PRINCIPLES AND METHODS OF DATA ANALYSIS IN THE ENERGY INDUSTRY

Laboratory Work No. 2

Histograms and Probability Mass Functions of Inverter Sensor Data

Related lecture: Lecture 2 — Plotting, DataFrames, and Distributions

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

Purpose of the work

To practice reading a real sensor dataset into a pandas DataFrame and visualizing the distribution of several variables with histograms; to build intuition for how the number of bins changes the picture a histogram presents; to recognize when a variable's distribution is dominated by a large spike of zero values, and to handle that deliberately; and to see, side by side, that a probability mass function (PMF) is the same shape as a histogram, only rescaled onto a probability axis.

Background

1. The three variables for this lab

Every sheet of the workbook records, among others, the following three columns, each telling 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 it is 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: TmpMdul C never has a reason to be exactly zero, while Fac and Pac are legitimately zero for a large part of every day (whenever the inverter has nothing to convert). Recognizing which situation you are in is exactly what step 2.3 of the task asks you to decide for yourself.

2. Histograms and the number of bins (Lecture 2 / Think Stats Ch. 2)

A histogram groups a variable's values into equal-width intervals (“bins”) and counts how many observations fall into each one. In pandas/Matplotlib, `series.hist(bins=n)` or `plt.hist(series, bins=n)` draws it directly.

```
df['TmpMdul C'].hist(bins=15)
plt.xlabel('Module temperature, C'); plt.ylabel('count')
```

Why the bin count matters

Too few bins hide real structure (everything blurs into one or two bars); too many bins make the picture noisy (each bar reflects only a handful of samples, and random fluctuation dominates the shape). There is no single “correct” number — the right choice is the one that reveals the structure you actually care about, found by trying a few values and comparing, exactly as step 2.2 of the task asks.

3. Filtering out zeros

A Boolean condition filters a Series exactly as in Lab 1: `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 you are studying — otherwise that single spike visually swallows the entire rest of the distribution, no matter how many bins you choose.

4. From histogram to PMF

A histogram shows counts; a **probability mass function (PMF)** shows the same shape rescaled so that the bars sum to 1 — each bar is the *fraction* of observations in that bin rather than their raw count. The simplest way to draw one with Matplotlib, without any extra library, is to pass every observation an equal weight of 1/n:

```
import numpy as np

weights = np.ones_like(data) / len(data)
plt.hist(data, bins=n, weights=weights)   # same bins as the histogram above
plt.ylabel('probability')
```

Using the same data and the same number of bins for both plots is essential — that is precisely what makes the comparison in step 2.4 meaningful: the two charts should look identical in shape, differing only in what the y-axis measures.

Task

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

- 2.1 — Visualize the column's data using a histogram.
- 2.2 — Try several different numbers of bins; keep the one that gives the clearest picture of the data's shape.

- 2.3 — Decide whether the column needs its zero values filtered out before the picture is meaningful, and filter them if so (see the Background section above for how to decide).
- 2.4 — Using the same number of bins chosen in 2.2 (and the same filtering decision from 2.3), redraw the same data as a PMF instead of a histogram.

Place your code and all resulting graphs (histograms and PMFs, for all three columns) into your report.

Optional / bonus tasks

- For Fac and Pac, show the 'raw vs. filtered' histograms side by side and explain in one or two sentences why filtering changed the picture so much.
- For one of the three columns, show three histograms with different bin counts (e.g. too few, about right, too many) side by side, and briefly justify your final choice.
- Compute the mean and standard deviation of each column before and after filtering zeros, and comment on how much the zeros were distorting the summary statistics (compare to the special-code cleaning example from Lab 1).
- TmpMdul C never reaches exactly zero in this dataset — but can you think of a sensor or situation in an energy system where a temperature reading of exactly 0 would itself be a sign of a data problem, not a real measurement? Explain briefly.

Report requirements

Your submitted report must contain:

- the topic and purpose of the work;
- your variant number and assigned date (as in Lab 1);
- your program code for all of steps 2.1-2.4, for all three columns;
- all resulting graphs: one histogram and one PMF per column, using your final bin-count choice;
- a short conclusion: which columns needed zero-filtering and why, what bin count you settled on for each column and why, and anything the shape of a distribution told you that a single summary number (mean, max) 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 worked example.

Column F — TmpMdul C (module temperature)

Quick check first: `(df['TmpMdul C'] == 0).sum()` returns **0** — there are no zero readings at all (min = 15.64 °C, max = 26.51 °C), so step 2.3 requires no filtering for this column.

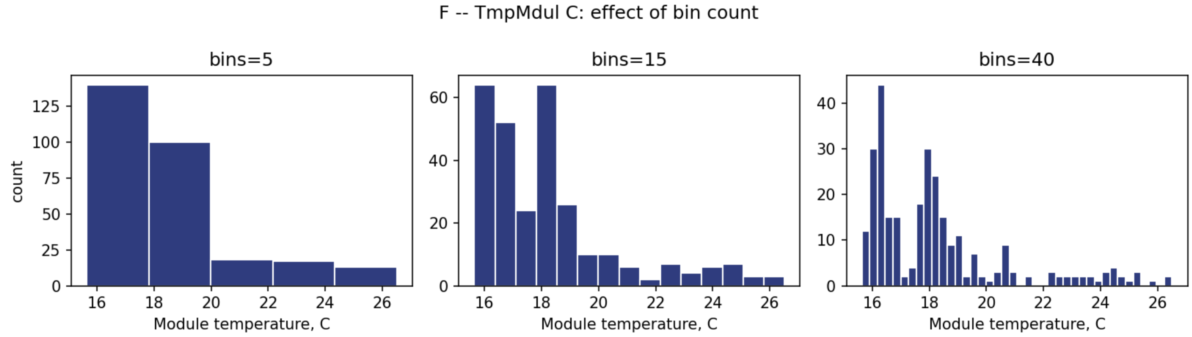

Figure 1. TmpMdul C histogram at 5, 15, and 40 bins. 5 bins hides the dip around 19-20 C; 40 bins is noisy. 15 bins was kept as the clearest picture.

```
tmod = df['TmpMdul C']
tmod.hist(bins=15)

weights = np.ones_like(tmod) / len(tmod)
plt.hist(tmod, bins=15, weights=weights)
```

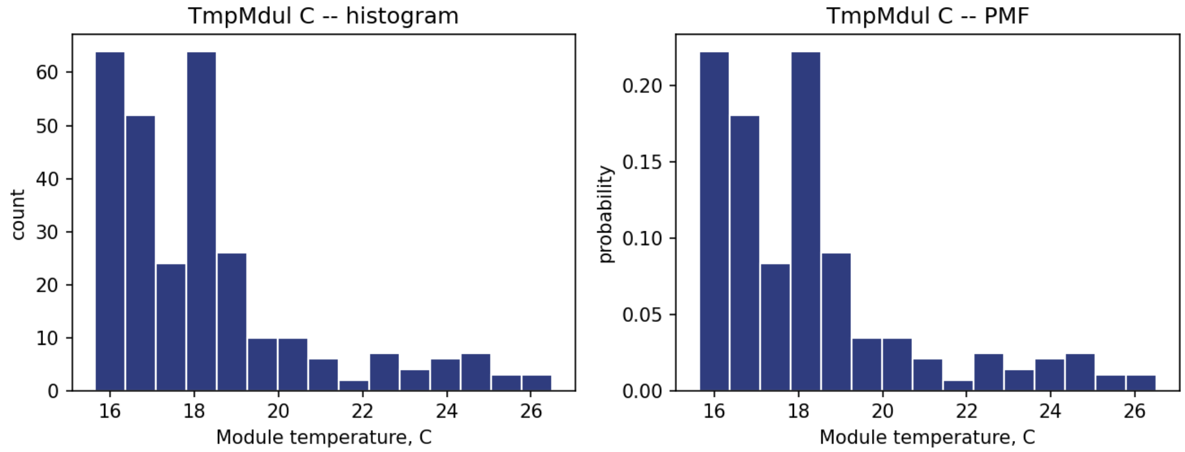

Figure 2. TmpMdul C: histogram (left) and PMF (right) at 15 bins — identical shape, rescaled y-axis.

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. This is exactly the case described in the Background section: the zero values represent a different *regime*, not real variability of the frequency.

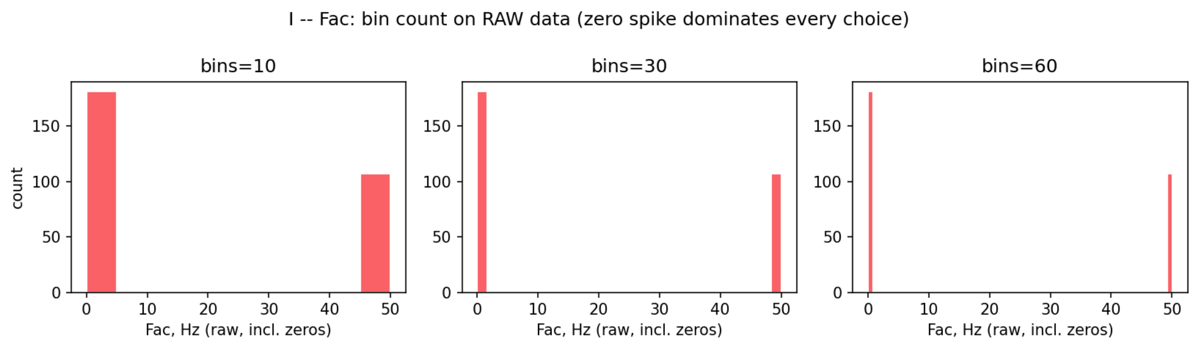

Figure 3. Fac histogram on RAW data at 10, 30, and 60 bins. No bin count helps — the zero spike (idle inverter) dominates every choice equally.

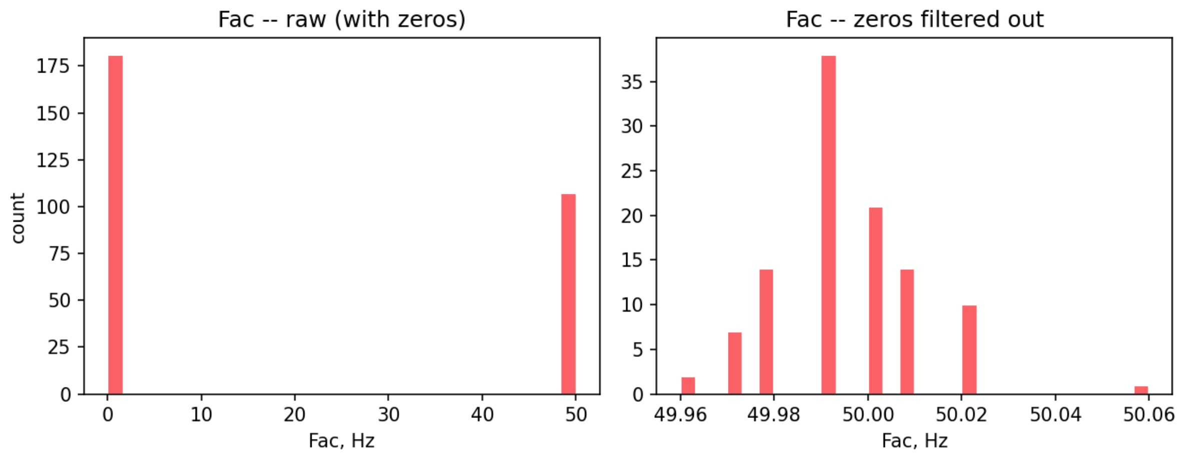

Figure 4. Fac before and after removing zero values (bins=30). Only once the zeros are removed does the real structure — the grid frequency held tightly between 49.96 and 50.06 Hz — become visible.

```
fac = df['Fac']
fac_nz = fac[fac > 0]          # step 2.3: filter out the idle-inverter zeros

fac_nz.hist(bins=15)
weights = np.ones_like(fac_nz) / len(fac_nz)
plt.hist(fac_nz, bins=15, weights=weights)
```

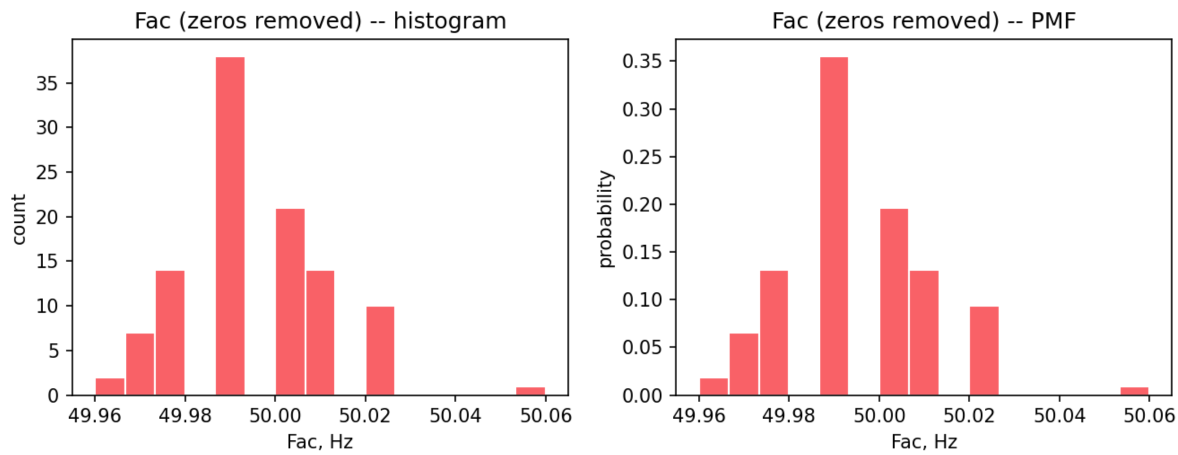

Figure 5. Fac (zeros removed): histogram (left) and PMF (right) at 15 bins.

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.

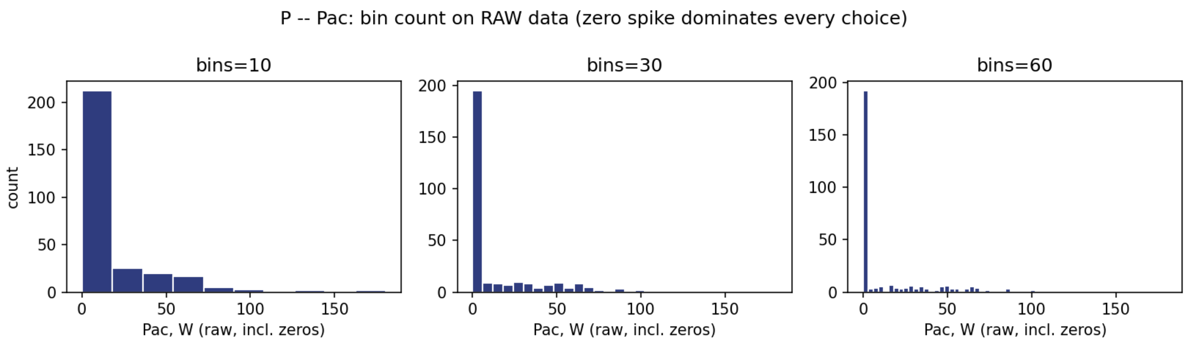

Figure 6. Pac histogram on RAW data at 10, 30, and 60 bins — again, the zero spike (no generation at night) dominates regardless of bin count.

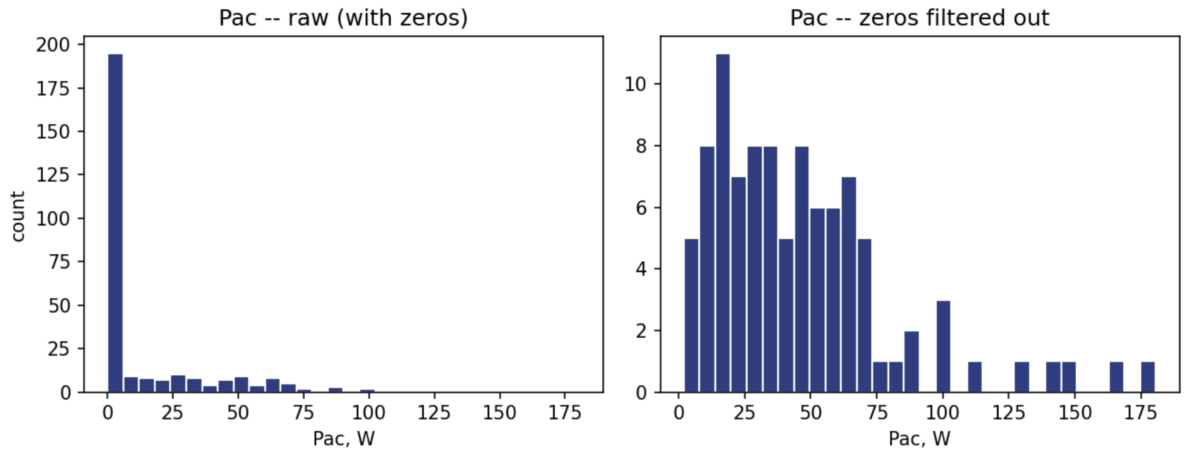

Figure 7. Pac before and after removing zeros (bins=30). The filtered distribution is right-skewed: many low-to-moderate power readings and a long tail of high-power readings around solar noon — consistent with the irradiance curve from Lab 1.

```
pac = df['Pac']
pac_nz = pac[pac > 0]            # step 2.3

pac_nz.hist(bins=20)
weights = np.ones_like(pac_nz) / len(pac_nz)
plt.hist(pac_nz, bins=20, weights=weights)
```

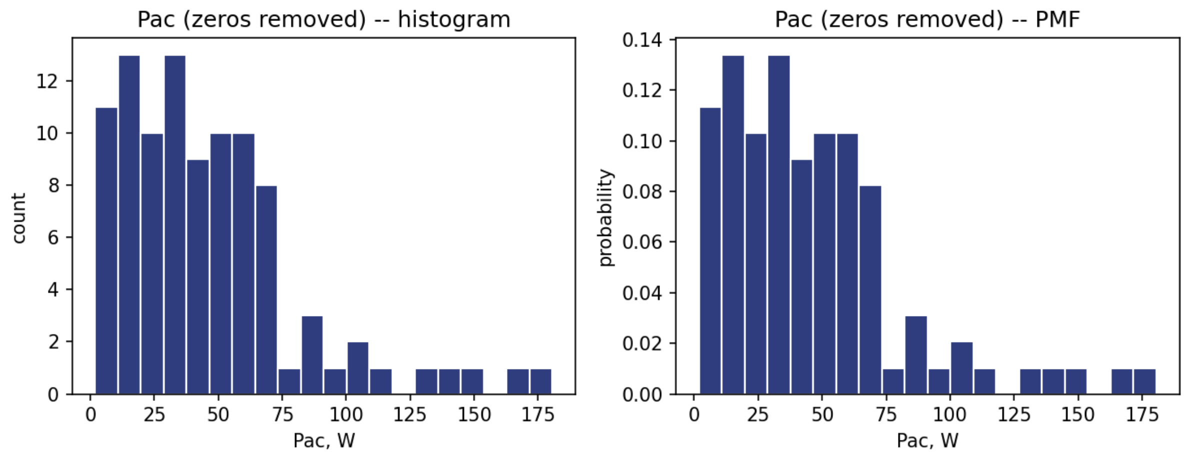

Figure 8. Pac (zeros removed): histogram (left) and PMF (right) at 20 bins.

Summary of decisions

Column	Zeros filtered?	Bins used	Nonzero mean
F — TmpMdul C	No (no zeros present)	15	18.24 °C
I — Fac	Yes (181/288 zeros removed)	15	49.99 Hz
P — Pac	Yes (191/288 zeros removed)	20	46.48 W

Interpretation

The three columns illustrate three different reasons to look at a distribution rather than a single average. TmpMdul C needed no cleaning at all, and its histogram shows a temperature that spends most of the (cold January) day clustered low, with a shorter tail of warmer daytime readings. Fac, once cleaned, is not a smooth bell curve at all — it clusters into a few discrete steps very close to 50 Hz, reflecting how the grid frequency is regulated in narrow, almost quantized bands rather than varying freely. Pac, once cleaned, shows the shape you would expect from a solar inverter: many samples at low-to-moderate output and a long right tail toward its midday maximum, mirroring the irradiance curve from Lab 1. None of this — the near-zero spread of Fac, or the skew of Pac — would be visible from the mean and maximum alone.

Control questions

- Why does TmpMdul C not need zero-filtering while Fac and Pac do? What general rule would you use to decide, for a column you have never seen before, whether its zeros need filtering?
- Why does increasing the number of bins on the raw (unfiltered) Fac or Pac data fail to reveal any more structure, no matter how high you set it?
- What exactly changes, and what stays the same, when you turn a histogram into a PMF using the same bin count?
- If two students used a different number of bins for the same column, could they reach different conclusions about the shape of the distribution? Give a concrete example using one of this lab's columns.
- Why is the Fac distribution (after filtering) so much narrower, relative to its mean, than the Pac distribution?
- Suppose a fourth column in this dataset is exactly zero for every single row. What would that most likely indicate, and what would you check before analyzing it further?