# Package: vizintel (via r-universe)

October 31, 2026

**Title** Visualization Intelligence Toolkit

**Version** 0.1.1

**Description** Provides automated data profiling and visualization recommendation tools for exploratory data analysis (EDA). The package inspects a data frame, classifies each variable as numeric, categorical, or datetime, and computes distributional statistics including skewness, excess kurtosis, Shapiro-Wilk normality test p-values (Royston 1995) <doi:10.2307/2986146>, and cardinality. Based on these profiles, the package scores and ranks candidate chart types (histogram, density, boxplot, violin, bar, lollipop, scatter, hexbin, line, heatmap) using an internal taxonomy that accounts for variable types, analysis goal, skewness, cardinality, and data size. Additional functions audit existing ggplot objects for common visualization integrity problems such as truncated axes, lie factors, overplotting, and dual y-axes following principles described in Tufte (1983, ISBN:0961392142) and Wilke (2019) <https://clauswilke.com/dataviz/>. The package also checks visualizations for accessibility issues and suggests alternative chart types for a given plot.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en-US

**Imports** dplyr, e1071, ggplot2, gridExtra, rlang

**Suggests** colorspace, hexbin, patchwork, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Config/roxygen2/version** 8.1.0

**NeedsCompilation** no

**Author** Pooja Thirumal [aut, cre], Ronesha Navaneethan [aut], Amrit Pal [ctb] (Supervisor)

**Maintainer** Pooja Thirumal <poojathirumal70@gmail.com>

**Repository** https://cran.r-universe.dev

**Date/Publication** 2026-10-31 17:50:46 UTC

**RemoteUrl** https://github.com/cran/vizintel

**RemoteRef** HEAD

**RemoteSha** e4e75caac55a503ff19a4f3b6de9808ef91364e5

# Contents

plot.viz_recommend_result . . . 2
print.viz_audit_report . . . 3
print.viz_recommend_result . . . 3
summary.viz_audit_report . . . 4
viz_accessibility . . . 4
viz_alternatives . . . 5
viz_audit . . . 6
viz_explore . . . 7
viz_recommend . . . 8

**Index** **10**

---

`plot.viz_recommend_result` *Plot recommended charts from a viz_recommend_result*

---

## Description

Renders each recommended chart type as a real `ggplot` using the data stored inside the result object, then arranges all panels into a single view with `gridExtra::grid.arrange()`. Each panel is labelled with the chart type and its fit score.

## Usage

```
## S3 method for class 'viz_recommend_result'
plot(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | A `viz_recommend_result` object returned by `viz_recommend()`. |
| `...` | Additional arguments passed to `gridExtra::grid.arrange()` (e.g. `ncol`, `nrow`). |

## Details

Call `plot()` directly on the object returned by `viz_recommend()` – the data is stored automatically so you do not need to pass it again.

## Value

Invisibly returns `NULL`. Called for its side-effect of rendering the arranged plot grid.

## Examples

```
res <- viz_recommend(mtcars["mpg"])
plot(res)

# Paired mode -- scatter / hexbin arranged together
res2 <- viz_recommend(mtcars, x = "wt", y = "mpg")
plot(res2)
```

---

`print.viz_audit_report` *Print a viz_audit_report*

---

## Description

Prints each detected issue with its severity prefix and technical message, followed by a plain-English explanation (the `explanation` field) on the next line prefixed with `"In plain terms: "`.

## Usage

```
## S3 method for class 'viz_audit_report'
print(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | A `viz_audit_report` object returned by `viz_audit()`. |
| `...` | Ignored. |

## Value

`x`, invisibly.

---

`print.viz_recommend_result` *Print a viz_recommend_result*

---

## Description

Print a viz_recommend_result

## Usage

```
## S3 method for class 'viz_recommend_result'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | A viz_recommend_result object returned by viz_recommend(). |
| ... | Ignored. |

### Value

x, invisibly.

---

## summary.viz_audit_report *Summarise a viz_audit_report*

---

### Description

Summarise a viz_audit_report

### Usage

```
## S3 method for class 'viz_audit_report'
summary(object, ...)
```

### Arguments

| | |
|---|---|
| object | A viz_audit_report object returned by viz_audit(). |
| ... | Ignored. |

### Value

object, invisibly.

---

## viz_accessibility *Check visualization accessibility*

---

### Description

Checks an existing ggplot for common visualization accessibility and readability issues, including excessive colours, colour-only encoding, excessive shapes, small text, excessive text labels, and overplotting.

### Usage

```
viz_accessibility(plot)
```

### Arguments

| | |
|---|---|
| plot | A ggplot object to check for common accessibility issues. |

## Value

A list containing a `summary` and an `issues` element. Each issue contains a severity level, check name, and descriptive message.

## Examples

```
library(ggplot2)

p <- ggplot(mtcars, aes(x = wt, y = mpg, colour = factor(cyl))) +
  geom_point()

viz_accessibility(p)
```

---

# viz_alternatives *Suggest alternative visualizations*

---

## Description

Suggests and generates alternative visualization types for an existing ggplot object. The generated alternatives are displayed via the active graphics device and are also returned for further use. Informational messages about the alternatives can be suppressed with `suppressMessages()`.

## Usage

```
viz_alternatives(plot)
```

## Arguments

| | |
|---|---|
| `plot` | A ggplot object. |

## Value

A list containing:

`recommendations` Character vector of suggested alternative visualization names.

`plots` A named list of generated `ggplot` objects for the alternatives.

## Examples

```
library(ggplot2)

p <- ggplot(mtcars, aes(wt, mpg)) +
  geom_point()

result <- suppressMessages(viz_alternatives(p))
result$recommendations
```

`viz_audit` *Audit a ggplot object for common visualization integrity problems*

## Description

Inspects a `ggplot` object by rendering it internally via `ggplot2::ggplot_build()` and running four integrity checks drawn from data-visualization best practices (Tufte, Cairo, Wilke). Returns a structured S3 report with one entry per detected issue.

## Usage

```
viz_audit(plot)
```

## Arguments

| | |
|---|---|
| `plot` | A `ggplot` object to audit. |

## Details

### Checks performed:

| ID | Check | Triggers | S |
|---|---|---|---|
| `truncated_axis` | Y-axis does not include 0 on a bar/column plot | Bar/col geom + displayed y-min > min(0, data min) | " |
| `lie_factor` | Displayed axis range distorts data range | Range ratio outside [0.95, 1.05] | " |
| `overplotting` | Many points overlap in a scatter plot | Point/jitter geom + collision fraction > 0.30 | " |
| `dual_y_axis` | A secondary y-axis is defined | `sec.axis` present on y scale | " |

### Lie factor note:

The lie factor is computed in **data space** (displayed axis range vs actual data range), not pixel space. Pixel-based computation requires a live graphics device and is not reproducible in a non-interactive context.

### Plain-English explanations:

Every issue includes an `explanation` field with a plain-English description of what the problem means for how misleading the chart looks. This is printed after the technical message when you call `print()` on the report.

## Value

An S3 object of class `"viz_audit_report"`, which is a list with:

`n_issues` Total number of issues detected (integer).

`n_critical` Number of critical issues (integer).

`n_warning` Number of warning-level issues (integer).

`n_info` Number of info-level issues (integer).

`issues` A list of issue lists, each with fields `check`, `severity`, `message`, `details`, and `explanation`.

`plot_geoms` Character vector of geom class names in the plot.

## Examples

```
library(ggplot2)

# A bar chart with a truncated y-axis -- should flag "truncated_axis"
bad <- ggplot(data.frame(x = c("A", "B"), y = c(50, 80)), aes(x, y)) +
  geom_col() +
  coord_cartesian(ylim = c(40, 85))
report <- viz_audit(bad)
print(report)

# A clean scatter plot -- should return no critical issues
good <- ggplot(mtcars, aes(wt, mpg)) + geom_point()
viz_audit(good)
```

---

`viz_explore` *Explore Relationships Between Numeric Variables*

---

## Description

Explores relationships between numeric variables using correlation analysis and a heatmap.

## Usage

```
viz_explore(data)
```

## Arguments

| | |
|---|---|
| `data` | A data frame or tibble containing the variables to explore. |

## Value

A list containing:

- `correlation`: A correlation matrix.
- `plot`: A ggplot2 correlation heatmap.

## Examples

```
result <- viz_explore(mtcars)
result$correlation

result$plot
```

`viz_recommend` *Recommend chart types for a data frame's columns*

## Description

Calls `profile_data()` internally to understand each column, then scores candidate chart types from an internal taxonomy against the column profile and any goal or variable hint the user provides. Returns a ranked list of recommendations, highest scoring first.

## Usage

```
viz_recommend(df, goal = NULL, x = NULL, y = NULL)
```

## Arguments

| | |
|---|---|
| `df` | A `data.frame` or tibble with at least one column. |
| `goal` | One of `"distribution"`, `"compare"`, `"trend"`, `"relationship"`, `"composition"`, or `NULL` (default). When `NULL`, the best-fitting goal is inferred from the column types. |
| `x` | Name of the column to use as the x variable (character). When supplied together with `y`, a paired recommendation is returned for that specific combination instead of per-column recommendations. |
| `y` | Name of the column to use as the y variable (character). See `x`. |

## Details

**How scoring works:**

An internal taxonomy table lists chart types alongside the conditions under which each chart is a good fit (variable types, goal, skewness, cardinality, data size). Each applicable chart starts with a base score and is adjusted:

- +0.15 if the goal matches exactly rather than being a generic match
- -0.20 if the data is skewed (|skewness| > 1) and the chart does not handle skew well
- -0.20 if the column has more than 15 unique categories and the chart becomes unreadable at that cardinality
- -0.15 if there are more than 1000 rows and the chart does not scale well

Charts with a final score below 0.10 are dropped. The top 3 per variable/pair are returned.

**Taxonomy overview:**

| Chart | Best for |
|---|---|
| `histogram` | Single numeric, distribution |
| `log_histogram` | Skewed numeric (suggests log scale) |
| `density` | Single numeric, smooth distribution shape |
| `boxplot` | Numeric + categorical group comparison |
| `violin` | Same as boxplot, better for larger samples |

| | |
|---|---|
| bar | Categorical counts or low-cardinality cat vs numeric |
| lollipop | High-cardinality categorical (bar gets overcrowded) |
| scatter | Numeric vs numeric, relationship |
| hexbin | Numeric vs numeric with many points (overplotting risk) |
| line | Datetime vs numeric, trend over time |
| heatmap | Two categoricals or categorical vs datetime |

## Value

A list of class "viz_recommend_result". Each element is a list of class "viz_recommendation" with fields:

variable Column name, or "x vs y" string for pairs.

chart Chart type string.

fit_score Numeric in [0, 1]. Higher is better.

rationale One sentence explaining why this chart fits.

goal The goal used when scoring this recommendation.

The result also has attributes df, x, and y (the data and variable names stored so that plot() can be called on the result directly, without passing data again).

**Side effect:** when called in an interactive session, the recommended charts are automatically rendered in the Plots pane via plot.viz_recommend_result(). You can also call plot() manually on the returned object at any time.

## Examples

```
# Per-column recommendations for the whole data frame
viz_recommend(mtcars)


# Two specific columns as a pair
viz_recommend(mtcars, x = "wt", y = "mpg")

# Force a goal
viz_recommend(mtcars["mpg"], goal = "distribution")
```

# Index

ggplot2::ggplot_build(), *6*
gridExtra::grid.arrange(), *2*

plot.viz_recommend_result, 2
print.viz_audit_report, 3
print.viz_recommend_result, 3

summary.viz_audit_report, 4

viz_accessibility, 4
viz_alternatives, 5
viz_audit, 6
viz_explore, 7
viz_recommend, 8
viz_recommend(), *2*