

**I also need a tree and workflow for my programmer**

Here is a **workflow tree** and **annotated code workflow** tailored for your Relative Strength (RS) analysis and visualization suite, based on your scripts and files. This will help your programmer understand implementation, data flow, and integration.[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)[[4]](#fn4)

**Relative Strength Analysis Workflow Tree**

1. Data Ingestion  
 └─ Load RS CSVs (daily, weekly, monthly)  
 ├─ rs\_ibd\_stocks\_0\_daily.csv  
 ├─ rs\_ibd\_stocks\_0\_weekly.csv  
 ├─ rs\_ibd\_stocks\_0\_monthly.csv  
 ├─ rs\_ibd\_sectors\_0\_daily.csv  
 ├─ rs\_ibd\_sectors\_0\_weekly.csv  
 ├─ rs\_ibd\_sectors\_0\_monthly.csv  
 ├─ rs\_ibd\_industries\_0\_daily.csv  
 ├─ rs\_ibd\_industries\_0\_weekly.csv  
 └─ rs\_ibd\_industries\_0\_monthly.csv  
  
2. Data Processing  
 ├─ For each timeframe (daily, weekly, monthly):  
 ├─ Calculate/aggregate RS percentiles per stock, sector, industry  
 ├─ Select top/bottom performers  
 └─ Prepare summary tables for charts  
  
3. Analysis & Feature Engineering  
 ├─ Multi-timeframe momentum/value structure  
 ├─ Sector and industry rotation signals  
 ├─ Identify elite leaders, watchlists, and value turnaround candidates  
  
4. Visualization  
 ├─ chart\_heatmap\_per.py (Multi-Timeframe RS Heatmap for stocks)  
 ├─ chart\_momentum\_vs\_value\_scatter.py (Momentum vs Value scatter plot)  
 ├─ chart\_sector\_performace\_by\_time.py (Sector RS Radar Chart across timeframes)  
 └─ chart\_top10\_per.py (Top Daily RS Leaders Bar Chart)  
  
5. Output & Integration  
 ├─ Export chart visualizations as .png or .svg files  
 ├─ (Optional) Publish reports, dashboards, or web visualization  
 └─ Save processed tables for use in further analysis

**Annotated Code Workflow For Implementation**

**1. Data Ingestion**

* **File(s) loaded via pandas:**

df\_stocks = pd.read\_csv('rs\_ibd\_stocks\_0\_daily.csv')  
df\_sectors = pd.read\_csv('rs\_ibd\_sectors\_0\_daily.csv')  
df\_industries = pd.read\_csv('rs\_ibd\_industries\_0\_daily.csv')  
# Repeat for weekly/monthly as needed

**2. Data Processing**

* **Aggregate/calculate RS percentiles:**
  + Use pandas groupby or selection/filter for percentiles per ticker/sector/industry.
  + Summarize top RS values for "top 10" or multi-timeframe analysis dataframes.

**3. Analysis**

* **Identify leadership and rotation:**
  + Compare short, medium, and long-term RS using columns (1D\_RS%, 7D\_RS%, etc).
  + Construct tables for heatmap, scatter, radar, bar chart as needed by visual scripts.

**4. Visualization**

* **Run specific scripts for each chart:**
  + **Heatmap:**

# chart\_heatmap\_per.py: input summary table, output PNG heatmap  
fig = go.Figure(data=go.Heatmap(...))  
fig.write\_image('rs\_heatmap.png')

* + **Momentum vs Value Scatter Plot:**

fig = go.Figure()  
fig.add\_trace(go.Scatter(...))  
fig.write\_image('rs\_percentiles\_scatter.png')

* + **Sector Radar Chart:**

fig = go.Figure()  
fig.add\_trace(go.Scatterpolar(...))  
fig.write\_image('sector\_radar\_chart.png')

* + **Top 10 Bar Chart:**

fig = px.bar(...)  
fig.write\_image('top\_10\_stocks\_rs\_percentile.png')

**5. Output & Integration**

* **Save visualization images locally/in cloud**
* **Export summary tables (.csv/.json) for dashboards or distribution**
* **(Optional) Integrate charts in reporting workflows or web dashboards (Plotly Dash, Flask, etc.)**

**Data/Chart Processing Diagram Example**

RS CSVs (daily/weekly/monthly)  
 │  
 pandas DataFrames  
 │  
 ├─ Prepare tables (multi-timeframe, top performers, rotation)  
 │  
 │  
 ├─ chart\_heatmap\_per.py → rs\_heatmap.png  
 ├─ chart\_momentum\_vs\_value\_scatter.py → rs\_percentiles\_scatter.png  
 ├─ chart\_sector\_performace\_by\_time.py → sector\_radar\_chart.png  
 ├─ chart\_top10\_per.py → top\_10\_stocks\_rs\_percentile.png  
 │  
 Output summary images & tables  
 │  
 Dashboard/report integration (optional)

**Implementation Notes**

* All chart scripts rely on **pandas** and **plotly**—ensure those libraries are installed.
* Place the relevant **RS CSVs** in the project/data directory.
* Each chart script expects pre-processed tables (see script examples for required column names).
* Output charts are generated via write\_image() functions in Plotly.

This workflow tree and code diagram gives your programmer clear guidance for the RS analysis project, regardless of the platform (local batch, dashboard, reporting pipeline).[[4]](#fn4)[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)

⁂

1. chart\_heatmap\_per.py

1. chart\_momentum\_vs\_value\_scatter.py

1. chart\_sector\_performace\_by\_time.py

1. chart\_top10\_per.py

1. Comprehensive-Relative-Strength-Analysis-Market.docx

1. Comprehensive-Relative-Strength-Analysis-Visuali.docx