Contents

1 FigRecipe MCP Server + Claude Code Demo

2 Request (DO NOT CHANGE THIS SECTION)

- Verify that FigRecipe MCP Server is available
- Demonstrate the MCP server live under your responsibility
  - 1. Overview
  - 2. Types of Plots
  - 3. Representative Recipe Contents and File/Directory Structure
  - 4. Panel Composition
  - 5. Figure- and Panel-Level Captions
  - 6. Statistical Annotations
  - 7. Reproduction
- Create and use demo CSV data and refer from recipes instead of inline data
- Add contents/documentation to this org file interactively
- Proceed step by step. One plot and one narration form a set.
- Play narration between sections
- Store all artifacts in ./demo$_{\text{mcpout}}$/ (remove it if it already exists)
- Add documentation and place inline figures
- Style: width 40mm, height 28mm, white background instead of transparent, SCITEX STYLE

- Do not check source code, unless explicitly instructed, to purely check MCP server capabilities

3 Emacs Org Mode Setup (DO NOT CHANGE THIS SECTION)

```
(defun my/setup-demo-org ()
  (auto-revert-mode t)
  (run-with-timer 0 5
    (lambda ()
      (when (derived-mode-p 'org-mode)
        (org-display-inline-images)))))

(my/setup-demo-org)
```

4 Section 1: Overview

FigRecipe MCP Server enables declarative figure creation via Claude Code.

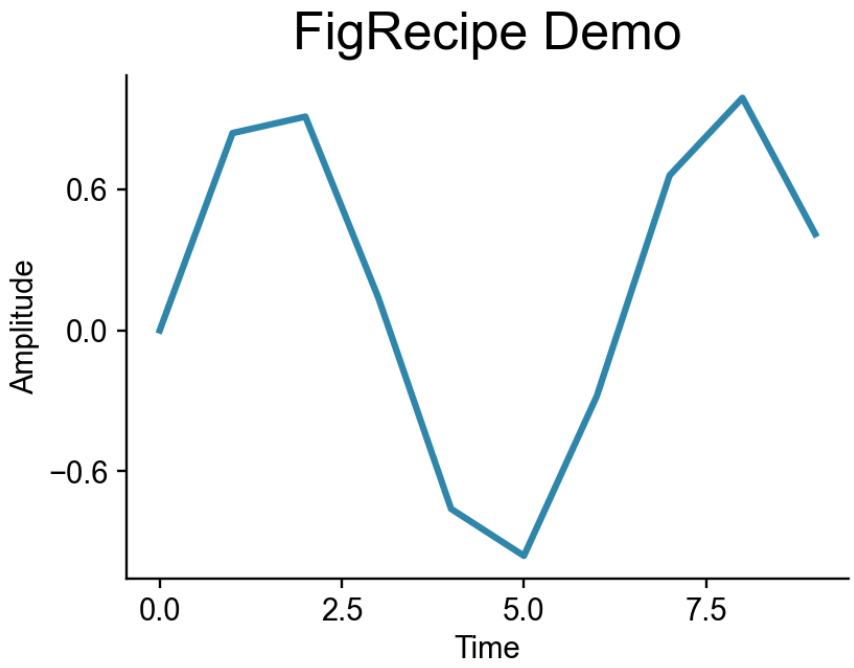
4.1 Available Plot Types (27 types in 7 categories)

Category	Plot Types
Line/Curve	line, plot, step, fill, $\text{fill}_{\text{between}}$, errorbar
Scatter	scatter
Bar	bar, barh
Distribution	hist, hist2d, boxplot, violinplot
Image/Matrix	imshow, matshow, heatmap, pcolormesh
Contour	contour, contourf
Special	pie, stem, eventplot, hexbin

4.2 Available Styles

- MATPLOTLIB (default)
- SCITEX (publication-ready)

4.3 Simple Example

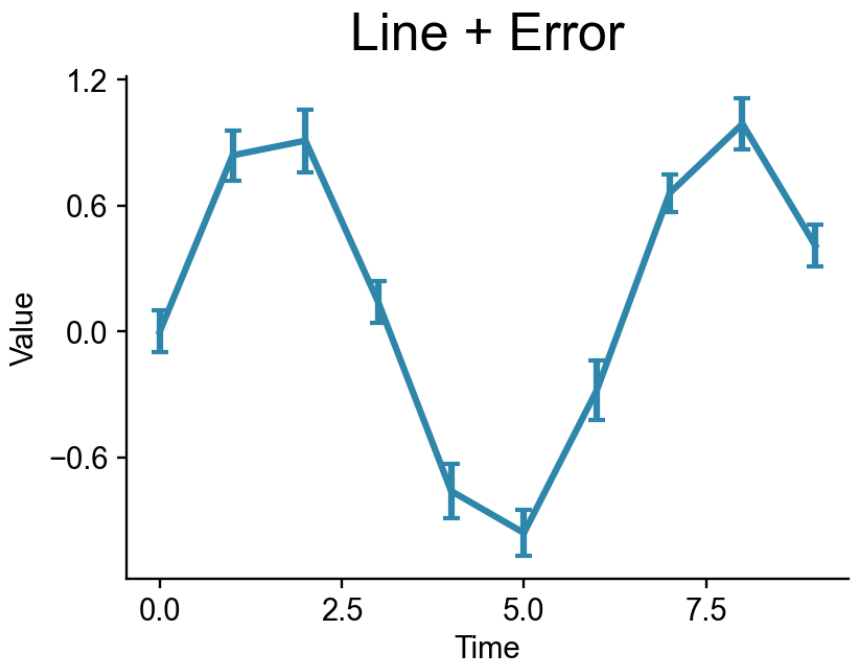


5 Section 2: Types of Plots

Demonstrating different plot types available in FigRecipe MCP Server.

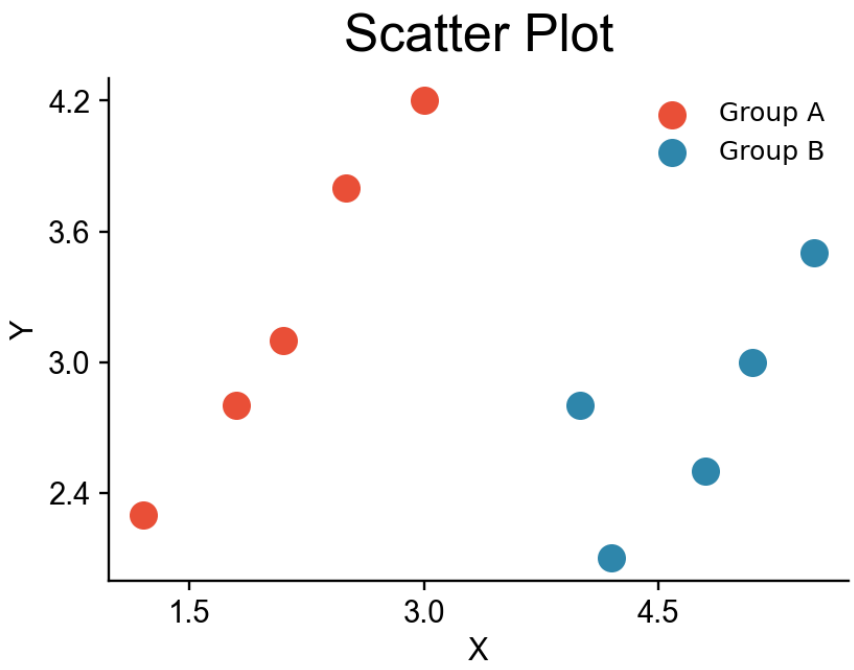
5.1 Line Plot with Error Bars

```
plots:
  - type: errorbar
    x: [0, 1, 2, ...]
    y: [values]
    yerr: [errors]
```



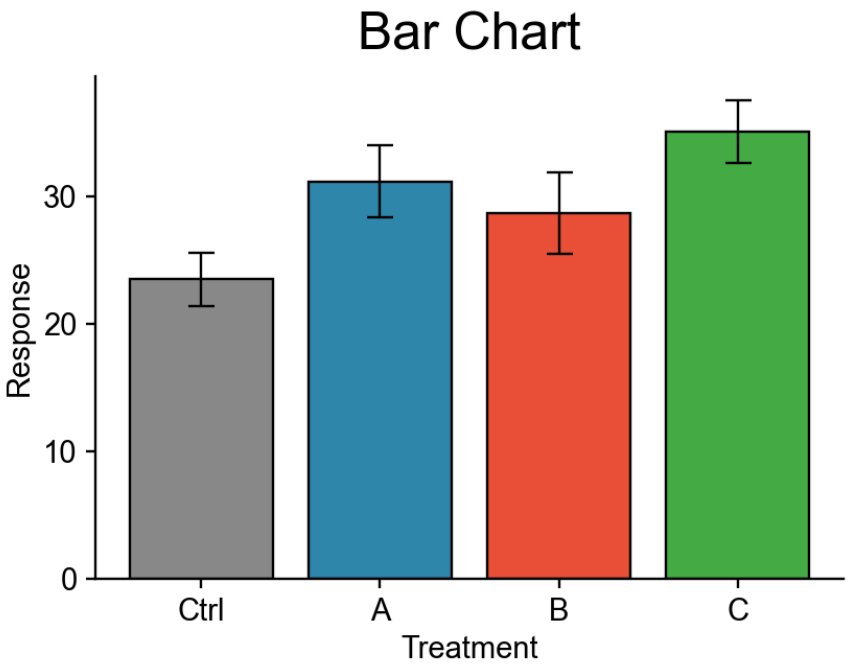
5.2 Scatter Plot

```
plots:
  - type: scatter
    x: [x_values]
    y: [y_values]
    color: "#E94F37"
```



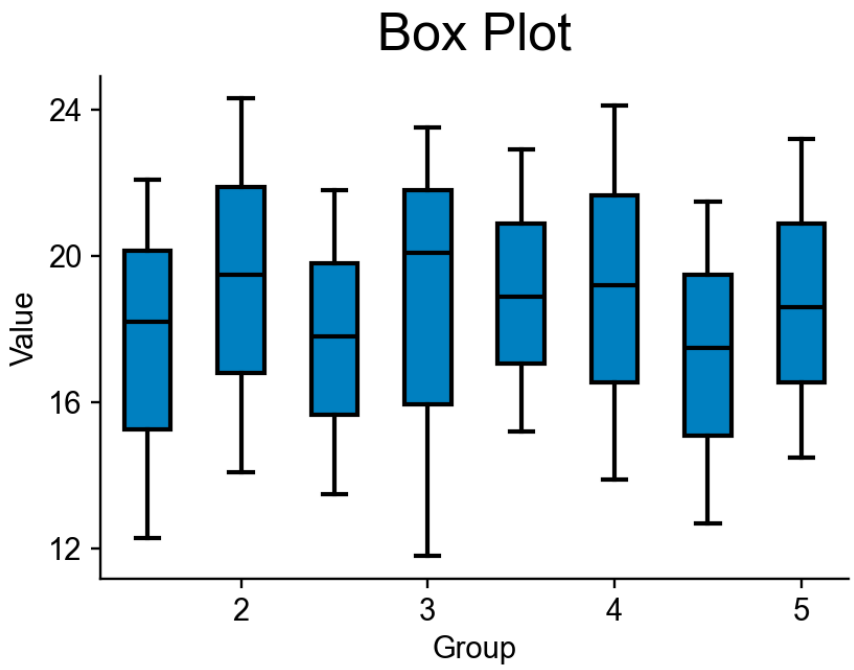
5.3 Bar Chart

```
plots:
  - type: bar
    x: ["Ctrl", "A", "B", "C"]
    height: [23.5, 31.2, 28.7, 35.1]
    yerr: [2.1, 2.8, 3.2, 2.5]
```



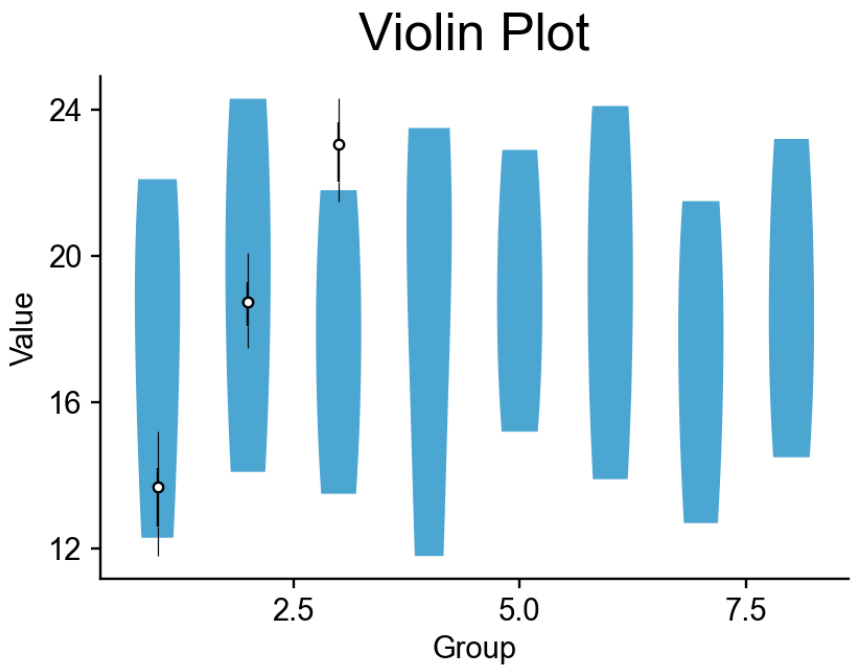
5.4 Box Plot

```
plots:
  - type: boxplot
    data: [[group1], [group2], [group3]]
```



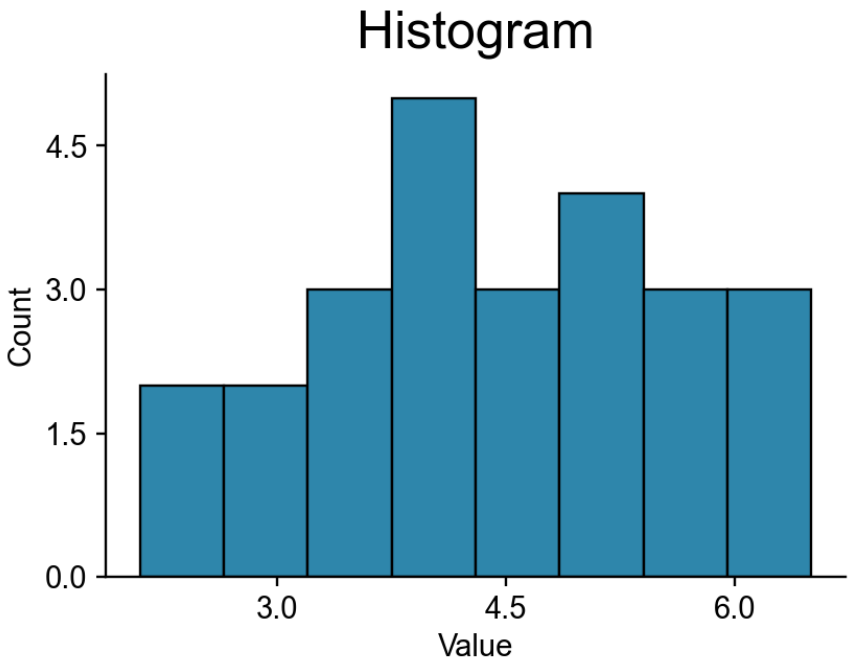
5.5 Violin Plot

```
plots:
  - type: violinplot
    data: [[group1], [group2], [group3]]
```



5.6 Histogram

```
plots:
  - type: hist
    x: [values]
    bins: 8
```



6 Section 3: Recipe Contents and File Structure

6.1 Directory Structure

When you create figures, FigRecipe automatically saves:

- PNG/PDF image file
- YAML recipe file (for reproduction)
- Data directory with CSV files for each plot

```
demo_mcp_out/
 data/                    # Source data files
    barplot.csv
    groups.csv
    timeseries.csv
 figures/
    01_overview.png      # Image file
    01_overview.yaml     # Recipe file
    01_overview_data/    # Extracted plot data
```

```
        plot_000_x.csv
        plot_000_y.csv
 recipes/
```

6.2 Recipe YAML Structure

A recipe file contains:

1. **Figure metadata**: ID, creation timestamp, matplotlib version
2. **Figure settings**: Size (mm), DPI, style configuration
3. **Axes data**: Plot calls with function names, args, kwargs
4. **Decorations**: Labels, titles, legend settings

```yaml
figrecipe: '1.0'
id: fig_3e964862
created: '2026-01-16T05:58:25'
figure:
  figsize: [2.76, 2.17]  # inches (from mm conversion)
  dpi: 300
  style: {...}           # SCITEX style parameters
axes:
  ax_0_0:
    calls:
      - function: errorbar
        args: [{data: errorbar_000_x.csv}, ...]
        kwargs: {color: '#2E86AB', capsize: 2}
    decorations:
      - function: set_xlabel
        args: [{data: "Time"}]
```

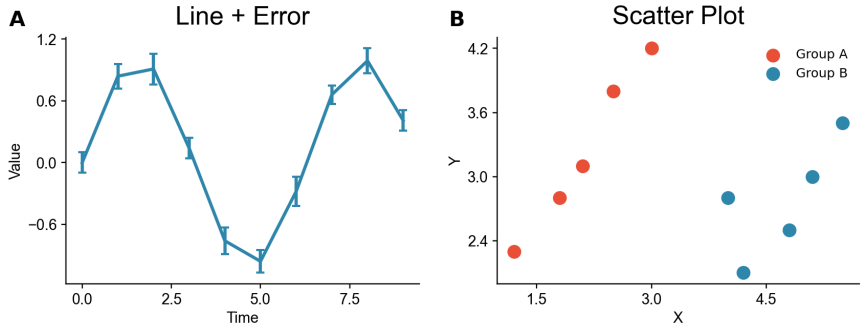
7 Section 4: Panel Composition

FigRecipe can compose multiple figures into a single multi-panel figure.

7.1 Horizontal Layout (2 panels)

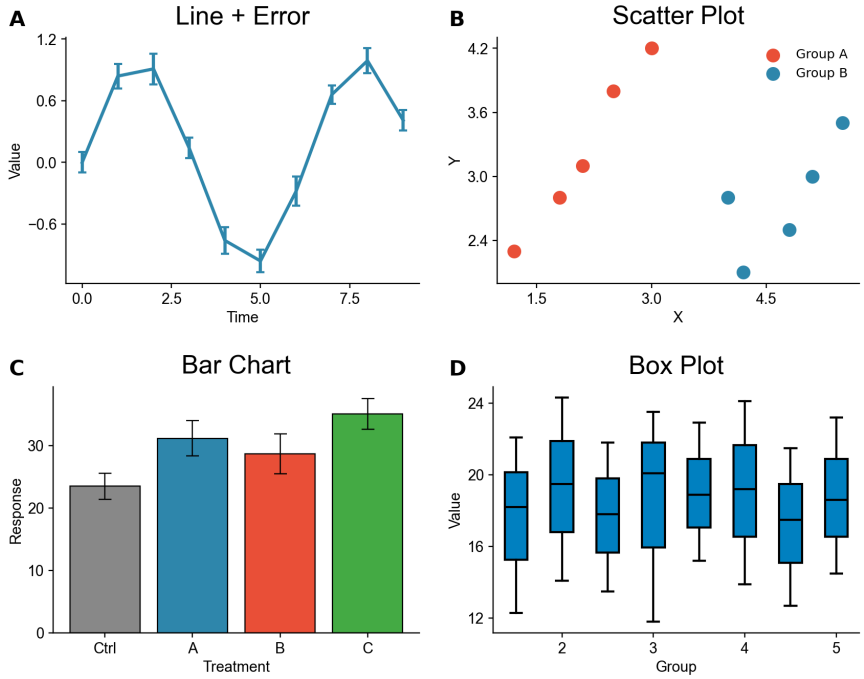
```yaml
compose:
  sources: [figure1.png, figure2.png]
  layout: horizontal
```

```
  gap_mm: 3
  panel_labels: true
  label_style: uppercase  # A, B, C...
```



7.2 Grid Layout (2x2 panels)

```
compose:
  sources: [fig1.png, fig2.png, fig3.png, fig4.png]
  layout: grid
  gap_mm: 3
  panel_labels: true
```



7.3 Layout Options

Layout	Description
horizontal	Figures side by side
vertical	Figures stacked vertically
grid	Automatic grid arrangement

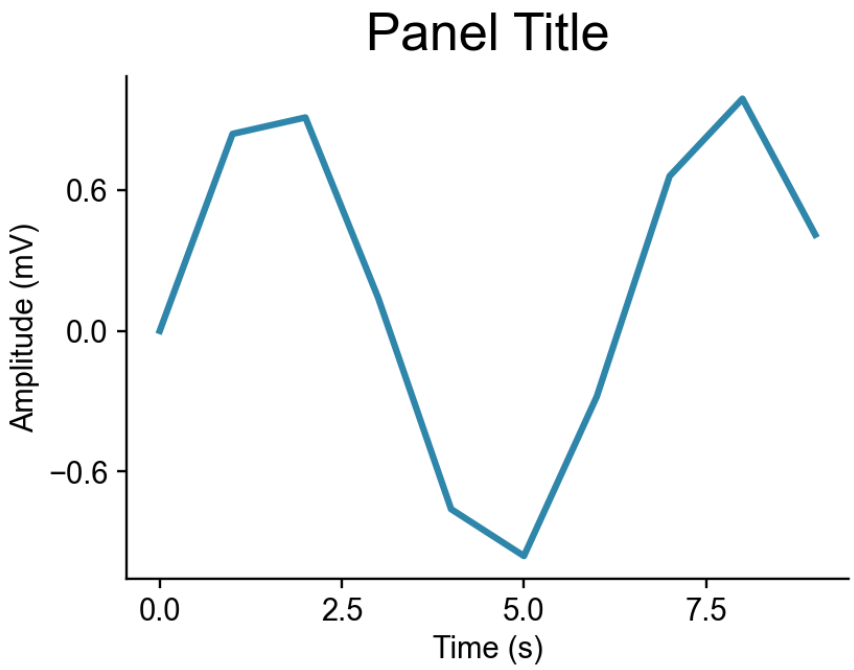
7.4 Label Styles

- `uppercase`: A, B, C, D
- `lowercase`: a, b, c, d
- `numeric`: 1, 2, 3, 4

8 Section 5: Figure and Panel-Level Captions

8.1 Single Figure with Title

```
figure:
  width_mm: 40
  height_mm: 28
xlabel: "Time (s)"
ylabel: "Amplitude (mV)"
title: "Panel Title"        # Panel-level title
suptitle: "Figure Title"    # Figure-level title
```



8.2 Composed Figure with Caption

The `compose` tool supports adding figure captions below the panels:

```
compose:
  sources: [fig1.png, fig2.png]
  layout: horizontal
  panel_labels: true
  caption: "Figure 1. Description of panels (A) and (B)..."
```

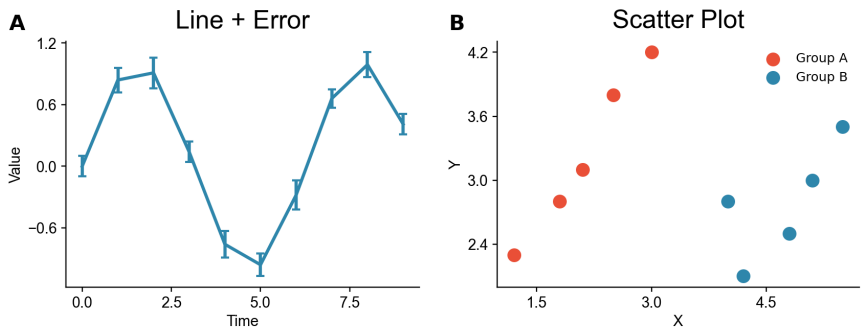


Figure 1. Demonstration of composed figures with caption. (A) Line plot with error bars showing temporal signal variation. (B) Scatter plot showing group distributions.

8.3 Caption Hierarchy

Level	Parameter	Description
Figure	suptitle	Overall figure title
Panel	title	Individual panel title
Composition	caption	Caption below composed fig

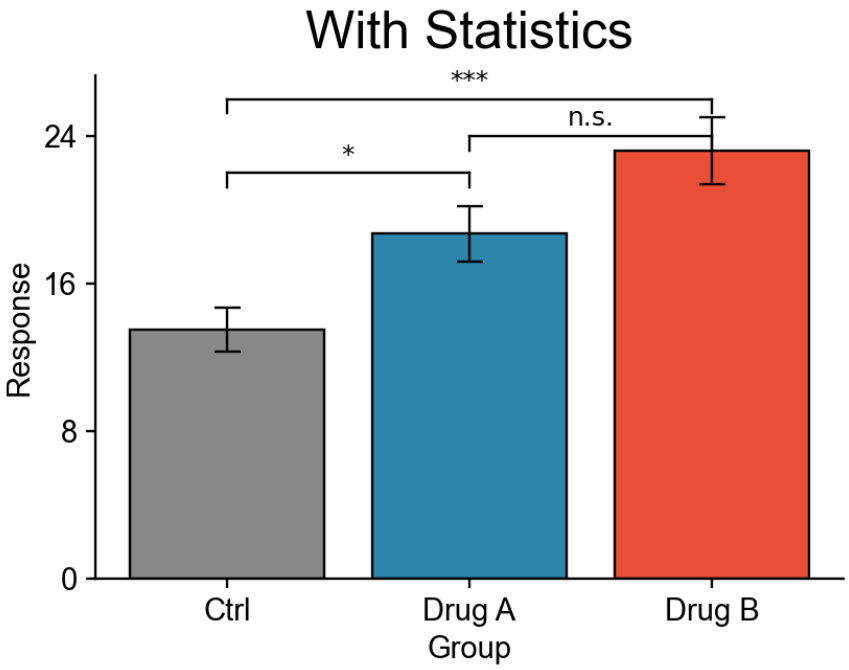
9 Section 6: Statistical Annotations

Add significance brackets and p-value indicators to your figures.

9.1 Syntax

```
stat_annotations:
  - x1: 0           # First bar/group index
    x2: 1           # Second bar/group index
    y: 22           # Y position of bracket
    text: "*"       # Significance text
  - x1: 0
    x2: 2
    y: 26
    text: "***"
  - x1: 1
    x2: 2
    y: 24
    text: "n.s."    # Not significant
```

9.2 Example with Significance Brackets



9.3 Common Significance Notations

Symbol	Meaning	P-value
n.s.	Not significant	$p > 0.05$
*	Significant	$p < 0.05$
**	Very significant	$p < 0.01$
***	Highly signif.	$p < 0.001$

10 Section 7: Reproduction

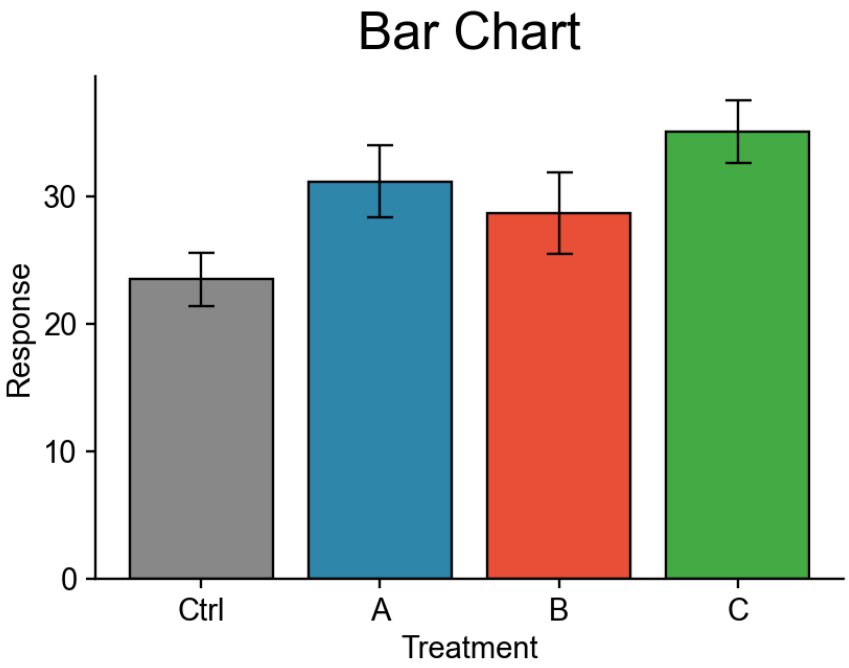
The key feature of FigRecipe is reproducibility - recreate any figure from its recipe.

10.1 Reproduce from Recipe

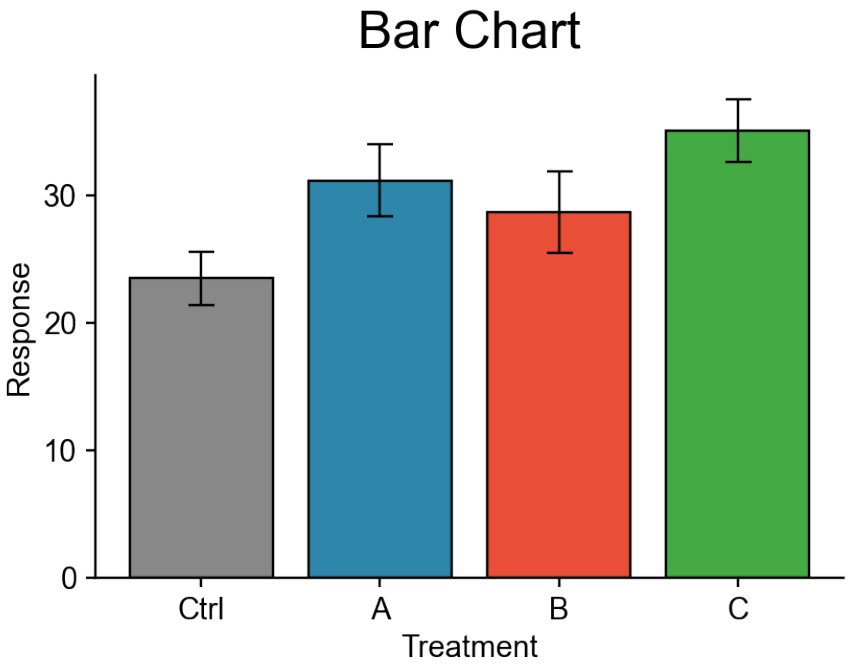
```
reproduce:
  recipe_path: "figures/02c_bar.yaml"
  output_path: "figures/07_reproduced.png"
  format: png
```

```
dpi: 300
```

10.1.1 Original Figure



10.1.2 Reproduced Figure



10.2 Validate Reproducibility

The `validate` tool checks if a recipe produces consistent output:

```
validate:
  recipe_path: "figures/02c_bar.yaml"
  mse_threshold: 100  # Maximum acceptable error
```

Result: `{"valid": true, "mse": 0, "message": "Recipe produces consistent output"}`

10.3 Extract Data from Recipe

Retrieve plotted data arrays from a saved recipe:

```
extract_data:
  recipe_path: "figures/02c_bar.yaml"
```

Result:

```
{
  "bar_000": {
    "x": ["Ctrl", "A", "B", "C"],
    "yerr": [2.1, 2.8, 3.2, 2.5]
  }
}
```

10.4 MCP Tools Summary

Tool	Description
plot	Create figure from spec
reproduce	Recreate figure from recipe
compose	Combine multiple figures
info	Get recipe information
validate	Check reproducibility
crop	Auto-crop whitespace
$\text{extract}_{\text{data}}$	Get plotted data from recipe
$\text{list}_{\text{styles}}$	List available style presets
$\text{get}_{\text{plottypes}}$	List supported plot types

11 Demo Complete

This demonstration covered all major FigRecipe MCP Server capabilities:

1. Overview and plot types
2. Recipe file structure
3. Panel composition
4. Captions and labels
5. Statistical annotations
6. Reproducibility

All artifacts are stored in `./demo_mcp_out/`.