UCLA COMPUTER SCIENCE FIELD NOTES

CS M148

Introduction to Data Science

Data questions, cleaning, visualization, statistical models, pipelines, ethics, and communication

How these notes are written. I wrote each chapter the way I wish the material had been explained the first time: the real idea, the point where I usually get stuck, the simplest mental model that still stays honest, and a worked decision instead of a polished answer appearing from nowhere. Every chapter closes with practice and a fully written solution. If a sentence sounds impressive but does not help me solve or explain something, it does not belong here.

Daniel Mastick

Curriculum map, working notes, practice, and solutions

Course map

Week	Core thread	What should be solid by the end
1	**Questions, populations, and data-generating processes**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
2	**Cleaning and reproducible data pipelines**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
3	**Exploration and visualization**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
4	**Probability, sampling, and uncertainty**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
5	**Regression and classification**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
6	**Evaluation and experimental design**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
7	**Unsupervised learning and dimensionality reduction**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
8	**Ethics, deployment, and communication**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
9	**Integration workshop**	Connect at least three chapters in one end-to-end problem and explain where their contracts meet.
10	**Synthesis and project review**	Audit assumptions and explain a complete solution from interface to failure handling.

Scope anchor: https://catalog.registrar.ucla.edu/course/2025/COMSCIM148. The sequence is aligned to the official catalog description and expanded into a practical ten-week study plan.

How I would use this packet

Treat each numbered section as one chapter. Read the formal explanation, pause at the student interpretation, and see whether the easy version still says the same thing. Rework the example without looking, then cover the solution in the chapter practice box. The cumulative set at the end is deliberately mixed: knowledge becomes durable when I have to choose the tool instead of being told which chapter I am in.

Contents

CHAPTER 1 Questions, populations, and data-generating processes

The idea

Data science begins by defining the decision, target population, unit of observation, and how measurements were produced. Selection, survivorship, leakage, and proxy bias can dominate model choice.

Rebuild the chapter from first principles

The claim I need to own is this: Data science begins by defining the decision, target population, unit of observation, and how measurements were produced. Selection, survivorship, leakage, and proxy bias can dominate model choice. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of questions, populations, and data-generating processes. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach questions, populations, and data-generating processes on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from the course foundations to cleaning and reproducible data pipelines. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Data science begins by defining the decision, target population, unit of observation, and how measurements were produced.
- **Mechanism.** Selection, survivorship, leakage, and proxy bias can dominate model choice.
- **Boundary.** The useful test is whether I can apply questions, populations, and data-generating processes to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when questions, populations, and data-generating processes is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I ask how each row came to exist before I calculate anything.

The easy version

Data science begins by defining the decision, target population, unit of observation, and how measurements were produced.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why can a large convenience sample still be biased? The conclusion is Size reduces sampling variance but not systematic selection error. If inclusion depends on the outcome or population traits, more rows estimate the wrong distribution precisely.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why can a large convenience sample still be biased?
Solution. Size reduces sampling variance but not systematic selection error. If inclusion depends on the outcome or population traits, more rows estimate the wrong distribution precisely.

Practice 2. Give a short oral explanation of questions, populations, and data-generating processes that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct questions, populations, and data-generating processes from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a questions, populations, and data-generating processes problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 2 Cleaning and reproducible data pipelines

The idea

Schemas, types, missingness, duplicates, joins, units, and provenance turn raw files into analyzable tables. Cleaning must be scripted, logged, testable, and separated from irreversible source data.

Rebuild the chapter from first principles

The claim I need to own is this: Schemas, types, missingness, duplicates, joins, units, and provenance turn raw files into analyzable tables. Cleaning must be scripted, logged, testable, and separated from irreversible source data. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of cleaning and reproducible data pipelines. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach cleaning and reproducible data pipelines on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from questions, populations, and data-generating processes to exploration and visualization. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Schemas, types, missingness, duplicates, joins, units, and provenance turn raw files into analyzable tables.
- **Mechanism.** Cleaning must be scripted, logged, testable, and separated from irreversible source data.
- **Boundary.** The useful test is whether I can apply cleaning and reproducible data pipelines to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when cleaning and reproducible data pipelines is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I treat a join as a claim about identity and cardinality, not a convenience operation.

The easy version

Schemas, types, missingness, duplicates, joins, units, and provenance turn raw files into analyzable tables.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: How do you detect an accidental many-to-many join? The conclusion is Assert key uniqueness on the expected side, compare row counts, inspect unmatched keys, and validate aggregate totals before and after the join.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. How do you detect an accidental many-to-many join?
Solution. Assert key uniqueness on the expected side, compare row counts, inspect unmatched keys, and validate aggregate totals before and after the join.

Practice 2. Give a short oral explanation of cleaning and reproducible data pipelines that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.

Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct cleaning and reproducible data pipelines from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a cleaning and reproducible data pipelines problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 3 Exploration and visualization

The idea

Exploratory analysis studies distributions, relationships, groups, missingness, and outliers. Visual encodings should match the question; scales, aggregation, uncertainty, and denominator choices can change the story.

Rebuild the chapter from first principles

The claim I need to own is this: Exploratory analysis studies distributions, relationships, groups, missingness, and outliers. Visual encodings should match the question; scales, aggregation, uncertainty, and denominator choices can change the story. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of exploration and visualization. List the known quantities, the unknown, the units or types, and any hidden constraint.

2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.

3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach exploration and visualization on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from cleaning and reproducible data pipelines to probability, sampling, and uncertainty. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Exploratory analysis studies distributions, relationships, groups, missingness, and outliers.
- **Mechanism.** Visual encodings should match the question; scales, aggregation, uncertainty, and denominator choices can change the story.

- **Boundary.** The useful test is whether I can apply exploration and visualization to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when exploration and visualization is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I write the sentence a chart should support before choosing its geometry.

The easy version

Exploratory analysis studies distributions, relationships, groups, missingness, and outliers.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why can a truncated y-axis mislead? The conclusion is It magnifies small absolute differences. Use a justified scale, label it clearly, and show context or an alternate view when magnitude matters.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why can a truncated y-axis mislead?
Solution. It magnifies small absolute differences. Use a justified scale, label it clearly, and show context or an alternate view when magnitude matters.

Practice 2. Give a short oral explanation of exploration and visualization that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct exploration and visualization from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a exploration and visualization problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output?

Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 4 Probability, sampling, and uncertainty

The idea

Random variables, conditional probability, expectation, variance, sampling distributions, confidence intervals, and hypothesis tests quantify uncertainty. Practical importance and uncertainty should accompany p-values.

Rebuild the chapter from first principles

The claim I need to own is this: Random variables, conditional probability, expectation, variance, sampling distributions, confidence intervals, and hypothesis tests quantify uncertainty. Practical importance and uncertainty should accompany p-values. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of probability, sampling, and uncertainty. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes,

the invariant breaks, or the model stops matching reality.

- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach probability, sampling, and uncertainty on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from exploration and visualization to regression and classification. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Random variables, conditional probability, expectation, variance, sampling distributions, confidence intervals, and hypothesis tests quantify uncertainty.
- **Mechanism.** Practical importance and uncertainty should accompany p-values.
- **Boundary.** The useful test is whether I can apply probability, sampling, and uncertainty to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when probability, sampling, and uncertainty is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I separate uncertainty about a population estimate from variability in individual outcomes.

The easy version

Random variables, conditional probability, expectation, variance, sampling distributions, confidence intervals, and hypothesis tests quantify uncertainty.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Interpret a 95 percent confidence interval. The conclusion is Under repeated use of the procedure, about 95 percent of constructed intervals cover the fixed parameter. It is not a 95 percent probability statement about this fixed interval under the frequentist model.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Interpret a 95 percent confidence interval.
Solution. Under repeated use of the procedure, about 95 percent of constructed intervals cover the fixed parameter. It is not a 95 percent probability statement about this fixed interval under the frequentist model.

Practice 2. Give a short oral explanation of probability, sampling, and uncertainty that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct probability, sampling, and uncertainty from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a probability, sampling, and uncertainty problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 5 Regression and classification

The idea

Regression estimates conditional numeric outcomes; classification estimates labels or probabilities. Loss functions define what errors cost. Regularization controls variance, while feature engineering and preprocessing must be learned only from training data.

Rebuild the chapter from first principles

The claim I need to own is this: Regression estimates conditional numeric outcomes; classification estimates labels or probabilities. Loss functions define what errors cost. Regularization controls variance, while feature engineering and preprocessing must be learned only from training data. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of regression and classification. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach regression and classification on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from probability, sampling, and uncertainty to evaluation and experimental design. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Regression estimates conditional numeric outcomes; classification estimates labels or probabilities.
- **Mechanism.** Loss functions define what errors cost.
- **Boundary.** Regularization controls variance, while feature engineering and preprocessing must be learned only from training data.
- **Decision test.** I should be able to say when regression and classification is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I define the prediction target, evaluation metric, and baseline before fitting a complicated model.

The easy version

Regression estimates conditional numeric outcomes; classification estimates labels or probabilities.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why must scaling occur inside cross-validation? The conclusion is Scaling on all data leaks validation distribution information into training. Fit preprocessing on each training fold and apply it to that fold's validation data.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why must scaling occur inside cross-validation?
Solution. Scaling on all data leaks validation distribution information into training. Fit preprocessing on each training fold and apply it to that fold's validation data.

Practice 2. Give a short oral explanation of regression and classification that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct regression and classification from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a regression and classification problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 6 Evaluation and experimental design

The idea

Train, validation, and test splits serve different roles. Cross-validation estimates selection uncertainty; calibration, class imbalance, subgroup performance, and cost-sensitive thresholds matter beyond accuracy. Randomized experiments support causal claims when implementation is sound.

Rebuild the chapter from first principles

The claim I need to own is this: Train, validation, and test splits serve different roles. Cross-validation estimates selection uncertainty; calibration, class imbalance, subgroup performance, and cost-sensitive thresholds matter beyond accuracy. Randomized experiments support causal claims when implementation is sound. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of evaluation and experimental design. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach evaluation and experimental design on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from regression and classification to unsupervised learning and dimensionality reduction. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Train, validation, and test splits serve different roles.
- **Mechanism.** Cross-validation estimates selection uncertainty; calibration, class imbalance, subgroup performance, and cost-sensitive thresholds matter beyond accuracy.
- **Boundary.** Randomized experiments support causal claims when implementation is sound.
- **Decision test.** I should be able to say when evaluation and experimental design is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I keep the test set closed until decisions are finished; repeated peeking turns it into another validation set.

The easy version

Train, validation, and test splits serve different roles.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: A classifier is 99 percent accurate on a 1 percent-positive dataset. Is it useful? The conclusion is Not necessarily: always predicting negative is 99 percent accurate. Inspect recall, precision, calibration, confusion costs, and performance against that baseline.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. A classifier is 99 percent accurate on a 1 percent-positive dataset. Is it useful?
Solution. Not necessarily: always predicting negative is 99 percent accurate. Inspect recall, precision, calibration, confusion costs, and performance against that baseline.

Practice 2. Give a short oral explanation of evaluation and experimental design that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that

licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct evaluation and experimental design from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a evaluation and experimental design problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 7 Unsupervised learning and dimensionality reduction

The idea

Clustering imposes a similarity structure; PCA finds orthogonal directions of maximum variance. Results depend on scaling, distance, chosen dimension, and stability. Unsupervised patterns are hypotheses, not automatically meaningful categories.

Rebuild the chapter from first principles

The claim I need to own is this: Clustering imposes a similarity structure; PCA finds orthogonal directions of maximum variance. Results depend on scaling, distance, chosen dimension, and stability. Unsupervised patterns are hypotheses, not automatically meaningful categories. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of unsupervised learning and dimensionality reduction. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that

I can audit it later.

4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach unsupervised learning and dimensionality reduction on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from evaluation and experimental design to ethics, deployment, and communication. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Clustering imposes a similarity structure; PCA finds orthogonal directions of maximum variance.
- **Mechanism.** Results depend on scaling, distance, chosen dimension, and stability.
- **Boundary.** Unsupervised patterns are hypotheses, not automatically meaningful categories.

- **Decision test.** I should be able to say when unsupervised learning and dimensionality reduction is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I ask whether the distance metric matches the domain before trusting a cluster plot.

The easy version

Clustering imposes a similarity structure; PCA finds orthogonal directions of maximum variance.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: What does the first principal component optimize? The conclusion is It is the unit direction maximizing projected sample variance, equivalently minimizing squared reconstruction error for a one-dimensional linear approximation.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. What does the first principal component optimize?
Solution. It is the unit direction maximizing projected sample variance, equivalently minimizing squared reconstruction error for a one-dimensional linear approximation.

Practice 2. Give a short oral explanation of unsupervised learning and dimensionality reduction that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct unsupervised learning and dimensionality reduction from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a unsupervised learning and dimensionality reduction problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output?

Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 8 Ethics, deployment, and communication

The idea

Privacy, consent, fairness, documentation, monitoring, and distribution shift are part of model quality. A useful report connects methods to a decision, states limitations, and makes uncertainty visible without hiding behind jargon.

Rebuild the chapter from first principles

The claim I need to own is this: Privacy, consent, fairness, documentation, monitoring, and distribution shift are part of model quality. A useful report connects methods to a decision, states limitations, and makes uncertainty visible without hiding behind jargon. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of ethics, deployment, and communication. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes,

the invariant breaks, or the model stops matching reality.

- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach ethics, deployment, and communication on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from unsupervised learning and dimensionality reduction to the cumulative course model. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Privacy, consent, fairness, documentation, monitoring, and distribution shift are part of model quality.
- **Mechanism.** A useful report connects methods to a decision, states limitations, and makes uncertainty visible without hiding behind jargon.
- **Boundary.** The useful test is whether I can apply ethics, deployment, and communication to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when ethics, deployment, and communication is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

If I cannot explain who bears the cost of an error, the analysis is not ready to ship.

The easy version

Privacy, consent, fairness, documentation, monitoring, and distribution shift are part of model quality.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: What belongs in a model card? The conclusion is Intended use, data and evaluation scope, metrics and subgroup results, limitations, ethical risks, training details, monitoring needs, and uses the model should not support.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. What belongs in a model card?
Solution. Intended use, data and evaluation scope, metrics and subgroup results, limitations, ethical risks, training details, monitoring needs, and uses the model should not support.

Practice 2. Give a short oral explanation of ethics, deployment, and communication that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct ethics, deployment, and communication from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a ethics, deployment, and communication problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

Final study check

I should be able to close this packet and reconstruct the core models, not merely recognize their names. My final check is to explain one complete problem aloud: what the inputs mean, which invariant or contract controls the work, why the method is legal, what it costs, how it can fail, and what evidence would convince another engineer.