universität innsbruck

# tinyplot

## Convenient and Customizable Base R Plots

Grant McDermott, Vincent Arel-Bundock, Achim Zeileis

https://grantmcdermott.com/tinyplot/

# Motivation

Ross Ihaka & Robert Gentleman (1996)

R: A Language for Data Analysis *and Graphics*

**Engines:** Base `graphics` vs. newer flexible `grid` (enabling `ggplot2` and `lattice`).

**Core of base graphics:** `plot()` generic function and corresponding methods.

**Default method:** Handles many basic plotting elements like points, lines, etc.

**Formula method:** Handles various `y ~ x` setups.

- Scatterplots (numeric `y` vs. numeric `x`).
- Boxplots (numeric `y` vs. categorical `x`).
- Spineplots/spinograms (categorical `y`).

# Motivation

**Illustration:** Determinants of student performance in end-term exam of an introductory mathematics course for business and economics students at Universität Innsbruck.

# Motivation

**Illustration:** Determinants of student performance in end-term exam of an introductory mathematics course for business and economics students at Universität Innsbruck.

```
data("MathExam14W", package = "psychotools")
math <- MathExam14W |>
  transform(
    attempt = factor(attempt, ordered = FALSE),
    points = rowSums(as.matrix(credits)^2 * (-1)^as.matrix(credits))
  ) |>
  transform(
    pass = factor(points >= 26, labels = c("fail", "pass")),
    score = points/52
  ) |>
  subset(select = c("score", "pass", "tests", "attempt", "gender"))
```

# Motivation

**Dependent variables:**

- Numeric `score` (proportion of points).
- Binary indicator for `pass`-ing the exam (`score >= 0.5`).

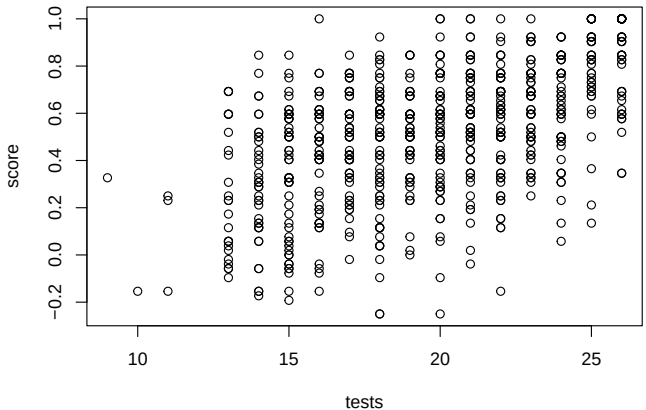
**Explanatory variables:** Points from previous online `tests`, `attempt`, `gender`.

```
summary(math)
```

```
     score            pass         tests          attempt     gender
 Min.   :-0.2500   fail:281   Min.   : 9.00   1:431   female:326
 1st Qu.: 0.3269   pass:448   1st Qu.:17.00   2: 52   male  :403
 Median : 0.5385              Median :20.00   3:121
 Mean   : 0.5194              Mean   :19.63   4:113
 3rd Qu.: 0.6923              3rd Qu.:23.00   5: 12
 Max.   : 1.0000              Max.   :26.00
```

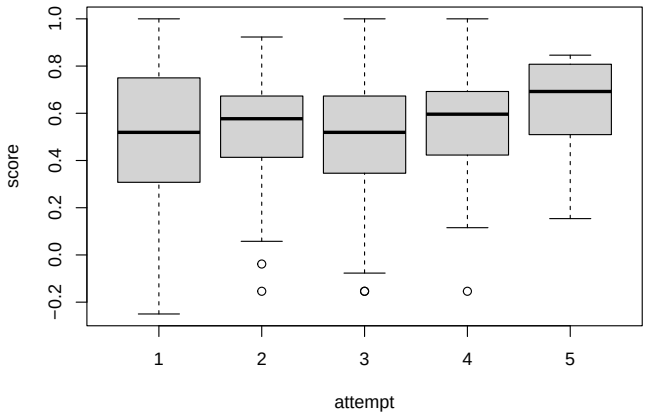
# Motivation: numeric ~ numeric

```
plot(score ~ tests, data = math)
```



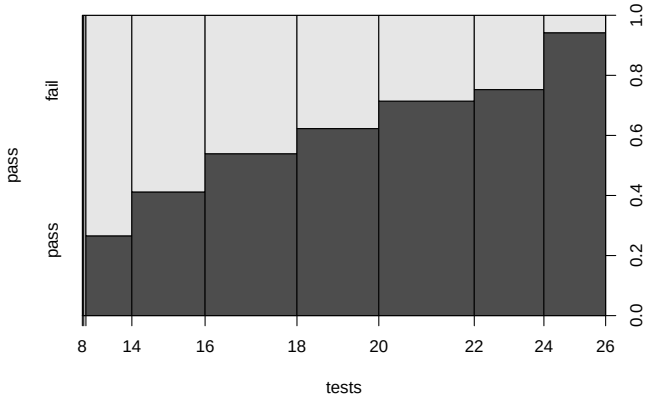
# Motivation: numeric ~ categorical

```
plot(score ~ attempt, data = math)
```



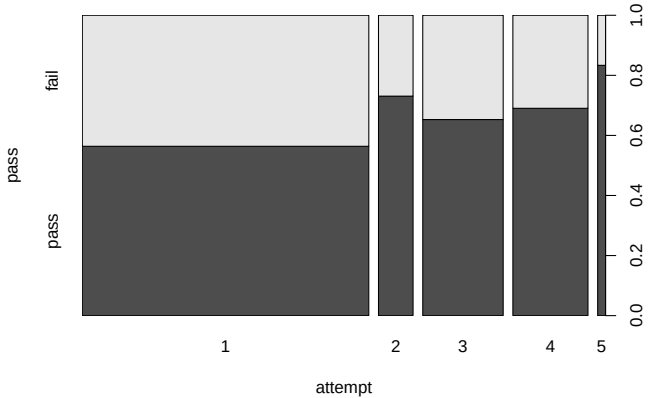
# Motivation: categorical ~ numeric

```
plot(pass ~ tests, data = math)
```



# Motivation: categorical ~ categorical

```
plot(pass ~ attempt, data = math)
```



# Motivation: Limitations

**So far:**

- Nifty data visualizations.
- Intuitive, concise syntax.

**Possible customizations:**

- Groups via shading, symbols, line types, etc.
- Legends, axes, annotation.
- Grid of faceted displays.
- Layers with additional elements.

**But:**

- Requires low-level drawing of such elements.
- Tedious without intuitive, concise syntax.

# tinyplot

Install:

```
install.packages("tinyplot") #or#
install.packages("tinyplot",
  repos = "https://grantmcdermott.R-universe.dev")
```


Load:

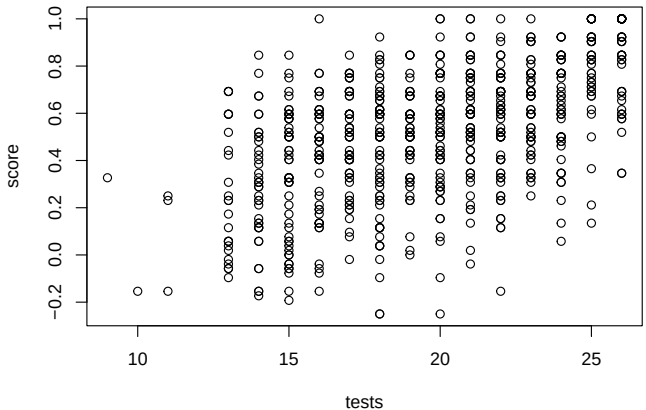
```
library("tinyplot")
```

**Starting point**

`tinyplot()` or its shorthand `plt()` as drop-in replacement for `plot()`.

# tinyplot

```
tinyplot(score ~ tests, data = math)
```



# tinyplot: Features

**Core ideas:**

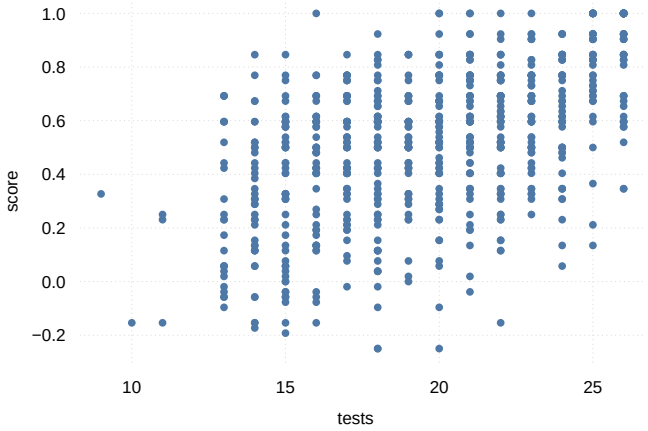
- Preservation of strengths of base R graphics.
- Lightweight extension with convenience features.
- No strong dependencies on non-base packages.
- Improved feature parity vs. `grid`-based `ggplot2` and `lattice`.
- Grouped plots with automatic legends and/or facets.
- Advanced visualization types.
- Customization via themes.

**Here:**

```
tinytheme("clean2")
```

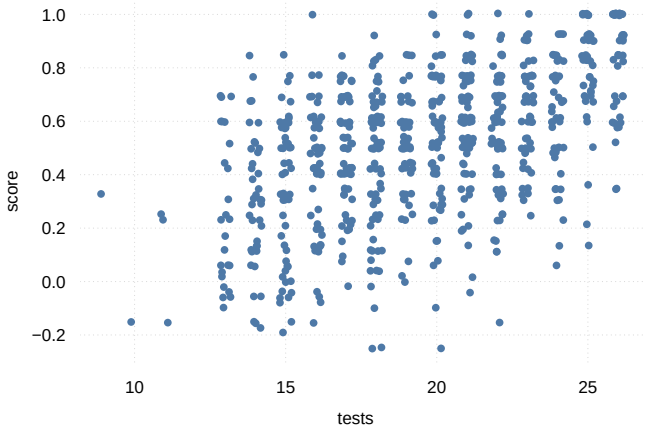
# tinyplot: Themes

```
tinyplot(score ~ tests, data = math)
```



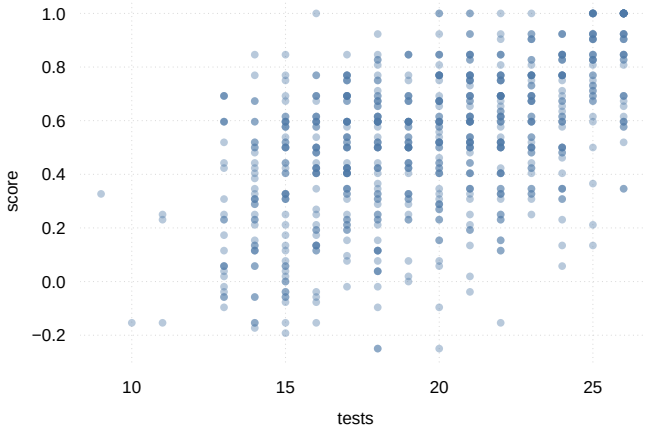
# tinyplot: More plot types

```
tinyplot(score ~ tests, data = math, type = "jitter")
```



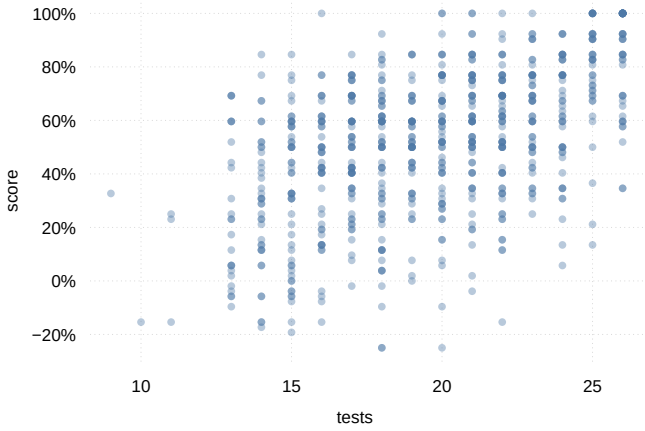
# tinyplot: Alpha transparency

```
tinyplot(score ~ tests, data = math, alpha = 0.4)
```



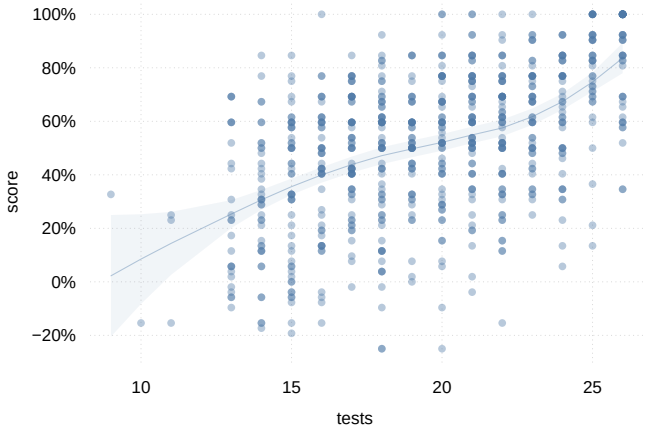
# tinyplot: Axis labels

```
tinyplot(score ~ tests, data = math, alpha = 0.4, yaxl = "%")
```



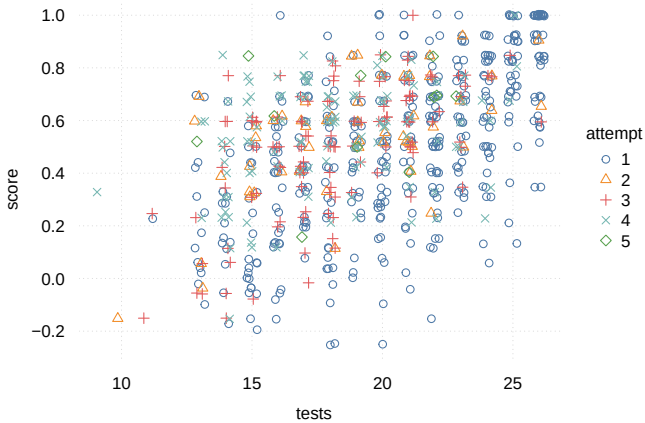
# tinyplot: Add layers

```
tinyplot(score ~ tests, data = math, alpha = 0.4, yaxl = "%")
tinyplot_add(type = "loess")
```



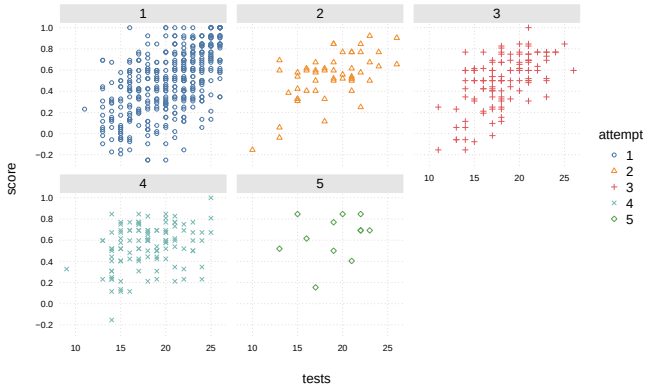
# tinyplot: Grouped plots with legends

```
tinyplot(score ~ tests | attempt, data = math, pch = "by", type = "jitter")
```



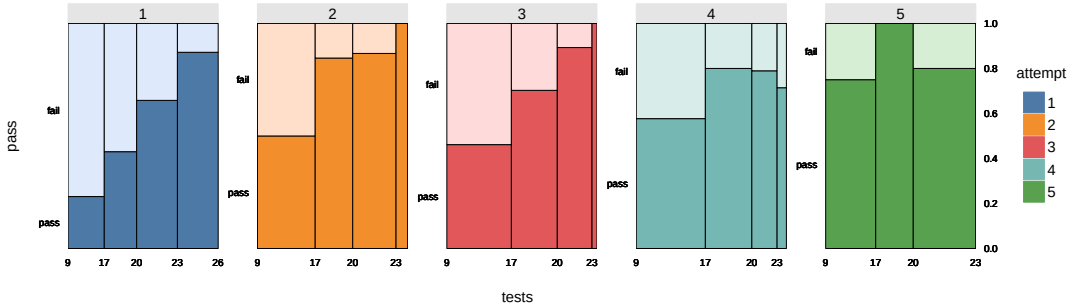
# tinyplot: Facets

```
tinyplot(score ~ tests | attempt, data = math, pch = "by", facet = "by")
```



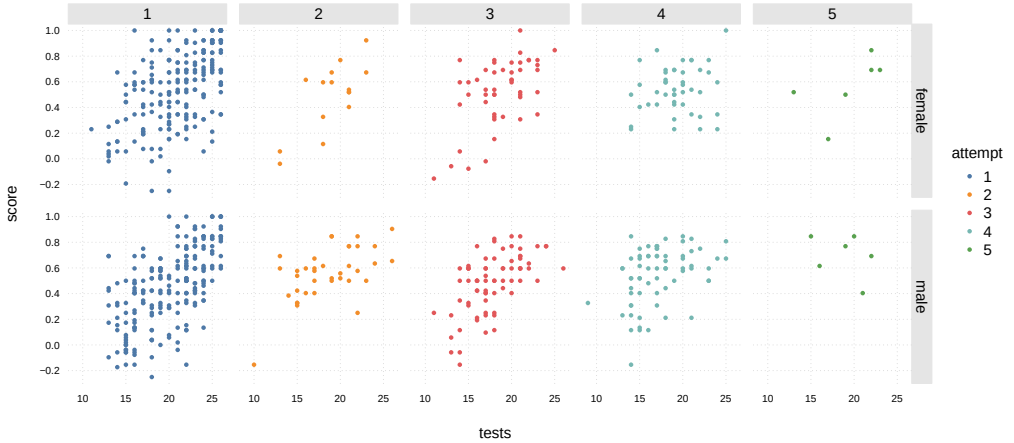
# tinyplot: Facets

```
tinyplot(pass ~ tests | attempt, data = math, facet = "by",
  breaks = c(9, 17, 20, 23, 26), facet.args = list(nrow = 1))
```



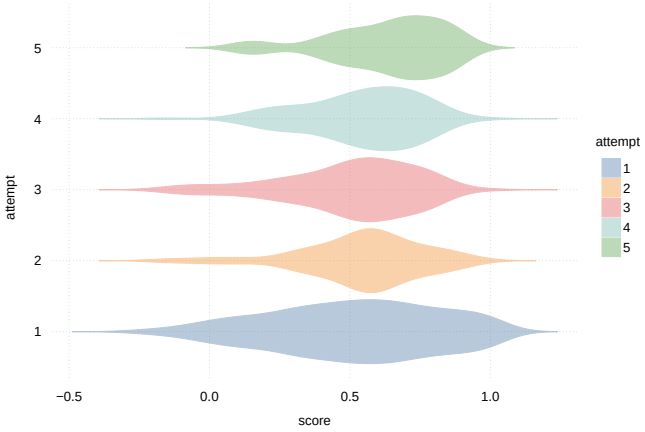
# tinyplot: Facets

```
tinyplot(score ~ tests | attempt, data = math, facet = gender ~ attempt)
```



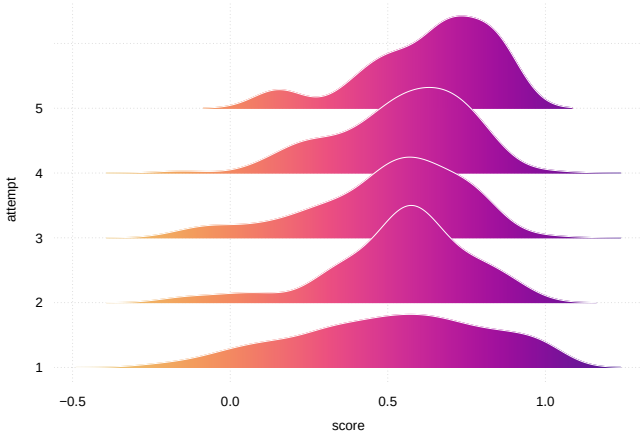
# tinyplot: More plot types

```
tinyplot(score ~ attempt | attempt, data = math,
  type = "violin", flip = TRUE, alpha = 0.4)
```



# tinyplot: More plot types

```
tinyplot(attempt ~ score, data = math, bg = "white",
  type = type_ridge(gradient = TRUE, col = "white"))
```



# tinyplot: More plot types

**Interface:** Types can be passed as either a *string* or *function*.

| | | | | |
|---|---|---|---|---|
| *String* | `"p"` | `"ridge"` | `"loess"` | `...` |
| *Function* | `type_points()` | `type_ridge()` | `type_loess()` | `...` |

**Arguments:** Can always be passed through the `type` function.

```
tinyplot(..., type = type_ridge(gradient = TRUE))
```

**Alternatively:** Through `tinyplot()` if there is no clash with top-level arguments.

```
tinyplot(..., type = "ridge", gradient = TRUE)
```

# Use cases

**Teaching and interactive usage:**

- Start with simple visualizations and fundamental principles in base graphics.
- Proceed to more complex display using the same type of interface.

**Package development:**

- Create flexible visualizations without introducing numerous dependencies.

**Web R:**

- Engine for appealing graphics without adding much overhead.

# References

McDermott G, Arel-Bundock V, Zeileis A (2025). *tinyplot: Lightweight Extension of the Base R Graphics System*. R package version 0.6.0.
doi:10.32614/CRAN.package.tinyplot

Zeileis A (2025). "Examining Exams Using Rasch Models and Assessment of Measurement Invariance." *Austrian Journal of Statistics*, **54**(3), 9-26.
doi:10.17713/ajs.v54i3.2055

**Mastodon:** @zeileis@fosstodon.org

**Bluesky:** @zeileis.org

**Web:** https://www.zeileis.org/