TYPES OF VISUALIZATION

1. Scatter Plots

A scatter plot is a fundamental type of visualization that displays the relationship between two continuous variables by plotting individual data points on an X-Y axis. In R, scatter plots can be created using the base `plot()` function or `ggplot2`'s `geom_point()`. They are particularly useful for identifying correlations, clusters, and outliers in data. For example, in a dataset containing height and weight measurements, a scatter plot can reveal whether taller individuals tend to weigh more. Customizations such as point size, color, and transparency (`alpha`) can enhance readability, while adding trend lines (`geom_smooth()`) can help visualize patterns.

2. Line Charts

Line charts are used to display trends over time or ordered categories by connecting data points with straight lines. In R, they are created using `geom_line()` in `ggplot2` or `plot(type = "l")` in base R. Line charts are ideal for time-series data, such as stock prices over months or temperature changes across days. Multiple lines can be plotted on the same graph to compare trends, with different colors or line styles (dashed, dotted) distinguishing between groups. Annotations (`annotate()`) and interactive tooltips (using `plotly`) can further improve interpretation.

3. Bar Charts

Bar charts represent categorical data with rectangular bars, where the height or length corresponds to the value of each category. In R, `geom_bar()` (for counts) or `geom_col()` (for pre-aggregated values) in `ggplot2` are commonly used, while `barplot()` serves a similar purpose in base R. Bar charts are useful for comparing quantities across groups, such as sales by region or survey responses by category. Variations include stacked bars (showing sub-category breakdowns) and grouped bars

(for side-by-side comparisons). Customizing fill colors, adding labels (`geom_text()`), and adjusting bar width enhance clarity.

4. Histograms

Histograms visualize the distribution of a continuous variable by dividing the data into bins (intervals) and displaying the frequency of values in each bin as bars. In R, `geom_histogram()` (ggplot2) or `hist()` (base R) are used. Histograms help identify skewness, central tendency, and outliers—for example, examining the distribution of exam scores in a class. The choice of bin width (`bins` or `binwidth`) significantly impacts the interpretation; too few bins may obscure patterns, while too many can overcomplicate. Density plots (`geom_density()`) are a smoothed alternative.

5. Box Plots (Box-and-Whisker Plots)

Box plots summarize the distribution of a continuous variable through quartiles, displaying the median (middle line), interquartile range (IQR, the box), and potential outliers (dots beyond "whiskers"). In R, `geom_boxplot()` (ggplot2) or `boxplot()` (base R) are used. They are excellent for comparing distributions across categories—for instance, comparing salaries across job roles. The plot highlights skewness, spread, and extreme values without requiring assumptions about the underlying distribution. Variations like violin plots (`geom_violin()`) combine box plots with density traces.

6. Heatmaps

Heatmaps use color gradients to represent matrix-like data, where each cell's color intensity reflects its value. In R, `geom_tile()` (ggplot2) or `heatmap()` (base R) are common functions. Heatmaps are widely used in genomics (gene expression), correlation matrices, or time-based patterns (e.g., hourly website traffic across days). Custom color palettes (`scale_fill_gradient()`) and clustering (`dendrogram` in `heatmap()`) can reveal hidden structures. Interactive versions (via `plotly` or `d3heatmap`) allow zooming and hovering for details.

7. Violin Plots

Violin plots combine features of box plots and density plots, showing the full distribution of data through mirrored density curves. In R, `geom_violin()` (ggplot2) is used, often layered with `geom_boxplot()` for quartile markers. They are useful for comparing distributions across groups, especially when multimodality (multiple peaks) exists. For example, violin plots can reveal bimodal age distributions in survey responses for different products. Adjusting smoothing parameters (`adjust`) and scaling (`scale = "area"`) fine-tunes the representation.

8. Density Plots

Density plots estimate the probability density function of a continuous variable, offering a smoothed version of histograms. In R, `geom_density()` (ggplot2) or `density()` (base R) create these curves. They are ideal for visualizing underlying distributions without binning artifacts—for example, comparing the salary distributions of two industries. Multiple densities can be overlaid with transparency (`alpha`) for comparison. Kernel density estimation (KDE) parameters (`bw` for bandwidth) control smoothness.

9. Network Diagrams

Network diagrams visualize relationships (edges) between entities (nodes), such as social networks or system dependencies. In R, packages like `igraph` (for layout algorithms) and `ggraph` (ggplot2 extension) create these plots. Nodes can be sized by importance (e.g., centrality metrics) and colored by group, while edges may represent connection strength (e.g., line width). Interactive tools (`visNetwork`) enable exploration of large networks.

10. Spatial Maps

R supports geographic visualizations using packages like `sf`, `leaflet`, and `tmap`. Choropleth maps (colored regions, e.g., `geom_sf()` in ggplot2) display aggregated data (e.g., election results by state), while point maps (e.g., `geom_point()` with

latitude/longitude) show individual locations. Interactive maps (`leaflet`) allow zooming and pop-ups. Projections (`coord_sf()`) ensure accurate spatial representation.

Each visualization type in R serves distinct analytical purposes, and choosing the right one depends on the data structure and the story to be conveyed. Customization options in R enable tailored, publication-quality graphics.