Laboratory Work No. 3

Percentiles and Cumulative Distribution Functions of Inverter Sensor Data

Related lecture: Lecture 3 — Probability Mass Functions, Cumulative Distribution Functions, Modeling Distributions, and Probability Density Functions

Dataset: the same GECAD Photovoltaic (PV) system file used in Labs 1 and 2, PV2013January.xlsx — same student, same assigned date as in Labs 1 and 2.

Purpose of the work

To practice reading a real sensor dataset into a pandas DataFrame and summarizing a variable with percentiles; to build and interpret the empirical cumulative distribution function (CDF) of several sensor variables; and to see directly how a large spike of zero values distorts both a variable's percentiles and its CDF, and how filtering those zeros out reveals the real shape underneath.

Background

1. The three variables for this lab. As in Lab 2, every sheet of the workbook records three columns that each tell a different part of the story of how the inverter behaves over a day:

Column	Meaning	Why it's interesting
F — TmpMdul C	Module (panel) temperature, °C	Varies continuously all day and night — a genuine physical quantity with no "off" state.
I — Fac	Frequency of the AC current the inverter generates, Hz	Is exactly 0 whenever the inverter is idle (no sunlight); only takes a real value (≈50 Hz) while actively feeding the grid.
P — Pac	AC power generated, W	Same pattern as Fac: 0 at night, a real distribution of values during the day.

This distinction matters, exactly as in Lab 2: TmpMdul C never has a reason to be exactly zero, while Fac and Pac are legitimately zero for a large part of every day. Step 2.3 of the task below asks you to decide, for each column, whether that zero spike needs to be filtered out before the percentiles and CDF are meaningful.

2. Percentiles (Lecture 3 / Think Stats Ch. 4). The *p*th percentile of a variable is the value below which *p*% of the observations fall. The 50th percentile is the median; the 25th and 75th percentiles (together, the interquartile range or IQR) describe where the middle half of the data sits. NumPy computes all three in one call:

```
import numpy as np

p25, p50, p75 = np.percentile(data, [25, 50, 75])
```

Percentiles are more informative than a single mean whenever a distribution is skewed or has a large spike at one value — exactly the situation with Fac and Pac in this dataset.

3. The empirical CDF (Lecture 3 / Think Stats Ch. 4). The cumulative distribution function of a sample is the function CDF(x) = fraction of observations ≤ x. Plotted against the sorted data values, it rises from 0 to 1 and its height at any x is exactly the percentile rank of x. Unlike a histogram, a CDF needs no bin-count choice at all — every single observation is represented individually as one small step, so two students who read the same data will always draw the identical curve.

```
xs = np.sort(data.to_numpy())
ys = np.arange(1, len(xs) + 1) / len(xs)
plt.plot(xs, ys)
```

A CDF that jumps steeply over a narrow range of x means many observations are packed into that range; a CDF that stays flat means very few observations lie there. A long flat segment starting at x = 0 is the signature of a large spike of zero values — watch for exactly this shape in the raw Fac and Pac plots below.

4. Filtering out zeros. A Boolean condition filters a Series exactly as in Labs 1 and 2: `nonzero = series[series > 0]`. This is worth doing whenever a large spike at zero represents a different regime ("inverter off") rather than genuine variability of the quantity being studied — otherwise that spike dominates both the percentiles (the 25th and even 50th percentile can collapse to exactly 0) and the CDF (a long flat shelf at the bottom of the plot), hiding the real distribution of the values that matter.

Task

Use the same date (variant) and the same CSV-reading approach as in Labs 1 and 2. For each of the three columns F (TmpMdul C), I (Fac), and P (Pac):

- 2.1 — Print the 25th, 50th, and 75th percentiles of the column.
- 2.2 — Plot the column's empirical CDF.
- 2.3 — Decide whether the column needs its zero values filtered out before the percentiles and CDF are meaningful, and filter them if so (see the Background section above for how to decide).
- 2.4 — Using the filtered data from 2.3 (where filtering was needed), repeat both the percentiles (2.1) and the CDF plot (2.2).

Place your code and all resulting graphs (CDFs, raw and — where applicable — filtered, for all three columns) into your report.

Optional / bonus tasks

- Plot all three raw CDFs (F, I, P) on one set of axes with a legend, and comment on why their shapes differ so much even though all three come from the same 288 rows of data.
- For Fac or Pac, mark the 25th, 50th, and 75th percentiles as horizontal and vertical guide lines directly on the CDF plot, before and after filtering.
- Compare the interquartile range (75th − 25th percentile) of Fac and Pac, before and after filtering zeros, and explain what the change tells you about each variable.
- The CDF of TmpMdul C should look like a smooth, mostly straight-ish curve rather than a curve with a flat shelf. Explain, in one or two sentences, why that column's CDF looks fundamentally different in shape from the raw Fac and Pac CDFs.

Report requirements

Your submitted report must contain:

- the topic and purpose of the work;
- your variant number and assigned date (as in Labs 1 and 2);
- your program code for all of steps 2.1–2.4, for all three columns;
- the printed 25th/50th/75th percentiles (raw and, where filtering was applied, filtered) for all three columns;
- all resulting CDF graphs — raw for every column, plus filtered for any column where zeros were removed;
- a short conclusion: which columns needed zero-filtering and why, how the percentiles changed after filtering, and what the shape of each CDF told you that the percentiles alone would not have.

Worked example (teacher's variant)

Date used: **15-01-2013** — the same date and the same `data_pv_15-01-2013.csv` file as the Lab 1 and Lab 2 worked examples.

Column F — TmpMdul C (module temperature)

Quick check first: `(df['TmpMdul C'] == 0).sum()` returns **0** — there are no zero readings at all, so step 2.3 requires no filtering for this column.

```
tmod = df['TmpMdul C']
print((tmod == 0).sum())                  # 0 -- no filtering needed
print(np.percentile(tmod, [25, 50, 75]))

xs = np.sort(tmod.to_numpy())
ys = np.arange(1, len(xs) + 1) / len(xs)
plt.plot(xs, ys)
```

Percentile	Value
25th	16.43 °C
50th (median)	17.84 °C
75th	18.79 °C

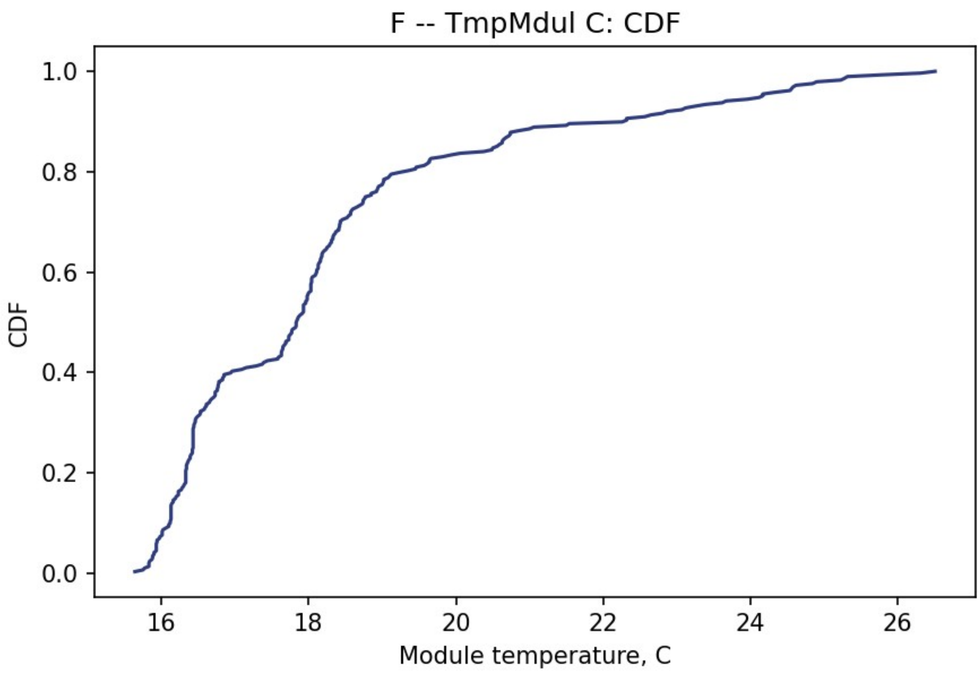

Figure 1. TmpMdul C: empirical CDF. No zero spike, no flat shelf -- a smooth, continuously-rising curve typical of a genuine physical measurement with no “off” state.

Column I — Fac (AC frequency)

Quick check: `(df['Fac'] == 0).sum()` returns **181** out of 288 readings (63%) — the inverter is idle most of the night and early morning, exactly as found in Lab 2. This is exactly the case described in the Background section: the zero values represent a different regime, not real variability of the frequency, so step 2.3 calls for filtering.

```
fac = df['Fac']
print((fac == 0).sum())                    # 181 / 288 (63%)
print(np.percentile(fac, [25, 50, 75]))     # raw

fac_nz = fac[fac > 0]                       # step 2.3: filter idle-inverter zeros
print(np.percentile(fac_nz, [25, 50, 75])) # filtered
```

Percentile	Raw (n = 288)	Filtered, Fac > 0 (n = 107)
25th	0.00 Hz	49.99 Hz
50th (median)	0.00 Hz	49.99 Hz
75th	49.99 Hz	50.00 Hz

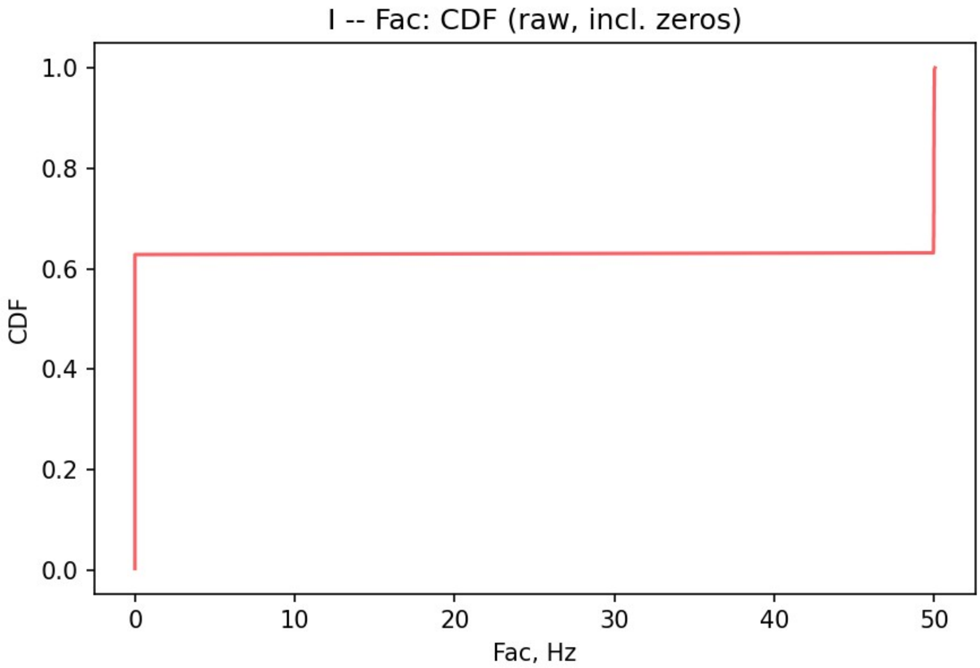

Figure 2. Fac: empirical CDF on RAW data. A long flat shelf at CDF = 0 up to Fac = 0, then a vertical jump to about 0.63, is the unmistakable signature of the 63% zero spike -- this is exactly why the raw 25th and 50th percentiles above are both 0.

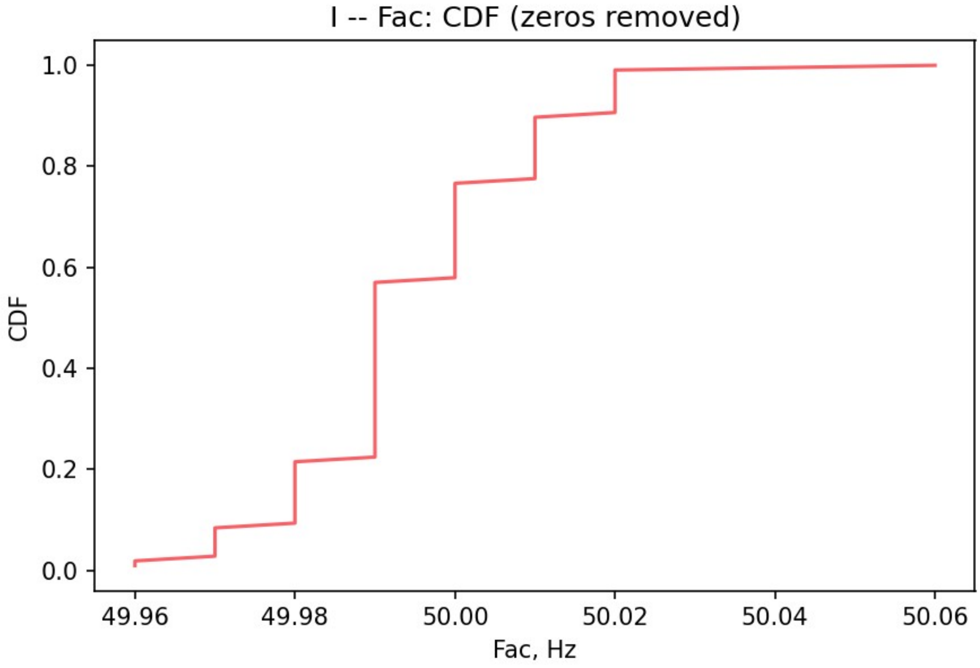

Figure 3. Fac: empirical CDF with zeros removed. The real structure -- grid frequency held tightly between 49.96 and 50.06 Hz, almost all of it packed between 49.98 and 50.02 Hz -- is only visible once the flat shelf is gone.

Column P — Pac (AC power)

Quick check: `(df['Pac'] == 0).sum()` returns **191** out of 288 (66%) — the same idle-at-night pattern as Fac, so the same filtering decision applies.

```
pac = df['Pac']
print((pac == 0).sum())                    # 191 / 288 (66%)
print(np.percentile(pac, [25, 50, 75]))     # raw

pac_nz = pac[pac > 0]                       # step 2.3
print(np.percentile(pac_nz, [25, 50, 75])) # filtered
```

Percentile	Raw (n = 288)	Filtered, Pac > 0 (n = 97)
25th	0.00 W	20.73 W
50th (median)	0.00 W	38.98 W
75th	21.04 W	63.13 W

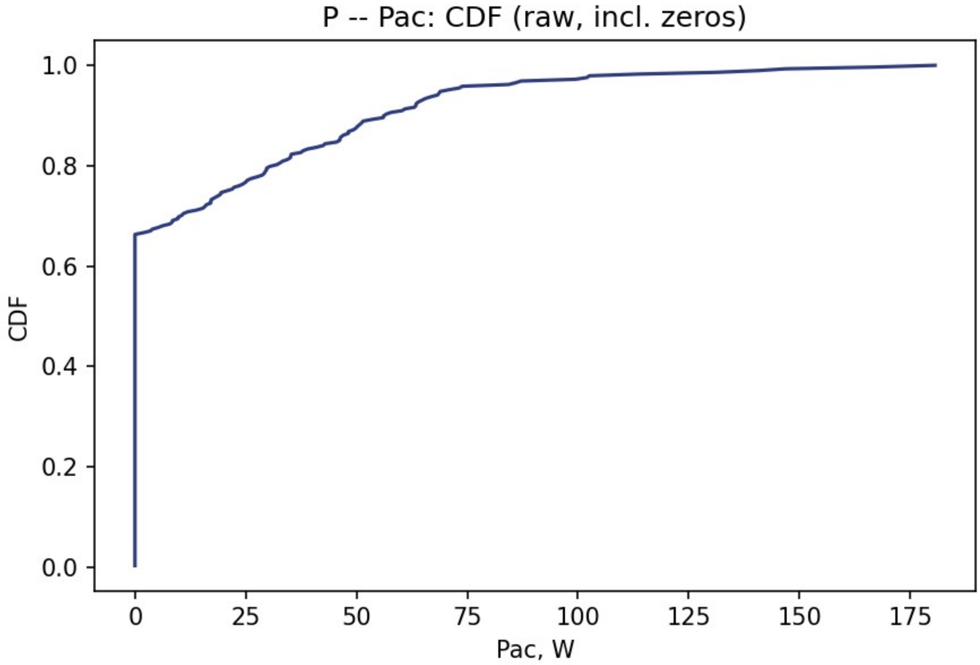

Figure 4. Pac: empirical CDF on RAW data. The same flat-shelf-then-jump shape as Fac -- the shelf reaches up to about 0.66 this time, matching the larger 66% zero share.

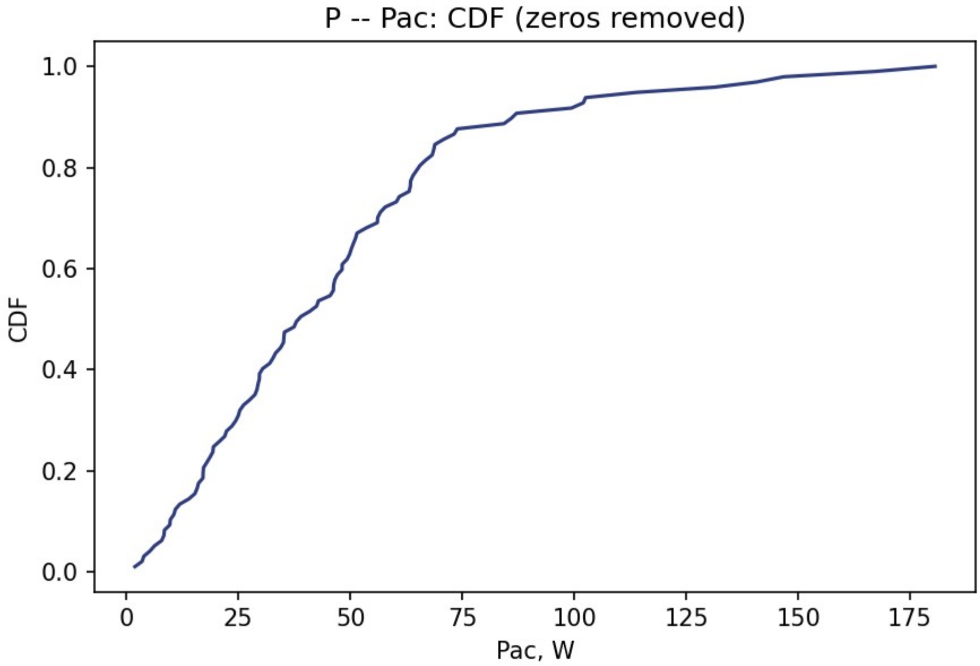

Figure 5. Pac: empirical CDF with zeros removed. A steadily, smoothly rising curve with no single dominant step -- consistent with a right-skewed but continuously-varying power output, unlike the narrowly clustered Fac.

Summary of decisions

Column	Zeros filtered?	25th → filtered	Median → filtered	75th → filtered
F — TmpMdul C	No (no zeros present)	16.43 (unchanged)	17.84 (unchanged)	18.79 (unchanged)
I — Fac	Yes (181/288, 63%)	0.00 → 49.99	0.00 → 49.99	49.99 → 50.00

Column	Zeros filtered?	25th → filtered	Median → filtered	75th → filtered
P — Pac	Yes (191/288, 66%)	0.00 → 20.73	0.00 → 38.98	21.04 → 63.13

Interpretation

The three columns again illustrate why a distribution should be looked at rather than trusted to a single summary number — this time through percentiles and the CDF rather than the histograms and PMFs of Lab 2. TmpMdul C needed no cleaning: its CDF rises smoothly across its whole range, and its percentiles are stable, informative numbers on their own. Fac and Pac both have a majority of readings at exactly zero (the inverter idle overnight), and this shows up in their raw CDFs as a long flat shelf sitting at CDF = 0 followed by a sudden near-vertical jump — a shape that also drags the raw 25th and even 50th percentiles down to exactly 0, hiding any information about how the inverter actually behaves while it is running. Filtering out the zeros restores a meaningful CDF and meaningful percentiles for both columns, and the two filtered CDFs also look different from each other: Fac's filtered CDF rises almost vertically over a very narrow band of frequencies (the grid frequency is tightly regulated), while Pac's filtered CDF climbs steadily across a wide range of power values (a continuously varying, right-skewed output). None of this — the flat shelf, the near-vertical jump, or the difference in filtered spread — would be visible from the mean or the raw percentiles alone.

Control questions

- Why do the raw 25th and 50th percentiles of Fac and Pac come out as exactly 0, while the raw 75th percentile of Fac does not?
- What feature of a CDF plot immediately tells you that a variable has a large spike of identical (e.g. zero) values, without looking at any percentile numbers at all?
- Unlike a histogram, a CDF does not require choosing a number of bins. Why does that make CDFs particularly useful for comparing distributions computed by different students, or at different times?
- Why does the filtered Fac CDF rise almost vertically, while the filtered Pac CDF rises much more gradually? What does that difference say about how tightly each quantity is controlled?
- If you only had the 25th, 50th, and 75th percentiles of Pac (raw, not filtered) and no plot at all, could you tell that the inverter is idle most of the day? What is the smallest change to which percentiles you'd need to compute to detect this?
- Suppose a future column in this dataset has its 25th percentile equal to its 75th percentile. What would that imply about the shape of its distribution?

> *Reminder: use your own assigned date's sheet (as in Labs 1 and 2) rather than 15-01-2013 when producing your own report — your zero counts, percentiles, and CDF shapes will differ from the numbers shown here, and that is expected.*