CASSION · COURSE

Food Security Analysis and IPC

Compute FCS, rCSI, HHS and LCS correctly, then use them the way an IPC analysis workshop does — as converging evidence, not as a single score.

Instructor	Pierre Robentz Cassion
Level	Intermediate
Learner hours	18 h
Taught in	Python · R
Updated	July 28, 2026
Sectors	Food Security · Livelihoods
Standards and methodologies	Integrated Food Security Phase Classification (IPC) · Sphere Standards · Core Humanitarian Standard (CHS)

Read and run the course online at data-analysis.cassion.dev/courses/food-security-ipc

What you will be able to do

- Compute the Food Consumption Score from its raw components, and state which threshold set you applied and why
- Score the Household Hunger Scale and reduced Coping Strategy Index with the exclusion rules each one requires
- Classify a household on the Livelihood Coping Strategies module by its most severe strategy rather than by a count
- Read a price series through its seasonality, and detect a market panel that changed composition
- Compute terms of trade and explain what they show that neither price shows alone
- Assemble an evidence table and say what a Phase 3 classification does and does not assert

Prerequisites

None. This course assumes no prior programming experience.

Contents

About this course

Food security is the sector with the most standardised indicators on this platform and the widest gap between computing one and understanding it. Every instrument here has a published definition, a scoring rule and a threshold, and almost every one of them is routinely applied with one step missing.

The course runs on three files. The **household food security survey** carries the raw components of the Food Consumption Score, the Household Hunger Scale and the reduced Coping Strategy Index, so all three have to be built rather than read off. It gains a **Livelihood Coping Strategies module** here, keyed on household, because the survey shipped without it and a module is a second file rather than a rewritten one. The **market price series** is new: twenty-four months, twelve markets, seven commodities and a wage.

Three results shape the course, all computed from those files.

Four indicators on the same 1,955 households give four different answers. Poor food consumption catches 7.4% of them; moderate or severe hunger 42.9%; a high coping index 51.0%; crisis or emergency livelihood strategies 48.2%. **76.3% are flagged by at least one and 3.4% by all four.** That gap is the reason the IPC convenes a working group rather than running a formula, and it is the argument this course is built around.

Seasonality is bigger than most programme effects. Median maize runs 75 gourdes per kilogram at the December harvest and 130 at the July lean-season peak. A comparison of two survey rounds taken at different points in that cycle measures the calendar.

A goat's price fell 18% and its purchasing power fell 62%. In 2023 a goat bought 69.4 kg of maize; at the 2024 lean-season peak it bought 26.7. Neither the goat series nor the maize series shows that on its own.

Both Python and R throughout. You should have done module 3 — this course assumes you can defend a denominator, put an interval on a proportion, and write an indicator reference sheet before you write the analysis.

Part I

Compute it before you believe it

CHAPTER 1

One score, two thresholds, eight times the answer

Lesson 1 of 8 · 135 min

On the 21/35 thresholds 0.9% of these households have poor food consumption. On the 28/42 thresholds 7.3% do. Both are correct applications of a published standard, and a report that does not say which it used is not.

1.1 The score, from its parts

The Food Consumption Score asks how many of the last seven days a household ate from each of eight food groups, and weights them by nutritional density.

```python
import pandas as pd

survey = pd.read_csv("food-security-survey-2024.v1.csv")

WEIGHTS = {
    "fcs_cereals_tubers": 2.0,
    "fcs_pulses": 3.0,
    "fcs_vegetables": 1.0,
    "fcs_fruit": 1.0,
    "fcs_meat_fish_eggs": 4.0,
    "fcs_dairy": 4.0,
    "fcs_oils_fats": 0.5,
    "fcs_sugar": 0.5,
}
```

```r
library(dplyr)

weights <- c(fcs_cereals_tubers = 2, fcs_pulses = 3, fcs_vegetables = 1,
             fcs_fruit = 1, fcs_meat_fish_eggs = 4, fcs_dairy = 4,
             fcs_oils_fats = 0.5, fcs_sugar = 0.5)
```

Meat and dairy carry four points a day and sugar carries half, because the score is a proxy for dietary quality rather than for quantity. A household eating cereals and oil every day scores 17.5; one eating meat twice a week scores 8 for those two days alone.

Two things have to happen before the multiplication, and both are skipped routinely.

1.2 Range-check first

```python
groups = list(WEIGHTS)
print(survey[groups].max())
print(f"values above 7: {(survey[groups] > 7).sum().sum()}")
```

```r
survey |> summarise(across(all_of(names(weights)), max, .names = "{.col}"))
```

Twenty-three cells hold a value above seven days, against a seven-day recall. They are impossible, and a score computed without checking inherits them: the highest FCS in this file is 114.5 uncorrected against 105.5 with the values capped.

The correction is a judgement. Capping at seven assumes a keying error in the last digit; blanking assumes the answer is unknown. **Say which you did**, and notice that capping is the more conservative choice here because it keeps the household in the denominator.

1.3 A blank is not a zero

```python
blanks = survey[groups].isna().sum()
print(blanks[blanks > 0])
print(f"households with any blank: {survey[groups].isna().any(axis=1).sum()}")
```

```r
survey |> summarise(across(all_of(names(weights)), ~ sum(is.na(.x))))
```

127 cells across dairy, fruit and meat are blank, in 123 households. Those are the three highest-weighted groups after pulses, which is not a coincidence — they are the questions an enumerator skips when the answer is obviously none and the form does not force a response.

Treating a blank as zero days scores a household as eating less than it did. `sum()` in pandas does exactly that by default, and `sum(na.rm = TRUE)` in R does it on request.

```python
# The silent one: NaN treated as zero.
naive = sum(survey[group].fillna(0) * weight for group, weight in WEIGHTS.items())

# The honest one: a household missing any group has no score.
capped = survey[groups].clip(upper=7)
fcs = sum(capped[group] * weight for group, weight in WEIGHTS.items())
fcs = fcs.where(capped.notna().all(axis=1))

print(f"analysable: {fcs.notna().sum()} of {len(survey)}")
```

```r
survey |>
  mutate(across(all_of(names(weights)), ~ pmin(.x, 7))) |>
  rowwise() |>
  mutate(fcs = if (anyNA(c_across(all_of(names(weights))))) NA_real_
               else sum(c_across(all_of(names(weights))) * weights))
```

1,989 households of 2,112 have a complete consumption module. Report that number. A prevalence computed on 1,989 and printed beside a demographic figure computed on 2,112 invites a subtraction that does not mean anything.

1.4 The two threshold sets

```python
for poor, borderline in [(21, 35), (28, 42)]:
    bands = pd.cut(fcs, [-1, poor, borderline, 200],
                   labels=["poor", "borderline", "acceptable"])
    print(f"{poor}/{borderline}: ",
          (bands.value_counts(normalize=True) * 100).round(1).to_dict())
```

```r
classify <- function(x, poor, borderline) {
  cut(x, c(-1, poor, borderline, Inf), labels = c("poor", "borderline", "acceptable"))
}
```

Thresholds	Poor	Borderline	Acceptable
21 / 35	**0.9%**	22.7%	76.4%
28 / 42	**7.3%**	38.5%	54.1%

The poor-consumption headline is eight times larger on one set than the other, and both are the published standard. The 28/42 set exists for contexts where oil and sugar are consumed near-universally: those two groups add 7 points to almost every household, which shifts the whole distribution right and makes the lower cut-offs too generous.

Median FCS here is 43.5, and oil is eaten a mean 4.7 days a week and sugar 3.8 — frequent enough that the two groups add about 4 points to a typical household before any nutrient-dense food is counted.

```python
staples = survey[["fcs_oils_fats", "fcs_sugar"]].mean()
print(f"oil eaten {staples['fcs_oils_fats']:.1f} days a week on average")
print(f"sugar eaten {staples['fcs_sugar']:.1f} days")
print("→ 28/42 is the defensible set here; say so in the methodology note")
```

```r
survey |> summarise(oil = mean(fcs_oils_fats), sugar = mean(fcs_sugar))
```

Choose on the evidence, state the choice, and never change it between rounds. A programme that reports 0.9% one year and 7.3% the next because someone switched threshold sets has reported a fourfold deterioration that did not happen.

1.5 What the score is not

Three things the FCS is regularly asked to do and cannot.

It is not a measure of quantity. A household eating small portions of eight food groups scores well. The score is about dietary diversity and frequency, and Sphere's kilocalorie standards are a different instrument.

It is not comparable across contexts with different diets. The weights are fixed globally and the food groups are not eaten in the same proportions everywhere, which is exactly why two threshold sets exist.

It is not an IPC phase. It is one outcome indicator feeding one of several evidence rows, and the last unit of this course is about the difference.

1.6 Report it with its rules

```
Food consumption, lean season 2024

  Analysable                  1,989 of 2,112 households (94.2%)
  Poor consumption               7.3%   146 households
  Borderline                    38.5%   766
  Acceptable                    54.1%  1,077

  FCS computed on the 28/42 threshold set: oil is eaten a mean 4.7 days
  a week and sugar 3.8, so the standard 21/35 cut-offs would classify
```

```
0.9% as poor and understate the caseload.
23 impossible values capped at 7 days; 123 households excluded for an
incomplete consumption module.
```

The thresholds named, the exclusions counted, and the reason for the choice in one sentence. **That paragraph is what makes the 7.3% auditable**, and it is three lines longer than the version most reports carry.

1.7 What comes next

Food consumption is what a household ate. The next lesson is what it went without and what it did to avoid going without — two more instruments on the same households, each with an exclusion rule of its own.

CHAPTER 2

The thirty-seven households zero-filling would misclassify

Lesson 2 of 8 · 135 min

Zero-filling an incomplete Household Hunger Scale moves the prevalence by three tenths of a point and can misclassify every one of the thirty-seven households it touches. Which of those matters depends on whether your output is a percentage or a list.

2.1 Two more instruments, two more rules

The Household Hunger Scale and the reduced Coping Strategy Index sit beside the Food Consumption Score in almost every food security survey, and each carries a scoring rule that a naive `sum()` breaks in a different way.

Instrument	What it asks	The rule that gets broken
HHS	Three questions on going without food, scored 0–2 each	Valid only when **all three** are answered
rCSI	Five behaviours, days in the last seven, weighted	The weights are not all 1, and the fifth is 3

2.2 The Household Hunger Scale

Three questions, deliberately few, deliberately severe: was there no food of any kind in the house, did anyone go to sleep hungry, did anyone go a whole day and night without eating. Each is scored 0 for never, 1 for rarely or sometimes, 2 for often.

```python
import pandas as pd

survey = pd.read_csv("food-security-survey-2024.v1.csv")
HHS = ["hhs_no_food_in_house", "hhs_sleep_hungry",
       "hhs_day_and_night_without_eating"]

complete = survey[HHS].notna().all(axis=1)
score = survey.loc[complete, HHS].sum(axis=1)

bands = pd.cut(score, [-1, 1, 3, 6],
               labels=["little or none", "moderate", "severe"])
print(f"complete: {complete.sum()} of {len(survey)}")
print((bands.value_counts(normalize=True) * 100).round(1))
```

```r
library(dplyr)

hhs <- c("hhs_no_food_in_house", "hhs_sleep_hungry",
         "hhs_day_and_night_without_eating")

survey |>
  filter(if_all(all_of(hhs), ~ !is.na(.x))) |>
  mutate(score = rowSums(across(all_of(hhs))),
         band = cut(score, c(-1, 1, 3, 6),
                    labels = c("little or none", "moderate", "severe"))) |>
  count(band) |> mutate(share = n / sum(n))
```

Band	Households	Share
Little or no hunger (0–1)	1,186	57.2%
Moderate (2–3)	564	27.2%
Severe (4–6)	325	**15.7%**

2,075 of 2,112 households answered all three. The thirty-seven that did not are the interesting ones.

2.3 What zero-filling actually costs

The instinct is that dropping the incomplete responses loses data, so fill the gap with zero and keep them. Try it and measure.

```python
zero_filled = survey[HHS].fillna(0).sum(axis=1)
naive = pd.cut(zero_filled, [-1, 1, 3, 6],
               labels=["little or none", "moderate", "severe"])
print((naive.value_counts(normalize=True) * 100).round(1))
```

```r
survey |>
  mutate(score = rowSums(across(all_of(hhs)), na.rm = TRUE)) |>
  count(band = cut(score, c(-1, 1, 3, 6)))
```

	Complete cases	Zero-filled
Little or none	57.2%	57.5%
Moderate	27.2%	26.9%
Severe	15.7%	15.5%

Three tenths of a percentage point. On a prevalence, zero-filling is practically harmless here, and if you stop at the table you will conclude it does not matter.

Now look at the households rather than the percentage.

```python
partial = survey.loc[~complete, HHS]
print(partial.sum(axis=1).value_counts().sort_index())
```

```r
survey |> filter(if_any(all_of(hhs), is.na)) |>
  mutate(observed = rowSums(across(all_of(hhs)), na.rm = TRUE)) |> count(observed)
```

Twenty-nine of the thirty-seven score 0 or 1 on the questions they did answer, so zero-filling classifies them as **little or no hunger**. The unanswered question is worth up to 2 points. A household sitting at 1 with one question missing could be at 1 or at 3 — little hunger or moderate — and nothing in the data decides it.

So the cost of zero-filling depends entirely on what the analysis produces.

- If the output is a **prevalence**, the damage is three tenths of a point and you should say you zero-filled and move on.
- If the output is a **targeting list**, you have just told thirty-seven households they are food secure on the strength of a question nobody asked.

Name the output before choosing the rule. This is the same decision as the CMAM cure-rate denominator and the water point functionality rate: the defensible choice is not a property of the data, it is a property of the decision the number feeds.

2.4 The reduced Coping Strategy Index

Five behaviours, days in the last seven, and the weights are the whole point.

```
RCSI = {
    "rcsi_less_preferred_food": 1,
    "rcsi_borrowed_food": 2,
    "rcsi_limit_portion_size": 1,
    "rcsi_restrict_adult_consumption": 3,
    "rcsi_reduce_meal_numbers": 1,
}
rcsi = sum(survey[column] * weight for column, weight in RCSI.items())
print(f"median {rcsi.median():.0f}, mean {rcsi.mean():.1f}, max {rcsi.max():.0f}")
print(f"rCSI 19 or above: {(rcsi >= 19).mean():.1%}")
```

```
rcsi_weights <- c(rcsi_less_preferred_food = 1, rcsi_borrowed_food = 2,
                  rcsi_limit_portion_size = 1,
                  rcsi_restrict_adult_consumption = 3,
                  rcsi_reduce_meal_numbers = 1)
```

Median 19, mean 19.5, and 50.9% at or above 19. The weight of 3 on *restricting adult consumption so children can eat* is not arbitrary — it is the behaviour most predictive of deterioration, and an unweighted sum would bury it among four behaviours weighted 1.

Two things about the rCSI that are routinely got wrong.

It has no universal threshold. Unlike the FCS, the rCSI's cut-offs are context-specific and usually set against the distribution in the same population. A report quoting "rCSI above 19" as though 19 were a standard has borrowed a number from another country's analysis.

It goes up before consumption goes down, and it can also go *down* in a crisis, when a household has exhausted the strategies. A falling rCSI beside a falling FCS is worse news than a rising one.

2.5 Three instruments, three analysable samples

```
denominators = pd.DataFrame({
    "instrument": ["Food Consumption Score", "Household Hunger Scale",
                   "reduced Coping Strategy Index"],
    "analysable": [1989, 2075, 2112],
    "excluded": [123, 37, 0],
    "rule": ["all eight groups answered", "all three questions answered",
             "complete in this file"],
})
print(denominators)
```

```
# Print this before the results, every time.
```

Three numbers on three denominators, and none of them is 2,112. The difference is small enough to be invisible in a table and large enough to matter when someone subtracts two of your percentages.

2.6 What comes next

Consumption and hunger say what a household is experiencing. Neither says what it is doing about it, and the next lesson is the module that does — where the scoring rule is not a sum at all.

Part II

Coping is a sequence, not a score

CHAPTER 3

A household is classified by its worst strategy

Lesson 3 of 8 · 135 min

Ten strategies, three severity phases, and the rule is the maximum rather than the sum. 18.2% of these households used an emergency strategy, and one district's answer to "not applicable" makes its prevalence figures wrong while leaving its classification intact.

3.1 The module, and why it is not another index

The Livelihood Coping Strategies module asks whether a household used any of ten strategies in the last thirty days. Each strategy is **pre-classified** into a severity phase before any data is collected, and the household is then classified by the most severe strategy it used.

Phase	Strategies	What they h
Stress	Sold household assets, spent savings, borrowed money, sold more animals than usual	Reduce the ab
Crisis	Sold productive assets, withdrew children from school, cut health spending	Reduce future
Emergency	Sold house or land, begged, sold last female breeding animals	Reduce future

The classification is a maximum, not a sum, and not a count. A household that sold its last breeding animals once is worse off than one that borrowed money four times, and any arithmetic adding the two together says the opposite.

```
import pandas as pd

coping = pd.read_csv("livelihood-coping-2024.v1.csv")

PHASE = {
    "sold_household_assets": "stress",
    "spent_savings": "stress",
    "borrowed_money": "stress",
    "sold_more_animals_than_usual": "stress",
    "sold_productive_assets": "crisis",
    "withdrew_children_from_school": "crisis",
    "reduced_health_expenditure": "crisis",
    "sold_house_or_land": "emergency",
    "begged": "emergency",
    "sold_last_female_animals": "emergency",
}
RANK = {"none": 0, "stress": 1, "crisis": 2, "emergency": 3}

used = coping[list(PHASE)].eq("yes")
severity = pd.Series("none", index=coping.index)
for strategy, phase in PHASE.items():
    higher = used[strategy] & (severity.map(RANK) < RANK[phase])
    severity[higher] = phase

print((severity.value_counts(normalize=True) * 100).round(1))
```

```
library(dplyr)
```

```
phase <- c(sold_household_assets = "stress", spent_savings = "stress",
           borrowed_money = "stress", sold_more_animals_than_usual = "stress",
           sold_productive_assets = "crisis",
           withdrew_children_from_school = "crisis",
           reduced_health_expenditure = "crisis",
           sold_house_or_land = "emergency", begged = "emergency",
           sold_last_female_animals = "emergency")
rank <- c(none = 0, stress = 1, crisis = 2, emergency = 3)
```

Classification	Households	Share
None	339	16.0%
Stress	754	35.5%
Crisis	642	30.3%
Emergency	386	**18.2%**

48.5% used a crisis or emergency strategy. That single figure is what the IPC evidence table takes from this module, and it is a very different picture from the 7.3% with poor food consumption.

3.2 Not applicable is not no

A household with no animals cannot sell its last breeding female. The module records that as `not-applicable`, and it is a third answer rather than a flavour of no.

```
answers = coping["sold_last_female_animals"].value_counts()
print(answers)

applicable = coping["sold_last_female_animals"].isin(["yes", "no"])
print(f"prevalence among applicable: "
      f"{coping.loc[applicable, 'sold_last_female_animals'].eq('yes').mean():.1%}")
print(f"prevalence if n/a counted as no: "
      f"{coping['sold_last_female_animals'].eq('yes').mean():.1%}")
```

```
coping |> count(sold_last_female_animals)

coping |>
  filter(sold_last_female_animals %in% c("yes", "no")) |>
  summarise(prevalence = mean(sold_last_female_animals == "yes"), n = n())
```

Strategy	Among households it applies to	If n/a counted as no
Sold more animals than usual	**37.4%**	27.5%
Sold productive assets	19.8%	17.3%
Withdrew children from school	19.7%	16.5%
Sold last female animals	8.6%	5.8%

Ten points of difference on the first row. The denominator for a per-strategy prevalence is the households the strategy is *available to*, and counting the rest as non-users understates every one of them.

The household-level classification survives this and the per-strategy prevalence does not, because a maximum over `yes` values does not care whether the non-yes values are `no` or `not-applicable`. That asymmetry is worth knowing: it is why the same file can support a correct classification and a wrong prevalence table on the same page.

3.3 The district where the distinction was lost

```python
by_district = coping.groupby("district")["sold_last_female_animals"].agg(
    yes=lambda s: (s == "yes").sum(),
    applicable=lambda s: s.isin(["yes", "no"]).sum(),
    not_applicable=lambda s: (s == "not-applicable").sum(),
)
by_district["correct"] = by_district["yes"] / by_district["applicable"]
by_district["naive"] = by_district["yes"] / coping.groupby("district").size()
print((by_district[["correct", "naive"]] * 100).round(1))
```

```r
coping |>
  summarise(yes = sum(sold_last_female_animals == "yes"),
            applicable = sum(sold_last_female_animals %in% c("yes", "no")),
            n = n(), .by = district) |>
  mutate(correct = yes / applicable, naive = yes / n)
```

District	Correct denominator	Every household
Nord-Ouest	**12.3%**	6.8%
Artibonite	9.7%	5.3%
Sud	7.5%	4.5%
Centre	6.5%	6.4%

Centre's two columns are almost identical, and every other district's are not. The reason is in the raw data: **Centre has no `not-applicable` cells at all**, because its enumerators recorded them as `no` throughout.

```python
print(coping.groupby("district")[list(PHASE)]
      .apply(lambda block: (block == "not-applicable").sum().sum()))
```

```r
coping |> summarise(across(all_of(names(phase)),
                    ~ sum(.x == "not-applicable")), .by = district)
```

On the correct denominator Nord-Ouest is nearly twice Centre. On the wrong one they are the same. A district comparison built from the naive column would conclude that livestock distress selling is uniform across the response area, and would be reading one team's data entry habit.

3.4 The partial module

Fifty-eight households have the three emergency questions blank.

```python
emergency = [s for s, phase in PHASE.items() if phase == "emergency"]
partial = coping[emergency].eq("").all(axis=1) | coping[emergency].isna().all(axis=1)
print(f"{partial.sum()} households cannot be classified above crisis")
```

```r
coping |> filter(if_all(c(sold_house_or_land, begged, sold_last_female_animals),
                        ~ is.na(.x))) |> nrow()
```

A maximum computed over what is present classifies them as crisis at worst. **So 18.2% in emergency is a lower bound**, and the honest report says so rather than presenting it as the estimate.

The alternative — dropping them — makes the emergency share an unbiased estimate of a slightly different population. Both are defensible; picking silently is not.

3.5 Why coping is measured separately from consumption

The instruments do not overlap the way people expect.

```
survey = pd.read_csv("food-security-survey-2024.v1.csv")
merged = survey.merge(coping, on="household_id", how="left", indicator=True)
print(merged["_merge"].value_counts())
```

```
survey |> left_join(coping, by = "household_id") |>
  summarise(matched = sum(!is.na(district.y)), n = n())
```

Households resort to coping strategies **before** their food consumption falls — that is the point of coping. A household selling assets to keep eating has intact consumption and a collapsing asset base, and an analysis reading only the FCS records it as food secure right up until the assets run out.

That is the argument for measuring both, and the next lesson is what happens when you do.

Two things to check on the join. Nine households appear in the module and not in the survey, because the module was administered from a separate listing — an inner join drops them without a word. And the survey's twelve duplicate households appear once in the module, so the join is not one-to-one.

3.6 What comes next

Four instruments now exist on these households: consumption, hunger, coping behaviour and livelihood strategies. The next lesson asks which of them agree about who is food insecure, and the answer is most of the reason the IPC exists.

CHAPTER 4

76.3% flagged by one, 3.4% by all four

Lesson 4 of 8 · 135 min

Four instruments on the same 1,955 households give prevalences of 7.4%, 42.9%, 51.0% and 48.2%. The union is 76.3% and the intersection is 3.4%. Every argument in the rest of this course starts from that gap.

4.1 Four answers to one question

Every instrument in the last two lessons claims to identify food-insecure households. Put them on the same households and count.

```python
import pandas as pd

survey = pd.read_csv("food-security-survey-2024.v1.csv")
coping = pd.read_csv("livelihood-coping-2024.v1.csv")
data = survey.merge(coping, on="household_id", how="inner", suffixes=("", "_lcs"))

flags = pd.DataFrame({
    "poor_consumption": fcs <= 28,          # FCS on the 28/42 set
    "hunger": hhs >= 2,                     # HHS moderate or severe
    "high_coping": rcsi >= 19,              # rCSI at or above the median
    "crisis_strategies": severity.isin(["crisis", "emergency"]),
})
flags = flags.dropna()
print(f"analysable on all four: {len(flags)} of {len(survey)}")
print((flags.mean() * 100).round(1))
```

```r
library(dplyr)

flags |> summarise(across(everything(), ~ mean(.x)), n = n())
```

Instrument	Households flagged	Share
Poor food consumption	144	**7.4%**
Moderate or severe hunger	839	42.9%
High coping index	997	**51.0%**
Crisis or emergency strategies	943	48.2%

A sevenfold range on the same 1,955 households. These are not four estimates of one quantity with sampling error between them. They are four different quantities, each correctly measured.

4.2 The overlap

```python
count = flags.sum(axis=1)
print(count.value_counts().sort_index())
print(f"any: {(count >= 1).mean():.1%}   all four: {(count == 4).mean():.1%}")
```

```
flags |> mutate(n_flags = rowSums(across(everything()))) |> count(n_flags)
```

Flagged by	Households	Share
None of the four	464	23.7%
One	591	30.2%
Two	434	22.2%
Three	400	20.5%
All four	66	**3.4%**

76.3% of households are flagged by at least one instrument and 3.4% by all four. Both numbers are defensible answers to "how many households are food insecure", and they are twenty-two times apart.

This is the single most important table in the course. Everything that follows — the evidence table, the convergence rule, the working group — exists because this table looks the way it does.

4.3 Why they disagree, in order of importance

They measure different moments in the same process. Coping runs ahead of consumption. A household sells its goats in June and is still eating in July, so the LCS flags it and the FCS does not. Reading only consumption sees the crisis about four months late.

They have different sensitivities. The FCS at 7.4% is the least sensitive instrument here by a wide margin, because its threshold was written to identify severe dietary deprivation rather than stress. That is not a defect — it is what makes a poor FCS a strong signal when it appears.

Some ask about behaviour and some about experience. rCSI and LCS ask what a household *did*; HHS asks what it *went through*. A household with resources copes without hunger; a household without resources goes hungry without coping, because there is nothing left to sell.

```
nothing_left = flags["hunger"] & ~flags["crisis_strategies"]
print(f"hungry, no crisis strategies: {nothing_left.sum()} households")
```

```
flags |> filter(hunger, !crisis_strategies) |> nrow()
```

282 households are hungry and using no crisis or emergency strategy. That is the group a coping-only analysis loses, and it is the group furthest along.

4.4 What not to do about it

Three repairs that all look principled.

Average them into a composite score. There is no defensible weighting, the instruments are on different scales, and a household at 7.4% on one and 51% on another is not "at 29%". Composite food security indices exist and every one of them buries the disagreement rather than resolving it.

Pick the one that gives the number you expected. This happens more often than the previous one and is harder to see, because each individual choice is defensible in isolation. **Choose the instrument before you compute it**, on the question being asked, and write the choice down.

Take the intersection to be safe. 3.4% is the most conservative estimate and it excludes the 400 households flagged by three instruments out of four. Being conservative about a caseload is not caution; it is a decision to serve fewer people, and it should be made as one.

4.5 What to do instead

Report all four, with what each is for.

```
Food security indicators, lean season 2024, 1,955 households

  Poor food consumption (FCS ≤28)          7.4%   severe dietary deprivation
  Moderate or severe hunger (HHS ≥2)      42.9%   experienced deprivation
  High coping (rCSI ≥19)                  51.0%   consumption-based coping
  Crisis or emergency strategies (LCS)    48.2%   asset depletion

  Flagged by at least one                 76.3%
  Flagged by all four                      3.4%

  These are four different quantities, not four estimates of one. The
  spread between them is the finding: coping and asset depletion are
  widespread while severe dietary deprivation is not yet, which is the
  profile of a population early in a deterioration rather than late in one.
```

The last sentence is the analysis. The pattern across the four — high coping, high asset depletion, low severe consumption deficit — is a *diagnosis*, and it is only available because the four disagree. A single composite would have produced one number and no diagnosis at all.

4.6 The rule this course is built on

Convergence of evidence means agreement across independent instruments raises confidence, and disagreement is information rather than error.

Where three instruments agree and one does not, ask what the fourth measures that the others do not, before deciding it is wrong. In this population the odd one out is the FCS, and what it measures that the others do not is *severity* — so the disagreement says the deterioration has not yet reached diets, which is a finding about timing and an argument for acting now.

4.7 What comes next

Everything so far is what households reported about themselves. The next unit is the market they buy from — an independent line of evidence that does not depend on anybody's recall, and that turns out to explain the timing of all four indicators.

Part III

The market is an outcome too

CHAPTER 5

The calendar is bigger than the programme

Lesson 5 of 8 · 135 min

Median maize is 75 gourdes at the December harvest and 130 at the July peak. Compare two survey rounds taken at different points in that cycle and you have measured the calendar — and one district's panel changed composition exactly when it mattered.

5.1 An independent line of evidence

Everything in the first two units is what households said about themselves. A price series is not: nobody is recalling anything, and a market that has doubled its maize price has done so whether or not a survey was in the field.

```python
import pandas as pd

prices = pd.read_csv("market-prices-2024.v1.csv")
print(f"{len(prices):,} observations")
print(f"{prices['market_id'].nunique()} markets, "
      f"{prices['commodity'].nunique()} series, "
      f"{prices['period'].nunique()} months")
```

```r
library(dplyr)

prices |> summarise(n = n(), markets = n_distinct(market_id),
                    series = n_distinct(commodity), months = n_distinct(period))
```

2,273 observations, twelve markets, eight series, twenty-four months. **Two years is the minimum useful length**, because a single year cannot distinguish a seasonal peak from a deterioration.

5.2 Clean three things before plotting anything

```python
prices["price_htg"] = pd.to_numeric(prices["price_htg"])

dollars = prices["price_htg"] < 10
marmite = (prices["commodity"] == "maize") & (prices["market_id"] == "MK005")
thin = prices["trader_quotes"] == 1

print(f"dollar entries: {dollars.sum()}")
print(f"marmite market maize rows: {marmite.sum()}")
print(f"single-quote observations: {thin.sum()}")
```

```r
prices |> summarise(
  dollars = sum(price_htg < 10),
  thin = sum(trader_quotes == 1)
)
```

Eight prices are in dollars with the currency column still saying HTG. They read as values under 10 and an outlier filter would delete them, when multiplying by about 132 recovers them.

One market records maize by the marmite, a volume measure of roughly 2.7 kg, and files it in a column labelled per kilogram. Its median maize price is 263 gourdes against about 100 everywhere else.

```python
by_market = prices[prices["commodity"] == "maize"].groupby("market_id")["price_htg"].
    median()
print(by_market.round(1).sort_values())
```

```
prices |> filter(commodity == "maize") |>
  summarise(median = median(price_htg), .by = market_id) |> arrange(median)
```

The unit error is invisible in one market and obvious across twelve. That is the general rule for a unit slip: it never looks wrong on its own row, and it always looks wrong beside its peers.

Thirteen observations rest on a single trader quote. They are not errors. A median of one is a quote rather than a price, and the decision is whether to exclude them or weight them down — either is defensible, silence is not.

5.3 The seasonal cycle

```python
clean = prices[~dollars & ~marmite]
maize = clean[clean["commodity"] == "maize"]
series = maize.groupby("period")["price_htg"].median()
print(series.round(0))
```

```
prices |>
  filter(commodity == "maize", price_htg >= 10, market_id != "MK005") |>
  summarise(median = median(price_htg), .by = period) |> arrange(period)
```

Month	2023	2024
January	66	70
April	97	110
July (lean peak)	**130**	**159**
October	92	104
December (harvest)	**75**	**81**

Maize costs 74% more at the July peak than at the December harvest in 2023, and 96% more in 2024. That is the size of the seasonal effect, and it is larger than almost any programme effect anyone will ask you to detect.

The consequence for survey design is direct. A baseline in December and an endline in July would show a catastrophic deterioration produced entirely by the calendar. Compare like with like: **July against July**.

```python
peaks = series.loc[["2023-07", "2024-07"]]
print(f"lean peak 2023: {peaks['2023-07']:.1f}")
print(f"lean peak 2024: {peaks['2024-07']:.1f}")
print(f"year on year: {peaks['2024-07'] / peaks['2023-07'] - 1:+.1%}")
```

```
# Same point in the season, one year apart. Everything else is the calendar.
```

130.1 against 159.3 — a 22% rise at the same point in the season. *That* is the deterioration, and it is a quarter of the size of the seasonal swing that would have been mistaken for it.

5.4 The panel that changed composition

```
coverage = maize.pivot_table(index="period", columns="market_id",
                             values="price_htg", aggfunc="size")
print(coverage.isna().sum(axis=1)[lambda s: s > 0])
```

```
prices |> filter(commodity == "maize") |> count(period, market_id) |>
  count(period) |> filter(n < 11)
```

MK001 stops reporting from June to September 2024, when its road is cut. The rows are absent rather than zero, so nothing in the file announces it.

It is also the most expensive market in Nord-Ouest, and Nord-Ouest has only three markets — so losing one moves the district mean by a third of its deviation rather than a twelfth.

```
nord = maize[maize["district"] == "Nord-Ouest"]
reported = nord.groupby("period")["price_htg"].mean()

balanced = nord[nord["market_id"] != "MK001"].groupby("period")["price_htg"].mean()
print(pd.DataFrame({"reported": reported, "balanced": balanced}).round(1)
      .loc["2024-04":"2024-10"])
```

```
prices |>
  filter(commodity == "maize", district == "Nord-Ouest") |>
  summarise(reported = mean(price_htg), .by = period)
```

Month	Whoever reported	The two that reported every month
May 2024	147.0	123.2
June 2024	172.1	172.1
May to June	**+17%**	**+40%**

The reported series understates the lean-season rise by more than half, and every number in it is a correct mean of the markets that reported. The composition changed underneath the series and the series did not say so.

Use a balanced panel — only the markets present in every month you are comparing — or chain month-on-month changes computed within markets. Both are more work than a `groupby` and both are the difference between a 17% rise and a 40% one.

5.5 Report the series with its panel

```
Maize price, gourdes per kg, median across markets

  2023 harvest (Dec)        75      2024 harvest (Dec)       81
```

```
2023 lean peak (Jul)     130      2024 lean peak (Jul)    159
Seasonal swing          +74%      Seasonal swing         +96%
Year on year at the lean peak: +22%

Balanced panel of 10 markets throughout. MK001 (Nord-Ouest) did not
report June to September 2024 and is excluded from all periods; MK005
records maize by the marmite and is excluded from maize.
8 dollar-denominated entries converted at 132 HTG; 13 single-quote
observations retained and flagged.
```

The panel stated, the exclusions counted, and the year-on-year comparison made at the same point in the season. **A price table without its panel is the same defect as a coverage figure without its reporting rate**, and this course has now met it three times.

5.6 What comes next

A price says what food cost. It does not say whether anybody could afford it, and the next lesson is the ratio that does — where a goat whose price fell 18% lost 62% of its purchasing power.

CHAPTER 6

The goat that lost 62% of its value while its price fell 18%

Lesson 6 of 8 · 135 min

A day's casual labour bought 4.82 kg of maize at the 2023 harvest and 2.48 kg at the 2024 lean peak. A goat bought 69.4 kg and then 26.7. Neither the wage series nor the goat series shows that, because it is a ratio.

6.1 A price is a number about a market, not about a household

Maize at 159 gourdes a kilogram tells you nothing about whether anyone can buy it. That question needs the other side of the transaction: what the household has to sell, or to earn, in order to buy.

Terms of trade are that ratio, expressed in the units the household thinks in.

```python
import pandas as pd

prices = pd.read_csv("market-prices-2024.v1.csv")
clean = prices[(prices["price_htg"] >= 10) &
               ~((prices["commodity"] == "maize") &
                 (prices["market_id"] == "MK005"))]

series = clean.pivot_table(index="period", columns="commodity",
                           values="price_htg", aggfunc="median")

series["wage_to_maize"] = series["daily-wage-casual-labour"] / series["maize"]
series["goat_to_maize"] = series["goat"] / series["maize"]
print(series[["maize", "daily-wage-casual-labour", "goat",
              "wage_to_maize", "goat_to_maize"]].round(2))
```

```r
library(dplyr)

prices |>
  filter(price_htg >= 10, !(commodity == "maize" & market_id == "MK005")) |>
  summarise(price = median(price_htg), .by = c(period, commodity)) |>
  tidyr::pivot_wider(names_from = commodity, values_from = price) |>
  mutate(wage_to_maize = `daily-wage-casual-labour` / maize,
         goat_to_maize = goat / maize)
```

Month	Maize	Wage	Goat	Wage → kg maize	Goat → kg maize
Dec 2023	74.9	361.3	5,195	**4.82**	**69.4**
Jul 2023	130.1	368.9	4,332	2.84	33.3
Dec 2024	81.4	388.5	5,900	4.77	72.5
Jul 2024	159.3	395.6	4,254	**2.48**	**26.7**

6.2 What the wage series alone would have told you

The daily wage rose 10% over two years, from 361 to 396 gourdes. Read on its own, that is stability, perhaps mild improvement.

A day's work bought 4.82 kg of maize at the 2023 harvest and 2.48 kg at the 2024 lean peak — a 49% fall in purchasing power while the wage went up.

```
wage = series["wage_to_maize"]
print(f"wage rose {series['daily-wage-casual-labour']['2024-07'] / series['daily-wage-
    casual-labour']['2023-12'] - 1:+.1%}")
print(f"purchasing power fell {wage['2024-07'] / wage['2023-12'] - 1:+.1%}")


# One number goes up, the other goes down, and they are the same households.
```

This is the single most useful ratio in a food security analysis of a labour-dependent population, and it takes two columns and a division.

6.3 What the goat series alone would have told you

Worse, because the goat price moves the wrong way.

Livestock prices *fall* during the lean season: everyone is selling at once, the animals are in poor condition, and buyers know both. So a pastoral or agro-pastoral household faces rising cereal prices and falling livestock prices at the same moment.

```
goat = series["goat_to_maize"]
print(f"goat price change: "
      f"{series['goat']['2024-07'] / series['goat']['2023-12'] - 1:+.1%}")
print(f"goat terms of trade: {goat['2024-07'] / goat['2023-12'] - 1:+.1%}")


# -18% and -62%. The second is the one the household experiences.
```

The goat price fell 18%. The goat's purchasing power fell 62%. A herder who sold one goat for 69 kg of maize in December 2023 needed to sell two and a half goats for the same maize at the 2024 peak — which is exactly the mechanism by which a herd disappears in one bad year.

Neither series shows that. The maize series shows a price rise; the goat series shows a modest price fall; only the ratio shows a collapse.

6.4 The three ratios worth computing

Ratio	Whose access it describes	Read it against
Wage to staple	Casual labourers, 26.7% of these households	Kilograms per day w
Livestock to staple	Livestock-dependent households, 7.8% here	Kilograms per anima
Cash-crop to staple	Farmers selling one crop and buying another, 29.5% subsistence here	Kilograms per kilogr

```
livelihoods = pd.read_csv("food-security-survey-2024.v1.csv")["main_livelihood"]
print(livelihoods.value_counts(normalize=True).round(3))
```

```
survey |> count(main_livelihood) |> mutate(share = n / sum(n)) |> arrange(-share)
```

Compute the ratio that matches the livelihood you are analysing. A wage-to-cereal figure describes nothing about a household living on remittances, and the survey's `main_livelihood` column is what tells you which ratio belongs to which group and how many people are in it.

6.5 Two warnings

Terms of trade are not a welfare measure. They say how much food a unit of income or of assets buys. They say nothing about how many units the household has, and a household with no goats is unaffected by a goat-to-maize collapse and may be worse off than one that has them.

The denominator has to be the staple people actually eat. Rice is imported here and moves with the exchange rate rather than the harvest; maize is local and moves with the season. A terms-of-trade series built on rice would show a much flatter picture and would be describing a different household.

```
series["wage_to_rice"] = (series["daily-wage-casual-labour"]
                          / series["rice-imported"])
print(series[["wage_to_maize", "wage_to_rice"]].loc[
    ["2023-12", "2023-07", "2024-12", "2024-07"]].round(2))
```

```
# Two staples, two stories. Say which one the population eats.
```

6.6 Report the ratio, not just the prices

```
Terms of trade, median across a balanced panel of 10 markets

                          Dec 2023   Jul 2024   Change
  Maize, HTG/kg               74.9      159.3    +113%
  Casual wage, HTG/day       361.3      395.6     +10%
  Goat, HTG/head           5,195.1    4,254.4     -18%

  Wage buys, kg maize          4.82       2.48     -49%
  Goat buys, kg maize          69.4       26.7     -62%

  Terms of trade are computed on maize, the local staple. On imported rice
  the wage ratio falls 29% over the same period rather than 49%. 34.5% of
  surveyed households depend on casual labour or livestock as their main
  livelihood; a further 29.5% on subsistence farming.
```

The last line is what turns a market table into an analysis. The ratio matters in proportion to how many households live on that side of it, and that number comes from the household survey rather than from the price file.

6.7 What comes next

You now have four household indicators and two market ratios, from two independent sources, all pointing at the same lean season. The last unit is what an IPC analysis does with them — and why the answer is a table rather than a formula.

Part IV

What a phase classification asserts

CHAPTER 7

Six rows, two sources, one table

Lesson 7 of 8 · 135 min

An IPC evidence table is not a spreadsheet of results. It is a structured argument in which each row carries an indicator, a value, a reliability score and a phase — and the analyst's job is the last column.

7.1 What the table is for

Lesson 4 established that four instruments give four answers. The IPC's response to that is not to pick one, average them, or find the true number. It is to lay them out so that a group of analysts can argue about them in a structured way and record what they concluded.

Every row of an evidence table carries five things:

Column	What it holds	Who supplies it
Indicator	The measure, by name	The standard
Value	The number, with its denominator and n	The analyst
Reference table	The phase thresholds for that indicator	The IPC manual
Phase	Which phase the value falls in	Arithmetic
Reliability	How much weight this row can bear	**The analyst's judgement**

The fifth column is the one that is not computed, and it is the one that makes the table an argument rather than a list.

7.2 Building the rows

```python
import pandas as pd

evidence = pd.DataFrame([
    {"indicator": "Food Consumption Score, poor",
     "value": "7.4%", "n": 1989, "source": "household survey"},
    {"indicator": "Household Hunger Scale, moderate or severe",
     "value": "42.9%", "n": 2075, "source": "household survey"},
    {"indicator": "reduced Coping Strategy Index, 19 or above",
     "value": "51.0%", "n": 2112, "source": "household survey"},
    {"indicator": "Livelihood coping, crisis or emergency",
     "value": "48.5%", "n": 2121, "source": "LCS module"},
    {"indicator": "Maize price, year on year at the lean peak",
     "value": "+22%", "n": 10, "source": "market monitoring"},
    {"indicator": "Wage to maize terms of trade, change",
     "value": "-49%", "n": 10, "source": "market monitoring"},
])
print(evidence)
```

```r
library(dplyr)

evidence <- tibble::tribble(
  ~indicator,                                ~value,   ~n,   ~source,
```

```
  "FCS poor",                              "7.4%",  1989, "household survey",
  "HHS moderate or severe",               "42.9%",  2075, "household survey",
  "rCSI >= 19",                           "51.0%",  2112, "household survey",
  "LCS crisis or emergency",              "48.5%",  2121, "LCS module",
  "Maize price, year on year",             "+22%",    10, "market monitoring",
  "Wage terms of trade, change",           "-49%",    10, "market monitoring"
)
```

Six rows, two sources, six different denominators. Building the table in code from the same script that computed the numbers is what stops the values drifting from the analysis — the same argument as the WASH denominator table, in a different sector.

7.3 Direct and indirect evidence

The IPC separates evidence by how close it sits to the outcome being classified.

Direct evidence measures the outcome itself: food consumption, nutritional status, mortality. The FCS and HHS rows are direct.

Indirect evidence measures something that causes or accompanies the outcome: prices, terms of trade, rainfall, displacement, conflict. The two market rows are indirect.

Indirect evidence cannot classify on its own. A 49% fall in wage terms of trade is a powerful argument that consumption *will* deteriorate, and it is not a measurement of consumption. The distinction matters because an analysis short of direct evidence often has plenty of indirect evidence and is tempted to promote it.

But indirect evidence is what makes a classification forward-looking. The household indicators describe the lean season that has happened; the price series describes the one still running. That is why an IPC analysis produces a current classification and a projection, and the projection leans on the indirect rows.

7.4 Reliability, and why it is scored

Not every row deserves equal weight, and the reasons are the ones this course has been accumulating.

```
evidence["reliability"] = [
    "high -- 1,989 households, standard instrument",
    "high -- 2,075 households, standard instrument",
    "medium -- no context-specific threshold established",
    "medium -- 58 households cannot be classified above crisis",
    "medium -- balanced panel of 10 of 12 markets",
    "medium -- same panel; wage series is a single occupation",
]
print(evidence[["indicator", "value", "reliability"]])
```

```
# The reliability column is prose, and it is the part that gets read.
```

Write the reason, not a score. "Medium" tells a reader nothing; "58 households cannot be classified above crisis" tells them exactly how much to discount the row and in which direction.

Four things that lower reliability, all of them met earlier in this course:

- **A denominator smaller than the sample** — the FCS's 1,989 of 2,112.
- **A threshold borrowed rather than established** — the rCSI's 19.

- **A panel that changed composition** — the price series without MK001.
- **A rule applied where the data cannot support it** — a maximum severity over a partly administered module.

7.5 What convergence looks like

```
evidence["direction"] = ["worse", "worse", "worse", "worse", "worse", "worse"]
print(f"rows pointing the same way: "
      f"{(evidence['direction'] == 'worse').sum()} of {len(evidence)}")
```

```
# Six of six. That is convergence -- and it is not the same as agreement on level.
```

All six rows point the same direction and they disagree about severity by a factor of seven. That combination — strong convergence on direction, wide disagreement on level — is the most common real pattern, and the table is what lets you say both at once.

The wrong readings of it are worth naming.

"They disagree, so the data is bad." They disagree because they measure different things, and the disagreement carries the diagnosis: coping and asset depletion high, severe consumption deficit not yet.

"Four of six say over 40%, so it is over 40%." Voting across indicators measuring different quantities is not an estimator of anything.

"The most severe row is the safest." Nor is the least severe. **The row you lead with should be the one with the highest reliability for the decision at hand**, and the others belong beside it.

7.6 The table as it goes into the workshop

```
Evidence table -- lean season 2024, four districts

  Direct evidence
    FCS poor (28/42 set)          7.4%   n=1,989   high reliability
    HHS moderate or severe       42.9%   n=2,075   high
    rCSI >= 19                   51.0%   n=2,112   medium, threshold not local
    LCS crisis or emergency      48.5%   n=2,121   medium, 58 partial modules

  Indirect evidence
    Maize, year on year at peak   +22%   10 markets, balanced panel
    Wage terms of trade           -49%   10 markets, same panel

  Convergence: six of six rows indicate deterioration.
  Divergence: severity estimates range from 7.4% to 51.0%, reflecting
  different constructs rather than measurement error.
  Not available: nutritional status, mortality, food access at the
  household level in physical quantities.
```

The "not available" block is not a formality. Nutrition and mortality are the two outcome indicators the IPC weights most heavily, and an analysis without them is classifying on consumption and coping alone. Saying so is what stops the next reader assuming they were considered and found unremarkable.

7.7 What comes next

The table is assembled. The last lesson is what a working group is entitled to conclude from it — the sentence a Phase 3 classification asserts, the one it does not, and the threshold rule most people quoting a phase have never read.

CHAPTER 8

What Phase 3 actually asserts

Lesson 8 of 8 · 135 min

A Phase 3 classification says at least one household in five in an area is in Phase 3 or worse. It does not say the area is in crisis, that four fifths are fine, or that anything about any particular household is known.

8.1 The sentence, in full

An area classified **IPC Phase 3 (Crisis)** asserts this and no more:

> At least 20% of households in the area are in Phase 3 or worse, having food consumption gaps with high or above-usual acute malnutrition, **or** are marginally able to meet minimum food needs only by depleting essential livelihood assets or through crisis coping strategies.

Four things follow from that sentence, and each is routinely lost.

8.2 It is a threshold, not an average

The 20% rule is the whole mechanism. An area is classified in the highest phase for which at least a fifth of the population meets the criteria.

```
import pandas as pd

# Illustrative distribution across an area
distribution = pd.Series({
    "Phase 1 -- None/Minimal": 0.34,
    "Phase 2 -- Stressed": 0.44,
    "Phase 3 -- Crisis": 0.18,
    "Phase 4 -- Emergency": 0.04,
})
cumulative = distribution[::-1].cumsum()[::-1]
print((cumulative * 100).round(1))
print("Highest phase reaching 20%:",
      cumulative[cumulative >= 0.20].index[-1])
```

```
# Cumulate from the worst phase down; the classification is the last one
# that still reaches 20%.
```

In that distribution, 22% are in Phase 3 or worse, so the area is Phase 3 — even though 78% are not. **The classification is about the tail, and it is designed to be.**

So "the area is in Phase 3" and "the average household is in Phase 3" are different claims, and only the first is made. A report saying "the population is in crisis" has changed a statement about a fifth into a statement about everyone.

8.3 It says nothing about any household

An area classification is an **area** classification. It is assigned to a geography from a body of evidence about that geography, and it does not classify the households in it.

This is the most consequential misuse. A Phase 3 area is not a targeting list; the households inside it range from Phase 1 to Phase 4, and identifying which is a separate exercise using household-level indicators — the ones the first two units of this course computed.

```
print("Area classification → geographic prioritisation")
print("Household indicators → who inside it is eligible")
print("These are different analyses on different units.")
```

```
# Two questions, two units of analysis, two datasets.
```

8.4 The 4 in Phase 4 is not four times the 1

The phases are **ordinal**, and the intervals between them are not equal or even defined. Phase 4 is not twice as bad as Phase 2, and an area moving from 2 to 3 has not deteriorated by "one unit".

So phases cannot be averaged, subtracted or trended arithmetically. A national figure of "mean phase 2.6" is meaningless, and the correct summary is the number of people in each phase, which is what the IPC publishes.

```
population = pd.Series({"Phase 1": 340_000, "Phase 2": 440_000,
                        "Phase 3": 180_000, "Phase 4": 40_000})
print(f"Phase 3 or above: {population[['Phase 3', 'Phase 4']].sum():,} people")
print(f"share: {population[['Phase 3', 'Phase 4']].sum() / population.sum():.1%}")
```

```
# People per phase. Never a mean phase.
```

8.5 A classification is a group's judgement, not a computation

Nothing in this course produces a phase, and that is not a gap in the course.

A phase is assigned by a technical working group that reads the evidence table, weighs the reliability of each row, applies the reference thresholds, argues, and records both the conclusion and the level of confidence in it. **The analyst's job is to build a table that makes the argument possible**, not to pre-empt it.

That is why the six-row table in the last lesson ends with a "not available" block. A working group that cannot see what is missing cannot weigh what is present.

8.6 What this analysis could and could not support

```
Lean season 2024, four districts

  What the evidence supports
    A deterioration is underway: six of six indicators point the same way,
    and the year-on-year maize comparison at the same point in the season
    is +22% with wage terms of trade down 49%.
    Coping and asset depletion are widespread: 48.5% of households used a
```

```
    crisis or emergency livelihood strategy.

What it does not support
  An area phase classification. Nutritional status and mortality are the
  outcome indicators the IPC weights most heavily and neither was
  collected. This is consumption and coping evidence only.
  A household targeting list. The survey identifies indicators, not
  eligibility, and four indicators disagree about which households are
  affected by a factor of seven.

What would settle it
  A SMART nutrition survey in the same districts, and a second market
  round in October to establish whether the peak has passed.
```

"What it does not support" is the section that gets an analysis taken seriously, and it is the section most likely to be deleted for length.

8.7 The habit these three courses were for

Module 4 has now applied the same three questions in three sectors.

1. **What did I count, over what?** Six denominators in this course alone, and no two of the four household indicators share one.
2. **What else could explain it?** Seasonality, which moves maize 74% within a year and would have been read as a programme effect.
3. **What should someone do differently because of this?** The evidence supports prioritising the districts and acting before consumption falls. It does not support a phase, and saying so is part of the answer.

The vocabulary changes by sector and the questions do not. MUAC and weight-for-height in nutrition, three functionality rates in WASH, four food security indicators here — every one of them is the same problem: a measure that looks like a fact until you ask what it counted.

8.8 What comes next

Module 4 continues with protection case management and education attendance. The instruments will be unfamiliar and the questions will not.

About this edition

This PDF is generated from the same source as the web version of the course, so the two cannot drift. The web edition carries the runnable code, the downloadable datasets and any corrections issued since this file was produced.

> Every dataset referenced in this course is synthetic or fully de-identified. No record traces to a real person.

This course is published in English and in French. The French edition is a professional adaptation using the terminology UNICEF, WHO, WFP, OCHA and national ministries publish in — not a literal translation.

Read and run the course online at data-analysis.cassion.dev/courses/food-security-ipc

Generated July 28, 2026.