# Backup slide — Key methodological choices (the “7 issues”)

- **Data source:** NTSB aviation accident database (event + cause tables), 2016–present
- **Population:** Filtered to **Part 91 accidents only** (`far_description = 'Part 91'`)
- **Accidents only:** Excludes incidents (`investigation_type = 'Accident'`)
- **Causation model:** Uses NTSB **probable cause (“C”) + contributing factors (“F”)**
- **Maintenance definition:** Text-based classification (maintenance, inspection, installation, etc.)
- **Accident-level counting:** Each accident counted once, even if multiple causes exist
- **Primary cause metric:** Restricted to `cause_level = 'C'` for “root cause” percentage

---

# Explanation (backup / Q&A slide)

If you’re wondering where those numbers come from, here’s exactly how they’re built.

This is straight out of the NTSB database—specifically the events table joined to the causes and factors table—and limited to 2016 forward.

I filtered to Part 91 only, and to accidents only, not incidents, so we’re looking at the same population we operate in.

Now, NTSB doesn’t assign a single root cause in isolation—they use a combination of probable cause and contributing factors. So for the broader number—the roughly seven percent—that includes anything where maintenance shows up anywhere in the causal chain.

To identify maintenance, I’m not using a predefined flag—because there isn’t one. It’s based on standard text matching: maintenance, inspection, installation errors, that kind of thing. That’s consistent with how most published analyses handle it.

Each accident is counted once, even if it has multiple causes, so we don’t inflate the numbers.

And then for the smaller number—the roughly 2.7 percent—that’s restricted to where maintenance is actually listed as the probable cause, not just a contributing factor.

So if anything, this method is slightly conservative, but it’s transparent and reproducible. If you run the same query on the NTSB data, you’ll land in essentially the same place.”

---

```
-- Part 91 accidents, 2016-present, with maintenance classification

WITH part91_accidents AS (
    SELECT
        e.ev_id,
        e.event_date,
        e.total_fatal_injuries
    FROM events e
    WHERE e.event_date >= '2016-01-01'
      AND e.far_description = 'Part 91'
      AND e.investigation_type = 'Accident'
),

maintenance_flags AS (
    SELECT DISTINCT
        c.ev_id,
        MAX(CASE
            WHEN LOWER(c.cause_factor_description) SIMILAR TO
'%(maint|inspection|install)%'
            THEN 1 ELSE 0 END) AS has_maintenance,
        MAX(CASE
            WHEN LOWER(c.cause_factor_description) SIMILAR TO
'%(maint|inspection|install)%'
             AND c.cause_level = 'C'
```

```
        THEN 1 ELSE 0 END) AS is_primary_maintenance
    FROM causes c
    GROUP BY c.ev_id
)

SELECT
    COUNT(*) AS total_accidents,

    SUM(CASE WHEN m.has_maintenance = 1 THEN 1 ELSE 0 END) AS
maintenance_accidents,
    ROUND(100.0 * SUM(CASE WHEN m.has_maintenance = 1 THEN 1 ELSE 0 END) /
COUNT(*), 1)
        AS pct_maintenance_any,

    SUM(CASE WHEN m.is_primary_maintenance = 1 THEN 1 ELSE 0 END) AS
primary_maintenance_accidents,
    ROUND(100.0 * SUM(CASE WHEN m.is_primary_maintenance = 1 THEN 1 ELSE 0 END)
/ COUNT(*), 1)
        AS pct_primary_maintenance,

    SUM(CASE WHEN p.total_fatal_injuries > 0 THEN 1 ELSE 0 END) AS fatal_accidents,

    SUM(CASE
        WHEN p.total_fatal_injuries > 0 AND m.has_maintenance = 1
        THEN 1 ELSE 0 END) AS fatal_with_maintenance,
```

```
    ROUND(
        100.0 * SUM(CASE
            WHEN p.total_fatal_injuries > 0 AND m.has_maintenance = 1
            THEN 1 ELSE 0 END)
        / NULLIF(SUM(CASE WHEN p.total_fatal_injuries > 0 THEN 1 ELSE 0 END), 0),
        1
    ) AS pct_fatal_with_maintenance

FROM part91_accidents p
LEFT JOIN maintenance_flags m ON p.ev_id = m.ev_id;
```