Visualizing Your Data: Creating Effective Plots with ggplot2 in R

Thursday, February 5, 2026

4:00pm - 5:00pm **(Online)**

Sherman Centre for Digital Scholarship

Land Acknowledgement

McMaster University is located on the traditional Territories of the Mississauga and Haudenosaunee Nations, and within the lands protected by the “Dish With One Spoon” wampum agreement.

[Consider customizing this land acknowledgement. An example has been included in the notes below, courtesy of Danica Evering.]

McMaster's anities and ity Libraries | McMaster University | Sherman Centre for Digital Scholarship

Code of Conduct

The Sherman Centre and the McMaster University Libraries are committed to fostering a supportive and inclusive environment for its presenters and participants.

As a participant in this session, you agree to support and help cultivate an experience that is collaborative, respectful, and inclusive, as well as free of harassment, discrimination, and oppression. We reserve the right to remove participants who exhibit harassing, malicious, or persistently disruptive behaviour.

Please refer to our code of conduct webpage for more information: **scds.ca/code-of-conduct/**

Certificate Programs

The Sherman Centre for Digital Scholarship Certificate of Attendance

The Sherman Centre's certificate program recognizes attendance at our workshops. It complements degree training, supports the development of critical competencies in data analysis, research data management, and digital scholarship, and formalizes core skills fostered by our workshops.

Participants are invited to attend seven workshops and receive a certificate of attendance. To verify your participation in today's workshop, we will provide a code and additional instructions at the end of the session.

You can learn more about the certificate program at **scds.ca/certificate-program**

The Canadian Certificate for Digital Humanities

This workshop is also eligible for the Canadian Certificate for Digital Humanities. To learn more about the certificate, visit **ccdhhn.ca**. You can also contact local liaison Alexis-Carlota Cochrane at scds@mcmaster.ca

Winter 2026: Upcoming Workshops

Data Analysis Support Hub

Feb 10: Finding, Accessing, and Adding GIS Data to Your Project

Feb 12: Introduction to Python Programming

March 10: Create an Interactive Dashboard using ArcGIS

Digital Research

February 11: Visualizing Bibliometric Networks with VOSviewer

Research Data Management

February 19: Communities Empowered by Data 101: Tools and Best Practices

May 12: Data Management Plan Bootcamp (In-Person)

May 19: Data Deposit Bootcamp (In-Person)

Do More with Digital Scholarship

February 6: Create a Digital Exhibition with Omeka S

February 9: Rethinking "Good" Data: Power, Vulnerability, and Queer Data Care

scds.ca
scds@mcmaster.ca

A joint initiative of McMaster's Faculty of Humanities and McMaster University Libraries

Register for Upcoming Workshops: https://u.mcmaster.ca/scds-workshops

Book an Appointment with the DASH Team

Receive help from a member of the DASH team! DASH can assist with the following topics:

- Creating data visualizations, including charts, graphs, and scatter plots
- Figuring out which statistical tests to run (e.g., t-test, chi-square, etc.).
- Analyzing data with software including SPSS, Python, R, SAS, ArcGIS, MATLAB, and Excel
- Choosing which software package to use, including free and open-source software
- Troubleshooting problems related to file formats, data retrieval, and download
- Selecting methodology and type of data analysis to use in a thesis project

Book an appointment: **https://library.mcmaster.ca/services/dash**

Session Recording and Privacy

This session is being recorded with the intention of being shared publicly via the web for future audiences. In respect of your privacy, participant lists will not be shared outside of this session, nor will question or chat transcripts.

Questions asked via the chat box will be read by the facilitator without identifying you. Note that you may be identifiable when asking a question during the session in an audio or visual format.

Data Visualization

Using R (ggplot2)

Sahar Khademioore

PhD Candidate in Health Research Methodology

McMaster University

Sherman Centre for Digital Scholarship | DASH Workshop Series

Workshop Objectives

By the end of this session, you will be able to:

Understand why we visualize data

Learn the purpose behind visualization and when charts reveal what numbers cannot.

Recognize common chart types

Identify line, bar, scatter, pie, histogram, and box plots — and know when to use each.

Learn the Grammar of Graphics

Understand the layered framework that powers ggplot2: data, aesthetics, geometries, and more.

Create plots with ggplot2

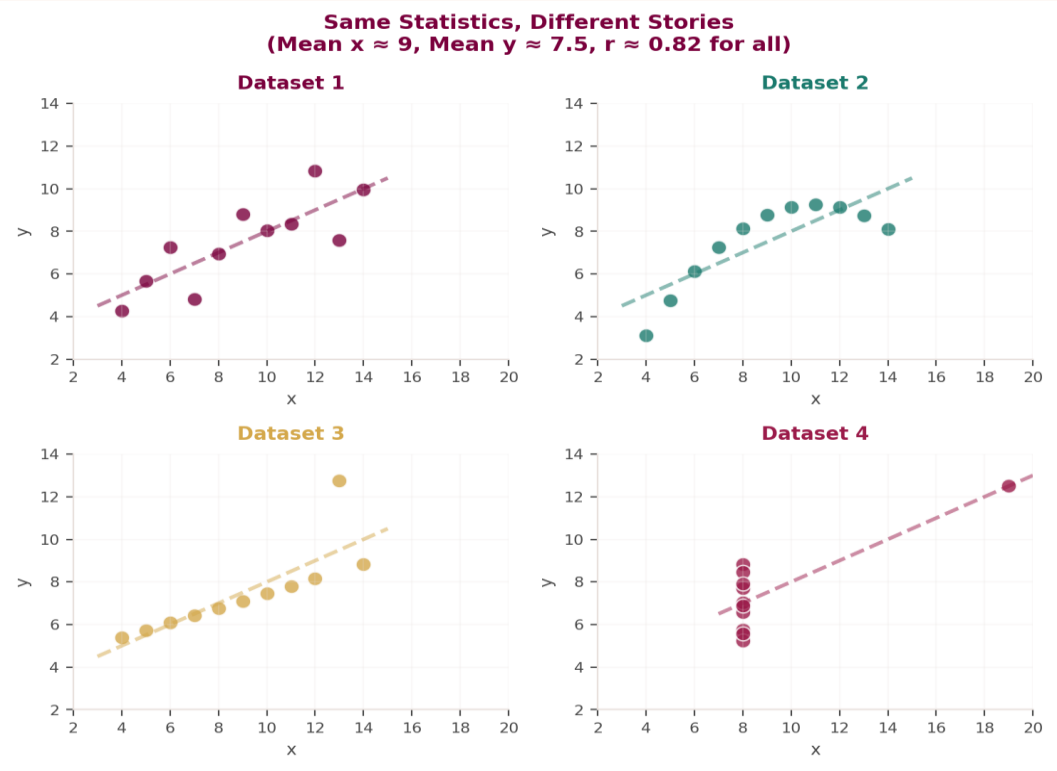
Write R code to produce publication-quality visualizations step by step.

Why Visualize Data?

Turning numbers into understanding

The Power of Visualization

Why looking at your data matters more than summary statistics



Anscombe's Quartet

All four datasets have:

- Same mean of x (9.0)
- Same mean of y (7.5)
- Same correlation (0.82)
- Same regression line

Yet they look completely different!

This is why you should always visualize your data — summary statistics alone can be misleading.

What Can Visualization Do?

Show trends

How has hospital admission rate changed over 10 years?

Reveal relationships

Is there a connection between BMI and blood pressure?

Compare groups

Which treatment group had better outcomes?

Show composition

What percentage of patients fall in each age category?

Display distributions

How are test scores spread across the class?

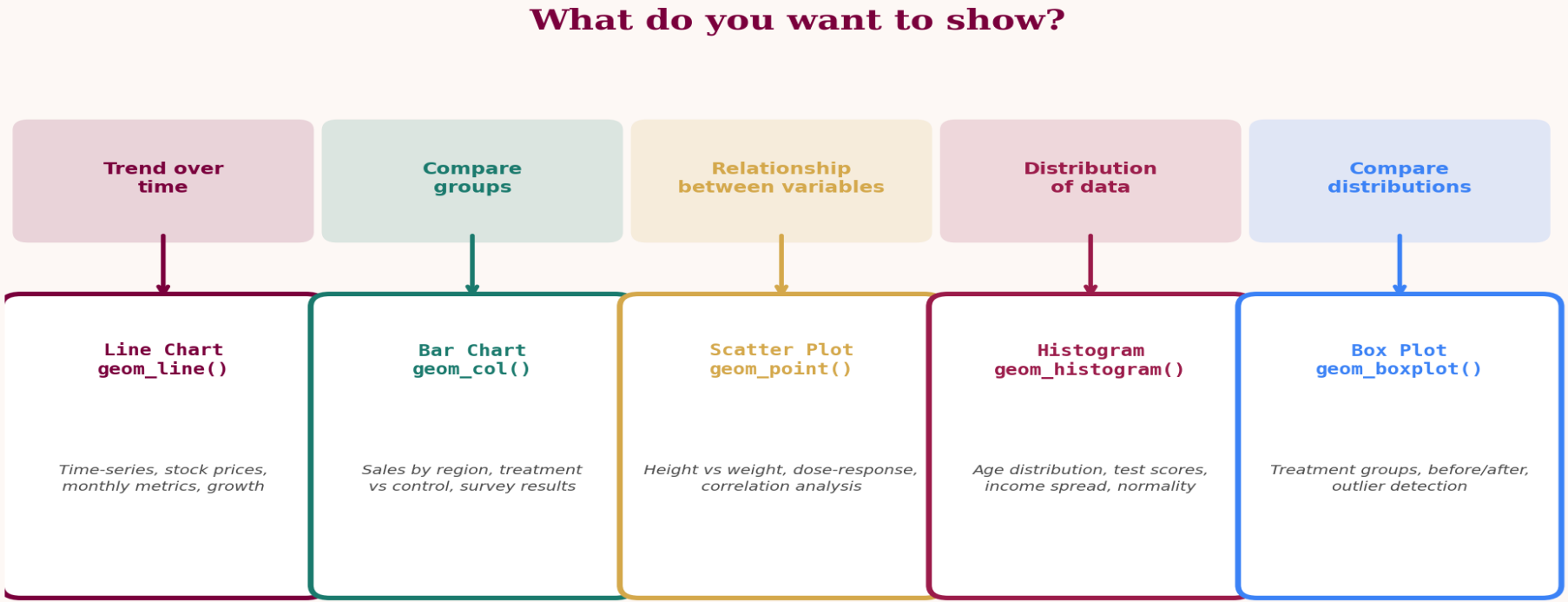
A good visualization can communicate in seconds what might take paragraphs to explain in words.

Common Chart Types

Choosing the right visualization for your data

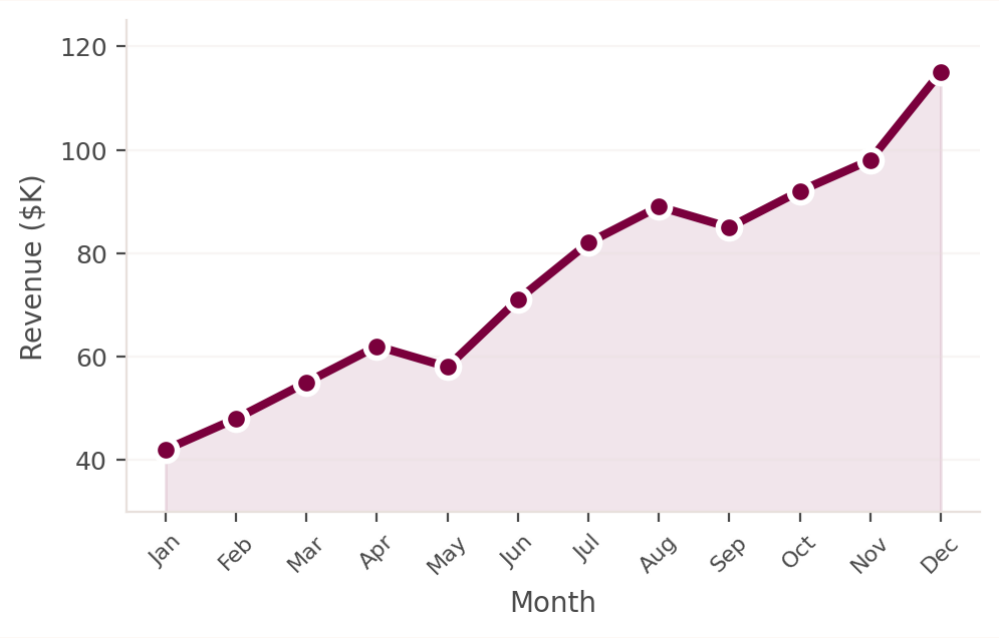
Choosing the Right Chart

Ask yourself: "What do I want to show?" — then pick accordingly



Line Charts

Best for showing trends over time or continuous sequences



When to use:

- Time-series data (monthly revenue, daily temps)
- Tracking metrics over sequential intervals
- Comparing multiple trends on same timeline

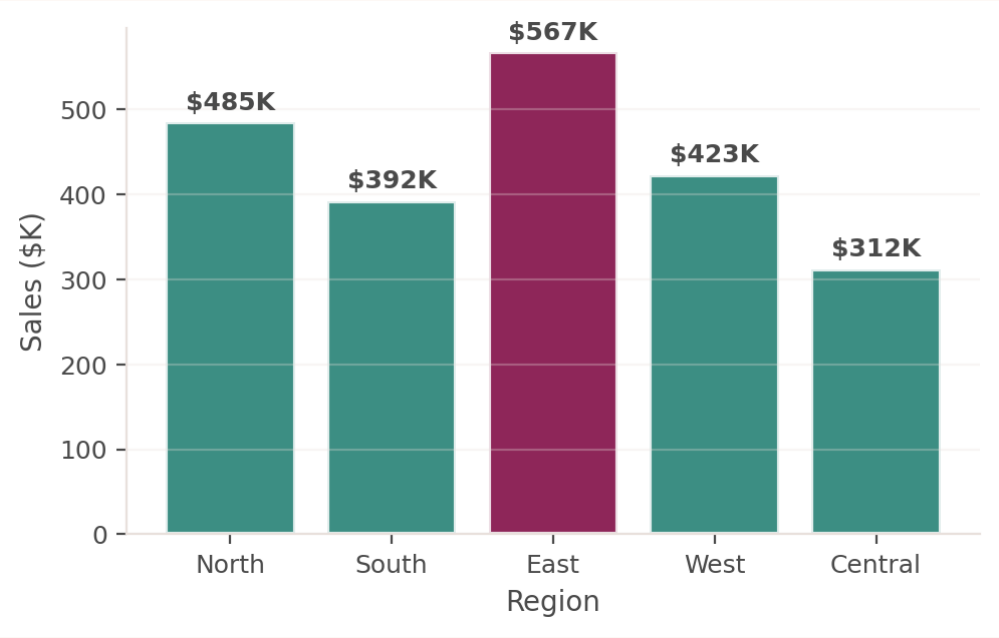
Avoid when:

- X-axis is categorical with no natural order
- You have very few data points (< 3)

```
ggplot(data, aes(x = month, y = revenue))
+
  geom_line(color = "#7A003C", linewidth
= 1) +
  geom_point(color = "#7A003C", size = 2)
```

Bar Charts

Perfect for comparing values across discrete categories



When to use:

- Comparing quantities across categories
- Showing ranked or ordered values
- Survey results, group comparisons

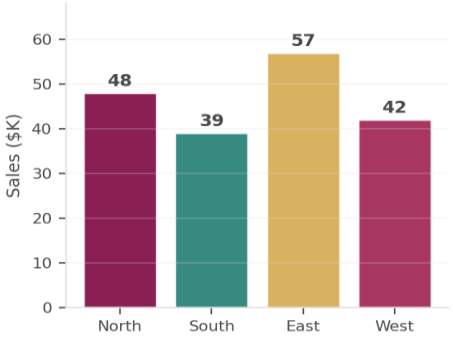
Variations:

- Grouped bars → compare sub-categories
- Stacked bars → show composition within
- Horizontal bars → long category labels

```
ggplot(data, aes(x = region, y = sales))
+
  geom_col(fill = "#1A7A6D")
```

Types of Bar Charts

Four variations of bar charts and when to use each one



Simple (Vertical) Bar

Compare values across categories

```
geom_col(fill = color)
```

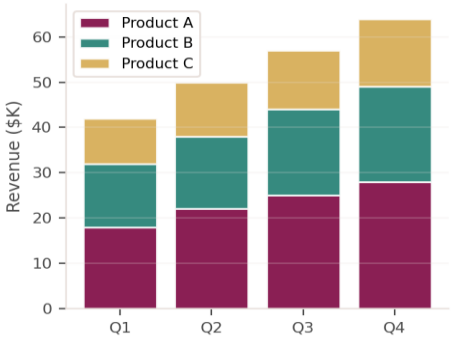
Ranking, survey results, single-variable comparisons

Grouped Bar

Compare sub-groups side by side

```
geom_col(position =
"dodge")
```

Year-over-year, treatment vs. control by category

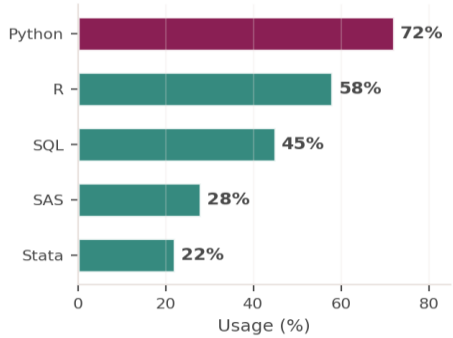


Stacked Bar

Show composition within totals

```
geom_col(position =
"stack")
```

Budget breakdown, market share over time periods



Horizontal Bar

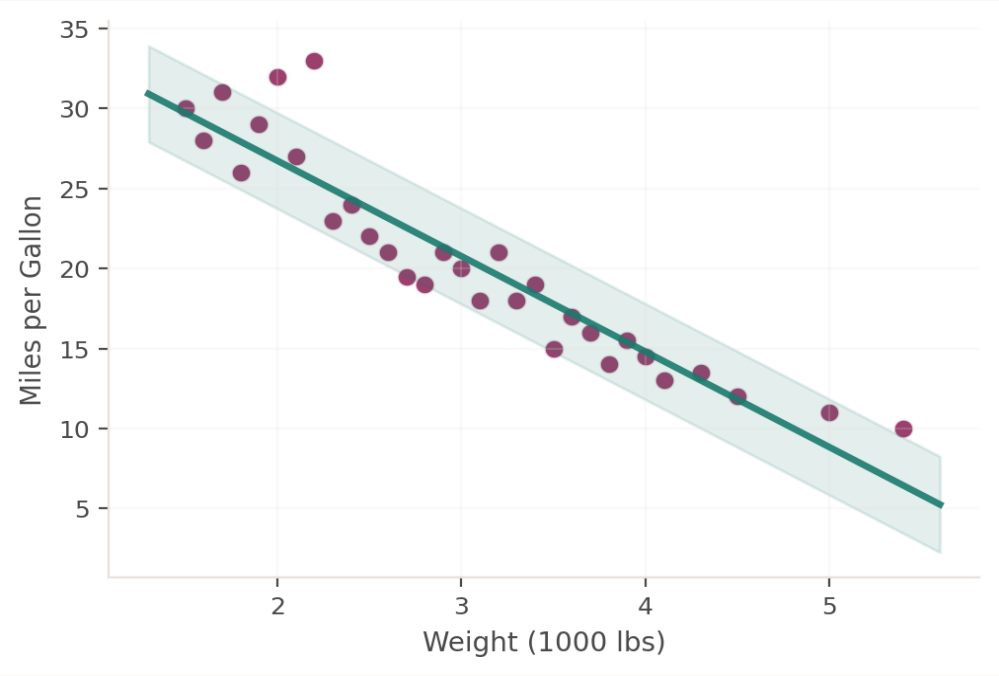
Best for long category labels

```
geom_col() +
coord_flip()
```

Top-10 lists, names, long text categories

Scatter Plots

Reveals relationships between two continuous variables



When to use:

- Two continuous numeric variables
- Checking for correlation or clusters
- Identifying outliers in data
- Dose-response, height vs. weight

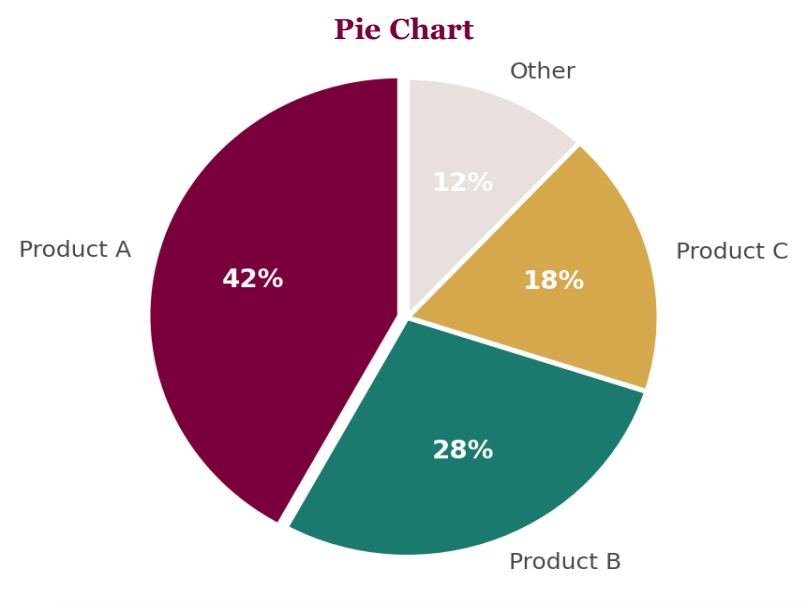
Enhance with:

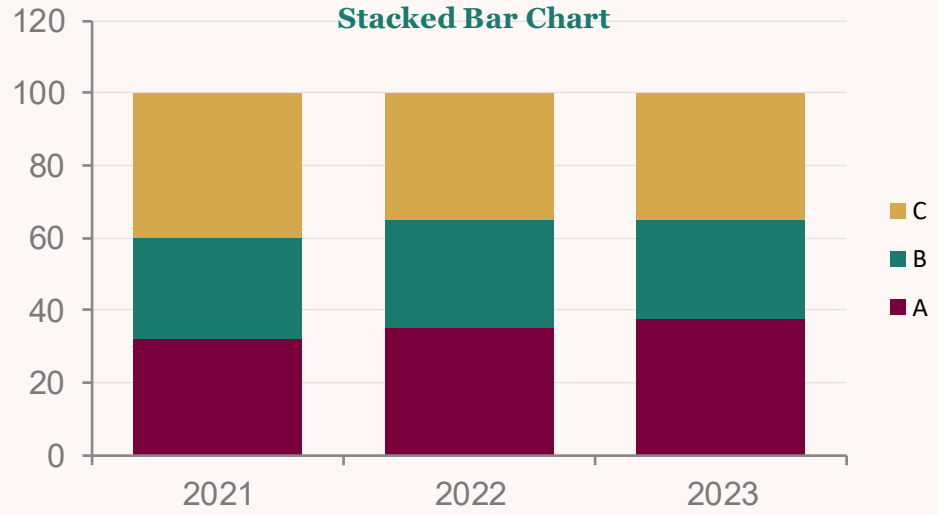
- geom_smooth() for trend lines
- Color/size aesthetics for extra variables

```
ggplot(data, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm")
```

Part-to-Whole Charts

Showing how parts make up a whole

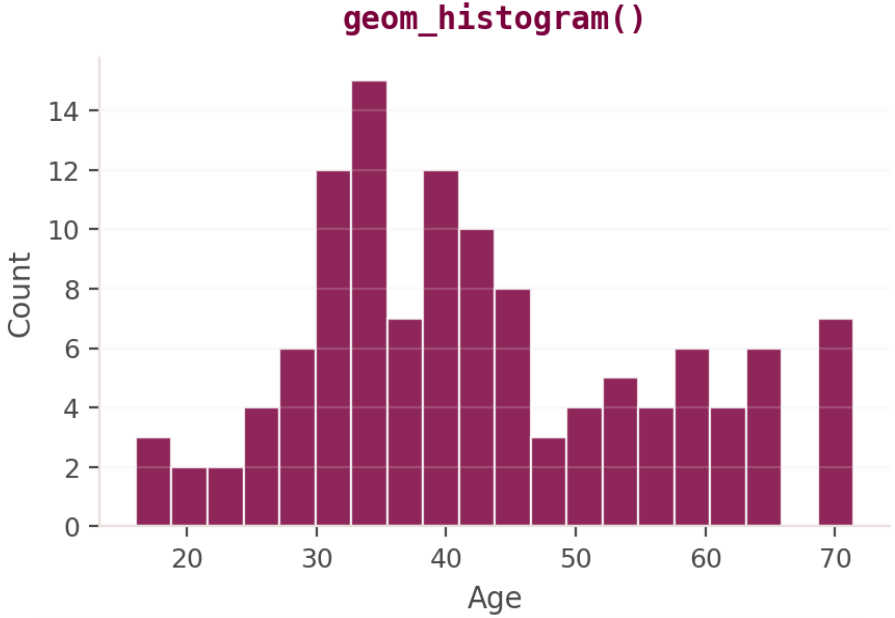


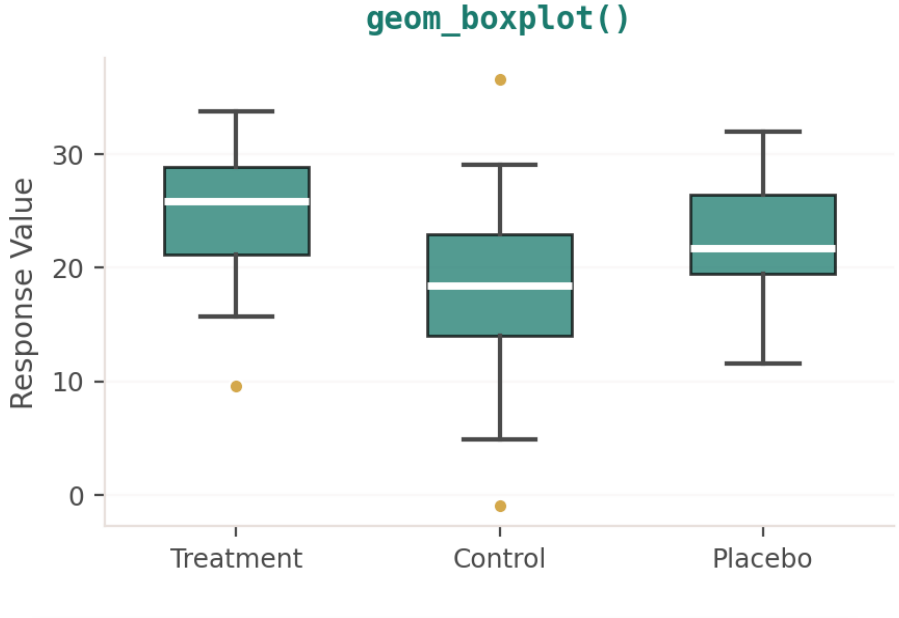


Tip: Pie charts work best with 2-5 slices. For more categories, comparing across time, or precise comparisons, use a stacked or grouped bar chart instead.

Distribution Plots

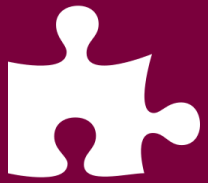
Understanding how your data is spread





Histogram: Explore the shape of one continuous variable. Is it normal, skewed, bimodal? Adjust bins to find the right detail level.

Box Plot: Compare distributions across groups. Shows median, quartiles, and outliers. Great for treatment vs. control.



The Grammar of Graphics

A layered approach to building visualizations

What Is the Grammar of Graphics?

Just like sentences have a grammar (subject, verb, object), data visualizations have a grammar too. Developed by Leland Wilkinson, it breaks every chart into modular building blocks.

Data

Your dataset

Aesthetics

Map to axes, color, size

Geometries

Points, bars, lines

Facets

Split into sub-plots

Statistics

Summarize/transform

Coordinates

Cartesian, polar, etc.

Themes

Fonts, colors, style

Think of it like building with LEGO blocks — each piece has a specific role, and you snap them together to build your chart.

What Is ggplot2?

ggplot2 is an R package that implements the Grammar of Graphics. Created by Hadley Wickham, it is the most popular visualization tool in R and one of the most powerful plotting libraries in any language.

The basic ggplot2 formula:

```
ggplot(data = my_data, aes(x = var1, y = var2)) +
  geom_point()    # Add a layer of points
  # + more layers, facets, themes...
```

`ggplot()`	Initializes the plot and sets the data source.
`aes()`	Maps variables to visual properties (axes, color, size, shape).
`geom_*()`	Adds visual layers — points, lines, bars, etc. Stack multiple geoms!
`+ operator`	Layers are added with +. Each addition builds on the previous.

Layer 1: Data

Every visualization starts with a data frame — a table of rows and columns

Example: mtcars dataset

mpg	cyl	wt	hp
21.0	6	2.62	110
22.8	4	2.32	93
18.7	8	3.44	175
21.4	6	2.78	110
14.3	8	3.57	245

```
ggplot(data = mtcars)
```

Key points:

- Each row = one observation
- Each column = one variable

R has many built-in datasets you can practice with: mtcars, iris, diamonds, airquality, etc.

Use head(mtcars) to preview the first 6 rows of any data frame.

At this stage, ggplot(data = mtcars) creates an empty canvas. No visual elements appear until you add a geom!

Layer 2: Aesthetics (aes)

Map your data variables to visual properties

Aesthetic	What it does	Example
x	Position on x-axis	`aes(x = wt)`
y	Position on y-axis	`aes(y = mpg)`
color	Color of points/lines	`aes(color = cyl)`
fill	Fill color of bars/areas	`aes(fill = cyl)`
size	Size of points	`aes(size = hp)`
shape	Shape of points	`aes(shape = gear)`

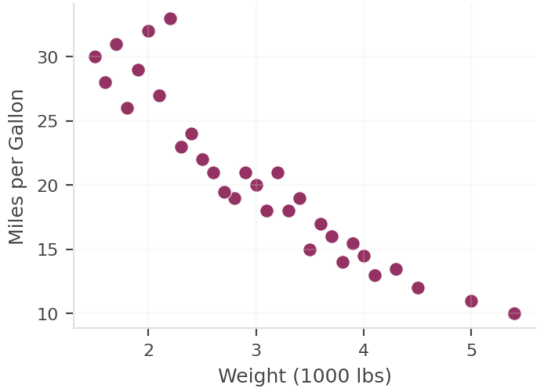
```
ggplot(data = mtcars, aes(x = wt, y = mpg, color = factor(cyl)))
```

Place aes() inside ggplot() for global mappings that apply to all layers, or inside a specific geom for layer-specific mappings.

Aesthetics in Action

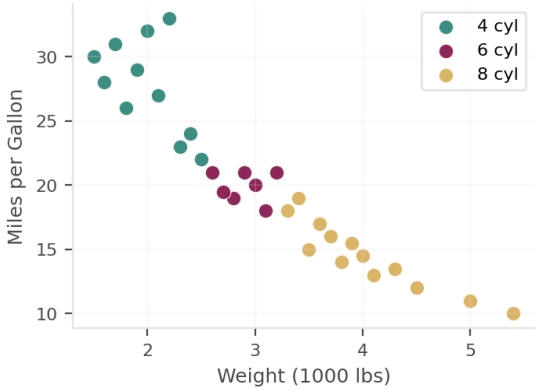
The same data tells different stories depending on which aesthetics you map

`aes(x, y) only`



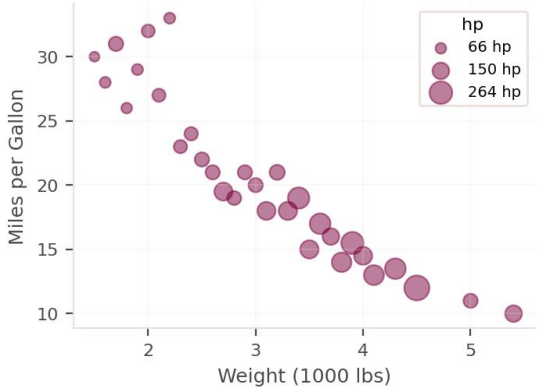
Just position — all points identical

`aes(x, y, color = cyl)`



Color reveals groups in the data

`aes(x, y, size = hp)`



Size encodes a third variable (hp)

Same data, same x and y — but different aesthetics reveal hidden patterns. This is the power of the grammar!

Layer 3: Geometries (geom)

Geometries define the visual marks — this is where you choose the chart type

Continuous Variables

- `geom_point()` – scatter
- `geom_line()` – line plot
- `geom_smooth()` – trend line
- `geom_area()` – area chart

Categorical Variables

- `geom_bar()` – bar (counts)
- `geom_col()` – bar (values)
- `geom_boxplot()` – box plot
- `geom_violin()` – violin plot

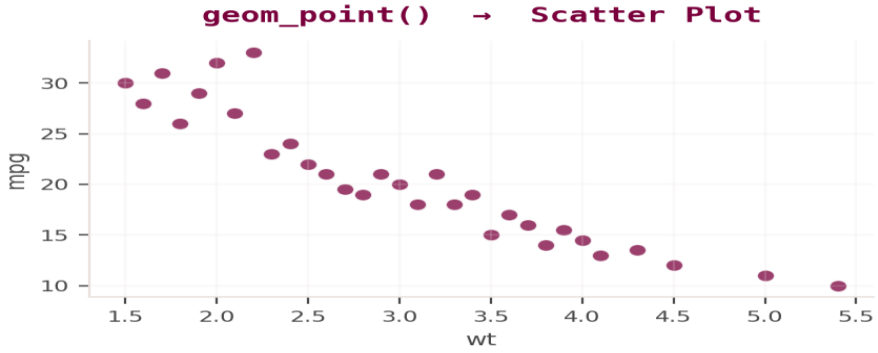
Distributions

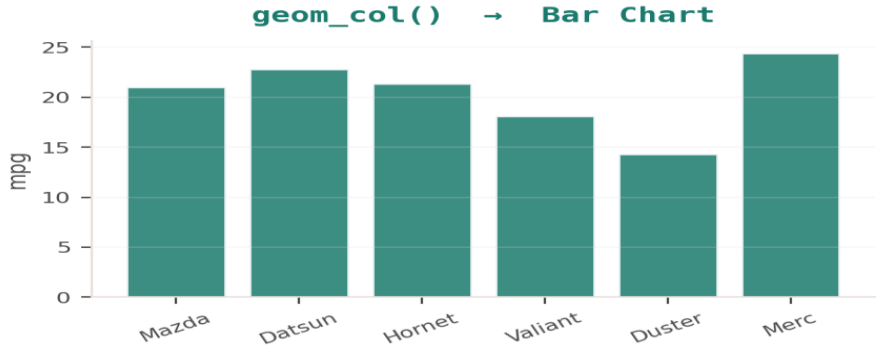
- `geom_histogram()` – histogram
- `geom_density()` – density curve
- `geom_freqpoly()` – freq polygon
- `geom_qq()` – Q-Q plot

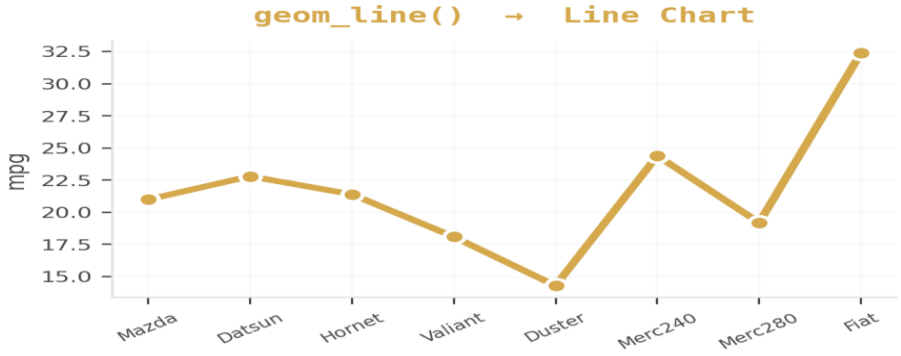
```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm")
```

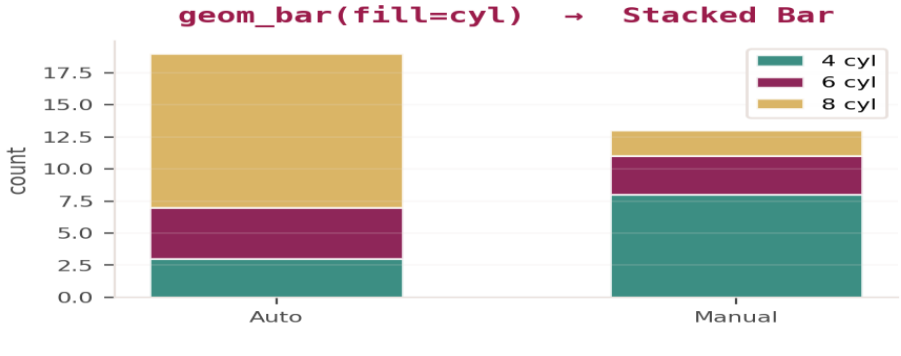
Geometries Gallery

Changing the geom changes the entire story your chart tells









Building Up: Combining Layers

The power of ggplot2 lies in stacking layers — each "+" adds a new element

Step 1: Data + aesthetics

```
ggplot(mtcars, aes(x = wt, y = mpg))
```

Step 2: Add points

```
  + geom_point()
```

Step 3: Add trend line

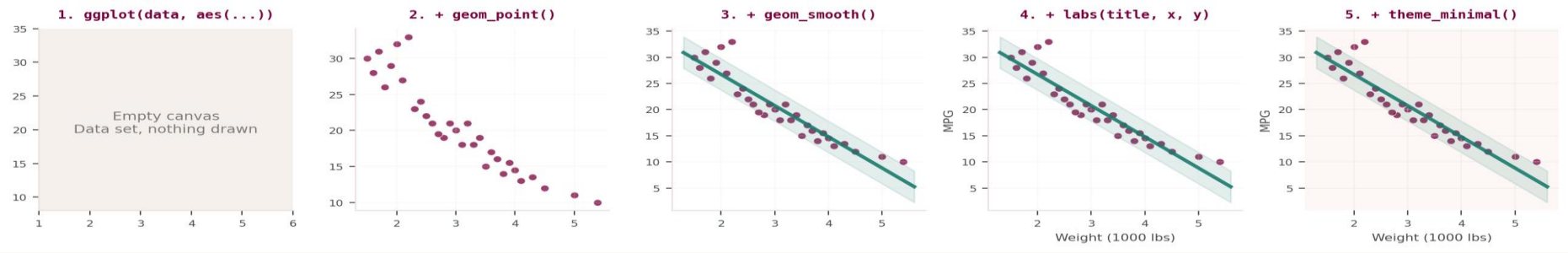
```
  + geom_smooth(method = "lm")
```

Step 4: Add labels

```
  + labs(title = "MPG vs Weight", x = "Weight", y = "MPG")
```

Step 5: Apply theme

```
  + theme_minimal()
```



Facets: Small Multiples

Split your plot into sub-plots based on a categorical variable

`facet_wrap()`

Wraps panels into rows/columns automatically.

Use for one grouping variable.

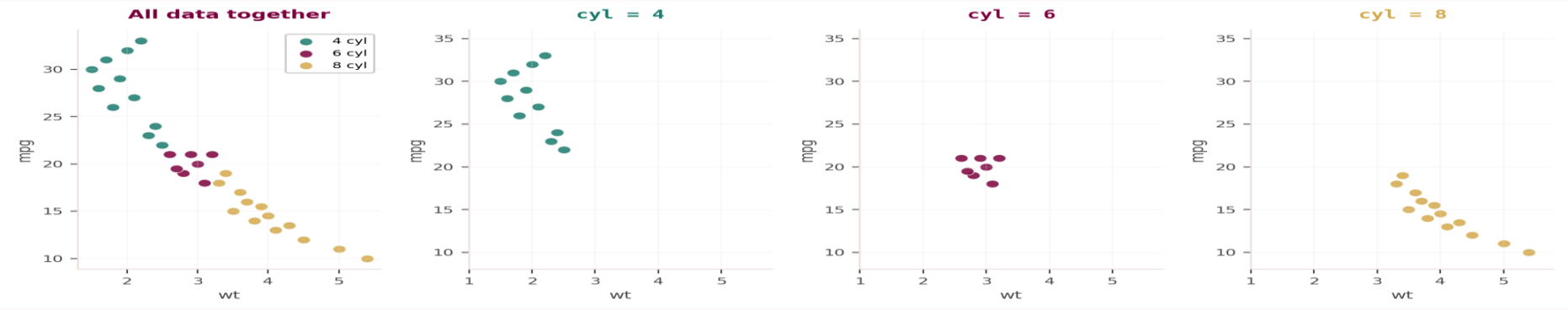
```
facet_wrap(~cyl)
```

`facet_grid()`

Creates a grid using two variables.

Use for comparing across two dimensions.

```
facet_grid(gear ~ cyl)
```

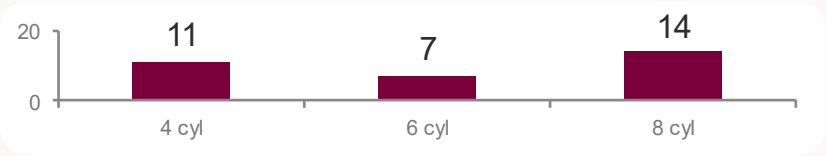


Statistical Transformations

Some geoms automatically compute statistics — you don't need to calculate them yourself

Geom	Default Stat	What it does
`geom_histogram()`	`stat_bin()`	Counts observations in bins
`geom_bar()`	`stat_count()`	Counts observations per category
`geom_smooth()`	`stat_smooth()`	Fits a smoothed line (loess or lm)
`geom_boxplot()`	`stat_boxplot()`	Computes five-number summary
`geom_density()`	`stat_density()`	Estimates probability density

Override default stat with the stat argument. geom_bar(stat = "identity") is equivalent to geom_col().



geom_bar() counts automatically

Themes: Polishing Your Plot

Themes control non-data elements: background, grid, fonts, and colors

`theme_minimal()`

Clean, no background. Great default.

`theme_bw()`

Black and white. Good for print.

`theme_classic()`

No grid lines. Journal-style.

`theme_void()`

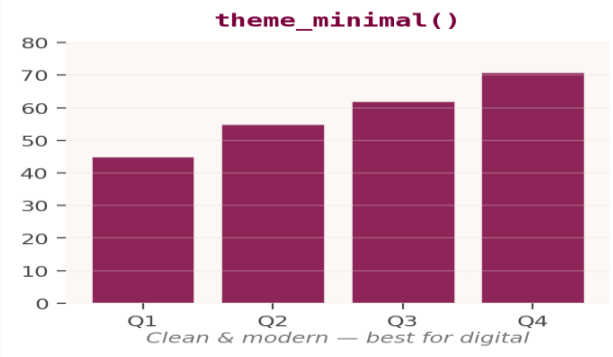
Nothing but data. For maps.

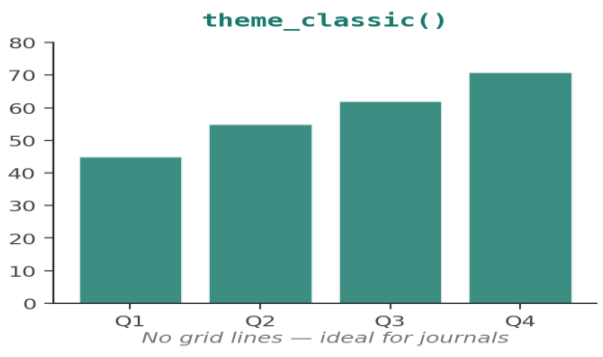
`theme_dark()`

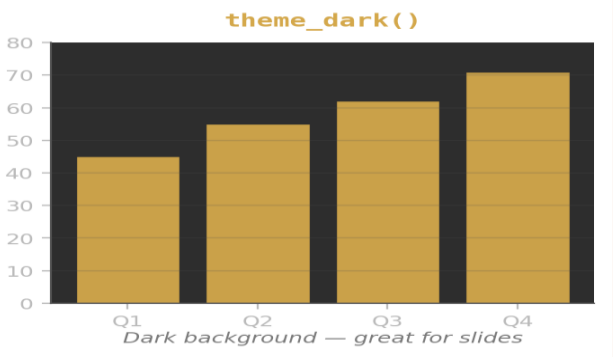
Dark background. For presentations.

`theme_light()`

Light gray background, subtle grid.







Labels and Annotations

Clear labels turn a chart from confusing to informative

```
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  labs(
    title = "Car Weight vs. Fuel Efficiency",
    subtitle = "Data from the mtcars dataset",
    x = "Weight (1000 lbs)",
    y = "Miles per Gallon",
    caption = "Source: Motor Trend, 1974"
  ) +
  theme_minimal()
```

labs() arguments:

title — Main plot title
subtitle — Secondary title
x — X-axis label
y — Y-axis label
caption — Bottom note/source
color / fill / size — Legend titles

Always include axis labels with units, and a descriptive title. Your future self (and your readers) will thank you!

Let's Practice!

Time to write some R code

Thank You!

Questions?

Email: Khades1@mcmaster.ca

Book DASH appointment: library.mcmaster.ca/services/dash

Contact DASH: libdash@mcmaster.ca