# Radial Climatological Rings Chart (Python)

Concentric “seasonality rings” for monthly data. Each ring is a year (inner = most recent), and each ring has 12 wedges (Dec at 12 o’clock, Jan at 1 o’clock, …, Nov at 11 o’clock). Designed for anomalies **and** absolute values (e.g., CO₂), with smart color scaling, publication-ready output, and a friendly CLI.

## ✨ Features

* **Rings by year**: inner = latest, outer = earliest.
* **Clockwise month layout**: **Dec at 12 o’clock**.
* **Smart color scaling**:
  + *Smart symmetric* (default): auto-detects if data crosses 0; uses symmetric scale for anomalies, non‑symmetric otherwise.
  + Optional robust min/max via quantiles.
  + Exact colorbar step control (e.g., **0.1**), with automatic **tick thinning** to avoid over-plotting.
* **Aligned colorbar**: ticks and segment boundaries line up.
* **Month labels** around the outside.
* **Configurable year axis**: angle, tick spacing, label offset, bold font, widths.
* **Headless**: saves straight to PNG; no GUI window.

## 🚀 Quick start

# 1) Install dependencies (Python 3.9+)  
pip install numpy pandas matplotlib  
  
# 2) Run on an anomaly dataset  
python seasonal\_ring\_chart.py \  
 --data HadSST.4.2.0.0\_monthly\_GLOBE.csv \  
 --value-col anomaly --year-col year --month-col month \  
 --year-min 1970 --year-max 2025 \  
 --title "Global SST anomaly (HadSST4)" \  
 --hide-center-label --month-labels \  
 --year-axis-angle-deg 165 --year-label-step 10 \  
 --cbar-step 0.1 --cbar-shrink 0.8 --cbar-fraction 0.045 --cbar-pad 0.03 \  
 --year-label-fontsize 6 --year-label-weight bold --year-label-offset 0.22 \  
 --year-axis-linewidth 1.4 --year-tick-width 1.4 --year-tick-length 0.10  
  
# 3) Run on a positive-only dataset (e.g., CO₂ ppm)  
python seasonal\_ring\_chart.py \  
 --data co2\_mm\_mlo.csv \  
 --value-col average --year-col year --month-col month \  
 --year-min 1970 --year-max 2025 \  
 --title "Mauna Loa CO₂ (ppm)" \  
 --hide-center-label --month-labels \  
 --year-axis-angle-deg 165 --year-label-step 10 \  
 --cbar-shrink 0.6 --cbar-fraction 0.05 --cbar-pad 0.03 \  
 --year-label-fontsize 8 --year-label-weight bold --year-label-offset 0.20 \  
 --year-axis-linewidth 1.2 --year-tick-width 1.2 --year-tick-length 0.10

The script auto-picks sensible colorbar steps when --cbar-step is omitted and caps the number of segments/ticks to keep plots crisp.

## 📦 Installation

* Python **3.9+**
* pip install numpy pandas matplotlib

Optional: create and activate a virtualenv/conda env before installing.

## 🧭 Usage

python seasonal\_ring\_chart.py --data <file.csv> --value-col <name> --year-col <name> --month-col <name> [options]

### Data expectations

* CSV with at least **year**, **month**, and **value** columns.
* Month can be **1–12** or strings (e.g., Jan, January). Use --month-format if needed.
* Missing months are drawn in a neutral grey.

### Month layout

* The plot is rotated so **December sits at 12 o’clock** (top). January is at roughly 1 o’clock, etc.

## 🔧 Most useful options (short list)

* --year-min/--year-max : clamp the time window.
* --hide-center-label : remove the center text.
* --month-labels : draw month labels around the outside (default on).
* --year-axis-angle-deg 165 : place year axis between **Sep & Oct**.
* --year-label-step 10 : label every 10 years (also controls tick placement).
* --year-label-offset 0.2 : nudge labels away from the axis line.
* --year-axis-linewidth/--year-tick-width/--year-tick-length : styling for the year axis.
* --cbar-step 0.1 : force exact colorbar step (segments & labels aligned).
* --cbar-shrink 0.8 and --cbar-fraction 0.045 : colorbar length & thickness.

See **User Manual** for the full option reference.

## 🖼 Output

* Always saves a PNG (default filename: seasonal\_ring\_chart\_<value>\_<firstYear>-<lastYear>.png).
* --save out.png to set the filename.
* --dpi 600 by default for publication quality.

## 💡 Tips

* For **reproducible color scales across runs**, provide --vmin/--vmax.
* For small anomaly ranges, --cbar-step 0.1 looks clean.
* If labels feel crowded, increase --year-label-step, reduce --year-label-fontsize, or tweak --month-label-fontsize.
* Very long year ranges: consider shrinking --ring-width or increasing --figsize.

## 🧪 Development

* No GUI: the script sets matplotlib to the **Agg** backend.
* Layers: data wedges at low z-order; labels/axis on top (prevents the center hole from hiding labels).
* Complexity: O(12 × years). Even large datasets should render quickly.

## 🤝 Contributing

Issues and PRs welcome! Please include a minimal CSV sample and the exact command you ran.

## 📄 License

MIT License © Your Name. See LICENSE (add one if not present).