicting trends (e.g., temperature changes over a day, population growth over years, reaction rate changing with temperature). For each graph, ask: "What trend or pattern do you see in this graph? Is it increasing, decreasing, fluctuating, stable? What does this trend tell you about the data?" Focus on quickly identifying visual trends.  
Main Activity 1: "Anatomy of a Line Graph - Connecting the Dots of Change" (25 mins):  
  
Introduce the key components of a line graph:  
  
Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).  
Data Points: Plotted points representing data values for each time point or x-value.  
Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.  
Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.  
Title: Clear, concise title describing what the line graph represents.  
Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.  
  
Show examples of well-designed and poorly designed line graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of trends, and effective visual communication of change over time.  
Discuss choosing appropriate scales for line graphs: linear vs. logarithmic (briefly introduce log scale concept for future, focus on linear for Year 8). Discuss how scale choice can affect the visual appearance of trends.  
Discuss line styles: solid lines, dashed lines, dotted lines for distinguishing multiple datasets.  
  
Main Activity 2: "Line Graph Construction Challenge - Revealing Time-Based Trends" (20 mins):  
  
Provide students with datasets suitable for line graphs (different datasets for different groups - e.g., temperature readings over time, plant growth measurements over weeks, reaction rate data at different temperatures, population data over years). Data can be in tables or lists.  
Each group task: Construct a line graph (using paper and pencil initially, or digital tools - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose appropriate axes, scales, plot data points, connect lines, label axes, and add a title. Encourage them to focus on creating a clear line graph that effectively shows trends and changes over time.  
Groups share their constructed line graphs with the class. Class feedback and discussion – peer review of line graph design and effectiveness in communicating trends. Discuss different scale choices and their impact on visual interpretation.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why are line graphs the best choice for showing trends and changes over time or a continuous variable?  
What are the key elements of a well-designed line graph for scientific communication?  
When is it most appropriate to use a line graph compared to tables or bar graphs? What types of scientific questions are best answered with line graphs?  
How can misleading scales or poor design distort the trends shown in a line graph?  
  
Assessments:  
  
Formative Task (5 mins): "Line Graph Component Identification" - Provide a line graph. Ask students to label and identify key components (axes, data points, lines, labels, title, scale).  
Review of group-constructed line graphs – assess clarity, accuracy, appropriate scale choice, and effectiveness in showing trends.  
  
Student Workbook:  
  
Diagram of a well-designed line graph with labels for key components and descriptions.  
Checklist of criteria for good line graph design.  
Data sets for line graph construction exercises.  
"Scale Choice Challenge" - scenarios where students choose appropriate scales and explain their reasoning.  
  
Extension Activities:  
  
Students find line graphs in scientific publications or reports and analyse their design and interpretation. Critique their effectiveness in communicating trends and relationships.  
Challenge: Explore more advanced line graph techniques (e.g., multiple lines on one graph with legend, shaded areas under lines to represent uncertainty or ranges). Create examples of these more complex line graph types using digital tools. Investigate the concept of trendlines or best-fit lines (briefly introduced conceptually).

- Warm-Up (10 mins): "Trend Spotting Challenge" - Show students several simple line graphs depicting trends (e.g., temperature changes over a day, population growth over years, reaction rate changing with temperature). For each graph, ask: "What trend or pattern do you see in this graph? Is it increasing, decreasing, fluctuating, stable? What does this trend tell you about the data?" Focus on quickly identifying visual trends.

- Main Activity 1: "Anatomy of a Line Graph - Connecting the Dots of Change" (25 mins):  
  
Introduce the key components of a line graph:  
  
Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).  
Data Points: Plotted points representing data values for each time point or x-value.  
Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.  
Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.  
Title: Clear, concise title describing what the line graph represents.  
Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.  
  
Show examples of well-designed and poorly designed line graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of trends, and effective visual communication of change over time.  
Discuss choosing appropriate scales for line graphs: linear vs. logarithmic (briefly introduce log scale concept for future, focus on linear for Year 8). Discuss how scale choice can affect the visual appearance of trends.  
Discuss line styles: solid lines, dashed lines, dotted lines for distinguishing multiple datasets.

- Introduce the key components of a line graph:  
  
Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).  
Data Points: Plotted points representing data values for each time point or x-value.  
Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.  
Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.  
Title: Clear, concise title describing what the line graph represents.  
Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.

- Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).

- Data Points: Plotted points representing data values for each time point or x-value.

- Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.

- Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.

- Title: Clear, concise title describing what the line graph represents.

- Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.

- Show examples of well-designed and poorly designed line graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of trends, and effective visual communication of change over time.

- Discuss choosing appropriate scales for line graphs: linear vs. logarithmic (briefly introduce log scale concept for future, focus on linear for Year 8). Discuss how scale choice can affect the visual appearance of trends.

- Discuss line styles: solid lines, dashed lines, dotted lines for distinguishing multiple datasets.

- Main Activity 2: "Line Graph Construction Challenge - Revealing Time-Based Trends" (20 mins):  
  
Provide students with datasets suitable for line graphs (different datasets for different groups - e.g., temperature readings over time, plant growth measurements over weeks, reaction rate data at different temperatures, population data over years). Data can be in tables or lists.  
Each group task: Construct a line graph (using paper and pencil initially, or digital tools - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose appropriate axes, scales, plot data points, connect lines, label axes, and add a title. Encourage them to focus on creating a clear line graph that effectively shows trends and changes over time.  
Groups share their constructed line graphs with the class. Class feedback and discussion – peer review of line graph design and effectiveness in communicating trends. Discuss different scale choices and their impact on visual interpretation.

- Provide students with datasets suitable for line graphs (different datasets for different groups - e.g., temperature readings over time, plant growth measurements over weeks, reaction rate data at different temperatures, population data over years). Data can be in tables or lists.

- Each group task: Construct a line graph (using paper and pencil initially, or digital tools - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose appropriate axes, scales, plot data points, connect lines, label axes, and add a title. Encourage them to focus on creating a clear line graph that effectively shows trends and changes over time.

- Groups share their constructed line graphs with the class. Class feedback and discussion – peer review of line graph design and effectiveness in communicating trends. Discuss different scale choices and their impact on visual interpretation.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are line graphs the best choice for showing trends and changes over time or a continuous variable?  
What are the key elements of a well-designed line graph for scientific communication?  
When is it most appropriate to use a line graph compared to tables or bar graphs? What types of scientific questions are best answered with line graphs?  
How can misleading scales or poor design distort the trends shown in a line graph?

- Why are line graphs the best choice for showing trends and changes over time or a continuous variable?

- What are the key elements of a well-designed line graph for scientific communication?

- When is it most appropriate to use a line graph compared to tables or bar graphs? What types of scientific questions are best answered with line graphs?

- How can misleading scales or poor design distort the trends shown in a line graph?

- Assessments:  
  
Formative Task (5 mins): "Line Graph Component Identification" - Provide a line graph. Ask students to label and identify key components (axes, data points, lines, labels, title, scale).  
Review of group-constructed line graphs – assess clarity, accuracy, appropriate scale choice, and effectiveness in showing trends.

- Formative Task (5 mins): "Line Graph Component Identification" - Provide a line graph. Ask students to label and identify key components (axes, data points, lines, labels, title, scale).

- Review of group-constructed line graphs – assess clarity, accuracy, appropriate scale choice, and effectiveness in showing trends.

- Student Workbook:  
  
Diagram of a well-designed line graph with labels for key components and descriptions.  
Checklist of criteria for good line graph design.  
Data sets for line graph construction exercises.  
"Scale Choice Challenge" - scenarios where students choose appropriate scales and explain their reasoning.

- Diagram of a well-designed line graph with labels for key components and descriptions.

- Checklist of criteria for good line graph design.

- Data sets for line graph construction exercises.

- "Scale Choice Challenge" - scenarios where students choose appropriate scales and explain their reasoning.

- Extension Activities:  
  
Students find line graphs in scientific publications or reports and analyse their design and interpretation. Critique their effectiveness in communicating trends and relationships.  
Challenge: Explore more advanced line graph techniques (e.g., multiple lines on one graph with legend, shaded areas under lines to represent uncertainty or ranges). Create examples of these more complex line graph types using digital tools. Investigate the concept of trendlines or best-fit lines (briefly introduced conceptually).

- Students find line graphs in scientific publications or reports and analyse their design and interpretation. Critique their effectiveness in communicating trends and relationships.

- Challenge: Explore more advanced line graph techniques (e.g., multiple lines on one graph with legend, shaded areas under lines to represent uncertainty or ranges). Create examples of these more complex line graph types using digital tools. Investigate the concept of trendlines or best-fit lines (briefly introduced conceptually).

- Lesson Materials:  
  
Slide Deck: Introduction to line graphs, components of a line graph, examples of good and bad line graphs, line graph design principles, scale choice considerations.  
Handout: Data sets for line graph construction exercises, "Scale Choice Challenge" scenarios worksheet, checklist for good line graph design.  
Formative Task (printable or digital).  
Graph paper, rulers, coloured pencils (for paper-based construction) OR access to spreadsheet software, Observable, or online graph makers (for digital construction).

- Slide Deck: Introduction to line graphs, components of a line graph, examples of good and bad line graphs, line graph design principles, scale choice considerations.

- Handout: Data sets for line graph construction exercises, "Scale Choice Challenge" scenarios worksheet, checklist for good line graph design.

- Formative Task (printable or digital).

- Graph paper, rulers, coloured pencils (for paper-based construction) OR access to spreadsheet software, Observable, or online graph makers (for digital construction).

- Students will be able to construct line graphs to visually represent continuous data and show trends or changes over time or a continuous variable.

- Students will be able to identify the key components of a line graph (axes, data points, lines, labels, title, scale).

- Students will be able to choose appropriate scales and line styles for effective line graph construction.

- Students will be able to interpret line graphs to identify trends, patterns, and relationships in data over time or a continuous variable.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Warm-Up (10 mins): "Trend Spotting Challenge" - Show students several simple line graphs depicting trends (e.g., temperature changes over a day, population growth over years, reaction rate changing with temperature). For each graph, ask: "What trend or pattern do you see in this graph? Is it increasing, decreasing, fluctuating, stable? What does this trend tell you about the data?" Focus on quickly identifying visual trends.

- Main Activity 1: "Anatomy of a Line Graph - Connecting the Dots of Change" (25 mins):  
  
Introduce the key components of a line graph:  
  
Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).  
Data Points: Plotted points representing data values for each time point or x-value.  
Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.  
Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.  
Title: Clear, concise title describing what the line graph represents.  
Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.  
  
Show examples of well-designed and poorly designed line graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of trends, and effective visual communication of change over time.  
Discuss choosing appropriate scales for line graphs: linear vs. logarithmic (briefly introduce log scale concept for future, focus on linear for Year 8). Discuss how scale choice can affect the visual appearance of trends.  
Discuss line styles: solid lines, dashed lines, dotted lines for distinguishing multiple datasets.

- Introduce the key components of a line graph:  
  
Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).  
Data Points: Plotted points representing data values for each time point or x-value.  
Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.  
Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.  
Title: Clear, concise title describing what the line graph represents.  
Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.

- Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).

- Data Points: Plotted points representing data values for each time point or x-value.

- Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.

- Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.

- Title: Clear, concise title describing what the line graph represents.

- Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.

- Show examples of well-designed and poorly designed line graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of trends, and effective visual communication of change over time.

- Discuss choosing appropriate scales for line graphs: linear vs. logarithmic (briefly introduce log scale concept for future, focus on linear for Year 8). Discuss how scale choice can affect the visual appearance of trends.

- Discuss line styles: solid lines, dashed lines, dotted lines for distinguishing multiple datasets.

- Main Activity 2: "Line Graph Construction Challenge - Revealing Time-Based Trends" (20 mins):  
  
Provide students with datasets suitable for line graphs (different datasets for different groups - e.g., temperature readings over time, plant growth measurements over weeks, reaction rate data at different temperatures, population data over years). Data can be in tables or lists.  
Each group task: Construct a line graph (using paper and pencil initially, or digital tools - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose appropriate axes, scales, plot data points, connect lines, label axes, and add a title. Encourage them to focus on creating a clear line graph that effectively shows trends and changes over time.  
Groups share their constructed line graphs with the class. Class feedback and discussion – peer review of line graph design and effectiveness in communicating trends. Discuss different scale choices and their impact on visual interpretation.

- Provide students with datasets suitable for line graphs (different datasets for different groups - e.g., temperature readings over time, plant growth measurements over weeks, reaction rate data at different temperatures, population data over years). Data can be in tables or lists.

- Each group task: Construct a line graph (using paper and pencil initially, or digital tools - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose appropriate axes, scales, plot data points, connect lines, label axes, and add a title. Encourage them to focus on creating a clear line graph that effectively shows trends and changes over time.

- Groups share their constructed line graphs with the class. Class feedback and discussion – peer review of line graph design and effectiveness in communicating trends. Discuss different scale choices and their impact on visual interpretation.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are line graphs the best choice for showing trends and changes over time or a continuous variable?  
What are the key elements of a well-designed line graph for scientific communication?  
When is it most appropriate to use a line graph compared to tables or bar graphs? What types of scientific questions are best answered with line graphs?  
How can misleading scales or poor design distort the trends shown in a line graph?

- Why are line graphs the best choice for showing trends and changes over time or a continuous variable?

- What are the key elements of a well-designed line graph for scientific communication?

- When is it most appropriate to use a line graph compared to tables or bar graphs? What types of scientific questions are best answered with line graphs?

- How can misleading scales or poor design distort the trends shown in a line graph?

- Assessments:  
  
Formative Task (5 mins): "Line Graph Component Identification" - Provide a line graph. Ask students to label and identify key components (axes, data points, lines, labels, title, scale).  
Review of group-constructed line graphs – assess clarity, accuracy, appropriate scale choice, and effectiveness in showing trends.

- Formative Task (5 mins): "Line Graph Component Identification" - Provide a line graph. Ask students to label and identify key components (axes, data points, lines, labels, title, scale).

- Review of group-constructed line graphs – assess clarity, accuracy, appropriate scale choice, and effectiveness in showing trends.

- Student Workbook:  
  
Diagram of a well-designed line graph with labels for key components and descriptions.  
Checklist of criteria for good line graph design.  
Data sets for line graph construction exercises.  
"Scale Choice Challenge" - scenarios where students choose appropriate scales and explain their reasoning.

- Diagram of a well-designed line graph with labels for key components and descriptions.

- Checklist of criteria for good line graph design.

- Data sets for line graph construction exercises.

- "Scale Choice Challenge" - scenarios where students choose appropriate scales and explain their reasoning.

- Extension Activities:  
  
Students find line graphs in scientific publications or reports and analyse their design and interpretation. Critique their effectiveness in communicating trends and relationships.  
Challenge: Explore more advanced line graph techniques (e.g., multiple lines on one graph with legend, shaded areas under lines to represent uncertainty or ranges). Create examples of these more complex line graph types using digital tools. Investigate the concept of trendlines or best-fit lines (briefly introduced conceptually).

- Students find line graphs in scientific publications or reports and analyse their design and interpretation. Critique their effectiveness in communicating trends and relationships.

- Challenge: Explore more advanced line graph techniques (e.g., multiple lines on one graph with legend, shaded areas under lines to represent uncertainty or ranges). Create examples of these more complex line graph types using digital tools. Investigate the concept of trendlines or best-fit lines (briefly introduced conceptually).

- Introduce the key components of a line graph:  
  
Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).  
Data Points: Plotted points representing data values for each time point or x-value.  
Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.  
Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.  
Title: Clear, concise title describing what the line graph represents.  
Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.

- Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).

- Data Points: Plotted points representing data values for each time point or x-value.

- Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.

- Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.

- Title: Clear, concise title describing what the line graph represents.

- Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.

- Show examples of well-designed and poorly designed line graphs. For poorly designed examples, ask students to identify weaknesses and suggest improvements. Focus on clarity, accurate representation of trends, and effective visual communication of change over time.

- Discuss choosing appropriate scales for line graphs: linear vs. logarithmic (briefly introduce log scale concept for future, focus on linear for Year 8). Discuss how scale choice can affect the visual appearance of trends.

- Discuss line styles: solid lines, dashed lines, dotted lines for distinguishing multiple datasets.

- Axes: Horizontal (x-axis) usually for time or a continuous independent variable, Vertical (y-axis) for the dependent variable being measured. Label axes clearly (variable names, units).

- Data Points: Plotted points representing data values for each time point or x-value.

- Lines: Lines connecting consecutive data points to show trends. Use straight lines between points. For multiple datasets on the same graph, use different line styles or colours to distinguish them.

- Scale: Appropriate scales for both axes to accommodate data ranges, with clear tick marks and labels. Choose scales that spread out the data effectively and make trends visible.

- Title: Clear, concise title describing what the line graph represents.

- Labels: Axis labels with variable names and units, and potentially a legend if multiple lines are plotted.

- Provide students with datasets suitable for line graphs (different datasets for different groups - e.g., temperature readings over time, plant growth measurements over weeks, reaction rate data at different temperatures, population data over years). Data can be in tables or lists.

- Each group task: Construct a line graph (using paper and pencil initially, or digital tools - spreadsheet software, Observable) to visualize their assigned dataset. They need to choose appropriate axes, scales, plot data points, connect lines, label axes, and add a title. Encourage them to focus on creating a clear line graph that effectively shows trends and changes over time.

- Groups share their constructed line graphs with the class. Class feedback and discussion – peer review of line graph design and effectiveness in communicating trends. Discuss different scale choices and their impact on visual interpretation.

- Why are line graphs the best choice for showing trends and changes over time or a continuous variable?

- What are the key elements of a well-designed line graph for scientific communication?

- When is it most appropriate to use a line graph compared to tables or bar graphs? What types of scientific questions are best answered with line graphs?

- How can misleading scales or poor design distort the trends shown in a line graph?

- Formative Task (5 mins): "Line Graph Component Identification" - Provide a line graph. Ask students to label and identify key components (axes, data points, lines, labels, title, scale).

- Review of group-constructed line graphs – assess clarity, accuracy, appropriate scale choice, and effectiveness in showing trends.

- Diagram of a well-designed line graph with labels for key components and descriptions.

- Checklist of criteria for good line graph design.

- Data sets for line graph construction exercises.

- "Scale Choice Challenge" - scenarios where students choose appropriate scales and explain their reasoning.

- Students find line graphs in scientific publications or reports and analyse their design and interpretation. Critique their effectiveness in communicating trends and relationships.

- Challenge: Explore more advanced line graph techniques (e.g., multiple lines on one graph with legend, shaded areas under lines to represent uncertainty or ranges). Create examples of these more complex line graph types using digital tools. Investigate the concept of trendlines or best-fit lines (briefly introduced conceptually).

- Slide Deck: Introduction to line graphs, components of a line graph, examples of good and bad line graphs, line graph design principles, scale choice considerations.

- Handout: Data sets for line graph construction exercises, "Scale Choice Challenge" scenarios worksheet, checklist for good line graph design.

- Formative Task (printable or digital).

- Graph paper, rulers, coloured pencils (for paper-based construction) OR access to spreadsheet software, Observable, or online graph makers (for digital construction).

### Week 4: Descriptive Statistics & Data Interpretation

### Lesson 4.1: Central Tendency: Mean, Median, Mode - Finding the "Average"

- Title: Unlocking the Center: Mean, Median, and Mode – Finding the Typical Value

- Learning Outcomes:  
  
Students will be able to define and calculate the mean, median, and mode for a given dataset.  
Students will be able to explain the differences between mean, median, and mode and identify situations where each measure is most appropriate.  
Students will be able to calculate mean, median, and mode using Python in Observable notebooks.  
Students will be able to interpret the mean, median, and mode in the context of scientific data and draw basic conclusions about central tendency.  
NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Students will be able to define and calculate the mean, median, and mode for a given dataset.

- Students will be able to explain the differences between mean, median, and mode and identify situations where each measure is most appropriate.

- Students will be able to calculate mean, median, and mode using Python in Observable notebooks.

- Students will be able to interpret the mean, median, and mode in the context of scientific data and draw basic conclusions about central tendency.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Overview: This lesson introduces the concept of central tendency and the three key measures: mean, median, and mode. Students will learn to calculate each measure, understand their differences, and apply them to scientific data, using Python in Observable for calculations.

- Activities:  
  
Warm-Up (10 mins): "Guess the Average" - Present a few sets of simple numbers (e.g., ages of students in a small group, number of petals on different flowers). Ask students to quickly estimate the "average" value for each set. Discuss: What does "average" mean in everyday language? Are there different ways to think about "average"?  
Main Activity 1: "Mean, Median, Mode Definitions and Calculations" (25 mins):  
  
Introduce and define Mean (arithmetic average), Median (middle value when data is ordered), and Mode (most frequent value).  
Step-by-step calculation examples for each measure using small, simple datasets (done by hand initially).  
  
Mean: Sum of values divided by the number of values.  
Median: Ordering data and finding the middle value (or average of two middle values for even datasets).  
Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).  
  
Worksheet practice: Provide students with several small datasets. Students calculate the mean, median, and mode for each dataset by hand. Check answers as a class.  
  
Main Activity 2: "Mean, Median, Mode in Python (Observable)" (20 mins):  
  
Introduce how to calculate mean, median, and mode using Python in Observable notebooks.  
Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).  
Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.  
Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).  
  
Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the mean, median, and mode.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are the similarities and differences between mean, median, and mode?  
When might the mean be the most appropriate measure of central tendency? When might the median be better? When might the mode be useful?  
How can understanding central tendency help us analyse and interpret scientific data? What kind of information does it give us?  
Are there any limitations to using just central tendency to describe a dataset? (Lead to the idea of data dispersion in the next lesson).  
  
Assessments:  
  
Formative Quiz (5 mins): Short multiple-choice quiz on definitions of mean, median, mode and their basic calculation. AI-generated quiz.  
Review of student Python code in Observable notebooks – check for correct calculations and code syntax.  
  
Student Workbook:  
  
Definitions of mean, median, and mode with formulas.  
Step-by-step calculation examples for each measure.  
Worksheet with datasets for manual calculation practice.  
Observable notebook exercises with Python code templates for calculating mean, median, and mode.  
"Choosing the Right Measure" scenario worksheet with prompts to decide when to use mean, median, or mode.  
  
Extension Activities:  
  
Students research real-world examples where mean, median, and mode are used in different scientific fields (e.g., average temperature, median income, modal species in an ecosystem).  
Challenge: Investigate the concept of weighted mean and when it is used. Calculate a weighted mean for a given dataset.

- Warm-Up (10 mins): "Guess the Average" - Present a few sets of simple numbers (e.g., ages of students in a small group, number of petals on different flowers). Ask students to quickly estimate the "average" value for each set. Discuss: What does "average" mean in everyday language? Are there different ways to think about "average"?

- Main Activity 1: "Mean, Median, Mode Definitions and Calculations" (25 mins):  
  
Introduce and define Mean (arithmetic average), Median (middle value when data is ordered), and Mode (most frequent value).  
Step-by-step calculation examples for each measure using small, simple datasets (done by hand initially).  
  
Mean: Sum of values divided by the number of values.  
Median: Ordering data and finding the middle value (or average of two middle values for even datasets).  
Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).  
  
Worksheet practice: Provide students with several small datasets. Students calculate the mean, median, and mode for each dataset by hand. Check answers as a class.

- Introduce and define Mean (arithmetic average), Median (middle value when data is ordered), and Mode (most frequent value).

- Step-by-step calculation examples for each measure using small, simple datasets (done by hand initially).  
  
Mean: Sum of values divided by the number of values.  
Median: Ordering data and finding the middle value (or average of two middle values for even datasets).  
Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).

- Mean: Sum of values divided by the number of values.

- Median: Ordering data and finding the middle value (or average of two middle values for even datasets).

- Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).

- Worksheet practice: Provide students with several small datasets. Students calculate the mean, median, and mode for each dataset by hand. Check answers as a class.

- Main Activity 2: "Mean, Median, Mode in Python (Observable)" (20 mins):  
  
Introduce how to calculate mean, median, and mode using Python in Observable notebooks.  
Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).  
Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.  
Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).  
  
Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the mean, median, and mode.

- Introduce how to calculate mean, median, and mode using Python in Observable notebooks.

- Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).  
Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.  
Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).

- Create a code cell.

- Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).

- Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.

- Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).

- Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the mean, median, and mode.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the similarities and differences between mean, median, and mode?  
When might the mean be the most appropriate measure of central tendency? When might the median be better? When might the mode be useful?  
How can understanding central tendency help us analyse and interpret scientific data? What kind of information does it give us?  
Are there any limitations to using just central tendency to describe a dataset? (Lead to the idea of data dispersion in the next lesson).

- What are the similarities and differences between mean, median, and mode?

- When might the mean be the most appropriate measure of central tendency? When might the median be better? When might the mode be useful?

- How can understanding central tendency help us analyse and interpret scientific data? What kind of information does it give us?

- Are there any limitations to using just central tendency to describe a dataset? (Lead to the idea of data dispersion in the next lesson).

- Assessments:  
  
Formative Quiz (5 mins): Short multiple-choice quiz on definitions of mean, median, mode and their basic calculation. AI-generated quiz.  
Review of student Python code in Observable notebooks – check for correct calculations and code syntax.

- Formative Quiz (5 mins): Short multiple-choice quiz on definitions of mean, median, mode and their basic calculation. AI-generated quiz.

- Review of student Python code in Observable notebooks – check for correct calculations and code syntax.

- Student Workbook:  
  
Definitions of mean, median, and mode with formulas.  
Step-by-step calculation examples for each measure.  
Worksheet with datasets for manual calculation practice.  
Observable notebook exercises with Python code templates for calculating mean, median, and mode.  
"Choosing the Right Measure" scenario worksheet with prompts to decide when to use mean, median, or mode.

- Definitions of mean, median, and mode with formulas.

- Step-by-step calculation examples for each measure.

- Worksheet with datasets for manual calculation practice.

- Observable notebook exercises with Python code templates for calculating mean, median, and mode.

- "Choosing the Right Measure" scenario worksheet with prompts to decide when to use mean, median, or mode.

- Extension Activities:  
  
Students research real-world examples where mean, median, and mode are used in different scientific fields (e.g., average temperature, median income, modal species in an ecosystem).  
Challenge: Investigate the concept of weighted mean and when it is used. Calculate a weighted mean for a given dataset.

- Students research real-world examples where mean, median, and mode are used in different scientific fields (e.g., average temperature, median income, modal species in an ecosystem).

- Challenge: Investigate the concept of weighted mean and when it is used. Calculate a weighted mean for a given dataset.

- Lesson Materials:  
  
Slide Deck: Introduction to central tendency, definitions of mean, median, mode, calculation examples, choosing the appropriate measure, Python code examples.  
Handout: Worksheet with datasets for manual calculation practice, "Choosing the Right Measure" scenario worksheet, Python code templates for Observable exercises.  
Formative Quiz (printable or digital).  
Computers/Tablets with internet access for Observable.

- Slide Deck: Introduction to central tendency, definitions of mean, median, mode, calculation examples, choosing the appropriate measure, Python code examples.

- Handout: Worksheet with datasets for manual calculation practice, "Choosing the Right Measure" scenario worksheet, Python code templates for Observable exercises.

- Formative Quiz (printable or digital).

- Computers/Tablets with internet access for Observable.

- Students will be able to define and calculate the mean, median, and mode for a given dataset.

- Students will be able to explain the differences between mean, median, and mode and identify situations where each measure is most appropriate.

- Students will be able to calculate mean, median, and mode using Python in Observable notebooks.

- Students will be able to interpret the mean, median, and mode in the context of scientific data and draw basic conclusions about central tendency.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Warm-Up (10 mins): "Guess the Average" - Present a few sets of simple numbers (e.g., ages of students in a small group, number of petals on different flowers). Ask students to quickly estimate the "average" value for each set. Discuss: What does "average" mean in everyday language? Are there different ways to think about "average"?

- Main Activity 1: "Mean, Median, Mode Definitions and Calculations" (25 mins):  
  
Introduce and define Mean (arithmetic average), Median (middle value when data is ordered), and Mode (most frequent value).  
Step-by-step calculation examples for each measure using small, simple datasets (done by hand initially).  
  
Mean: Sum of values divided by the number of values.  
Median: Ordering data and finding the middle value (or average of two middle values for even datasets).  
Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).  
  
Worksheet practice: Provide students with several small datasets. Students calculate the mean, median, and mode for each dataset by hand. Check answers as a class.

- Introduce and define Mean (arithmetic average), Median (middle value when data is ordered), and Mode (most frequent value).

- Step-by-step calculation examples for each measure using small, simple datasets (done by hand initially).  
  
Mean: Sum of values divided by the number of values.  
Median: Ordering data and finding the middle value (or average of two middle values for even datasets).  
Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).

- Mean: Sum of values divided by the number of values.

- Median: Ordering data and finding the middle value (or average of two middle values for even datasets).

- Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).

- Worksheet practice: Provide students with several small datasets. Students calculate the mean, median, and mode for each dataset by hand. Check answers as a class.

- Main Activity 2: "Mean, Median, Mode in Python (Observable)" (20 mins):  
  
Introduce how to calculate mean, median, and mode using Python in Observable notebooks.  
Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).  
Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.  
Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).  
  
Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the mean, median, and mode.

- Introduce how to calculate mean, median, and mode using Python in Observable notebooks.

- Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).  
Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.  
Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).

- Create a code cell.

- Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).

- Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.

- Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).

- Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the mean, median, and mode.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the similarities and differences between mean, median, and mode?  
When might the mean be the most appropriate measure of central tendency? When might the median be better? When might the mode be useful?  
How can understanding central tendency help us analyse and interpret scientific data? What kind of information does it give us?  
Are there any limitations to using just central tendency to describe a dataset? (Lead to the idea of data dispersion in the next lesson).

- What are the similarities and differences between mean, median, and mode?

- When might the mean be the most appropriate measure of central tendency? When might the median be better? When might the mode be useful?

- How can understanding central tendency help us analyse and interpret scientific data? What kind of information does it give us?

- Are there any limitations to using just central tendency to describe a dataset? (Lead to the idea of data dispersion in the next lesson).

- Assessments:  
  
Formative Quiz (5 mins): Short multiple-choice quiz on definitions of mean, median, mode and their basic calculation. AI-generated quiz.  
Review of student Python code in Observable notebooks – check for correct calculations and code syntax.

- Formative Quiz (5 mins): Short multiple-choice quiz on definitions of mean, median, mode and their basic calculation. AI-generated quiz.

- Review of student Python code in Observable notebooks – check for correct calculations and code syntax.

- Student Workbook:  
  
Definitions of mean, median, and mode with formulas.  
Step-by-step calculation examples for each measure.  
Worksheet with datasets for manual calculation practice.  
Observable notebook exercises with Python code templates for calculating mean, median, and mode.  
"Choosing the Right Measure" scenario worksheet with prompts to decide when to use mean, median, or mode.

- Definitions of mean, median, and mode with formulas.

- Step-by-step calculation examples for each measure.

- Worksheet with datasets for manual calculation practice.

- Observable notebook exercises with Python code templates for calculating mean, median, and mode.

- "Choosing the Right Measure" scenario worksheet with prompts to decide when to use mean, median, or mode.

- Extension Activities:  
  
Students research real-world examples where mean, median, and mode are used in different scientific fields (e.g., average temperature, median income, modal species in an ecosystem).  
Challenge: Investigate the concept of weighted mean and when it is used. Calculate a weighted mean for a given dataset.

- Students research real-world examples where mean, median, and mode are used in different scientific fields (e.g., average temperature, median income, modal species in an ecosystem).

- Challenge: Investigate the concept of weighted mean and when it is used. Calculate a weighted mean for a given dataset.

- Introduce and define Mean (arithmetic average), Median (middle value when data is ordered), and Mode (most frequent value).

- Step-by-step calculation examples for each measure using small, simple datasets (done by hand initially).  
  
Mean: Sum of values divided by the number of values.  
Median: Ordering data and finding the middle value (or average of two middle values for even datasets).  
Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).

- Mean: Sum of values divided by the number of values.

- Median: Ordering data and finding the middle value (or average of two middle values for even datasets).

- Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).

- Worksheet practice: Provide students with several small datasets. Students calculate the mean, median, and mode for each dataset by hand. Check answers as a class.

- Mean: Sum of values divided by the number of values.

- Median: Ordering data and finding the middle value (or average of two middle values for even datasets).

- Mode: Identifying the value that appears most frequently (datasets can have no mode, one mode, or multiple modes).

- Introduce how to calculate mean, median, and mode using Python in Observable notebooks.

- Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).  
Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.  
Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).

- Create a code cell.

- Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).

- Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.

- Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).

- Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the mean, median, and mode.

- Create a code cell.

- Create a Python list of numbers representing a scientific dataset (e.g., temperature readings, plant heights).

- Use numpy.mean(), numpy.median(), and scipy.stats.mode() functions to calculate the measures.

- Print the results with clear labels (e.g., "Mean Temperature:", mean\_temp).

- What are the similarities and differences between mean, median, and mode?

- When might the mean be the most appropriate measure of central tendency? When might the median be better? When might the mode be useful?

- How can understanding central tendency help us analyse and interpret scientific data? What kind of information does it give us?

- Are there any limitations to using just central tendency to describe a dataset? (Lead to the idea of data dispersion in the next lesson).

- Formative Quiz (5 mins): Short multiple-choice quiz on definitions of mean, median, mode and their basic calculation. AI-generated quiz.

- Review of student Python code in Observable notebooks – check for correct calculations and code syntax.

- Definitions of mean, median, and mode with formulas.

- Step-by-step calculation examples for each measure.

- Worksheet with datasets for manual calculation practice.

- Observable notebook exercises with Python code templates for calculating mean, median, and mode.

- "Choosing the Right Measure" scenario worksheet with prompts to decide when to use mean, median, or mode.

- Students research real-world examples where mean, median, and mode are used in different scientific fields (e.g., average temperature, median income, modal species in an ecosystem).

- Challenge: Investigate the concept of weighted mean and when it is used. Calculate a weighted mean for a given dataset.

- Slide Deck: Introduction to central tendency, definitions of mean, median, mode, calculation examples, choosing the appropriate measure, Python code examples.

- Handout: Worksheet with datasets for manual calculation practice, "Choosing the Right Measure" scenario worksheet, Python code templates for Observable exercises.

- Formative Quiz (printable or digital).

- Computers/Tablets with internet access for Observable.

### Lesson 4.2: Data Dispersion: Range - Understanding Data Spread

- Title: Beyond the Average: Range – Measuring the Spread of Data

- Learning Outcomes:  
  
Students will be able to define and calculate the range of a dataset.  
Students will be able to explain how range describes the dispersion or spread of data.  
Students will be able to calculate range using Python in Observable notebooks.  
Students will be able to interpret the range in the context of scientific data and compare the spread of different datasets.  
Students will understand the limitations of range as a measure of dispersion and recognize the need for more robust measures (to be introduced later).  
NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Students will be able to define and calculate the range of a dataset.

- Students will be able to explain how range describes the dispersion or spread of data.

- Students will be able to calculate range using Python in Observable notebooks.

- Students will be able to interpret the range in the context of scientific data and compare the spread of different datasets.

- Students will understand the limitations of range as a measure of dispersion and recognize the need for more robust measures (to be introduced later).

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Overview: This lesson introduces the concept of data dispersion, focusing on the range as a simple measure of spread. Students will learn to calculate range, understand its meaning, use Python in Observable for calculations, and recognize its limitations, setting the stage for more advanced measures of dispersion.

- Activities:  
  
Warm-Up (10 mins): "Data Spread Visualisation" - Show two sets of data visually represented (e.g., two line graphs with same mean but different spread, or two sets of bar graphs with different variability). Ask students: "Which dataset is more spread out? How can you describe or measure this 'spread'?" Lead to the idea of data dispersion and the need for measures beyond central tendency.  
Main Activity 1: "Range Definition and Calculation" (25 mins):  
  
Introduce and define Range: The difference between the maximum and minimum values in a dataset.  
Step-by-step calculation examples for range using small datasets (by hand). Example: Dataset: [12, 15, 18, 20, 25]. Range = 25 - 12 = 13.  
Explain how range describes the total spread of the data – how much the values vary from the lowest to the highest. A larger range indicates greater dispersion.  
Worksheet practice: Provide students with several datasets. Students calculate the range for each dataset by hand. Check answers as a class.  
  
Main Activity 2: "Range in Python (Observable)" (20 mins):  
  
Introduce how to calculate range using Python in Observable. (Note: NumPy doesn't directly have a 'range' function in the statistical sense, so we'll use max() and min() and subtraction).  
Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset.  
Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.  
Print the result with a clear label (e.g., "Data Range:", data\_range).  
  
Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the range.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What does the range tell us about a dataset that central tendency measures (mean, median, mode) don't?  
Compare datasets with the same mean but different ranges. What does the range reveal in this case?  
What are the limitations of using range as the only measure of data dispersion? (e.g., sensitive to outliers, doesn't describe the distribution within the range).  
When might range be a useful measure despite its limitations? (e.g., quick overview of spread, in situations where outliers are not a major concern).  
  
Assessments:  
  
Formative Task (5 mins): "Range Calculation Quiz" - Quick calculation of range for a very simple dataset. AI-generated quiz.  
Review of student Python code in Observable notebooks – check for correct range calculations and code syntax.  
  
Student Workbook:  
  
Definition of range with formula.  
Step-by-step calculation examples for range.  
Worksheet with datasets for manual range calculation practice.  
Observable notebook exercises with Python code templates for calculating range.  
"Interpreting Range" scenario worksheet with prompts to compare datasets based on their range and central tendency.  
  
Extension Activities:  
  
Students compare the range of different real-world scientific datasets (e.g., temperature ranges in different cities, height ranges of different plant species).  
Challenge: Investigate and compare range to other measures of dispersion like interquartile range (IQR) and standard deviation (brief conceptual introduction to IQR).

- Warm-Up (10 mins): "Data Spread Visualisation" - Show two sets of data visually represented (e.g., two line graphs with same mean but different spread, or two sets of bar graphs with different variability). Ask students: "Which dataset is more spread out? How can you describe or measure this 'spread'?" Lead to the idea of data dispersion and the need for measures beyond central tendency.

- Main Activity 1: "Range Definition and Calculation" (25 mins):  
  
Introduce and define Range: The difference between the maximum and minimum values in a dataset.  
Step-by-step calculation examples for range using small datasets (by hand). Example: Dataset: [12, 15, 18, 20, 25]. Range = 25 - 12 = 13.  
Explain how range describes the total spread of the data – how much the values vary from the lowest to the highest. A larger range indicates greater dispersion.  
Worksheet practice: Provide students with several datasets. Students calculate the range for each dataset by hand. Check answers as a class.

- Introduce and define Range: The difference between the maximum and minimum values in a dataset.

- Step-by-step calculation examples for range using small datasets (by hand). Example: Dataset: [12, 15, 18, 20, 25]. Range = 25 - 12 = 13.

- Explain how range describes the total spread of the data – how much the values vary from the lowest to the highest. A larger range indicates greater dispersion.

- Worksheet practice: Provide students with several datasets. Students calculate the range for each dataset by hand. Check answers as a class.

- Main Activity 2: "Range in Python (Observable)" (20 mins):  
  
Introduce how to calculate range using Python in Observable. (Note: NumPy doesn't directly have a 'range' function in the statistical sense, so we'll use max() and min() and subtraction).  
Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset.  
Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.  
Print the result with a clear label (e.g., "Data Range:", data\_range).  
  
Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the range.

- Introduce how to calculate range using Python in Observable. (Note: NumPy doesn't directly have a 'range' function in the statistical sense, so we'll use max() and min() and subtraction).

- Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset.  
Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.  
Print the result with a clear label (e.g., "Data Range:", data\_range).

- Create a code cell.

- Create a Python list of numbers representing a scientific dataset.

- Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.

- Print the result with a clear label (e.g., "Data Range:", data\_range).

- Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the range.

- Reflection & Discussion (10 mins): Class discussion:  
  
What does the range tell us about a dataset that central tendency measures (mean, median, mode) don't?  
Compare datasets with the same mean but different ranges. What does the range reveal in this case?  
What are the limitations of using range as the only measure of data dispersion? (e.g., sensitive to outliers, doesn't describe the distribution within the range).  
When might range be a useful measure despite its limitations? (e.g., quick overview of spread, in situations where outliers are not a major concern).

- What does the range tell us about a dataset that central tendency measures (mean, median, mode) don't?

- Compare datasets with the same mean but different ranges. What does the range reveal in this case?

- What are the limitations of using range as the only measure of data dispersion? (e.g., sensitive to outliers, doesn't describe the distribution within the range).

- When might range be a useful measure despite its limitations? (e.g., quick overview of spread, in situations where outliers are not a major concern).

- Assessments:  
  
Formative Task (5 mins): "Range Calculation Quiz" - Quick calculation of range for a very simple dataset. AI-generated quiz.  
Review of student Python code in Observable notebooks – check for correct range calculations and code syntax.

- Formative Task (5 mins): "Range Calculation Quiz" - Quick calculation of range for a very simple dataset. AI-generated quiz.

- Review of student Python code in Observable notebooks – check for correct range calculations and code syntax.

- Student Workbook:  
  
Definition of range with formula.  
Step-by-step calculation examples for range.  
Worksheet with datasets for manual range calculation practice.  
Observable notebook exercises with Python code templates for calculating range.  
"Interpreting Range" scenario worksheet with prompts to compare datasets based on their range and central tendency.

- Definition of range with formula.

- Step-by-step calculation examples for range.

- Worksheet with datasets for manual range calculation practice.

- Observable notebook exercises with Python code templates for calculating range.

- "Interpreting Range" scenario worksheet with prompts to compare datasets based on their range and central tendency.

- Extension Activities:  
  
Students compare the range of different real-world scientific datasets (e.g., temperature ranges in different cities, height ranges of different plant species).  
Challenge: Investigate and compare range to other measures of dispersion like interquartile range (IQR) and standard deviation (brief conceptual introduction to IQR).

- Students compare the range of different real-world scientific datasets (e.g., temperature ranges in different cities, height ranges of different plant species).

- Challenge: Investigate and compare range to other measures of dispersion like interquartile range (IQR) and standard deviation (brief conceptual introduction to IQR).

- Lesson Materials:  
  
Slide Deck: Introduction to data dispersion, definition of range, calculation examples, interpreting range, Python code examples, limitations of range.  
Handout: Worksheet with datasets for manual range calculation practice, "Interpreting Range" scenario worksheet, Python code templates for Observable exercises.  
Formative Quiz (printable or digital).  
Computers/Tablets with internet access for Observable.

- Slide Deck: Introduction to data dispersion, definition of range, calculation examples, interpreting range, Python code examples, limitations of range.

- Handout: Worksheet with datasets for manual range calculation practice, "Interpreting Range" scenario worksheet, Python code templates for Observable exercises.

- Formative Quiz (printable or digital).

- Computers/Tablets with internet access for Observable.

- Students will be able to define and calculate the range of a dataset.

- Students will be able to explain how range describes the dispersion or spread of data.

- Students will be able to calculate range using Python in Observable notebooks.

- Students will be able to interpret the range in the context of scientific data and compare the spread of different datasets.

- Students will understand the limitations of range as a measure of dispersion and recognize the need for more robust measures (to be introduced later).

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Warm-Up (10 mins): "Data Spread Visualisation" - Show two sets of data visually represented (e.g., two line graphs with same mean but different spread, or two sets of bar graphs with different variability). Ask students: "Which dataset is more spread out? How can you describe or measure this 'spread'?" Lead to the idea of data dispersion and the need for measures beyond central tendency.

- Main Activity 1: "Range Definition and Calculation" (25 mins):  
  
Introduce and define Range: The difference between the maximum and minimum values in a dataset.  
Step-by-step calculation examples for range using small datasets (by hand). Example: Dataset: [12, 15, 18, 20, 25]. Range = 25 - 12 = 13.  
Explain how range describes the total spread of the data – how much the values vary from the lowest to the highest. A larger range indicates greater dispersion.  
Worksheet practice: Provide students with several datasets. Students calculate the range for each dataset by hand. Check answers as a class.

- Introduce and define Range: The difference between the maximum and minimum values in a dataset.

- Step-by-step calculation examples for range using small datasets (by hand). Example: Dataset: [12, 15, 18, 20, 25]. Range = 25 - 12 = 13.

- Explain how range describes the total spread of the data – how much the values vary from the lowest to the highest. A larger range indicates greater dispersion.

- Worksheet practice: Provide students with several datasets. Students calculate the range for each dataset by hand. Check answers as a class.

- Main Activity 2: "Range in Python (Observable)" (20 mins):  
  
Introduce how to calculate range using Python in Observable. (Note: NumPy doesn't directly have a 'range' function in the statistical sense, so we'll use max() and min() and subtraction).  
Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset.  
Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.  
Print the result with a clear label (e.g., "Data Range:", data\_range).  
  
Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the range.

- Introduce how to calculate range using Python in Observable. (Note: NumPy doesn't directly have a 'range' function in the statistical sense, so we'll use max() and min() and subtraction).

- Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset.  
Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.  
Print the result with a clear label (e.g., "Data Range:", data\_range).

- Create a code cell.

- Create a Python list of numbers representing a scientific dataset.

- Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.

- Print the result with a clear label (e.g., "Data Range:", data\_range).

- Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the range.

- Reflection & Discussion (10 mins): Class discussion:  
  
What does the range tell us about a dataset that central tendency measures (mean, median, mode) don't?  
Compare datasets with the same mean but different ranges. What does the range reveal in this case?  
What are the limitations of using range as the only measure of data dispersion? (e.g., sensitive to outliers, doesn't describe the distribution within the range).  
When might range be a useful measure despite its limitations? (e.g., quick overview of spread, in situations where outliers are not a major concern).

- What does the range tell us about a dataset that central tendency measures (mean, median, mode) don't?

- Compare datasets with the same mean but different ranges. What does the range reveal in this case?

- What are the limitations of using range as the only measure of data dispersion? (e.g., sensitive to outliers, doesn't describe the distribution within the range).

- When might range be a useful measure despite its limitations? (e.g., quick overview of spread, in situations where outliers are not a major concern).

- Assessments:  
  
Formative Task (5 mins): "Range Calculation Quiz" - Quick calculation of range for a very simple dataset. AI-generated quiz.  
Review of student Python code in Observable notebooks – check for correct range calculations and code syntax.

- Formative Task (5 mins): "Range Calculation Quiz" - Quick calculation of range for a very simple dataset. AI-generated quiz.

- Review of student Python code in Observable notebooks – check for correct range calculations and code syntax.

- Student Workbook:  
  
Definition of range with formula.  
Step-by-step calculation examples for range.  
Worksheet with datasets for manual range calculation practice.  
Observable notebook exercises with Python code templates for calculating range.  
"Interpreting Range" scenario worksheet with prompts to compare datasets based on their range and central tendency.

- Definition of range with formula.

- Step-by-step calculation examples for range.

- Worksheet with datasets for manual range calculation practice.

- Observable notebook exercises with Python code templates for calculating range.

- "Interpreting Range" scenario worksheet with prompts to compare datasets based on their range and central tendency.

- Extension Activities:  
  
Students compare the range of different real-world scientific datasets (e.g., temperature ranges in different cities, height ranges of different plant species).  
Challenge: Investigate and compare range to other measures of dispersion like interquartile range (IQR) and standard deviation (brief conceptual introduction to IQR).

- Students compare the range of different real-world scientific datasets (e.g., temperature ranges in different cities, height ranges of different plant species).

- Challenge: Investigate and compare range to other measures of dispersion like interquartile range (IQR) and standard deviation (brief conceptual introduction to IQR).

- Introduce and define Range: The difference between the maximum and minimum values in a dataset.

- Step-by-step calculation examples for range using small datasets (by hand). Example: Dataset: [12, 15, 18, 20, 25]. Range = 25 - 12 = 13.

- Explain how range describes the total spread of the data – how much the values vary from the lowest to the highest. A larger range indicates greater dispersion.

- Worksheet practice: Provide students with several datasets. Students calculate the range for each dataset by hand. Check answers as a class.

- Introduce how to calculate range using Python in Observable. (Note: NumPy doesn't directly have a 'range' function in the statistical sense, so we'll use max() and min() and subtraction).

- Guided code-along in Observable:  
  
Create a code cell.  
Create a Python list of numbers representing a scientific dataset.  
Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.  
Print the result with a clear label (e.g., "Data Range:", data\_range).

- Create a code cell.

- Create a Python list of numbers representing a scientific dataset.

- Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.

- Print the result with a clear label (e.g., "Data Range:", data\_range).

- Student practice: Provide students with new datasets in Observable. Students write Python code to calculate and display the range.

- Create a code cell.

- Create a Python list of numbers representing a scientific dataset.

- Use numpy.max() to find the maximum value, numpy.min() to find the minimum value, and subtract to calculate the range.

- Print the result with a clear label (e.g., "Data Range:", data\_range).

- What does the range tell us about a dataset that central tendency measures (mean, median, mode) don't?

- Compare datasets with the same mean but different ranges. What does the range reveal in this case?

- What are the limitations of using range as the only measure of data dispersion? (e.g., sensitive to outliers, doesn't describe the distribution within the range).

- When might range be a useful measure despite its limitations? (e.g., quick overview of spread, in situations where outliers are not a major concern).

- Formative Task (5 mins): "Range Calculation Quiz" - Quick calculation of range for a very simple dataset. AI-generated quiz.

- Review of student Python code in Observable notebooks – check for correct range calculations and code syntax.

- Definition of range with formula.

- Step-by-step calculation examples for range.

- Worksheet with datasets for manual range calculation practice.

- Observable notebook exercises with Python code templates for calculating range.

- "Interpreting Range" scenario worksheet with prompts to compare datasets based on their range and central tendency.

- Students compare the range of different real-world scientific datasets (e.g., temperature ranges in different cities, height ranges of different plant species).

- Challenge: Investigate and compare range to other measures of dispersion like interquartile range (IQR) and standard deviation (brief conceptual introduction to IQR).

- Slide Deck: Introduction to data dispersion, definition of range, calculation examples, interpreting range, Python code examples, limitations of range.

- Handout: Worksheet with datasets for manual range calculation practice, "Interpreting Range" scenario worksheet, Python code templates for Observable exercises.

- Formative Quiz (printable or digital).

- Computers/Tablets with internet access for Observable.

### Lesson 4.3: Outliers: Identifying and Interpreting Unusual Data Points

- Title: Data Detectives: Unmasking Outliers – Spotting the Unusual Suspects in Data

- Learning Outcomes:  
  
Students will be able to define outliers as unusual or extreme values in a dataset.  
Students will be able to identify outliers visually using graphs (e.g., box plots, scatter plots) and numerically using simple rules (e.g., beyond a certain number of standard deviations from the mean – conceptual).  
Students will be able to interpret potential causes of outliers (errors, natural variation, interesting anomalies).  
Students will be able to discuss appropriate strategies for handling outliers in data analysis (investigate, correct, remove with justification, or keep).  
NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Students will be able to define outliers as unusual or extreme values in a dataset.

- Students will be able to identify outliers visually using graphs (e.g., box plots, scatter plots) and numerically using simple rules (e.g., beyond a certain number of standard deviations from the mean – conceptual).

- Students will be able to interpret potential causes of outliers (errors, natural variation, interesting anomalies).

- Students will be able to discuss appropriate strategies for handling outliers in data analysis (investigate, correct, remove with justification, or keep).

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Overview: This lesson introduces outliers as unusual data points and equips students with methods to identify and interpret them. Students will learn to spot outliers visually and numerically (conceptually), understand their potential causes, and discuss responsible strategies for handling outliers in scientific data analysis.

- Activities:  
  
Warm-Up (10 mins): "Spot the Odd One Out" - Present a series of lists of numbers or objects where one item is clearly an outlier (e.g., ages of students in a class with one value of 90, heights of plants with one value much taller than others). Ask students: "In each list, which value seems unusual or doesn't fit with the others? Why do you think it's unusual?" Lead to the concept of outliers as data points that are different from the rest.  
Main Activity 1: "What are Outliers? Definition and Visual Identification" (25 mins):  
  
Define Outliers: Data points that are significantly different from other values in a dataset. They lie far away from the main cluster of data.  
Discuss potential causes of outliers:  
  
Errors in data collection or measurement: Mistakes in recording data, faulty equipment.  
Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).  
Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).  
  
Visual identification of outliers using graphs:  
  
Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.  
Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.  
Demonstrate creating simple box plots in Observable (using Plot library - basic example).  
  
"Outlier Spotting Challenge (Visual)" - Present students with graphs (box plots, scatter plots) showing datasets with outliers. Students visually identify the outliers and describe their location on the graph.  
  
Main Activity 2: "Numerical Outlier Identification (Conceptual)" (20 mins):  
  
Introduce the concept of numerical outlier detection (without going into complex formulas at Year 8 level).  
Explain the idea of using standard deviation (conceptually - "how spread out the data is") or IQR (Interquartile Range - if introduced in previous extension activity) to define outlier boundaries.  
Simplified rule example: "Values that are much further than the typical spread from the mean or median might be outliers." (Avoid precise formulas at this stage).  
"Outlier Spotting Challenge (Numerical - Simplified Rule)" - Provide datasets and a simplified rule (e.g., "values more than 2 times the range away from the median are potential outliers"). Students apply the rule to identify potential outliers. Discuss that these are just guidelines, not definitive rules.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to identify outliers in scientific data?  
What are some possible reasons for outliers to occur?  
What should scientists do when they find outliers in their data? (Discuss different approaches - investigate, correct, remove, keep - and the importance of justification and transparency).  
Is it always "bad" to have outliers? Can outliers sometimes be important or interesting?  
  
Assessments:  
  
Formative Task (5 mins): "Outlier Definition Check" - Short answer question defining outliers and listing possible causes. AI-generated question.  
Observation of student participation in outlier spotting activities and discussions.  
  
Student Workbook:  
  
Definition of outliers and potential causes.  
Examples of outliers in different types of graphs (box plots, scatter plots).  
"Outlier Spotting Challenge (Visual)" worksheet with graphs.  
Simplified rule for numerical outlier identification (conceptual).  
"Outlier Spotting Challenge (Numerical - Simplified Rule)" worksheet.  
"Handling Outliers" scenario worksheet with prompts to discuss appropriate actions for different outlier situations.  
  
Extension Activities:  
  
Students research real-world examples of significant scientific discoveries that were initially considered outliers or anomalies (e.g., discovery of penicillin, unexpected astronomical observations).  
Challenge: Investigate more formal methods for numerical outlier detection (e.g., using z-scores, IQR rule – more detailed explanation of IQR). Implement a simple outlier detection function in Python in Observable using one of these methods (if appropriate for Year 8 level).

- Warm-Up (10 mins): "Spot the Odd One Out" - Present a series of lists of numbers or objects where one item is clearly an outlier (e.g., ages of students in a class with one value of 90, heights of plants with one value much taller than others). Ask students: "In each list, which value seems unusual or doesn't fit with the others? Why do you think it's unusual?" Lead to the concept of outliers as data points that are different from the rest.

- Main Activity 1: "What are Outliers? Definition and Visual Identification" (25 mins):  
  
Define Outliers: Data points that are significantly different from other values in a dataset. They lie far away from the main cluster of data.  
Discuss potential causes of outliers:  
  
Errors in data collection or measurement: Mistakes in recording data, faulty equipment.  
Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).  
Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).  
  
Visual identification of outliers using graphs:  
  
Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.  
Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.  
Demonstrate creating simple box plots in Observable (using Plot library - basic example).  
  
"Outlier Spotting Challenge (Visual)" - Present students with graphs (box plots, scatter plots) showing datasets with outliers. Students visually identify the outliers and describe their location on the graph.

- Define Outliers: Data points that are significantly different from other values in a dataset. They lie far away from the main cluster of data.

- Discuss potential causes of outliers:  
  
Errors in data collection or measurement: Mistakes in recording data, faulty equipment.  
Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).  
Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).

- Errors in data collection or measurement: Mistakes in recording data, faulty equipment.

- Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).

- Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).

- Visual identification of outliers using graphs:  
  
Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.  
Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.  
Demonstrate creating simple box plots in Observable (using Plot library - basic example).

- Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.

- Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.

- Demonstrate creating simple box plots in Observable (using Plot library - basic example).

- "Outlier Spotting Challenge (Visual)" - Present students with graphs (box plots, scatter plots) showing datasets with outliers. Students visually identify the outliers and describe their location on the graph.

- Main Activity 2: "Numerical Outlier Identification (Conceptual)" (20 mins):  
  
Introduce the concept of numerical outlier detection (without going into complex formulas at Year 8 level).  
Explain the idea of using standard deviation (conceptually - "how spread out the data is") or IQR (Interquartile Range - if introduced in previous extension activity) to define outlier boundaries.  
Simplified rule example: "Values that are much further than the typical spread from the mean or median might be outliers." (Avoid precise formulas at this stage).  
"Outlier Spotting Challenge (Numerical - Simplified Rule)" - Provide datasets and a simplified rule (e.g., "values more than 2 times the range away from the median are potential outliers"). Students apply the rule to identify potential outliers. Discuss that these are just guidelines, not definitive rules.

- Introduce the concept of numerical outlier detection (without going into complex formulas at Year 8 level).

- Explain the idea of using standard deviation (conceptually - "how spread out the data is") or IQR (Interquartile Range - if introduced in previous extension activity) to define outlier boundaries.

- Simplified rule example: "Values that are much further than the typical spread from the mean or median might be outliers." (Avoid precise formulas at this stage).

- "Outlier Spotting Challenge (Numerical - Simplified Rule)" - Provide datasets and a simplified rule (e.g., "values more than 2 times the range away from the median are potential outliers"). Students apply the rule to identify potential outliers. Discuss that these are just guidelines, not definitive rules.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to identify outliers in scientific data?  
What are some possible reasons for outliers to occur?  
What should scientists do when they find outliers in their data? (Discuss different approaches - investigate, correct, remove, keep - and the importance of justification and transparency).  
Is it always "bad" to have outliers? Can outliers sometimes be important or interesting?

- Why is it important to identify outliers in scientific data?

- What are some possible reasons for outliers to occur?

- What should scientists do when they find outliers in their data? (Discuss different approaches - investigate, correct, remove, keep - and the importance of justification and transparency).

- Is it always "bad" to have outliers? Can outliers sometimes be important or interesting?

- Assessments:  
  
Formative Task (5 mins): "Outlier Definition Check" - Short answer question defining outliers and listing possible causes. AI-generated question.  
Observation of student participation in outlier spotting activities and discussions.

- Formative Task (5 mins): "Outlier Definition Check" - Short answer question defining outliers and listing possible causes. AI-generated question.

- Observation of student participation in outlier spotting activities and discussions.

- Student Workbook:  
  
Definition of outliers and potential causes.  
Examples of outliers in different types of graphs (box plots, scatter plots).  
"Outlier Spotting Challenge (Visual)" worksheet with graphs.  
Simplified rule for numerical outlier identification (conceptual).  
"Outlier Spotting Challenge (Numerical - Simplified Rule)" worksheet.  
"Handling Outliers" scenario worksheet with prompts to discuss appropriate actions for different outlier situations.

- Definition of outliers and potential causes.

- Examples of outliers in different types of graphs (box plots, scatter plots).

- "Outlier Spotting Challenge (Visual)" worksheet with graphs.

- Simplified rule for numerical outlier identification (conceptual).

- "Outlier Spotting Challenge (Numerical - Simplified Rule)" worksheet.

- "Handling Outliers" scenario worksheet with prompts to discuss appropriate actions for different outlier situations.

- Extension Activities:  
  
Students research real-world examples of significant scientific discoveries that were initially considered outliers or anomalies (e.g., discovery of penicillin, unexpected astronomical observations).  
Challenge: Investigate more formal methods for numerical outlier detection (e.g., using z-scores, IQR rule – more detailed explanation of IQR). Implement a simple outlier detection function in Python in Observable using one of these methods (if appropriate for Year 8 level).

- Students research real-world examples of significant scientific discoveries that were initially considered outliers or anomalies (e.g., discovery of penicillin, unexpected astronomical observations).

- Challenge: Investigate more formal methods for numerical outlier detection (e.g., using z-scores, IQR rule – more detailed explanation of IQR). Implement a simple outlier detection function in Python in Observable using one of these methods (if appropriate for Year 8 level).

- Lesson Materials:  
  
Slide Deck: Introduction to outliers, definition, causes, visual outlier identification (box plots, scatter plots), numerical outlier identification (conceptual), handling outliers, importance of outliers.  
Handout: "Outlier Spotting Challenge (Visual)" worksheet with graphs, "Outlier Spotting Challenge (Numerical - Simplified Rule)" worksheet, "Handling Outliers" scenario worksheet, definitions of outliers and causes.  
Formative Quiz (printable or digital).  
Computers/Tablets with internet access for Observable.

- Slide Deck: Introduction to outliers, definition, causes, visual outlier identification (box plots, scatter plots), numerical outlier identification (conceptual), handling outliers, importance of outliers.

- Handout: "Outlier Spotting Challenge (Visual)" worksheet with graphs, "Outlier Spotting Challenge (Numerical - Simplified Rule)" worksheet, "Handling Outliers" scenario worksheet, definitions of outliers and causes.

- Formative Quiz (printable or digital).

- Computers/Tablets with internet access for Observable.

- Students will be able to define outliers as unusual or extreme values in a dataset.

- Students will be able to identify outliers visually using graphs (e.g., box plots, scatter plots) and numerically using simple rules (e.g., beyond a certain number of standard deviations from the mean – conceptual).

- Students will be able to interpret potential causes of outliers (errors, natural variation, interesting anomalies).

- Students will be able to discuss appropriate strategies for handling outliers in data analysis (investigate, correct, remove with justification, or keep).

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Analyse a model to identify data and trends)

- Warm-Up (10 mins): "Spot the Odd One Out" - Present a series of lists of numbers or objects where one item is clearly an outlier (e.g., ages of students in a class with one value of 90, heights of plants with one value much taller than others). Ask students: "In each list, which value seems unusual or doesn't fit with the others? Why do you think it's unusual?" Lead to the concept of outliers as data points that are different from the rest.

- Main Activity 1: "What are Outliers? Definition and Visual Identification" (25 mins):  
  
Define Outliers: Data points that are significantly different from other values in a dataset. They lie far away from the main cluster of data.  
Discuss potential causes of outliers:  
  
Errors in data collection or measurement: Mistakes in recording data, faulty equipment.  
Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).  
Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).  
  
Visual identification of outliers using graphs:  
  
Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.  
Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.  
Demonstrate creating simple box plots in Observable (using Plot library - basic example).  
  
"Outlier Spotting Challenge (Visual)" - Present students with graphs (box plots, scatter plots) showing datasets with outliers. Students visually identify the outliers and describe their location on the graph.

- Define Outliers: Data points that are significantly different from other values in a dataset. They lie far away from the main cluster of data.

- Discuss potential causes of outliers:  
  
Errors in data collection or measurement: Mistakes in recording data, faulty equipment.  
Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).  
Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).

- Errors in data collection or measurement: Mistakes in recording data, faulty equipment.

- Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).

- Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).

- Visual identification of outliers using graphs:  
  
Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.  
Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.  
Demonstrate creating simple box plots in Observable (using Plot library - basic example).

- Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.

- Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.

- Demonstrate creating simple box plots in Observable (using Plot library - basic example).

- "Outlier Spotting Challenge (Visual)" - Present students with graphs (box plots, scatter plots) showing datasets with outliers. Students visually identify the outliers and describe their location on the graph.

- Main Activity 2: "Numerical Outlier Identification (Conceptual)" (20 mins):  
  
Introduce the concept of numerical outlier detection (without going into complex formulas at Year 8 level).  
Explain the idea of using standard deviation (conceptually - "how spread out the data is") or IQR (Interquartile Range - if introduced in previous extension activity) to define outlier boundaries.  
Simplified rule example: "Values that are much further than the typical spread from the mean or median might be outliers." (Avoid precise formulas at this stage).  
"Outlier Spotting Challenge (Numerical - Simplified Rule)" - Provide datasets and a simplified rule (e.g., "values more than 2 times the range away from the median are potential outliers"). Students apply the rule to identify potential outliers. Discuss that these are just guidelines, not definitive rules.

- Introduce the concept of numerical outlier detection (without going into complex formulas at Year 8 level).

- Explain the idea of using standard deviation (conceptually - "how spread out the data is") or IQR (Interquartile Range - if introduced in previous extension activity) to define outlier boundaries.

- Simplified rule example: "Values that are much further than the typical spread from the mean or median might be outliers." (Avoid precise formulas at this stage).

- "Outlier Spotting Challenge (Numerical - Simplified Rule)" - Provide datasets and a simplified rule (e.g., "values more than 2 times the range away from the median are potential outliers"). Students apply the rule to identify potential outliers. Discuss that these are just guidelines, not definitive rules.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to identify outliers in scientific data?  
What are some possible reasons for outliers to occur?  
What should scientists do when they find outliers in their data? (Discuss different approaches - investigate, correct, remove, keep - and the importance of justification and transparency).  
Is it always "bad" to have outliers? Can outliers sometimes be important or interesting?

- Why is it important to identify outliers in scientific data?

- What are some possible reasons for outliers to occur?

- What should scientists do when they find outliers in their data? (Discuss different approaches - investigate, correct, remove, keep - and the importance of justification and transparency).

- Is it always "bad" to have outliers? Can outliers sometimes be important or interesting?

- Assessments:  
  
Formative Task (5 mins): "Outlier Definition Check" - Short answer question defining outliers and listing possible causes. AI-generated question.  
Observation of student participation in outlier spotting activities and discussions.

- Formative Task (5 mins): "Outlier Definition Check" - Short answer question defining outliers and listing possible causes. AI-generated question.

- Observation of student participation in outlier spotting activities and discussions.

- Student Workbook:  
  
Definition of outliers and potential causes.  
Examples of outliers in different types of graphs (box plots, scatter plots).  
"Outlier Spotting Challenge (Visual)" worksheet with graphs.  
Simplified rule for numerical outlier identification (conceptual).  
"Outlier Spotting Challenge (Numerical - Simplified Rule)" worksheet.  
"Handling Outliers" scenario worksheet with prompts to discuss appropriate actions for different outlier situations.

- Definition of outliers and potential causes.

- Examples of outliers in different types of graphs (box plots, scatter plots).

- "Outlier Spotting Challenge (Visual)" worksheet with graphs.

- Simplified rule for numerical outlier identification (conceptual).

- "Outlier Spotting Challenge (Numerical - Simplified Rule)" worksheet.

- "Handling Outliers" scenario worksheet with prompts to discuss appropriate actions for different outlier situations.

- Extension Activities:  
  
Students research real-world examples of significant scientific discoveries that were initially considered outliers or anomalies (e.g., discovery of penicillin, unexpected astronomical observations).  
Challenge: Investigate more formal methods for numerical outlier detection (e.g., using z-scores, IQR rule – more detailed explanation of IQR). Implement a simple outlier detection function in Python in Observable using one of these methods (if appropriate for Year 8 level).

- Students research real-world examples of significant scientific discoveries that were initially considered outliers or anomalies (e.g., discovery of penicillin, unexpected astronomical observations).

- Challenge: Investigate more formal methods for numerical outlier detection (e.g., using z-scores, IQR rule – more detailed explanation of IQR). Implement a simple outlier detection function in Python in Observable using one of these methods (if appropriate for Year 8 level).

- Define Outliers: Data points that are significantly different from other values in a dataset. They lie far away from the main cluster of data.

- Discuss potential causes of outliers:  
  
Errors in data collection or measurement: Mistakes in recording data, faulty equipment.  
Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).  
Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).

- Errors in data collection or measurement: Mistakes in recording data, faulty equipment.

- Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).

- Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).

- Visual identification of outliers using graphs:  
  
Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.  
Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.  
Demonstrate creating simple box plots in Observable (using Plot library - basic example).

- Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.

- Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.

- Demonstrate creating simple box plots in Observable (using Plot library - basic example).

- "Outlier Spotting Challenge (Visual)" - Present students with graphs (box plots, scatter plots) showing datasets with outliers. Students visually identify the outliers and describe their location on the graph.

- Errors in data collection or measurement: Mistakes in recording data, faulty equipment.

- Natural variation: Genuine extreme values within a distribution (e.g., exceptionally tall person in a height dataset).

- Interesting anomalies or new discoveries: Outliers might represent something significant or unexpected that needs further investigation (e.g., unexpected experimental result, indication of a new phenomenon).

- Box Plots: Show how outliers are represented as points beyond the whiskers of a box plot.

- Scatter Plots: Show how outliers are points far away from the main cluster in a scatter plot.

- Demonstrate creating simple box plots in Observable (using Plot library - basic example).

- Introduce the concept of numerical outlier detection (without going into complex formulas at Year 8 level).

- Explain the idea of using standard deviation (conceptually - "how spread out the data is") or IQR (Interquartile Range - if introduced in previous extension activity) to define outlier boundaries.

- Simplified rule example: "Values that are much further than the typical spread from the mean or median might be outliers." (Avoid precise formulas at this stage).

- "Outlier Spotting Challenge (Numerical - Simplified Rule)" - Provide datasets and a simplified rule (e.g., "values more than 2 times the range away from the median are potential outliers"). Students apply the rule to identify potential outliers. Discuss that these are just guidelines, not definitive rules.

- Why is it important to identify outliers in scientific data?

- What are some possible reasons for outliers to occur?

- What should scientists do when they find outliers in their data? (Discuss different approaches - investigate, correct, remove, keep - and the importance of justification and transparency).

- Is it always "bad" to have outliers? Can outliers sometimes be important or interesting?

- Formative Task (5 mins): "Outlier Definition Check" - Short answer question defining outliers and listing possible causes. AI-generated question.

- Observation of student participation in outlier spotting activities and discussions.

- Definition of outliers and potential causes.

- Examples of outliers in different types of graphs (box plots, scatter plots).

- "Outlier Spotting Challenge (Visual)" worksheet with graphs.

- Simplified rule for numerical outlier identification (conceptual).

- "Outlier Spotting Challenge (Numerical - Simplified Rule)" worksheet.

- "Handling Outliers" scenario worksheet with prompts to discuss appropriate actions for different outlier situations.

- Students research real-world examples of significant scientific discoveries that were initially considered outliers or anomalies (e.g., discovery of penicillin, unexpected astronomical observations).

- Challenge: Investigate more formal methods for numerical outlier detection (e.g., using z-scores, IQR rule – more detailed explanation of IQR). Implement a simple outlier detection function in Python in Observable using one of these methods (if appropriate for Year 8 level).

- Slide Deck: Introduction to outliers, definition, causes, visual outlier identification (box plots, scatter plots), numerical outlier identification (conceptual), handling outliers, importance of outliers.

- Handout: "Outlier Spotting Challenge (Visual)" worksheet with graphs, "Outlier Spotting Challenge (Numerical - Simplified Rule)" worksheet, "Handling Outliers" scenario worksheet, definitions of outliers and causes.

- Formative Quiz (printable or digital).

- Computers/Tablets with internet access for Observable.

### Week 5: Scientific Question Formulation & Experimental Design - Advanced

### Lesson 5.1: Asking Testable Questions: From Observation to Inquiry

- Title: Igniting Inquiry: Asking Powerful, Testable Scientific Questions

- Learning Outcomes:  
  
Students will be able to formulate scientific questions that are testable and answerable through data collection and analysis.  
Students will be able to differentiate between descriptive, comparative, correlational, and causal types of scientific questions.  
Students will be able to refine broad or vague questions into focused and testable scientific questions.  
NSW Curriculum Outcomes: SC4-WS-01 (Observing and questioning), ACSIS124 (Formulating questions).

- Students will be able to formulate scientific questions that are testable and answerable through data collection and analysis.

- Students will be able to differentiate between descriptive, comparative, correlational, and causal types of scientific questions.

- Students will be able to refine broad or vague questions into focused and testable scientific questions.

- NSW Curriculum Outcomes: SC4-WS-01 (Observing and questioning), ACSIS124 (Formulating questions).

- Overview: This lesson focuses on the crucial skill of formulating strong scientific questions. Students will learn the characteristics of testable questions, explore different question types, and practice refining questions to be specific and answerable through scientific investigation.

- Activities:  
  
Warm-Up (10 mins): "Observation Challenge - Nature Walk (Virtual or Real)" - Show a short video clip of a natural environment (forest, garden, ocean scene) or, if feasible, take students on a brief walk in the schoolyard. Ask students to make detailed observations and write down at least three questions that come to mind based on their observations. Share a few examples and discuss the initial questions.  
Main Activity 1: "Characteristics of Testable Scientific Questions" (25 mins):  
  
Introduce the key characteristics of a good testable scientific question:  
  
Testable: Can be investigated through experiments, observations, or data collection.  
Focused: Specific and not too broad.  
Measurable/Observable: Involves variables that can be measured or observed.  
Clear and Unambiguous: Easy to understand what is being asked.  
Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.  
  
Present examples of good and bad scientific questions (related to various science topics). For each example, discuss whether it meets the criteria of a testable question and why.  
"Question Critique" activity: Provide students with a list of questions (some good, some bad). In pairs, students critique each question against the criteria and classify them as "testable" or "not testable" with justifications. Discuss as a class.  
  
Main Activity 2: "Question Types and Refinement" (20 mins):  
  
Introduce different types of scientific questions (conceptually, no rigid definitions needed at Year 8):  
  
Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").  
Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").  
Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").  
Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").  
  
Provide examples of each question type. Discuss: What kind of data would you collect to answer each type of question? What kind of analysis would be appropriate?  
"Question Refinement Challenge": Provide students with broad or vague initial questions (e.g., "How does the environment affect living things?"). In groups, students refine these broad questions into more focused and testable scientific questions, specifying the question type and how they could be investigated. Share and discuss refined questions as a class.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to ask testable questions in science? Why not just ask any question?  
How does understanding different question types help in designing scientific investigations?  
What are some common mistakes to avoid when formulating scientific questions?  
How can you improve a question that is not initially testable?  
  
Assessments:  
  
Formative Task (5 mins): "Testable Question Identification" - Quick multiple-choice quiz: Students identify which of several questions is the most testable scientific question. AI-generated quiz.  
Review of group-refined questions from "Question Refinement Challenge" – assess focus, testability, and clarity.  
  
Student Workbook:  
  
Characteristics of testable scientific questions (checklist).  
Examples of good and bad scientific questions with critique prompts.  
Descriptions of different question types (descriptive, comparative, correlational, causal – simplified).  
"Question Critique" worksheet with question lists.  
"Question Refinement Challenge" worksheet with broad questions and prompts for refinement.  
  
Extension Activities:  
  
Students choose a scientific topic they are interested in and formulate a range of testable scientific questions related to that topic (aim for different question types).  
Challenge: Research famous scientific questions that have driven major scientific discoveries throughout history. Analyse why these questions were so important and effective in guiding research.

- Warm-Up (10 mins): "Observation Challenge - Nature Walk (Virtual or Real)" - Show a short video clip of a natural environment (forest, garden, ocean scene) or, if feasible, take students on a brief walk in the schoolyard. Ask students to make detailed observations and write down at least three questions that come to mind based on their observations. Share a few examples and discuss the initial questions.

- Main Activity 1: "Characteristics of Testable Scientific Questions" (25 mins):  
  
Introduce the key characteristics of a good testable scientific question:  
  
Testable: Can be investigated through experiments, observations, or data collection.  
Focused: Specific and not too broad.  
Measurable/Observable: Involves variables that can be measured or observed.  
Clear and Unambiguous: Easy to understand what is being asked.  
Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.  
  
Present examples of good and bad scientific questions (related to various science topics). For each example, discuss whether it meets the criteria of a testable question and why.  
"Question Critique" activity: Provide students with a list of questions (some good, some bad). In pairs, students critique each question against the criteria and classify them as "testable" or "not testable" with justifications. Discuss as a class.

- Introduce the key characteristics of a good testable scientific question:  
  
Testable: Can be investigated through experiments, observations, or data collection.  
Focused: Specific and not too broad.  
Measurable/Observable: Involves variables that can be measured or observed.  
Clear and Unambiguous: Easy to understand what is being asked.  
Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.

- Testable: Can be investigated through experiments, observations, or data collection.

- Focused: Specific and not too broad.

- Measurable/Observable: Involves variables that can be measured or observed.

- Clear and Unambiguous: Easy to understand what is being asked.

- Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.

- Present examples of good and bad scientific questions (related to various science topics). For each example, discuss whether it meets the criteria of a testable question and why.

- "Question Critique" activity: Provide students with a list of questions (some good, some bad). In pairs, students critique each question against the criteria and classify them as "testable" or "not testable" with justifications. Discuss as a class.

- Main Activity 2: "Question Types and Refinement" (20 mins):  
  
Introduce different types of scientific questions (conceptually, no rigid definitions needed at Year 8):  
  
Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").  
Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").  
Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").  
Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").  
  
Provide examples of each question type. Discuss: What kind of data would you collect to answer each type of question? What kind of analysis would be appropriate?  
"Question Refinement Challenge": Provide students with broad or vague initial questions (e.g., "How does the environment affect living things?"). In groups, students refine these broad questions into more focused and testable scientific questions, specifying the question type and how they could be investigated. Share and discuss refined questions as a class.

- Introduce different types of scientific questions (conceptually, no rigid definitions needed at Year 8):  
  
Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").  
Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").  
Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").  
Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").

- Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").

- Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").

- Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").

- Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").

- Provide examples of each question type. Discuss: What kind of data would you collect to answer each type of question? What kind of analysis would be appropriate?

- "Question Refinement Challenge": Provide students with broad or vague initial questions (e.g., "How does the environment affect living things?"). In groups, students refine these broad questions into more focused and testable scientific questions, specifying the question type and how they could be investigated. Share and discuss refined questions as a class.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to ask testable questions in science? Why not just ask any question?  
How does understanding different question types help in designing scientific investigations?  
What are some common mistakes to avoid when formulating scientific questions?  
How can you improve a question that is not initially testable?

- Why is it important to ask testable questions in science? Why not just ask any question?

- How does understanding different question types help in designing scientific investigations?

- What are some common mistakes to avoid when formulating scientific questions?

- How can you improve a question that is not initially testable?

- Assessments:  
  
Formative Task (5 mins): "Testable Question Identification" - Quick multiple-choice quiz: Students identify which of several questions is the most testable scientific question. AI-generated quiz.  
Review of group-refined questions from "Question Refinement Challenge" – assess focus, testability, and clarity.

- Formative Task (5 mins): "Testable Question Identification" - Quick multiple-choice quiz: Students identify which of several questions is the most testable scientific question. AI-generated quiz.

- Review of group-refined questions from "Question Refinement Challenge" – assess focus, testability, and clarity.

- Student Workbook:  
  
Characteristics of testable scientific questions (checklist).  
Examples of good and bad scientific questions with critique prompts.  
Descriptions of different question types (descriptive, comparative, correlational, causal – simplified).  
"Question Critique" worksheet with question lists.  
"Question Refinement Challenge" worksheet with broad questions and prompts for refinement.

- Characteristics of testable scientific questions (checklist).

- Examples of good and bad scientific questions with critique prompts.

- Descriptions of different question types (descriptive, comparative, correlational, causal – simplified).

- "Question Critique" worksheet with question lists.

- "Question Refinement Challenge" worksheet with broad questions and prompts for refinement.

- Extension Activities:  
  
Students choose a scientific topic they are interested in and formulate a range of testable scientific questions related to that topic (aim for different question types).  
Challenge: Research famous scientific questions that have driven major scientific discoveries throughout history. Analyse why these questions were so important and effective in guiding research.

- Students choose a scientific topic they are interested in and formulate a range of testable scientific questions related to that topic (aim for different question types).

- Challenge: Research famous scientific questions that have driven major scientific discoveries throughout history. Analyse why these questions were so important and effective in guiding research.

- Lesson Materials:  
  
Slide Deck: Introduction to testable questions, characteristics of good questions, examples of good and bad questions, question types, question refinement process.  
Handout: "Question Critique" worksheet with question lists, "Question Refinement Challenge" worksheet with broad questions, checklist for testable questions.  
Formative Quiz (printable or digital).  
Video clip of natural environment (optional).

- Slide Deck: Introduction to testable questions, characteristics of good questions, examples of good and bad questions, question types, question refinement process.

- Handout: "Question Critique" worksheet with question lists, "Question Refinement Challenge" worksheet with broad questions, checklist for testable questions.

- Formative Quiz (printable or digital).

- Video clip of natural environment (optional).

- Students will be able to formulate scientific questions that are testable and answerable through data collection and analysis.

- Students will be able to differentiate between descriptive, comparative, correlational, and causal types of scientific questions.

- Students will be able to refine broad or vague questions into focused and testable scientific questions.

- NSW Curriculum Outcomes: SC4-WS-01 (Observing and questioning), ACSIS124 (Formulating questions).

- Warm-Up (10 mins): "Observation Challenge - Nature Walk (Virtual or Real)" - Show a short video clip of a natural environment (forest, garden, ocean scene) or, if feasible, take students on a brief walk in the schoolyard. Ask students to make detailed observations and write down at least three questions that come to mind based on their observations. Share a few examples and discuss the initial questions.

- Main Activity 1: "Characteristics of Testable Scientific Questions" (25 mins):  
  
Introduce the key characteristics of a good testable scientific question:  
  
Testable: Can be investigated through experiments, observations, or data collection.  
Focused: Specific and not too broad.  
Measurable/Observable: Involves variables that can be measured or observed.  
Clear and Unambiguous: Easy to understand what is being asked.  
Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.  
  
Present examples of good and bad scientific questions (related to various science topics). For each example, discuss whether it meets the criteria of a testable question and why.  
"Question Critique" activity: Provide students with a list of questions (some good, some bad). In pairs, students critique each question against the criteria and classify them as "testable" or "not testable" with justifications. Discuss as a class.

- Introduce the key characteristics of a good testable scientific question:  
  
Testable: Can be investigated through experiments, observations, or data collection.  
Focused: Specific and not too broad.  
Measurable/Observable: Involves variables that can be measured or observed.  
Clear and Unambiguous: Easy to understand what is being asked.  
Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.

- Testable: Can be investigated through experiments, observations, or data collection.

- Focused: Specific and not too broad.

- Measurable/Observable: Involves variables that can be measured or observed.

- Clear and Unambiguous: Easy to understand what is being asked.

- Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.

- Present examples of good and bad scientific questions (related to various science topics). For each example, discuss whether it meets the criteria of a testable question and why.

- "Question Critique" activity: Provide students with a list of questions (some good, some bad). In pairs, students critique each question against the criteria and classify them as "testable" or "not testable" with justifications. Discuss as a class.

- Main Activity 2: "Question Types and Refinement" (20 mins):  
  
Introduce different types of scientific questions (conceptually, no rigid definitions needed at Year 8):  
  
Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").  
Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").  
Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").  
Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").  
  
Provide examples of each question type. Discuss: What kind of data would you collect to answer each type of question? What kind of analysis would be appropriate?  
"Question Refinement Challenge": Provide students with broad or vague initial questions (e.g., "How does the environment affect living things?"). In groups, students refine these broad questions into more focused and testable scientific questions, specifying the question type and how they could be investigated. Share and discuss refined questions as a class.

- Introduce different types of scientific questions (conceptually, no rigid definitions needed at Year 8):  
  
Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").  
Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").  
Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").  
Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").

- Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").

- Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").

- Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").

- Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").

- Provide examples of each question type. Discuss: What kind of data would you collect to answer each type of question? What kind of analysis would be appropriate?

- "Question Refinement Challenge": Provide students with broad or vague initial questions (e.g., "How does the environment affect living things?"). In groups, students refine these broad questions into more focused and testable scientific questions, specifying the question type and how they could be investigated. Share and discuss refined questions as a class.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to ask testable questions in science? Why not just ask any question?  
How does understanding different question types help in designing scientific investigations?  
What are some common mistakes to avoid when formulating scientific questions?  
How can you improve a question that is not initially testable?

- Why is it important to ask testable questions in science? Why not just ask any question?

- How does understanding different question types help in designing scientific investigations?

- What are some common mistakes to avoid when formulating scientific questions?

- How can you improve a question that is not initially testable?

- Assessments:  
  
Formative Task (5 mins): "Testable Question Identification" - Quick multiple-choice quiz: Students identify which of several questions is the most testable scientific question. AI-generated quiz.  
Review of group-refined questions from "Question Refinement Challenge" – assess focus, testability, and clarity.

- Formative Task (5 mins): "Testable Question Identification" - Quick multiple-choice quiz: Students identify which of several questions is the most testable scientific question. AI-generated quiz.

- Review of group-refined questions from "Question Refinement Challenge" – assess focus, testability, and clarity.

- Student Workbook:  
  
Characteristics of testable scientific questions (checklist).  
Examples of good and bad scientific questions with critique prompts.  
Descriptions of different question types (descriptive, comparative, correlational, causal – simplified).  
"Question Critique" worksheet with question lists.  
"Question Refinement Challenge" worksheet with broad questions and prompts for refinement.

- Characteristics of testable scientific questions (checklist).

- Examples of good and bad scientific questions with critique prompts.

- Descriptions of different question types (descriptive, comparative, correlational, causal – simplified).

- "Question Critique" worksheet with question lists.

- "Question Refinement Challenge" worksheet with broad questions and prompts for refinement.

- Extension Activities:  
  
Students choose a scientific topic they are interested in and formulate a range of testable scientific questions related to that topic (aim for different question types).  
Challenge: Research famous scientific questions that have driven major scientific discoveries throughout history. Analyse why these questions were so important and effective in guiding research.

- Students choose a scientific topic they are interested in and formulate a range of testable scientific questions related to that topic (aim for different question types).

- Challenge: Research famous scientific questions that have driven major scientific discoveries throughout history. Analyse why these questions were so important and effective in guiding research.

- Introduce the key characteristics of a good testable scientific question:  
  
Testable: Can be investigated through experiments, observations, or data collection.  
Focused: Specific and not too broad.  
Measurable/Observable: Involves variables that can be measured or observed.  
Clear and Unambiguous: Easy to understand what is being asked.  
Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.

- Testable: Can be investigated through experiments, observations, or data collection.

- Focused: Specific and not too broad.

- Measurable/Observable: Involves variables that can be measured or observed.

- Clear and Unambiguous: Easy to understand what is being asked.

- Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.

- Present examples of good and bad scientific questions (related to various science topics). For each example, discuss whether it meets the criteria of a testable question and why.

- "Question Critique" activity: Provide students with a list of questions (some good, some bad). In pairs, students critique each question against the criteria and classify them as "testable" or "not testable" with justifications. Discuss as a class.

- Testable: Can be investigated through experiments, observations, or data collection.

- Focused: Specific and not too broad.

- Measurable/Observable: Involves variables that can be measured or observed.

- Clear and Unambiguous: Easy to understand what is being asked.

- Relevant and Interesting (Ideally): Connects to scientific concepts and sparks curiosity.

- Introduce different types of scientific questions (conceptually, no rigid definitions needed at Year 8):  
  
Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").  
Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").  
Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").  
Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").

- Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").

- Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").

- Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").

- Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").

- Provide examples of each question type. Discuss: What kind of data would you collect to answer each type of question? What kind of analysis would be appropriate?

- "Question Refinement Challenge": Provide students with broad or vague initial questions (e.g., "How does the environment affect living things?"). In groups, students refine these broad questions into more focused and testable scientific questions, specifying the question type and how they could be investigated. Share and discuss refined questions as a class.

- Descriptive Questions: Aim to describe something (e.g., "What is the biodiversity of insects in our schoolyard?").

- Comparative Questions: Aim to compare two or more things (e.g., "Is the plant growth rate different in sunny vs. shady areas?").

- Correlational Questions: Aim to investigate relationships between variables (e.g., "Is there a relationship between air temperature and cricket chirping rate?").

- Causal Questions: Aim to determine cause-and-effect relationships (e.g., "Does fertilizer X cause plants to grow taller?").

- Why is it important to ask testable questions in science? Why not just ask any question?

- How does understanding different question types help in designing scientific investigations?

- What are some common mistakes to avoid when formulating scientific questions?

- How can you improve a question that is not initially testable?

- Formative Task (5 mins): "Testable Question Identification" - Quick multiple-choice quiz: Students identify which of several questions is the most testable scientific question. AI-generated quiz.

- Review of group-refined questions from "Question Refinement Challenge" – assess focus, testability, and clarity.

- Characteristics of testable scientific questions (checklist).

- Examples of good and bad scientific questions with critique prompts.

- Descriptions of different question types (descriptive, comparative, correlational, causal – simplified).

- "Question Critique" worksheet with question lists.

- "Question Refinement Challenge" worksheet with broad questions and prompts for refinement.

- Students choose a scientific topic they are interested in and formulate a range of testable scientific questions related to that topic (aim for different question types).

- Challenge: Research famous scientific questions that have driven major scientific discoveries throughout history. Analyse why these questions were so important and effective in guiding research.

- Slide Deck: Introduction to testable questions, characteristics of good questions, examples of good and bad questions, question types, question refinement process.

- Handout: "Question Critique" worksheet with question lists, "Question Refinement Challenge" worksheet with broad questions, checklist for testable questions.

- Formative Quiz (printable or digital).

- Video clip of natural environment (optional).

### Lesson 5.2: Experimental Design: Variables, Controls, and Groups

- Title: Blueprint for Investigation: Designing Robust Scientific Experiments

- Learning Outcomes:  
  
Students will be able to define and differentiate between independent, dependent, and controlled variables in an experiment.  
Students will be able to explain the purpose of control groups and experimental groups in controlled experiments.  
Students will be able to design basic controlled experiments to investigate scientific questions, identifying variables and controls.  
NSW Curriculum Outcomes: SC4-WS-02 (Planning investigations), ACSIS125 (Conducting investigations safely).

- Students will be able to define and differentiate between independent, dependent, and controlled variables in an experiment.

- Students will be able to explain the purpose of control groups and experimental groups in controlled experiments.

- Students will be able to design basic controlled experiments to investigate scientific questions, identifying variables and controls.

- NSW Curriculum Outcomes: SC4-WS-02 (Planning investigations), ACSIS125 (Conducting investigations safely).

- Overview: This lesson focuses on the fundamental principles of experimental design, emphasizing variables, controls, and experimental groups. Students will learn to identify different types of variables, understand the importance of controls, and design their own controlled experiments.

- Activities:  
  
Warm-Up (10 mins): "Variable Scenarios" - Present several simple experimental scenarios (verbally or on slides - e.g., plant growth experiment, reaction rate experiment, survey on sleep habits). For each scenario, ask students: "What factor are we changing or manipulating in this experiment? What factor are we measuring or observing to see if it changes? What factors should we keep the same to make it a fair test?" Lead to the concepts of independent, dependent, and controlled variables.  
Main Activity 1: "Variables Unpacked: Independent, Dependent, Controlled" (25 mins):  
  
Introduce and define:  
  
Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").  
Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").  
Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.  
  
Use real-world examples and analogies to explain each variable type (e.g., in a recipe, changing oven temperature (independent) and measuring cake rise (dependent), while keeping ingredients and baking time constant (controlled)).  
"Variable Identification Challenge": Provide students with descriptions of different experiments (from various science topics). For each experiment, students identify the independent variable, dependent variable, and list several important controlled variables. Discuss as a class and clarify variable types.  
  
Main Activity 2: "Control Groups and Experimental Groups - The Power of Comparison" (20 mins):  
  
Explain the purpose of control groups and experimental groups in controlled experiments:  
  
Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).  
Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.  
  
Explain why control groups are essential for drawing valid conclusions in experiments: To isolate the effect of the independent variable and rule out other factors.  
"Control Group Design Challenge": Present scientific questions that require controlled experiments (e.g., "Does a new fertilizer increase plant growth?", "Does a new drug reduce reaction time?"). For each question, students design a basic controlled experiment, specifying:  
  
Independent Variable and how it will be manipulated.  
Dependent Variable and how it will be measured.  
Controlled Variables (list at least 3).  
Experimental Group(s) and Control Group(s) description.  
  
Groups share their experimental designs. Class feedback and discussion – peer review of experimental designs, focusing on clarity of variables and effective use of control groups.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why are variables and controls so important in experimental design?  
What happens if you don't control variables properly in an experiment? What kind of conclusions can you draw?  
Why is a control group essential for a valid controlled experiment?  
What are some real-world examples where controlled experiments are used to test new ideas or products?  
  
Assessments:  
  
Formative Task (5 mins): "Variable Matching Quiz" - Matching exercise: Students match variable types (independent, dependent, controlled) to their definitions. AI-generated quiz.  
Review of group-designed experiments from "Control Group Design Challenge" – assess clarity of variable identification, effective use of control groups, and feasibility of design.  
  
Student Workbook:  
  
Definitions of independent, dependent, and controlled variables with examples.  
Explanation of control groups and experimental groups and their purpose.  
"Variable Identification Challenge" worksheet with experiment descriptions.  
"Control Group Design Challenge" worksheet with scientific questions and prompts for experimental design.  
Checklist for designing a controlled experiment (variables, controls, groups).  
  
Extension Activities:  
  
Students research famous controlled experiments in science history (e.g., Pasteur's experiments, clinical trials for new medicines). Analyse the experimental design, identifying variables, controls, and the importance of these elements in the study's conclusions.  
Challenge: Design experiments with more complex experimental designs (e.g., experiments with multiple levels of the independent variable, factorial designs – conceptually introduced). Discuss the advantages of these more complex designs.

- Warm-Up (10 mins): "Variable Scenarios" - Present several simple experimental scenarios (verbally or on slides - e.g., plant growth experiment, reaction rate experiment, survey on sleep habits). For each scenario, ask students: "What factor are we changing or manipulating in this experiment? What factor are we measuring or observing to see if it changes? What factors should we keep the same to make it a fair test?" Lead to the concepts of independent, dependent, and controlled variables.

- Main Activity 1: "Variables Unpacked: Independent, Dependent, Controlled" (25 mins):  
  
Introduce and define:  
  
Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").  
Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").  
Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.  
  
Use real-world examples and analogies to explain each variable type (e.g., in a recipe, changing oven temperature (independent) and measuring cake rise (dependent), while keeping ingredients and baking time constant (controlled)).  
"Variable Identification Challenge": Provide students with descriptions of different experiments (from various science topics). For each experiment, students identify the independent variable, dependent variable, and list several important controlled variables. Discuss as a class and clarify variable types.

- Introduce and define:  
  
Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").  
Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").  
Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.

- Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").

- Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").

- Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.

- Use real-world examples and analogies to explain each variable type (e.g., in a recipe, changing oven temperature (independent) and measuring cake rise (dependent), while keeping ingredients and baking time constant (controlled)).

- "Variable Identification Challenge": Provide students with descriptions of different experiments (from various science topics). For each experiment, students identify the independent variable, dependent variable, and list several important controlled variables. Discuss as a class and clarify variable types.

- Main Activity 2: "Control Groups and Experimental Groups - The Power of Comparison" (20 mins):  
  
Explain the purpose of control groups and experimental groups in controlled experiments:  
  
Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).  
Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.  
  
Explain why control groups are essential for drawing valid conclusions in experiments: To isolate the effect of the independent variable and rule out other factors.  
"Control Group Design Challenge": Present scientific questions that require controlled experiments (e.g., "Does a new fertilizer increase plant growth?", "Does a new drug reduce reaction time?"). For each question, students design a basic controlled experiment, specifying:  
  
Independent Variable and how it will be manipulated.  
Dependent Variable and how it will be measured.  
Controlled Variables (list at least 3).  
Experimental Group(s) and Control Group(s) description.  
  
Groups share their experimental designs. Class feedback and discussion – peer review of experimental designs, focusing on clarity of variables and effective use of control groups.

- Explain the purpose of control groups and experimental groups in controlled experiments:  
  
Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).  
Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.

- Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).

- Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.

- Explain why control groups are essential for drawing valid conclusions in experiments: To isolate the effect of the independent variable and rule out other factors.

- "Control Group Design Challenge": Present scientific questions that require controlled experiments (e.g., "Does a new fertilizer increase plant growth?", "Does a new drug reduce reaction time?"). For each question, students design a basic controlled experiment, specifying:  
  
Independent Variable and how it will be manipulated.  
Dependent Variable and how it will be measured.  
Controlled Variables (list at least 3).  
Experimental Group(s) and Control Group(s) description.

- Independent Variable and how it will be manipulated.

- Dependent Variable and how it will be measured.

- Controlled Variables (list at least 3).

- Experimental Group(s) and Control Group(s) description.

- Groups share their experimental designs. Class feedback and discussion – peer review of experimental designs, focusing on clarity of variables and effective use of control groups.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are variables and controls so important in experimental design?  
What happens if you don't control variables properly in an experiment? What kind of conclusions can you draw?  
Why is a control group essential for a valid controlled experiment?  
What are some real-world examples where controlled experiments are used to test new ideas or products?

- Why are variables and controls so important in experimental design?

- What happens if you don't control variables properly in an experiment? What kind of conclusions can you draw?

- Why is a control group essential for a valid controlled experiment?

- What are some real-world examples where controlled experiments are used to test new ideas or products?

- Assessments:  
  
Formative Task (5 mins): "Variable Matching Quiz" - Matching exercise: Students match variable types (independent, dependent, controlled) to their definitions. AI-generated quiz.  
Review of group-designed experiments from "Control Group Design Challenge" – assess clarity of variable identification, effective use of control groups, and feasibility of design.

- Formative Task (5 mins): "Variable Matching Quiz" - Matching exercise: Students match variable types (independent, dependent, controlled) to their definitions. AI-generated quiz.

- Review of group-designed experiments from "Control Group Design Challenge" – assess clarity of variable identification, effective use of control groups, and feasibility of design.

- Student Workbook:  
  
Definitions of independent, dependent, and controlled variables with examples.  
Explanation of control groups and experimental groups and their purpose.  
"Variable Identification Challenge" worksheet with experiment descriptions.  
"Control Group Design Challenge" worksheet with scientific questions and prompts for experimental design.  
Checklist for designing a controlled experiment (variables, controls, groups).

- Definitions of independent, dependent, and controlled variables with examples.

- Explanation of control groups and experimental groups and their purpose.

- "Variable Identification Challenge" worksheet with experiment descriptions.

- "Control Group Design Challenge" worksheet with scientific questions and prompts for experimental design.

- Checklist for designing a controlled experiment (variables, controls, groups).

- Extension Activities:  
  
Students research famous controlled experiments in science history (e.g., Pasteur's experiments, clinical trials for new medicines). Analyse the experimental design, identifying variables, controls, and the importance of these elements in the study's conclusions.  
Challenge: Design experiments with more complex experimental designs (e.g., experiments with multiple levels of the independent variable, factorial designs – conceptually introduced). Discuss the advantages of these more complex designs.

- Students research famous controlled experiments in science history (e.g., Pasteur's experiments, clinical trials for new medicines). Analyse the experimental design, identifying variables, controls, and the importance of these elements in the study's conclusions.

- Challenge: Design experiments with more complex experimental designs (e.g., experiments with multiple levels of the independent variable, factorial designs – conceptually introduced). Discuss the advantages of these more complex designs.

- Lesson Materials:  
  
Slide Deck: Introduction to experimental design, definitions of independent, dependent, controlled variables, control groups, experimental groups, examples of controlled experiments.  
Handout: "Variable Identification Challenge" worksheet with experiment descriptions, "Control Group Design Challenge" worksheet with scientific questions, checklist for designing controlled experiments.  
Formative Quiz (printable or digital).

- Slide Deck: Introduction to experimental design, definitions of independent, dependent, controlled variables, control groups, experimental groups, examples of controlled experiments.

- Handout: "Variable Identification Challenge" worksheet with experiment descriptions, "Control Group Design Challenge" worksheet with scientific questions, checklist for designing controlled experiments.

- Formative Quiz (printable or digital).

- Students will be able to define and differentiate between independent, dependent, and controlled variables in an experiment.

- Students will be able to explain the purpose of control groups and experimental groups in controlled experiments.

- Students will be able to design basic controlled experiments to investigate scientific questions, identifying variables and controls.

- NSW Curriculum Outcomes: SC4-WS-02 (Planning investigations), ACSIS125 (Conducting investigations safely).

- Warm-Up (10 mins): "Variable Scenarios" - Present several simple experimental scenarios (verbally or on slides - e.g., plant growth experiment, reaction rate experiment, survey on sleep habits). For each scenario, ask students: "What factor are we changing or manipulating in this experiment? What factor are we measuring or observing to see if it changes? What factors should we keep the same to make it a fair test?" Lead to the concepts of independent, dependent, and controlled variables.

- Main Activity 1: "Variables Unpacked: Independent, Dependent, Controlled" (25 mins):  
  
Introduce and define:  
  
Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").  
Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").  
Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.  
  
Use real-world examples and analogies to explain each variable type (e.g., in a recipe, changing oven temperature (independent) and measuring cake rise (dependent), while keeping ingredients and baking time constant (controlled)).  
"Variable Identification Challenge": Provide students with descriptions of different experiments (from various science topics). For each experiment, students identify the independent variable, dependent variable, and list several important controlled variables. Discuss as a class and clarify variable types.

- Introduce and define:  
  
Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").  
Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").  
Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.

- Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").

- Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").

- Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.

- Use real-world examples and analogies to explain each variable type (e.g., in a recipe, changing oven temperature (independent) and measuring cake rise (dependent), while keeping ingredients and baking time constant (controlled)).

- "Variable Identification Challenge": Provide students with descriptions of different experiments (from various science topics). For each experiment, students identify the independent variable, dependent variable, and list several important controlled variables. Discuss as a class and clarify variable types.

- Main Activity 2: "Control Groups and Experimental Groups - The Power of Comparison" (20 mins):  
  
Explain the purpose of control groups and experimental groups in controlled experiments:  
  
Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).  
Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.  
  
Explain why control groups are essential for drawing valid conclusions in experiments: To isolate the effect of the independent variable and rule out other factors.  
"Control Group Design Challenge": Present scientific questions that require controlled experiments (e.g., "Does a new fertilizer increase plant growth?", "Does a new drug reduce reaction time?"). For each question, students design a basic controlled experiment, specifying:  
  
Independent Variable and how it will be manipulated.  
Dependent Variable and how it will be measured.  
Controlled Variables (list at least 3).  
Experimental Group(s) and Control Group(s) description.  
  
Groups share their experimental designs. Class feedback and discussion – peer review of experimental designs, focusing on clarity of variables and effective use of control groups.

- Explain the purpose of control groups and experimental groups in controlled experiments:  
  
Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).  
Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.

- Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).

- Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.

- Explain why control groups are essential for drawing valid conclusions in experiments: To isolate the effect of the independent variable and rule out other factors.

- "Control Group Design Challenge": Present scientific questions that require controlled experiments (e.g., "Does a new fertilizer increase plant growth?", "Does a new drug reduce reaction time?"). For each question, students design a basic controlled experiment, specifying:  
  
Independent Variable and how it will be manipulated.  
Dependent Variable and how it will be measured.  
Controlled Variables (list at least 3).  
Experimental Group(s) and Control Group(s) description.

- Independent Variable and how it will be manipulated.

- Dependent Variable and how it will be measured.

- Controlled Variables (list at least 3).

- Experimental Group(s) and Control Group(s) description.

- Groups share their experimental designs. Class feedback and discussion – peer review of experimental designs, focusing on clarity of variables and effective use of control groups.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are variables and controls so important in experimental design?  
What happens if you don't control variables properly in an experiment? What kind of conclusions can you draw?  
Why is a control group essential for a valid controlled experiment?  
What are some real-world examples where controlled experiments are used to test new ideas or products?

- Why are variables and controls so important in experimental design?

- What happens if you don't control variables properly in an experiment? What kind of conclusions can you draw?

- Why is a control group essential for a valid controlled experiment?

- What are some real-world examples where controlled experiments are used to test new ideas or products?

- Assessments:  
  
Formative Task (5 mins): "Variable Matching Quiz" - Matching exercise: Students match variable types (independent, dependent, controlled) to their definitions. AI-generated quiz.  
Review of group-designed experiments from "Control Group Design Challenge" – assess clarity of variable identification, effective use of control groups, and feasibility of design.

- Formative Task (5 mins): "Variable Matching Quiz" - Matching exercise: Students match variable types (independent, dependent, controlled) to their definitions. AI-generated quiz.

- Review of group-designed experiments from "Control Group Design Challenge" – assess clarity of variable identification, effective use of control groups, and feasibility of design.

- Student Workbook:  
  
Definitions of independent, dependent, and controlled variables with examples.  
Explanation of control groups and experimental groups and their purpose.  
"Variable Identification Challenge" worksheet with experiment descriptions.  
"Control Group Design Challenge" worksheet with scientific questions and prompts for experimental design.  
Checklist for designing a controlled experiment (variables, controls, groups).

- Definitions of independent, dependent, and controlled variables with examples.

- Explanation of control groups and experimental groups and their purpose.

- "Variable Identification Challenge" worksheet with experiment descriptions.

- "Control Group Design Challenge" worksheet with scientific questions and prompts for experimental design.

- Checklist for designing a controlled experiment (variables, controls, groups).

- Extension Activities:  
  
Students research famous controlled experiments in science history (e.g., Pasteur's experiments, clinical trials for new medicines). Analyse the experimental design, identifying variables, controls, and the importance of these elements in the study's conclusions.  
Challenge: Design experiments with more complex experimental designs (e.g., experiments with multiple levels of the independent variable, factorial designs – conceptually introduced). Discuss the advantages of these more complex designs.

- Students research famous controlled experiments in science history (e.g., Pasteur's experiments, clinical trials for new medicines). Analyse the experimental design, identifying variables, controls, and the importance of these elements in the study's conclusions.

- Challenge: Design experiments with more complex experimental designs (e.g., experiments with multiple levels of the independent variable, factorial designs – conceptually introduced). Discuss the advantages of these more complex designs.

- Introduce and define:  
  
Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").  
Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").  
Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.

- Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").

- Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").

- Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.

- Use real-world examples and analogies to explain each variable type (e.g., in a recipe, changing oven temperature (independent) and measuring cake rise (dependent), while keeping ingredients and baking time constant (controlled)).

- "Variable Identification Challenge": Provide students with descriptions of different experiments (from various science topics). For each experiment, students identify the independent variable, dependent variable, and list several important controlled variables. Discuss as a class and clarify variable types.

- Independent Variable: The factor that is deliberately changed or manipulated by the experimenter (the "cause").

- Dependent Variable: The factor that is measured or observed to see if it is affected by the independent variable (the "effect").

- Controlled Variables (Constants): Factors that are kept constant throughout the experiment to ensure a fair test and isolate the effect of the independent variable.

- Explain the purpose of control groups and experimental groups in controlled experiments:  
  
Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).  
Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.

- Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).

- Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.

- Explain why control groups are essential for drawing valid conclusions in experiments: To isolate the effect of the independent variable and rule out other factors.

- "Control Group Design Challenge": Present scientific questions that require controlled experiments (e.g., "Does a new fertilizer increase plant growth?", "Does a new drug reduce reaction time?"). For each question, students design a basic controlled experiment, specifying:  
  
Independent Variable and how it will be manipulated.  
Dependent Variable and how it will be measured.  
Controlled Variables (list at least 3).  
Experimental Group(s) and Control Group(s) description.

- Independent Variable and how it will be manipulated.

- Dependent Variable and how it will be measured.

- Controlled Variables (list at least 3).

- Experimental Group(s) and Control Group(s) description.

- Groups share their experimental designs. Class feedback and discussion – peer review of experimental designs, focusing on clarity of variables and effective use of control groups.

- Experimental Group(s): Group(s) that receive the treatment or manipulation (the independent variable is changed).

- Control Group: A group that does not receive the treatment or manipulation (independent variable is kept at a standard or baseline level). Serves as a baseline for comparison.

- Independent Variable and how it will be manipulated.

- Dependent Variable and how it will be measured.

- Controlled Variables (list at least 3).

- Experimental Group(s) and Control Group(s) description.

- Why are variables and controls so important in experimental design?

- What happens if you don't control variables properly in an experiment? What kind of conclusions can you draw?

- Why is a control group essential for a valid controlled experiment?

- What are some real-world examples where controlled experiments are used to test new ideas or products?

- Formative Task (5 mins): "Variable Matching Quiz" - Matching exercise: Students match variable types (independent, dependent, controlled) to their definitions. AI-generated quiz.

- Review of group-designed experiments from "Control Group Design Challenge" – assess clarity of variable identification, effective use of control groups, and feasibility of design.

- Definitions of independent, dependent, and controlled variables with examples.

- Explanation of control groups and experimental groups and their purpose.

- "Variable Identification Challenge" worksheet with experiment descriptions.

- "Control Group Design Challenge" worksheet with scientific questions and prompts for experimental design.

- Checklist for designing a controlled experiment (variables, controls, groups).

- Students research famous controlled experiments in science history (e.g., Pasteur's experiments, clinical trials for new medicines). Analyse the experimental design, identifying variables, controls, and the importance of these elements in the study's conclusions.

- Challenge: Design experiments with more complex experimental designs (e.g., experiments with multiple levels of the independent variable, factorial designs – conceptually introduced). Discuss the advantages of these more complex designs.

- Slide Deck: Introduction to experimental design, definitions of independent, dependent, controlled variables, control groups, experimental groups, examples of controlled experiments.

- Handout: "Variable Identification Challenge" worksheet with experiment descriptions, "Control Group Design Challenge" worksheet with scientific questions, checklist for designing controlled experiments.

- Formative Quiz (printable or digital).

### Lesson 5.3: Designing Virtual Experiments in Observable

- Title: Virtual Labs: Conducting Experiments in Observable Notebooks

- Learning Outcomes:  
  
Students will be able to design a simple virtual experiment within an Observable notebook environment.  
Students will be able to define and manipulate variables (independent and dependent) using Python code in Observable to simulate experimental conditions.  
Students will be able to simulate data collection within Observable by generating or inputting data based on their virtual experiment design.  
Students will be able to analyse and interpret simulated data in Observable to draw conclusions about their virtual experiment.  
NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-05 (Using data to develop conclusions), ACSIS125 (Planning and conducting investigations), ACSIS126 (Processing and analysing data).

- Students will be able to design a simple virtual experiment within an Observable notebook environment.

- Students will be able to define and manipulate variables (independent and dependent) using Python code in Observable to simulate experimental conditions.

- Students will be able to simulate data collection within Observable by generating or inputting data based on their virtual experiment design.

- Students will be able to analyse and interpret simulated data in Observable to draw conclusions about their virtual experiment.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-05 (Using data to develop conclusions), ACSIS125 (Planning and conducting investigations), ACSIS126 (Processing and analysing data).

- Overview: This lesson provides a hands-on experience in designing and conducting virtual experiments using Observable notebooks. Students will learn to translate experimental designs into code, manipulate variables, simulate data collection, and analyse the resulting data within the interactive Observable environment.

- Activities:  
  
Warm-Up (10 mins): "Why Virtual Experiments?" - Class discussion: "What are the advantages of using computer simulations or virtual experiments in science? When might virtual experiments be useful or necessary compared to real-world experiments?" (e.g., safety, cost, time, ethical reasons, studying complex systems). Brainstorm examples of scientific areas where simulations are widely used (climate modelling, physics simulations, disease spread models).  
Main Activity 1: "Guided Virtual Experiment Design - Plant Growth Simulation" (25 mins):  
  
Guided, step-by-step activity to design a simple virtual experiment in Observable simulating plant growth under different light conditions.  
Provide a template Observable notebook (partially completed notebook with markdown instructions and code cells).  
Guide students through the steps:  
  
Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).  
Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).  
Design Control: Other factors kept constant (water, soil type, temperature – conceptually).  
Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.  
  
Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})  
  
Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.  
  
  
Students work through the guided virtual experiment in Observable, running code cells, modifying parameters (e.g., changing means, standard deviations in data generation code), and observing how changes in variables affect the simulated data and visualizations.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What did you learn about designing and conducting experiments from this virtual experiment activity?  
How does Observable help in conducting virtual experiments and analysing data? What are the advantages of using Observable for this?  
What are the limitations of this simple virtual experiment? How could you make it more realistic or complex?  
In what other scientific contexts could you use virtual experiments and simulations to investigate questions?  
  
Assessments:  
  
Formative Check (5 mins): Review student Observable notebooks to check for completion of the guided virtual experiment, correct code execution, and basic data interpretation in markdown cells.  
Observation of student engagement during the virtual experiment activity.  
  
Student Workbook:  
  
Step-by-step instructions for the guided virtual experiment (plant growth simulation).  
Observable notebook template (partially completed).  
Code snippets for data generation and visualization (to be copied and adapted by students).  
Prompts for data analysis and interpretation in markdown cells.  
Reflection questions on virtual experiments and Observable.  
  
Extension Activities:  
  
Students design their own virtual experiments in Observable to investigate a scientific question of their choice. Encourage them to choose a question and design that is slightly more complex than the guided example.  
Challenge: Extend the plant growth simulation to include more variables (e.g., different types of fertilizer, different soil pH levels) and more complex interactions between variables. Explore how to model these interactions in code and visualize the multi-variable data.

- Warm-Up (10 mins): "Why Virtual Experiments?" - Class discussion: "What are the advantages of using computer simulations or virtual experiments in science? When might virtual experiments be useful or necessary compared to real-world experiments?" (e.g., safety, cost, time, ethical reasons, studying complex systems). Brainstorm examples of scientific areas where simulations are widely used (climate modelling, physics simulations, disease spread models).

- Main Activity 1: "Guided Virtual Experiment Design - Plant Growth Simulation" (25 mins):  
  
Guided, step-by-step activity to design a simple virtual experiment in Observable simulating plant growth under different light conditions.  
Provide a template Observable notebook (partially completed notebook with markdown instructions and code cells).  
Guide students through the steps:  
  
Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).  
Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).  
Design Control: Other factors kept constant (water, soil type, temperature – conceptually).  
Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.  
  
Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})  
  
Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.  
  
  
Students work through the guided virtual experiment in Observable, running code cells, modifying parameters (e.g., changing means, standard deviations in data generation code), and observing how changes in variables affect the simulated data and visualizations.

- Guided, step-by-step activity to design a simple virtual experiment in Observable simulating plant growth under different light conditions.

- Provide a template Observable notebook (partially completed notebook with markdown instructions and code cells).

- Guide students through the steps:  
  
Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).  
Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).  
Design Control: Other factors kept constant (water, soil type, temperature – conceptually).  
Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.  
  
Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})  
  
Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.

- Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).

- Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).

- Design Control: Other factors kept constant (water, soil type, temperature – conceptually).

- Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.

- Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")

- Store the simulated data in a dictionary or Pandas DataFrame.

- Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})

- Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.

- Describe the trends they observe in the simulated data from the graph and the printed data.

- Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.

- Discuss limitations of the virtual experiment and how it could be improved or extended.

- Students work through the guided virtual experiment in Observable, running code cells, modifying parameters (e.g., changing means, standard deviations in data generation code), and observing how changes in variables affect the simulated data and visualizations.

- Reflection & Discussion (10 mins): Class discussion:  
  
What did you learn about designing and conducting experiments from this virtual experiment activity?  
How does Observable help in conducting virtual experiments and analysing data? What are the advantages of using Observable for this?  
What are the limitations of this simple virtual experiment? How could you make it more realistic or complex?  
In what other scientific contexts could you use virtual experiments and simulations to investigate questions?

- What did you learn about designing and conducting experiments from this virtual experiment activity?

- How does Observable help in conducting virtual experiments and analysing data? What are the advantages of using Observable for this?

- What are the limitations of this simple virtual experiment? How could you make it more realistic or complex?

- In what other scientific contexts could you use virtual experiments and simulations to investigate questions?

- Assessments:  
  
Formative Check (5 mins): Review student Observable notebooks to check for completion of the guided virtual experiment, correct code execution, and basic data interpretation in markdown cells.  
Observation of student engagement during the virtual experiment activity.

- Formative Check (5 mins): Review student Observable notebooks to check for completion of the guided virtual experiment, correct code execution, and basic data interpretation in markdown cells.

- Observation of student engagement during the virtual experiment activity.

- Student Workbook:  
  
Step-by-step instructions for the guided virtual experiment (plant growth simulation).  
Observable notebook template (partially completed).  
Code snippets for data generation and visualization (to be copied and adapted by students).  
Prompts for data analysis and interpretation in markdown cells.  
Reflection questions on virtual experiments and Observable.

- Step-by-step instructions for the guided virtual experiment (plant growth simulation).

- Observable notebook template (partially completed).

- Code snippets for data generation and visualization (to be copied and adapted by students).

- Prompts for data analysis and interpretation in markdown cells.

- Reflection questions on virtual experiments and Observable.

- Extension Activities:  
  
Students design their own virtual experiments in Observable to investigate a scientific question of their choice. Encourage them to choose a question and design that is slightly more complex than the guided example.  
Challenge: Extend the plant growth simulation to include more variables (e.g., different types of fertilizer, different soil pH levels) and more complex interactions between variables. Explore how to model these interactions in code and visualize the multi-variable data.

- Students design their own virtual experiments in Observable to investigate a scientific question of their choice. Encourage them to choose a question and design that is slightly more complex than the guided example.

- Challenge: Extend the plant growth simulation to include more variables (e.g., different types of fertilizer, different soil pH levels) and more complex interactions between variables. Explore how to model these interactions in code and visualize the multi-variable data.

- Lesson Materials:  
  
Slide Deck: Introduction to virtual experiments, advantages of simulations, step-by-step guide to the plant growth virtual experiment in Observable, code examples for data generation and visualization in Observable.  
Handout: Step-by-step instructions for the virtual experiment, Observable notebook template link, code snippets to copy and adapt.  
Computers/Tablets with internet access for Observable.

- Slide Deck: Introduction to virtual experiments, advantages of simulations, step-by-step guide to the plant growth virtual experiment in Observable, code examples for data generation and visualization in Observable.

- Handout: Step-by-step instructions for the virtual experiment, Observable notebook template link, code snippets to copy and adapt.

- Computers/Tablets with internet access for Observable.

- Students will be able to design a simple virtual experiment within an Observable notebook environment.

- Students will be able to define and manipulate variables (independent and dependent) using Python code in Observable to simulate experimental conditions.

- Students will be able to simulate data collection within Observable by generating or inputting data based on their virtual experiment design.

- Students will be able to analyse and interpret simulated data in Observable to draw conclusions about their virtual experiment.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-05 (Using data to develop conclusions), ACSIS125 (Planning and conducting investigations), ACSIS126 (Processing and analysing data).

- Warm-Up (10 mins): "Why Virtual Experiments?" - Class discussion: "What are the advantages of using computer simulations or virtual experiments in science? When might virtual experiments be useful or necessary compared to real-world experiments?" (e.g., safety, cost, time, ethical reasons, studying complex systems). Brainstorm examples of scientific areas where simulations are widely used (climate modelling, physics simulations, disease spread models).

- Main Activity 1: "Guided Virtual Experiment Design - Plant Growth Simulation" (25 mins):  
  
Guided, step-by-step activity to design a simple virtual experiment in Observable simulating plant growth under different light conditions.  
Provide a template Observable notebook (partially completed notebook with markdown instructions and code cells).  
Guide students through the steps:  
  
Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).  
Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).  
Design Control: Other factors kept constant (water, soil type, temperature – conceptually).  
Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.  
  
Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})  
  
Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.  
  
  
Students work through the guided virtual experiment in Observable, running code cells, modifying parameters (e.g., changing means, standard deviations in data generation code), and observing how changes in variables affect the simulated data and visualizations.

- Guided, step-by-step activity to design a simple virtual experiment in Observable simulating plant growth under different light conditions.

- Provide a template Observable notebook (partially completed notebook with markdown instructions and code cells).

- Guide students through the steps:  
  
Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).  
Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).  
Design Control: Other factors kept constant (water, soil type, temperature – conceptually).  
Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.  
  
Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})  
  
Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.

- Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).

- Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).

- Design Control: Other factors kept constant (water, soil type, temperature – conceptually).

- Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.

- Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")

- Store the simulated data in a dictionary or Pandas DataFrame.

- Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})

- Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.

- Describe the trends they observe in the simulated data from the graph and the printed data.

- Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.

- Discuss limitations of the virtual experiment and how it could be improved or extended.

- Students work through the guided virtual experiment in Observable, running code cells, modifying parameters (e.g., changing means, standard deviations in data generation code), and observing how changes in variables affect the simulated data and visualizations.

- Reflection & Discussion (10 mins): Class discussion:  
  
What did you learn about designing and conducting experiments from this virtual experiment activity?  
How does Observable help in conducting virtual experiments and analysing data? What are the advantages of using Observable for this?  
What are the limitations of this simple virtual experiment? How could you make it more realistic or complex?  
In what other scientific contexts could you use virtual experiments and simulations to investigate questions?

- What did you learn about designing and conducting experiments from this virtual experiment activity?

- How does Observable help in conducting virtual experiments and analysing data? What are the advantages of using Observable for this?

- What are the limitations of this simple virtual experiment? How could you make it more realistic or complex?

- In what other scientific contexts could you use virtual experiments and simulations to investigate questions?

- Assessments:  
  
Formative Check (5 mins): Review student Observable notebooks to check for completion of the guided virtual experiment, correct code execution, and basic data interpretation in markdown cells.  
Observation of student engagement during the virtual experiment activity.

- Formative Check (5 mins): Review student Observable notebooks to check for completion of the guided virtual experiment, correct code execution, and basic data interpretation in markdown cells.

- Observation of student engagement during the virtual experiment activity.

- Student Workbook:  
  
Step-by-step instructions for the guided virtual experiment (plant growth simulation).  
Observable notebook template (partially completed).  
Code snippets for data generation and visualization (to be copied and adapted by students).  
Prompts for data analysis and interpretation in markdown cells.  
Reflection questions on virtual experiments and Observable.

- Step-by-step instructions for the guided virtual experiment (plant growth simulation).

- Observable notebook template (partially completed).

- Code snippets for data generation and visualization (to be copied and adapted by students).

- Prompts for data analysis and interpretation in markdown cells.

- Reflection questions on virtual experiments and Observable.

- Extension Activities:  
  
Students design their own virtual experiments in Observable to investigate a scientific question of their choice. Encourage them to choose a question and design that is slightly more complex than the guided example.  
Challenge: Extend the plant growth simulation to include more variables (e.g., different types of fertilizer, different soil pH levels) and more complex interactions between variables. Explore how to model these interactions in code and visualize the multi-variable data.

- Students design their own virtual experiments in Observable to investigate a scientific question of their choice. Encourage them to choose a question and design that is slightly more complex than the guided example.

- Challenge: Extend the plant growth simulation to include more variables (e.g., different types of fertilizer, different soil pH levels) and more complex interactions between variables. Explore how to model these interactions in code and visualize the multi-variable data.

- Guided, step-by-step activity to design a simple virtual experiment in Observable simulating plant growth under different light conditions.

- Provide a template Observable notebook (partially completed notebook with markdown instructions and code cells).

- Guide students through the steps:  
  
Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).  
Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).  
Design Control: Other factors kept constant (water, soil type, temperature – conceptually).  
Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.  
  
Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})  
  
Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.

- Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).

- Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).

- Design Control: Other factors kept constant (water, soil type, temperature – conceptually).

- Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.

- Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")

- Store the simulated data in a dictionary or Pandas DataFrame.

- Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})

- Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.

- Describe the trends they observe in the simulated data from the graph and the printed data.

- Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.

- Discuss limitations of the virtual experiment and how it could be improved or extended.

- Students work through the guided virtual experiment in Observable, running code cells, modifying parameters (e.g., changing means, standard deviations in data generation code), and observing how changes in variables affect the simulated data and visualizations.

- Define the Scientific Question: "Does light intensity affect plant growth?" (or similar, more specific question).

- Identify Variables: Independent Variable (light intensity - categories: Low, Medium, High), Dependent Variable (plant height after 2 weeks - in cm).

- Design Control: Other factors kept constant (water, soil type, temperature – conceptually).

- Simulate Data Collection (using Python code): In a code cell, provide code to:  
  
Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")  
  
Store the simulated data in a dictionary or Pandas DataFrame.

- Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")

- Store the simulated data in a dictionary or Pandas DataFrame.

- Data Visualization: In a code cell, provide code to create a bar graph (or box plot) in Observable using Plot library to visualize the simulated plant heights for each light level and compare them. Example using Plot:  
python  
import {Plot} from "@observablehq/plot"  
Plot.plot({  
 marks: [  
 Plot.barY(  
 light\_levels.flatMap(level => plant\_heights[level].map(height => ({level: level, height: height}))),  
 {x: "level", y: "height", fill: "level", tip: true}  
 )  
 ],  
 x: {label: "Light Intensity"},  
 y: {label: "Plant Height (cm)"},  
 color: {domain: light\_levels}  
})

- Data Analysis and Interpretation: In markdown cells, guide students to:  
  
Describe the trends they observe in the simulated data from the graph and the printed data.  
Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.  
Discuss limitations of the virtual experiment and how it could be improved or extended.

- Describe the trends they observe in the simulated data from the graph and the printed data.

- Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.

- Discuss limitations of the virtual experiment and how it could be improved or extended.

- Generate simulated plant height data for each light intensity level. Use numpy.random to create somewhat variable data to mimic real-world variability (e.g., using numpy.random.normal() with different means for each light level and some standard deviation). Example code snippet:  
python  
import numpy as np  
light\_levels = ["Low", "Medium", "High"]  
plant\_heights = {}  
for level in light\_levels:  
 if level == "Low":  
 plant\_heights[level] = np.random.normal(5, 1.5, 20) # Mean 5cm, SD 1.5cm, 20 samples  
 elif level == "Medium":  
 plant\_heights[level] = np.random.normal(8, 2, 20) # Mean 8cm, SD 2cm, 20 samples  
 else: # High  
 plant\_heights[level] = np.random.normal(12, 2.5, 20) # Mean 12cm, SD 2.5cm, 20 samples  
 print(f"Simulated plant heights for {level} light: {plant\_heights[level]}")

- Store the simulated data in a dictionary or Pandas DataFrame.

- Describe the trends they observe in the simulated data from the graph and the printed data.

- Answer the scientific question: "Does light intensity affect plant growth?" based on the simulated data.

- Discuss limitations of the virtual experiment and how it could be improved or extended.

- What did you learn about designing and conducting experiments from this virtual experiment activity?

- How does Observable help in conducting virtual experiments and analysing data? What are the advantages of using Observable for this?

- What are the limitations of this simple virtual experiment? How could you make it more realistic or complex?

- In what other scientific contexts could you use virtual experiments and simulations to investigate questions?

- Formative Check (5 mins): Review student Observable notebooks to check for completion of the guided virtual experiment, correct code execution, and basic data interpretation in markdown cells.

- Observation of student engagement during the virtual experiment activity.

- Step-by-step instructions for the guided virtual experiment (plant growth simulation).

- Observable notebook template (partially completed).

- Code snippets for data generation and visualization (to be copied and adapted by students).

- Prompts for data analysis and interpretation in markdown cells.

- Reflection questions on virtual experiments and Observable.

- Students design their own virtual experiments in Observable to investigate a scientific question of their choice. Encourage them to choose a question and design that is slightly more complex than the guided example.

- Challenge: Extend the plant growth simulation to include more variables (e.g., different types of fertilizer, different soil pH levels) and more complex interactions between variables. Explore how to model these interactions in code and visualize the multi-variable data.

- Slide Deck: Introduction to virtual experiments, advantages of simulations, step-by-step guide to the plant growth virtual experiment in Observable, code examples for data generation and visualization in Observable.

- Handout: Step-by-step instructions for the virtual experiment, Observable notebook template link, code snippets to copy and adapt.

- Computers/Tablets with internet access for Observable.

### Week 6: Advanced Data Handling (Cleaning, Sorting, Filtering)

### Lesson 6.1: Real-World Data Challenges: "Messy" Datasets and Data Quality Issues

- Title: Taming the Data Jungle: Facing Real-World Data Challenges

- Learning Outcomes:  
  
Students will be able to recognize that real-world datasets are often "messy" and contain various data quality issues.  
Students will be able to identify common data quality problems, including missing values, inconsistent formats, errors, and biases.  
Students will understand the importance of data cleaning and data quality control in ensuring reliable data analysis.  
NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Examine a range of sources of data).

- Students will be able to recognize that real-world datasets are often "messy" and contain various data quality issues.

- Students will be able to identify common data quality problems, including missing values, inconsistent formats, errors, and biases.

- Students will understand the importance of data cleaning and data quality control in ensuring reliable data analysis.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Examine a range of sources of data).

- Overview: This lesson introduces students to the realities of working with real-world data. They will explore common data quality issues, understand why data cleaning is essential, and begin to appreciate the complexities of data preparation for analysis.

- Activities:  
  
Warm-Up (10 mins): "Data Disaster Stories" - Briefly share a few real-world examples of situations where poor data quality led to problems or incorrect conclusions (e.g., a flawed scientific study due to data errors, a business decision based on inaccurate data, a software bug caused by unexpected data input). Discuss: What are the potential consequences of using "messy" data? Why is data quality important?  
Main Activity 1: "The Messy Dataset Exploration" (25 mins):  
  
Provide students with a pre-prepared "messy" dataset (digital file - CSV or similar). The dataset should contain various data quality issues (examples below). The dataset can be related to a scientific topic relevant to Year 8 (e.g., environmental data, survey data, experimental results).  
Guide students to explore the dataset in Observable (using a Pandas DataFrame in a code cell).  
Ask students to investigate and identify different types of data quality issues within the dataset. Guide them to look for:  
  
Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.  
Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.  
Errors and Typos: Point out obvious errors, impossible values, or typos in text data.  
Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.  
Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).  
  
Students document the data quality issues they find in a markdown cell in their Observable notebook, describing the type of issue and providing specific examples from the dataset.  
  
Main Activity 2: "Categorizing Data Quality Issues and Brainstorming Solutions" (20 mins):  
  
Class discussion: Based on the messy dataset exploration, categorize the identified data quality issues into broader categories (e.g., Missingness, Inconsistency, Errors, Bias – simplified categories for Year 8).  
For each category, brainstorm potential reasons why these issues might occur in real-world data collection.  
For each category, brainstorm general strategies or approaches that data scientists use to address these data quality issues (e.g., for missing values: remove, fill in/impute; for inconsistencies: standardize formats; for errors: correct if possible, remove if not; for bias: be aware, try to mitigate if possible, acknowledge limitations). Focus on conceptual solutions at this stage, not specific coding techniques yet.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to be aware of data quality issues before analysing data?  
What are the potential consequences of ignoring data quality problems in scientific research or real-world applications?  
Is it always possible to completely "clean" a messy dataset? What are the trade-offs and limitations?  
How can we design data collection processes to minimize data quality issues from the start? (Preventative measures).  
  
Assessments:  
  
Formative Task (5 mins): "Data Quality Issue Identification Quiz" - Short multiple-choice quiz identifying different types of data quality issues from descriptions. AI-generated quiz.  
Review of student documentation of data quality issues in their Observable notebooks – assess their ability to identify and categorize issues in the messy dataset.  
  
Student Workbook:  
  
Descriptions of common data quality issues (missing values, inconsistent formats, errors, bias – simplified).  
"Messy Dataset Exploration" worksheet with prompts to identify different issue types in a provided dataset.  
Table to categorize data quality issues and brainstorm potential causes and solutions (conceptual strategies).  
Reflection questions on data quality and its importance.  
  
Extension Activities:  
  
Students research real-world case studies where data quality issues led to significant problems or errors in decision-making or scientific conclusions. Present findings to the class.  
Challenge: Find publicly available "messy" datasets online (e.g., from government data portals, Kaggle datasets with data cleaning challenges). Explore these datasets and identify the data quality issues present.

- Warm-Up (10 mins): "Data Disaster Stories" - Briefly share a few real-world examples of situations where poor data quality led to problems or incorrect conclusions (e.g., a flawed scientific study due to data errors, a business decision based on inaccurate data, a software bug caused by unexpected data input). Discuss: What are the potential consequences of using "messy" data? Why is data quality important?

- Main Activity 1: "The Messy Dataset Exploration" (25 mins):  
  
Provide students with a pre-prepared "messy" dataset (digital file - CSV or similar). The dataset should contain various data quality issues (examples below). The dataset can be related to a scientific topic relevant to Year 8 (e.g., environmental data, survey data, experimental results).  
Guide students to explore the dataset in Observable (using a Pandas DataFrame in a code cell).  
Ask students to investigate and identify different types of data quality issues within the dataset. Guide them to look for:  
  
Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.  
Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.  
Errors and Typos: Point out obvious errors, impossible values, or typos in text data.  
Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.  
Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).  
  
Students document the data quality issues they find in a markdown cell in their Observable notebook, describing the type of issue and providing specific examples from the dataset.

- Provide students with a pre-prepared "messy" dataset (digital file - CSV or similar). The dataset should contain various data quality issues (examples below). The dataset can be related to a scientific topic relevant to Year 8 (e.g., environmental data, survey data, experimental results).

- Guide students to explore the dataset in Observable (using a Pandas DataFrame in a code cell).

- Ask students to investigate and identify different types of data quality issues within the dataset. Guide them to look for:  
  
Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.  
Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.  
Errors and Typos: Point out obvious errors, impossible values, or typos in text data.  
Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.  
Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).

- Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.

- Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.

- Errors and Typos: Point out obvious errors, impossible values, or typos in text data.

- Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.

- Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).

- Students document the data quality issues they find in a markdown cell in their Observable notebook, describing the type of issue and providing specific examples from the dataset.

- Main Activity 2: "Categorizing Data Quality Issues and Brainstorming Solutions" (20 mins):  
  
Class discussion: Based on the messy dataset exploration, categorize the identified data quality issues into broader categories (e.g., Missingness, Inconsistency, Errors, Bias – simplified categories for Year 8).  
For each category, brainstorm potential reasons why these issues might occur in real-world data collection.  
For each category, brainstorm general strategies or approaches that data scientists use to address these data quality issues (e.g., for missing values: remove, fill in/impute; for inconsistencies: standardize formats; for errors: correct if possible, remove if not; for bias: be aware, try to mitigate if possible, acknowledge limitations). Focus on conceptual solutions at this stage, not specific coding techniques yet.

- Class discussion: Based on the messy dataset exploration, categorize the identified data quality issues into broader categories (e.g., Missingness, Inconsistency, Errors, Bias – simplified categories for Year 8).

- For each category, brainstorm potential reasons why these issues might occur in real-world data collection.

- For each category, brainstorm general strategies or approaches that data scientists use to address these data quality issues (e.g., for missing values: remove, fill in/impute; for inconsistencies: standardize formats; for errors: correct if possible, remove if not; for bias: be aware, try to mitigate if possible, acknowledge limitations). Focus on conceptual solutions at this stage, not specific coding techniques yet.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to be aware of data quality issues before analysing data?  
What are the potential consequences of ignoring data quality problems in scientific research or real-world applications?  
Is it always possible to completely "clean" a messy dataset? What are the trade-offs and limitations?  
How can we design data collection processes to minimize data quality issues from the start? (Preventative measures).

- Why is it important to be aware of data quality issues before analysing data?

- What are the potential consequences of ignoring data quality problems in scientific research or real-world applications?

- Is it always possible to completely "clean" a messy dataset? What are the trade-offs and limitations?

- How can we design data collection processes to minimize data quality issues from the start? (Preventative measures).

- Assessments:  
  
Formative Task (5 mins): "Data Quality Issue Identification Quiz" - Short multiple-choice quiz identifying different types of data quality issues from descriptions. AI-generated quiz.  
Review of student documentation of data quality issues in their Observable notebooks – assess their ability to identify and categorize issues in the messy dataset.

- Formative Task (5 mins): "Data Quality Issue Identification Quiz" - Short multiple-choice quiz identifying different types of data quality issues from descriptions. AI-generated quiz.

- Review of student documentation of data quality issues in their Observable notebooks – assess their ability to identify and categorize issues in the messy dataset.

- Student Workbook:  
  
Descriptions of common data quality issues (missing values, inconsistent formats, errors, bias – simplified).  
"Messy Dataset Exploration" worksheet with prompts to identify different issue types in a provided dataset.  
Table to categorize data quality issues and brainstorm potential causes and solutions (conceptual strategies).  
Reflection questions on data quality and its importance.

- Descriptions of common data quality issues (missing values, inconsistent formats, errors, bias – simplified).

- "Messy Dataset Exploration" worksheet with prompts to identify different issue types in a provided dataset.

- Table to categorize data quality issues and brainstorm potential causes and solutions (conceptual strategies).

- Reflection questions on data quality and its importance.

- Extension Activities:  
  
Students research real-world case studies where data quality issues led to significant problems or errors in decision-making or scientific conclusions. Present findings to the class.  
Challenge: Find publicly available "messy" datasets online (e.g., from government data portals, Kaggle datasets with data cleaning challenges). Explore these datasets and identify the data quality issues present.

- Students research real-world case studies where data quality issues led to significant problems or errors in decision-making or scientific conclusions. Present findings to the class.

- Challenge: Find publicly available "messy" datasets online (e.g., from government data portals, Kaggle datasets with data cleaning challenges). Explore these datasets and identify the data quality issues present.

- Lesson Materials:  
  
Slide Deck: Introduction to real-world data challenges, definition of "messy" data, common data quality issues (missing values, inconsistencies, errors, bias), importance of data cleaning.  
Handout: "Messy Dataset Exploration" worksheet with prompts, table for categorizing issues and brainstorming solutions, descriptions of data quality issues.  
Formative Quiz (printable or digital).  
Digital "messy" dataset file (CSV or similar) for student exploration in Observable.  
Computers/Tablets with internet access for Observable.

- Slide Deck: Introduction to real-world data challenges, definition of "messy" data, common data quality issues (missing values, inconsistencies, errors, bias), importance of data cleaning.

- Handout: "Messy Dataset Exploration" worksheet with prompts, table for categorizing issues and brainstorming solutions, descriptions of data quality issues.

- Formative Quiz (printable or digital).

- Digital "messy" dataset file (CSV or similar) for student exploration in Observable.

- Computers/Tablets with internet access for Observable.

- Students will be able to recognize that real-world datasets are often "messy" and contain various data quality issues.

- Students will be able to identify common data quality problems, including missing values, inconsistent formats, errors, and biases.

- Students will understand the importance of data cleaning and data quality control in ensuring reliable data analysis.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), SC4-DA1-01 (Examine a range of sources of data).

- Warm-Up (10 mins): "Data Disaster Stories" - Briefly share a few real-world examples of situations where poor data quality led to problems or incorrect conclusions (e.g., a flawed scientific study due to data errors, a business decision based on inaccurate data, a software bug caused by unexpected data input). Discuss: What are the potential consequences of using "messy" data? Why is data quality important?

- Main Activity 1: "The Messy Dataset Exploration" (25 mins):  
  
Provide students with a pre-prepared "messy" dataset (digital file - CSV or similar). The dataset should contain various data quality issues (examples below). The dataset can be related to a scientific topic relevant to Year 8 (e.g., environmental data, survey data, experimental results).  
Guide students to explore the dataset in Observable (using a Pandas DataFrame in a code cell).  
Ask students to investigate and identify different types of data quality issues within the dataset. Guide them to look for:  
  
Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.  
Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.  
Errors and Typos: Point out obvious errors, impossible values, or typos in text data.  
Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.  
Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).  
  
Students document the data quality issues they find in a markdown cell in their Observable notebook, describing the type of issue and providing specific examples from the dataset.

- Provide students with a pre-prepared "messy" dataset (digital file - CSV or similar). The dataset should contain various data quality issues (examples below). The dataset can be related to a scientific topic relevant to Year 8 (e.g., environmental data, survey data, experimental results).

- Guide students to explore the dataset in Observable (using a Pandas DataFrame in a code cell).

- Ask students to investigate and identify different types of data quality issues within the dataset. Guide them to look for:  
  
Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.  
Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.  
Errors and Typos: Point out obvious errors, impossible values, or typos in text data.  
Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.  
Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).

- Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.

- Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.

- Errors and Typos: Point out obvious errors, impossible values, or typos in text data.

- Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.

- Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).

- Students document the data quality issues they find in a markdown cell in their Observable notebook, describing the type of issue and providing specific examples from the dataset.

- Main Activity 2: "Categorizing Data Quality Issues and Brainstorming Solutions" (20 mins):  
  
Class discussion: Based on the messy dataset exploration, categorize the identified data quality issues into broader categories (e.g., Missingness, Inconsistency, Errors, Bias – simplified categories for Year 8).  
For each category, brainstorm potential reasons why these issues might occur in real-world data collection.  
For each category, brainstorm general strategies or approaches that data scientists use to address these data quality issues (e.g., for missing values: remove, fill in/impute; for inconsistencies: standardize formats; for errors: correct if possible, remove if not; for bias: be aware, try to mitigate if possible, acknowledge limitations). Focus on conceptual solutions at this stage, not specific coding techniques yet.

- Class discussion: Based on the messy dataset exploration, categorize the identified data quality issues into broader categories (e.g., Missingness, Inconsistency, Errors, Bias – simplified categories for Year 8).

- For each category, brainstorm potential reasons why these issues might occur in real-world data collection.

- For each category, brainstorm general strategies or approaches that data scientists use to address these data quality issues (e.g., for missing values: remove, fill in/impute; for inconsistencies: standardize formats; for errors: correct if possible, remove if not; for bias: be aware, try to mitigate if possible, acknowledge limitations). Focus on conceptual solutions at this stage, not specific coding techniques yet.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is it important to be aware of data quality issues before analysing data?  
What are the potential consequences of ignoring data quality problems in scientific research or real-world applications?  
Is it always possible to completely "clean" a messy dataset? What are the trade-offs and limitations?  
How can we design data collection processes to minimize data quality issues from the start? (Preventative measures).

- Why is it important to be aware of data quality issues before analysing data?

- What are the potential consequences of ignoring data quality problems in scientific research or real-world applications?

- Is it always possible to completely "clean" a messy dataset? What are the trade-offs and limitations?

- How can we design data collection processes to minimize data quality issues from the start? (Preventative measures).

- Assessments:  
  
Formative Task (5 mins): "Data Quality Issue Identification Quiz" - Short multiple-choice quiz identifying different types of data quality issues from descriptions. AI-generated quiz.  
Review of student documentation of data quality issues in their Observable notebooks – assess their ability to identify and categorize issues in the messy dataset.

- Formative Task (5 mins): "Data Quality Issue Identification Quiz" - Short multiple-choice quiz identifying different types of data quality issues from descriptions. AI-generated quiz.

- Review of student documentation of data quality issues in their Observable notebooks – assess their ability to identify and categorize issues in the messy dataset.

- Student Workbook:  
  
Descriptions of common data quality issues (missing values, inconsistent formats, errors, bias – simplified).  
"Messy Dataset Exploration" worksheet with prompts to identify different issue types in a provided dataset.  
Table to categorize data quality issues and brainstorm potential causes and solutions (conceptual strategies).  
Reflection questions on data quality and its importance.

- Descriptions of common data quality issues (missing values, inconsistent formats, errors, bias – simplified).

- "Messy Dataset Exploration" worksheet with prompts to identify different issue types in a provided dataset.

- Table to categorize data quality issues and brainstorm potential causes and solutions (conceptual strategies).

- Reflection questions on data quality and its importance.

- Extension Activities:  
  
Students research real-world case studies where data quality issues led to significant problems or errors in decision-making or scientific conclusions. Present findings to the class.  
Challenge: Find publicly available "messy" datasets online (e.g., from government data portals, Kaggle datasets with data cleaning challenges). Explore these datasets and identify the data quality issues present.

- Students research real-world case studies where data quality issues led to significant problems or errors in decision-making or scientific conclusions. Present findings to the class.

- Challenge: Find publicly available "messy" datasets online (e.g., from government data portals, Kaggle datasets with data cleaning challenges). Explore these datasets and identify the data quality issues present.

- Provide students with a pre-prepared "messy" dataset (digital file - CSV or similar). The dataset should contain various data quality issues (examples below). The dataset can be related to a scientific topic relevant to Year 8 (e.g., environmental data, survey data, experimental results).

- Guide students to explore the dataset in Observable (using a Pandas DataFrame in a code cell).

- Ask students to investigate and identify different types of data quality issues within the dataset. Guide them to look for:  
  
Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.  
Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.  
Errors and Typos: Point out obvious errors, impossible values, or typos in text data.  
Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.  
Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).

- Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.

- Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.

- Errors and Typos: Point out obvious errors, impossible values, or typos in text data.

- Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.

- Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).

- Students document the data quality issues they find in a markdown cell in their Observable notebook, describing the type of issue and providing specific examples from the dataset.

- Missing Values (NaN, empty cells): Use pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count missing values per column.

- Inconsistent Formats (e.g., dates, text case): Show examples of inconsistent date formats (e.g., MM/DD/YYYY, DD-MM-YYYY), inconsistent text casing (e.g., "Yes", "yes", "YES"). Use pandas.DataFrame.info() to check data types.

- Errors and Typos: Point out obvious errors, impossible values, or typos in text data.

- Outliers (Review from Week 4): Remind students about outliers and how they might be data quality issues or genuine extreme values.

- Potential Biases (Introduce Conceptually): Briefly introduce the idea that data can be biased in how it is collected or represented. (More detail on bias in later lessons).

- Class discussion: Based on the messy dataset exploration, categorize the identified data quality issues into broader categories (e.g., Missingness, Inconsistency, Errors, Bias – simplified categories for Year 8).

- For each category, brainstorm potential reasons why these issues might occur in real-world data collection.

- For each category, brainstorm general strategies or approaches that data scientists use to address these data quality issues (e.g., for missing values: remove, fill in/impute; for inconsistencies: standardize formats; for errors: correct if possible, remove if not; for bias: be aware, try to mitigate if possible, acknowledge limitations). Focus on conceptual solutions at this stage, not specific coding techniques yet.

- Why is it important to be aware of data quality issues before analysing data?

- What are the potential consequences of ignoring data quality problems in scientific research or real-world applications?

- Is it always possible to completely "clean" a messy dataset? What are the trade-offs and limitations?

- How can we design data collection processes to minimize data quality issues from the start? (Preventative measures).

- Formative Task (5 mins): "Data Quality Issue Identification Quiz" - Short multiple-choice quiz identifying different types of data quality issues from descriptions. AI-generated quiz.

- Review of student documentation of data quality issues in their Observable notebooks – assess their ability to identify and categorize issues in the messy dataset.

- Descriptions of common data quality issues (missing values, inconsistent formats, errors, bias – simplified).

- "Messy Dataset Exploration" worksheet with prompts to identify different issue types in a provided dataset.

- Table to categorize data quality issues and brainstorm potential causes and solutions (conceptual strategies).

- Reflection questions on data quality and its importance.

- Students research real-world case studies where data quality issues led to significant problems or errors in decision-making or scientific conclusions. Present findings to the class.

- Challenge: Find publicly available "messy" datasets online (e.g., from government data portals, Kaggle datasets with data cleaning challenges). Explore these datasets and identify the data quality issues present.

- Slide Deck: Introduction to real-world data challenges, definition of "messy" data, common data quality issues (missing values, inconsistencies, errors, bias), importance of data cleaning.

- Handout: "Messy Dataset Exploration" worksheet with prompts, table for categorizing issues and brainstorming solutions, descriptions of data quality issues.

- Formative Quiz (printable or digital).

- Digital "messy" dataset file (CSV or similar) for student exploration in Observable.

- Computers/Tablets with internet access for Observable.

### Lesson 6.2: Data Cleaning Techniques in Python (Pandas - Basic)

- Title: Data Spa Treatment: Cleaning Data with Python and Pandas

- Learning Outcomes:  
  
Students will be able to use basic Python and Pandas techniques in Observable notebooks to clean datasets.  
Students will be able to handle missing values in Pandas DataFrames using methods like dropna() and fillna().  
Students will be able to standardize data formats (e.g., converting data types, standardizing text case) using Pandas.  
Students will be able to apply basic data validation techniques to identify and correct errors in datasets using Pandas.  
NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information).

- Students will be able to use basic Python and Pandas techniques in Observable notebooks to clean datasets.

- Students will be able to handle missing values in Pandas DataFrames using methods like dropna() and fillna().

- Students will be able to standardize data formats (e.g., converting data types, standardizing text case) using Pandas.

- Students will be able to apply basic data validation techniques to identify and correct errors in datasets using Pandas.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information).

- Overview: This hands-on lesson introduces basic data cleaning techniques using Python and the Pandas library within Observable. Students will learn practical coding skills to handle missing values, standardize formats, and correct simple errors in datasets, building upon the data quality issues explored in the previous lesson.

- Activities:  
  
Warm-Up (10 mins): "Data Cleaning Scenarios" - Present a few short scenarios describing data quality issues in a dataset (e.g., "A column has missing values", "Date formats are inconsistent", "Text data has mixed casing"). For each scenario, ask students: "What is the data quality issue? What is a possible strategy to address this issue?" Brief class discussion to review conceptual solutions from the previous lesson.  
Main Activity 1: "Handling Missing Values with Pandas" (25 mins):  
  
Guided code-along in Observable:  
  
Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.  
Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.  
Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).  
Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.  
  
Student practice: Provide students with datasets containing missing values in Observable. Students write Python code in their notebooks to:  
  
Detect and count missing values.  
Experiment with both dropna() and fillna() to handle the missing values.  
Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.  
  
  
Main Activity 2: "Standardizing Data Formats with Pandas" (20 mins):  
  
Guided code-along in Observable:  
  
Continue using the same dataset (or load a new one with format inconsistencies).  
Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.  
  
  
Student practice: Provide datasets with format inconsistencies in Observable. Students write Python code to:  
  
Identify columns with incorrect or inconsistent data types.  
Convert columns to appropriate data types using astype() and to\_datetime().  
Standardize text casing in text columns using .str.lower() or .str.upper().  
  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are the advantages of using Python and Pandas for data cleaning tasks?  
What are some key considerations when deciding how to handle missing values (dropping vs. filling)?  
Why is standardizing data formats important for data analysis?  
What are some other common data cleaning tasks that data scientists often perform (beyond what we covered today)? (Briefly mention data validation, handling duplicates, more complex text cleaning).  
  
Assessments:  
  
Formative Task (5 mins): "Data Cleaning Technique Matching" - Matching exercise: Students match data cleaning techniques (e.g., dropna(), fillna(), astype()) to their descriptions or purposes. AI-generated quiz.  
Review of student Python code in Observable notebooks – check for correct implementation of data cleaning techniques and code syntax.  
  
Student Workbook:  
  
Code examples and explanations for Pandas data cleaning techniques: dropna(), fillna(), astype(), to\_datetime(), .str.lower(), .str.upper().  
Observable notebook exercises with "messy" datasets and coding prompts for data cleaning tasks (handling missing values, standardizing formats).  
Code templates for data cleaning operations in Pandas.  
"Choosing Data Cleaning Methods" scenario worksheet with prompts to decide appropriate cleaning techniques for different data issues.  
  
Extension Activities:  
  
Students explore more advanced Pandas data cleaning functions (e.g., replace(), apply(), more sophisticated imputation techniques).  
Challenge: Design a more comprehensive data cleaning pipeline in Observable for a real-world "messy" dataset, combining multiple data cleaning techniques to prepare the data for analysis. Document the steps and justify their choices.

- Warm-Up (10 mins): "Data Cleaning Scenarios" - Present a few short scenarios describing data quality issues in a dataset (e.g., "A column has missing values", "Date formats are inconsistent", "Text data has mixed casing"). For each scenario, ask students: "What is the data quality issue? What is a possible strategy to address this issue?" Brief class discussion to review conceptual solutions from the previous lesson.

- Main Activity 1: "Handling Missing Values with Pandas" (25 mins):  
  
Guided code-along in Observable:  
  
Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.  
Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.  
Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).  
Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.  
  
Student practice: Provide students with datasets containing missing values in Observable. Students write Python code in their notebooks to:  
  
Detect and count missing values.  
Experiment with both dropna() and fillna() to handle the missing values.  
Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.

- Guided code-along in Observable:  
  
Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.  
Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.  
Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).  
Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.

- Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.

- Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.

- Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).

- Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.

- Student practice: Provide students with datasets containing missing values in Observable. Students write Python code in their notebooks to:  
  
Detect and count missing values.  
Experiment with both dropna() and fillna() to handle the missing values.  
Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.

- Detect and count missing values.

- Experiment with both dropna() and fillna() to handle the missing values.

- Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.

- Main Activity 2: "Standardizing Data Formats with Pandas" (20 mins):  
  
Guided code-along in Observable:  
  
Continue using the same dataset (or load a new one with format inconsistencies).  
Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.  
  
  
Student practice: Provide datasets with format inconsistencies in Observable. Students write Python code to:  
  
Identify columns with incorrect or inconsistent data types.  
Convert columns to appropriate data types using astype() and to\_datetime().  
Standardize text casing in text columns using .str.lower() or .str.upper().

- Guided code-along in Observable:  
  
Continue using the same dataset (or load a new one with format inconsistencies).  
Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Continue using the same dataset (or load a new one with format inconsistencies).

- Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).

- Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Student practice: Provide datasets with format inconsistencies in Observable. Students write Python code to:  
  
Identify columns with incorrect or inconsistent data types.  
Convert columns to appropriate data types using astype() and to\_datetime().  
Standardize text casing in text columns using .str.lower() or .str.upper().

- Identify columns with incorrect or inconsistent data types.

- Convert columns to appropriate data types using astype() and to\_datetime().

- Standardize text casing in text columns using .str.lower() or .str.upper().

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the advantages of using Python and Pandas for data cleaning tasks?  
What are some key considerations when deciding how to handle missing values (dropping vs. filling)?  
Why is standardizing data formats important for data analysis?  
What are some other common data cleaning tasks that data scientists often perform (beyond what we covered today)? (Briefly mention data validation, handling duplicates, more complex text cleaning).

- What are the advantages of using Python and Pandas for data cleaning tasks?

- What are some key considerations when deciding how to handle missing values (dropping vs. filling)?

- Why is standardizing data formats important for data analysis?

- What are some other common data cleaning tasks that data scientists often perform (beyond what we covered today)? (Briefly mention data validation, handling duplicates, more complex text cleaning).

- Assessments:  
  
Formative Task (5 mins): "Data Cleaning Technique Matching" - Matching exercise: Students match data cleaning techniques (e.g., dropna(), fillna(), astype()) to their descriptions or purposes. AI-generated quiz.  
Review of student Python code in Observable notebooks – check for correct implementation of data cleaning techniques and code syntax.

- Formative Task (5 mins): "Data Cleaning Technique Matching" - Matching exercise: Students match data cleaning techniques (e.g., dropna(), fillna(), astype()) to their descriptions or purposes. AI-generated quiz.

- Review of student Python code in Observable notebooks – check for correct implementation of data cleaning techniques and code syntax.

- Student Workbook:  
  
Code examples and explanations for Pandas data cleaning techniques: dropna(), fillna(), astype(), to\_datetime(), .str.lower(), .str.upper().  
Observable notebook exercises with "messy" datasets and coding prompts for data cleaning tasks (handling missing values, standardizing formats).  
Code templates for data cleaning operations in Pandas.  
"Choosing Data Cleaning Methods" scenario worksheet with prompts to decide appropriate cleaning techniques for different data issues.

- Code examples and explanations for Pandas data cleaning techniques: dropna(), fillna(), astype(), to\_datetime(), .str.lower(), .str.upper().

- Observable notebook exercises with "messy" datasets and coding prompts for data cleaning tasks (handling missing values, standardizing formats).

- Code templates for data cleaning operations in Pandas.

- "Choosing Data Cleaning Methods" scenario worksheet with prompts to decide appropriate cleaning techniques for different data issues.

- Extension Activities:  
  
Students explore more advanced Pandas data cleaning functions (e.g., replace(), apply(), more sophisticated imputation techniques).  
Challenge: Design a more comprehensive data cleaning pipeline in Observable for a real-world "messy" dataset, combining multiple data cleaning techniques to prepare the data for analysis. Document the steps and justify their choices.

- Students explore more advanced Pandas data cleaning functions (e.g., replace(), apply(), more sophisticated imputation techniques).

- Challenge: Design a more comprehensive data cleaning pipeline in Observable for a real-world "messy" dataset, combining multiple data cleaning techniques to prepare the data for analysis. Document the steps and justify their choices.

- Lesson Materials:  
  
Slide Deck: Introduction to data cleaning with Python and Pandas, code examples for dropna(), fillna(), astype(), to\_datetime(), .str.lower(), .str.upper(), data cleaning workflow.  
Handout: Observable notebook exercises with "messy" datasets and coding prompts, code templates for data cleaning operations, "Choosing Data Cleaning Methods" scenario worksheet, Pandas cheat sheet (basic cleaning functions).  
Formative Quiz (printable or digital).  
Digital "messy" datasets (CSV or similar) for student practice in Observable.  
Computers/Tablets with internet access for Observable.

- Slide Deck: Introduction to data cleaning with Python and Pandas, code examples for dropna(), fillna(), astype(), to\_datetime(), .str.lower(), .str.upper(), data cleaning workflow.

- Handout: Observable notebook exercises with "messy" datasets and coding prompts, code templates for data cleaning operations, "Choosing Data Cleaning Methods" scenario worksheet, Pandas cheat sheet (basic cleaning functions).

- Formative Quiz (printable or digital).

- Digital "messy" datasets (CSV or similar) for student practice in Observable.

- Computers/Tablets with internet access for Observable.

- Students will be able to use basic Python and Pandas techniques in Observable notebooks to clean datasets.

- Students will be able to handle missing values in Pandas DataFrames using methods like dropna() and fillna().

- Students will be able to standardize data formats (e.g., converting data types, standardizing text case) using Pandas.

- Students will be able to apply basic data validation techniques to identify and correct errors in datasets using Pandas.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information).

- Warm-Up (10 mins): "Data Cleaning Scenarios" - Present a few short scenarios describing data quality issues in a dataset (e.g., "A column has missing values", "Date formats are inconsistent", "Text data has mixed casing"). For each scenario, ask students: "What is the data quality issue? What is a possible strategy to address this issue?" Brief class discussion to review conceptual solutions from the previous lesson.

- Main Activity 1: "Handling Missing Values with Pandas" (25 mins):  
  
Guided code-along in Observable:  
  
Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.  
Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.  
Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).  
Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.  
  
Student practice: Provide students with datasets containing missing values in Observable. Students write Python code in their notebooks to:  
  
Detect and count missing values.  
Experiment with both dropna() and fillna() to handle the missing values.  
Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.

- Guided code-along in Observable:  
  
Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.  
Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.  
Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).  
Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.

- Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.

- Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.

- Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).

- Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.

- Student practice: Provide students with datasets containing missing values in Observable. Students write Python code in their notebooks to:  
  
Detect and count missing values.  
Experiment with both dropna() and fillna() to handle the missing values.  
Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.

- Detect and count missing values.

- Experiment with both dropna() and fillna() to handle the missing values.

- Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.

- Main Activity 2: "Standardizing Data Formats with Pandas" (20 mins):  
  
Guided code-along in Observable:  
  
Continue using the same dataset (or load a new one with format inconsistencies).  
Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.  
  
  
Student practice: Provide datasets with format inconsistencies in Observable. Students write Python code to:  
  
Identify columns with incorrect or inconsistent data types.  
Convert columns to appropriate data types using astype() and to\_datetime().  
Standardize text casing in text columns using .str.lower() or .str.upper().

- Guided code-along in Observable:  
  
Continue using the same dataset (or load a new one with format inconsistencies).  
Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Continue using the same dataset (or load a new one with format inconsistencies).

- Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).

- Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Student practice: Provide datasets with format inconsistencies in Observable. Students write Python code to:  
  
Identify columns with incorrect or inconsistent data types.  
Convert columns to appropriate data types using astype() and to\_datetime().  
Standardize text casing in text columns using .str.lower() or .str.upper().

- Identify columns with incorrect or inconsistent data types.

- Convert columns to appropriate data types using astype() and to\_datetime().

- Standardize text casing in text columns using .str.lower() or .str.upper().

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the advantages of using Python and Pandas for data cleaning tasks?  
What are some key considerations when deciding how to handle missing values (dropping vs. filling)?  
Why is standardizing data formats important for data analysis?  
What are some other common data cleaning tasks that data scientists often perform (beyond what we covered today)? (Briefly mention data validation, handling duplicates, more complex text cleaning).

- What are the advantages of using Python and Pandas for data cleaning tasks?

- What are some key considerations when deciding how to handle missing values (dropping vs. filling)?

- Why is standardizing data formats important for data analysis?

- What are some other common data cleaning tasks that data scientists often perform (beyond what we covered today)? (Briefly mention data validation, handling duplicates, more complex text cleaning).

- Assessments:  
  
Formative Task (5 mins): "Data Cleaning Technique Matching" - Matching exercise: Students match data cleaning techniques (e.g., dropna(), fillna(), astype()) to their descriptions or purposes. AI-generated quiz.  
Review of student Python code in Observable notebooks – check for correct implementation of data cleaning techniques and code syntax.

- Formative Task (5 mins): "Data Cleaning Technique Matching" - Matching exercise: Students match data cleaning techniques (e.g., dropna(), fillna(), astype()) to their descriptions or purposes. AI-generated quiz.

- Review of student Python code in Observable notebooks – check for correct implementation of data cleaning techniques and code syntax.

- Student Workbook:  
  
Code examples and explanations for Pandas data cleaning techniques: dropna(), fillna(), astype(), to\_datetime(), .str.lower(), .str.upper().  
Observable notebook exercises with "messy" datasets and coding prompts for data cleaning tasks (handling missing values, standardizing formats).  
Code templates for data cleaning operations in Pandas.  
"Choosing Data Cleaning Methods" scenario worksheet with prompts to decide appropriate cleaning techniques for different data issues.

- Code examples and explanations for Pandas data cleaning techniques: dropna(), fillna(), astype(), to\_datetime(), .str.lower(), .str.upper().

- Observable notebook exercises with "messy" datasets and coding prompts for data cleaning tasks (handling missing values, standardizing formats).

- Code templates for data cleaning operations in Pandas.

- "Choosing Data Cleaning Methods" scenario worksheet with prompts to decide appropriate cleaning techniques for different data issues.

- Extension Activities:  
  
Students explore more advanced Pandas data cleaning functions (e.g., replace(), apply(), more sophisticated imputation techniques).  
Challenge: Design a more comprehensive data cleaning pipeline in Observable for a real-world "messy" dataset, combining multiple data cleaning techniques to prepare the data for analysis. Document the steps and justify their choices.

- Students explore more advanced Pandas data cleaning functions (e.g., replace(), apply(), more sophisticated imputation techniques).

- Challenge: Design a more comprehensive data cleaning pipeline in Observable for a real-world "messy" dataset, combining multiple data cleaning techniques to prepare the data for analysis. Document the steps and justify their choices.

- Guided code-along in Observable:  
  
Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.  
Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.  
Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).  
Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.

- Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.

- Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.

- Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).

- Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.

- Student practice: Provide students with datasets containing missing values in Observable. Students write Python code in their notebooks to:  
  
Detect and count missing values.  
Experiment with both dropna() and fillna() to handle the missing values.  
Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.

- Detect and count missing values.

- Experiment with both dropna() and fillna() to handle the missing values.

- Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.

- Load a "messy" dataset (or a cleaned version from previous lesson, but ensure it still has missing values) into a Pandas DataFrame.

- Demonstrate how to detect missing values using pandas.DataFrame.isnull() or pandas.DataFrame.isna() and sum() to count them.

- Introduce pandas.DataFrame.dropna() to remove rows or columns with missing values. Demonstrate both dropna() with default settings and using axis parameter to drop columns. Discuss when dropping rows vs. columns might be appropriate (and potential drawbacks of data loss).

- Introduce pandas.DataFrame.fillna() to fill missing values with a specific value (e.g., mean, median, 0, a placeholder string). Demonstrate filling with a constant value and with the mean of a column. Discuss when imputation might be preferred over dropping data.

- Detect and count missing values.

- Experiment with both dropna() and fillna() to handle the missing values.

- Compare the results of different missing value handling techniques and discuss which approach might be most suitable for different situations.

- Guided code-along in Observable:  
  
Continue using the same dataset (or load a new one with format inconsistencies).  
Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Continue using the same dataset (or load a new one with format inconsistencies).

- Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).

- Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Student practice: Provide datasets with format inconsistencies in Observable. Students write Python code to:  
  
Identify columns with incorrect or inconsistent data types.  
Convert columns to appropriate data types using astype() and to\_datetime().  
Standardize text casing in text columns using .str.lower() or .str.upper().

- Identify columns with incorrect or inconsistent data types.

- Convert columns to appropriate data types using astype() and to\_datetime().

- Standardize text casing in text columns using .str.lower() or .str.upper().

- Continue using the same dataset (or load a new one with format inconsistencies).

- Demonstrate:  
  
Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).  
Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).

- Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Converting Data Types: Use pandas.Series.astype() to convert columns to appropriate data types (e.g., converting a column of strings representing numbers to numeric type float or int, converting date strings to datetime objects using pandas.to\_datetime()).

- Standardizing Text Case: Use pandas.Series.str.lower() or pandas.Series.str.upper() to standardize text casing in text columns.

- Identify columns with incorrect or inconsistent data types.

- Convert columns to appropriate data types using astype() and to\_datetime().

- Standardize text casing in text columns using .str.lower() or .str.upper().

- What are the advantages of using Python and Pandas for data cleaning tasks?

- What are some key considerations when deciding how to handle missing values (dropping vs. filling)?

- Why is standardizing data formats important for data analysis?

- What are some other common data cleaning tasks that data scientists often perform (beyond what we covered today)? (Briefly mention data validation, handling duplicates, more complex text cleaning).

- Formative Task (5 mins): "Data Cleaning Technique Matching" - Matching exercise: Students match data cleaning techniques (e.g., dropna(), fillna(), astype()) to their descriptions or purposes. AI-generated quiz.

- Review of student Python code in Observable notebooks – check for correct implementation of data cleaning techniques and code syntax.

- Code examples and explanations for Pandas data cleaning techniques: dropna(), fillna(), astype(), to\_datetime(), .str.lower(), .str.upper().

- Observable notebook exercises with "messy" datasets and coding prompts for data cleaning tasks (handling missing values, standardizing formats).

- Code templates for data cleaning operations in Pandas.

- "Choosing Data Cleaning Methods" scenario worksheet with prompts to decide appropriate cleaning techniques for different data issues.

- Students explore more advanced Pandas data cleaning functions (e.g., replace(), apply(), more sophisticated imputation techniques).

- Challenge: Design a more comprehensive data cleaning pipeline in Observable for a real-world "messy" dataset, combining multiple data cleaning techniques to prepare the data for analysis. Document the steps and justify their choices.

- Slide Deck: Introduction to data cleaning with Python and Pandas, code examples for dropna(), fillna(), astype(), to\_datetime(), .str.lower(), .str.upper(), data cleaning workflow.

- Handout: Observable notebook exercises with "messy" datasets and coding prompts, code templates for data cleaning operations, "Choosing Data Cleaning Methods" scenario worksheet, Pandas cheat sheet (basic cleaning functions).

- Formative Quiz (printable or digital).

- Digital "messy" datasets (CSV or similar) for student practice in Observable.

- Computers/Tablets with internet access for Observable.

### Lesson 6.3: Data Sorting and Filtering for Focused Analysis

- Title: Focusing the Lens: Sorting and Filtering Data for Deeper Insights

- Learning Outcomes:  
  
Students will be able to sort Pandas DataFrames by one or more columns using pandas.DataFrame.sort\_values().  
Students will be able to filter Pandas DataFrames to select specific subsets of data based on conditions using boolean indexing or pandas.DataFrame.query().  
Students will be able to combine sorting and filtering techniques in Python to extract and analyze focused subsets of data relevant to specific research questions.  
NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information).

- Students will be able to sort Pandas DataFrames by one or more columns using pandas.DataFrame.sort\_values().

- Students will be able to filter Pandas DataFrames to select specific subsets of data based on conditions using boolean indexing or pandas.DataFrame.query().

- Students will be able to combine sorting and filtering techniques in Python to extract and analyze focused subsets of data relevant to specific research questions.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information).

- Overview: This lesson focuses on data sorting and filtering as essential techniques for focusing data analysis. Students will learn to use Python and Pandas in Observable to sort data by different criteria and filter data based on conditions, enabling them to extract and analyse specific subsets of data to answer focused research questions.

- Activities:  
  
Warm-Up (10 mins): "Information Filtering Challenge" - Present a scenario where students need to find specific information from a large, unsorted list or table (e.g., find all students with a certain grade in a long list, find products within a specific price range in a product catalogue). Discuss: How do you efficiently find the information you need in a large dataset? What strategies do you use to "filter" or "sort" the information? Lead to the concepts of data sorting and filtering.  
Main Activity 1: "Data Sorting with Pandas" (25 mins):  
  
Guided code-along in Observable:  
  
Load a dataset into a Pandas DataFrame.  
Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.  
Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).  
  
Student practice: Provide students with datasets in Observable. Students write Python code to:  
  
Sort DataFrames by different columns in ascending and descending order.  
Sort DataFrames by multiple columns to organize data according to different criteria.  
Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.  
  
  
Main Activity 2: "Data Filtering with Pandas" (20 mins):  
  
Guided code-along in Observable:  
  
Continue using the same dataset.  
Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).  
  
Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().  
  
Student practice: Provide students with research questions that require filtering data. Students write Python code to:  
  
Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.  
Combine multiple filtering conditions to narrow down data subsets further.  
Use both boolean indexing and query() methods for filtering and compare their usage.  
  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why are data sorting and filtering important techniques for data analysis?  
How does sorting help in exploring and understanding data patterns?  
How does filtering allow us to focus analysis on specific subsets of data relevant to research questions?  
In what types of scientific investigations or real-world scenarios would data sorting and filtering be particularly useful?  
Can sorting and filtering be used together effectively? How?  
  
Assessments:  
  
Formative Task (5 mins): "Sorting and Filtering Techniques Check" - Short answer questions or matching exercise on the purpose and basic syntax of sort\_values() and data filtering techniques in Pandas. AI-generated questions.  
Review of student Python code in Observable notebooks – check for correct implementation of sorting and filtering techniques and code syntax.  
  
Student Workbook:  
  
Code examples and explanations for Pandas data sorting (sort\_values()) and filtering (boolean indexing, query()) techniques.  
Observable notebook exercises with datasets and coding prompts for data sorting and filtering tasks.  
Code templates for sorting and filtering operations in Pandas.  
"Data Exploration with Sorting and Filtering" worksheet with research questions and prompts to use sorting and filtering to answer them.  
  
Extension Activities:  
  
Students explore more advanced sorting options in Pandas (e.g., sorting by index, custom sorting functions).  
Challenge: Combine data cleaning, sorting, and filtering techniques to prepare a "messy" dataset for a specific data analysis task (e.g., answering a research question that requires specific data subsets and cleaned data). Document the entire data preparation workflow in an Observable notebook.

- Warm-Up (10 mins): "Information Filtering Challenge" - Present a scenario where students need to find specific information from a large, unsorted list or table (e.g., find all students with a certain grade in a long list, find products within a specific price range in a product catalogue). Discuss: How do you efficiently find the information you need in a large dataset? What strategies do you use to "filter" or "sort" the information? Lead to the concepts of data sorting and filtering.

- Main Activity 1: "Data Sorting with Pandas" (25 mins):  
  
Guided code-along in Observable:  
  
Load a dataset into a Pandas DataFrame.  
Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.  
Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).  
  
Student practice: Provide students with datasets in Observable. Students write Python code to:  
  
Sort DataFrames by different columns in ascending and descending order.  
Sort DataFrames by multiple columns to organize data according to different criteria.  
Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.

- Guided code-along in Observable:  
  
Load a dataset into a Pandas DataFrame.  
Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.  
Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).

- Load a dataset into a Pandas DataFrame.

- Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.

- Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).

- Student practice: Provide students with datasets in Observable. Students write Python code to:  
  
Sort DataFrames by different columns in ascending and descending order.  
Sort DataFrames by multiple columns to organize data according to different criteria.  
Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.

- Sort DataFrames by different columns in ascending and descending order.

- Sort DataFrames by multiple columns to organize data according to different criteria.

- Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.

- Main Activity 2: "Data Filtering with Pandas" (20 mins):  
  
Guided code-along in Observable:  
  
Continue using the same dataset.  
Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).  
  
Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().  
  
Student practice: Provide students with research questions that require filtering data. Students write Python code to:  
  
Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.  
Combine multiple filtering conditions to narrow down data subsets further.  
Use both boolean indexing and query() methods for filtering and compare their usage.

- Guided code-along in Observable:  
  
Continue using the same dataset.  
Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).  
  
Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().

- Continue using the same dataset.

- Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).

- Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').

- Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].

- Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).

- Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().

- Student practice: Provide students with research questions that require filtering data. Students write Python code to:  
  
Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.  
Combine multiple filtering conditions to narrow down data subsets further.  
Use both boolean indexing and query() methods for filtering and compare their usage.

- Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.

- Combine multiple filtering conditions to narrow down data subsets further.

- Use both boolean indexing and query() methods for filtering and compare their usage.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are data sorting and filtering important techniques for data analysis?  
How does sorting help in exploring and understanding data patterns?  
How does filtering allow us to focus analysis on specific subsets of data relevant to research questions?  
In what types of scientific investigations or real-world scenarios would data sorting and filtering be particularly useful?  
Can sorting and filtering be used together effectively? How?

- Why are data sorting and filtering important techniques for data analysis?

- How does sorting help in exploring and understanding data patterns?

- How does filtering allow us to focus analysis on specific subsets of data relevant to research questions?

- In what types of scientific investigations or real-world scenarios would data sorting and filtering be particularly useful?

- Can sorting and filtering be used together effectively? How?

- Assessments:  
  
Formative Task (5 mins): "Sorting and Filtering Techniques Check" - Short answer questions or matching exercise on the purpose and basic syntax of sort\_values() and data filtering techniques in Pandas. AI-generated questions.  
Review of student Python code in Observable notebooks – check for correct implementation of sorting and filtering techniques and code syntax.

- Formative Task (5 mins): "Sorting and Filtering Techniques Check" - Short answer questions or matching exercise on the purpose and basic syntax of sort\_values() and data filtering techniques in Pandas. AI-generated questions.

- Review of student Python code in Observable notebooks – check for correct implementation of sorting and filtering techniques and code syntax.

- Student Workbook:  
  
Code examples and explanations for Pandas data sorting (sort\_values()) and filtering (boolean indexing, query()) techniques.  
Observable notebook exercises with datasets and coding prompts for data sorting and filtering tasks.  
Code templates for sorting and filtering operations in Pandas.  
"Data Exploration with Sorting and Filtering" worksheet with research questions and prompts to use sorting and filtering to answer them.

- Code examples and explanations for Pandas data sorting (sort\_values()) and filtering (boolean indexing, query()) techniques.

- Observable notebook exercises with datasets and coding prompts for data sorting and filtering tasks.

- Code templates for sorting and filtering operations in Pandas.

- "Data Exploration with Sorting and Filtering" worksheet with research questions and prompts to use sorting and filtering to answer them.

- Extension Activities:  
  
Students explore more advanced sorting options in Pandas (e.g., sorting by index, custom sorting functions).  
Challenge: Combine data cleaning, sorting, and filtering techniques to prepare a "messy" dataset for a specific data analysis task (e.g., answering a research question that requires specific data subsets and cleaned data). Document the entire data preparation workflow in an Observable notebook.

- Students explore more advanced sorting options in Pandas (e.g., sorting by index, custom sorting functions).

- Challenge: Combine data cleaning, sorting, and filtering techniques to prepare a "messy" dataset for a specific data analysis task (e.g., answering a research question that requires specific data subsets and cleaned data). Document the entire data preparation workflow in an Observable notebook.

- Lesson Materials:  
  
Slide Deck: Introduction to data sorting and filtering, code examples for sort\_values(), boolean indexing, query(), combining sorting and filtering, data exploration workflow.  
Handout: Observable notebook exercises with datasets and coding prompts, code templates for sorting and filtering operations, "Data Exploration with Sorting and Filtering" worksheet with research questions, Pandas cheat sheet (sorting and filtering functions).  
Formative Quiz (printable or digital).  
Digital datasets (CSV or similar) for student practice in Observable.  
Computers/Tablets with internet access for Observable.

- Slide Deck: Introduction to data sorting and filtering, code examples for sort\_values(), boolean indexing, query(), combining sorting and filtering, data exploration workflow.

- Handout: Observable notebook exercises with datasets and coding prompts, code templates for sorting and filtering operations, "Data Exploration with Sorting and Filtering" worksheet with research questions, Pandas cheat sheet (sorting and filtering functions).

- Formative Quiz (printable or digital).

- Digital datasets (CSV or similar) for student practice in Observable.

- Computers/Tablets with internet access for Observable.

- Students will be able to sort Pandas DataFrames by one or more columns using pandas.DataFrame.sort\_values().

- Students will be able to filter Pandas DataFrames to select specific subsets of data based on conditions using boolean indexing or pandas.DataFrame.query().

- Students will be able to combine sorting and filtering techniques in Python to extract and analyze focused subsets of data relevant to specific research questions.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information).

- Warm-Up (10 mins): "Information Filtering Challenge" - Present a scenario where students need to find specific information from a large, unsorted list or table (e.g., find all students with a certain grade in a long list, find products within a specific price range in a product catalogue). Discuss: How do you efficiently find the information you need in a large dataset? What strategies do you use to "filter" or "sort" the information? Lead to the concepts of data sorting and filtering.

- Main Activity 1: "Data Sorting with Pandas" (25 mins):  
  
Guided code-along in Observable:  
  
Load a dataset into a Pandas DataFrame.  
Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.  
Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).  
  
Student practice: Provide students with datasets in Observable. Students write Python code to:  
  
Sort DataFrames by different columns in ascending and descending order.  
Sort DataFrames by multiple columns to organize data according to different criteria.  
Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.

- Guided code-along in Observable:  
  
Load a dataset into a Pandas DataFrame.  
Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.  
Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).

- Load a dataset into a Pandas DataFrame.

- Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.

- Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).

- Student practice: Provide students with datasets in Observable. Students write Python code to:  
  
Sort DataFrames by different columns in ascending and descending order.  
Sort DataFrames by multiple columns to organize data according to different criteria.  
Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.

- Sort DataFrames by different columns in ascending and descending order.

- Sort DataFrames by multiple columns to organize data according to different criteria.

- Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.

- Main Activity 2: "Data Filtering with Pandas" (20 mins):  
  
Guided code-along in Observable:  
  
Continue using the same dataset.  
Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).  
  
Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().  
  
Student practice: Provide students with research questions that require filtering data. Students write Python code to:  
  
Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.  
Combine multiple filtering conditions to narrow down data subsets further.  
Use both boolean indexing and query() methods for filtering and compare their usage.

- Guided code-along in Observable:  
  
Continue using the same dataset.  
Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).  
  
Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().

- Continue using the same dataset.

- Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).

- Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').

- Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].

- Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).

- Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().

- Student practice: Provide students with research questions that require filtering data. Students write Python code to:  
  
Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.  
Combine multiple filtering conditions to narrow down data subsets further.  
Use both boolean indexing and query() methods for filtering and compare their usage.

- Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.

- Combine multiple filtering conditions to narrow down data subsets further.

- Use both boolean indexing and query() methods for filtering and compare their usage.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are data sorting and filtering important techniques for data analysis?  
How does sorting help in exploring and understanding data patterns?  
How does filtering allow us to focus analysis on specific subsets of data relevant to research questions?  
In what types of scientific investigations or real-world scenarios would data sorting and filtering be particularly useful?  
Can sorting and filtering be used together effectively? How?

- Why are data sorting and filtering important techniques for data analysis?

- How does sorting help in exploring and understanding data patterns?

- How does filtering allow us to focus analysis on specific subsets of data relevant to research questions?

- In what types of scientific investigations or real-world scenarios would data sorting and filtering be particularly useful?

- Can sorting and filtering be used together effectively? How?

- Assessments:  
  
Formative Task (5 mins): "Sorting and Filtering Techniques Check" - Short answer questions or matching exercise on the purpose and basic syntax of sort\_values() and data filtering techniques in Pandas. AI-generated questions.  
Review of student Python code in Observable notebooks – check for correct implementation of sorting and filtering techniques and code syntax.

- Formative Task (5 mins): "Sorting and Filtering Techniques Check" - Short answer questions or matching exercise on the purpose and basic syntax of sort\_values() and data filtering techniques in Pandas. AI-generated questions.

- Review of student Python code in Observable notebooks – check for correct implementation of sorting and filtering techniques and code syntax.

- Student Workbook:  
  
Code examples and explanations for Pandas data sorting (sort\_values()) and filtering (boolean indexing, query()) techniques.  
Observable notebook exercises with datasets and coding prompts for data sorting and filtering tasks.  
Code templates for sorting and filtering operations in Pandas.  
"Data Exploration with Sorting and Filtering" worksheet with research questions and prompts to use sorting and filtering to answer them.

- Code examples and explanations for Pandas data sorting (sort\_values()) and filtering (boolean indexing, query()) techniques.

- Observable notebook exercises with datasets and coding prompts for data sorting and filtering tasks.

- Code templates for sorting and filtering operations in Pandas.

- "Data Exploration with Sorting and Filtering" worksheet with research questions and prompts to use sorting and filtering to answer them.

- Extension Activities:  
  
Students explore more advanced sorting options in Pandas (e.g., sorting by index, custom sorting functions).  
Challenge: Combine data cleaning, sorting, and filtering techniques to prepare a "messy" dataset for a specific data analysis task (e.g., answering a research question that requires specific data subsets and cleaned data). Document the entire data preparation workflow in an Observable notebook.

- Students explore more advanced sorting options in Pandas (e.g., sorting by index, custom sorting functions).

- Challenge: Combine data cleaning, sorting, and filtering techniques to prepare a "messy" dataset for a specific data analysis task (e.g., answering a research question that requires specific data subsets and cleaned data). Document the entire data preparation workflow in an Observable notebook.

- Guided code-along in Observable:  
  
Load a dataset into a Pandas DataFrame.  
Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.  
Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).

- Load a dataset into a Pandas DataFrame.

- Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.

- Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).

- Student practice: Provide students with datasets in Observable. Students write Python code to:  
  
Sort DataFrames by different columns in ascending and descending order.  
Sort DataFrames by multiple columns to organize data according to different criteria.  
Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.

- Sort DataFrames by different columns in ascending and descending order.

- Sort DataFrames by multiple columns to organize data according to different criteria.

- Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.

- Load a dataset into a Pandas DataFrame.

- Demonstrate how to sort a DataFrame by a single column using pandas.DataFrame.sort\_values(by='column\_name'). Show sorting in ascending and descending order using ascending=True/False.

- Demonstrate sorting by multiple columns using sort\_values(by=['column1', 'column2']). Explain the sorting order (primary sort by column1, then secondary sort by column2 within groups of column1).

- Sort DataFrames by different columns in ascending and descending order.

- Sort DataFrames by multiple columns to organize data according to different criteria.

- Observe how sorting helps in understanding data order and identifying trends or patterns related to sorted columns.

- Guided code-along in Observable:  
  
Continue using the same dataset.  
Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).  
  
Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().

- Continue using the same dataset.

- Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).

- Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').

- Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].

- Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).

- Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().

- Student practice: Provide students with research questions that require filtering data. Students write Python code to:  
  
Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.  
Combine multiple filtering conditions to narrow down data subsets further.  
Use both boolean indexing and query() methods for filtering and compare their usage.

- Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.

- Combine multiple filtering conditions to narrow down data subsets further.

- Use both boolean indexing and query() methods for filtering and compare their usage.

- Continue using the same dataset.

- Demonstrate data filtering using boolean indexing:  
  
Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').  
Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].  
Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).

- Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').

- Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].

- Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).

- Introduce pandas.DataFrame.query() as an alternative method for filtering using string-based queries. Demonstrate basic usage of query().

- Create boolean conditions based on column values (e.g., df['column\_name'] > value, df['category\_column'] == 'specific\_category').

- Apply boolean conditions to filter the DataFrame: filtered\_df = df[boolean\_condition].

- Show examples of filtering based on single conditions and multiple conditions combined with logical operators (& for AND, | for OR).

- Filter DataFrames to select subsets of data that meet specific criteria related to the research questions.

- Combine multiple filtering conditions to narrow down data subsets further.

- Use both boolean indexing and query() methods for filtering and compare their usage.

- Why are data sorting and filtering important techniques for data analysis?

- How does sorting help in exploring and understanding data patterns?

- How does filtering allow us to focus analysis on specific subsets of data relevant to research questions?

- In what types of scientific investigations or real-world scenarios would data sorting and filtering be particularly useful?

- Can sorting and filtering be used together effectively? How?

- Formative Task (5 mins): "Sorting and Filtering Techniques Check" - Short answer questions or matching exercise on the purpose and basic syntax of sort\_values() and data filtering techniques in Pandas. AI-generated questions.

- Review of student Python code in Observable notebooks – check for correct implementation of sorting and filtering techniques and code syntax.

- Code examples and explanations for Pandas data sorting (sort\_values()) and filtering (boolean indexing, query()) techniques.

- Observable notebook exercises with datasets and coding prompts for data sorting and filtering tasks.

- Code templates for sorting and filtering operations in Pandas.

- "Data Exploration with Sorting and Filtering" worksheet with research questions and prompts to use sorting and filtering to answer them.

- Students explore more advanced sorting options in Pandas (e.g., sorting by index, custom sorting functions).

- Challenge: Combine data cleaning, sorting, and filtering techniques to prepare a "messy" dataset for a specific data analysis task (e.g., answering a research question that requires specific data subsets and cleaned data). Document the entire data preparation workflow in an Observable notebook.

- Slide Deck: Introduction to data sorting and filtering, code examples for sort\_values(), boolean indexing, query(), combining sorting and filtering, data exploration workflow.

- Handout: Observable notebook exercises with datasets and coding prompts, code templates for sorting and filtering operations, "Data Exploration with Sorting and Filtering" worksheet with research questions, Pandas cheat sheet (sorting and filtering functions).

- Formative Quiz (printable or digital).

- Digital datasets (CSV or similar) for student practice in Observable.

- Computers/Tablets with internet access for Observable.

### Week 7: AI-Powered Data Analysis & Predictive Modelling (Introduction)

### Lesson 7.1: Introduction to AI in Data Analysis: Pattern Recognition and Insights

- Title: The AI Data Detective: Uncovering Patterns with Artificial Intelligence

- Learning Outcomes:  
  
Students will be able to define Artificial Intelligence (AI) in the context of data analysis.  
Students will be able to explain the concept of pattern recognition and how AI can be used to identify patterns in data.  
Students will be able to describe how AI can be used for data summarization and to gain insights from large datasets.  
Students will be able to recognize examples of AI applications in data analysis across different scientific fields.  
NSW Curriculum Outcomes: SC4-DA1-01 (explains how data is used by scientists to model and predict), SC4-WS-06 (uses data to identify trends, patterns and relationships).

- Students will be able to define Artificial Intelligence (AI) in the context of data analysis.

- Students will be able to explain the concept of pattern recognition and how AI can be used to identify patterns in data.

- Students will be able to describe how AI can be used for data summarization and to gain insights from large datasets.

- Students will be able to recognize examples of AI applications in data analysis across different scientific fields.

- NSW Curriculum Outcomes: SC4-DA1-01 (explains how data is used by scientists to model and predict), SC4-WS-06 (uses data to identify trends, patterns and relationships).

- Overview: This lesson provides a conceptual introduction to Artificial Intelligence (AI) and its role in modern data analysis. Students will explore how AI is used for pattern recognition, data summarization, and extracting insights from complex datasets, focusing on applications in scientific domains.

- Activities:  
  
Warm-Up (10 mins): "Pattern Recognition Challenge" - Show students a series of visual patterns (simple geometric patterns, sequences, or even more complex images with patterns like animal camouflage or constellations). Ask students to identify the patterns they see and describe how they recognized them. Discuss: What does it mean to recognize a pattern? How do our brains do it? Lead to the idea of AI mimicking this ability.  
Main Activity 1: "What is AI in Data Analysis? - The Intelligent Assistant" (25 mins):  
  
Introduce a simplified definition of Artificial Intelligence (AI) in data analysis: "Using computers to perform tasks that typically require human intelligence when working with data, such as recognizing patterns, learning from data, and making predictions or decisions."  
Emphasize that AI is not magic, but algorithms and computer programs designed to process data in sophisticated ways.  
Focus on AI as a tool to assist data scientists, not replace them.  
Explain the concept of Pattern Recognition: AI algorithms can be trained to identify complex patterns, relationships, and anomalies in data that might be difficult or impossible for humans to detect manually, especially in large datasets.  
Examples of AI pattern recognition in science:  
  
Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.  
Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.  
Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.  
  
  
Main Activity 2: "AI for Data Summarization and Insights" (20 mins):  
  
Explain how AI can be used for Data Summarization: AI algorithms can automatically summarize large datasets, identify key trends and features, and generate concise reports or visualizations, saving time and effort for data scientists.  
Examples of AI for data summarization and insights in science:  
  
Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.  
Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.  
Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.  
  
"AI in Science Case Study Exploration": Divide students into groups. Assign each group a brief case study of AI being used in data analysis in a scientific field (e.g., AI in genomics, AI in materials science, AI in robotics data analysis). Groups research and present a short summary of their case study to the class, highlighting:  
  
What type of data is being analysed?  
How is AI being used for pattern recognition or data summarization?  
What insights or benefits are gained from using AI in this context?  
  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are the most impressive or exciting applications of AI in data analysis you learned about today?  
What are the potential advantages of using AI for data analysis in science compared to traditional methods?  
Are there any limitations or potential concerns about using AI for data analysis? (Lead to ethical considerations in later lessons).  
Do you think AI will replace data scientists in the future? Why or why not? (Emphasize AI as a tool to augment human abilities).  
  
Assessments:  
  
Formative Task (5 mins): "AI in Data Analysis - Key Concepts Quiz" - Short multiple-choice quiz on the definition of AI in data analysis, pattern recognition, and data summarization. AI-generated quiz.  
Observation of student participation in case study research and presentations, assessing their understanding of AI applications.  
  
Student Workbook:  
  
Simplified definition of AI in data analysis.  
Explanations of pattern recognition and data summarization using AI.  
Summary table for case study examples of AI in science, identifying data types, AI applications, and insights gained.  
Reflection questions on the potential and limitations of AI in data analysis.  
  
Extension Activities:  
  
Students research a specific AI technique used in data analysis (e.g., clustering algorithms, neural networks – simplified conceptual overview). Present a brief explanation of the technique and its applications.  
Challenge: Explore online demos or interactive tools that showcase AI pattern recognition or data summarization (if accessible and age-appropriate). Experiment with these tools and discuss their capabilities and limitations.

- Warm-Up (10 mins): "Pattern Recognition Challenge" - Show students a series of visual patterns (simple geometric patterns, sequences, or even more complex images with patterns like animal camouflage or constellations). Ask students to identify the patterns they see and describe how they recognized them. Discuss: What does it mean to recognize a pattern? How do our brains do it? Lead to the idea of AI mimicking this ability.

- Main Activity 1: "What is AI in Data Analysis? - The Intelligent Assistant" (25 mins):  
  
Introduce a simplified definition of Artificial Intelligence (AI) in data analysis: "Using computers to perform tasks that typically require human intelligence when working with data, such as recognizing patterns, learning from data, and making predictions or decisions."  
Emphasize that AI is not magic, but algorithms and computer programs designed to process data in sophisticated ways.  
Focus on AI as a tool to assist data scientists, not replace them.  
Explain the concept of Pattern Recognition: AI algorithms can be trained to identify complex patterns, relationships, and anomalies in data that might be difficult or impossible for humans to detect manually, especially in large datasets.  
Examples of AI pattern recognition in science:  
  
Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.  
Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.  
Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.

- Introduce a simplified definition of Artificial Intelligence (AI) in data analysis: "Using computers to perform tasks that typically require human intelligence when working with data, such as recognizing patterns, learning from data, and making predictions or decisions."

- Emphasize that AI is not magic, but algorithms and computer programs designed to process data in sophisticated ways.

- Focus on AI as a tool to assist data scientists, not replace them.

- Explain the concept of Pattern Recognition: AI algorithms can be trained to identify complex patterns, relationships, and anomalies in data that might be difficult or impossible for humans to detect manually, especially in large datasets.

- Examples of AI pattern recognition in science:  
  
Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.  
Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.  
Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.

- Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.

- Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.

- Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.

- Main Activity 2: "AI for Data Summarization and Insights" (20 mins):  
  
Explain how AI can be used for Data Summarization: AI algorithms can automatically summarize large datasets, identify key trends and features, and generate concise reports or visualizations, saving time and effort for data scientists.  
Examples of AI for data summarization and insights in science:  
  
Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.  
Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.  
Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.  
  
"AI in Science Case Study Exploration": Divide students into groups. Assign each group a brief case study of AI being used in data analysis in a scientific field (e.g., AI in genomics, AI in materials science, AI in robotics data analysis). Groups research and present a short summary of their case study to the class, highlighting:  
  
What type of data is being analysed?  
How is AI being used for pattern recognition or data summarization?  
What insights or benefits are gained from using AI in this context?

- Explain how AI can be used for Data Summarization: AI algorithms can automatically summarize large datasets, identify key trends and features, and generate concise reports or visualizations, saving time and effort for data scientists.

- Examples of AI for data summarization and insights in science:  
  
Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.  
Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.  
Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.

- Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.

- Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.

- Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.

- "AI in Science Case Study Exploration": Divide students into groups. Assign each group a brief case study of AI being used in data analysis in a scientific field (e.g., AI in genomics, AI in materials science, AI in robotics data analysis). Groups research and present a short summary of their case study to the class, highlighting:  
  
What type of data is being analysed?  
How is AI being used for pattern recognition or data summarization?  
What insights or benefits are gained from using AI in this context?

- What type of data is being analysed?

- How is AI being used for pattern recognition or data summarization?

- What insights or benefits are gained from using AI in this context?

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the most impressive or exciting applications of AI in data analysis you learned about today?  
What are the potential advantages of using AI for data analysis in science compared to traditional methods?  
Are there any limitations or potential concerns about using AI for data analysis? (Lead to ethical considerations in later lessons).  
Do you think AI will replace data scientists in the future? Why or why not? (Emphasize AI as a tool to augment human abilities).

- What are the most impressive or exciting applications of AI in data analysis you learned about today?

- What are the potential advantages of using AI for data analysis in science compared to traditional methods?

- Are there any limitations or potential concerns about using AI for data analysis? (Lead to ethical considerations in later lessons).

- Do you think AI will replace data scientists in the future? Why or why not? (Emphasize AI as a tool to augment human abilities).

- Assessments:  
  
Formative Task (5 mins): "AI in Data Analysis - Key Concepts Quiz" - Short multiple-choice quiz on the definition of AI in data analysis, pattern recognition, and data summarization. AI-generated quiz.  
Observation of student participation in case study research and presentations, assessing their understanding of AI applications.

- Formative Task (5 mins): "AI in Data Analysis - Key Concepts Quiz" - Short multiple-choice quiz on the definition of AI in data analysis, pattern recognition, and data summarization. AI-generated quiz.

- Observation of student participation in case study research and presentations, assessing their understanding of AI applications.

- Student Workbook:  
  
Simplified definition of AI in data analysis.  
Explanations of pattern recognition and data summarization using AI.  
Summary table for case study examples of AI in science, identifying data types, AI applications, and insights gained.  
Reflection questions on the potential and limitations of AI in data analysis.

- Simplified definition of AI in data analysis.

- Explanations of pattern recognition and data summarization using AI.

- Summary table for case study examples of AI in science, identifying data types, AI applications, and insights gained.

- Reflection questions on the potential and limitations of AI in data analysis.

- Extension Activities:  
  
Students research a specific AI technique used in data analysis (e.g., clustering algorithms, neural networks – simplified conceptual overview). Present a brief explanation of the technique and its applications.  
Challenge: Explore online demos or interactive tools that showcase AI pattern recognition or data summarization (if accessible and age-appropriate). Experiment with these tools and discuss their capabilities and limitations.

- Students research a specific AI technique used in data analysis (e.g., clustering algorithms, neural networks – simplified conceptual overview). Present a brief explanation of the technique and its applications.

- Challenge: Explore online demos or interactive tools that showcase AI pattern recognition or data summarization (if accessible and age-appropriate). Experiment with these tools and discuss their capabilities and limitations.

- Lesson Materials:  
  
Slide Deck: Introduction to AI in data analysis, definition of AI, pattern recognition concept, data summarization concept, examples of AI applications in science, AI as a tool.  
Handout: Case study descriptions for group research, summary table for case study analysis, definitions of AI concepts.  
Formative Quiz (printable or digital).  
(Optional) Links to online demos or interactive AI tools.  
Computers/Tablets with internet access for case study research.

- Slide Deck: Introduction to AI in data analysis, definition of AI, pattern recognition concept, data summarization concept, examples of AI applications in science, AI as a tool.

- Handout: Case study descriptions for group research, summary table for case study analysis, definitions of AI concepts.

- Formative Quiz (printable or digital).

- (Optional) Links to online demos or interactive AI tools.

- Computers/Tablets with internet access for case study research.

- Students will be able to define Artificial Intelligence (AI) in the context of data analysis.

- Students will be able to explain the concept of pattern recognition and how AI can be used to identify patterns in data.

- Students will be able to describe how AI can be used for data summarization and to gain insights from large datasets.

- Students will be able to recognize examples of AI applications in data analysis across different scientific fields.

- NSW Curriculum Outcomes: SC4-DA1-01 (explains how data is used by scientists to model and predict), SC4-WS-06 (uses data to identify trends, patterns and relationships).

- Warm-Up (10 mins): "Pattern Recognition Challenge" - Show students a series of visual patterns (simple geometric patterns, sequences, or even more complex images with patterns like animal camouflage or constellations). Ask students to identify the patterns they see and describe how they recognized them. Discuss: What does it mean to recognize a pattern? How do our brains do it? Lead to the idea of AI mimicking this ability.

- Main Activity 1: "What is AI in Data Analysis? - The Intelligent Assistant" (25 mins):  
  
Introduce a simplified definition of Artificial Intelligence (AI) in data analysis: "Using computers to perform tasks that typically require human intelligence when working with data, such as recognizing patterns, learning from data, and making predictions or decisions."  
Emphasize that AI is not magic, but algorithms and computer programs designed to process data in sophisticated ways.  
Focus on AI as a tool to assist data scientists, not replace them.  
Explain the concept of Pattern Recognition: AI algorithms can be trained to identify complex patterns, relationships, and anomalies in data that might be difficult or impossible for humans to detect manually, especially in large datasets.  
Examples of AI pattern recognition in science:  
  
Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.  
Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.  
Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.

- Introduce a simplified definition of Artificial Intelligence (AI) in data analysis: "Using computers to perform tasks that typically require human intelligence when working with data, such as recognizing patterns, learning from data, and making predictions or decisions."

- Emphasize that AI is not magic, but algorithms and computer programs designed to process data in sophisticated ways.

- Focus on AI as a tool to assist data scientists, not replace them.

- Explain the concept of Pattern Recognition: AI algorithms can be trained to identify complex patterns, relationships, and anomalies in data that might be difficult or impossible for humans to detect manually, especially in large datasets.

- Examples of AI pattern recognition in science:  
  
Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.  
Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.  
Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.

- Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.

- Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.

- Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.

- Main Activity 2: "AI for Data Summarization and Insights" (20 mins):  
  
Explain how AI can be used for Data Summarization: AI algorithms can automatically summarize large datasets, identify key trends and features, and generate concise reports or visualizations, saving time and effort for data scientists.  
Examples of AI for data summarization and insights in science:  
  
Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.  
Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.  
Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.  
  
"AI in Science Case Study Exploration": Divide students into groups. Assign each group a brief case study of AI being used in data analysis in a scientific field (e.g., AI in genomics, AI in materials science, AI in robotics data analysis). Groups research and present a short summary of their case study to the class, highlighting:  
  
What type of data is being analysed?  
How is AI being used for pattern recognition or data summarization?  
What insights or benefits are gained from using AI in this context?

- Explain how AI can be used for Data Summarization: AI algorithms can automatically summarize large datasets, identify key trends and features, and generate concise reports or visualizations, saving time and effort for data scientists.

- Examples of AI for data summarization and insights in science:  
  
Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.  
Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.  
Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.

- Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.

- Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.

- Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.

- "AI in Science Case Study Exploration": Divide students into groups. Assign each group a brief case study of AI being used in data analysis in a scientific field (e.g., AI in genomics, AI in materials science, AI in robotics data analysis). Groups research and present a short summary of their case study to the class, highlighting:  
  
What type of data is being analysed?  
How is AI being used for pattern recognition or data summarization?  
What insights or benefits are gained from using AI in this context?

- What type of data is being analysed?

- How is AI being used for pattern recognition or data summarization?

- What insights or benefits are gained from using AI in this context?

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the most impressive or exciting applications of AI in data analysis you learned about today?  
What are the potential advantages of using AI for data analysis in science compared to traditional methods?  
Are there any limitations or potential concerns about using AI for data analysis? (Lead to ethical considerations in later lessons).  
Do you think AI will replace data scientists in the future? Why or why not? (Emphasize AI as a tool to augment human abilities).

- What are the most impressive or exciting applications of AI in data analysis you learned about today?

- What are the potential advantages of using AI for data analysis in science compared to traditional methods?

- Are there any limitations or potential concerns about using AI for data analysis? (Lead to ethical considerations in later lessons).

- Do you think AI will replace data scientists in the future? Why or why not? (Emphasize AI as a tool to augment human abilities).

- Assessments:  
  
Formative Task (5 mins): "AI in Data Analysis - Key Concepts Quiz" - Short multiple-choice quiz on the definition of AI in data analysis, pattern recognition, and data summarization. AI-generated quiz.  
Observation of student participation in case study research and presentations, assessing their understanding of AI applications.

- Formative Task (5 mins): "AI in Data Analysis - Key Concepts Quiz" - Short multiple-choice quiz on the definition of AI in data analysis, pattern recognition, and data summarization. AI-generated quiz.

- Observation of student participation in case study research and presentations, assessing their understanding of AI applications.

- Student Workbook:  
  
Simplified definition of AI in data analysis.  
Explanations of pattern recognition and data summarization using AI.  
Summary table for case study examples of AI in science, identifying data types, AI applications, and insights gained.  
Reflection questions on the potential and limitations of AI in data analysis.

- Simplified definition of AI in data analysis.

- Explanations of pattern recognition and data summarization using AI.

- Summary table for case study examples of AI in science, identifying data types, AI applications, and insights gained.

- Reflection questions on the potential and limitations of AI in data analysis.

- Extension Activities:  
  
Students research a specific AI technique used in data analysis (e.g., clustering algorithms, neural networks – simplified conceptual overview). Present a brief explanation of the technique and its applications.  
Challenge: Explore online demos or interactive tools that showcase AI pattern recognition or data summarization (if accessible and age-appropriate). Experiment with these tools and discuss their capabilities and limitations.

- Students research a specific AI technique used in data analysis (e.g., clustering algorithms, neural networks – simplified conceptual overview). Present a brief explanation of the technique and its applications.

- Challenge: Explore online demos or interactive tools that showcase AI pattern recognition or data summarization (if accessible and age-appropriate). Experiment with these tools and discuss their capabilities and limitations.

- Introduce a simplified definition of Artificial Intelligence (AI) in data analysis: "Using computers to perform tasks that typically require human intelligence when working with data, such as recognizing patterns, learning from data, and making predictions or decisions."

- Emphasize that AI is not magic, but algorithms and computer programs designed to process data in sophisticated ways.

- Focus on AI as a tool to assist data scientists, not replace them.

- Explain the concept of Pattern Recognition: AI algorithms can be trained to identify complex patterns, relationships, and anomalies in data that might be difficult or impossible for humans to detect manually, especially in large datasets.

- Examples of AI pattern recognition in science:  
  
Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.  
Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.  
Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.

- Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.

- Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.

- Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.

- Image Recognition in Biology/Medicine: AI identifying patterns in medical images (X-rays, scans) to detect diseases like cancer.

- Anomaly Detection in Environmental Science: AI identifying unusual patterns in sensor data to detect pollution events or environmental changes.

- Classification in Astronomy: AI classifying celestial objects (galaxies, stars) from telescope images based on patterns in their light spectra.

- Explain how AI can be used for Data Summarization: AI algorithms can automatically summarize large datasets, identify key trends and features, and generate concise reports or visualizations, saving time and effort for data scientists.

- Examples of AI for data summarization and insights in science:  
  
Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.  
Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.  
Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.

- Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.

- Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.

- Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.

- "AI in Science Case Study Exploration": Divide students into groups. Assign each group a brief case study of AI being used in data analysis in a scientific field (e.g., AI in genomics, AI in materials science, AI in robotics data analysis). Groups research and present a short summary of their case study to the class, highlighting:  
  
What type of data is being analysed?  
How is AI being used for pattern recognition or data summarization?  
What insights or benefits are gained from using AI in this context?

- What type of data is being analysed?

- How is AI being used for pattern recognition or data summarization?

- What insights or benefits are gained from using AI in this context?

- Trend Analysis in Climate Science: AI summarizing massive climate datasets to identify long-term temperature trends, sea level rise patterns, and extreme weather event frequencies.

- Text Summarization in Scientific Literature: AI summarizing large volumes of scientific research papers to extract key findings and identify research trends.

- Customer Insights in Social Science/Marketing (Relatable Example): AI summarizing customer data (purchase history, demographics) to identify customer segments and preferences for targeted marketing.

- What type of data is being analysed?

- How is AI being used for pattern recognition or data summarization?

- What insights or benefits are gained from using AI in this context?

- What are the most impressive or exciting applications of AI in data analysis you learned about today?

- What are the potential advantages of using AI for data analysis in science compared to traditional methods?

- Are there any limitations or potential concerns about using AI for data analysis? (Lead to ethical considerations in later lessons).

- Do you think AI will replace data scientists in the future? Why or why not? (Emphasize AI as a tool to augment human abilities).

- Formative Task (5 mins): "AI in Data Analysis - Key Concepts Quiz" - Short multiple-choice quiz on the definition of AI in data analysis, pattern recognition, and data summarization. AI-generated quiz.

- Observation of student participation in case study research and presentations, assessing their understanding of AI applications.

- Simplified definition of AI in data analysis.

- Explanations of pattern recognition and data summarization using AI.

- Summary table for case study examples of AI in science, identifying data types, AI applications, and insights gained.

- Reflection questions on the potential and limitations of AI in data analysis.

- Students research a specific AI technique used in data analysis (e.g., clustering algorithms, neural networks – simplified conceptual overview). Present a brief explanation of the technique and its applications.

- Challenge: Explore online demos or interactive tools that showcase AI pattern recognition or data summarization (if accessible and age-appropriate). Experiment with these tools and discuss their capabilities and limitations.

- Slide Deck: Introduction to AI in data analysis, definition of AI, pattern recognition concept, data summarization concept, examples of AI applications in science, AI as a tool.

- Handout: Case study descriptions for group research, summary table for case study analysis, definitions of AI concepts.

- Formative Quiz (printable or digital).

- (Optional) Links to online demos or interactive AI tools.

- Computers/Tablets with internet access for case study research.

### Lesson 7.2: Simplified Predictive Modelling (Conceptual Introduction - Visual Tools if feasible)

- Title: Predicting the Future (Simply): Introduction to Predictive Modelling

- Learning Outcomes:  
  
Students will be able to explain the basic concept of predictive modelling in data science.  
Students will be able to identify the key steps in building a predictive model (training, prediction, evaluation – conceptually).  
Students will be able to understand the role of training data, features, and target variables in predictive modelling.  
Students will be able to explore and use a simplified visual tool (if available) to build and test a basic predictive model.  
NSW Curriculum Outcomes: SC4-DA1-01 (explains how data is used by scientists to model and predict), SC4-WS-07 (identifies problem-solving strategies and proposes solutions).

- Students will be able to explain the basic concept of predictive modelling in data science.

- Students will be able to identify the key steps in building a predictive model (training, prediction, evaluation – conceptually).

- Students will be able to understand the role of training data, features, and target variables in predictive modelling.

- Students will be able to explore and use a simplified visual tool (if available) to build and test a basic predictive model.

- NSW Curriculum Outcomes: SC4-DA1-01 (explains how data is used by scientists to model and predict), SC4-WS-07 (identifies problem-solving strategies and proposes solutions).

- Overview: This lesson provides a simplified, conceptual introduction to predictive modelling, a core application of AI in data science. Students will learn the basic principles of building predictive models, understand key terminology, and ideally get hands-on experience with a visual, user-friendly predictive modelling tool (if feasible and accessible).

- Activities:  
  
Warm-Up (10 mins): "Prediction Scenarios" - Present several everyday prediction scenarios (e.g., weather forecast, predicting sports game outcome, predicting traffic congestion, predicting movie box office success). For each scenario, ask students: "What information or data do you think is used to make this prediction? What is being predicted?" Discuss the idea of using data to make predictions.  
Main Activity 1: "What is Predictive Modelling? - Looking into the Crystal Ball" (25 mins):  
  
Introduce the concept of Predictive Modelling: "Using data and algorithms to build models that can predict future outcomes or unknown values based on patterns learned from past data."  
Explain the key steps in building a predictive model (simplified, conceptual level):  
  
Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.  
Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).  
Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).  
Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.  
Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."  
  
Use analogies to explain these concepts (e.g., training a dog to recognize commands, learning from examples in a textbook).  
Examples of Predictive Modelling in Science:  
  
Weather Forecasting: Predicting future weather conditions based on current and past weather data.  
Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.  
Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.  
  
  
Main Activity 2: "Hands-on Predictive Modelling with a Visual Tool (If Feasible)" (20 mins):  
  
If a simplified, user-friendly visual predictive modelling tool is available and suitable for Year 8 (e.g., a drag-and-drop online tool, or a very basic feature in a data visualization platform), guide students to use it for a simple predictive modelling task.  
Example task: Predicting house prices based on features like size and location (using a simplified dataset and tool).  
Steps (if using a visual tool):  
  
Load a simplified dataset into the tool.  
Identify features (input variables) and target variable (output to predict).  
Use the tool's interface to "train" a basic predictive model.  
Use the trained model to make predictions for new data points.  
Evaluate the model's performance (if the tool provides any basic evaluation metrics).  
  
If a visual tool is not feasible, demonstrate a very simplified predictive modelling example conceptually using a simple dataset and a basic rule or pattern that could be used for prediction.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are the key steps involved in building a predictive model?  
What is the role of training data and features in predictive modelling?  
What are some real-world applications of predictive modelling in science and other fields that you find interesting?  
What are some limitations or challenges of predictive modelling? Are predictions always accurate? Why or why not? (Introduce the idea of model accuracy, uncertainty, and the fact that predictions are not guarantees).  
  
Assessments:  
  
Formative Task (5 mins): "Predictive Modelling Concepts Quiz" - Short multiple-choice quiz on the definition of predictive modelling, training data, features, and target variables. AI-generated quiz.  
Observation of student engagement and participation during the hands-on activity with the visual predictive modelling tool (if used).  
  
Student Workbook:  
  
Simplified explanation of predictive modelling and its key steps (training, prediction, evaluation).  
Definitions of training data, features, and target variables.  
Examples of predictive modelling applications in science.  
Instructions for using the visual predictive modelling tool (if used) and prompts for the hands-on activity.  
Reflection questions on predictive modelling and its limitations.  
  
Extension Activities:  
  
Students research real-world examples of predictive models used in a scientific field they are interested in (e.g., climate change prediction models, disease outbreak prediction models). Present a summary of their findings.  
Challenge: Explore more advanced concepts related to predictive modelling (e.g., different types of predictive models – linear regression, classification, etc. - conceptually; overfitting and underfitting – conceptually).

- Warm-Up (10 mins): "Prediction Scenarios" - Present several everyday prediction scenarios (e.g., weather forecast, predicting sports game outcome, predicting traffic congestion, predicting movie box office success). For each scenario, ask students: "What information or data do you think is used to make this prediction? What is being predicted?" Discuss the idea of using data to make predictions.

- Main Activity 1: "What is Predictive Modelling? - Looking into the Crystal Ball" (25 mins):  
  
Introduce the concept of Predictive Modelling: "Using data and algorithms to build models that can predict future outcomes or unknown values based on patterns learned from past data."  
Explain the key steps in building a predictive model (simplified, conceptual level):  
  
Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.  
Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).  
Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).  
Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.  
Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."  
  
Use analogies to explain these concepts (e.g., training a dog to recognize commands, learning from examples in a textbook).  
Examples of Predictive Modelling in Science:  
  
Weather Forecasting: Predicting future weather conditions based on current and past weather data.  
Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.  
Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.

- Introduce the concept of Predictive Modelling: "Using data and algorithms to build models that can predict future outcomes or unknown values based on patterns learned from past data."

- Explain the key steps in building a predictive model (simplified, conceptual level):  
  
Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.  
Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).  
Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).  
Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.  
Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."

- Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.

- Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).

- Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).

- Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.

- Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."

- Use analogies to explain these concepts (e.g., training a dog to recognize commands, learning from examples in a textbook).

- Examples of Predictive Modelling in Science:  
  
Weather Forecasting: Predicting future weather conditions based on current and past weather data.  
Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.  
Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.

- Weather Forecasting: Predicting future weather conditions based on current and past weather data.

- Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.

- Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.

- Main Activity 2: "Hands-on Predictive Modelling with a Visual Tool (If Feasible)" (20 mins):  
  
If a simplified, user-friendly visual predictive modelling tool is available and suitable for Year 8 (e.g., a drag-and-drop online tool, or a very basic feature in a data visualization platform), guide students to use it for a simple predictive modelling task.  
Example task: Predicting house prices based on features like size and location (using a simplified dataset and tool).  
Steps (if using a visual tool):  
  
Load a simplified dataset into the tool.  
Identify features (input variables) and target variable (output to predict).  
Use the tool's interface to "train" a basic predictive model.  
Use the trained model to make predictions for new data points.  
Evaluate the model's performance (if the tool provides any basic evaluation metrics).  
  
If a visual tool is not feasible, demonstrate a very simplified predictive modelling example conceptually using a simple dataset and a basic rule or pattern that could be used for prediction.

- If a simplified, user-friendly visual predictive modelling tool is available and suitable for Year 8 (e.g., a drag-and-drop online tool, or a very basic feature in a data visualization platform), guide students to use it for a simple predictive modelling task.

- Example task: Predicting house prices based on features like size and location (using a simplified dataset and tool).

- Steps (if using a visual tool):  
  
Load a simplified dataset into the tool.  
Identify features (input variables) and target variable (output to predict).  
Use the tool's interface to "train" a basic predictive model.  
Use the trained model to make predictions for new data points.  
Evaluate the model's performance (if the tool provides any basic evaluation metrics).

- Load a simplified dataset into the tool.

- Identify features (input variables) and target variable (output to predict).

- Use the tool's interface to "train" a basic predictive model.

- Use the trained model to make predictions for new data points.

- Evaluate the model's performance (if the tool provides any basic evaluation metrics).

- If a visual tool is not feasible, demonstrate a very simplified predictive modelling example conceptually using a simple dataset and a basic rule or pattern that could be used for prediction.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the key steps involved in building a predictive model?  
What is the role of training data and features in predictive modelling?  
What are some real-world applications of predictive modelling in science and other fields that you find interesting?  
What are some limitations or challenges of predictive modelling? Are predictions always accurate? Why or why not? (Introduce the idea of model accuracy, uncertainty, and the fact that predictions are not guarantees).

- What are the key steps involved in building a predictive model?

- What is the role of training data and features in predictive modelling?

- What are some real-world applications of predictive modelling in science and other fields that you find interesting?

- What are some limitations or challenges of predictive modelling? Are predictions always accurate? Why or why not? (Introduce the idea of model accuracy, uncertainty, and the fact that predictions are not guarantees).

- Assessments:  
  
Formative Task (5 mins): "Predictive Modelling Concepts Quiz" - Short multiple-choice quiz on the definition of predictive modelling, training data, features, and target variables. AI-generated quiz.  
Observation of student engagement and participation during the hands-on activity with the visual predictive modelling tool (if used).

- Formative Task (5 mins): "Predictive Modelling Concepts Quiz" - Short multiple-choice quiz on the definition of predictive modelling, training data, features, and target variables. AI-generated quiz.

- Observation of student engagement and participation during the hands-on activity with the visual predictive modelling tool (if used).

- Student Workbook:  
  
Simplified explanation of predictive modelling and its key steps (training, prediction, evaluation).  
Definitions of training data, features, and target variables.  
Examples of predictive modelling applications in science.  
Instructions for using the visual predictive modelling tool (if used) and prompts for the hands-on activity.  
Reflection questions on predictive modelling and its limitations.

- Simplified explanation of predictive modelling and its key steps (training, prediction, evaluation).

- Definitions of training data, features, and target variables.

- Examples of predictive modelling applications in science.

- Instructions for using the visual predictive modelling tool (if used) and prompts for the hands-on activity.

- Reflection questions on predictive modelling and its limitations.

- Extension Activities:  
  
Students research real-world examples of predictive models used in a scientific field they are interested in (e.g., climate change prediction models, disease outbreak prediction models). Present a summary of their findings.  
Challenge: Explore more advanced concepts related to predictive modelling (e.g., different types of predictive models – linear regression, classification, etc. - conceptually; overfitting and underfitting – conceptually).

- Students research real-world examples of predictive models used in a scientific field they are interested in (e.g., climate change prediction models, disease outbreak prediction models). Present a summary of their findings.

- Challenge: Explore more advanced concepts related to predictive modelling (e.g., different types of predictive models – linear regression, classification, etc. - conceptually; overfitting and underfitting – conceptually).

- Lesson Materials:  
  
Slide Deck: Introduction to predictive modelling, definition, key steps (training, prediction, evaluation), features, target variable, examples of predictive models in science.  
Handout: Instructions for using visual predictive modelling tool (if used), simplified dataset for hands-on activity, definitions of predictive modelling concepts.  
Formative Quiz (printable or digital).  
Access to a simplified visual predictive modelling tool (if feasible and available).  
(Optional) Example datasets for predictive modelling.

- Slide Deck: Introduction to predictive modelling, definition, key steps (training, prediction, evaluation), features, target variable, examples of predictive models in science.

- Handout: Instructions for using visual predictive modelling tool (if used), simplified dataset for hands-on activity, definitions of predictive modelling concepts.

- Formative Quiz (printable or digital).

- Access to a simplified visual predictive modelling tool (if feasible and available).

- (Optional) Example datasets for predictive modelling.

- Students will be able to explain the basic concept of predictive modelling in data science.

- Students will be able to identify the key steps in building a predictive model (training, prediction, evaluation – conceptually).

- Students will be able to understand the role of training data, features, and target variables in predictive modelling.

- Students will be able to explore and use a simplified visual tool (if available) to build and test a basic predictive model.

- NSW Curriculum Outcomes: SC4-DA1-01 (explains how data is used by scientists to model and predict), SC4-WS-07 (identifies problem-solving strategies and proposes solutions).

- Warm-Up (10 mins): "Prediction Scenarios" - Present several everyday prediction scenarios (e.g., weather forecast, predicting sports game outcome, predicting traffic congestion, predicting movie box office success). For each scenario, ask students: "What information or data do you think is used to make this prediction? What is being predicted?" Discuss the idea of using data to make predictions.

- Main Activity 1: "What is Predictive Modelling? - Looking into the Crystal Ball" (25 mins):  
  
Introduce the concept of Predictive Modelling: "Using data and algorithms to build models that can predict future outcomes or unknown values based on patterns learned from past data."  
Explain the key steps in building a predictive model (simplified, conceptual level):  
  
Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.  
Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).  
Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).  
Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.  
Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."  
  
Use analogies to explain these concepts (e.g., training a dog to recognize commands, learning from examples in a textbook).  
Examples of Predictive Modelling in Science:  
  
Weather Forecasting: Predicting future weather conditions based on current and past weather data.  
Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.  
Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.

- Introduce the concept of Predictive Modelling: "Using data and algorithms to build models that can predict future outcomes or unknown values based on patterns learned from past data."

- Explain the key steps in building a predictive model (simplified, conceptual level):  
  
Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.  
Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).  
Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).  
Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.  
Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."

- Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.

- Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).

- Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).

- Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.

- Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."

- Use analogies to explain these concepts (e.g., training a dog to recognize commands, learning from examples in a textbook).

- Examples of Predictive Modelling in Science:  
  
Weather Forecasting: Predicting future weather conditions based on current and past weather data.  
Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.  
Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.

- Weather Forecasting: Predicting future weather conditions based on current and past weather data.

- Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.

- Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.

- Main Activity 2: "Hands-on Predictive Modelling with a Visual Tool (If Feasible)" (20 mins):  
  
If a simplified, user-friendly visual predictive modelling tool is available and suitable for Year 8 (e.g., a drag-and-drop online tool, or a very basic feature in a data visualization platform), guide students to use it for a simple predictive modelling task.  
Example task: Predicting house prices based on features like size and location (using a simplified dataset and tool).  
Steps (if using a visual tool):  
  
Load a simplified dataset into the tool.  
Identify features (input variables) and target variable (output to predict).  
Use the tool's interface to "train" a basic predictive model.  
Use the trained model to make predictions for new data points.  
Evaluate the model's performance (if the tool provides any basic evaluation metrics).  
  
If a visual tool is not feasible, demonstrate a very simplified predictive modelling example conceptually using a simple dataset and a basic rule or pattern that could be used for prediction.

- If a simplified, user-friendly visual predictive modelling tool is available and suitable for Year 8 (e.g., a drag-and-drop online tool, or a very basic feature in a data visualization platform), guide students to use it for a simple predictive modelling task.

- Example task: Predicting house prices based on features like size and location (using a simplified dataset and tool).

- Steps (if using a visual tool):  
  
Load a simplified dataset into the tool.  
Identify features (input variables) and target variable (output to predict).  
Use the tool's interface to "train" a basic predictive model.  
Use the trained model to make predictions for new data points.  
Evaluate the model's performance (if the tool provides any basic evaluation metrics).

- Load a simplified dataset into the tool.

- Identify features (input variables) and target variable (output to predict).

- Use the tool's interface to "train" a basic predictive model.

- Use the trained model to make predictions for new data points.

- Evaluate the model's performance (if the tool provides any basic evaluation metrics).

- If a visual tool is not feasible, demonstrate a very simplified predictive modelling example conceptually using a simple dataset and a basic rule or pattern that could be used for prediction.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the key steps involved in building a predictive model?  
What is the role of training data and features in predictive modelling?  
What are some real-world applications of predictive modelling in science and other fields that you find interesting?  
What are some limitations or challenges of predictive modelling? Are predictions always accurate? Why or why not? (Introduce the idea of model accuracy, uncertainty, and the fact that predictions are not guarantees).

- What are the key steps involved in building a predictive model?

- What is the role of training data and features in predictive modelling?

- What are some real-world applications of predictive modelling in science and other fields that you find interesting?

- What are some limitations or challenges of predictive modelling? Are predictions always accurate? Why or why not? (Introduce the idea of model accuracy, uncertainty, and the fact that predictions are not guarantees).

- Assessments:  
  
Formative Task (5 mins): "Predictive Modelling Concepts Quiz" - Short multiple-choice quiz on the definition of predictive modelling, training data, features, and target variables. AI-generated quiz.  
Observation of student engagement and participation during the hands-on activity with the visual predictive modelling tool (if used).

- Formative Task (5 mins): "Predictive Modelling Concepts Quiz" - Short multiple-choice quiz on the definition of predictive modelling, training data, features, and target variables. AI-generated quiz.

- Observation of student engagement and participation during the hands-on activity with the visual predictive modelling tool (if used).

- Student Workbook:  
  
Simplified explanation of predictive modelling and its key steps (training, prediction, evaluation).  
Definitions of training data, features, and target variables.  
Examples of predictive modelling applications in science.  
Instructions for using the visual predictive modelling tool (if used) and prompts for the hands-on activity.  
Reflection questions on predictive modelling and its limitations.

- Simplified explanation of predictive modelling and its key steps (training, prediction, evaluation).

- Definitions of training data, features, and target variables.

- Examples of predictive modelling applications in science.

- Instructions for using the visual predictive modelling tool (if used) and prompts for the hands-on activity.

- Reflection questions on predictive modelling and its limitations.

- Extension Activities:  
  
Students research real-world examples of predictive models used in a scientific field they are interested in (e.g., climate change prediction models, disease outbreak prediction models). Present a summary of their findings.  
Challenge: Explore more advanced concepts related to predictive modelling (e.g., different types of predictive models – linear regression, classification, etc. - conceptually; overfitting and underfitting – conceptually).

- Students research real-world examples of predictive models used in a scientific field they are interested in (e.g., climate change prediction models, disease outbreak prediction models). Present a summary of their findings.

- Challenge: Explore more advanced concepts related to predictive modelling (e.g., different types of predictive models – linear regression, classification, etc. - conceptually; overfitting and underfitting – conceptually).

- Introduce the concept of Predictive Modelling: "Using data and algorithms to build models that can predict future outcomes or unknown values based on patterns learned from past data."

- Explain the key steps in building a predictive model (simplified, conceptual level):  
  
Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.  
Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).  
Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).  
Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.  
Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."

- Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.

- Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).

- Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).

- Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.

- Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."

- Use analogies to explain these concepts (e.g., training a dog to recognize commands, learning from examples in a textbook).

- Examples of Predictive Modelling in Science:  
  
Weather Forecasting: Predicting future weather conditions based on current and past weather data.  
Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.  
Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.

- Weather Forecasting: Predicting future weather conditions based on current and past weather data.

- Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.

- Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.

- Training the Model: The model "learns" patterns and relationships from a dataset called the "training data." This is like teaching a computer to recognize patterns.

- Features: These are the input variables or characteristics used to make predictions (e.g., in weather prediction: temperature, humidity, wind speed; in predicting house price: size, location, number of bedrooms).

- Target Variable: This is the variable we want to predict (e.g., weather: temperature tomorrow; house price: selling price).

- Making Predictions: Once trained, the model can use new data (features) to predict the target variable for unseen situations.

- Evaluating the Model: After building a model, we need to check how accurate its predictions are. This is called "model evaluation."

- Weather Forecasting: Predicting future weather conditions based on current and past weather data.

- Disease Prediction in Medicine: Predicting patient risk of developing a disease based on medical history, genetics, lifestyle data.

- Predicting Species Distribution in Ecology: Predicting where different species are likely to be found based on environmental factors and species occurrence data.

- If a simplified, user-friendly visual predictive modelling tool is available and suitable for Year 8 (e.g., a drag-and-drop online tool, or a very basic feature in a data visualization platform), guide students to use it for a simple predictive modelling task.

- Example task: Predicting house prices based on features like size and location (using a simplified dataset and tool).

- Steps (if using a visual tool):  
  
Load a simplified dataset into the tool.  
Identify features (input variables) and target variable (output to predict).  
Use the tool's interface to "train" a basic predictive model.  
Use the trained model to make predictions for new data points.  
Evaluate the model's performance (if the tool provides any basic evaluation metrics).

- Load a simplified dataset into the tool.

- Identify features (input variables) and target variable (output to predict).

- Use the tool's interface to "train" a basic predictive model.

- Use the trained model to make predictions for new data points.

- Evaluate the model's performance (if the tool provides any basic evaluation metrics).

- If a visual tool is not feasible, demonstrate a very simplified predictive modelling example conceptually using a simple dataset and a basic rule or pattern that could be used for prediction.

- Load a simplified dataset into the tool.

- Identify features (input variables) and target variable (output to predict).

- Use the tool's interface to "train" a basic predictive model.

- Use the trained model to make predictions for new data points.

- Evaluate the model's performance (if the tool provides any basic evaluation metrics).

- What are the key steps involved in building a predictive model?

- What is the role of training data and features in predictive modelling?

- What are some real-world applications of predictive modelling in science and other fields that you find interesting?

- What are some limitations or challenges of predictive modelling? Are predictions always accurate? Why or why not? (Introduce the idea of model accuracy, uncertainty, and the fact that predictions are not guarantees).

- Formative Task (5 mins): "Predictive Modelling Concepts Quiz" - Short multiple-choice quiz on the definition of predictive modelling, training data, features, and target variables. AI-generated quiz.

- Observation of student engagement and participation during the hands-on activity with the visual predictive modelling tool (if used).

- Simplified explanation of predictive modelling and its key steps (training, prediction, evaluation).

- Definitions of training data, features, and target variables.

- Examples of predictive modelling applications in science.

- Instructions for using the visual predictive modelling tool (if used) and prompts for the hands-on activity.

- Reflection questions on predictive modelling and its limitations.

- Students research real-world examples of predictive models used in a scientific field they are interested in (e.g., climate change prediction models, disease outbreak prediction models). Present a summary of their findings.

- Challenge: Explore more advanced concepts related to predictive modelling (e.g., different types of predictive models – linear regression, classification, etc. - conceptually; overfitting and underfitting – conceptually).

- Slide Deck: Introduction to predictive modelling, definition, key steps (training, prediction, evaluation), features, target variable, examples of predictive models in science.

- Handout: Instructions for using visual predictive modelling tool (if used), simplified dataset for hands-on activity, definitions of predictive modelling concepts.

- Formative Quiz (printable or digital).

- Access to a simplified visual predictive modelling tool (if feasible and available).

- (Optional) Example datasets for predictive modelling.

### Lesson 7.3: Ethical Considerations of AI in Data Science

- Title: AI Ethics Compass: Navigating the Ethical Landscape of Data Science

- Learning Outcomes:  
  
Students will be able to identify and discuss ethical considerations related to the use of AI in data science.  
Students will be able to explain the concept of bias in AI and its potential sources and consequences.  
Students will be able to discuss data privacy concerns related to AI and data science applications.  
Students will be able to understand the importance of transparency and explainability in AI systems, especially in critical applications.  
NSW Curriculum Outcomes: SC4-WS-08 (Reflecting on scientific methods), SC4-3VA (Nature and Development of Science - Science as a human endeavour).

- Students will be able to identify and discuss ethical considerations related to the use of AI in data science.

- Students will be able to explain the concept of bias in AI and its potential sources and consequences.

- Students will be able to discuss data privacy concerns related to AI and data science applications.

- Students will be able to understand the importance of transparency and explainability in AI systems, especially in critical applications.

- NSW Curriculum Outcomes: SC4-WS-08 (Reflecting on scientific methods), SC4-3VA (Nature and Development of Science - Science as a human endeavour).

- Overview: This crucial lesson focuses on the ethical dimensions of AI in data science. Students will explore key ethical considerations, including bias in AI, data privacy, and the need for transparency and responsible AI development, fostering critical thinking about the societal impact of these technologies.

- Activities:  
  
Warm-Up (10 mins): "Ethical Dilemma Scenarios (General)" - Present a few short ethical dilemma scenarios that are not directly related to AI but introduce the concept of ethical decision-making and trade-offs (e.g., a scenario about resource allocation, a scenario about honesty vs. loyalty). Discuss: What are the ethical issues in this scenario? Are there different perspectives on what is "right"? How do you make an ethical decision when there are competing values? Lead to the idea of ethical considerations being complex and important.  
Main Activity 1: "Bias in AI - Unintended Consequences" (25 mins):  
  
Introduce the concept of Bias in AI: "AI systems can unintentionally learn and perpetuate biases that exist in the data they are trained on, leading to unfair or discriminatory outcomes."  
Explain potential sources of bias in AI:  
  
Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).  
Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).  
Human Bias: Biases of the people who design, develop, and deploy AI systems.  
  
Examples of AI Bias and its consequences:  
  
Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).  
Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.  
Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.  
  
"Bias Detection Challenge": Present simplified scenarios or short descriptions of AI systems. Ask students to identify potential sources of bias and possible consequences of biased outcomes in each scenario. Discuss: How can we detect and mitigate bias in AI systems?  
  
Main Activity 2: "Data Privacy and Transparency in AI" (20 mins):  
  
Discuss Data Privacy Concerns in AI: AI systems often rely on large amounts of personal data. It's crucial to consider how this data is collected, used, stored, and protected. Discuss issues like:  
  
Informed consent for data collection.  
Anonymization and de-identification of data.  
Data security and preventing data breaches.  
Potential for misuse of personal data.  
  
Explain the importance of Transparency and Explainability in AI, especially in critical applications (e.g., healthcare, justice). "Black Box" AI systems (where it's hard to understand how they make decisions) can raise ethical concerns. Transparency means making AI systems more understandable and explainable.  
Examples of ethical considerations related to privacy and transparency:  
  
Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.  
AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.  
Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.  
  
"Ethical AI Principles Brainstorm": In groups, students brainstorm a list of ethical principles or guidelines that should be followed when developing and using AI in data science to address issues of bias, privacy, and transparency. Share and discuss principles as a class.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why are ethical considerations so important in the field of AI and data science?  
What are the potential harms or negative consequences of unethical AI development and use?  
What steps can data scientists, developers, and society take to promote ethical AI and mitigate risks of bias, privacy violations, and lack of transparency?  
What role do you think you, as future data-literate citizens, can play in ensuring AI is used ethically and responsibly?  
  
Assessments:  
  
Formative Task (5 mins): "AI Ethics - Key Terms Check" - Short answer questions defining bias in AI and explaining data privacy concerns. AI-generated questions.  
Observation of student participation in ethical dilemma discussions and brainstorming activities, assessing their engagement with ethical considerations.  
  
Student Workbook:  
  
Explanations of bias in AI, data privacy concerns, and transparency in AI.  
Examples of AI bias and its consequences.  
Ethical dilemma scenarios related to AI in data science.  
"Bias Detection Challenge" worksheet with AI scenarios.  
"Ethical AI Principles Brainstorm" worksheet with prompts.  
Reflection questions on ethical AI and responsible innovation.  
  
Extension Activities:  
  
Students research current ethical debates and discussions around AI in the news or scientific literature (e.g., debates about AI regulation, algorithmic bias in specific AI systems, data privacy laws). Present findings and lead a class discussion.  
Challenge: Design a "Code of Ethics for AI Data Scientists" targeted at Year 8 students, outlining key ethical principles and responsible practices for working with data and AI.

- Warm-Up (10 mins): "Ethical Dilemma Scenarios (General)" - Present a few short ethical dilemma scenarios that are not directly related to AI but introduce the concept of ethical decision-making and trade-offs (e.g., a scenario about resource allocation, a scenario about honesty vs. loyalty). Discuss: What are the ethical issues in this scenario? Are there different perspectives on what is "right"? How do you make an ethical decision when there are competing values? Lead to the idea of ethical considerations being complex and important.

- Main Activity 1: "Bias in AI - Unintended Consequences" (25 mins):  
  
Introduce the concept of Bias in AI: "AI systems can unintentionally learn and perpetuate biases that exist in the data they are trained on, leading to unfair or discriminatory outcomes."  
Explain potential sources of bias in AI:  
  
Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).  
Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).  
Human Bias: Biases of the people who design, develop, and deploy AI systems.  
  
Examples of AI Bias and its consequences:  
  
Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).  
Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.  
Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.  
  
"Bias Detection Challenge": Present simplified scenarios or short descriptions of AI systems. Ask students to identify potential sources of bias and possible consequences of biased outcomes in each scenario. Discuss: How can we detect and mitigate bias in AI systems?

- Introduce the concept of Bias in AI: "AI systems can unintentionally learn and perpetuate biases that exist in the data they are trained on, leading to unfair or discriminatory outcomes."

- Explain potential sources of bias in AI:  
  
Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).  
Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).  
Human Bias: Biases of the people who design, develop, and deploy AI systems.

- Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).

- Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).

- Human Bias: Biases of the people who design, develop, and deploy AI systems.

- Examples of AI Bias and its consequences:  
  
Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).  
Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.  
Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.

- Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).

- Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.

- Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.

- "Bias Detection Challenge": Present simplified scenarios or short descriptions of AI systems. Ask students to identify potential sources of bias and possible consequences of biased outcomes in each scenario. Discuss: How can we detect and mitigate bias in AI systems?

- Main Activity 2: "Data Privacy and Transparency in AI" (20 mins):  
  
Discuss Data Privacy Concerns in AI: AI systems often rely on large amounts of personal data. It's crucial to consider how this data is collected, used, stored, and protected. Discuss issues like:  
  
Informed consent for data collection.  
Anonymization and de-identification of data.  
Data security and preventing data breaches.  
Potential for misuse of personal data.  
  
Explain the importance of Transparency and Explainability in AI, especially in critical applications (e.g., healthcare, justice). "Black Box" AI systems (where it's hard to understand how they make decisions) can raise ethical concerns. Transparency means making AI systems more understandable and explainable.  
Examples of ethical considerations related to privacy and transparency:  
  
Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.  
AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.  
Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.  
  
"Ethical AI Principles Brainstorm": In groups, students brainstorm a list of ethical principles or guidelines that should be followed when developing and using AI in data science to address issues of bias, privacy, and transparency. Share and discuss principles as a class.

- Discuss Data Privacy Concerns in AI: AI systems often rely on large amounts of personal data. It's crucial to consider how this data is collected, used, stored, and protected. Discuss issues like:  
  
Informed consent for data collection.  
Anonymization and de-identification of data.  
Data security and preventing data breaches.  
Potential for misuse of personal data.

- Informed consent for data collection.

- Anonymization and de-identification of data.

- Data security and preventing data breaches.

- Potential for misuse of personal data.

- Explain the importance of Transparency and Explainability in AI, especially in critical applications (e.g., healthcare, justice). "Black Box" AI systems (where it's hard to understand how they make decisions) can raise ethical concerns. Transparency means making AI systems more understandable and explainable.

- Examples of ethical considerations related to privacy and transparency:  
  
Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.  
AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.  
Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.

- Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.

- AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.

- Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.

- "Ethical AI Principles Brainstorm": In groups, students brainstorm a list of ethical principles or guidelines that should be followed when developing and using AI in data science to address issues of bias, privacy, and transparency. Share and discuss principles as a class.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are ethical considerations so important in the field of AI and data science?  
What are the potential harms or negative consequences of unethical AI development and use?  
What steps can data scientists, developers, and society take to promote ethical AI and mitigate risks of bias, privacy violations, and lack of transparency?  
What role do you think you, as future data-literate citizens, can play in ensuring AI is used ethically and responsibly?

- Why are ethical considerations so important in the field of AI and data science?

- What are the potential harms or negative consequences of unethical AI development and use?

- What steps can data scientists, developers, and society take to promote ethical AI and mitigate risks of bias, privacy violations, and lack of transparency?

- What role do you think you, as future data-literate citizens, can play in ensuring AI is used ethically and responsibly?

- Assessments:  
  
Formative Task (5 mins): "AI Ethics - Key Terms Check" - Short answer questions defining bias in AI and explaining data privacy concerns. AI-generated questions.  
Observation of student participation in ethical dilemma discussions and brainstorming activities, assessing their engagement with ethical considerations.

- Formative Task (5 mins): "AI Ethics - Key Terms Check" - Short answer questions defining bias in AI and explaining data privacy concerns. AI-generated questions.

- Observation of student participation in ethical dilemma discussions and brainstorming activities, assessing their engagement with ethical considerations.

- Student Workbook:  
  
Explanations of bias in AI, data privacy concerns, and transparency in AI.  
Examples of AI bias and its consequences.  
Ethical dilemma scenarios related to AI in data science.  
"Bias Detection Challenge" worksheet with AI scenarios.  
"Ethical AI Principles Brainstorm" worksheet with prompts.  
Reflection questions on ethical AI and responsible innovation.

- Explanations of bias in AI, data privacy concerns, and transparency in AI.

- Examples of AI bias and its consequences.

- Ethical dilemma scenarios related to AI in data science.

- "Bias Detection Challenge" worksheet with AI scenarios.

- "Ethical AI Principles Brainstorm" worksheet with prompts.

- Reflection questions on ethical AI and responsible innovation.

- Extension Activities:  
  
Students research current ethical debates and discussions around AI in the news or scientific literature (e.g., debates about AI regulation, algorithmic bias in specific AI systems, data privacy laws). Present findings and lead a class discussion.  
Challenge: Design a "Code of Ethics for AI Data Scientists" targeted at Year 8 students, outlining key ethical principles and responsible practices for working with data and AI.

- Students research current ethical debates and discussions around AI in the news or scientific literature (e.g., debates about AI regulation, algorithmic bias in specific AI systems, data privacy laws). Present findings and lead a class discussion.

- Challenge: Design a "Code of Ethics for AI Data Scientists" targeted at Year 8 students, outlining key ethical principles and responsible practices for working with data and AI.

- Lesson Materials:  
  
Slide Deck: Introduction to AI ethics, definition of bias in AI, sources and consequences of bias, data privacy concerns, transparency and explainability, responsible AI development.  
Handout: Ethical dilemma scenarios related to AI, "Bias Detection Challenge" worksheet, "Ethical AI Principles Brainstorm" worksheet, definitions of ethical AI concepts.  
Formative Quiz (printable or digital).  
(Optional) Short video clips or news reports related to AI ethics debates.

- Slide Deck: Introduction to AI ethics, definition of bias in AI, sources and consequences of bias, data privacy concerns, transparency and explainability, responsible AI development.

- Handout: Ethical dilemma scenarios related to AI, "Bias Detection Challenge" worksheet, "Ethical AI Principles Brainstorm" worksheet, definitions of ethical AI concepts.

- Formative Quiz (printable or digital).

- (Optional) Short video clips or news reports related to AI ethics debates.

- Students will be able to identify and discuss ethical considerations related to the use of AI in data science.

- Students will be able to explain the concept of bias in AI and its potential sources and consequences.

- Students will be able to discuss data privacy concerns related to AI and data science applications.

- Students will be able to understand the importance of transparency and explainability in AI systems, especially in critical applications.

- NSW Curriculum Outcomes: SC4-WS-08 (Reflecting on scientific methods), SC4-3VA (Nature and Development of Science - Science as a human endeavour).

- Warm-Up (10 mins): "Ethical Dilemma Scenarios (General)" - Present a few short ethical dilemma scenarios that are not directly related to AI but introduce the concept of ethical decision-making and trade-offs (e.g., a scenario about resource allocation, a scenario about honesty vs. loyalty). Discuss: What are the ethical issues in this scenario? Are there different perspectives on what is "right"? How do you make an ethical decision when there are competing values? Lead to the idea of ethical considerations being complex and important.

- Main Activity 1: "Bias in AI - Unintended Consequences" (25 mins):  
  
Introduce the concept of Bias in AI: "AI systems can unintentionally learn and perpetuate biases that exist in the data they are trained on, leading to unfair or discriminatory outcomes."  
Explain potential sources of bias in AI:  
  
Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).  
Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).  
Human Bias: Biases of the people who design, develop, and deploy AI systems.  
  
Examples of AI Bias and its consequences:  
  
Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).  
Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.  
Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.  
  
"Bias Detection Challenge": Present simplified scenarios or short descriptions of AI systems. Ask students to identify potential sources of bias and possible consequences of biased outcomes in each scenario. Discuss: How can we detect and mitigate bias in AI systems?

- Introduce the concept of Bias in AI: "AI systems can unintentionally learn and perpetuate biases that exist in the data they are trained on, leading to unfair or discriminatory outcomes."

- Explain potential sources of bias in AI:  
  
Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).  
Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).  
Human Bias: Biases of the people who design, develop, and deploy AI systems.

- Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).

- Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).

- Human Bias: Biases of the people who design, develop, and deploy AI systems.

- Examples of AI Bias and its consequences:  
  
Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).  
Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.  
Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.

- Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).

- Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.

- Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.

- "Bias Detection Challenge": Present simplified scenarios or short descriptions of AI systems. Ask students to identify potential sources of bias and possible consequences of biased outcomes in each scenario. Discuss: How can we detect and mitigate bias in AI systems?

- Main Activity 2: "Data Privacy and Transparency in AI" (20 mins):  
  
Discuss Data Privacy Concerns in AI: AI systems often rely on large amounts of personal data. It's crucial to consider how this data is collected, used, stored, and protected. Discuss issues like:  
  
Informed consent for data collection.  
Anonymization and de-identification of data.  
Data security and preventing data breaches.  
Potential for misuse of personal data.  
  
Explain the importance of Transparency and Explainability in AI, especially in critical applications (e.g., healthcare, justice). "Black Box" AI systems (where it's hard to understand how they make decisions) can raise ethical concerns. Transparency means making AI systems more understandable and explainable.  
Examples of ethical considerations related to privacy and transparency:  
  
Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.  
AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.  
Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.  
  
"Ethical AI Principles Brainstorm": In groups, students brainstorm a list of ethical principles or guidelines that should be followed when developing and using AI in data science to address issues of bias, privacy, and transparency. Share and discuss principles as a class.

- Discuss Data Privacy Concerns in AI: AI systems often rely on large amounts of personal data. It's crucial to consider how this data is collected, used, stored, and protected. Discuss issues like:  
  
Informed consent for data collection.  
Anonymization and de-identification of data.  
Data security and preventing data breaches.  
Potential for misuse of personal data.

- Informed consent for data collection.

- Anonymization and de-identification of data.

- Data security and preventing data breaches.

- Potential for misuse of personal data.

- Explain the importance of Transparency and Explainability in AI, especially in critical applications (e.g., healthcare, justice). "Black Box" AI systems (where it's hard to understand how they make decisions) can raise ethical concerns. Transparency means making AI systems more understandable and explainable.

- Examples of ethical considerations related to privacy and transparency:  
  
Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.  
AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.  
Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.

- Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.

- AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.

- Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.

- "Ethical AI Principles Brainstorm": In groups, students brainstorm a list of ethical principles or guidelines that should be followed when developing and using AI in data science to address issues of bias, privacy, and transparency. Share and discuss principles as a class.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why are ethical considerations so important in the field of AI and data science?  
What are the potential harms or negative consequences of unethical AI development and use?  
What steps can data scientists, developers, and society take to promote ethical AI and mitigate risks of bias, privacy violations, and lack of transparency?  
What role do you think you, as future data-literate citizens, can play in ensuring AI is used ethically and responsibly?

- Why are ethical considerations so important in the field of AI and data science?

- What are the potential harms or negative consequences of unethical AI development and use?

- What steps can data scientists, developers, and society take to promote ethical AI and mitigate risks of bias, privacy violations, and lack of transparency?

- What role do you think you, as future data-literate citizens, can play in ensuring AI is used ethically and responsibly?

- Assessments:  
  
Formative Task (5 mins): "AI Ethics - Key Terms Check" - Short answer questions defining bias in AI and explaining data privacy concerns. AI-generated questions.  
Observation of student participation in ethical dilemma discussions and brainstorming activities, assessing their engagement with ethical considerations.

- Formative Task (5 mins): "AI Ethics - Key Terms Check" - Short answer questions defining bias in AI and explaining data privacy concerns. AI-generated questions.

- Observation of student participation in ethical dilemma discussions and brainstorming activities, assessing their engagement with ethical considerations.

- Student Workbook:  
  
Explanations of bias in AI, data privacy concerns, and transparency in AI.  
Examples of AI bias and its consequences.  
Ethical dilemma scenarios related to AI in data science.  
"Bias Detection Challenge" worksheet with AI scenarios.  
"Ethical AI Principles Brainstorm" worksheet with prompts.  
Reflection questions on ethical AI and responsible innovation.

- Explanations of bias in AI, data privacy concerns, and transparency in AI.

- Examples of AI bias and its consequences.

- Ethical dilemma scenarios related to AI in data science.

- "Bias Detection Challenge" worksheet with AI scenarios.

- "Ethical AI Principles Brainstorm" worksheet with prompts.

- Reflection questions on ethical AI and responsible innovation.

- Extension Activities:  
  
Students research current ethical debates and discussions around AI in the news or scientific literature (e.g., debates about AI regulation, algorithmic bias in specific AI systems, data privacy laws). Present findings and lead a class discussion.  
Challenge: Design a "Code of Ethics for AI Data Scientists" targeted at Year 8 students, outlining key ethical principles and responsible practices for working with data and AI.

- Students research current ethical debates and discussions around AI in the news or scientific literature (e.g., debates about AI regulation, algorithmic bias in specific AI systems, data privacy laws). Present findings and lead a class discussion.

- Challenge: Design a "Code of Ethics for AI Data Scientists" targeted at Year 8 students, outlining key ethical principles and responsible practices for working with data and AI.

- Introduce the concept of Bias in AI: "AI systems can unintentionally learn and perpetuate biases that exist in the data they are trained on, leading to unfair or discriminatory outcomes."

- Explain potential sources of bias in AI:  
  
Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).  
Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).  
Human Bias: Biases of the people who design, develop, and deploy AI systems.

- Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).

- Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).

- Human Bias: Biases of the people who design, develop, and deploy AI systems.

- Examples of AI Bias and its consequences:  
  
Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).  
Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.  
Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.

- Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).

- Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.

- Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.

- "Bias Detection Challenge": Present simplified scenarios or short descriptions of AI systems. Ask students to identify potential sources of bias and possible consequences of biased outcomes in each scenario. Discuss: How can we detect and mitigate bias in AI systems?

- Data Bias: Biases present in the training data itself (e.g., historical data reflecting societal biases, underrepresentation of certain groups in datasets).

- Algorithm Bias: Biases introduced by the design of the AI algorithm itself (less common but possible).

- Human Bias: Biases of the people who design, develop, and deploy AI systems.

- Facial Recognition Bias: AI facial recognition systems being less accurate in recognizing faces of people with darker skin tones due to biased training data. (Show visual examples if appropriate and sensitive).

- Hiring Algorithm Bias: AI hiring tools inadvertently discriminating against certain demographic groups based on patterns learned from biased historical hiring data.

- Loan Application Bias: AI loan approval systems showing bias against certain groups, perpetuating existing inequalities.

- Discuss Data Privacy Concerns in AI: AI systems often rely on large amounts of personal data. It's crucial to consider how this data is collected, used, stored, and protected. Discuss issues like:  
  
Informed consent for data collection.  
Anonymization and de-identification of data.  
Data security and preventing data breaches.  
Potential for misuse of personal data.

- Informed consent for data collection.

- Anonymization and de-identification of data.

- Data security and preventing data breaches.

- Potential for misuse of personal data.

- Explain the importance of Transparency and Explainability in AI, especially in critical applications (e.g., healthcare, justice). "Black Box" AI systems (where it's hard to understand how they make decisions) can raise ethical concerns. Transparency means making AI systems more understandable and explainable.

- Examples of ethical considerations related to privacy and transparency:  
  
Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.  
AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.  
Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.

- Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.

- AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.

- Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.

- "Ethical AI Principles Brainstorm": In groups, students brainstorm a list of ethical principles or guidelines that should be followed when developing and using AI in data science to address issues of bias, privacy, and transparency. Share and discuss principles as a class.

- Informed consent for data collection.

- Anonymization and de-identification of data.

- Data security and preventing data breaches.

- Potential for misuse of personal data.

- Use of AI in Healthcare Diagnosis: Ethical need for transparency and explainability in AI diagnostic tools so doctors and patients can understand the reasoning behind AI recommendations.

- AI in Criminal Justice (Risk Assessment Tools): Ethical concerns about using "black box" AI algorithms to assess criminal risk without transparency about how these algorithms work and potential biases.

- Data Collection for AI Surveillance Systems: Ethical considerations around privacy and potential for misuse of large-scale AI-powered surveillance.

- Why are ethical considerations so important in the field of AI and data science?

- What are the potential harms or negative consequences of unethical AI development and use?

- What steps can data scientists, developers, and society take to promote ethical AI and mitigate risks of bias, privacy violations, and lack of transparency?

- What role do you think you, as future data-literate citizens, can play in ensuring AI is used ethically and responsibly?

- Formative Task (5 mins): "AI Ethics - Key Terms Check" - Short answer questions defining bias in AI and explaining data privacy concerns. AI-generated questions.

- Observation of student participation in ethical dilemma discussions and brainstorming activities, assessing their engagement with ethical considerations.

- Explanations of bias in AI, data privacy concerns, and transparency in AI.

- Examples of AI bias and its consequences.

- Ethical dilemma scenarios related to AI in data science.

- "Bias Detection Challenge" worksheet with AI scenarios.

- "Ethical AI Principles Brainstorm" worksheet with prompts.

- Reflection questions on ethical AI and responsible innovation.

- Students research current ethical debates and discussions around AI in the news or scientific literature (e.g., debates about AI regulation, algorithmic bias in specific AI systems, data privacy laws). Present findings and lead a class discussion.

- Challenge: Design a "Code of Ethics for AI Data Scientists" targeted at Year 8 students, outlining key ethical principles and responsible practices for working with data and AI.

- Slide Deck: Introduction to AI ethics, definition of bias in AI, sources and consequences of bias, data privacy concerns, transparency and explainability, responsible AI development.

- Handout: Ethical dilemma scenarios related to AI, "Bias Detection Challenge" worksheet, "Ethical AI Principles Brainstorm" worksheet, definitions of ethical AI concepts.

- Formative Quiz (printable or digital).

- (Optional) Short video clips or news reports related to AI ethics debates.

Okay, here are the detailed lesson plans for Weeks 8-10, completing the comprehensive 10-week Year 8 Data Science curriculum.

### Week 8: Group Project - Real-World Data Science Challenge

### Lesson 8.1: Project Launch: Defining Problems and Choosing Datasets

- Title: Challenge Accepted: Launching Your Real-World Data Science Project

- Learning Outcomes:  
  
Students will be able to collaboratively brainstorm and define a real-world scientific or societal problem that can be investigated using data science.  
Students will be able to work in teams to select a specific project topic and formulate a clear project question.  
Students will be able to identify and evaluate potential datasets relevant to their chosen project topic, considering data availability, quality, and ethical implications.  
Students will be able to outline a basic project proposal, including team roles, project question, and initial dataset ideas.  
NSW Curriculum Outcomes: SC4-WS-07 (identifies problem-solving strategies and proposes solutions), SC4-WS-08 (reflecting on scientific methods), ACSIS124 (Formulating questions), ACSIS125 (Planning investigations).

- Students will be able to collaboratively brainstorm and define a real-world scientific or societal problem that can be investigated using data science.

- Students will be able to work in teams to select a specific project topic and formulate a clear project question.

- Students will be able to identify and evaluate potential datasets relevant to their chosen project topic, considering data availability, quality, and ethical implications.

- Students will be able to outline a basic project proposal, including team roles, project question, and initial dataset ideas.

- NSW Curriculum Outcomes: SC4-WS-07 (identifies problem-solving strategies and proposes solutions), SC4-WS-08 (reflecting on scientific methods), ACSIS124 (Formulating questions), ACSIS125 (Planning investigations).

- Overview: This lesson marks the launch of the group data science project. Students will collaboratively brainstorm real-world problems, choose project topics, formulate research questions, and begin exploring potential datasets. This lesson emphasizes teamwork, problem definition, and initial project planning.

- Activities:  
  
Warm-Up (10 mins): "Real-World Problems Brainstorm" - Class brainstorm: "What are some important problems or challenges facing our world or our community today that might be investigated or addressed using data?" (Examples: environmental issues, health concerns, social issues, technological challenges, questions in a specific scientific domain like biology, physics, chemistry). Write brainstormed problems on the board. Encourage a wide range of ideas.  
Main Activity 1: "Problem Definition and Project Topic Selection" (25 mins):  
  
Divide students into pre-assigned groups (or allow self-selection into small teams of 3-4 students).  
Explain the group project: Students will work in teams to apply data science skills learned in the course to investigate a real-world problem using data. They will go through the entire data science process – from question formulation to data analysis, visualization, and communication of findings.  
Guide group discussion to narrow down the brainstormed problem list and choose a specific project topic. Encourage groups to consider:  
  
Interest: Choose a topic that genuinely interests the team members.  
Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.  
Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).  
Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.  
  
Each group task: Decide on a specific project topic from the brainstormed list or a related topic. Define the problem they want to investigate in more detail. Formulate a clear and focused project question related to their chosen topic. Project question should be answerable using data science methods.  
Groups share their chosen project topics and project questions with the class. Class feedback and discussion – brief peer feedback on topic choices and question clarity.  
  
Main Activity 2: "Dataset Exploration and Evaluation" (20 mins):  
  
Introduce the concept of datasets as the foundation for data science projects. Explain that for their projects, they will need to find and use real-world datasets.  
Provide a list of potential data sources (or guidance on how to find data sources):  
  
Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).  
Scientific data repositories (e.g., for environmental data, health data, biodiversity data).  
Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).  
School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).  
  
Guide groups to start exploring potential datasets related to their chosen project topics. Encourage them to consider:  
  
Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?  
Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).  
Data Relevance: Does the dataset contain variables and information that can help them answer their project question?  
Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?  
  
Each group task: Identify at least 2-3 potential datasets that could be used for their project. Briefly evaluate each dataset based on the criteria above. Choose one primary dataset to focus on initially.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are the most challenging aspects of defining a real-world data science project?  
What are some important factors to consider when choosing a project topic and formulating a project question?  
Why is dataset selection a crucial step in a data science project?  
What are some ethical considerations to keep in mind when working with real-world datasets for projects?  
  
Assessments:  
  
Formative Check (5 mins): Briefly check each group's chosen project topic and project question for clarity and feasibility.  
Review of group notes on potential datasets and their initial evaluations – assess their consideration of data availability, quality, and relevance.  
  
Student Workbook:  
  
Project guidelines and overview.  
Brainstorming prompts for real-world problems and project topics.  
Worksheet for defining project topic, formulating project question, and outlining project proposal.  
List of potential data sources and data evaluation criteria checklist.  
"Dataset Exploration and Evaluation" worksheet to document potential datasets and their assessment.  
  
Extension Activities:  
  
Students research examples of successful data science projects that address real-world problems in the area of their chosen topic. Present examples to the class and discuss the impact of these projects.  
Challenge: For a chosen project topic, explore APIs (Application Programming Interfaces) that could provide real-time or dynamic data for their project (e.g., weather APIs, environmental sensor data APIs – conceptually introduced).

- Warm-Up (10 mins): "Real-World Problems Brainstorm" - Class brainstorm: "What are some important problems or challenges facing our world or our community today that might be investigated or addressed using data?" (Examples: environmental issues, health concerns, social issues, technological challenges, questions in a specific scientific domain like biology, physics, chemistry). Write brainstormed problems on the board. Encourage a wide range of ideas.

- Main Activity 1: "Problem Definition and Project Topic Selection" (25 mins):  
  
Divide students into pre-assigned groups (or allow self-selection into small teams of 3-4 students).  
Explain the group project: Students will work in teams to apply data science skills learned in the course to investigate a real-world problem using data. They will go through the entire data science process – from question formulation to data analysis, visualization, and communication of findings.  
Guide group discussion to narrow down the brainstormed problem list and choose a specific project topic. Encourage groups to consider:  
  
Interest: Choose a topic that genuinely interests the team members.  
Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.  
Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).  
Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.  
  
Each group task: Decide on a specific project topic from the brainstormed list or a related topic. Define the problem they want to investigate in more detail. Formulate a clear and focused project question related to their chosen topic. Project question should be answerable using data science methods.  
Groups share their chosen project topics and project questions with the class. Class feedback and discussion – brief peer feedback on topic choices and question clarity.

- Divide students into pre-assigned groups (or allow self-selection into small teams of 3-4 students).

- Explain the group project: Students will work in teams to apply data science skills learned in the course to investigate a real-world problem using data. They will go through the entire data science process – from question formulation to data analysis, visualization, and communication of findings.

- Guide group discussion to narrow down the brainstormed problem list and choose a specific project topic. Encourage groups to consider:  
  
Interest: Choose a topic that genuinely interests the team members.  
Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.  
Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).  
Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.

- Interest: Choose a topic that genuinely interests the team members.

- Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.

- Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).

- Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.

- Each group task: Decide on a specific project topic from the brainstormed list or a related topic. Define the problem they want to investigate in more detail. Formulate a clear and focused project question related to their chosen topic. Project question should be answerable using data science methods.

- Groups share their chosen project topics and project questions with the class. Class feedback and discussion – brief peer feedback on topic choices and question clarity.

- Main Activity 2: "Dataset Exploration and Evaluation" (20 mins):  
  
Introduce the concept of datasets as the foundation for data science projects. Explain that for their projects, they will need to find and use real-world datasets.  
Provide a list of potential data sources (or guidance on how to find data sources):  
  
Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).  
Scientific data repositories (e.g., for environmental data, health data, biodiversity data).  
Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).  
School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).  
  
Guide groups to start exploring potential datasets related to their chosen project topics. Encourage them to consider:  
  
Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?  
Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).  
Data Relevance: Does the dataset contain variables and information that can help them answer their project question?  
Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?  
  
Each group task: Identify at least 2-3 potential datasets that could be used for their project. Briefly evaluate each dataset based on the criteria above. Choose one primary dataset to focus on initially.

- Introduce the concept of datasets as the foundation for data science projects. Explain that for their projects, they will need to find and use real-world datasets.

- Provide a list of potential data sources (or guidance on how to find data sources):  
  
Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).  
Scientific data repositories (e.g., for environmental data, health data, biodiversity data).  
Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).  
School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).

- Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).

- Scientific data repositories (e.g., for environmental data, health data, biodiversity data).

- Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).

- School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).

- Guide groups to start exploring potential datasets related to their chosen project topics. Encourage them to consider:  
  
Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?  
Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).  
Data Relevance: Does the dataset contain variables and information that can help them answer their project question?  
Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?

- Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?

- Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).

- Data Relevance: Does the dataset contain variables and information that can help them answer their project question?

- Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?

- Each group task: Identify at least 2-3 potential datasets that could be used for their project. Briefly evaluate each dataset based on the criteria above. Choose one primary dataset to focus on initially.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the most challenging aspects of defining a real-world data science project?  
What are some important factors to consider when choosing a project topic and formulating a project question?  
Why is dataset selection a crucial step in a data science project?  
What are some ethical considerations to keep in mind when working with real-world datasets for projects?

- What are the most challenging aspects of defining a real-world data science project?

- What are some important factors to consider when choosing a project topic and formulating a project question?

- Why is dataset selection a crucial step in a data science project?

- What are some ethical considerations to keep in mind when working with real-world datasets for projects?

- Assessments:  
  
Formative Check (5 mins): Briefly check each group's chosen project topic and project question for clarity and feasibility.  
Review of group notes on potential datasets and their initial evaluations – assess their consideration of data availability, quality, and relevance.

- Formative Check (5 mins): Briefly check each group's chosen project topic and project question for clarity and feasibility.

- Review of group notes on potential datasets and their initial evaluations – assess their consideration of data availability, quality, and relevance.

- Student Workbook:  
  
Project guidelines and overview.  
Brainstorming prompts for real-world problems and project topics.  
Worksheet for defining project topic, formulating project question, and outlining project proposal.  
List of potential data sources and data evaluation criteria checklist.  
"Dataset Exploration and Evaluation" worksheet to document potential datasets and their assessment.

- Project guidelines and overview.

- Brainstorming prompts for real-world problems and project topics.

- Worksheet for defining project topic, formulating project question, and outlining project proposal.

- List of potential data sources and data evaluation criteria checklist.

- "Dataset Exploration and Evaluation" worksheet to document potential datasets and their assessment.

- Extension Activities:  
  
Students research examples of successful data science projects that address real-world problems in the area of their chosen topic. Present examples to the class and discuss the impact of these projects.  
Challenge: For a chosen project topic, explore APIs (Application Programming Interfaces) that could provide real-time or dynamic data for their project (e.g., weather APIs, environmental sensor data APIs – conceptually introduced).

- Students research examples of successful data science projects that address real-world problems in the area of their chosen topic. Present examples to the class and discuss the impact of these projects.

- Challenge: For a chosen project topic, explore APIs (Application Programming Interfaces) that could provide real-time or dynamic data for their project (e.g., weather APIs, environmental sensor data APIs – conceptually introduced).

- Lesson Materials:  
  
Slide Deck: Introduction to group project, project overview, brainstorming real-world problems, project topic selection guidance, dataset exploration and evaluation, project proposal outline.  
Handout: Project guidelines, project proposal template, dataset evaluation checklist, list of potential data sources (or links).  
Computers/Tablets with internet access for dataset research and exploration.

- Slide Deck: Introduction to group project, project overview, brainstorming real-world problems, project topic selection guidance, dataset exploration and evaluation, project proposal outline.

- Handout: Project guidelines, project proposal template, dataset evaluation checklist, list of potential data sources (or links).

- Computers/Tablets with internet access for dataset research and exploration.

- Students will be able to collaboratively brainstorm and define a real-world scientific or societal problem that can be investigated using data science.

- Students will be able to work in teams to select a specific project topic and formulate a clear project question.

- Students will be able to identify and evaluate potential datasets relevant to their chosen project topic, considering data availability, quality, and ethical implications.

- Students will be able to outline a basic project proposal, including team roles, project question, and initial dataset ideas.

- NSW Curriculum Outcomes: SC4-WS-07 (identifies problem-solving strategies and proposes solutions), SC4-WS-08 (reflecting on scientific methods), ACSIS124 (Formulating questions), ACSIS125 (Planning investigations).

- Warm-Up (10 mins): "Real-World Problems Brainstorm" - Class brainstorm: "What are some important problems or challenges facing our world or our community today that might be investigated or addressed using data?" (Examples: environmental issues, health concerns, social issues, technological challenges, questions in a specific scientific domain like biology, physics, chemistry). Write brainstormed problems on the board. Encourage a wide range of ideas.

- Main Activity 1: "Problem Definition and Project Topic Selection" (25 mins):  
  
Divide students into pre-assigned groups (or allow self-selection into small teams of 3-4 students).  
Explain the group project: Students will work in teams to apply data science skills learned in the course to investigate a real-world problem using data. They will go through the entire data science process – from question formulation to data analysis, visualization, and communication of findings.  
Guide group discussion to narrow down the brainstormed problem list and choose a specific project topic. Encourage groups to consider:  
  
Interest: Choose a topic that genuinely interests the team members.  
Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.  
Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).  
Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.  
  
Each group task: Decide on a specific project topic from the brainstormed list or a related topic. Define the problem they want to investigate in more detail. Formulate a clear and focused project question related to their chosen topic. Project question should be answerable using data science methods.  
Groups share their chosen project topics and project questions with the class. Class feedback and discussion – brief peer feedback on topic choices and question clarity.

- Divide students into pre-assigned groups (or allow self-selection into small teams of 3-4 students).

- Explain the group project: Students will work in teams to apply data science skills learned in the course to investigate a real-world problem using data. They will go through the entire data science process – from question formulation to data analysis, visualization, and communication of findings.

- Guide group discussion to narrow down the brainstormed problem list and choose a specific project topic. Encourage groups to consider:  
  
Interest: Choose a topic that genuinely interests the team members.  
Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.  
Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).  
Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.

- Interest: Choose a topic that genuinely interests the team members.

- Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.

- Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).

- Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.

- Each group task: Decide on a specific project topic from the brainstormed list or a related topic. Define the problem they want to investigate in more detail. Formulate a clear and focused project question related to their chosen topic. Project question should be answerable using data science methods.

- Groups share their chosen project topics and project questions with the class. Class feedback and discussion – brief peer feedback on topic choices and question clarity.

- Main Activity 2: "Dataset Exploration and Evaluation" (20 mins):  
  
Introduce the concept of datasets as the foundation for data science projects. Explain that for their projects, they will need to find and use real-world datasets.  
Provide a list of potential data sources (or guidance on how to find data sources):  
  
Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).  
Scientific data repositories (e.g., for environmental data, health data, biodiversity data).  
Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).  
School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).  
  
Guide groups to start exploring potential datasets related to their chosen project topics. Encourage them to consider:  
  
Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?  
Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).  
Data Relevance: Does the dataset contain variables and information that can help them answer their project question?  
Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?  
  
Each group task: Identify at least 2-3 potential datasets that could be used for their project. Briefly evaluate each dataset based on the criteria above. Choose one primary dataset to focus on initially.

- Introduce the concept of datasets as the foundation for data science projects. Explain that for their projects, they will need to find and use real-world datasets.

- Provide a list of potential data sources (or guidance on how to find data sources):  
  
Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).  
Scientific data repositories (e.g., for environmental data, health data, biodiversity data).  
Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).  
School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).

- Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).

- Scientific data repositories (e.g., for environmental data, health data, biodiversity data).

- Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).

- School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).

- Guide groups to start exploring potential datasets related to their chosen project topics. Encourage them to consider:  
  
Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?  
Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).  
Data Relevance: Does the dataset contain variables and information that can help them answer their project question?  
Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?

- Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?

- Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).

- Data Relevance: Does the dataset contain variables and information that can help them answer their project question?

- Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?

- Each group task: Identify at least 2-3 potential datasets that could be used for their project. Briefly evaluate each dataset based on the criteria above. Choose one primary dataset to focus on initially.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the most challenging aspects of defining a real-world data science project?  
What are some important factors to consider when choosing a project topic and formulating a project question?  
Why is dataset selection a crucial step in a data science project?  
What are some ethical considerations to keep in mind when working with real-world datasets for projects?

- What are the most challenging aspects of defining a real-world data science project?

- What are some important factors to consider when choosing a project topic and formulating a project question?

- Why is dataset selection a crucial step in a data science project?

- What are some ethical considerations to keep in mind when working with real-world datasets for projects?

- Assessments:  
  
Formative Check (5 mins): Briefly check each group's chosen project topic and project question for clarity and feasibility.  
Review of group notes on potential datasets and their initial evaluations – assess their consideration of data availability, quality, and relevance.

- Formative Check (5 mins): Briefly check each group's chosen project topic and project question for clarity and feasibility.

- Review of group notes on potential datasets and their initial evaluations – assess their consideration of data availability, quality, and relevance.

- Student Workbook:  
  
Project guidelines and overview.  
Brainstorming prompts for real-world problems and project topics.  
Worksheet for defining project topic, formulating project question, and outlining project proposal.  
List of potential data sources and data evaluation criteria checklist.  
"Dataset Exploration and Evaluation" worksheet to document potential datasets and their assessment.

- Project guidelines and overview.

- Brainstorming prompts for real-world problems and project topics.

- Worksheet for defining project topic, formulating project question, and outlining project proposal.

- List of potential data sources and data evaluation criteria checklist.

- "Dataset Exploration and Evaluation" worksheet to document potential datasets and their assessment.

- Extension Activities:  
  
Students research examples of successful data science projects that address real-world problems in the area of their chosen topic. Present examples to the class and discuss the impact of these projects.  
Challenge: For a chosen project topic, explore APIs (Application Programming Interfaces) that could provide real-time or dynamic data for their project (e.g., weather APIs, environmental sensor data APIs – conceptually introduced).

- Students research examples of successful data science projects that address real-world problems in the area of their chosen topic. Present examples to the class and discuss the impact of these projects.

- Challenge: For a chosen project topic, explore APIs (Application Programming Interfaces) that could provide real-time or dynamic data for their project (e.g., weather APIs, environmental sensor data APIs – conceptually introduced).

- Divide students into pre-assigned groups (or allow self-selection into small teams of 3-4 students).

- Explain the group project: Students will work in teams to apply data science skills learned in the course to investigate a real-world problem using data. They will go through the entire data science process – from question formulation to data analysis, visualization, and communication of findings.

- Guide group discussion to narrow down the brainstormed problem list and choose a specific project topic. Encourage groups to consider:  
  
Interest: Choose a topic that genuinely interests the team members.  
Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.  
Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).  
Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.

- Interest: Choose a topic that genuinely interests the team members.

- Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.

- Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).

- Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.

- Each group task: Decide on a specific project topic from the brainstormed list or a related topic. Define the problem they want to investigate in more detail. Formulate a clear and focused project question related to their chosen topic. Project question should be answerable using data science methods.

- Groups share their chosen project topics and project questions with the class. Class feedback and discussion – brief peer feedback on topic choices and question clarity.

- Interest: Choose a topic that genuinely interests the team members.

- Relevance: Choose a topic that is relevant to real-world issues or scientific concepts.

- Feasibility: Consider if data is likely to be available for the chosen topic (at least potentially).

- Scope: Choose a topic that is manageable within the project timeframe and Year 8 skill level.

- Introduce the concept of datasets as the foundation for data science projects. Explain that for their projects, they will need to find and use real-world datasets.

- Provide a list of potential data sources (or guidance on how to find data sources):  
  
Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).  
Scientific data repositories (e.g., for environmental data, health data, biodiversity data).  
Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).  
School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).

- Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).

- Scientific data repositories (e.g., for environmental data, health data, biodiversity data).

- Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).

- School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).

- Guide groups to start exploring potential datasets related to their chosen project topics. Encourage them to consider:  
  
Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?  
Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).  
Data Relevance: Does the dataset contain variables and information that can help them answer their project question?  
Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?

- Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?

- Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).

- Data Relevance: Does the dataset contain variables and information that can help them answer their project question?

- Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?

- Each group task: Identify at least 2-3 potential datasets that could be used for their project. Briefly evaluate each dataset based on the criteria above. Choose one primary dataset to focus on initially.

- Open government data portals (e.g., data.gov, data.gov.au, data.gov.uk).

- Scientific data repositories (e.g., for environmental data, health data, biodiversity data).

- Publicly available datasets on platforms like Kaggle, UCI Machine Learning Repository (for simpler, introductory datasets).

- School or local community data (e.g., school energy consumption data, local weather data, survey data they can collect themselves – if feasible and ethically approved).

- Data Availability: Can they find a dataset that is publicly accessible and relevant to their question?

- Data Quality: Based on initial descriptions or previews, does the dataset seem to be of reasonable quality? (Initial assessment only, more detailed data exploration in next lesson).

- Data Relevance: Does the dataset contain variables and information that can help them answer their project question?

- Ethical Considerations: Are there any ethical concerns related to using this dataset (privacy, sensitive data, etc.)?

- What are the most challenging aspects of defining a real-world data science project?

- What are some important factors to consider when choosing a project topic and formulating a project question?

- Why is dataset selection a crucial step in a data science project?

- What are some ethical considerations to keep in mind when working with real-world datasets for projects?

- Formative Check (5 mins): Briefly check each group's chosen project topic and project question for clarity and feasibility.

- Review of group notes on potential datasets and their initial evaluations – assess their consideration of data availability, quality, and relevance.

- Project guidelines and overview.

- Brainstorming prompts for real-world problems and project topics.

- Worksheet for defining project topic, formulating project question, and outlining project proposal.

- List of potential data sources and data evaluation criteria checklist.

- "Dataset Exploration and Evaluation" worksheet to document potential datasets and their assessment.

- Students research examples of successful data science projects that address real-world problems in the area of their chosen topic. Present examples to the class and discuss the impact of these projects.

- Challenge: For a chosen project topic, explore APIs (Application Programming Interfaces) that could provide real-time or dynamic data for their project (e.g., weather APIs, environmental sensor data APIs – conceptually introduced).

- Slide Deck: Introduction to group project, project overview, brainstorming real-world problems, project topic selection guidance, dataset exploration and evaluation, project proposal outline.

- Handout: Project guidelines, project proposal template, dataset evaluation checklist, list of potential data sources (or links).

- Computers/Tablets with internet access for dataset research and exploration.

### Lesson 8.2: Data Exploration and Question Formulation for Group Projects

- Title: Data Expedition: Exploring Datasets and Defining Project Questions

- Learning Outcomes:  
  
Students will be able to load and explore their chosen project dataset in Observable notebooks using Pandas.  
Students will be able to perform initial data exploration tasks, including examining data structure, data types, descriptive statistics, and visualizing data distributions.  
Students will be able to refine their project question based on their initial dataset exploration, ensuring it is answerable with the available data.  
Students will be able to develop a more detailed project plan, outlining data analysis steps, visualization strategies, and a project timeline.  
NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), ACSIS126 (Processing and analysing data), ACSIS125 (Planning investigations).

- Students will be able to load and explore their chosen project dataset in Observable notebooks using Pandas.

- Students will be able to perform initial data exploration tasks, including examining data structure, data types, descriptive statistics, and visualizing data distributions.

- Students will be able to refine their project question based on their initial dataset exploration, ensuring it is answerable with the available data.

- Students will be able to develop a more detailed project plan, outlining data analysis steps, visualization strategies, and a project timeline.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), ACSIS126 (Processing and analysing data), ACSIS125 (Planning investigations).

- Overview: This lesson is dedicated to in-class group work on project data exploration and question refinement. Students will load their chosen datasets into Observable, conduct initial data exploration using Pandas and basic visualizations, and refine their project questions based on their data insights. They will also develop a more detailed project plan.

- Activities:  
  
Warm-Up (10 mins): "Data Exploration Techniques Review" - Quick review of data exploration techniques learned in the course so far:  
  
Loading data into Pandas DataFrames in Observable.  
Examining DataFrame structure (.head(), .info(), .describe()).  
Calculating descriptive statistics (mean, median, range).  
Creating basic visualizations (bar graphs, line graphs, histograms – review relevant graph types for their project data).  
Sorting and Filtering data (review relevant techniques for focusing data exploration).  
Briefly discuss: Why is data exploration an essential step before starting in-depth analysis? What are you looking for during data exploration?  
  
Main Activity 1: "Group Data Exploration in Observable" (35 mins):  
  
In-class group work session. Each group works on their chosen dataset in Observable notebooks.  
Guide students to perform the following data exploration tasks (provide a checklist or worksheet with prompts):  
  
Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).  
Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.  
Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.  
Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).  
Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).  
Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?  
  
Teacher circulates to provide guidance, answer questions, and ensure groups are effectively exploring their data in Observable.  
  
Main Activity 2: "Project Question Refinement and Project Plan Development" (15 mins):  
  
Based on their data exploration findings, groups revisit their initial project questions.  
Guide groups to refine their project questions to be more specific, focused, and directly answerable using the data they have explored. Project questions may need to be narrowed down or modified based on data availability and insights from exploration.  
Each group task: Refine their project question. Develop a more detailed project plan outlining the next steps:  
  
Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).  
Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).  
Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.  
  
Groups briefly share their refined project questions and project plans with the class. Class feedback and discussion – brief peer feedback on question refinement and project plan feasibility.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
Why is data exploration such an important step in the data science process?  
How did your initial data exploration help you refine your project question? Did you need to change your question based on what you found in the data?  
What are the key elements of a good project plan for a data science project?  
What are you most looking forward to doing in the next stages of your project? What challenges do you anticipate?  
  
Assessments:  
  
Formative Check (5 mins): Briefly review each group's refined project question and project plan for clarity, feasibility, and alignment with data exploration findings.  
Review of student Observable notebooks – assess the extent and quality of their data exploration, documentation of findings in markdown cells, and initial visualizations.  
  
Student Workbook:  
  
Checklist of data exploration tasks to perform in Observable (load data, examine structure, statistics, visualizations, data quality check).  
Worksheet for documenting key findings from data exploration.  
"Project Question Refinement" worksheet with prompts to revisit and refine project questions based on data insights.  
"Project Plan Development" template to outline data analysis plan, visualization plan, and project timeline.  
  
Extension Activities:  
  
Students explore more advanced data exploration techniques in Pandas (e.g., grouping and aggregation, correlation analysis).  
Challenge: For their project dataset, try to identify potential data quality issues in more detail and brainstorm specific data cleaning steps that might be needed in the next project stage.

- Warm-Up (10 mins): "Data Exploration Techniques Review" - Quick review of data exploration techniques learned in the course so far:  
  
Loading data into Pandas DataFrames in Observable.  
Examining DataFrame structure (.head(), .info(), .describe()).  
Calculating descriptive statistics (mean, median, range).  
Creating basic visualizations (bar graphs, line graphs, histograms – review relevant graph types for their project data).  
Sorting and Filtering data (review relevant techniques for focusing data exploration).  
Briefly discuss: Why is data exploration an essential step before starting in-depth analysis? What are you looking for during data exploration?

- Loading data into Pandas DataFrames in Observable.

- Examining DataFrame structure (.head(), .info(), .describe()).

- Calculating descriptive statistics (mean, median, range).

- Creating basic visualizations (bar graphs, line graphs, histograms – review relevant graph types for their project data).

- Sorting and Filtering data (review relevant techniques for focusing data exploration).

- Briefly discuss: Why is data exploration an essential step before starting in-depth analysis? What are you looking for during data exploration?

- Main Activity 1: "Group Data Exploration in Observable" (35 mins):  
  
In-class group work session. Each group works on their chosen dataset in Observable notebooks.  
Guide students to perform the following data exploration tasks (provide a checklist or worksheet with prompts):  
  
Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).  
Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.  
Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.  
Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).  
Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).  
Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?  
  
Teacher circulates to provide guidance, answer questions, and ensure groups are effectively exploring their data in Observable.

- In-class group work session. Each group works on their chosen dataset in Observable notebooks.

- Guide students to perform the following data exploration tasks (provide a checklist or worksheet with prompts):  
  
Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).  
Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.  
Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.  
Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).  
Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).  
Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?

- Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).

- Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.

- Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.

- Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).

- Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).

- Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?

- Teacher circulates to provide guidance, answer questions, and ensure groups are effectively exploring their data in Observable.

- Main Activity 2: "Project Question Refinement and Project Plan Development" (15 mins):  
  
Based on their data exploration findings, groups revisit their initial project questions.  
Guide groups to refine their project questions to be more specific, focused, and directly answerable using the data they have explored. Project questions may need to be narrowed down or modified based on data availability and insights from exploration.  
Each group task: Refine their project question. Develop a more detailed project plan outlining the next steps:  
  
Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).  
Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).  
Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.  
  
Groups briefly share their refined project questions and project plans with the class. Class feedback and discussion – brief peer feedback on question refinement and project plan feasibility.

- Based on their data exploration findings, groups revisit their initial project questions.

- Guide groups to refine their project questions to be more specific, focused, and directly answerable using the data they have explored. Project questions may need to be narrowed down or modified based on data availability and insights from exploration.

- Each group task: Refine their project question. Develop a more detailed project plan outlining the next steps:  
  
Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).  
Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).  
Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.

- Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).

- Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).

- Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.

- Groups briefly share their refined project questions and project plans with the class. Class feedback and discussion – brief peer feedback on question refinement and project plan feasibility.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is data exploration such an important step in the data science process?  
How did your initial data exploration help you refine your project question? Did you need to change your question based on what you found in the data?  
What are the key elements of a good project plan for a data science project?  
What are you most looking forward to doing in the next stages of your project? What challenges do you anticipate?

- Why is data exploration such an important step in the data science process?

- How did your initial data exploration help you refine your project question? Did you need to change your question based on what you found in the data?

- What are the key elements of a good project plan for a data science project?

- What are you most looking forward to doing in the next stages of your project? What challenges do you anticipate?

- Assessments:  
  
Formative Check (5 mins): Briefly review each group's refined project question and project plan for clarity, feasibility, and alignment with data exploration findings.  
Review of student Observable notebooks – assess the extent and quality of their data exploration, documentation of findings in markdown cells, and initial visualizations.

- Formative Check (5 mins): Briefly review each group's refined project question and project plan for clarity, feasibility, and alignment with data exploration findings.

- Review of student Observable notebooks – assess the extent and quality of their data exploration, documentation of findings in markdown cells, and initial visualizations.

- Student Workbook:  
  
Checklist of data exploration tasks to perform in Observable (load data, examine structure, statistics, visualizations, data quality check).  
Worksheet for documenting key findings from data exploration.  
"Project Question Refinement" worksheet with prompts to revisit and refine project questions based on data insights.  
"Project Plan Development" template to outline data analysis plan, visualization plan, and project timeline.

- Checklist of data exploration tasks to perform in Observable (load data, examine structure, statistics, visualizations, data quality check).

- Worksheet for documenting key findings from data exploration.

- "Project Question Refinement" worksheet with prompts to revisit and refine project questions based on data insights.

- "Project Plan Development" template to outline data analysis plan, visualization plan, and project timeline.

- Extension Activities:  
  
Students explore more advanced data exploration techniques in Pandas (e.g., grouping and aggregation, correlation analysis).  
Challenge: For their project dataset, try to identify potential data quality issues in more detail and brainstorm specific data cleaning steps that might be needed in the next project stage.

- Students explore more advanced data exploration techniques in Pandas (e.g., grouping and aggregation, correlation analysis).

- Challenge: For their project dataset, try to identify potential data quality issues in more detail and brainstorm specific data cleaning steps that might be needed in the next project stage.

- Lesson Materials:  
  
Slide Deck: Review of data exploration techniques, step-by-step guide for data exploration in Observable, project question refinement guidance, project plan development outline.  
Handout: Checklist of data exploration tasks, "Project Question Refinement" worksheet, "Project Plan Development" template.  
Computers/Tablets with internet access for Observable and project datasets.

- Slide Deck: Review of data exploration techniques, step-by-step guide for data exploration in Observable, project question refinement guidance, project plan development outline.

- Handout: Checklist of data exploration tasks, "Project Question Refinement" worksheet, "Project Plan Development" template.

- Computers/Tablets with internet access for Observable and project datasets.

- Students will be able to load and explore their chosen project dataset in Observable notebooks using Pandas.

- Students will be able to perform initial data exploration tasks, including examining data structure, data types, descriptive statistics, and visualizing data distributions.

- Students will be able to refine their project question based on their initial dataset exploration, ensuring it is answerable with the available data.

- Students will be able to develop a more detailed project plan, outlining data analysis steps, visualization strategies, and a project timeline.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), ACSIS126 (Processing and analysing data), ACSIS125 (Planning investigations).

- Warm-Up (10 mins): "Data Exploration Techniques Review" - Quick review of data exploration techniques learned in the course so far:  
  
Loading data into Pandas DataFrames in Observable.  
Examining DataFrame structure (.head(), .info(), .describe()).  
Calculating descriptive statistics (mean, median, range).  
Creating basic visualizations (bar graphs, line graphs, histograms – review relevant graph types for their project data).  
Sorting and Filtering data (review relevant techniques for focusing data exploration).  
Briefly discuss: Why is data exploration an essential step before starting in-depth analysis? What are you looking for during data exploration?

- Loading data into Pandas DataFrames in Observable.

- Examining DataFrame structure (.head(), .info(), .describe()).

- Calculating descriptive statistics (mean, median, range).

- Creating basic visualizations (bar graphs, line graphs, histograms – review relevant graph types for their project data).

- Sorting and Filtering data (review relevant techniques for focusing data exploration).

- Briefly discuss: Why is data exploration an essential step before starting in-depth analysis? What are you looking for during data exploration?

- Main Activity 1: "Group Data Exploration in Observable" (35 mins):  
  
In-class group work session. Each group works on their chosen dataset in Observable notebooks.  
Guide students to perform the following data exploration tasks (provide a checklist or worksheet with prompts):  
  
Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).  
Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.  
Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.  
Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).  
Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).  
Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?  
  
Teacher circulates to provide guidance, answer questions, and ensure groups are effectively exploring their data in Observable.

- In-class group work session. Each group works on their chosen dataset in Observable notebooks.

- Guide students to perform the following data exploration tasks (provide a checklist or worksheet with prompts):  
  
Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).  
Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.  
Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.  
Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).  
Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).  
Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?

- Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).

- Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.

- Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.

- Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).

- Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).

- Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?

- Teacher circulates to provide guidance, answer questions, and ensure groups are effectively exploring their data in Observable.

- Main Activity 2: "Project Question Refinement and Project Plan Development" (15 mins):  
  
Based on their data exploration findings, groups revisit their initial project questions.  
Guide groups to refine their project questions to be more specific, focused, and directly answerable using the data they have explored. Project questions may need to be narrowed down or modified based on data availability and insights from exploration.  
Each group task: Refine their project question. Develop a more detailed project plan outlining the next steps:  
  
Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).  
Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).  
Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.  
  
Groups briefly share their refined project questions and project plans with the class. Class feedback and discussion – brief peer feedback on question refinement and project plan feasibility.

- Based on their data exploration findings, groups revisit their initial project questions.

- Guide groups to refine their project questions to be more specific, focused, and directly answerable using the data they have explored. Project questions may need to be narrowed down or modified based on data availability and insights from exploration.

- Each group task: Refine their project question. Develop a more detailed project plan outlining the next steps:  
  
Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).  
Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).  
Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.

- Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).

- Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).

- Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.

- Groups briefly share their refined project questions and project plans with the class. Class feedback and discussion – brief peer feedback on question refinement and project plan feasibility.

- Reflection & Discussion (10 mins): Class discussion:  
  
Why is data exploration such an important step in the data science process?  
How did your initial data exploration help you refine your project question? Did you need to change your question based on what you found in the data?  
What are the key elements of a good project plan for a data science project?  
What are you most looking forward to doing in the next stages of your project? What challenges do you anticipate?

- Why is data exploration such an important step in the data science process?

- How did your initial data exploration help you refine your project question? Did you need to change your question based on what you found in the data?

- What are the key elements of a good project plan for a data science project?

- What are you most looking forward to doing in the next stages of your project? What challenges do you anticipate?

- Assessments:  
  
Formative Check (5 mins): Briefly review each group's refined project question and project plan for clarity, feasibility, and alignment with data exploration findings.  
Review of student Observable notebooks – assess the extent and quality of their data exploration, documentation of findings in markdown cells, and initial visualizations.

- Formative Check (5 mins): Briefly review each group's refined project question and project plan for clarity, feasibility, and alignment with data exploration findings.

- Review of student Observable notebooks – assess the extent and quality of their data exploration, documentation of findings in markdown cells, and initial visualizations.

- Student Workbook:  
  
Checklist of data exploration tasks to perform in Observable (load data, examine structure, statistics, visualizations, data quality check).  
Worksheet for documenting key findings from data exploration.  
"Project Question Refinement" worksheet with prompts to revisit and refine project questions based on data insights.  
"Project Plan Development" template to outline data analysis plan, visualization plan, and project timeline.

- Checklist of data exploration tasks to perform in Observable (load data, examine structure, statistics, visualizations, data quality check).

- Worksheet for documenting key findings from data exploration.

- "Project Question Refinement" worksheet with prompts to revisit and refine project questions based on data insights.

- "Project Plan Development" template to outline data analysis plan, visualization plan, and project timeline.

- Extension Activities:  
  
Students explore more advanced data exploration techniques in Pandas (e.g., grouping and aggregation, correlation analysis).  
Challenge: For their project dataset, try to identify potential data quality issues in more detail and brainstorm specific data cleaning steps that might be needed in the next project stage.

- Students explore more advanced data exploration techniques in Pandas (e.g., grouping and aggregation, correlation analysis).

- Challenge: For their project dataset, try to identify potential data quality issues in more detail and brainstorm specific data cleaning steps that might be needed in the next project stage.

- Loading data into Pandas DataFrames in Observable.

- Examining DataFrame structure (.head(), .info(), .describe()).

- Calculating descriptive statistics (mean, median, range).

- Creating basic visualizations (bar graphs, line graphs, histograms – review relevant graph types for their project data).

- Sorting and Filtering data (review relevant techniques for focusing data exploration).

- Briefly discuss: Why is data exploration an essential step before starting in-depth analysis? What are you looking for during data exploration?

- In-class group work session. Each group works on their chosen dataset in Observable notebooks.

- Guide students to perform the following data exploration tasks (provide a checklist or worksheet with prompts):  
  
Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).  
Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.  
Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.  
Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).  
Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).  
Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?

- Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).

- Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.

- Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.

- Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).

- Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).

- Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?

- Teacher circulates to provide guidance, answer questions, and ensure groups are effectively exploring their data in Observable.

- Load Dataset: Load their chosen dataset into a Pandas DataFrame in an Observable notebook (using code to read CSV, Excel, or other data file format).

- Examine Data Structure: Use .head(), .tail(), .sample() to view the first/last few rows and random samples of the data. Use .info() to check data types, column names, and missing values. Use .shape to check dataset dimensions.

- Descriptive Statistics: Use .describe() to calculate descriptive statistics for relevant numerical columns. Calculate mean, median, range for key variables using Pandas functions.

- Data Visualization (Initial): Create basic visualizations in Observable using Plot library (or other suitable libraries) to get an initial visual overview of the data distributions and relationships. Suggest relevant graph types based on their data types and project questions (histograms for distributions, scatter plots for relationships between numerical variables, bar graphs for categorical data summaries).

- Data Quality Check (Initial): Based on their exploration, note down any initial observations about data quality: Are there missing values? Are data formats consistent? Are there any obvious errors or outliers? (More detailed data cleaning and quality control in later project stages).

- Record Key Findings: In markdown cells in their Observable notebooks, groups document their key findings from data exploration. What are the main variables in the dataset? What do initial statistics and visualizations reveal about the data? What are some interesting or potentially relevant patterns or trends they observe?

- Based on their data exploration findings, groups revisit their initial project questions.

- Guide groups to refine their project questions to be more specific, focused, and directly answerable using the data they have explored. Project questions may need to be narrowed down or modified based on data availability and insights from exploration.

- Each group task: Refine their project question. Develop a more detailed project plan outlining the next steps:  
  
Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).  
Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).  
Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.

- Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).

- Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).

- Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.

- Groups briefly share their refined project questions and project plans with the class. Class feedback and discussion – brief peer feedback on question refinement and project plan feasibility.

- Data Analysis Plan: What specific data analysis techniques will they use to answer their refined project question? (e.g., descriptive statistics, comparisons between groups, correlation analysis – based on course content).

- Visualization Plan: What types of visualizations will they create to communicate their findings effectively? (Tables, bar graphs, line graphs, scatter plots – choose appropriate types for their data and message).

- Project Timeline: Create a basic timeline for completing the remaining project tasks (data cleaning, in-depth analysis, visualization, presentation preparation). Assign roles within the team for different project tasks.

- Why is data exploration such an important step in the data science process?

- How did your initial data exploration help you refine your project question? Did you need to change your question based on what you found in the data?

- What are the key elements of a good project plan for a data science project?

- What are you most looking forward to doing in the next stages of your project? What challenges do you anticipate?

- Formative Check (5 mins): Briefly review each group's refined project question and project plan for clarity, feasibility, and alignment with data exploration findings.

- Review of student Observable notebooks – assess the extent and quality of their data exploration, documentation of findings in markdown cells, and initial visualizations.

- Checklist of data exploration tasks to perform in Observable (load data, examine structure, statistics, visualizations, data quality check).

- Worksheet for documenting key findings from data exploration.

- "Project Question Refinement" worksheet with prompts to revisit and refine project questions based on data insights.

- "Project Plan Development" template to outline data analysis plan, visualization plan, and project timeline.

- Students explore more advanced data exploration techniques in Pandas (e.g., grouping and aggregation, correlation analysis).

- Challenge: For their project dataset, try to identify potential data quality issues in more detail and brainstorm specific data cleaning steps that might be needed in the next project stage.

- Slide Deck: Review of data exploration techniques, step-by-step guide for data exploration in Observable, project question refinement guidance, project plan development outline.

- Handout: Checklist of data exploration tasks, "Project Question Refinement" worksheet, "Project Plan Development" template.

- Computers/Tablets with internet access for Observable and project datasets.

### Lesson 8.3: Data Analysis and Visualization - Project Work Session

- Title: Data in Action: Analyzing and Visualizing Project Data

- Learning Outcomes:  
  
Students will be able to perform relevant data analysis techniques on their project datasets using Python and Pandas in Observable notebooks, guided by their project plans.  
Students will be able to create effective data visualizations (tables, graphs) in Observable to communicate their project findings, choosing appropriate visualization types for their data and message.  
Students will be able to collaboratively work within their groups to analyse data, create visualizations, and document their project progress in Observable notebooks.  
NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), ACSIS126 (Processing and analysing data), ACSIS127 (Communicating findings).

- Students will be able to perform relevant data analysis techniques on their project datasets using Python and Pandas in Observable notebooks, guided by their project plans.

- Students will be able to create effective data visualizations (tables, graphs) in Observable to communicate their project findings, choosing appropriate visualization types for their data and message.

- Students will be able to collaboratively work within their groups to analyse data, create visualizations, and document their project progress in Observable notebooks.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), ACSIS126 (Processing and analysing data), ACSIS127 (Communicating findings).

- Overview: This lesson is a dedicated in-class work session focused on data analysis and visualization for the group projects. Students will work within their teams to implement their data analysis plans in Observable, create visualizations to answer their project questions, and document their progress. Teacher facilitates and provides support.

- Activities:  
  
Warm-Up (10 mins): "Data Analysis and Visualization Techniques - Quick Recap" - Briefly review the data analysis and visualization techniques learned in the course that are most relevant to the projects:  
  
Descriptive Statistics (mean, median, range).  
Data Sorting and Filtering.  
Bar Graphs (for categorical comparisons).  
Line Graphs (for trends over time or continuous variables).  
Scatter Plots (for relationships between numerical variables).  
Tables (for summarizing data and presenting key values).  
Quickly remind students of when each technique is most appropriate and how to implement them in Python/Observable.  
  
Main Activity 1: "Group Data Analysis and Visualization Work Session" (60 mins):  
  
In-class group work session. Each group works on their Observable notebooks to implement their data analysis and visualization plans.  
Guide students to focus on:  
  
Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.  
Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.  
  
Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.  
  
Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.  
  
  
Teacher circulates to provide individualized guidance, answer questions, troubleshoot coding issues, and provide feedback on data analysis and visualization approaches. Encourage peer-to-peer support and collaboration within groups.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What data analysis and visualization techniques did you use in your project work session today?  
What were some of the key findings or interesting patterns you discovered in your data analysis so far?  
What were some challenges you encountered during data analysis or visualization? How did you overcome them or plan to address them?  
What are your next steps for your project? What do you plan to focus on in the remaining project time?  
  
Assessments:  
  
Formative Check (5 mins): Briefly check each group's Observable notebook to assess their progress in data analysis and visualization. Review the code they have written and the visualizations they have created. Provide quick feedback and suggestions.  
Observation of group work during the session – assess their collaboration, problem-solving, and engagement with data analysis and visualization tasks.  
  
Student Workbook:  
  
Checklist of data analysis and visualization tasks to complete for the project.  
Prompts for documenting data analysis steps, findings, and visualization interpretations in Observable notebooks.  
Guidance on choosing appropriate visualization types for different data and messages.  
Project progress tracking section to record completed tasks, challenges, and next steps.  
  
Extension Activities:  
  
Students explore more advanced data analysis techniques that might be relevant to their project (e.g., correlation analysis in more detail, basic statistical tests – conceptually introduced if appropriate).  
Challenge: Experiment with adding interactive elements to their visualizations in Observable (e.g., tooltips, sliders, dropdown menus) to create more dynamic and informative dashboards or interactive visualizations.

- Warm-Up (10 mins): "Data Analysis and Visualization Techniques - Quick Recap" - Briefly review the data analysis and visualization techniques learned in the course that are most relevant to the projects:  
  
Descriptive Statistics (mean, median, range).  
Data Sorting and Filtering.  
Bar Graphs (for categorical comparisons).  
Line Graphs (for trends over time or continuous variables).  
Scatter Plots (for relationships between numerical variables).  
Tables (for summarizing data and presenting key values).  
Quickly remind students of when each technique is most appropriate and how to implement them in Python/Observable.

- Descriptive Statistics (mean, median, range).

- Data Sorting and Filtering.

- Bar Graphs (for categorical comparisons).

- Line Graphs (for trends over time or continuous variables).

- Scatter Plots (for relationships between numerical variables).

- Tables (for summarizing data and presenting key values).

- Quickly remind students of when each technique is most appropriate and how to implement them in Python/Observable.

- Main Activity 1: "Group Data Analysis and Visualization Work Session" (60 mins):  
  
In-class group work session. Each group works on their Observable notebooks to implement their data analysis and visualization plans.  
Guide students to focus on:  
  
Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.  
Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.  
  
Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.  
  
Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.  
  
  
Teacher circulates to provide individualized guidance, answer questions, troubleshoot coding issues, and provide feedback on data analysis and visualization approaches. Encourage peer-to-peer support and collaboration within groups.

- In-class group work session. Each group works on their Observable notebooks to implement their data analysis and visualization plans.

- Guide students to focus on:  
  
Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.  
Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.  
  
Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.  
  
Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.

- Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.

- Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.

- Calculating descriptive statistics for specific variables or groups.

- Sorting and filtering data to focus on relevant subsets.

- Performing comparisons between groups or categories using Pandas and statistical measures.

- Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.

- Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.

- Creating tables to summarize key statistics or values.

- Generating bar graphs to compare categories or groups.

- Producing line graphs to show trends over time or continuous variables.

- Creating scatter plots to visualize relationships between numerical variables.

- Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.

- Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.

- Explaining the data analysis steps they are performing.

- Interpreting the results of their analysis.

- Describing the patterns or trends revealed by their visualizations.

- Relating their findings back to their project question.

- Teacher circulates to provide individualized guidance, answer questions, troubleshoot coding issues, and provide feedback on data analysis and visualization approaches. Encourage peer-to-peer support and collaboration within groups.

- Reflection & Discussion (10 mins): Class discussion:  
  
What data analysis and visualization techniques did you use in your project work session today?  
What were some of the key findings or interesting patterns you discovered in your data analysis so far?  
What were some challenges you encountered during data analysis or visualization? How did you overcome them or plan to address them?  
What are your next steps for your project? What do you plan to focus on in the remaining project time?

- What data analysis and visualization techniques did you use in your project work session today?

- What were some of the key findings or interesting patterns you discovered in your data analysis so far?

- What were some challenges you encountered during data analysis or visualization? How did you overcome them or plan to address them?

- What are your next steps for your project? What do you plan to focus on in the remaining project time?

- Assessments:  
  
Formative Check (5 mins): Briefly check each group's Observable notebook to assess their progress in data analysis and visualization. Review the code they have written and the visualizations they have created. Provide quick feedback and suggestions.  
Observation of group work during the session – assess their collaboration, problem-solving, and engagement with data analysis and visualization tasks.

- Formative Check (5 mins): Briefly check each group's Observable notebook to assess their progress in data analysis and visualization. Review the code they have written and the visualizations they have created. Provide quick feedback and suggestions.

- Observation of group work during the session – assess their collaboration, problem-solving, and engagement with data analysis and visualization tasks.

- Student Workbook:  
  
Checklist of data analysis and visualization tasks to complete for the project.  
Prompts for documenting data analysis steps, findings, and visualization interpretations in Observable notebooks.  
Guidance on choosing appropriate visualization types for different data and messages.  
Project progress tracking section to record completed tasks, challenges, and next steps.

- Checklist of data analysis and visualization tasks to complete for the project.

- Prompts for documenting data analysis steps, findings, and visualization interpretations in Observable notebooks.

- Guidance on choosing appropriate visualization types for different data and messages.

- Project progress tracking section to record completed tasks, challenges, and next steps.

- Extension Activities:  
  
Students explore more advanced data analysis techniques that might be relevant to their project (e.g., correlation analysis in more detail, basic statistical tests – conceptually introduced if appropriate).  
Challenge: Experiment with adding interactive elements to their visualizations in Observable (e.g., tooltips, sliders, dropdown menus) to create more dynamic and informative dashboards or interactive visualizations.

- Students explore more advanced data analysis techniques that might be relevant to their project (e.g., correlation analysis in more detail, basic statistical tests – conceptually introduced if appropriate).

- Challenge: Experiment with adding interactive elements to their visualizations in Observable (e.g., tooltips, sliders, dropdown menus) to create more dynamic and informative dashboards or interactive visualizations.

- Students will be able to perform relevant data analysis techniques on their project datasets using Python and Pandas in Observable notebooks, guided by their project plans.

- Students will be able to create effective data visualizations (tables, graphs) in Observable to communicate their project findings, choosing appropriate visualization types for their data and message.

- Students will be able to collaboratively work within their groups to analyse data, create visualizations, and document their project progress in Observable notebooks.

- NSW Curriculum Outcomes: SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), ACSIS126 (Processing and analysing data), ACSIS127 (Communicating findings).

- Warm-Up (10 mins): "Data Analysis and Visualization Techniques - Quick Recap" - Briefly review the data analysis and visualization techniques learned in the course that are most relevant to the projects:  
  
Descriptive Statistics (mean, median, range).  
Data Sorting and Filtering.  
Bar Graphs (for categorical comparisons).  
Line Graphs (for trends over time or continuous variables).  
Scatter Plots (for relationships between numerical variables).  
Tables (for summarizing data and presenting key values).  
Quickly remind students of when each technique is most appropriate and how to implement them in Python/Observable.

- Descriptive Statistics (mean, median, range).

- Data Sorting and Filtering.

- Bar Graphs (for categorical comparisons).

- Line Graphs (for trends over time or continuous variables).

- Scatter Plots (for relationships between numerical variables).

- Tables (for summarizing data and presenting key values).

- Quickly remind students of when each technique is most appropriate and how to implement them in Python/Observable.

- Main Activity 1: "Group Data Analysis and Visualization Work Session" (60 mins):  
  
In-class group work session. Each group works on their Observable notebooks to implement their data analysis and visualization plans.  
Guide students to focus on:  
  
Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.  
Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.  
  
Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.  
  
Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.  
  
  
Teacher circulates to provide individualized guidance, answer questions, troubleshoot coding issues, and provide feedback on data analysis and visualization approaches. Encourage peer-to-peer support and collaboration within groups.

- In-class group work session. Each group works on their Observable notebooks to implement their data analysis and visualization plans.

- Guide students to focus on:  
  
Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.  
Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.  
  
Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.  
  
Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.

- Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.

- Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.

- Calculating descriptive statistics for specific variables or groups.

- Sorting and filtering data to focus on relevant subsets.

- Performing comparisons between groups or categories using Pandas and statistical measures.

- Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.

- Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.

- Creating tables to summarize key statistics or values.

- Generating bar graphs to compare categories or groups.

- Producing line graphs to show trends over time or continuous variables.

- Creating scatter plots to visualize relationships between numerical variables.

- Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.

- Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.

- Explaining the data analysis steps they are performing.

- Interpreting the results of their analysis.

- Describing the patterns or trends revealed by their visualizations.

- Relating their findings back to their project question.

- Teacher circulates to provide individualized guidance, answer questions, troubleshoot coding issues, and provide feedback on data analysis and visualization approaches. Encourage peer-to-peer support and collaboration within groups.

- Reflection & Discussion (10 mins): Class discussion:  
  
What data analysis and visualization techniques did you use in your project work session today?  
What were some of the key findings or interesting patterns you discovered in your data analysis so far?  
What were some challenges you encountered during data analysis or visualization? How did you overcome them or plan to address them?  
What are your next steps for your project? What do you plan to focus on in the remaining project time?

- What data analysis and visualization techniques did you use in your project work session today?

- What were some of the key findings or interesting patterns you discovered in your data analysis so far?

- What were some challenges you encountered during data analysis or visualization? How did you overcome them or plan to address them?

- What are your next steps for your project? What do you plan to focus on in the remaining project time?

- Assessments:  
  
Formative Check (5 mins): Briefly check each group's Observable notebook to assess their progress in data analysis and visualization. Review the code they have written and the visualizations they have created. Provide quick feedback and suggestions.  
Observation of group work during the session – assess their collaboration, problem-solving, and engagement with data analysis and visualization tasks.

- Formative Check (5 mins): Briefly check each group's Observable notebook to assess their progress in data analysis and visualization. Review the code they have written and the visualizations they have created. Provide quick feedback and suggestions.

- Observation of group work during the session – assess their collaboration, problem-solving, and engagement with data analysis and visualization tasks.

- Student Workbook:  
  
Checklist of data analysis and visualization tasks to complete for the project.  
Prompts for documenting data analysis steps, findings, and visualization interpretations in Observable notebooks.  
Guidance on choosing appropriate visualization types for different data and messages.  
Project progress tracking section to record completed tasks, challenges, and next steps.

- Checklist of data analysis and visualization tasks to complete for the project.

- Prompts for documenting data analysis steps, findings, and visualization interpretations in Observable notebooks.

- Guidance on choosing appropriate visualization types for different data and messages.

- Project progress tracking section to record completed tasks, challenges, and next steps.

- Extension Activities:  
  
Students explore more advanced data analysis techniques that might be relevant to their project (e.g., correlation analysis in more detail, basic statistical tests – conceptually introduced if appropriate).  
Challenge: Experiment with adding interactive elements to their visualizations in Observable (e.g., tooltips, sliders, dropdown menus) to create more dynamic and informative dashboards or interactive visualizations.

- Students explore more advanced data analysis techniques that might be relevant to their project (e.g., correlation analysis in more detail, basic statistical tests – conceptually introduced if appropriate).

- Challenge: Experiment with adding interactive elements to their visualizations in Observable (e.g., tooltips, sliders, dropdown menus) to create more dynamic and informative dashboards or interactive visualizations.

- Descriptive Statistics (mean, median, range).

- Data Sorting and Filtering.

- Bar Graphs (for categorical comparisons).

- Line Graphs (for trends over time or continuous variables).

- Scatter Plots (for relationships between numerical variables).

- Tables (for summarizing data and presenting key values).

- Quickly remind students of when each technique is most appropriate and how to implement them in Python/Observable.

- In-class group work session. Each group works on their Observable notebooks to implement their data analysis and visualization plans.

- Guide students to focus on:  
  
Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.  
Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.  
  
Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.  
  
Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.

- Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.

- Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.

- Calculating descriptive statistics for specific variables or groups.

- Sorting and filtering data to focus on relevant subsets.

- Performing comparisons between groups or categories using Pandas and statistical measures.

- Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.

- Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.

- Creating tables to summarize key statistics or values.

- Generating bar graphs to compare categories or groups.

- Producing line graphs to show trends over time or continuous variables.

- Creating scatter plots to visualize relationships between numerical variables.

- Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.

- Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.

- Explaining the data analysis steps they are performing.

- Interpreting the results of their analysis.

- Describing the patterns or trends revealed by their visualizations.

- Relating their findings back to their project question.

- Teacher circulates to provide individualized guidance, answer questions, troubleshoot coding issues, and provide feedback on data analysis and visualization approaches. Encourage peer-to-peer support and collaboration within groups.

- Data Cleaning (if needed): Based on their data exploration, perform any necessary data cleaning steps (handling missing values, standardizing formats) using Pandas techniques learned in Week 6.

- Data Analysis Implementation: Implement their planned data analysis techniques in Python code within Observable. This might include:  
  
Calculating descriptive statistics for specific variables or groups.  
Sorting and filtering data to focus on relevant subsets.  
Performing comparisons between groups or categories using Pandas and statistical measures.  
Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.

- Calculating descriptive statistics for specific variables or groups.

- Sorting and filtering data to focus on relevant subsets.

- Performing comparisons between groups or categories using Pandas and statistical measures.

- Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.

- Visualization Creation: Create visualizations in Observable to communicate their data analysis findings effectively. This might include:  
  
Creating tables to summarize key statistics or values.  
Generating bar graphs to compare categories or groups.  
Producing line graphs to show trends over time or continuous variables.  
Creating scatter plots to visualize relationships between numerical variables.  
Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.

- Creating tables to summarize key statistics or values.

- Generating bar graphs to compare categories or groups.

- Producing line graphs to show trends over time or continuous variables.

- Creating scatter plots to visualize relationships between numerical variables.

- Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.

- Documentation: Throughout their work session, groups should continuously document their progress and findings in markdown cells in their Observable notebooks. This includes:  
  
Explaining the data analysis steps they are performing.  
Interpreting the results of their analysis.  
Describing the patterns or trends revealed by their visualizations.  
Relating their findings back to their project question.

- Explaining the data analysis steps they are performing.

- Interpreting the results of their analysis.

- Describing the patterns or trends revealed by their visualizations.

- Relating their findings back to their project question.

- Calculating descriptive statistics for specific variables or groups.

- Sorting and filtering data to focus on relevant subsets.

- Performing comparisons between groups or categories using Pandas and statistical measures.

- Investigating relationships between variables using correlation analysis (if appropriate and conceptually accessible) or by examining scatter plots.

- Creating tables to summarize key statistics or values.

- Generating bar graphs to compare categories or groups.

- Producing line graphs to show trends over time or continuous variables.

- Creating scatter plots to visualize relationships between numerical variables.

- Ensure visualizations are clear, well-labeled, and effectively communicate the intended message.

- Explaining the data analysis steps they are performing.

- Interpreting the results of their analysis.

- Describing the patterns or trends revealed by their visualizations.

- Relating their findings back to their project question.

- What data analysis and visualization techniques did you use in your project work session today?

- What were some of the key findings or interesting patterns you discovered in your data analysis so far?

- What were some challenges you encountered during data analysis or visualization? How did you overcome them or plan to address them?

- What are your next steps for your project? What do you plan to focus on in the remaining project time?

- Formative Check (5 mins): Briefly check each group's Observable notebook to assess their progress in data analysis and visualization. Review the code they have written and the visualizations they have created. Provide quick feedback and suggestions.

- Observation of group work during the session – assess their collaboration, problem-solving, and engagement with data analysis and visualization tasks.

- Checklist of data analysis and visualization tasks to complete for the project.

- Prompts for documenting data analysis steps, findings, and visualization interpretations in Observable notebooks.

- Guidance on choosing appropriate visualization types for different data and messages.

- Project progress tracking section to record completed tasks, challenges, and next steps.

- Students explore more advanced data analysis techniques that might be relevant to their project (e.g., correlation analysis in more detail, basic statistical tests – conceptually introduced if appropriate).

- Challenge: Experiment with adding interactive elements to their visualizations in Observable (e.g., tooltips, sliders, dropdown menus) to create more dynamic and informative dashboards or interactive visualizations.

Okay, here are the detailed lesson plans for Weeks 9 and 10, completing the comprehensive 10-week Year 8 Data Science curriculum.

### Week 9: Project Refinement & Advanced Review

### Lesson 9.1: Project Presentations - Peer Feedback and Iteration

- Title: Showcase and Sharpen: Project Presentations and Peer Review

- Learning Outcomes:  
  
Students will be able to present their group data science project findings clearly and concisely to their peers.  
Students will be able to provide constructive and specific feedback to their peers on their project presentations, focusing on data analysis, visualizations, and communication effectiveness.  
Students will be able to reflect on peer feedback and identify areas for improvement in their own project.  
Students will be able to plan iterative refinements to their project based on peer feedback to enhance their final presentations and project outcomes.  
NSW Curriculum Outcomes: SC4-WS-07 (Communicate information), ACSIS127 (Communicating findings), SC4-WS-08 (Reflecting on scientific methods), ACSIS131 (Evaluating methods and conclusions).

- Students will be able to present their group data science project findings clearly and concisely to their peers.

- Students will be able to provide constructive and specific feedback to their peers on their project presentations, focusing on data analysis, visualizations, and communication effectiveness.

- Students will be able to reflect on peer feedback and identify areas for improvement in their own project.

- Students will be able to plan iterative refinements to their project based on peer feedback to enhance their final presentations and project outcomes.

- NSW Curriculum Outcomes: SC4-WS-07 (Communicate information), ACSIS127 (Communicating findings), SC4-WS-08 (Reflecting on scientific methods), ACSIS131 (Evaluating methods and conclusions).

- Overview: This lesson focuses on project presentations and peer review, a critical part of the scientific process and collaborative learning. Students will present their projects to the class, receive feedback from peers, and learn to use this feedback to improve their projects iteratively.

- Activities:  
  
Warm-Up (10 mins): "Effective Presentation Skills Brainstorm" - Class brainstorm: "What makes a presentation effective and engaging? What are some key elements of a good scientific presentation?" (Examples: clear visuals, concise language, logical flow, engaging delivery, clear explanation of findings, evidence-based conclusions). List brainstormed elements on the board. Emphasize both content and delivery.  
Main Activity 1: "Group Project Presentations (Part 1)" (35 mins):  
  
Group project presentations begin. Divide the class time to allow each group a set time for presentation (e.g., 5-7 minutes presentation + 3-5 minutes for Q&A and feedback per group, adjust based on class size and number of groups).  
Groups present their projects to the class. Presentations should ideally cover:  
  
Project Question: Clearly state the real-world problem and project question they investigated.  
Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.  
Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).  
Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.  
Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.  
(Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.  
  
While groups are presenting, the rest of the class acts as peer reviewers. Provide students with a "Peer Feedback Form" (see Student Workbook).  
For each presentation, after the group finishes presenting, facilitate a brief Q&A session (1-2 questions from the teacher or class) followed by structured peer feedback:  
  
Each student (and teacher) completes the Peer Feedback Form for the presenting group.  
Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.  
  
Continue presentations for the allocated time, aiming to have as many groups present as possible in this lesson.  
  
Main Activity 2: "Peer Feedback Reflection and Iteration Planning" (20 mins):  
  
After the first set of presentations (or after all presentations if time allows in one lesson), guide students to reflect on the peer feedback they received (and the feedback they gave to others).  
Individual reflection time: Each student reviews the Peer Feedback Forms they received for their group's presentation.  
Group discussion: Within their project groups, students discuss the peer feedback they received. Focus on:  
  
Identifying common themes or consistent feedback points.  
Prioritizing feedback points that are most important to address for project improvement.  
Brainstorming specific, actionable steps they can take to refine their project based on the feedback.  
  
Each group task: Develop an "Iteration Plan" outlining specific actions they will take to revise their project based on peer feedback, focusing on improving data analysis, visualizations, and presentation clarity. Assign tasks within the team for project refinement.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What did you learn from presenting your project to your peers and receiving feedback?  
What kind of feedback did you find most helpful or valuable? Why?  
How does peer review help in improving scientific work and communication?  
What are your group's plans for refining your project based on the feedback you received? What specific changes will you make?  
  
Assessments:  
  
Formative Assessment: Quality of group project presentations (assess clarity, content coverage, use of visualizations, communication skills based on presentation rubric - if used).  
Formative Assessment: Quality and specificity of peer feedback provided by students (assess completeness and constructiveness of feedback based on Peer Feedback Forms).  
Review of group "Iteration Plans" – assess the extent to which they have incorporated peer feedback and planned actionable revisions.  
  
Student Workbook:  
  
Project presentation guidelines and rubric (if used).  
Peer Feedback Form template (with specific feedback criteria related to data analysis, visualizations, communication clarity, etc.).  
"Peer Feedback Reflection" worksheet with prompts to analyze received feedback and identify areas for improvement.  
"Project Iteration Plan" template to outline specific revisions based on feedback and task assignments.  
  
Extension Activities:  
  
Students research examples of scientific peer review processes in professional science and discuss the importance of peer review in ensuring quality and validity in scientific research.  
Challenge: Groups provide "expert" feedback on another group's Observable notebook code and data analysis workflow, focusing on code clarity, efficiency, and correctness of analysis techniques (more advanced peer review).

- Warm-Up (10 mins): "Effective Presentation Skills Brainstorm" - Class brainstorm: "What makes a presentation effective and engaging? What are some key elements of a good scientific presentation?" (Examples: clear visuals, concise language, logical flow, engaging delivery, clear explanation of findings, evidence-based conclusions). List brainstormed elements on the board. Emphasize both content and delivery.

- Main Activity 1: "Group Project Presentations (Part 1)" (35 mins):  
  
Group project presentations begin. Divide the class time to allow each group a set time for presentation (e.g., 5-7 minutes presentation + 3-5 minutes for Q&A and feedback per group, adjust based on class size and number of groups).  
Groups present their projects to the class. Presentations should ideally cover:  
  
Project Question: Clearly state the real-world problem and project question they investigated.  
Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.  
Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).  
Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.  
Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.  
(Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.  
  
While groups are presenting, the rest of the class acts as peer reviewers. Provide students with a "Peer Feedback Form" (see Student Workbook).  
For each presentation, after the group finishes presenting, facilitate a brief Q&A session (1-2 questions from the teacher or class) followed by structured peer feedback:  
  
Each student (and teacher) completes the Peer Feedback Form for the presenting group.  
Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.  
  
Continue presentations for the allocated time, aiming to have as many groups present as possible in this lesson.

- Group project presentations begin. Divide the class time to allow each group a set time for presentation (e.g., 5-7 minutes presentation + 3-5 minutes for Q&A and feedback per group, adjust based on class size and number of groups).

- Groups present their projects to the class. Presentations should ideally cover:  
  
Project Question: Clearly state the real-world problem and project question they investigated.  
Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.  
Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).  
Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.  
Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.  
(Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.

- Project Question: Clearly state the real-world problem and project question they investigated.

- Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.

- Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).

- Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.

- Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.

- (Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.

- While groups are presenting, the rest of the class acts as peer reviewers. Provide students with a "Peer Feedback Form" (see Student Workbook).

- For each presentation, after the group finishes presenting, facilitate a brief Q&A session (1-2 questions from the teacher or class) followed by structured peer feedback:  
  
Each student (and teacher) completes the Peer Feedback Form for the presenting group.  
Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.

- Each student (and teacher) completes the Peer Feedback Form for the presenting group.

- Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.

- Continue presentations for the allocated time, aiming to have as many groups present as possible in this lesson.

- Main Activity 2: "Peer Feedback Reflection and Iteration Planning" (20 mins):  
  
After the first set of presentations (or after all presentations if time allows in one lesson), guide students to reflect on the peer feedback they received (and the feedback they gave to others).  
Individual reflection time: Each student reviews the Peer Feedback Forms they received for their group's presentation.  
Group discussion: Within their project groups, students discuss the peer feedback they received. Focus on:  
  
Identifying common themes or consistent feedback points.  
Prioritizing feedback points that are most important to address for project improvement.  
Brainstorming specific, actionable steps they can take to refine their project based on the feedback.  
  
Each group task: Develop an "Iteration Plan" outlining specific actions they will take to revise their project based on peer feedback, focusing on improving data analysis, visualizations, and presentation clarity. Assign tasks within the team for project refinement.

- After the first set of presentations (or after all presentations if time allows in one lesson), guide students to reflect on the peer feedback they received (and the feedback they gave to others).

- Individual reflection time: Each student reviews the Peer Feedback Forms they received for their group's presentation.

- Group discussion: Within their project groups, students discuss the peer feedback they received. Focus on:  
  
Identifying common themes or consistent feedback points.  
Prioritizing feedback points that are most important to address for project improvement.  
Brainstorming specific, actionable steps they can take to refine their project based on the feedback.

- Identifying common themes or consistent feedback points.

- Prioritizing feedback points that are most important to address for project improvement.

- Brainstorming specific, actionable steps they can take to refine their project based on the feedback.

- Each group task: Develop an "Iteration Plan" outlining specific actions they will take to revise their project based on peer feedback, focusing on improving data analysis, visualizations, and presentation clarity. Assign tasks within the team for project refinement.

- Reflection & Discussion (10 mins): Class discussion:  
  
What did you learn from presenting your project to your peers and receiving feedback?  
What kind of feedback did you find most helpful or valuable? Why?  
How does peer review help in improving scientific work and communication?  
What are your group's plans for refining your project based on the feedback you received? What specific changes will you make?

- What did you learn from presenting your project to your peers and receiving feedback?

- What kind of feedback did you find most helpful or valuable? Why?

- How does peer review help in improving scientific work and communication?

- What are your group's plans for refining your project based on the feedback you received? What specific changes will you make?

- Assessments:  
  
Formative Assessment: Quality of group project presentations (assess clarity, content coverage, use of visualizations, communication skills based on presentation rubric - if used).  
Formative Assessment: Quality and specificity of peer feedback provided by students (assess completeness and constructiveness of feedback based on Peer Feedback Forms).  
Review of group "Iteration Plans" – assess the extent to which they have incorporated peer feedback and planned actionable revisions.

- Formative Assessment: Quality of group project presentations (assess clarity, content coverage, use of visualizations, communication skills based on presentation rubric - if used).

- Formative Assessment: Quality and specificity of peer feedback provided by students (assess completeness and constructiveness of feedback based on Peer Feedback Forms).

- Review of group "Iteration Plans" – assess the extent to which they have incorporated peer feedback and planned actionable revisions.

- Student Workbook:  
  
Project presentation guidelines and rubric (if used).  
Peer Feedback Form template (with specific feedback criteria related to data analysis, visualizations, communication clarity, etc.).  
"Peer Feedback Reflection" worksheet with prompts to analyze received feedback and identify areas for improvement.  
"Project Iteration Plan" template to outline specific revisions based on feedback and task assignments.

- Project presentation guidelines and rubric (if used).

- Peer Feedback Form template (with specific feedback criteria related to data analysis, visualizations, communication clarity, etc.).

- "Peer Feedback Reflection" worksheet with prompts to analyze received feedback and identify areas for improvement.

- "Project Iteration Plan" template to outline specific revisions based on feedback and task assignments.

- Extension Activities:  
  
Students research examples of scientific peer review processes in professional science and discuss the importance of peer review in ensuring quality and validity in scientific research.  
Challenge: Groups provide "expert" feedback on another group's Observable notebook code and data analysis workflow, focusing on code clarity, efficiency, and correctness of analysis techniques (more advanced peer review).

- Students research examples of scientific peer review processes in professional science and discuss the importance of peer review in ensuring quality and validity in scientific research.

- Challenge: Groups provide "expert" feedback on another group's Observable notebook code and data analysis workflow, focusing on code clarity, efficiency, and correctness of analysis techniques (more advanced peer review).

- Lesson Materials:  
  
Slide Deck: Introduction to project presentations and peer review, guidelines for effective presentations, peer feedback process, iteration planning.  
Handout: Project presentation guidelines/rubric (if used), Peer Feedback Form template, "Peer Feedback Reflection" worksheet, "Project Iteration Plan" template.  
Computers/Tablets with internet access for presenting groups (if using digital presentations).

- Slide Deck: Introduction to project presentations and peer review, guidelines for effective presentations, peer feedback process, iteration planning.

- Handout: Project presentation guidelines/rubric (if used), Peer Feedback Form template, "Peer Feedback Reflection" worksheet, "Project Iteration Plan" template.

- Computers/Tablets with internet access for presenting groups (if using digital presentations).

- Students will be able to present their group data science project findings clearly and concisely to their peers.

- Students will be able to provide constructive and specific feedback to their peers on their project presentations, focusing on data analysis, visualizations, and communication effectiveness.

- Students will be able to reflect on peer feedback and identify areas for improvement in their own project.

- Students will be able to plan iterative refinements to their project based on peer feedback to enhance their final presentations and project outcomes.

- NSW Curriculum Outcomes: SC4-WS-07 (Communicate information), ACSIS127 (Communicating findings), SC4-WS-08 (Reflecting on scientific methods), ACSIS131 (Evaluating methods and conclusions).

- Warm-Up (10 mins): "Effective Presentation Skills Brainstorm" - Class brainstorm: "What makes a presentation effective and engaging? What are some key elements of a good scientific presentation?" (Examples: clear visuals, concise language, logical flow, engaging delivery, clear explanation of findings, evidence-based conclusions). List brainstormed elements on the board. Emphasize both content and delivery.

- Main Activity 1: "Group Project Presentations (Part 1)" (35 mins):  
  
Group project presentations begin. Divide the class time to allow each group a set time for presentation (e.g., 5-7 minutes presentation + 3-5 minutes for Q&A and feedback per group, adjust based on class size and number of groups).  
Groups present their projects to the class. Presentations should ideally cover:  
  
Project Question: Clearly state the real-world problem and project question they investigated.  
Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.  
Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).  
Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.  
Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.  
(Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.  
  
While groups are presenting, the rest of the class acts as peer reviewers. Provide students with a "Peer Feedback Form" (see Student Workbook).  
For each presentation, after the group finishes presenting, facilitate a brief Q&A session (1-2 questions from the teacher or class) followed by structured peer feedback:  
  
Each student (and teacher) completes the Peer Feedback Form for the presenting group.  
Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.  
  
Continue presentations for the allocated time, aiming to have as many groups present as possible in this lesson.

- Group project presentations begin. Divide the class time to allow each group a set time for presentation (e.g., 5-7 minutes presentation + 3-5 minutes for Q&A and feedback per group, adjust based on class size and number of groups).

- Groups present their projects to the class. Presentations should ideally cover:  
  
Project Question: Clearly state the real-world problem and project question they investigated.  
Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.  
Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).  
Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.  
Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.  
(Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.

- Project Question: Clearly state the real-world problem and project question they investigated.

- Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.

- Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).

- Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.

- Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.

- (Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.

- While groups are presenting, the rest of the class acts as peer reviewers. Provide students with a "Peer Feedback Form" (see Student Workbook).

- For each presentation, after the group finishes presenting, facilitate a brief Q&A session (1-2 questions from the teacher or class) followed by structured peer feedback:  
  
Each student (and teacher) completes the Peer Feedback Form for the presenting group.  
Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.

- Each student (and teacher) completes the Peer Feedback Form for the presenting group.

- Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.

- Continue presentations for the allocated time, aiming to have as many groups present as possible in this lesson.

- Main Activity 2: "Peer Feedback Reflection and Iteration Planning" (20 mins):  
  
After the first set of presentations (or after all presentations if time allows in one lesson), guide students to reflect on the peer feedback they received (and the feedback they gave to others).  
Individual reflection time: Each student reviews the Peer Feedback Forms they received for their group's presentation.  
Group discussion: Within their project groups, students discuss the peer feedback they received. Focus on:  
  
Identifying common themes or consistent feedback points.  
Prioritizing feedback points that are most important to address for project improvement.  
Brainstorming specific, actionable steps they can take to refine their project based on the feedback.  
  
Each group task: Develop an "Iteration Plan" outlining specific actions they will take to revise their project based on peer feedback, focusing on improving data analysis, visualizations, and presentation clarity. Assign tasks within the team for project refinement.

- After the first set of presentations (or after all presentations if time allows in one lesson), guide students to reflect on the peer feedback they received (and the feedback they gave to others).

- Individual reflection time: Each student reviews the Peer Feedback Forms they received for their group's presentation.

- Group discussion: Within their project groups, students discuss the peer feedback they received. Focus on:  
  
Identifying common themes or consistent feedback points.  
Prioritizing feedback points that are most important to address for project improvement.  
Brainstorming specific, actionable steps they can take to refine their project based on the feedback.

- Identifying common themes or consistent feedback points.

- Prioritizing feedback points that are most important to address for project improvement.

- Brainstorming specific, actionable steps they can take to refine their project based on the feedback.

- Each group task: Develop an "Iteration Plan" outlining specific actions they will take to revise their project based on peer feedback, focusing on improving data analysis, visualizations, and presentation clarity. Assign tasks within the team for project refinement.

- Reflection & Discussion (10 mins): Class discussion:  
  
What did you learn from presenting your project to your peers and receiving feedback?  
What kind of feedback did you find most helpful or valuable? Why?  
How does peer review help in improving scientific work and communication?  
What are your group's plans for refining your project based on the feedback you received? What specific changes will you make?

- What did you learn from presenting your project to your peers and receiving feedback?

- What kind of feedback did you find most helpful or valuable? Why?

- How does peer review help in improving scientific work and communication?

- What are your group's plans for refining your project based on the feedback you received? What specific changes will you make?

- Assessments:  
  
Formative Assessment: Quality of group project presentations (assess clarity, content coverage, use of visualizations, communication skills based on presentation rubric - if used).  
Formative Assessment: Quality and specificity of peer feedback provided by students (assess completeness and constructiveness of feedback based on Peer Feedback Forms).  
Review of group "Iteration Plans" – assess the extent to which they have incorporated peer feedback and planned actionable revisions.

- Formative Assessment: Quality of group project presentations (assess clarity, content coverage, use of visualizations, communication skills based on presentation rubric - if used).

- Formative Assessment: Quality and specificity of peer feedback provided by students (assess completeness and constructiveness of feedback based on Peer Feedback Forms).

- Review of group "Iteration Plans" – assess the extent to which they have incorporated peer feedback and planned actionable revisions.

- Student Workbook:  
  
Project presentation guidelines and rubric (if used).  
Peer Feedback Form template (with specific feedback criteria related to data analysis, visualizations, communication clarity, etc.).  
"Peer Feedback Reflection" worksheet with prompts to analyze received feedback and identify areas for improvement.  
"Project Iteration Plan" template to outline specific revisions based on feedback and task assignments.

- Project presentation guidelines and rubric (if used).

- Peer Feedback Form template (with specific feedback criteria related to data analysis, visualizations, communication clarity, etc.).

- "Peer Feedback Reflection" worksheet with prompts to analyze received feedback and identify areas for improvement.

- "Project Iteration Plan" template to outline specific revisions based on feedback and task assignments.

- Extension Activities:  
  
Students research examples of scientific peer review processes in professional science and discuss the importance of peer review in ensuring quality and validity in scientific research.  
Challenge: Groups provide "expert" feedback on another group's Observable notebook code and data analysis workflow, focusing on code clarity, efficiency, and correctness of analysis techniques (more advanced peer review).

- Students research examples of scientific peer review processes in professional science and discuss the importance of peer review in ensuring quality and validity in scientific research.

- Challenge: Groups provide "expert" feedback on another group's Observable notebook code and data analysis workflow, focusing on code clarity, efficiency, and correctness of analysis techniques (more advanced peer review).

- Group project presentations begin. Divide the class time to allow each group a set time for presentation (e.g., 5-7 minutes presentation + 3-5 minutes for Q&A and feedback per group, adjust based on class size and number of groups).

- Groups present their projects to the class. Presentations should ideally cover:  
  
Project Question: Clearly state the real-world problem and project question they investigated.  
Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.  
Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).  
Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.  
Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.  
(Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.

- Project Question: Clearly state the real-world problem and project question they investigated.

- Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.

- Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).

- Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.

- Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.

- (Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.

- While groups are presenting, the rest of the class acts as peer reviewers. Provide students with a "Peer Feedback Form" (see Student Workbook).

- For each presentation, after the group finishes presenting, facilitate a brief Q&A session (1-2 questions from the teacher or class) followed by structured peer feedback:  
  
Each student (and teacher) completes the Peer Feedback Form for the presenting group.  
Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.

- Each student (and teacher) completes the Peer Feedback Form for the presenting group.

- Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.

- Continue presentations for the allocated time, aiming to have as many groups present as possible in this lesson.

- Project Question: Clearly state the real-world problem and project question they investigated.

- Dataset and Data Exploration: Briefly describe the dataset they used and key findings from initial data exploration.

- Data Analysis Methods: Explain the data analysis techniques they used (e.g., descriptive statistics, sorting, filtering, visualizations).

- Key Findings and Visualizations: Present their key findings, using data visualizations (tables, graphs) to support their conclusions.

- Conclusions and Insights: Summarize their main conclusions and insights related to their project question, based on their data analysis.

- (Optional - if applicable) Limitations or challenges encountered during the project, potential future directions.

- Each student (and teacher) completes the Peer Feedback Form for the presenting group.

- Share 1-2 specific pieces of constructive feedback with the presenting group based on the forms. Focus on actionable feedback for improvement.

- After the first set of presentations (or after all presentations if time allows in one lesson), guide students to reflect on the peer feedback they received (and the feedback they gave to others).

- Individual reflection time: Each student reviews the Peer Feedback Forms they received for their group's presentation.

- Group discussion: Within their project groups, students discuss the peer feedback they received. Focus on:  
  
Identifying common themes or consistent feedback points.  
Prioritizing feedback points that are most important to address for project improvement.  
Brainstorming specific, actionable steps they can take to refine their project based on the feedback.

- Identifying common themes or consistent feedback points.

- Prioritizing feedback points that are most important to address for project improvement.

- Brainstorming specific, actionable steps they can take to refine their project based on the feedback.

- Each group task: Develop an "Iteration Plan" outlining specific actions they will take to revise their project based on peer feedback, focusing on improving data analysis, visualizations, and presentation clarity. Assign tasks within the team for project refinement.

- Identifying common themes or consistent feedback points.

- Prioritizing feedback points that are most important to address for project improvement.

- Brainstorming specific, actionable steps they can take to refine their project based on the feedback.

- What did you learn from presenting your project to your peers and receiving feedback?

- What kind of feedback did you find most helpful or valuable? Why?

- How does peer review help in improving scientific work and communication?

- What are your group's plans for refining your project based on the feedback you received? What specific changes will you make?

- Formative Assessment: Quality of group project presentations (assess clarity, content coverage, use of visualizations, communication skills based on presentation rubric - if used).

- Formative Assessment: Quality and specificity of peer feedback provided by students (assess completeness and constructiveness of feedback based on Peer Feedback Forms).

- Review of group "Iteration Plans" – assess the extent to which they have incorporated peer feedback and planned actionable revisions.

- Project presentation guidelines and rubric (if used).

- Peer Feedback Form template (with specific feedback criteria related to data analysis, visualizations, communication clarity, etc.).

- "Peer Feedback Reflection" worksheet with prompts to analyze received feedback and identify areas for improvement.

- "Project Iteration Plan" template to outline specific revisions based on feedback and task assignments.

- Students research examples of scientific peer review processes in professional science and discuss the importance of peer review in ensuring quality and validity in scientific research.

- Challenge: Groups provide "expert" feedback on another group's Observable notebook code and data analysis workflow, focusing on code clarity, efficiency, and correctness of analysis techniques (more advanced peer review).

- Slide Deck: Introduction to project presentations and peer review, guidelines for effective presentations, peer feedback process, iteration planning.

- Handout: Project presentation guidelines/rubric (if used), Peer Feedback Form template, "Peer Feedback Reflection" worksheet, "Project Iteration Plan" template.

- Computers/Tablets with internet access for presenting groups (if using digital presentations).

### Lesson 9.2: AI-Driven Project Review and Refinement

- Title: AI Project Consultant: Leveraging AI for Project Enhancement

- Learning Outcomes:  
  
Students will be able to utilize AI-powered review tools (if available) to receive automated feedback on their project Observable notebooks, focusing on code quality, data analysis, and visualizations.  
Students will be able to analyse and interpret AI-generated feedback and identify specific areas for project improvement.  
Students will be able to integrate AI feedback with peer feedback to develop a comprehensive revision plan for their project.  
Students will be able to work collaboratively to implement project refinements based on both AI and peer feedback, enhancing their project notebooks and presentations.  
NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-07 (Communicate information), SC4-WS-08 (Reflecting on scientific methods), ACSIS131 (Evaluating methods and conclusions).

- Students will be able to utilize AI-powered review tools (if available) to receive automated feedback on their project Observable notebooks, focusing on code quality, data analysis, and visualizations.

- Students will be able to analyse and interpret AI-generated feedback and identify specific areas for project improvement.

- Students will be able to integrate AI feedback with peer feedback to develop a comprehensive revision plan for their project.

- Students will be able to work collaboratively to implement project refinements based on both AI and peer feedback, enhancing their project notebooks and presentations.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-07 (Communicate information), SC4-WS-08 (Reflecting on scientific methods), ACSIS131 (Evaluating methods and conclusions).

- Overview: This lesson introduces an innovative element – AI-driven project review. If feasible, students will use AI tools to get automated feedback on their project notebooks, complementing peer feedback. Students will learn to interpret AI feedback and integrate it into their project refinement process, further enhancing their work.

- Activities:  
  
Warm-Up (10 mins): "AI Feedback - Expectations and Limitations" - Class discussion: "What do you think AI could be good at providing feedback on for your data science projects? What might AI not be good at or what limitations might it have in giving feedback? What kind of feedback is best suited for humans vs. AI?" Discuss the strengths and weaknesses of automated feedback vs. human feedback. Emphasize that AI feedback is a tool to assist, not replace, human judgment.  
Main Activity 1: "AI-Driven Project Review Session" (40 mins):  
  
Introduce the AI-powered project review tool (if available - specific tool and access method will depend on chosen AI integration).  
Guide students on how to submit their Observable project notebooks to the AI review tool (e.g., uploading notebook file, sharing notebook link with the AI tool).  
Students submit their group project notebooks to the AI review tool.  
Explain the types of feedback the AI tool is expected to provide (depending on tool capabilities, examples):  
  
Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).  
Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).  
Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).  
(Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).  
  
Students receive and review the AI-generated feedback on their project notebooks. Encourage them to read through the feedback carefully and try to understand the AI's suggestions.  
Group discussion: Within their project groups, students discuss the AI feedback they received. Focus on:  
  
Understanding the AI's feedback points.  
Identifying specific, actionable suggestions from the AI feedback.  
Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?  
  
  
Main Activity 2: "Integrating AI and Peer Feedback - Refinement Action Plan" (20 mins):  
  
Guide groups to integrate both AI feedback and peer feedback into a comprehensive project refinement plan.  
Group task: Refine their "Iteration Plan" from the previous lesson, now incorporating insights from the AI feedback. Prioritize feedback points from both sources. Determine specific actions to address both peer and AI suggestions.  
Encourage groups to consider:  
  
Which feedback points are most critical to address for improving project quality and clarity.  
Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.  
What are the most efficient and effective ways to implement the refinements within the remaining project time?  
  
Groups finalize their "Refinement Action Plan," now informed by both peer and AI feedback.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What was your experience of receiving AI-driven feedback on your project? What kind of feedback was most helpful or surprising?  
How did the AI feedback compare to the peer feedback you received? Were there similarities or differences?  
How did you integrate both AI and peer feedback into your project refinement plan?  
Do you think AI feedback is a valuable tool for learning and improving data science work? What are its potential benefits and limitations?  
  
Assessments:  
  
Formative Assessment: Review of student engagement with AI feedback – assess their ability to access, interpret, and analyze AI-generated feedback reports (if AI tool provides reports).  
Formative Assessment: Quality of group "Refinement Action Plans" – assess the extent to which they have incorporated both AI and peer feedback and planned actionable revisions.  
Observation of group work during the session – assess their collaboration in analyzing feedback and planning project refinements.  
  
Student Workbook:  
  
Instructions for accessing and using the AI project review tool (if applicable).  
Worksheet for analyzing and interpreting AI-generated feedback reports.  
"Feedback Integration Worksheet" to combine AI and peer feedback and prioritize revision points.  
Revised "Refinement Action Plan" template (updated to incorporate AI feedback).  
Reflection questions on the value and limitations of AI feedback in project work.  
  
Extension Activities:  
  
Students critically evaluate the quality and usefulness of the AI feedback they received. Discuss: Was the AI feedback accurate? Specific enough? Actionable? What types of feedback was the AI tool good at providing, and where did it fall short?  
Challenge: If possible, compare the feedback from different AI review tools (if multiple tools are available) on the same project notebook. Discuss the similarities and differences in feedback provided by different AI systems.

- Warm-Up (10 mins): "AI Feedback - Expectations and Limitations" - Class discussion: "What do you think AI could be good at providing feedback on for your data science projects? What might AI not be good at or what limitations might it have in giving feedback? What kind of feedback is best suited for humans vs. AI?" Discuss the strengths and weaknesses of automated feedback vs. human feedback. Emphasize that AI feedback is a tool to assist, not replace, human judgment.

- Main Activity 1: "AI-Driven Project Review Session" (40 mins):  
  
Introduce the AI-powered project review tool (if available - specific tool and access method will depend on chosen AI integration).  
Guide students on how to submit their Observable project notebooks to the AI review tool (e.g., uploading notebook file, sharing notebook link with the AI tool).  
Students submit their group project notebooks to the AI review tool.  
Explain the types of feedback the AI tool is expected to provide (depending on tool capabilities, examples):  
  
Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).  
Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).  
Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).  
(Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).  
  
Students receive and review the AI-generated feedback on their project notebooks. Encourage them to read through the feedback carefully and try to understand the AI's suggestions.  
Group discussion: Within their project groups, students discuss the AI feedback they received. Focus on:  
  
Understanding the AI's feedback points.  
Identifying specific, actionable suggestions from the AI feedback.  
Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?

- Introduce the AI-powered project review tool (if available - specific tool and access method will depend on chosen AI integration).

- Guide students on how to submit their Observable project notebooks to the AI review tool (e.g., uploading notebook file, sharing notebook link with the AI tool).

- Students submit their group project notebooks to the AI review tool.

- Explain the types of feedback the AI tool is expected to provide (depending on tool capabilities, examples):  
  
Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).  
Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).  
Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).  
(Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).

- Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).

- Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).

- Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).

- (Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).

- Students receive and review the AI-generated feedback on their project notebooks. Encourage them to read through the feedback carefully and try to understand the AI's suggestions.

- Group discussion: Within their project groups, students discuss the AI feedback they received. Focus on:  
  
Understanding the AI's feedback points.  
Identifying specific, actionable suggestions from the AI feedback.  
Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?

- Understanding the AI's feedback points.

- Identifying specific, actionable suggestions from the AI feedback.

- Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?

- Main Activity 2: "Integrating AI and Peer Feedback - Refinement Action Plan" (20 mins):  
  
Guide groups to integrate both AI feedback and peer feedback into a comprehensive project refinement plan.  
Group task: Refine their "Iteration Plan" from the previous lesson, now incorporating insights from the AI feedback. Prioritize feedback points from both sources. Determine specific actions to address both peer and AI suggestions.  
Encourage groups to consider:  
  
Which feedback points are most critical to address for improving project quality and clarity.  
Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.  
What are the most efficient and effective ways to implement the refinements within the remaining project time?  
  
Groups finalize their "Refinement Action Plan," now informed by both peer and AI feedback.

- Guide groups to integrate both AI feedback and peer feedback into a comprehensive project refinement plan.

- Group task: Refine their "Iteration Plan" from the previous lesson, now incorporating insights from the AI feedback. Prioritize feedback points from both sources. Determine specific actions to address both peer and AI suggestions.

- Encourage groups to consider:  
  
Which feedback points are most critical to address for improving project quality and clarity.  
Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.  
What are the most efficient and effective ways to implement the refinements within the remaining project time?

- Which feedback points are most critical to address for improving project quality and clarity.

- Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.

- What are the most efficient and effective ways to implement the refinements within the remaining project time?

- Groups finalize their "Refinement Action Plan," now informed by both peer and AI feedback.

- Reflection & Discussion (10 mins): Class discussion:  
  
What was your experience of receiving AI-driven feedback on your project? What kind of feedback was most helpful or surprising?  
How did the AI feedback compare to the peer feedback you received? Were there similarities or differences?  
How did you integrate both AI and peer feedback into your project refinement plan?  
Do you think AI feedback is a valuable tool for learning and improving data science work? What are its potential benefits and limitations?

- What was your experience of receiving AI-driven feedback on your project? What kind of feedback was most helpful or surprising?

- How did the AI feedback compare to the peer feedback you received? Were there similarities or differences?

- How did you integrate both AI and peer feedback into your project refinement plan?

- Do you think AI feedback is a valuable tool for learning and improving data science work? What are its potential benefits and limitations?

- Assessments:  
  
Formative Assessment: Review of student engagement with AI feedback – assess their ability to access, interpret, and analyze AI-generated feedback reports (if AI tool provides reports).  
Formative Assessment: Quality of group "Refinement Action Plans" – assess the extent to which they have incorporated both AI and peer feedback and planned actionable revisions.  
Observation of group work during the session – assess their collaboration in analyzing feedback and planning project refinements.

- Formative Assessment: Review of student engagement with AI feedback – assess their ability to access, interpret, and analyze AI-generated feedback reports (if AI tool provides reports).

- Formative Assessment: Quality of group "Refinement Action Plans" – assess the extent to which they have incorporated both AI and peer feedback and planned actionable revisions.

- Observation of group work during the session – assess their collaboration in analyzing feedback and planning project refinements.

- Student Workbook:  
  
Instructions for accessing and using the AI project review tool (if applicable).  
Worksheet for analyzing and interpreting AI-generated feedback reports.  
"Feedback Integration Worksheet" to combine AI and peer feedback and prioritize revision points.  
Revised "Refinement Action Plan" template (updated to incorporate AI feedback).  
Reflection questions on the value and limitations of AI feedback in project work.

- Instructions for accessing and using the AI project review tool (if applicable).

- Worksheet for analyzing and interpreting AI-generated feedback reports.

- "Feedback Integration Worksheet" to combine AI and peer feedback and prioritize revision points.

- Revised "Refinement Action Plan" template (updated to incorporate AI feedback).

- Reflection questions on the value and limitations of AI feedback in project work.

- Extension Activities:  
  
Students critically evaluate the quality and usefulness of the AI feedback they received. Discuss: Was the AI feedback accurate? Specific enough? Actionable? What types of feedback was the AI tool good at providing, and where did it fall short?  
Challenge: If possible, compare the feedback from different AI review tools (if multiple tools are available) on the same project notebook. Discuss the similarities and differences in feedback provided by different AI systems.

- Students critically evaluate the quality and usefulness of the AI feedback they received. Discuss: Was the AI feedback accurate? Specific enough? Actionable? What types of feedback was the AI tool good at providing, and where did it fall short?

- Challenge: If possible, compare the feedback from different AI review tools (if multiple tools are available) on the same project notebook. Discuss the similarities and differences in feedback provided by different AI systems.

- Lesson Materials:  
  
Slide Deck: Introduction to AI-driven project review, accessing and using the AI review tool (if applicable), interpreting AI feedback, integrating AI and peer feedback, project refinement planning.  
Handout: Instructions for AI review tool (if applicable), worksheet for analyzing AI feedback, "Feedback Integration Worksheet", revised "Refinement Action Plan" template.  
Access to AI project review tool (if applicable) and student project Observable notebooks.  
Computers/Tablets with internet access for accessing AI tools and project notebooks.

- Slide Deck: Introduction to AI-driven project review, accessing and using the AI review tool (if applicable), interpreting AI feedback, integrating AI and peer feedback, project refinement planning.

- Handout: Instructions for AI review tool (if applicable), worksheet for analyzing AI feedback, "Feedback Integration Worksheet", revised "Refinement Action Plan" template.

- Access to AI project review tool (if applicable) and student project Observable notebooks.

- Computers/Tablets with internet access for accessing AI tools and project notebooks.

- Students will be able to utilize AI-powered review tools (if available) to receive automated feedback on their project Observable notebooks, focusing on code quality, data analysis, and visualizations.

- Students will be able to analyse and interpret AI-generated feedback and identify specific areas for project improvement.

- Students will be able to integrate AI feedback with peer feedback to develop a comprehensive revision plan for their project.

- Students will be able to work collaboratively to implement project refinements based on both AI and peer feedback, enhancing their project notebooks and presentations.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-07 (Communicate information), SC4-WS-08 (Reflecting on scientific methods), ACSIS131 (Evaluating methods and conclusions).

- Warm-Up (10 mins): "AI Feedback - Expectations and Limitations" - Class discussion: "What do you think AI could be good at providing feedback on for your data science projects? What might AI not be good at or what limitations might it have in giving feedback? What kind of feedback is best suited for humans vs. AI?" Discuss the strengths and weaknesses of automated feedback vs. human feedback. Emphasize that AI feedback is a tool to assist, not replace, human judgment.

- Main Activity 1: "AI-Driven Project Review Session" (40 mins):  
  
Introduce the AI-powered project review tool (if available - specific tool and access method will depend on chosen AI integration).  
Guide students on how to submit their Observable project notebooks to the AI review tool (e.g., uploading notebook file, sharing notebook link with the AI tool).  
Students submit their group project notebooks to the AI review tool.  
Explain the types of feedback the AI tool is expected to provide (depending on tool capabilities, examples):  
  
Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).  
Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).  
Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).  
(Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).  
  
Students receive and review the AI-generated feedback on their project notebooks. Encourage them to read through the feedback carefully and try to understand the AI's suggestions.  
Group discussion: Within their project groups, students discuss the AI feedback they received. Focus on:  
  
Understanding the AI's feedback points.  
Identifying specific, actionable suggestions from the AI feedback.  
Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?

- Introduce the AI-powered project review tool (if available - specific tool and access method will depend on chosen AI integration).

- Guide students on how to submit their Observable project notebooks to the AI review tool (e.g., uploading notebook file, sharing notebook link with the AI tool).

- Students submit their group project notebooks to the AI review tool.

- Explain the types of feedback the AI tool is expected to provide (depending on tool capabilities, examples):  
  
Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).  
Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).  
Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).  
(Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).

- Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).

- Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).

- Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).

- (Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).

- Students receive and review the AI-generated feedback on their project notebooks. Encourage them to read through the feedback carefully and try to understand the AI's suggestions.

- Group discussion: Within their project groups, students discuss the AI feedback they received. Focus on:  
  
Understanding the AI's feedback points.  
Identifying specific, actionable suggestions from the AI feedback.  
Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?

- Understanding the AI's feedback points.

- Identifying specific, actionable suggestions from the AI feedback.

- Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?

- Main Activity 2: "Integrating AI and Peer Feedback - Refinement Action Plan" (20 mins):  
  
Guide groups to integrate both AI feedback and peer feedback into a comprehensive project refinement plan.  
Group task: Refine their "Iteration Plan" from the previous lesson, now incorporating insights from the AI feedback. Prioritize feedback points from both sources. Determine specific actions to address both peer and AI suggestions.  
Encourage groups to consider:  
  
Which feedback points are most critical to address for improving project quality and clarity.  
Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.  
What are the most efficient and effective ways to implement the refinements within the remaining project time?  
  
Groups finalize their "Refinement Action Plan," now informed by both peer and AI feedback.

- Guide groups to integrate both AI feedback and peer feedback into a comprehensive project refinement plan.

- Group task: Refine their "Iteration Plan" from the previous lesson, now incorporating insights from the AI feedback. Prioritize feedback points from both sources. Determine specific actions to address both peer and AI suggestions.

- Encourage groups to consider:  
  
Which feedback points are most critical to address for improving project quality and clarity.  
Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.  
What are the most efficient and effective ways to implement the refinements within the remaining project time?

- Which feedback points are most critical to address for improving project quality and clarity.

- Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.

- What are the most efficient and effective ways to implement the refinements within the remaining project time?

- Groups finalize their "Refinement Action Plan," now informed by both peer and AI feedback.

- Reflection & Discussion (10 mins): Class discussion:  
  
What was your experience of receiving AI-driven feedback on your project? What kind of feedback was most helpful or surprising?  
How did the AI feedback compare to the peer feedback you received? Were there similarities or differences?  
How did you integrate both AI and peer feedback into your project refinement plan?  
Do you think AI feedback is a valuable tool for learning and improving data science work? What are its potential benefits and limitations?

- What was your experience of receiving AI-driven feedback on your project? What kind of feedback was most helpful or surprising?

- How did the AI feedback compare to the peer feedback you received? Were there similarities or differences?

- How did you integrate both AI and peer feedback into your project refinement plan?

- Do you think AI feedback is a valuable tool for learning and improving data science work? What are its potential benefits and limitations?

- Assessments:  
  
Formative Assessment: Review of student engagement with AI feedback – assess their ability to access, interpret, and analyze AI-generated feedback reports (if AI tool provides reports).  
Formative Assessment: Quality of group "Refinement Action Plans" – assess the extent to which they have incorporated both AI and peer feedback and planned actionable revisions.  
Observation of group work during the session – assess their collaboration in analyzing feedback and planning project refinements.

- Formative Assessment: Review of student engagement with AI feedback – assess their ability to access, interpret, and analyze AI-generated feedback reports (if AI tool provides reports).

- Formative Assessment: Quality of group "Refinement Action Plans" – assess the extent to which they have incorporated both AI and peer feedback and planned actionable revisions.

- Observation of group work during the session – assess their collaboration in analyzing feedback and planning project refinements.

- Student Workbook:  
  
Instructions for accessing and using the AI project review tool (if applicable).  
Worksheet for analyzing and interpreting AI-generated feedback reports.  
"Feedback Integration Worksheet" to combine AI and peer feedback and prioritize revision points.  
Revised "Refinement Action Plan" template (updated to incorporate AI feedback).  
Reflection questions on the value and limitations of AI feedback in project work.

- Instructions for accessing and using the AI project review tool (if applicable).

- Worksheet for analyzing and interpreting AI-generated feedback reports.

- "Feedback Integration Worksheet" to combine AI and peer feedback and prioritize revision points.

- Revised "Refinement Action Plan" template (updated to incorporate AI feedback).

- Reflection questions on the value and limitations of AI feedback in project work.

- Extension Activities:  
  
Students critically evaluate the quality and usefulness of the AI feedback they received. Discuss: Was the AI feedback accurate? Specific enough? Actionable? What types of feedback was the AI tool good at providing, and where did it fall short?  
Challenge: If possible, compare the feedback from different AI review tools (if multiple tools are available) on the same project notebook. Discuss the similarities and differences in feedback provided by different AI systems.

- Students critically evaluate the quality and usefulness of the AI feedback they received. Discuss: Was the AI feedback accurate? Specific enough? Actionable? What types of feedback was the AI tool good at providing, and where did it fall short?

- Challenge: If possible, compare the feedback from different AI review tools (if multiple tools are available) on the same project notebook. Discuss the similarities and differences in feedback provided by different AI systems.

- Introduce the AI-powered project review tool (if available - specific tool and access method will depend on chosen AI integration).

- Guide students on how to submit their Observable project notebooks to the AI review tool (e.g., uploading notebook file, sharing notebook link with the AI tool).

- Students submit their group project notebooks to the AI review tool.

- Explain the types of feedback the AI tool is expected to provide (depending on tool capabilities, examples):  
  
Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).  
Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).  
Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).  
(Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).

- Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).

- Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).

- Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).

- (Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).

- Students receive and review the AI-generated feedback on their project notebooks. Encourage them to read through the feedback carefully and try to understand the AI's suggestions.

- Group discussion: Within their project groups, students discuss the AI feedback they received. Focus on:  
  
Understanding the AI's feedback points.  
Identifying specific, actionable suggestions from the AI feedback.  
Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?

- Understanding the AI's feedback points.

- Identifying specific, actionable suggestions from the AI feedback.

- Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?

- Code Quality Feedback: Basic code syntax checks, suggestions for code clarity or efficiency (if AI tool has code analysis capabilities).

- Data Analysis Feedback: Suggestions on data analysis techniques, potential missing analyses, or alternative approaches (if AI tool has data analysis awareness).

- Visualization Feedback: Suggestions on visualization clarity, appropriateness of graph types, labeling, potential improvements in visual communication (if AI tool has visualization analysis capabilities).

- (Potentially) Conceptual Feedback: High-level feedback on project question, data relevance, or conclusions (if AI tool has more advanced natural language understanding and project context awareness - less likely in a basic tool).

- Understanding the AI's feedback points.

- Identifying specific, actionable suggestions from the AI feedback.

- Comparing AI feedback to peer feedback they received in the previous lesson. Are there overlaps? Are there different perspectives?

- Guide groups to integrate both AI feedback and peer feedback into a comprehensive project refinement plan.

- Group task: Refine their "Iteration Plan" from the previous lesson, now incorporating insights from the AI feedback. Prioritize feedback points from both sources. Determine specific actions to address both peer and AI suggestions.

- Encourage groups to consider:  
  
Which feedback points are most critical to address for improving project quality and clarity.  
Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.  
What are the most efficient and effective ways to implement the refinements within the remaining project time?

- Which feedback points are most critical to address for improving project quality and clarity.

- Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.

- What are the most efficient and effective ways to implement the refinements within the remaining project time?

- Groups finalize their "Refinement Action Plan," now informed by both peer and AI feedback.

- Which feedback points are most critical to address for improving project quality and clarity.

- Are there any conflicting feedback points (from peers vs. AI)? How to resolve these conflicts or prioritize different types of feedback.

- What are the most efficient and effective ways to implement the refinements within the remaining project time?

- What was your experience of receiving AI-driven feedback on your project? What kind of feedback was most helpful or surprising?

- How did the AI feedback compare to the peer feedback you received? Were there similarities or differences?

- How did you integrate both AI and peer feedback into your project refinement plan?

- Do you think AI feedback is a valuable tool for learning and improving data science work? What are its potential benefits and limitations?

- Formative Assessment: Review of student engagement with AI feedback – assess their ability to access, interpret, and analyze AI-generated feedback reports (if AI tool provides reports).

- Formative Assessment: Quality of group "Refinement Action Plans" – assess the extent to which they have incorporated both AI and peer feedback and planned actionable revisions.

- Observation of group work during the session – assess their collaboration in analyzing feedback and planning project refinements.

- Instructions for accessing and using the AI project review tool (if applicable).

- Worksheet for analyzing and interpreting AI-generated feedback reports.

- "Feedback Integration Worksheet" to combine AI and peer feedback and prioritize revision points.

- Revised "Refinement Action Plan" template (updated to incorporate AI feedback).

- Reflection questions on the value and limitations of AI feedback in project work.

- Students critically evaluate the quality and usefulness of the AI feedback they received. Discuss: Was the AI feedback accurate? Specific enough? Actionable? What types of feedback was the AI tool good at providing, and where did it fall short?

- Challenge: If possible, compare the feedback from different AI review tools (if multiple tools are available) on the same project notebook. Discuss the similarities and differences in feedback provided by different AI systems.

- Slide Deck: Introduction to AI-driven project review, accessing and using the AI review tool (if applicable), interpreting AI feedback, integrating AI and peer feedback, project refinement planning.

- Handout: Instructions for AI review tool (if applicable), worksheet for analyzing AI feedback, "Feedback Integration Worksheet", revised "Refinement Action Plan" template.

- Access to AI project review tool (if applicable) and student project Observable notebooks.

- Computers/Tablets with internet access for accessing AI tools and project notebooks.

### Lesson 9.3: Exam Preparation and Advanced Data Analysis Practice

- Title: Data Science Mastery: Exam Prep and Advanced Practice

- Learning Outcomes:  
  
Students will be able to review key concepts and skills covered throughout the Data Science course in preparation for the end-of-semester exam.  
Students will be able to practice solving advanced data analysis problems that are representative of exam-style questions, applying their learned techniques.  
Students will be able to identify areas of strength and weakness in their data science knowledge and skills and focus their final exam preparation efforts effectively.  
NSW Curriculum Outcomes: SC4-WS-05 (Using data to develop conclusions), SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), ACSIS126 (Processing and analysing data), ACSIS131 (Evaluating methods and conclusions).

- Students will be able to review key concepts and skills covered throughout the Data Science course in preparation for the end-of-semester exam.

- Students will be able to practice solving advanced data analysis problems that are representative of exam-style questions, applying their learned techniques.

- Students will be able to identify areas of strength and weakness in their data science knowledge and skills and focus their final exam preparation efforts effectively.

- NSW Curriculum Outcomes: SC4-WS-05 (Using data to develop conclusions), SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), ACSIS126 (Processing and analysing data), ACSIS131 (Evaluating methods and conclusions).

- Overview: This lesson is dedicated to exam preparation and advanced data analysis practice. Students will review key course concepts, work through practice problems that mirror the exam format, and identify areas where they need to focus their final revision efforts to maximize exam performance.

- Activities:  
  
Warm-Up (10 mins): "Data Science Concepts Brain Dump" - Quick individual activity: Students spend 5 minutes writing down as many key data science concepts, techniques, and terms they can remember from the entire course (e.g., data types, data cleaning, descriptive statistics, graph types, experimental design, AI concepts). After 5 minutes, students briefly share a few items from their lists. This activates prior knowledge and highlights key topics for review.  
Main Activity 1: "Comprehensive Exam Review Session" (35 mins):  
  
Teacher-led or AI-assisted (if feasible) review session covering the key topics and skills from the 10-week course. Structure the review around the main themes and weeks of the curriculum:  
  
Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.  
Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.  
Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.  
Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.  
  
For each topic, briefly:  
  
Recap key concepts and definitions.  
Review relevant techniques and skills.  
Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).  
Ask a few quick review questions to check understanding.  
  
Use a slide deck or AI-generated review materials (if available) to structure the review session and keep it focused and efficient.  
  
Main Activity 2: "Advanced Data Analysis Practice Problems (Exam-Style)" (25 mins):  
  
Provide students with a set of practice data analysis problems that are designed to be similar in style and difficulty to questions they might encounter in the end-of-semester exam. Problems should cover:  
  
Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.  
Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.  
Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.  
  
Students work individually or in pairs on the practice problems in Observable notebooks.  
Teacher circulates to provide guidance and answer questions. Encourage students to apply the techniques they have learned throughout the course to solve the problems.  
After a set time, review the solutions to the practice problems as a class. Discuss different approaches and correct answers, clarifying any misunderstandings.  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are the key data science concepts and skills that you feel most confident about for the exam?  
What are the areas where you still feel less confident or need more review? Identify 1-2 specific topics or skills you plan to focus on in your final exam preparation.  
What are some effective strategies for reviewing and preparing for the data science exam in the remaining time? (e.g., reviewing notes, re-doing exercises, practicing coding, focusing on areas of weakness).  
Any final questions about the course content or exam format?  
  
Assessments:  
  
Formative Assessment: Review of student participation in the exam review session – assess their engagement and understanding of reviewed concepts through their responses to review questions and discussions.  
Formative Assessment: Review of student work on practice data analysis problems – assess their ability to apply learned techniques to solve exam-style problems and their understanding of data analysis methods.  
  
Student Workbook:  
  
Comprehensive exam review checklist covering key topics and skills from the course.  
Practice data analysis problems (exam-style) with datasets provided.  
Example solutions or answer keys for the practice problems.  
"Exam Preparation Action Plan" template to outline their individual study plan and focus areas for final revision.  
Formula and concept review sheets summarizing key definitions, formulas, and techniques.  
  
Lesson Materials:  
  
Slide Deck: Exam review slides covering key concepts and skills from the course, example code snippets, review questions.  
Handout: Practice data analysis problems (exam-style) with datasets, example solutions/answer keys, exam review checklist, "Exam Preparation Action Plan" template, formula and concept review sheets.  
Computers/Tablets with internet access for Observable (for practice problems).  
(Optional) AI-generated exam review materials or interactive review tools.

- Warm-Up (10 mins): "Data Science Concepts Brain Dump" - Quick individual activity: Students spend 5 minutes writing down as many key data science concepts, techniques, and terms they can remember from the entire course (e.g., data types, data cleaning, descriptive statistics, graph types, experimental design, AI concepts). After 5 minutes, students briefly share a few items from their lists. This activates prior knowledge and highlights key topics for review.

- Main Activity 1: "Comprehensive Exam Review Session" (35 mins):  
  
Teacher-led or AI-assisted (if feasible) review session covering the key topics and skills from the 10-week course. Structure the review around the main themes and weeks of the curriculum:  
  
Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.  
Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.  
Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.  
Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.  
  
For each topic, briefly:  
  
Recap key concepts and definitions.  
Review relevant techniques and skills.  
Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).  
Ask a few quick review questions to check understanding.  
  
Use a slide deck or AI-generated review materials (if available) to structure the review session and keep it focused and efficient.

- Teacher-led or AI-assisted (if feasible) review session covering the key topics and skills from the 10-week course. Structure the review around the main themes and weeks of the curriculum:  
  
Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.  
Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.  
Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.  
Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.

- Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.

- Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.

- Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.

- Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.

- For each topic, briefly:  
  
Recap key concepts and definitions.  
Review relevant techniques and skills.  
Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).  
Ask a few quick review questions to check understanding.

- Recap key concepts and definitions.

- Review relevant techniques and skills.

- Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).

- Ask a few quick review questions to check understanding.

- Use a slide deck or AI-generated review materials (if available) to structure the review session and keep it focused and efficient.

- Main Activity 2: "Advanced Data Analysis Practice Problems (Exam-Style)" (25 mins):  
  
Provide students with a set of practice data analysis problems that are designed to be similar in style and difficulty to questions they might encounter in the end-of-semester exam. Problems should cover:  
  
Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.  
Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.  
Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.  
  
Students work individually or in pairs on the practice problems in Observable notebooks.  
Teacher circulates to provide guidance and answer questions. Encourage students to apply the techniques they have learned throughout the course to solve the problems.  
After a set time, review the solutions to the practice problems as a class. Discuss different approaches and correct answers, clarifying any misunderstandings.

- Provide students with a set of practice data analysis problems that are designed to be similar in style and difficulty to questions they might encounter in the end-of-semester exam. Problems should cover:  
  
Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.  
Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.  
Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.

- Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.

- Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.

- Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.

- Students work individually or in pairs on the practice problems in Observable notebooks.

- Teacher circulates to provide guidance and answer questions. Encourage students to apply the techniques they have learned throughout the course to solve the problems.

- After a set time, review the solutions to the practice problems as a class. Discuss different approaches and correct answers, clarifying any misunderstandings.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the key data science concepts and skills that you feel most confident about for the exam?  
What are the areas where you still feel less confident or need more review? Identify 1-2 specific topics or skills you plan to focus on in your final exam preparation.  
What are some effective strategies for reviewing and preparing for the data science exam in the remaining time? (e.g., reviewing notes, re-doing exercises, practicing coding, focusing on areas of weakness).  
Any final questions about the course content or exam format?

- What are the key data science concepts and skills that you feel most confident about for the exam?

- What are the areas where you still feel less confident or need more review? Identify 1-2 specific topics or skills you plan to focus on in your final exam preparation.

- What are some effective strategies for reviewing and preparing for the data science exam in the remaining time? (e.g., reviewing notes, re-doing exercises, practicing coding, focusing on areas of weakness).

- Any final questions about the course content or exam format?

- Assessments:  
  
Formative Assessment: Review of student participation in the exam review session – assess their engagement and understanding of reviewed concepts through their responses to review questions and discussions.  
Formative Assessment: Review of student work on practice data analysis problems – assess their ability to apply learned techniques to solve exam-style problems and their understanding of data analysis methods.

- Formative Assessment: Review of student participation in the exam review session – assess their engagement and understanding of reviewed concepts through their responses to review questions and discussions.

- Formative Assessment: Review of student work on practice data analysis problems – assess their ability to apply learned techniques to solve exam-style problems and their understanding of data analysis methods.

- Student Workbook:  
  
Comprehensive exam review checklist covering key topics and skills from the course.  
Practice data analysis problems (exam-style) with datasets provided.  
Example solutions or answer keys for the practice problems.  
"Exam Preparation Action Plan" template to outline their individual study plan and focus areas for final revision.  
Formula and concept review sheets summarizing key definitions, formulas, and techniques.

- Comprehensive exam review checklist covering key topics and skills from the course.

- Practice data analysis problems (exam-style) with datasets provided.

- Example solutions or answer keys for the practice problems.

- "Exam Preparation Action Plan" template to outline their individual study plan and focus areas for final revision.

- Formula and concept review sheets summarizing key definitions, formulas, and techniques.

- Lesson Materials:  
  
Slide Deck: Exam review slides covering key concepts and skills from the course, example code snippets, review questions.  
Handout: Practice data analysis problems (exam-style) with datasets, example solutions/answer keys, exam review checklist, "Exam Preparation Action Plan" template, formula and concept review sheets.  
Computers/Tablets with internet access for Observable (for practice problems).  
(Optional) AI-generated exam review materials or interactive review tools.

- Slide Deck: Exam review slides covering key concepts and skills from the course, example code snippets, review questions.

- Handout: Practice data analysis problems (exam-style) with datasets, example solutions/answer keys, exam review checklist, "Exam Preparation Action Plan" template, formula and concept review sheets.

- Computers/Tablets with internet access for Observable (for practice problems).

- (Optional) AI-generated exam review materials or interactive review tools.

- Students will be able to review key concepts and skills covered throughout the Data Science course in preparation for the end-of-semester exam.

- Students will be able to practice solving advanced data analysis problems that are representative of exam-style questions, applying their learned techniques.

- Students will be able to identify areas of strength and weakness in their data science knowledge and skills and focus their final exam preparation efforts effectively.

- NSW Curriculum Outcomes: SC4-WS-05 (Using data to develop conclusions), SC4-WS-06 (Process data and information), SC4-WS-07 (Communicate information), ACSIS126 (Processing and analysing data), ACSIS131 (Evaluating methods and conclusions).

- Warm-Up (10 mins): "Data Science Concepts Brain Dump" - Quick individual activity: Students spend 5 minutes writing down as many key data science concepts, techniques, and terms they can remember from the entire course (e.g., data types, data cleaning, descriptive statistics, graph types, experimental design, AI concepts). After 5 minutes, students briefly share a few items from their lists. This activates prior knowledge and highlights key topics for review.

- Main Activity 1: "Comprehensive Exam Review Session" (35 mins):  
  
Teacher-led or AI-assisted (if feasible) review session covering the key topics and skills from the 10-week course. Structure the review around the main themes and weeks of the curriculum:  
  
Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.  
Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.  
Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.  
Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.  
  
For each topic, briefly:  
  
Recap key concepts and definitions.  
Review relevant techniques and skills.  
Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).  
Ask a few quick review questions to check understanding.  
  
Use a slide deck or AI-generated review materials (if available) to structure the review session and keep it focused and efficient.

- Teacher-led or AI-assisted (if feasible) review session covering the key topics and skills from the 10-week course. Structure the review around the main themes and weeks of the curriculum:  
  
Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.  
Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.  
Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.  
Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.

- Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.

- Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.

- Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.

- Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.

- For each topic, briefly:  
  
Recap key concepts and definitions.  
Review relevant techniques and skills.  
Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).  
Ask a few quick review questions to check understanding.

- Recap key concepts and definitions.

- Review relevant techniques and skills.

- Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).

- Ask a few quick review questions to check understanding.

- Use a slide deck or AI-generated review materials (if available) to structure the review session and keep it focused and efficient.

- Main Activity 2: "Advanced Data Analysis Practice Problems (Exam-Style)" (25 mins):  
  
Provide students with a set of practice data analysis problems that are designed to be similar in style and difficulty to questions they might encounter in the end-of-semester exam. Problems should cover:  
  
Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.  
Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.  
Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.  
  
Students work individually or in pairs on the practice problems in Observable notebooks.  
Teacher circulates to provide guidance and answer questions. Encourage students to apply the techniques they have learned throughout the course to solve the problems.  
After a set time, review the solutions to the practice problems as a class. Discuss different approaches and correct answers, clarifying any misunderstandings.

- Provide students with a set of practice data analysis problems that are designed to be similar in style and difficulty to questions they might encounter in the end-of-semester exam. Problems should cover:  
  
Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.  
Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.  
Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.

- Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.

- Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.

- Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.

- Students work individually or in pairs on the practice problems in Observable notebooks.

- Teacher circulates to provide guidance and answer questions. Encourage students to apply the techniques they have learned throughout the course to solve the problems.

- After a set time, review the solutions to the practice problems as a class. Discuss different approaches and correct answers, clarifying any misunderstandings.

- Reflection & Discussion (10 mins): Class discussion:  
  
What are the key data science concepts and skills that you feel most confident about for the exam?  
What are the areas where you still feel less confident or need more review? Identify 1-2 specific topics or skills you plan to focus on in your final exam preparation.  
What are some effective strategies for reviewing and preparing for the data science exam in the remaining time? (e.g., reviewing notes, re-doing exercises, practicing coding, focusing on areas of weakness).  
Any final questions about the course content or exam format?

- What are the key data science concepts and skills that you feel most confident about for the exam?

- What are the areas where you still feel less confident or need more review? Identify 1-2 specific topics or skills you plan to focus on in your final exam preparation.

- What are some effective strategies for reviewing and preparing for the data science exam in the remaining time? (e.g., reviewing notes, re-doing exercises, practicing coding, focusing on areas of weakness).

- Any final questions about the course content or exam format?

- Assessments:  
  
Formative Assessment: Review of student participation in the exam review session – assess their engagement and understanding of reviewed concepts through their responses to review questions and discussions.  
Formative Assessment: Review of student work on practice data analysis problems – assess their ability to apply learned techniques to solve exam-style problems and their understanding of data analysis methods.

- Formative Assessment: Review of student participation in the exam review session – assess their engagement and understanding of reviewed concepts through their responses to review questions and discussions.

- Formative Assessment: Review of student work on practice data analysis problems – assess their ability to apply learned techniques to solve exam-style problems and their understanding of data analysis methods.

- Student Workbook:  
  
Comprehensive exam review checklist covering key topics and skills from the course.  
Practice data analysis problems (exam-style) with datasets provided.  
Example solutions or answer keys for the practice problems.  
"Exam Preparation Action Plan" template to outline their individual study plan and focus areas for final revision.  
Formula and concept review sheets summarizing key definitions, formulas, and techniques.

- Comprehensive exam review checklist covering key topics and skills from the course.

- Practice data analysis problems (exam-style) with datasets provided.

- Example solutions or answer keys for the practice problems.

- "Exam Preparation Action Plan" template to outline their individual study plan and focus areas for final revision.

- Formula and concept review sheets summarizing key definitions, formulas, and techniques.

- Lesson Materials:  
  
Slide Deck: Exam review slides covering key concepts and skills from the course, example code snippets, review questions.  
Handout: Practice data analysis problems (exam-style) with datasets, example solutions/answer keys, exam review checklist, "Exam Preparation Action Plan" template, formula and concept review sheets.  
Computers/Tablets with internet access for Observable (for practice problems).  
(Optional) AI-generated exam review materials or interactive review tools.

- Slide Deck: Exam review slides covering key concepts and skills from the course, example code snippets, review questions.

- Handout: Practice data analysis problems (exam-style) with datasets, example solutions/answer keys, exam review checklist, "Exam Preparation Action Plan" template, formula and concept review sheets.

- Computers/Tablets with internet access for Observable (for practice problems).

- (Optional) AI-generated exam review materials or interactive review tools.

- Teacher-led or AI-assisted (if feasible) review session covering the key topics and skills from the 10-week course. Structure the review around the main themes and weeks of the curriculum:  
  
Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.  
Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.  
Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.  
Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.

- Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.

- Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.

- Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.

- Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.

- For each topic, briefly:  
  
Recap key concepts and definitions.  
Review relevant techniques and skills.  
Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).  
Ask a few quick review questions to check understanding.

- Recap key concepts and definitions.

- Review relevant techniques and skills.

- Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).

- Ask a few quick review questions to check understanding.

- Use a slide deck or AI-generated review materials (if available) to structure the review session and keep it focused and efficient.

- Week 1-2: Foundations of Data Science and Data Collection: Review definition of data science, data types, data sources, digital footprints (briefly), data quality, accuracy, precision, validity, scientific questions.

- Week 3-4: Data Representation and Descriptive Statistics: Review tables, bar graphs, line graphs, scatter plots, mean, median, mode, range, outliers.

- Week 5-6: Experimental Design and Data Cleaning: Review experimental design principles (variables, controls), data cleaning techniques (missing values, format standardization), data sorting and filtering.

- Week 7: Introduction to AI in Data Science: Review basic AI concepts in data analysis (pattern recognition, summarization, predictive modelling – conceptually), ethical considerations of AI.

- Recap key concepts and definitions.

- Review relevant techniques and skills.

- Show a quick example or demonstration (e.g., a quick Python code snippet in Observable to calculate mean, create a graph, or filter data).

- Ask a few quick review questions to check understanding.

- Provide students with a set of practice data analysis problems that are designed to be similar in style and difficulty to questions they might encounter in the end-of-semester exam. Problems should cover:  
  
Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.  
Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.  
Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.

- Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.

- Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.

- Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.

- Students work individually or in pairs on the practice problems in Observable notebooks.

- Teacher circulates to provide guidance and answer questions. Encourage students to apply the techniques they have learned throughout the course to solve the problems.

- After a set time, review the solutions to the practice problems as a class. Discuss different approaches and correct answers, clarifying any misunderstandings.

- Data Interpretation: Present tables or graphs and ask students to interpret data, identify trends, draw conclusions, and answer specific questions based on the data.

- Data Analysis Tasks: Present datasets (in CSV or table format) and ask students to perform data analysis tasks in Observable using Python/Pandas: calculate descriptive statistics, sort and filter data, create specific visualizations to answer given questions.

- Short Answer/Conceptual Questions: Include some short answer questions that test their understanding of key data science concepts, definitions, and ethical considerations.

- What are the key data science concepts and skills that you feel most confident about for the exam?

- What are the areas where you still feel less confident or need more review? Identify 1-2 specific topics or skills you plan to focus on in your final exam preparation.

- What are some effective strategies for reviewing and preparing for the data science exam in the remaining time? (e.g., reviewing notes, re-doing exercises, practicing coding, focusing on areas of weakness).

- Any final questions about the course content or exam format?

- Formative Assessment: Review of student participation in the exam review session – assess their engagement and understanding of reviewed concepts through their responses to review questions and discussions.

- Formative Assessment: Review of student work on practice data analysis problems – assess their ability to apply learned techniques to solve exam-style problems and their understanding of data analysis methods.

- Comprehensive exam review checklist covering key topics and skills from the course.

- Practice data analysis problems (exam-style) with datasets provided.

- Example solutions or answer keys for the practice problems.

- "Exam Preparation Action Plan" template to outline their individual study plan and focus areas for final revision.

- Formula and concept review sheets summarizing key definitions, formulas, and techniques.

- Slide Deck: Exam review slides covering key concepts and skills from the course, example code snippets, review questions.

- Handout: Practice data analysis problems (exam-style) with datasets, example solutions/answer keys, exam review checklist, "Exam Preparation Action Plan" template, formula and concept review sheets.

- Computers/Tablets with internet access for Observable (for practice problems).

- (Optional) AI-generated exam review materials or interactive review tools.

### Week 10: End-of-Semester Exam & Future Pathways

### Lesson 10.1: End-of-Semester Exam (Part 1 - Data Interpretation and Short Answer)

- Title: Exam Challenge Part 1: Data Interpretation and Knowledge Check

- Learning Outcomes:  
  
Students will be able to demonstrate their ability to interpret data presented in tables and graphs.  
Students will be able to answer short-answer questions accurately, demonstrating their understanding of key data science concepts and terminology.  
Students will be able to apply their data analysis and reasoning skills to solve data interpretation problems within the given time constraints.  
NSW Curriculum Outcomes: SC4-WS-05 (Using data to develop conclusions), SC4-WS-06 (Process data and information), ACSIS126 (Processing and analysing data), ACSIS131 (Evaluating methods and conclusions).

- Students will be able to demonstrate their ability to interpret data presented in tables and graphs.

- Students will be able to answer short-answer questions accurately, demonstrating their understanding of key data science concepts and terminology.

- Students will be able to apply their data analysis and reasoning skills to solve data interpretation problems within the given time constraints.

- NSW Curriculum Outcomes: SC4-WS-05 (Using data to develop conclusions), SC4-WS-06 (Process data and information), ACSIS126 (Processing and analysing data), ACSIS131 (Evaluating methods and conclusions).

- Overview: This lesson is dedicated to Part 1 of the end-of-semester exam, focusing on assessing data interpretation skills and conceptual understanding through data interpretation questions and short answer questions.

- Activities:  
  
Main Activity 1: "End-of-Semester Exam - Part 1 (Data Interpretation and Short Answer)" (70 mins):  
  
Administer Part 1 of the End-of-Semester Exam in a supervised exam setting.  
Exam format (as outlined in the Course Plan):  
  
Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.  
Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.  
  
Provide clear instructions for the exam, time limits for each section (if applicable), and any specific materials allowed (e.g., non-programmable calculator, reference sheets - as per school exam policy).  
Ensure a quiet and focused exam environment.  
  
Post-Exam (Optional - if time allows): If there is any remaining time after the exam, students can have a quiet reflection period or begin working on other tasks.  
Assessments:  
  
Summative Assessment: End-of-Semester Exam - Part 1. Automated grading (if using AI-generated questions and digital exam format) and/or teacher grading based on pre-defined rubrics for data interpretation and short answer questions.  
  
Student Workbook:  
  
(No specific workbook activities for this lesson - exam session).  
  
Lesson Materials:  
  
End-of-Semester Exam - Part 1 paper (printed or digital format).  
Answer sheets or digital exam platform for student responses.  
Any allowed materials (calculators, reference sheets - as per exam policy).  
Supervision and exam administration materials.

- Main Activity 1: "End-of-Semester Exam - Part 1 (Data Interpretation and Short Answer)" (70 mins):  
  
Administer Part 1 of the End-of-Semester Exam in a supervised exam setting.  
Exam format (as outlined in the Course Plan):  
  
Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.  
Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.  
  
Provide clear instructions for the exam, time limits for each section (if applicable), and any specific materials allowed (e.g., non-programmable calculator, reference sheets - as per school exam policy).  
Ensure a quiet and focused exam environment.

- Administer Part 1 of the End-of-Semester Exam in a supervised exam setting.

- Exam format (as outlined in the Course Plan):  
  
Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.  
Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.

- Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.

- Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.

- Provide clear instructions for the exam, time limits for each section (if applicable), and any specific materials allowed (e.g., non-programmable calculator, reference sheets - as per school exam policy).

- Ensure a quiet and focused exam environment.

- Post-Exam (Optional - if time allows): If there is any remaining time after the exam, students can have a quiet reflection period or begin working on other tasks.

- Assessments:  
  
Summative Assessment: End-of-Semester Exam - Part 1. Automated grading (if using AI-generated questions and digital exam format) and/or teacher grading based on pre-defined rubrics for data interpretation and short answer questions.

- Summative Assessment: End-of-Semester Exam - Part 1. Automated grading (if using AI-generated questions and digital exam format) and/or teacher grading based on pre-defined rubrics for data interpretation and short answer questions.

- Student Workbook:  
  
(No specific workbook activities for this lesson - exam session).

- (No specific workbook activities for this lesson - exam session).

- Lesson Materials:  
  
End-of-Semester Exam - Part 1 paper (printed or digital format).  
Answer sheets or digital exam platform for student responses.  
Any allowed materials (calculators, reference sheets - as per exam policy).  
Supervision and exam administration materials.

- End-of-Semester Exam - Part 1 paper (printed or digital format).

- Answer sheets or digital exam platform for student responses.

- Any allowed materials (calculators, reference sheets - as per exam policy).

- Supervision and exam administration materials.

- Students will be able to demonstrate their ability to interpret data presented in tables and graphs.

- Students will be able to answer short-answer questions accurately, demonstrating their understanding of key data science concepts and terminology.

- Students will be able to apply their data analysis and reasoning skills to solve data interpretation problems within the given time constraints.

- NSW Curriculum Outcomes: SC4-WS-05 (Using data to develop conclusions), SC4-WS-06 (Process data and information), ACSIS126 (Processing and analysing data), ACSIS131 (Evaluating methods and conclusions).

- Main Activity 1: "End-of-Semester Exam - Part 1 (Data Interpretation and Short Answer)" (70 mins):  
  
Administer Part 1 of the End-of-Semester Exam in a supervised exam setting.  
Exam format (as outlined in the Course Plan):  
  
Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.  
Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.  
  
Provide clear instructions for the exam, time limits for each section (if applicable), and any specific materials allowed (e.g., non-programmable calculator, reference sheets - as per school exam policy).  
Ensure a quiet and focused exam environment.

- Administer Part 1 of the End-of-Semester Exam in a supervised exam setting.

- Exam format (as outlined in the Course Plan):  
  
Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.  
Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.

- Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.

- Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.

- Provide clear instructions for the exam, time limits for each section (if applicable), and any specific materials allowed (e.g., non-programmable calculator, reference sheets - as per school exam policy).

- Ensure a quiet and focused exam environment.

- Post-Exam (Optional - if time allows): If there is any remaining time after the exam, students can have a quiet reflection period or begin working on other tasks.

- Assessments:  
  
Summative Assessment: End-of-Semester Exam - Part 1. Automated grading (if using AI-generated questions and digital exam format) and/or teacher grading based on pre-defined rubrics for data interpretation and short answer questions.

- Summative Assessment: End-of-Semester Exam - Part 1. Automated grading (if using AI-generated questions and digital exam format) and/or teacher grading based on pre-defined rubrics for data interpretation and short answer questions.

- Student Workbook:  
  
(No specific workbook activities for this lesson - exam session).

- (No specific workbook activities for this lesson - exam session).

- Lesson Materials:  
  
End-of-Semester Exam - Part 1 paper (printed or digital format).  
Answer sheets or digital exam platform for student responses.  
Any allowed materials (calculators, reference sheets - as per exam policy).  
Supervision and exam administration materials.

- End-of-Semester Exam - Part 1 paper (printed or digital format).

- Answer sheets or digital exam platform for student responses.

- Any allowed materials (calculators, reference sheets - as per exam policy).

- Supervision and exam administration materials.

- Administer Part 1 of the End-of-Semester Exam in a supervised exam setting.

- Exam format (as outlined in the Course Plan):  
  
Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.  
Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.

- Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.

- Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.

- Provide clear instructions for the exam, time limits for each section (if applicable), and any specific materials allowed (e.g., non-programmable calculator, reference sheets - as per school exam policy).

- Ensure a quiet and focused exam environment.

- Section 1: Data Interpretation: Questions based on provided tables and graphs. Students will need to interpret data, identify trends, draw conclusions, and answer specific questions based on the presented data visualizations. Question types may include multiple-choice, short answer, and graph/table analysis questions.

- Section 2: Short Answer Questions: Short answer questions testing students' understanding of key data science concepts, definitions, terminology, and ethical considerations covered throughout the course.

- Summative Assessment: End-of-Semester Exam - Part 1. Automated grading (if using AI-generated questions and digital exam format) and/or teacher grading based on pre-defined rubrics for data interpretation and short answer questions.

- (No specific workbook activities for this lesson - exam session).

- End-of-Semester Exam - Part 1 paper (printed or digital format).

- Answer sheets or digital exam platform for student responses.

- Any allowed materials (calculators, reference sheets - as per exam policy).

- Supervision and exam administration materials.

### Lesson 10.2: End-of-Semester Exam (Part 2 - Coding and Notebook Tasks)

- Title: Exam Challenge Part 2: Coding and Data Analysis in Action

- Learning Outcomes:  
  
Students will be able to demonstrate their ability to perform basic coding tasks in Python within Observable notebooks to solve data analysis problems.  
Students will be able to apply data analysis techniques learned in the course (descriptive statistics, sorting, filtering, basic visualizations) to analyze a given dataset in Observable and answer specific questions.  
Students will be able to complete coding and notebook tasks effectively within the given time constraints, demonstrating their practical data science skills.  
NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-05 (Using data to develop conclusions), SC4-WS-06 (Process data and information), ACSIS126 (Processing and analysing data).

- Students will be able to demonstrate their ability to perform basic coding tasks in Python within Observable notebooks to solve data analysis problems.

- Students will be able to apply data analysis techniques learned in the course (descriptive statistics, sorting, filtering, basic visualizations) to analyze a given dataset in Observable and answer specific questions.

- Students will be able to complete coding and notebook tasks effectively within the given time constraints, demonstrating their practical data science skills.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-05 (Using data to develop conclusions), SC4-WS-06 (Process data and information), ACSIS126 (Processing and analysing data).

- Overview: This lesson is dedicated to Part 2 of the end-of-semester exam, focusing on assessing practical data science skills through coding and notebook tasks in Observable.

- Activities:  
  
Main Activity 1: "End-of-Semester Exam - Part 2 (Coding and Notebook Tasks)" (70 mins):  
  
Administer Part 2 of the End-of-Semester Exam in a supervised exam setting, with students working on computers/tablets with access to Observable.  
Exam format (as outlined in the Course Plan):  
  
Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.  
  
  
Provide clear instructions for the exam, the dataset(s) to be used, specific tasks to be completed in Observable notebooks, time limits, and any allowed resources (e.g., access to Pandas documentation - as per exam policy).  
Ensure students have access to computers/tablets with internet and Observable. Provide technical support as needed for basic access issues but not for coding problem-solving during the exam itself.  
Students work individually to complete the coding and notebook tasks within the exam time.  
  
Post-Exam (Optional - if time allows): If there is any remaining time after the exam, students can have a quiet reflection period or begin working on other tasks.  
Assessments:  
  
Summative Assessment: End-of-Semester Exam - Part 2. Automated grading (if using AI-assisted grading for code and notebook content) and/or teacher grading based on pre-defined rubrics for code correctness, data analysis accuracy, visualization quality, and interpretation of results.  
  
Student Workbook:  
  
(No specific workbook activities for this lesson - exam session).  
  
Lesson Materials:  
  
End-of-Semester Exam - Part 2 paper (printed or digital format) with instructions and data analysis tasks.  
Digital datasets (CSV or similar) to be used for the exam tasks (accessible within the exam environment).  
Computers/Tablets with internet access and Observable for each student.  
Supervision and exam administration materials.

- Main Activity 1: "End-of-Semester Exam - Part 2 (Coding and Notebook Tasks)" (70 mins):  
  
Administer Part 2 of the End-of-Semester Exam in a supervised exam setting, with students working on computers/tablets with access to Observable.  
Exam format (as outlined in the Course Plan):  
  
Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.  
  
  
Provide clear instructions for the exam, the dataset(s) to be used, specific tasks to be completed in Observable notebooks, time limits, and any allowed resources (e.g., access to Pandas documentation - as per exam policy).  
Ensure students have access to computers/tablets with internet and Observable. Provide technical support as needed for basic access issues but not for coding problem-solving during the exam itself.  
Students work individually to complete the coding and notebook tasks within the exam time.

- Administer Part 2 of the End-of-Semester Exam in a supervised exam setting, with students working on computers/tablets with access to Observable.

- Exam format (as outlined in the Course Plan):  
  
Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Loading a provided dataset into a Pandas DataFrame.

- Calculating descriptive statistics for specific columns.

- Sorting and filtering data based on given criteria.

- Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.

- Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Provide clear instructions for the exam, the dataset(s) to be used, specific tasks to be completed in Observable notebooks, time limits, and any allowed resources (e.g., access to Pandas documentation - as per exam policy).

- Ensure students have access to computers/tablets with internet and Observable. Provide technical support as needed for basic access issues but not for coding problem-solving during the exam itself.

- Students work individually to complete the coding and notebook tasks within the exam time.

- Post-Exam (Optional - if time allows): If there is any remaining time after the exam, students can have a quiet reflection period or begin working on other tasks.

- Assessments:  
  
Summative Assessment: End-of-Semester Exam - Part 2. Automated grading (if using AI-assisted grading for code and notebook content) and/or teacher grading based on pre-defined rubrics for code correctness, data analysis accuracy, visualization quality, and interpretation of results.

- Summative Assessment: End-of-Semester Exam - Part 2. Automated grading (if using AI-assisted grading for code and notebook content) and/or teacher grading based on pre-defined rubrics for code correctness, data analysis accuracy, visualization quality, and interpretation of results.

- Student Workbook:  
  
(No specific workbook activities for this lesson - exam session).

- (No specific workbook activities for this lesson - exam session).

- Lesson Materials:  
  
End-of-Semester Exam - Part 2 paper (printed or digital format) with instructions and data analysis tasks.  
Digital datasets (CSV or similar) to be used for the exam tasks (accessible within the exam environment).  
Computers/Tablets with internet access and Observable for each student.  
Supervision and exam administration materials.

- End-of-Semester Exam - Part 2 paper (printed or digital format) with instructions and data analysis tasks.

- Digital datasets (CSV or similar) to be used for the exam tasks (accessible within the exam environment).

- Computers/Tablets with internet access and Observable for each student.

- Supervision and exam administration materials.

- Students will be able to demonstrate their ability to perform basic coding tasks in Python within Observable notebooks to solve data analysis problems.

- Students will be able to apply data analysis techniques learned in the course (descriptive statistics, sorting, filtering, basic visualizations) to analyze a given dataset in Observable and answer specific questions.

- Students will be able to complete coding and notebook tasks effectively within the given time constraints, demonstrating their practical data science skills.

- NSW Curriculum Outcomes: SC4-WS-03 (Using digital technologies), SC4-WS-05 (Using data to develop conclusions), SC4-WS-06 (Process data and information), ACSIS126 (Processing and analysing data).

- Main Activity 1: "End-of-Semester Exam - Part 2 (Coding and Notebook Tasks)" (70 mins):  
  
Administer Part 2 of the End-of-Semester Exam in a supervised exam setting, with students working on computers/tablets with access to Observable.  
Exam format (as outlined in the Course Plan):  
  
Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.  
  
  
Provide clear instructions for the exam, the dataset(s) to be used, specific tasks to be completed in Observable notebooks, time limits, and any allowed resources (e.g., access to Pandas documentation - as per exam policy).  
Ensure students have access to computers/tablets with internet and Observable. Provide technical support as needed for basic access issues but not for coding problem-solving during the exam itself.  
Students work individually to complete the coding and notebook tasks within the exam time.

- Administer Part 2 of the End-of-Semester Exam in a supervised exam setting, with students working on computers/tablets with access to Observable.

- Exam format (as outlined in the Course Plan):  
  
Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Loading a provided dataset into a Pandas DataFrame.

- Calculating descriptive statistics for specific columns.

- Sorting and filtering data based on given criteria.

- Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.

- Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Provide clear instructions for the exam, the dataset(s) to be used, specific tasks to be completed in Observable notebooks, time limits, and any allowed resources (e.g., access to Pandas documentation - as per exam policy).

- Ensure students have access to computers/tablets with internet and Observable. Provide technical support as needed for basic access issues but not for coding problem-solving during the exam itself.

- Students work individually to complete the coding and notebook tasks within the exam time.

- Post-Exam (Optional - if time allows): If there is any remaining time after the exam, students can have a quiet reflection period or begin working on other tasks.

- Assessments:  
  
Summative Assessment: End-of-Semester Exam - Part 2. Automated grading (if using AI-assisted grading for code and notebook content) and/or teacher grading based on pre-defined rubrics for code correctness, data analysis accuracy, visualization quality, and interpretation of results.

- Summative Assessment: End-of-Semester Exam - Part 2. Automated grading (if using AI-assisted grading for code and notebook content) and/or teacher grading based on pre-defined rubrics for code correctness, data analysis accuracy, visualization quality, and interpretation of results.

- Student Workbook:  
  
(No specific workbook activities for this lesson - exam session).

- (No specific workbook activities for this lesson - exam session).

- Lesson Materials:  
  
End-of-Semester Exam - Part 2 paper (printed or digital format) with instructions and data analysis tasks.  
Digital datasets (CSV or similar) to be used for the exam tasks (accessible within the exam environment).  
Computers/Tablets with internet access and Observable for each student.  
Supervision and exam administration materials.

- End-of-Semester Exam - Part 2 paper (printed or digital format) with instructions and data analysis tasks.

- Digital datasets (CSV or similar) to be used for the exam tasks (accessible within the exam environment).

- Computers/Tablets with internet access and Observable for each student.

- Supervision and exam administration materials.

- Administer Part 2 of the End-of-Semester Exam in a supervised exam setting, with students working on computers/tablets with access to Observable.

- Exam format (as outlined in the Course Plan):  
  
Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Loading a provided dataset into a Pandas DataFrame.

- Calculating descriptive statistics for specific columns.

- Sorting and filtering data based on given criteria.

- Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.

- Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Provide clear instructions for the exam, the dataset(s) to be used, specific tasks to be completed in Observable notebooks, time limits, and any allowed resources (e.g., access to Pandas documentation - as per exam policy).

- Ensure students have access to computers/tablets with internet and Observable. Provide technical support as needed for basic access issues but not for coding problem-solving during the exam itself.

- Students work individually to complete the coding and notebook tasks within the exam time.

- Section 2: Coding and Notebook Tasks: Practical data analysis tasks to be completed in Observable notebooks using Python/Pandas. Tasks may include:  
  
Loading a provided dataset into a Pandas DataFrame.  
Calculating descriptive statistics for specific columns.  
Sorting and filtering data based on given criteria.  
Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.  
Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Loading a provided dataset into a Pandas DataFrame.

- Calculating descriptive statistics for specific columns.

- Sorting and filtering data based on given criteria.

- Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.

- Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Loading a provided dataset into a Pandas DataFrame.

- Calculating descriptive statistics for specific columns.

- Sorting and filtering data based on given criteria.

- Creating specific visualizations (bar graphs, line graphs, scatter plots) to answer questions about the data.

- Interpreting the results of their code and visualizations to answer exam questions in markdown cells.

- Summative Assessment: End-of-Semester Exam - Part 2. Automated grading (if using AI-assisted grading for code and notebook content) and/or teacher grading based on pre-defined rubrics for code correctness, data analysis accuracy, visualization quality, and interpretation of results.

- (No specific workbook activities for this lesson - exam session).

- End-of-Semester Exam - Part 2 paper (printed or digital format) with instructions and data analysis tasks.

- Digital datasets (CSV or similar) to be used for the exam tasks (accessible within the exam environment).

- Computers/Tablets with internet access and Observable for each student.

- Supervision and exam administration materials.

### Lesson 10.3: Future Pathways in Data Science & Course Reflection

- Title: Beyond Year 8: Data Science Futures and Course Reflection

- Learning Outcomes:  
  
Students will be able to explore potential future pathways and career options related to data science and STEM fields.  
Students will be able to reflect on their learning and skill development throughout the Data Science course, identifying their strengths and areas for growth.  
Students will be able to articulate their interest in pursuing further learning or exploration in data science and related fields.  
NSW Curriculum Outcomes: SC4-WS-08 (Reflecting on scientific methods), SC4-3VA (Nature and Development of Science - Science as a human endeavour), SC4-1VA, SC4-2VA, SC4-3VA (Values and attitudes in science).

- Students will be able to explore potential future pathways and career options related to data science and STEM fields.

- Students will be able to reflect on their learning and skill development throughout the Data Science course, identifying their strengths and areas for growth.

- Students will be able to articulate their interest in pursuing further learning or exploration in data science and related fields.

- NSW Curriculum Outcomes: SC4-WS-08 (Reflecting on scientific methods), SC4-3VA (Nature and Development of Science - Science as a human endeavour), SC4-1VA, SC4-2VA, SC4-3VA (Values and attitudes in science).

- Overview: This final lesson provides closure to the course, focusing on future pathways in data science and STEM, and student reflection on their learning journey. It aims to inspire continued interest in data science and encourage self-assessment of skills developed throughout the program.

- Activities:  
  
Warm-Up (10 mins): "Data Science Careers Brainstorm" - Class brainstorm: "What kinds of jobs or careers do you think might involve data science skills? What types of industries or fields use data science?" (Examples: technology, science research, business, healthcare, government, environmental organizations, sports analytics, etc.). List brainstormed careers and industries on the board.  
Main Activity 1: "Exploring Future Pathways in Data Science and STEM" (30 mins):  
  
Present information on future pathways and career options related to data science and STEM fields. This could include:  
  
Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.  
Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.  
Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.  
Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.  
  
Show short videos or presentations featuring data scientists talking about their work and career paths (if available and engaging for Year 8).  
Invite guest speakers (if feasible) – data scientists from industry or academia to share their experiences and career advice with students.  
  
Main Activity 2: "Course Reflection and Skill Self-Assessment" (30 mins):  
  
Guide students through a structured self-reflection activity on their learning and skill development throughout the 10-week Data Science course.  
Provide a "Course Reflection Questionnaire" (see Student Workbook). Questionnaire should include prompts for students to reflect on:  
  
Key concepts and skills they learned in the course.  
Specific activities or projects they enjoyed or found most valuable.  
Data science skills they feel they have developed most strongly.  
Areas where they feel they have improved and areas where they still want to learn more.  
Challenges they overcame during the course.  
Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.  
Their potential interest in pursuing further learning or career paths related to data science or STEM.  
  
Students individually complete the Course Reflection Questionnaire.  
Optional: After individual reflection, facilitate a brief class discussion where students can voluntarily share some of their reflections and key takeaways from the course (no pressure to share anything too personal).  
  
Reflection & Discussion (10 mins): Class discussion:  
  
What are your key takeaways from the Year 8 Data Science course overall? What are you most proud of learning or accomplishing?  
Has this course changed your perspective on data science or STEM fields? How?  
Are you interested in exploring data science further in the future? What are some next steps you might consider (e.g., further courses, online resources, projects)?  
Any final thoughts or feedback about the course itself? What did you find most effective or enjoyable? What could be improved? (Collect student feedback to inform future course iterations).  
  
Assessments:  
  
Formative Assessment: Review of student completed "Course Reflection Questionnaires" – assess the depth and thoughtfulness of their self-reflection on learning and skill development.  
Observation of student participation in class discussions about future pathways and course reflection.  
  
Student Workbook:  
  
Information handouts or links about future pathways and career options in data science and STEM fields.  
"Course Reflection Questionnaire" template with prompts for self-assessment and reflection on learning.  
List of resources for continued learning in data science (online courses, websites, communities, etc.).  
Course feedback form for students to provide constructive feedback on the program.  
  
Lesson Materials:  
  
Slide Deck: Future pathways in data science and STEM (higher education, career options, job market trends, societal impact), course reflection prompts, resources for continued learning.  
Handout: Information handouts on future pathways, "Course Reflection Questionnaire" template, course feedback form, list of resources for continued learning.  
(Optional) Video clips or presentations featuring data scientists.  
(Optional) Guest speaker from data science or STEM field.

- Warm-Up (10 mins): "Data Science Careers Brainstorm" - Class brainstorm: "What kinds of jobs or careers do you think might involve data science skills? What types of industries or fields use data science?" (Examples: technology, science research, business, healthcare, government, environmental organizations, sports analytics, etc.). List brainstormed careers and industries on the board.

- Main Activity 1: "Exploring Future Pathways in Data Science and STEM" (30 mins):  
  
Present information on future pathways and career options related to data science and STEM fields. This could include:  
  
Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.  
Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.  
Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.  
Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.  
  
Show short videos or presentations featuring data scientists talking about their work and career paths (if available and engaging for Year 8).  
Invite guest speakers (if feasible) – data scientists from industry or academia to share their experiences and career advice with students.

- Present information on future pathways and career options related to data science and STEM fields. This could include:  
  
Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.  
Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.  
Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.  
Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.

- Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.

- Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.

- Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.

- Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.

- Show short videos or presentations featuring data scientists talking about their work and career paths (if available and engaging for Year 8).

- Invite guest speakers (if feasible) – data scientists from industry or academia to share their experiences and career advice with students.

- Main Activity 2: "Course Reflection and Skill Self-Assessment" (30 mins):  
  
Guide students through a structured self-reflection activity on their learning and skill development throughout the 10-week Data Science course.  
Provide a "Course Reflection Questionnaire" (see Student Workbook). Questionnaire should include prompts for students to reflect on:  
  
Key concepts and skills they learned in the course.  
Specific activities or projects they enjoyed or found most valuable.  
Data science skills they feel they have developed most strongly.  
Areas where they feel they have improved and areas where they still want to learn more.  
Challenges they overcame during the course.  
Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.  
Their potential interest in pursuing further learning or career paths related to data science or STEM.  
  
Students individually complete the Course Reflection Questionnaire.  
Optional: After individual reflection, facilitate a brief class discussion where students can voluntarily share some of their reflections and key takeaways from the course (no pressure to share anything too personal).

- Guide students through a structured self-reflection activity on their learning and skill development throughout the 10-week Data Science course.

- Provide a "Course Reflection Questionnaire" (see Student Workbook). Questionnaire should include prompts for students to reflect on:  
  
Key concepts and skills they learned in the course.  
Specific activities or projects they enjoyed or found most valuable.  
Data science skills they feel they have developed most strongly.  
Areas where they feel they have improved and areas where they still want to learn more.  
Challenges they overcame during the course.  
Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.  
Their potential interest in pursuing further learning or career paths related to data science or STEM.

- Key concepts and skills they learned in the course.

- Specific activities or projects they enjoyed or found most valuable.

- Data science skills they feel they have developed most strongly.

- Areas where they feel they have improved and areas where they still want to learn more.

- Challenges they overcame during the course.

- Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.

- Their potential interest in pursuing further learning or career paths related to data science or STEM.

- Students individually complete the Course Reflection Questionnaire.

- Optional: After individual reflection, facilitate a brief class discussion where students can voluntarily share some of their reflections and key takeaways from the course (no pressure to share anything too personal).

- Reflection & Discussion (10 mins): Class discussion:  
  
What are your key takeaways from the Year 8 Data Science course overall? What are you most proud of learning or accomplishing?  
Has this course changed your perspective on data science or STEM fields? How?  
Are you interested in exploring data science further in the future? What are some next steps you might consider (e.g., further courses, online resources, projects)?  
Any final thoughts or feedback about the course itself? What did you find most effective or enjoyable? What could be improved? (Collect student feedback to inform future course iterations).

- What are your key takeaways from the Year 8 Data Science course overall? What are you most proud of learning or accomplishing?

- Has this course changed your perspective on data science or STEM fields? How?

- Are you interested in exploring data science further in the future? What are some next steps you might consider (e.g., further courses, online resources, projects)?

- Any final thoughts or feedback about the course itself? What did you find most effective or enjoyable? What could be improved? (Collect student feedback to inform future course iterations).

- Assessments:  
  
Formative Assessment: Review of student completed "Course Reflection Questionnaires" – assess the depth and thoughtfulness of their self-reflection on learning and skill development.  
Observation of student participation in class discussions about future pathways and course reflection.

- Formative Assessment: Review of student completed "Course Reflection Questionnaires" – assess the depth and thoughtfulness of their self-reflection on learning and skill development.

- Observation of student participation in class discussions about future pathways and course reflection.

- Student Workbook:  
  
Information handouts or links about future pathways and career options in data science and STEM fields.  
"Course Reflection Questionnaire" template with prompts for self-assessment and reflection on learning.  
List of resources for continued learning in data science (online courses, websites, communities, etc.).  
Course feedback form for students to provide constructive feedback on the program.

- Information handouts or links about future pathways and career options in data science and STEM fields.

- "Course Reflection Questionnaire" template with prompts for self-assessment and reflection on learning.

- List of resources for continued learning in data science (online courses, websites, communities, etc.).

- Course feedback form for students to provide constructive feedback on the program.

- Lesson Materials:  
  
Slide Deck: Future pathways in data science and STEM (higher education, career options, job market trends, societal impact), course reflection prompts, resources for continued learning.  
Handout: Information handouts on future pathways, "Course Reflection Questionnaire" template, course feedback form, list of resources for continued learning.  
(Optional) Video clips or presentations featuring data scientists.  
(Optional) Guest speaker from data science or STEM field.

- Slide Deck: Future pathways in data science and STEM (higher education, career options, job market trends, societal impact), course reflection prompts, resources for continued learning.

- Handout: Information handouts on future pathways, "Course Reflection Questionnaire" template, course feedback form, list of resources for continued learning.

- (Optional) Video clips or presentations featuring data scientists.

- (Optional) Guest speaker from data science or STEM field.

- Students will be able to explore potential future pathways and career options related to data science and STEM fields.

- Students will be able to reflect on their learning and skill development throughout the Data Science course, identifying their strengths and areas for growth.

- Students will be able to articulate their interest in pursuing further learning or exploration in data science and related fields.

- NSW Curriculum Outcomes: SC4-WS-08 (Reflecting on scientific methods), SC4-3VA (Nature and Development of Science - Science as a human endeavour), SC4-1VA, SC4-2VA, SC4-3VA (Values and attitudes in science).

- Warm-Up (10 mins): "Data Science Careers Brainstorm" - Class brainstorm: "What kinds of jobs or careers do you think might involve data science skills? What types of industries or fields use data science?" (Examples: technology, science research, business, healthcare, government, environmental organizations, sports analytics, etc.). List brainstormed careers and industries on the board.

- Main Activity 1: "Exploring Future Pathways in Data Science and STEM" (30 mins):  
  
Present information on future pathways and career options related to data science and STEM fields. This could include:  
  
Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.  
Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.  
Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.  
Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.  
  
Show short videos or presentations featuring data scientists talking about their work and career paths (if available and engaging for Year 8).  
Invite guest speakers (if feasible) – data scientists from industry or academia to share their experiences and career advice with students.

- Present information on future pathways and career options related to data science and STEM fields. This could include:  
  
Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.  
Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.  
Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.  
Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.

- Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.

- Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.

- Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.

- Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.

- Show short videos or presentations featuring data scientists talking about their work and career paths (if available and engaging for Year 8).

- Invite guest speakers (if feasible) – data scientists from industry or academia to share their experiences and career advice with students.

- Main Activity 2: "Course Reflection and Skill Self-Assessment" (30 mins):  
  
Guide students through a structured self-reflection activity on their learning and skill development throughout the 10-week Data Science course.  
Provide a "Course Reflection Questionnaire" (see Student Workbook). Questionnaire should include prompts for students to reflect on:  
  
Key concepts and skills they learned in the course.  
Specific activities or projects they enjoyed or found most valuable.  
Data science skills they feel they have developed most strongly.  
Areas where they feel they have improved and areas where they still want to learn more.  
Challenges they overcame during the course.  
Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.  
Their potential interest in pursuing further learning or career paths related to data science or STEM.  
  
Students individually complete the Course Reflection Questionnaire.  
Optional: After individual reflection, facilitate a brief class discussion where students can voluntarily share some of their reflections and key takeaways from the course (no pressure to share anything too personal).

- Guide students through a structured self-reflection activity on their learning and skill development throughout the 10-week Data Science course.

- Provide a "Course Reflection Questionnaire" (see Student Workbook). Questionnaire should include prompts for students to reflect on:  
  
Key concepts and skills they learned in the course.  
Specific activities or projects they enjoyed or found most valuable.  
Data science skills they feel they have developed most strongly.  
Areas where they feel they have improved and areas where they still want to learn more.  
Challenges they overcame during the course.  
Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.  
Their potential interest in pursuing further learning or career paths related to data science or STEM.

- Key concepts and skills they learned in the course.

- Specific activities or projects they enjoyed or found most valuable.

- Data science skills they feel they have developed most strongly.

- Areas where they feel they have improved and areas where they still want to learn more.

- Challenges they overcame during the course.

- Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.

- Their potential interest in pursuing further learning or career paths related to data science or STEM.

- Students individually complete the Course Reflection Questionnaire.

- Optional: After individual reflection, facilitate a brief class discussion where students can voluntarily share some of their reflections and key takeaways from the course (no pressure to share anything too personal).

- Reflection & Discussion (10 mins): Class discussion:  
  
What are your key takeaways from the Year 8 Data Science course overall? What are you most proud of learning or accomplishing?  
Has this course changed your perspective on data science or STEM fields? How?  
Are you interested in exploring data science further in the future? What are some next steps you might consider (e.g., further courses, online resources, projects)?  
Any final thoughts or feedback about the course itself? What did you find most effective or enjoyable? What could be improved? (Collect student feedback to inform future course iterations).

- What are your key takeaways from the Year 8 Data Science course overall? What are you most proud of learning or accomplishing?

- Has this course changed your perspective on data science or STEM fields? How?

- Are you interested in exploring data science further in the future? What are some next steps you might consider (e.g., further courses, online resources, projects)?

- Any final thoughts or feedback about the course itself? What did you find most effective or enjoyable? What could be improved? (Collect student feedback to inform future course iterations).

- Assessments:  
  
Formative Assessment: Review of student completed "Course Reflection Questionnaires" – assess the depth and thoughtfulness of their self-reflection on learning and skill development.  
Observation of student participation in class discussions about future pathways and course reflection.

- Formative Assessment: Review of student completed "Course Reflection Questionnaires" – assess the depth and thoughtfulness of their self-reflection on learning and skill development.

- Observation of student participation in class discussions about future pathways and course reflection.

- Student Workbook:  
  
Information handouts or links about future pathways and career options in data science and STEM fields.  
"Course Reflection Questionnaire" template with prompts for self-assessment and reflection on learning.  
List of resources for continued learning in data science (online courses, websites, communities, etc.).  
Course feedback form for students to provide constructive feedback on the program.

- Information handouts or links about future pathways and career options in data science and STEM fields.

- "Course Reflection Questionnaire" template with prompts for self-assessment and reflection on learning.

- List of resources for continued learning in data science (online courses, websites, communities, etc.).

- Course feedback form for students to provide constructive feedback on the program.

- Lesson Materials:  
  
Slide Deck: Future pathways in data science and STEM (higher education, career options, job market trends, societal impact), course reflection prompts, resources for continued learning.  
Handout: Information handouts on future pathways, "Course Reflection Questionnaire" template, course feedback form, list of resources for continued learning.  
(Optional) Video clips or presentations featuring data scientists.  
(Optional) Guest speaker from data science or STEM field.

- Slide Deck: Future pathways in data science and STEM (higher education, career options, job market trends, societal impact), course reflection prompts, resources for continued learning.

- Handout: Information handouts on future pathways, "Course Reflection Questionnaire" template, course feedback form, list of resources for continued learning.

- (Optional) Video clips or presentations featuring data scientists.

- (Optional) Guest speaker from data science or STEM field.

- Present information on future pathways and career options related to data science and STEM fields. This could include:  
  
Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.  
Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.  
Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.  
Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.

- Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.

- Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.

- Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.

- Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.

- Show short videos or presentations featuring data scientists talking about their work and career paths (if available and engaging for Year 8).

- Invite guest speakers (if feasible) – data scientists from industry or academia to share their experiences and career advice with students.

- Higher Education Pathways: Discuss relevant university degrees and courses in data science, computer science, statistics, mathematics, and other STEM fields that build upon data science foundations. Highlight specific degree programs and universities known for data science education.

- Emerging Job Roles in Data Science: Describe some common job titles in data science and related fields (e.g., Data Scientist, Data Analyst, Business Analyst, Data Engineer, Machine Learning Engineer, AI Researcher, Bioinformatics Analyst, Environmental Data Scientist, etc.). Briefly explain the typical responsibilities and skills required for these roles.

- Growing Demand for Data Science Skills: Present data or statistics showing the increasing demand for data science skills across various industries and the projected job growth in these fields. Emphasize the strong career prospects for individuals with data science expertise.

- Impact of Data Science on Society: Reiterate the broad impact of data science in addressing real-world problems, driving innovation, and shaping the future across diverse sectors. Inspire students to consider how they can use data science skills to make a positive impact.

- Guide students through a structured self-reflection activity on their learning and skill development throughout the 10-week Data Science course.

- Provide a "Course Reflection Questionnaire" (see Student Workbook). Questionnaire should include prompts for students to reflect on:  
  
Key concepts and skills they learned in the course.  
Specific activities or projects they enjoyed or found most valuable.  
Data science skills they feel they have developed most strongly.  
Areas where they feel they have improved and areas where they still want to learn more.  
Challenges they overcame during the course.  
Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.  
Their potential interest in pursuing further learning or career paths related to data science or STEM.

- Key concepts and skills they learned in the course.

- Specific activities or projects they enjoyed or found most valuable.

- Data science skills they feel they have developed most strongly.

- Areas where they feel they have improved and areas where they still want to learn more.

- Challenges they overcame during the course.

- Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.

- Their potential interest in pursuing further learning or career paths related to data science or STEM.

- Students individually complete the Course Reflection Questionnaire.

- Optional: After individual reflection, facilitate a brief class discussion where students can voluntarily share some of their reflections and key takeaways from the course (no pressure to share anything too personal).

- Key concepts and skills they learned in the course.

- Specific activities or projects they enjoyed or found most valuable.

- Data science skills they feel they have developed most strongly.

- Areas where they feel they have improved and areas where they still want to learn more.

- Challenges they overcame during the course.

- Their overall experience in the course and their level of interest in data science now compared to the beginning of the course.

- Their potential interest in pursuing further learning or career paths related to data science or STEM.

- What are your key takeaways from the Year 8 Data Science course overall? What are you most proud of learning or accomplishing?

- Has this course changed your perspective on data science or STEM fields? How?

- Are you interested in exploring data science further in the future? What are some next steps you might consider (e.g., further courses, online resources, projects)?

- Any final thoughts or feedback about the course itself? What did you find most effective or enjoyable? What could be improved? (Collect student feedback to inform future course iterations).

- Formative Assessment: Review of student completed "Course Reflection Questionnaires" – assess the depth and thoughtfulness of their self-reflection on learning and skill development.

- Observation of student participation in class discussions about future pathways and course reflection.

- Information handouts or links about future pathways and career options in data science and STEM fields.

- "Course Reflection Questionnaire" template with prompts for self-assessment and reflection on learning.

- List of resources for continued learning in data science (online courses, websites, communities, etc.).

- Course feedback form for students to provide constructive feedback on the program.

- Slide Deck: Future pathways in data science and STEM (higher education, career options, job market trends, societal impact), course reflection prompts, resources for continued learning.

- Handout: Information handouts on future pathways, "Course Reflection Questionnaire" template, course feedback form, list of resources for continued learning.

- (Optional) Video clips or presentations featuring data scientists.

- (Optional) Guest speaker from data science or STEM field.