Data visualization

“The simple graph has brought more information to the data analyst’s mind than any other device.” — John Tukey

MKT 566

Instructor: Davide Proserpio

What we will learn

- This chapter will teach you how to visualize your data
  - (We are going to use ggplot2, an R library for data viz)
  - You can find an intro to ggplot2 here: https://raw.githack.com/dadepro/mkt-615/main/lectures/07-dataviz/07-dataviz.html#4

- What types of charts exist & what they are used for
- How to pick the best visual option for different types of data
- How to create compelling figures
- Content partially based on Chapter 3 of R for Data Science

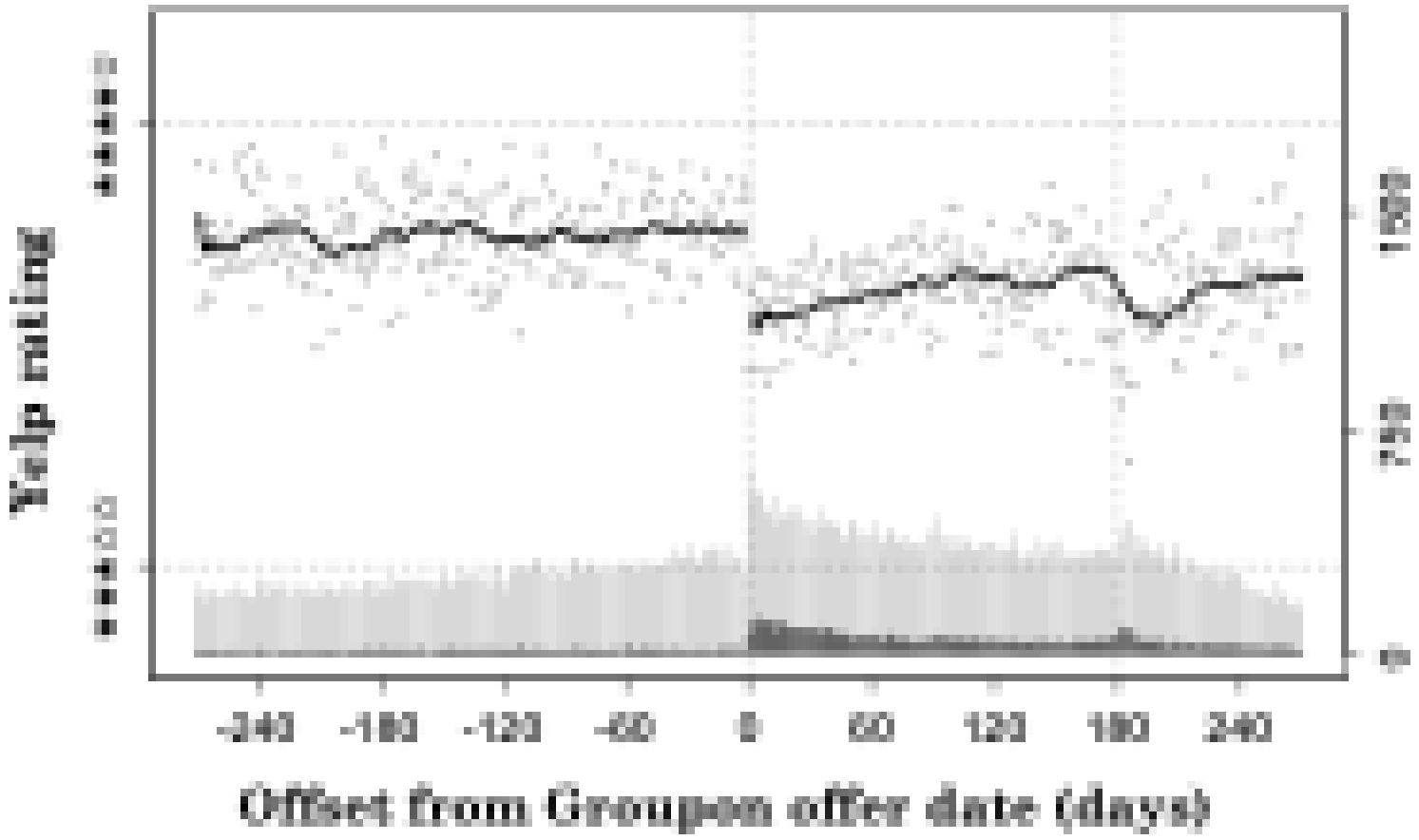
R Scripts

There are two R scripts on the course website:

- Chart types (reproduces all the different charts we will discuss today)
- Beautify figure (reproduces a simple figure beautification process)

- Download and open them with RStudio
- (Try to) Install the required libraries

An (almost) perfect example



Source: [The Groupon Effect on Yelp Ratings: A Root Cause Analysis (Byers et al. 2012)]

Chart types

Category Comparisons: Show how discrete groups or items stack up against one another.

Part-to-Whole & Composition: Break down totals into components.

Chart types

Category Comparisons: Show how discrete groups or items stack up against one another.

Part-to-Whole & Composition: Break down totals into components.

- Bar Chart
- Pareto Chart (Sorted bars + cumulative line)
- Treemap Chart
- Pie Chart
- Waterfall Chart
- Heatmap (for categorical grids)

Chart types

- **Bar Chart**
- Pareto Chart
- Treemap Chart
- Pie Chart
- Waterfall Chart

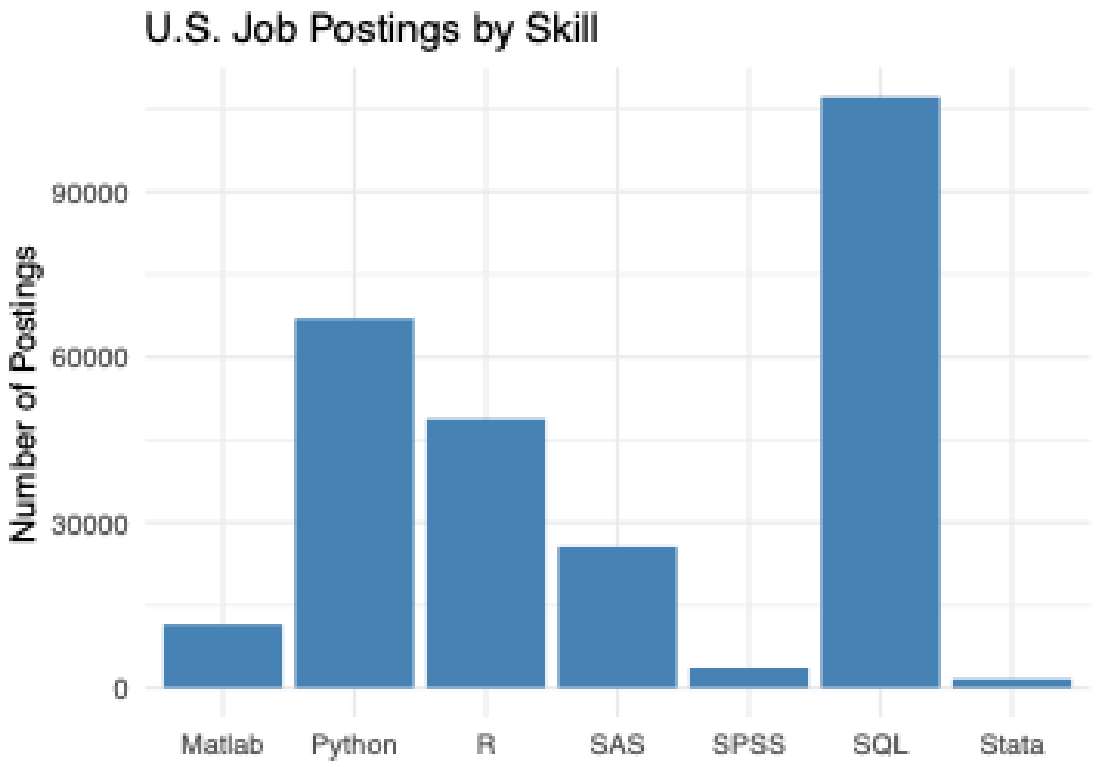


Chart types

- **Bar Chart**
- Pareto Chart
- Treemap Chart
- Pie Chart
- Waterfall Chart

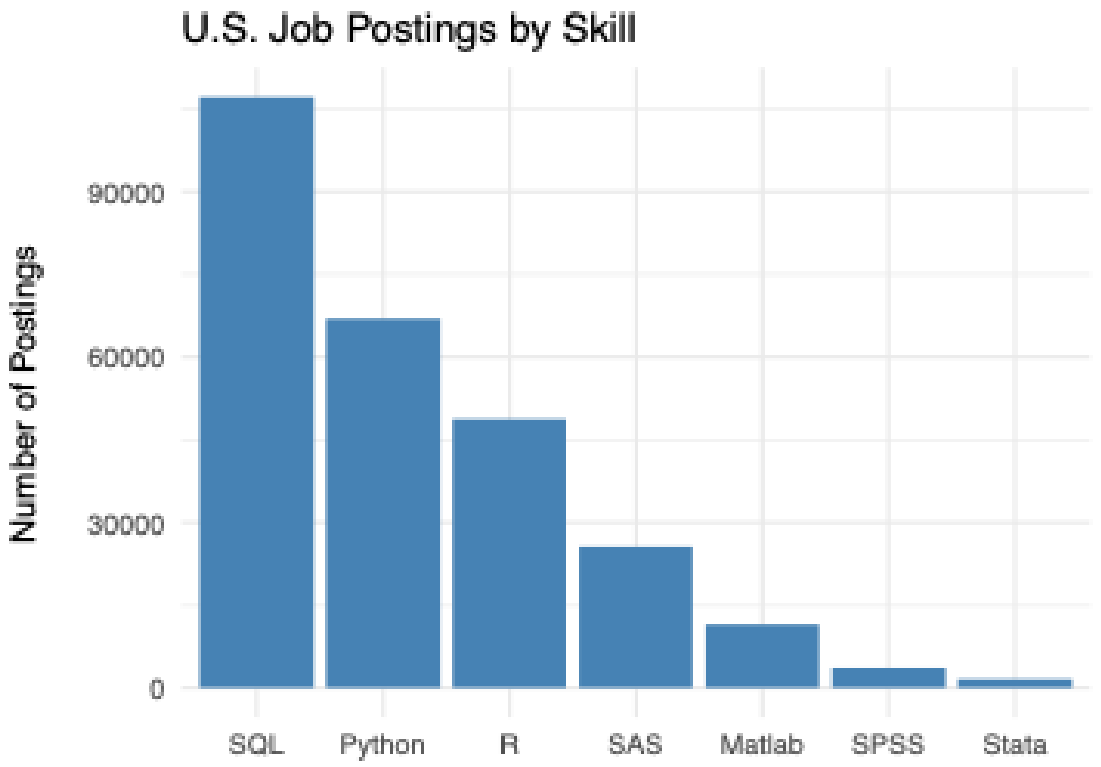


Chart types

- Bar Chart
- **Pareto Chart**
- Treemap Chart
- Pie Chart
- Waterfall Chart

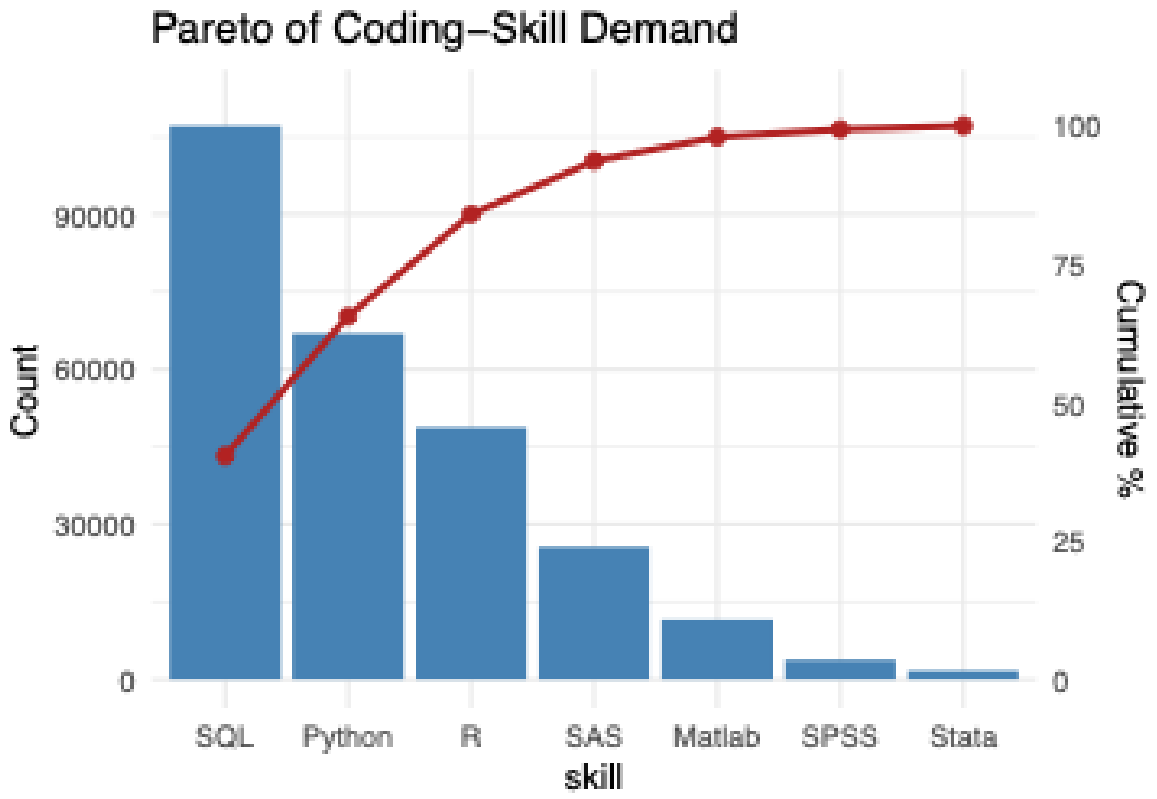


Chart types

- Bar Chart
- Pareto Chart
- **Treemap Chart**
- Pie Chart
- Waterfall Chart

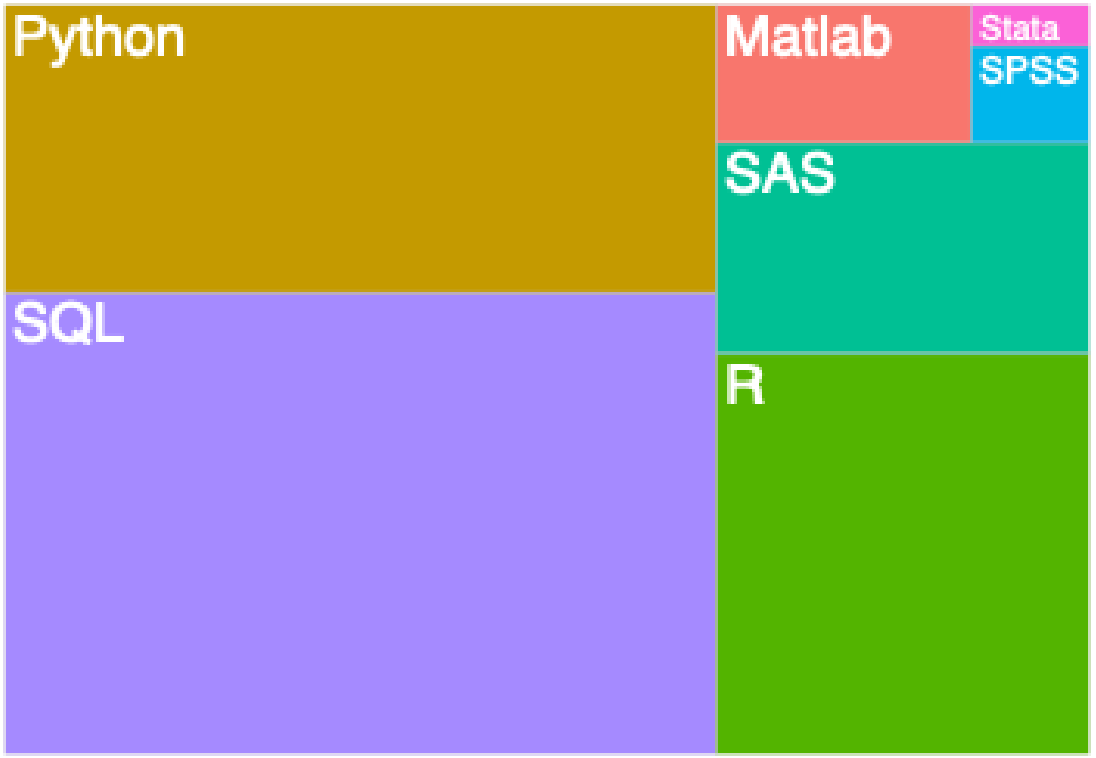


Chart types

- Bar Chart
- Pareto Chart
- Treemap Chart
- **Pie Chart**
- Waterfall Chart

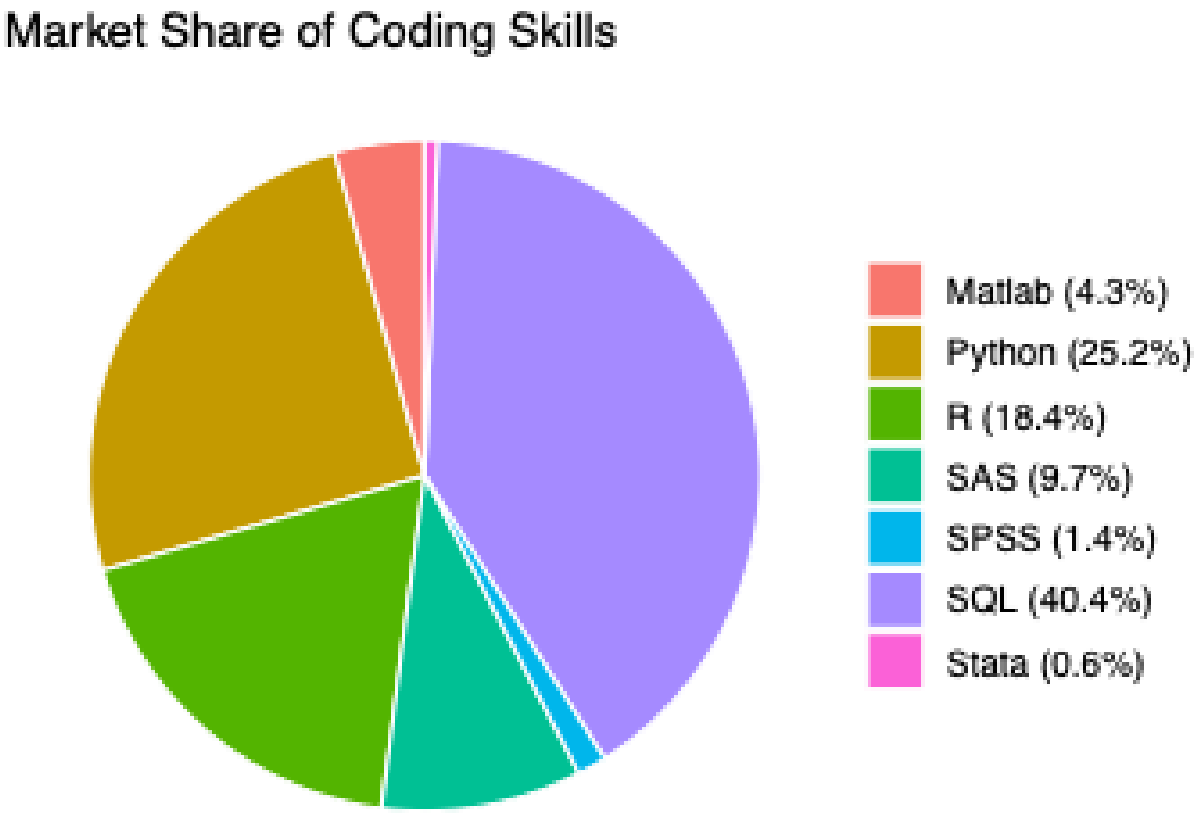


Chart types

- Bar Chart
- Pareto Chart
- Treemap Chart
- Pie Chart
- **Waterfall Chart**

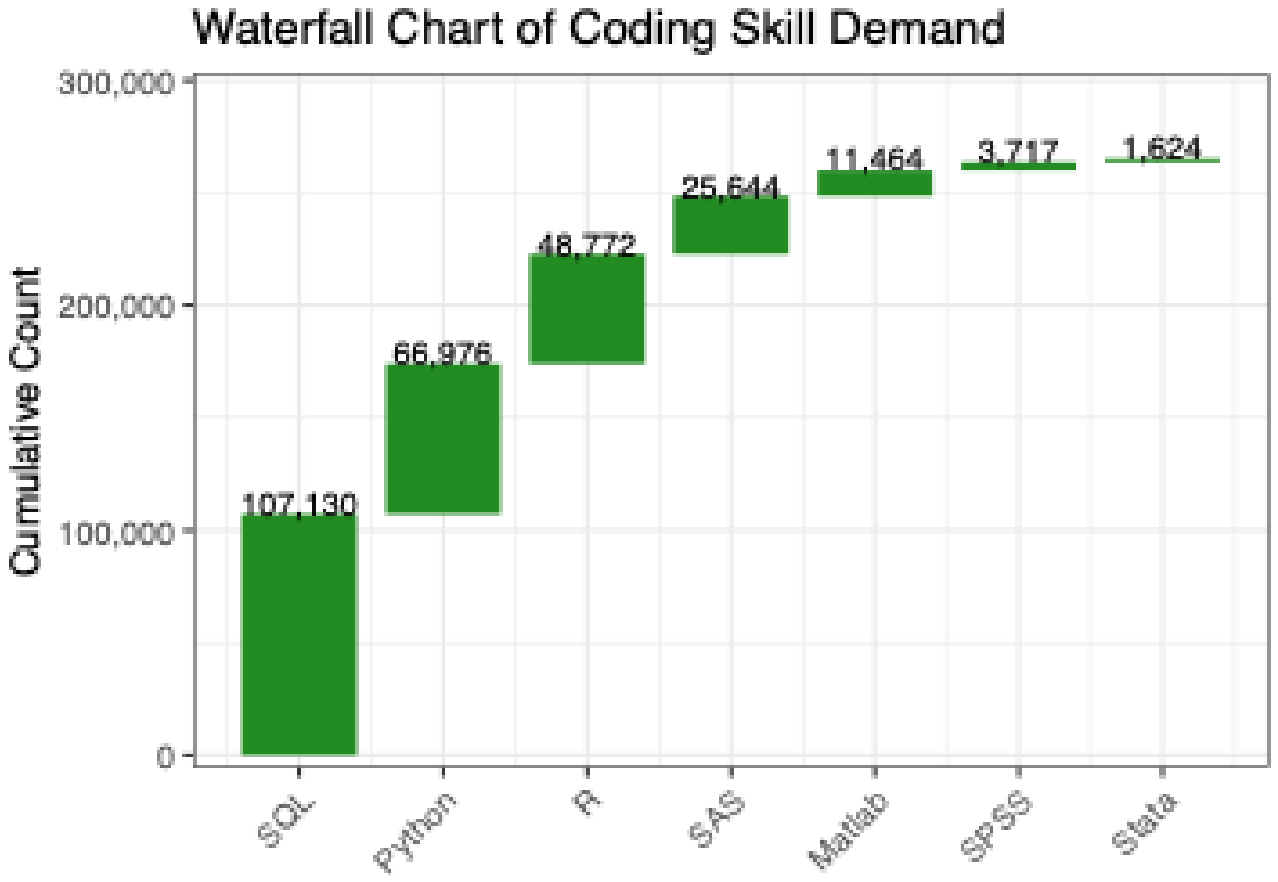


Chart types

Trends Over Time: Reveal how values evolve or accumulate.

Chart types

Trends Over Time: Reveal how values evolve or accumulate.

- Line Chart
- Area Chart (stacked or cumulative)
- Bar + Line Combo

New dataset

```
> store.df <- read.csv("http://goo.gl/QPDdMl")
> head(store.df)
  storeNum Year Week p1sales p2sales p1price p2price p1prom p2prom country
1      101    1    1     127     106    2.29    2.29      0      0      US
2      101    1    2     137     105    2.49    2.49      0      0      US
3      101    1    3     156      97    2.99    2.99      1      0      US
4      101    1    4     117     106    2.99    3.19      0      0      US
5      101    1    5     138     100    2.49    2.59      0      1      US
6      101    1    6     115     127    2.79    2.49      0      0      US
>
```

* Sales are in units counts

Chart types

- **Line Chart**
- Area Chart
- Bar + Line Combo

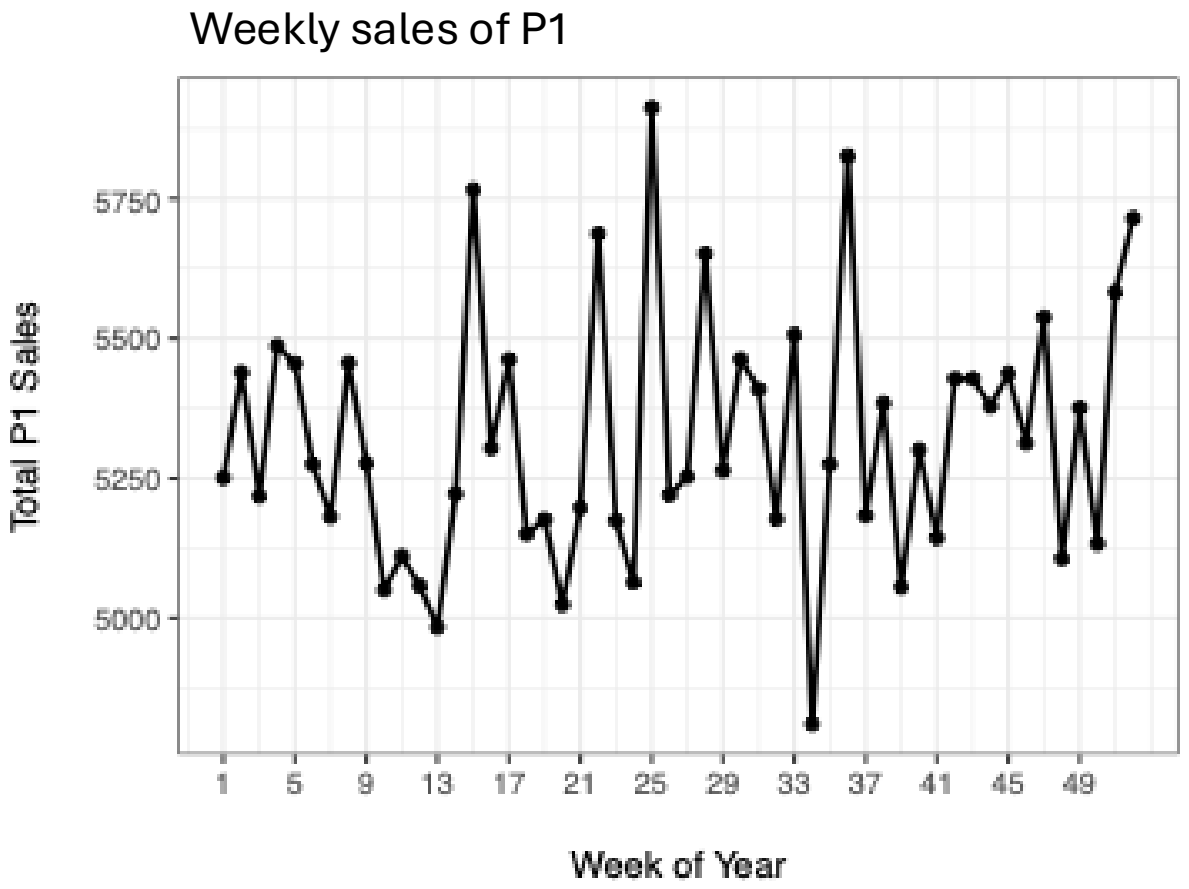


Chart types

- Line Chart
- **Area Chart**
- Bar + Line Combo

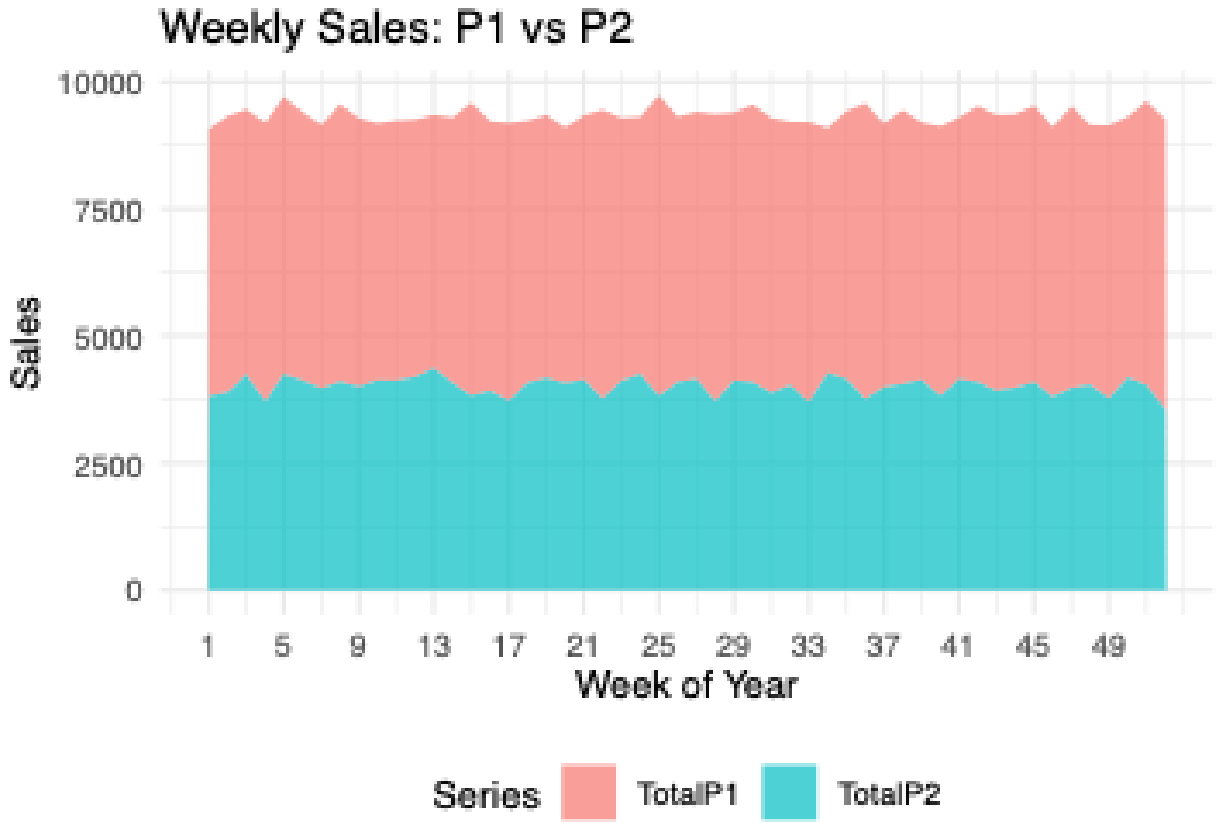


Chart types

- Line Chart
- Area Chart
- **Bar + Line Combo**

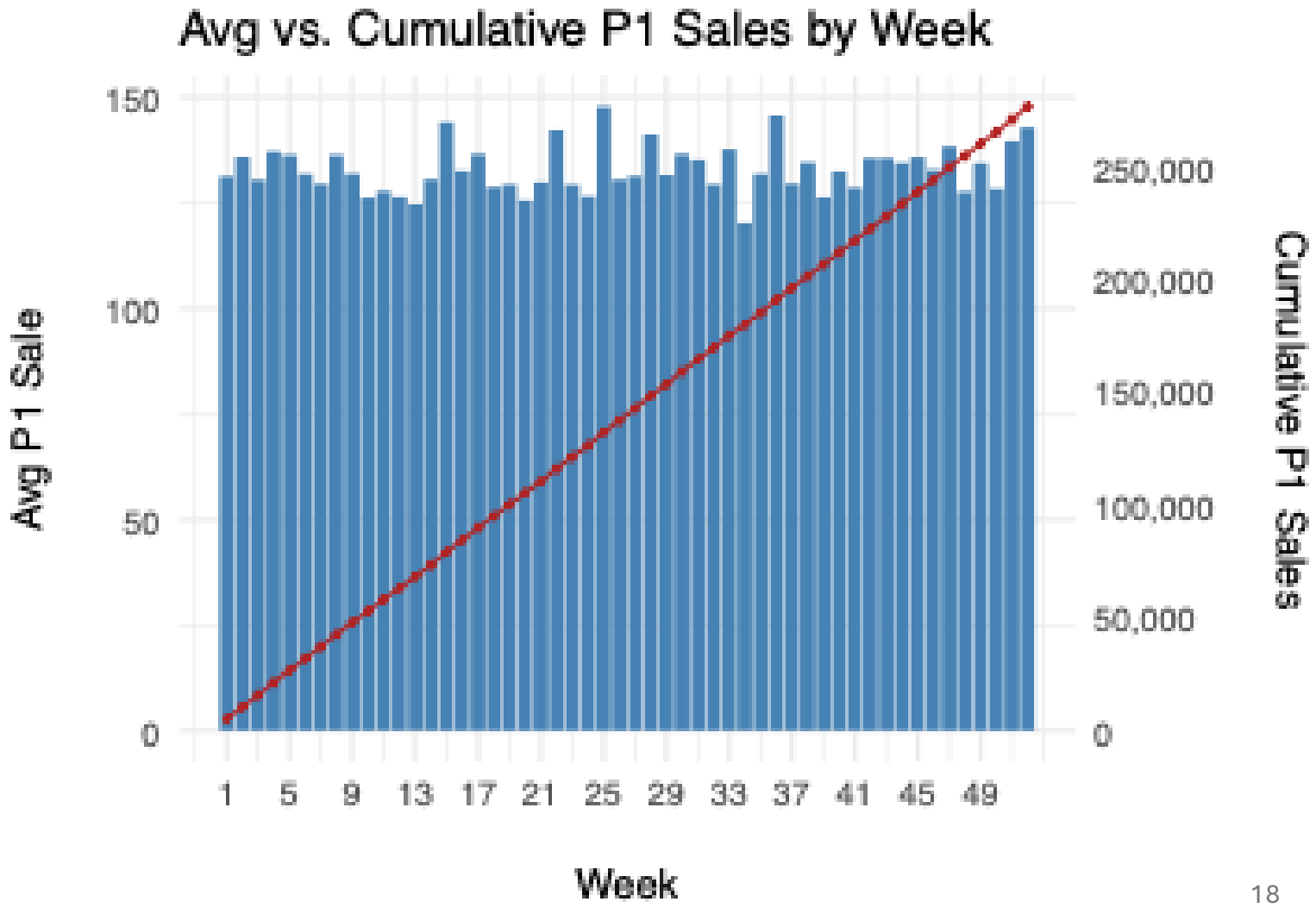


Chart types

Distribution & Density: Understand the shape, spread, and outliers of a variable.

Chart types

Distribution & Density: Understand the shape, spread, and outliers of a variable.

- Histogram
- Density Plot
- Box Plot
- Violin Plot

Chart types

- **Histogram**
- Density Plot
- Box Plot
- Violin Plot

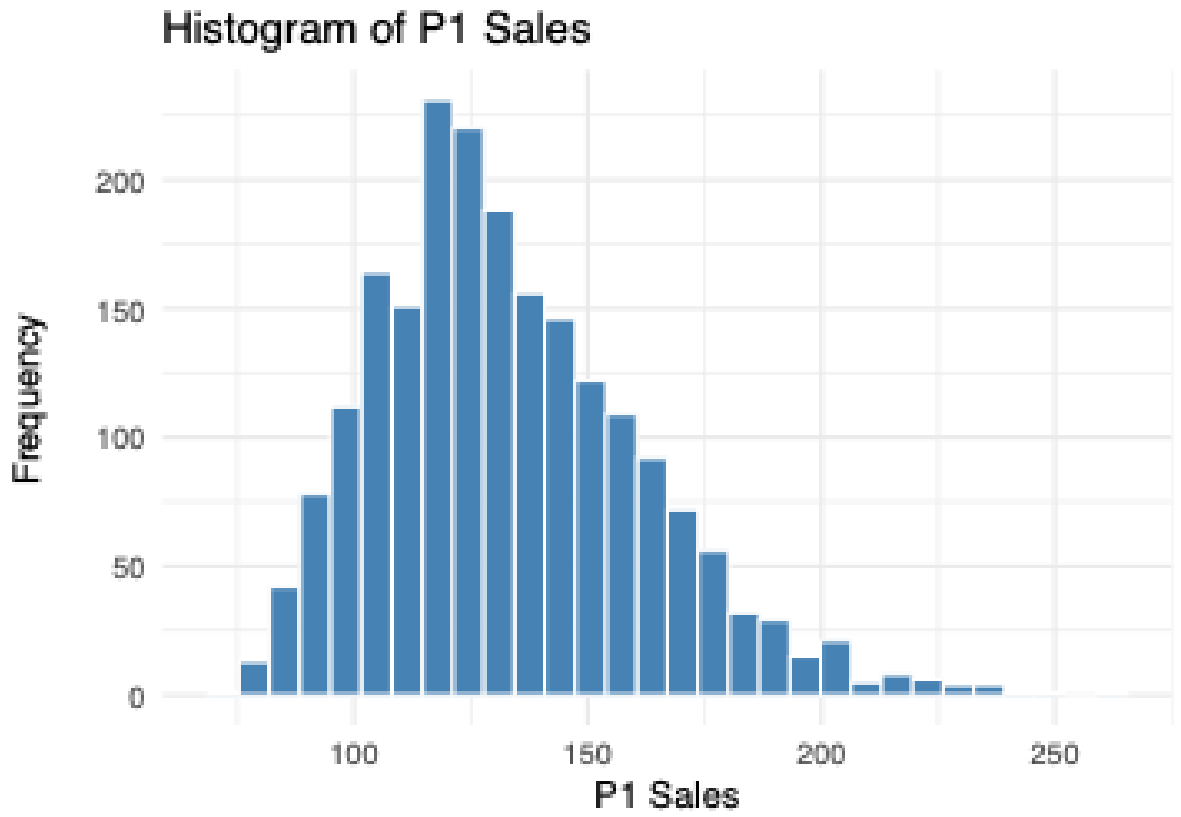


Chart types

- Histogram
- **Density Plot**
- Box Plot
- Violin Plot

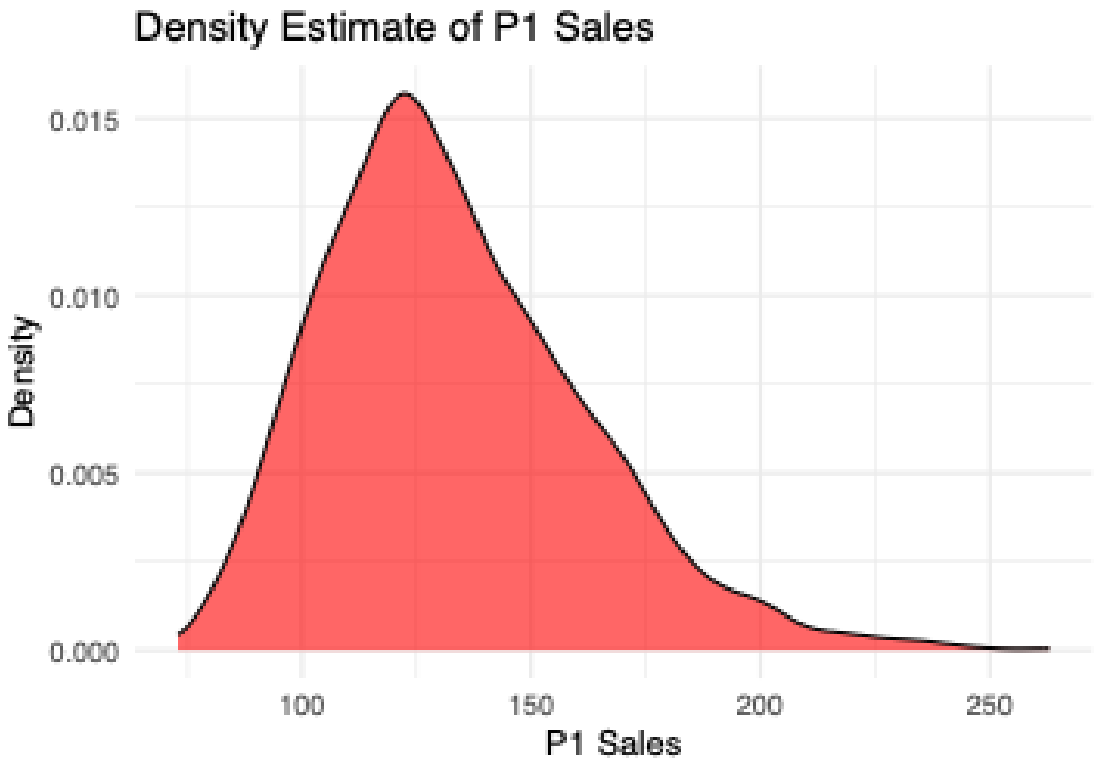
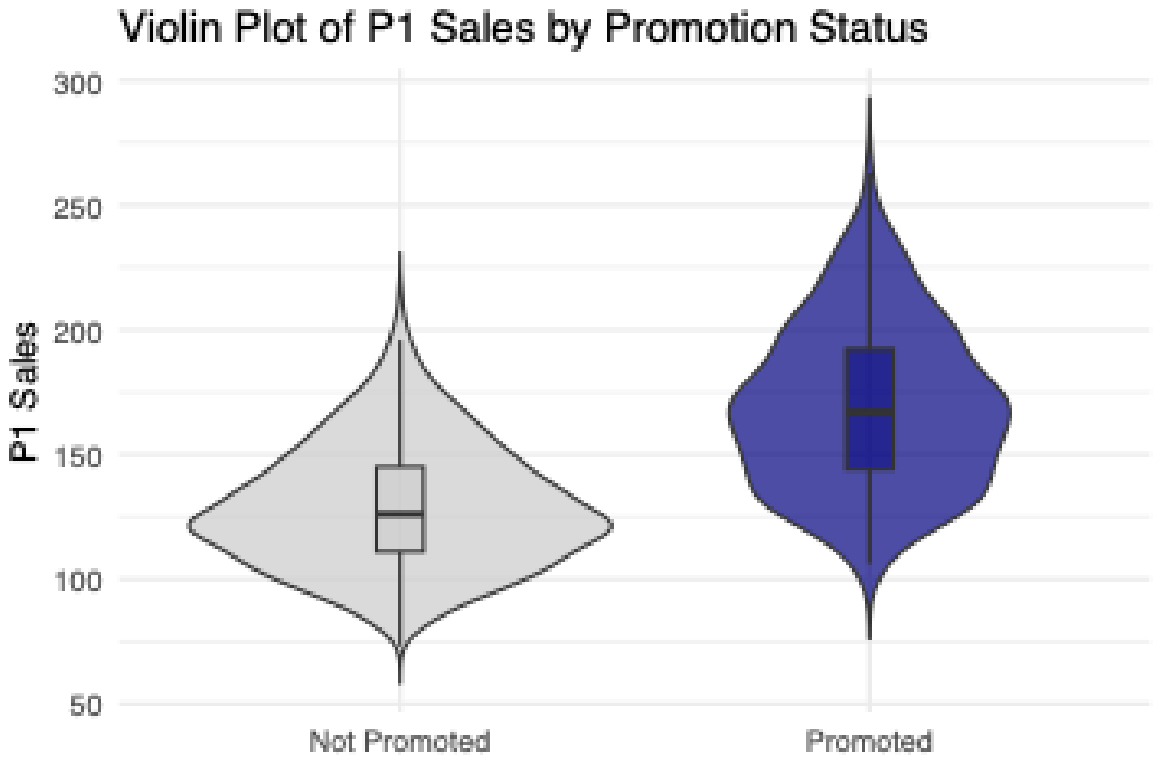


Chart types

- Histogram
- Density Plot
- **Box Plot**
- Violin Plot

Chart types

- Histogram
- Density Plot
- Box Plot
- **Violin Plot**



Box plot vs violin plot

Aspect	Boxplot	Violin plot
What it shows	**Five-number summary** (Q1, median, Q3; whiskers; outliers)	**Distribution shape** (smoothed density), can show quantiles if you add them
Outliers	Explicit points beyond whiskers (1.5×IQR rule)	Not shown by default
Multimodality	Hard to see	**Easy to see** (multiple “bulges”)
Robustness	**Robust**: based on quantiles	Depends on **bandwidth** and smoothing
Small samples	**Reliable**	Can be misleading (noisy density)
When to use	Compare medians/spread cleanly	Understand **shape** and differences beyond the median

Chart types

Relationships & Correlation: Explore how two (or more) variables move together.

Chart types

Relationships & Correlation: Explore how two (or more) variables move together.

- Scatter Plot
- Bubble Chart (scatter + size)

Chart types

- **Scatter Plot**
- Bubble Chart

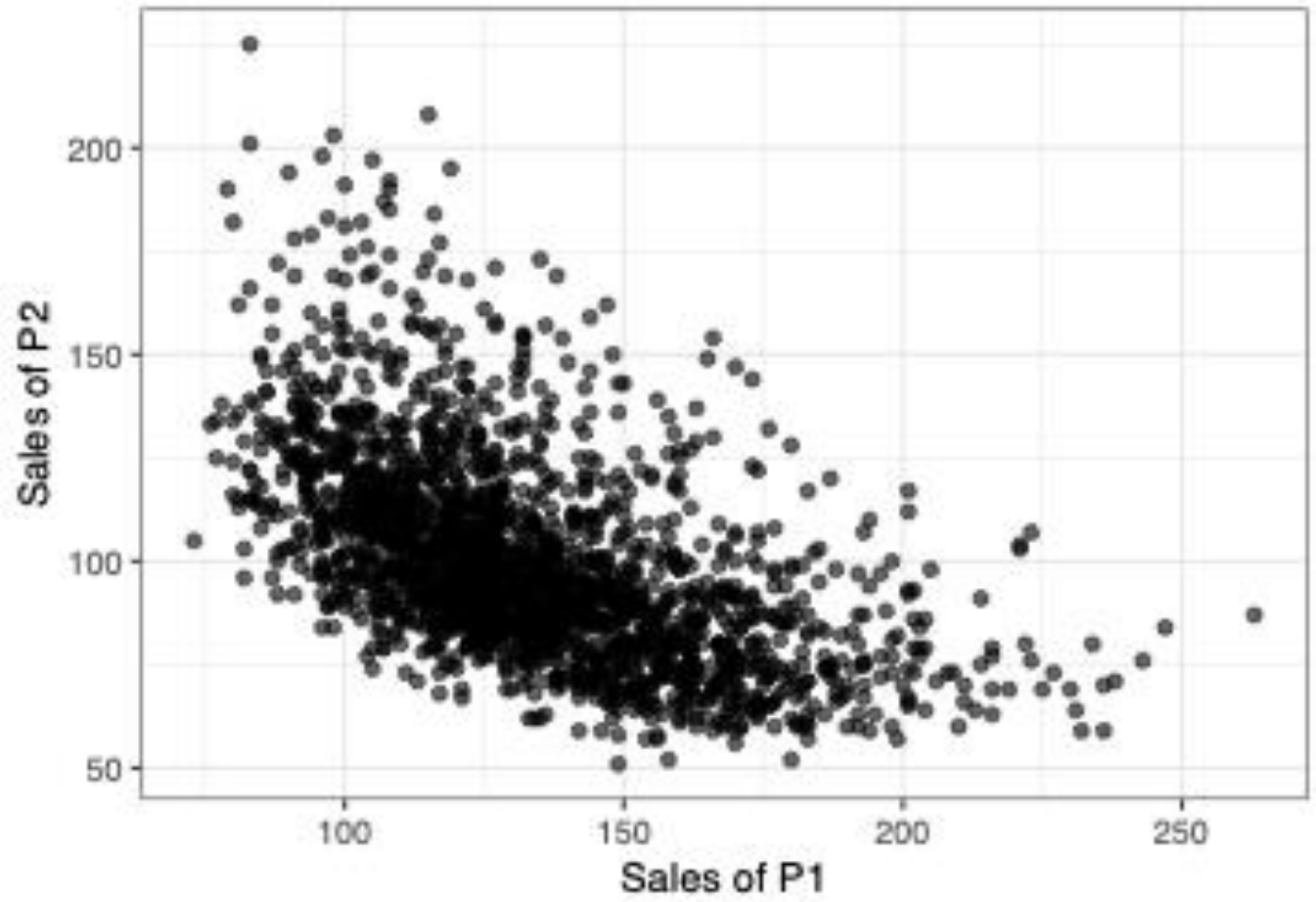


Chart types

- Scatter Plot
- **Bubble Chart**

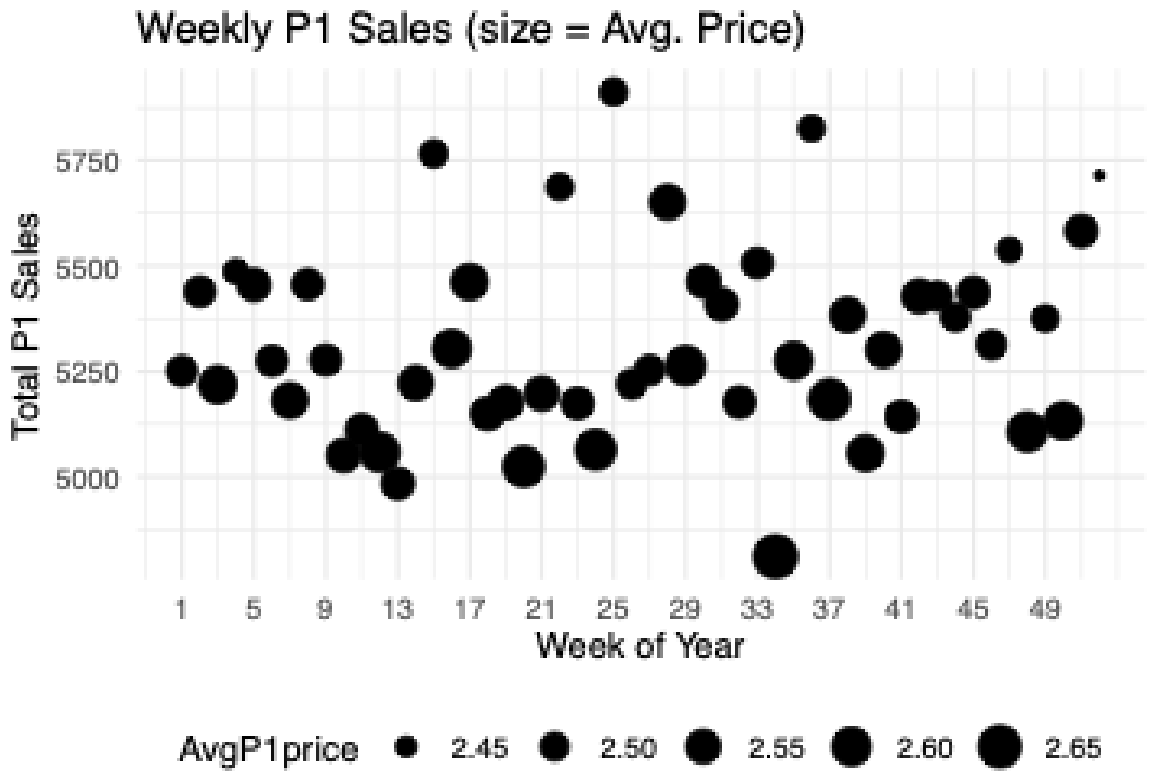


Chart types

- **Geospatial & Matrix Data:** Map values over space or grid layouts.

Chart types

- **Geospatial & Matrix Data:** Map values over space or grid layouts.
  - Geospatial Map (choropleth, points)
  - Heatmap (correlation matrix or spatial grid)

Chart types

- **Geospatial Map**
- Heatmap

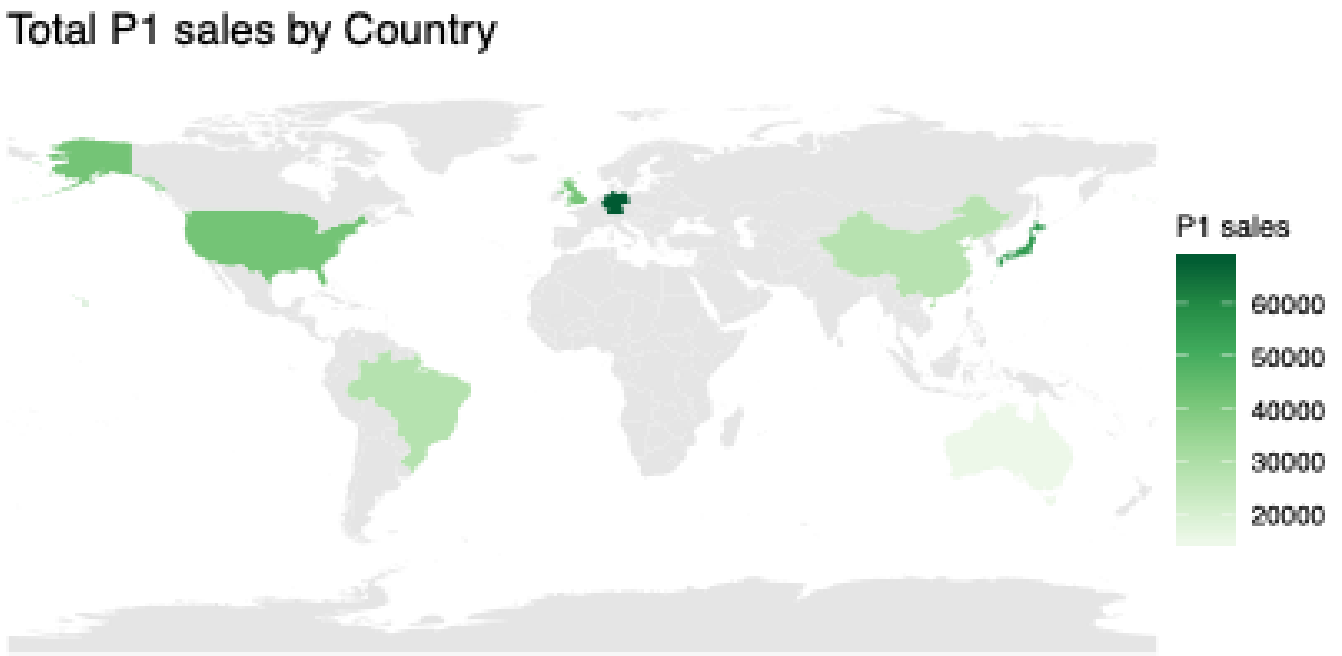
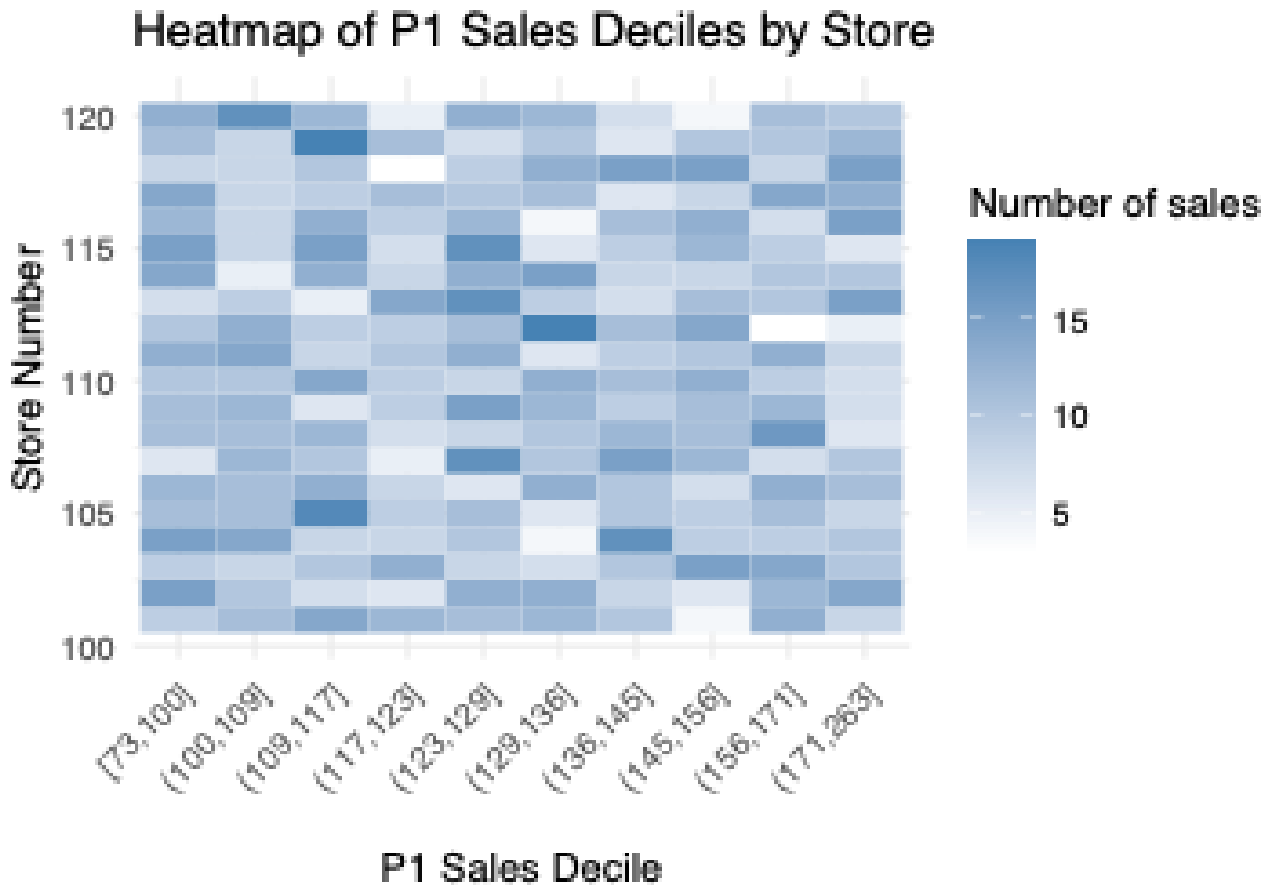


Chart types

- Geospatial Map
- **Heatmap**



Choosing the best chart

- **Define your question**: Comparison? Trend? Distribution? Relationship?
- **Inspect your variables**: Categorical vs. numeric; panel vs cross-section, time vs. location vs. hierarchy

- For example:
  - You want to see **price trends over time** → go with a **line chart**.
  - You want to **compare current job-post counts across languages** → a **bar chart**
  - You're exploring **salary distributions by city** → a **box** or **violin plot**

Best practice for good viz

Simplify & Declutter

- **Reduce “chart junk”**: eliminate unnecessary gridlines, backgrounds, and 3D effects
  - I often use **theme_few()** in R
- **Legends only when needed**: if you label directly on the plot, drop the legend, don’t be redundant

Best practice for good viz

Use Readable Scales & Labels

- **Descriptive titles & subtitles**: tell viewers what they're looking at and why it matters.
- **Clear axis labels**: include units (e.g., "Sales (USD Millions)") and avoid abbreviations when possible
- **Consistent breaks**: choose nice, round numbers or evenly spaced dates
- Always add **figure notes** at the bottom of the figure in documents and reports

Best practice for good viz

Choose Accessible Color & Style

- **Color-blind–friendly palettes**
  - In R, palettes from RColorBrewer (“Set2”, “Dark2”) or viridis.
- **Limit colors**: no more than 4–6 distinct colors in a single plot. For many categories, consider facets or small multiples instead
- **Transparency** to manage overplotting in dense scatter or area charts
  - Alpha parameter in R

Best practice for good viz

Leverage Facets

- In R, facet_wrap() / facet_grid() for splitting by a categorical variable rather than cramming everything into one panel.
- Ensures consistent scales and easy side-by-side comparisons

Best practice for good viz

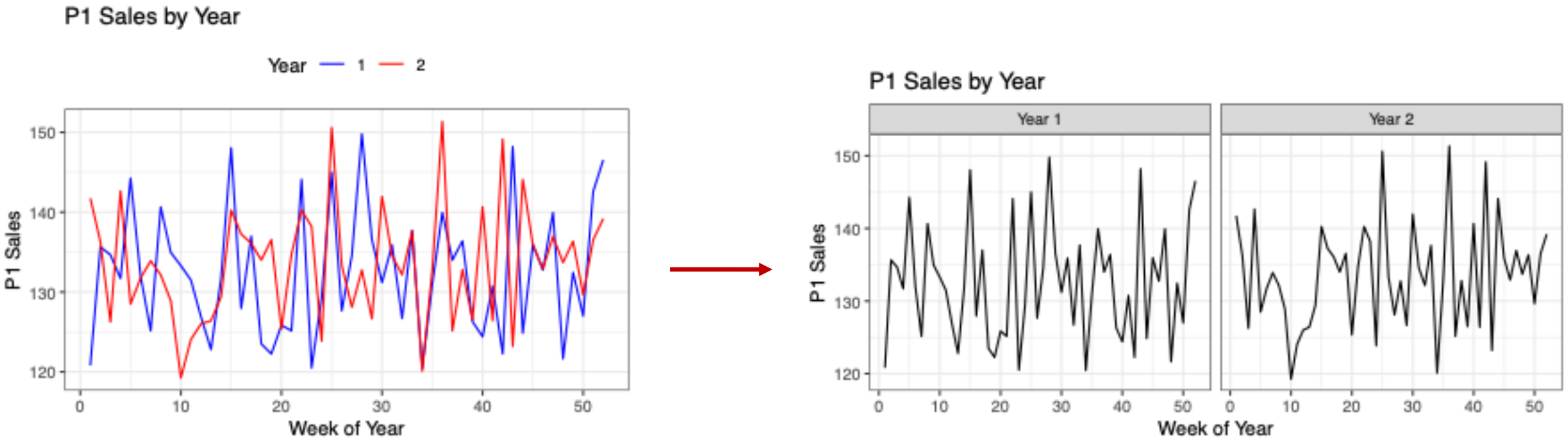
Leverage Facets

- In R, facet_wrap() / facet_grid() for splitting by a categorical variable rather than cramming everything into one panel.
- Ensures consistent scales and easy side-by-side comparisons

Best practice for good viz

Leverage Facets

- In R, facet_wrap() / facet_grid() for splitting by a categorical variable rather than cramming everything into one panel.
- Ensures consistent scales and easy side-by-side comparisons



Best practice for good viz

Annotate & Highlight Key Insights

- **Direct labels** with geom_text() or ggrepel for calling out peaks, thresholds, or outliers
- **Annotations** (annotate() or geom_vline()/geom_hline()) to mark events—product launches, policy changes, seasonal holidays

Best practice for good viz

Consistency Across Plots

- Define a **custom theme** and apply it to every chart to ensure that the colors, fonts, and margins are consistent.
- Use the same color mapping for the same variables across multiple plots.

Best practice for good viz

Validate & Iterate

- **Peer review**: show rough drafts to classmates: do they “get” the story without explanation?
- **Test in grayscale**: to verify that patterns and contrasts remain readable when printed without color

Best practice for good viz

Break the rules **only** when doing so tells a clearer story. Good visualization is as much art as science!

Dataset: mpg

Dataset of car manufacturers and car models information: https://rpubs.com/shailesh/mpg-exploration

This dataset provides fuel economy data from 1999 and 2008 for 38 popular models of cars. The dataset is shipped with *ggplot2* package.

Variable	Type	Description	Details
manufacturer	string	car manufacturer	15 manufacturers
model	string	model name	38 models
displ	numeric	engine displacement in liters	1.6 - 7.0, median: 3.3
year	integer	year of manufacturing	1999, 2008
cyl		number of cylinders	4, 5, 6, 8
trans	string	type of transmission	automatic, manual (many sub types)
drv	string	drive type	f, r, 4, f=front wheel, r=rear wheel, 4=4 wheel
cty	integer	city mileage	miles per gallon
hwy	integer	highway mileage	miles per gallon
fl	string	fuel type	5 fuel types (diesel, petrol, electric, etc.)
class	string	vehicle class	7 types (compact, SUV, minivan etc.)

Dataset: mpg

Dataset of car manufacturers and car models information: https://rpubs.com/shailesh/mpg-exploration

```
> head(mpg)
# A tibble: 6 × 11
  manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
  <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compact
2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compact
3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compact
4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compact
5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compact
6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compact
```

Dataset: mpg

```
> str(mpg)
tibble [234 × 11] (S3: tbl_df/tbl/data.frame)
 $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
 $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
 $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
 $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
 $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
 $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
 $ drv         : chr [1:234] "f" "f" "f" "f" ...
 $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
 $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
 $ fl          : chr [1:234] "p" "p" "p" "p" ...
 $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

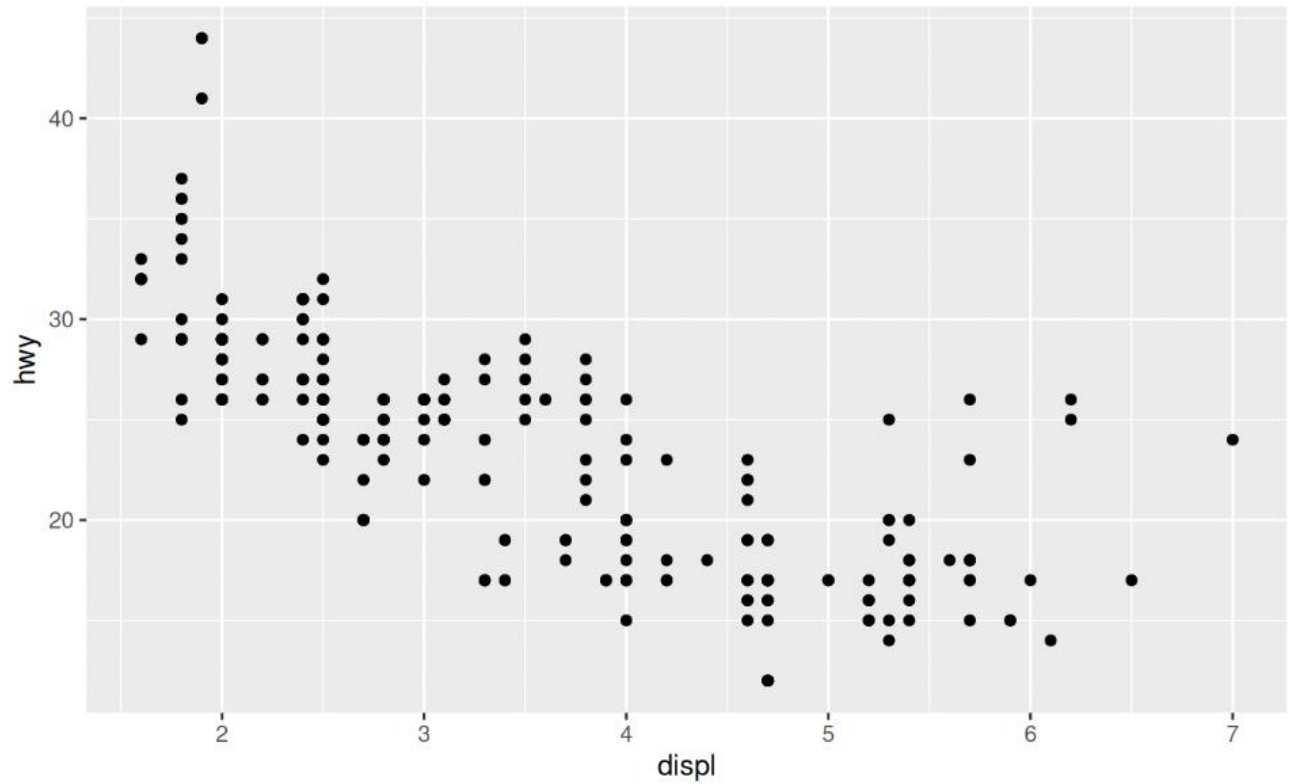
Scatterplot

- Useful to understand the relationship between two variables

- Let’s try to use it to learn the relationship between engine size (displ) and highway fuel efficiency (hwy)

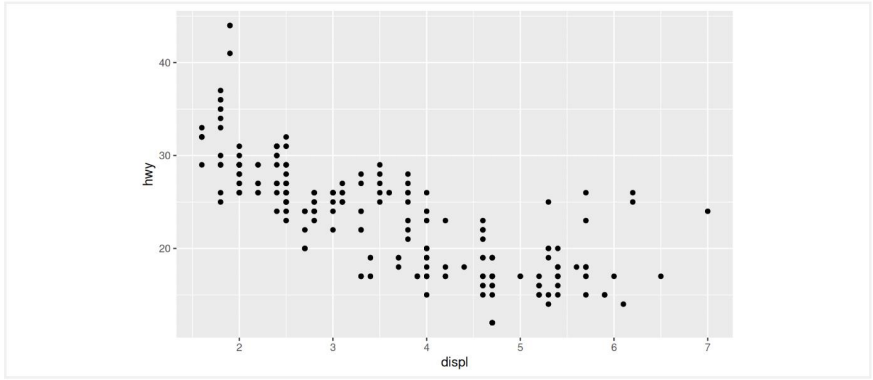
Scatterplot

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



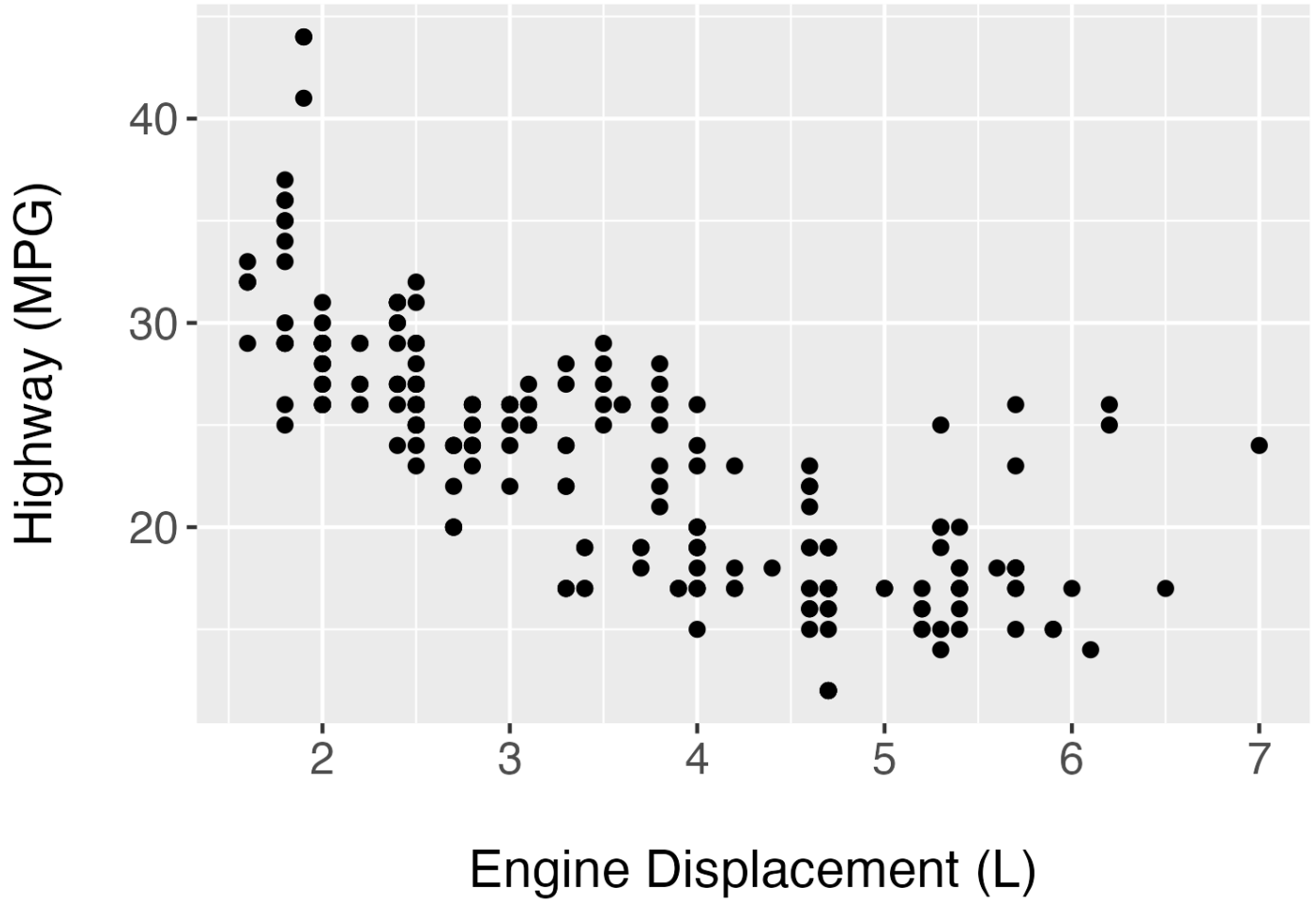
Scatterplot

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

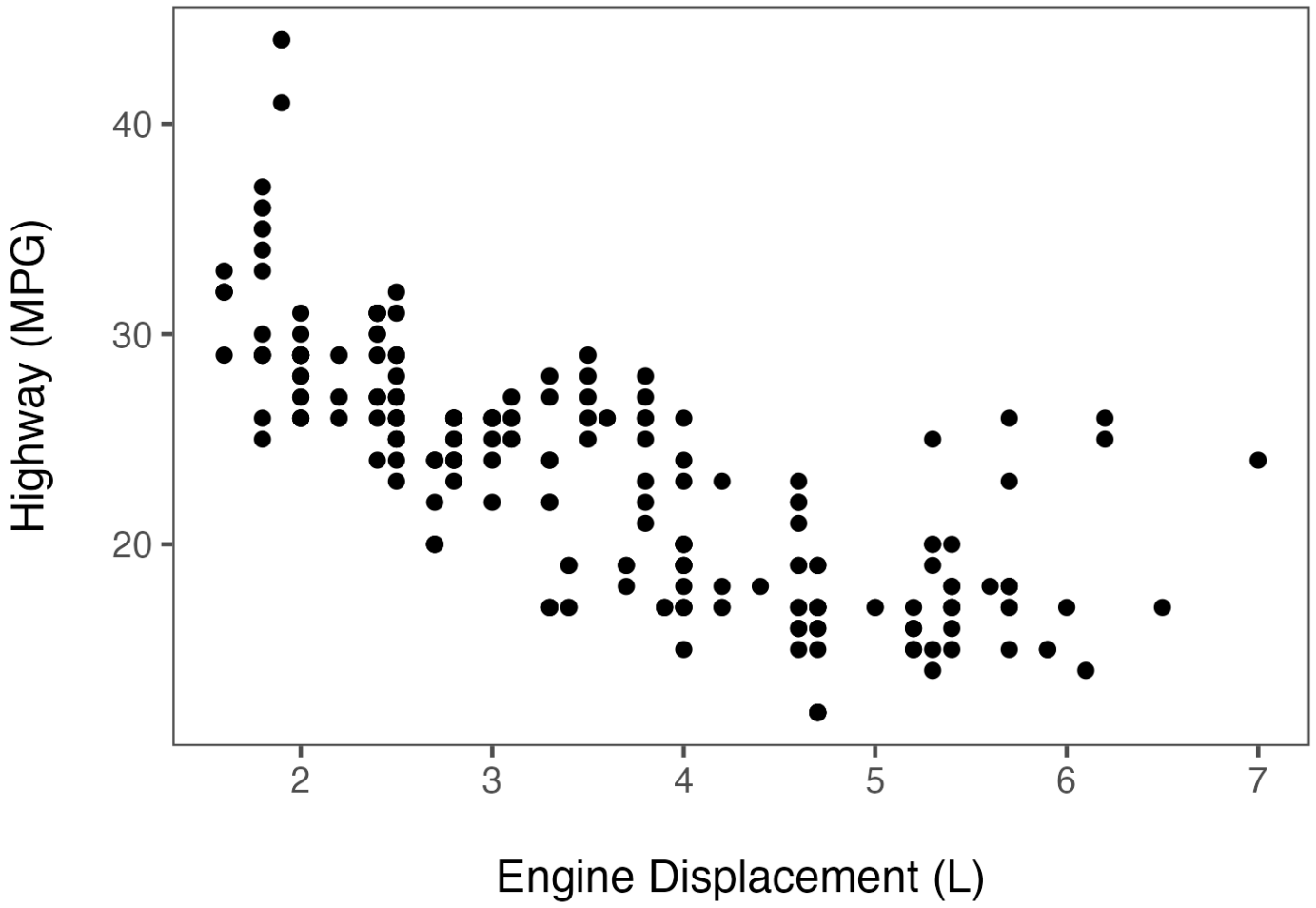


- Axis labels are not easy to interpret
- The units in which the variables are measured are unclear
- Axis font is very small
- Little space between axis labels and axis names
- Let’s fix these issues…

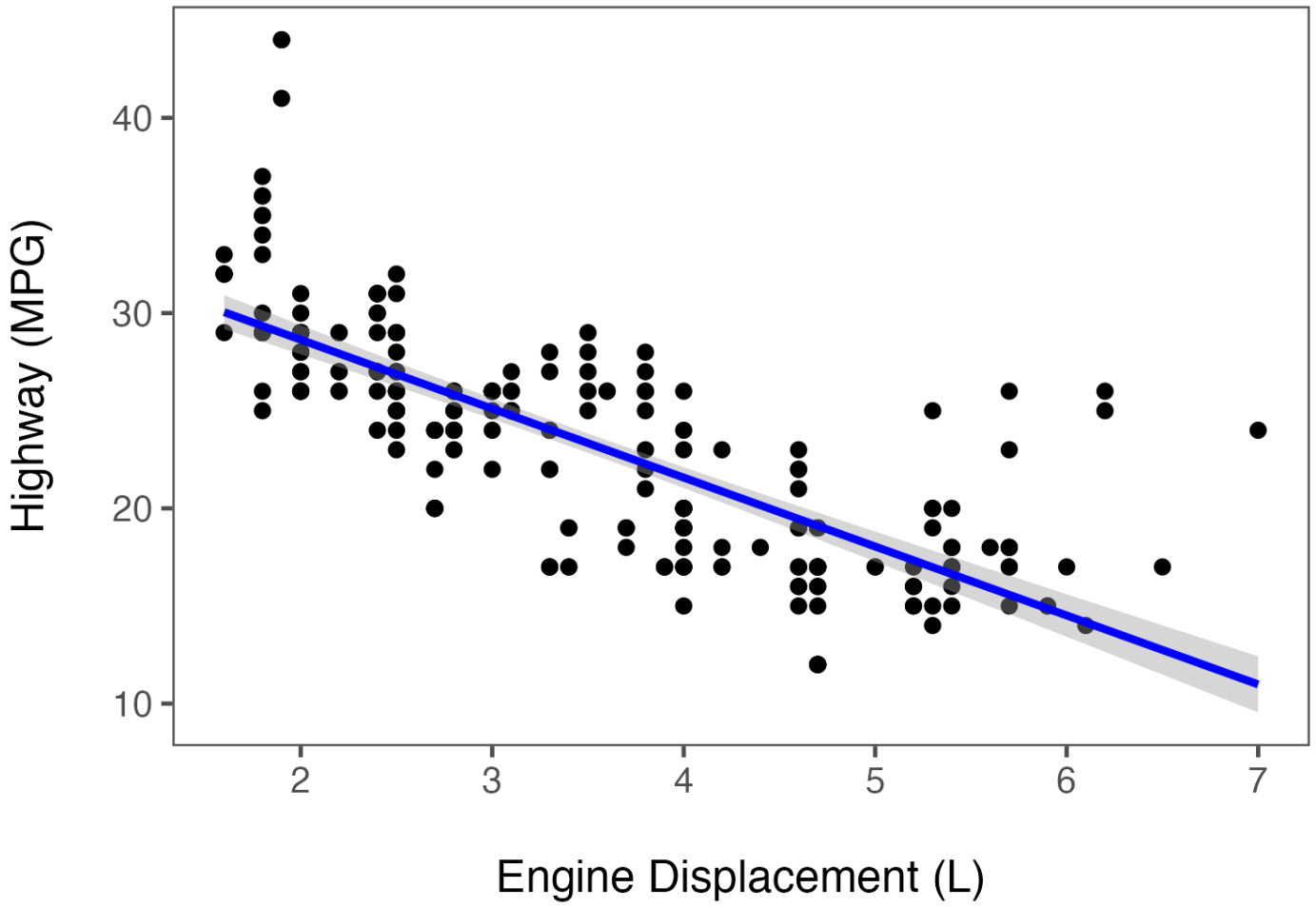
Scatterplot



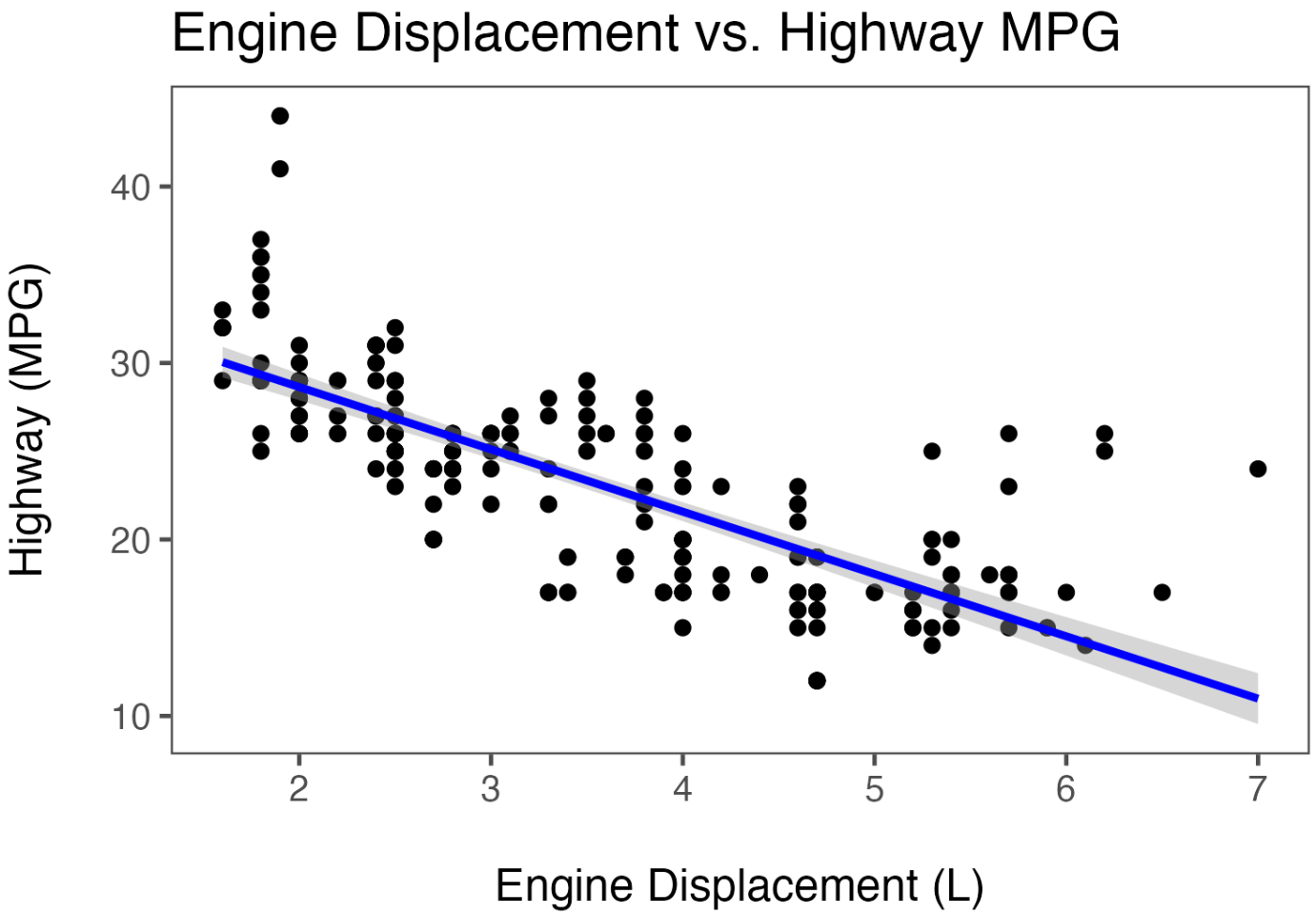
Scatterplot



Scatterplot



Scatterplot



Scatterplot

- Code to reproduce this exercise is: w1-2-data-viz.R