# Visual Forms

S. Santoni[1]

[1]Bayes Business School

MSc in Business Analytics, 2024/25

# Outline

# Outline

# Grammar of Graphics

- The Grammar of Graphics is an analytical framework that considers the design of a chart as a 'bundle of choices'
- Leland Wilkinson proposed GoF in 1999 [4]
- Since then, many data visualization textbooks and packages have built on it
- GoF advantages:
  - GoF facilitates reasoning about a chart design
  - GoF is a discursive tool that help individuals discuss charts

# Software Adopting GoF

## Python's Plotnine

plotnine 0.13.6 Reference Gallery Blog Tutorials Pages

### A Grammar of Graphics for Python

On this page
Example
Installation

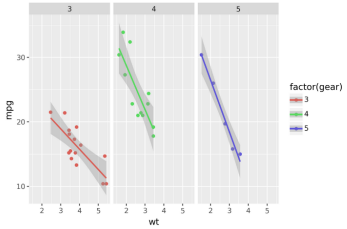
plotnine is an implementation of a *grammar of graphics* in Python based on ggplot2. The grammar allows you to compose plots by explicitly mapping variables in a dataframe to the visual objects that make up the plot.


Plotting with a *grammar of graphics* is powerful. Custom (and otherwise complex) plots are easy to think about and build incrementally, while the simple plots remain simple to create.

### Example

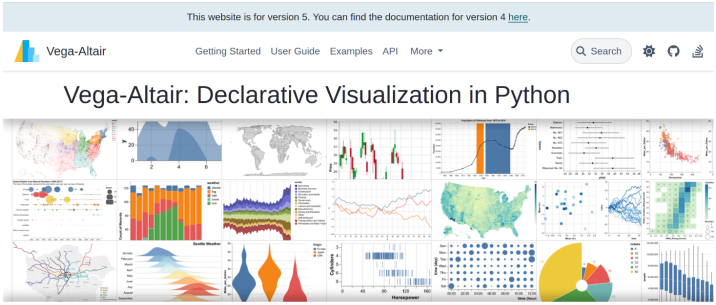
```
from plotnine import ggplot, geom_point, aes, stat_smooth, facet_wrap
from plotnine.data import mtcars

(
    ggplot(mtcars, aes("wt", "mpg", color="factor(gear)"))
    + geom_point()
    + stat_smooth(method="lm")
    + facet_wrap("gear")
)
```



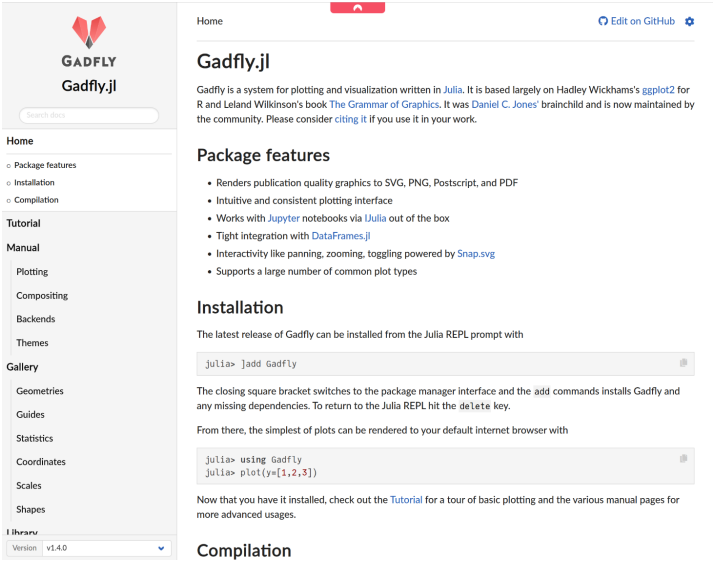
# Software Adopting GoF

## Python's Vega-Altair



# Software Adopting GoF

## Julia's Gadfly



GADFLY

Gadfly.jl

Search docs

Home
- Package features
- Installation
- Compilation

Tutorial

Manual
- Plotting
- Compositing
- Backends
- Themes

Gallery
- Geometries
- Guides
- Statistics
- Coordinates
- Scales
- Shapes

Library

Version v1.4.0

Home

Edit on GitHub

# Gadfly.jl

Gadfly is a system for plotting and visualization written in Julia. It is based largely on Hadley Wickhams's ggplot2 for R and Leland Wilkinson's book The Grammar of Graphics. It was Daniel C. Jones' brainchild and is now maintained by the community. Please consider citing it if you use it in your work.

## Package features

- Renders publication quality graphics to SVG, PNG, Postscript, and PDF
- Intuitive and consistent plotting interface
- Works with Jupyter notebooks via IJulia out of the box
- Tight integration with DataFrames.jl
- Interactivity like panning, zooming, toggling powered by Snap.svg
- Supports a large number of common plot types

## Installation

The latest release of Gadfly can be installed from the Julia REPL prompt with

```
julia> ]add Gadfly
```

The closing square bracket switches to the package manager interface and the `add` commands installs Gadfly and any missing dependencies. To return to the Julia REPL hit the `delete` key.

From there, the simplest of plots can be rendered to your default internet browser with

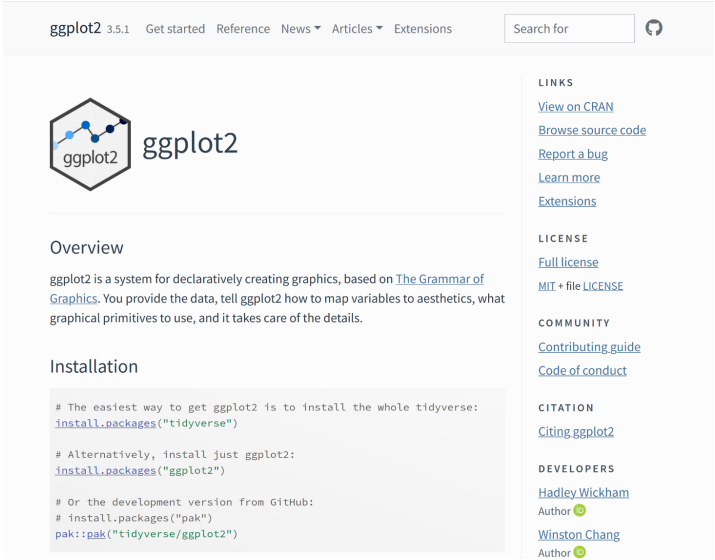
```
julia> using Gadfly
julia> plot(y=[1,2,3])
```

Now that you have it installed, check out the Tutorial for a tour of basic plotting and the various manual pages for more advanced usages.

## Compilation

# Software Adopting GoF

## R's ggplot2



ggplot2 3.5.1 Get started Reference News Articles Extensions Search for

ggplot2

### Overview

ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

### Installation

```
# The easiest way to get ggplot2 is to install the whole tidyverse:
install.packages("tidyverse")

# Alternatively, install just ggplot2:
install.packages("ggplot2")

# Or the development version from GitHub:
# install.packages("pak")
pak::pak("tidyverse/ggplot2")
```

LINKS

View on CRAN
Browse source code
Report a bug
Learn more
Extensions

LICENSE

Full license
MIT + file LICENSE

COMMUNITY

Contributing guide
Code of conduct

CITATION

Citing ggplot2

DEVELOPERS

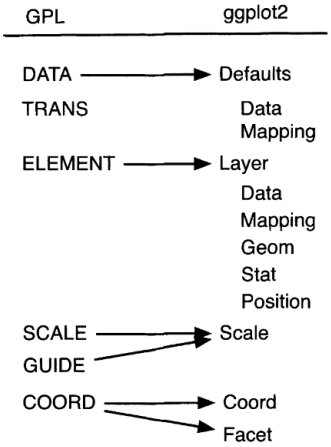
Hadley Wickham
Author

Winston Chang
Author

# Porting GoF to Specific Software

## Mapping between Wilkinson's GoF and ggplot2's GoF

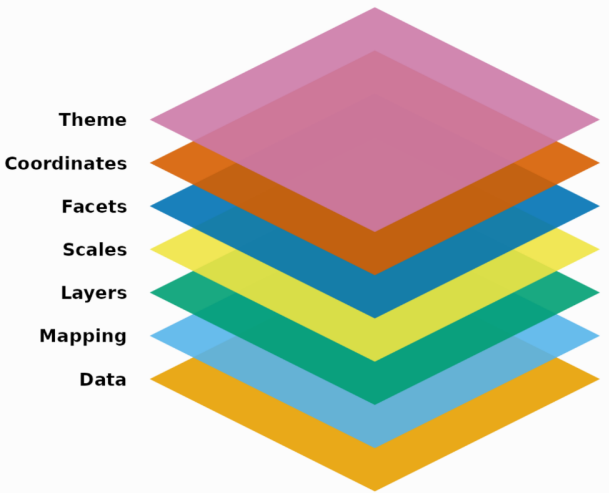





Source is [2]

# ggplot2's Internal Structure






Source is `https://ggplot2.tidyverse.org/articles/ggplot2.html`

# A Minimal ggplot2 Snippet



```
1  library(ggplot2)
2
3  ggplot(
4    data = TIDYDATA,
5    mapping = aes(x = COL_1, y = COL_2, colour = COL_3)
6    ) +
7    geom_point()
8
```

# Data



*"The system works best if the data is provided in a tidy format, which briefly means a rectangular data frame structure where rows are observations and columns are variables."*

*"As the first step in many plots, you would pass the data to the ggplot() function, which stores the data to be used later by other parts of the plotting system."*

```
1 ggplot(data = TIDYDATA)
2
```

# Mapping



*"The mapping of a plot is a set of instructions on how parts of the data are mapped onto aesthetic attributes of geometric objects. It is the 'dictionary' to translate tidy data to the graphics system."*

```
1 ggplot(data = TIDYDATA, mapping = aes(x = COL_1, y = COL_2))
2
```

# Layers

*"The heart of any graphic is the layers. They take the mapped data and display it in something humans can understand as a representation of the data. Every layer consists of three important parts:*

1. *The geometry that determines how data are displayed, such as points, lines, or rectanglesThe*
2. *The statistical transformation that may compute new variables from the data and affect what of the data is displayed*
3. *The position adjustment that primarily determines where a piece of data is being displayed."*

# Outline

# What Are the Core Visual Forms in Data Visualization?

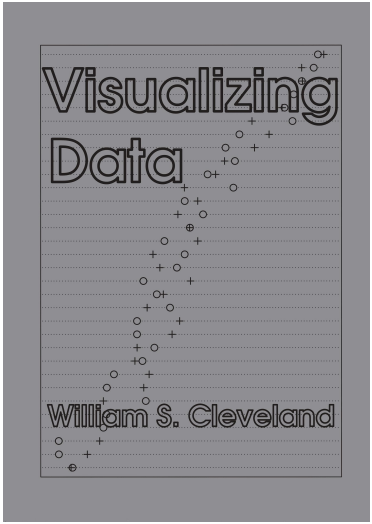


# A Taxonomy of Visual Forms



Cleveland [1] proposes a taxonomy of the visual forms based on the cardinality of a plot's **mapping** $\Phi$:

- $||\Phi|| = 1 \rightarrow$ univariate plots
- $||\Phi|| = 2 \rightarrow$ bivariate plots
- $||\Phi|| = 3 \rightarrow$ trivariate plots
- $||\Phi|| \geq 4 \rightarrow$ hypervariate plots



# Outline

# A Toy Dataset

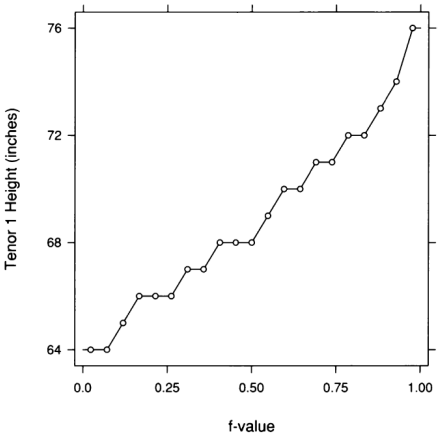
'Singer' – singer's height and vocal extension data (N=235)



| **Height** | **Voice part** |
|---|---|
| 64 | Soprano 1 |
| 62 | Soprano 2 |
| 65 | Alto 2 |
| 67 | Alto 1 |
| 72 | Tenor 1 |
| 69 | Tenor 2 |
| 75 | Basso 1 |
| 74 | Basso 2 |
| ... | ... |

# Quantile Plot

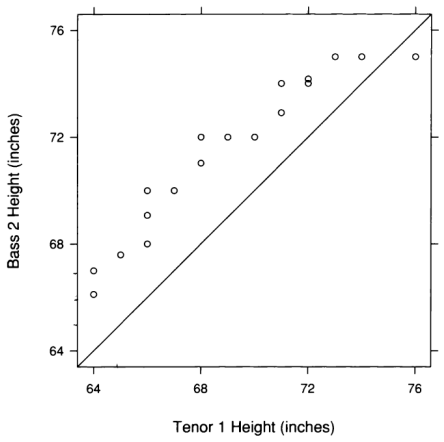





- Quantile plots are essential to visualize a distribution, i.e., the collection of positions in the data
- The quantile $f$, $q(f)$, of a set of data is a value along the measurement scale of the data with the property that approximately a fraction of the data are less than of equl to $q(f)$
- The property has to be approximate because there might not be a value with exactly a fraction / of the data less than or equal to it

# Quantile-Quantile Plot

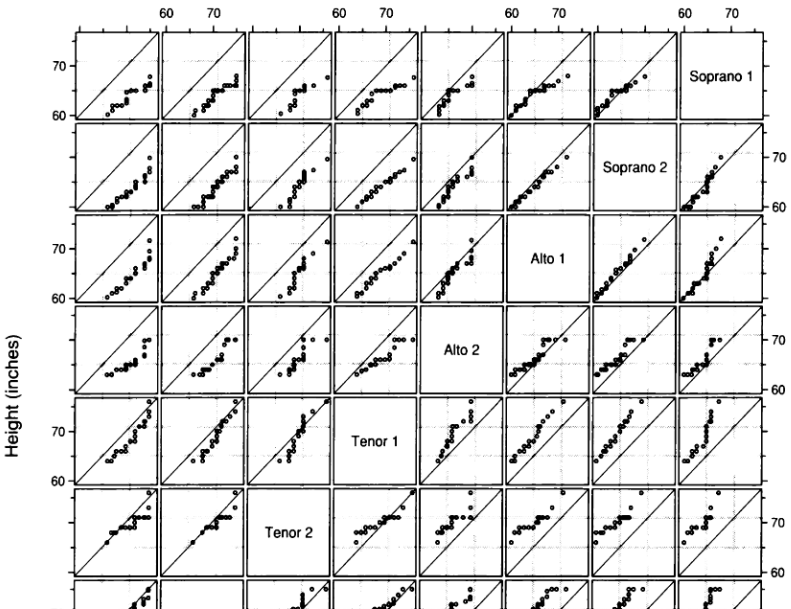





- The quantile-quantile plot, or q-q plot [3], is a powerful visualization method for comparing the distributions of two or more sets of univariate measurements
- When distributions are compared, the goal is to understand how the distributions shift in going from one data set to the next
- For the singers, the goal is to understand how the height distributions shift with voice part

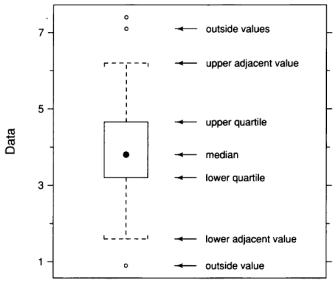
# A QQ Plot Extension: The Pairwise QQ Plot





# The Boxplot




Data
7
5
3
1
outside values
upper adjacent value
upper quartile
median
lower quartile
lower adjacent value
outside value

# Outline

# Outline

# Outline

# Outline

# Time to wrap up!

# References



[1] William S. Cleveland. *Visualizing Data*. Murray Hill, N.J. : [Summit, N.JclevelandVisualizingData1993: At&T Bell Laboratories ; Published by Hobart Press, 1993. 360 pp.

[2] Hadley Wickham. "A Layered Grammar of Graphics". In: *Journal of Computational and Graphical Statistics* 19.1 (Jan. 2010), pp. 3–28.

[3] Martin B Wilk and Ram Gnanadesikan. "Probability plotting methods for the analysis of data". In: *Biometrika* 55.1 (1968), pp. 1–17.

[4] Leland Wilkinson. *The Grammar of Graphics*. Springer, 2012.