Rethinking your plotting habits

An introduction to the Grammar of Graphics
Aim

Make plots / visualisations of **data:**

- reproducible

- more flexible for exploration

- publication-ready (no editing by hand)

- principles apply across different languages: MATLAB, PYTHON, R, JULIA, …

- and different kinds of data: fMRI, EEG, psychophys, …
The ideas

A Layered Grammar of Graphics

Hadley Wickham

A grammar of graphics is a tool that enables us to concisely describe the components of a graphic. Such a grammar allows us to move beyond named graphics (e.g., the "scatterplot") and gain insight into the deep structure that underlies statistical graphics. This article builds on Wilkinson, Anand, and Grossman (2005), describing extensions and refinements developed while building an open source implementation of the grammar of graphics for R, ggplot2.

The topics in this article include an introduction to the grammar by working through the process of creating a plot, and discussing the components that we need. The grammar is then presented formally and compared to Wilkinson’s grammar, highlighting the hierarchy of defaults, and the implications of embedding a graphical grammar into a programming language. The power of the grammar is illustrated with a selection of examples that explore different components and their interactions, in more detail. The article concludes by discussing some perceptual issues, and thinking about how we can build on the grammar to learn how to create graphical “poetry.”

Key Words: Grammar of graphics, Statistical graphics.

1. INTRODUCTION

What is a graphic? How can we succinctly describe a graphic? And how can we create the graphic that we have described? These are important questions for the field of statistical graphics.

One way to answer these questions is to develop a grammar “the fundamental principles or rules of an art or science” (OED Online 1989). A good grammar will allow us to gain insight into the composition of complicated graphics, and reveal unexpected connections between seemingly different graphics (Cox 1978). A grammar provides a strong foundation for understanding a diverse range of graphics. A grammar may also help us to distinguish between a well-formed or correct graphic looks like, but there will still be many grammatically correct but nonsensical graphics. This is easy to see by analogy to the English language: good grammar is just the first step in creating a good sentence.

© 2010 American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America

J. Comput. Graph. Stat., Volume 19, Number 1, Pages 3–28
DOI: 10.1198/jcgs.2009.07098

Wickham (2010)
The ideas

A Layered Grammar of Graphics

Hadley Wickham

A grammar of graphics is a tool that enables us to concisely describe the components of a graphic. Such a grammar allows us to move beyond named graphics (e.g., the "scatterplot") and gain insight into the deep structure that underlies statistical graphics. This article builds on Wilkinson, Anand, and Grossman (2005), describing extensions and refinements developed while building an open source implementation of the grammar of graphics for R, ggplot2.

The topics in this article include an introduction to the grammar by working through the process of creating a plot, and discussing the components that we need. The grammar is then presented formally and compared to Wilkinson’s grammar, highlighting the hierarchy of defaults, and the implications of embedding a graphical grammar into a programming language. The power of the grammar is illustrated with a selection of examples that explore different components and their interactions, in more detail. The article concludes by discussing some perceptual issues, and thinking about how we can build on the grammar to learn how to create graphical “poems.”

Supplemental materials are available online.

Key Words: Grammar of graphics, Statistical graphics.

1. INTRODUCTION

What is a graphic? How can we succinctly describe a graphic? And how can we create the graphic that we have described? These are important questions for the field of statistical graphics.

One way to answer these questions is to develop a grammar: “the fundamental principles or rules of an art or science” (OED Online 1989). A good grammar will allow us to gain insight into the composition of complicated graphics, and reveal unexpected connections between seemingly different graphics (Cox 1978). A grammar provides a strong foundation for understanding a diverse range of graphics. A grammar may also help us on what a well-formed or correct graphic looks like, but there will still be many grammatically correct but nonsensical graphics. This is easy to see by analogy to the English language: good grammar is just the first step in creating a good sentence.

Hadley Wickham is Assistant Professor of Statistics, Rice University, Houston, TX 77030 (E-mail: h.wickham@gmail.com).

© 2010 American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America

Journal of Computational and Graphical Statistics, Volume 19, Number 1, Pages 3–28
DOI: 10.1198/jcgs.2009.07098

Wickham (2010)
imperative

https://uk.mathworks.com/products/matlab/plot-gallery.html
**imperative**

- decide on plot type, build plot step by step, ...

**declarative**

- **–ve:** leads to repetitive code / work
instead: common framework

geometry
aesthetics
data
instead: common framework

coordinates
stats
facets
scales
geometry
aesthetics
data
instead: common framework
an example

Figure 3b

recreating this graphic in R/ggplot2

code: https://gist.github.com/schluppeck/9a54b9b7a37793d8959779629b4cd2fc
data, d

head(d)

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>subject</th>
<th>coherence</th>
<th>absent</th>
<th>present</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>4</td>
<td>0.035996 0.087079</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td>7</td>
<td>0.026042 0.056272</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td>13</td>
<td>0.028604 0.047791</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td>25</td>
<td>0.029221 0.063149</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>4</td>
<td>0.025157 0.096832</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td></td>
<td>7</td>
<td>0.041558 0.091160</td>
</tr>
</tbody>
</table>

4 variables that we want to map into a plot
aesthetics

- x, y (position)
- alpha, color, fill
- size
- shape
- linetype

http://docs.ggplot2.org/current/vignettes/ggplot2-specs.html
**data**

```r
head(d)
```

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0.035996</td>
<td>0.087079</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>0.026042</td>
<td>0.056272</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>13</td>
<td>0.028604</td>
<td>0.047791</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>25</td>
<td>0.029221</td>
<td>0.063149</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>0.025157</td>
<td>0.096832</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>0.041558</td>
<td>0.091160</td>
</tr>
</tbody>
</table>

4 variables that we want to **map** into a plot
data

head(d)

<table>
<thead>
<tr>
<th>X</th>
<th>subject</th>
<th>coherence</th>
<th>absent</th>
<th>present</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0.035996 0.087079</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>0.026042 0.056272</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>13</td>
<td>0.028604 0.047791</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>25</td>
<td>0.029221 0.063149</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>0.025157 0.096832</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>0.041558 0.091160</td>
</tr>
</tbody>
</table>

4 variables that we want to **map** into a plot

x, y (position)
data

head(d)

X subject coherence absent present
1 1 1 4 0.035996 0.087079
2 2 2 7 0.026042 0.056272
3 3 3 13 0.028604 0.047791
4 4 4 25 0.029221 0.063149
5 5 5 4 0.025157 0.096832
6 6 1 7 0.041558 0.091160

4 variables that we want to map into a plot

x, y (position)
data

head(d)

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>subject</th>
<th>coherence</th>
<th>absent</th>
<th>present</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>4</td>
<td>0.035996 0.087079</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td>7</td>
<td>0.026042 0.056272</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td>13</td>
<td>0.028604 0.047791</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td>25</td>
<td>0.029221 0.063149</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>4</td>
<td>0.025157 0.096832</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td></td>
<td>7</td>
<td>0.041558 0.091160</td>
</tr>
</tbody>
</table>

4 variables that we want to map into a plot

- x, y (position)
- size
- color
geometry

- 0d: points, text
- 1d: lines, paths
- 2d: polygons, intervals

Wickham (2010)
geometry

- 0d: points, text
- 1d: lines, paths
- 2d: polygons, intervals
with default settings
themes / look of plot
worth the hassle?

- I think yes: already for basic plotting
- for data exploration we often slice across different dimensions:
  - subjects, regions of interest, …
  - measures: RT, % correct, fMRI response amplitude, …
facet (lattice)

Data for different subjects

coherence
- 4
- 7
- 13
- 25

subject
- 1
- 2
- 3
- 4
- 5
facet (lattice)

Data by coherence

coherence
- 4
- 7
- 13
- 25

subject
- 1
- 2
- 3
- 4
- 5
rearrange

subject

statistic

condition

absent

present
But I do use Matlab
# bash

```bash
cd ~/matlab
git clone https://github.com/piermorel/gramm
```

[ or download + extract zip file ]

% in matlab

```matlab
addpath(genpath(‘~/matlab/gramm’))
```

[ or put this in your startup.m file ]
using the same approach in *matlab/GRAMM*

code: https://gist.github.com/schluppeck/9a54b9b7a37793d8959779629b4cd2fc
% Load example dataset
load carbig.mat

% CREATE a gramm object with data -> AES
g=gramm('x', Model_Year,'y', . . . .)

% Plot raw data as points
g.geom_point()

% Do the actual drawing
g.draw()
Figure 1

Fuel economy of new cars between 1970 and 1982
figure
h=gramm('x',Model_Year,'y',.. . .)

h.geom_point()
% Plot linear fits of the data
% with associated confidence intervals
h.stat_glm()

% Subdivide the data in subplots
% horizontally by region of origin
h.facet_grid([],origin_region)

% and draw this one
h.draw()
Visualization of X densities

geom_raster()

stat_bin()

europe

# Cyl

Japan

# Cyl

USA

stat_density()

stat_qq()

europe

# Cyl

Japan

# Cyl

USA

Horsepower

Horsepower

Horsepower

-5

-5

To: https://github.com/piermorel/gramm
Visualization of repeated trajectories

- **geom_point()**
- **geom_line()**
- **stat_smooth()**
- **stat_summary()**

Color legend:
- Red: A
- Green: B
- Blue: C
https://github.com/piermorel/gramm
(tidy) data

head(e)

<table>
<thead>
<tr>
<th>X</th>
<th>subject</th>
<th>coherence</th>
<th>condition</th>
<th>statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>absent</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>absent</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>13</td>
<td>absent</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>25</td>
<td>absent</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>absent</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>absent</td>
</tr>
</tbody>
</table>

4 variables

side note: to fully make use of ggplot, it’s best if the data are tidy (for details see http://vita.had.co.nz/papers/tidy-data.pdf)