Graphing Techniques: Adjustment to Ocular Melanoma

Stanton Lab Meeting

Joshua Wiley

Department of Psychology

University of California, Los Angeles

wileyj@ucla.edu

16 February, 2011

Table of Contents

Introduction

- Visualization provides an intuitive interface to data
- Computational advances in the past two decades have made data visualization practical

Introduction

- Visualization provides an intuitive interface to data
- Computational advances in the past two decades have made data visualization practical
- Mainstream use has remained limited
- In a survey of wall posters in the UCLA Psychology department, $\frac{62}{67} = 92.5\%$ included at least one graph, $\frac{47}{62} = 75.8\%$ included bar plots and $\frac{26}{62} = 41.9\%$ were exclusively bar plots.

What makes a good graph?

A good graph...

- accurately and *clearly* conveys information
- is succinct, but does not waste space
- often does not require advanced training to understand

What makes a good graph?

A good graph...

- accurately and *clearly* conveys information
- is succinct, but does not waste space
- often does not require advanced training to understand
- **must be both accurately encoded and decoded**

Abstract Theory I

In Cleveland's model [1], graphs contain two types of information

1. Physical
   - Includes distance, shapes, colors, size, etc.
   - Visual system perceives these elements and detects patterns & relationships
   - From Gestalt Psychology, we know some patterns can be detected rapidly and accurately, but also at times inaccurately, though accurate information was encoded
   - Decoding other information may be slower and require sequential processing (often less accurate)

Abstract Theory II

2. Scale
    - Links physical data to something meaningful—visual functioning score; blue $=$ female, white $=$ male; "$+$" $=$ treatment, "o" $=$ control
    - Scales require looking up information in a table, legend, key, etc.
    - Scale information is typically held in short-term working memory
    - A complicated scale must be perceived piecewise, not as a whole

Concrete Theory

Aspects to consider in a graph

- **Discrimination**: are separate elements perceived separately?
- **Comparison**: can relative magnitudes be perceived (e.g., greater than, twice as high, etc.)

Concrete Theory

Aspects to consider in a graph

- **Discrimination**: are separate elements perceived separately?
- **Comparison**: can relative magnitudes be perceived (e.g., greater than, twice as high, etc.)
- **Texture (symbols)**: research has shown “o”, “+”, “<”, “s”, “w” are easily discriminated [1].
- **Color**: is there high contrast (discrimination)?

Concrete Theory

Aspects to consider in a graph

- **Discrimination**: are separate elements perceived separately?
- **Comparison**: can relative magnitudes be perceived (e.g., greater than, twice as high, etc.)
- **Texture (symbols)**: research has shown "o", "+", "<", "s", "w" are easily discriminated [1].
- **Color**: is there high contrast (discrimination)?
- **Order**: are categorical variables ordered or sorted to faciliate interpretation?
- **Scale**: is a common (i.e., consistent) scale used when possible?
- **Table**: is the table lookup easy & clear?

Table of Contents

Background

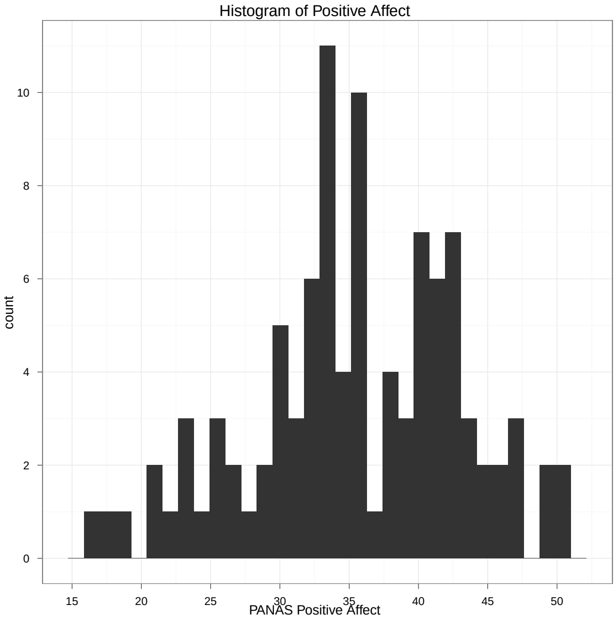
- There are many ways to visualize univariate distributions.
- **Histograms** use a binning algorithm and compute the frequency or density of points falling within each bin.
- **Density Plots** typically use a kernel density estimator[1] and provide information on density and the overall shape of the distribution.

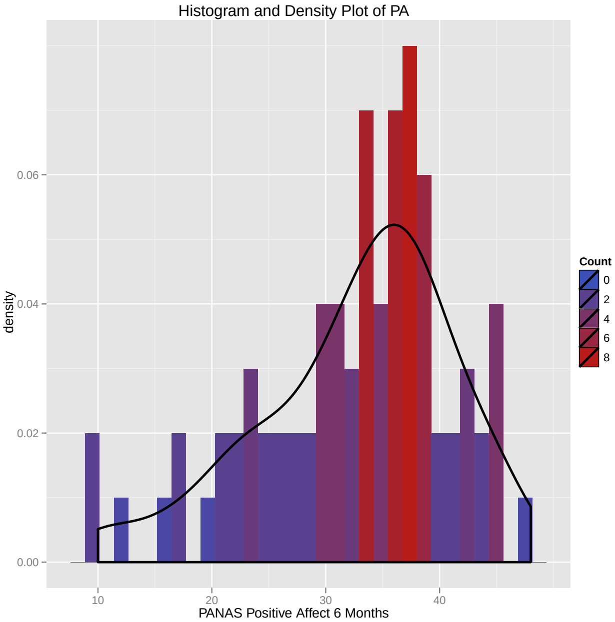
[1] There are numerous algorithms and options for degree of smoothing

Background

- There are many ways to visualize univariate distributions.
- **Histograms** use a binning algorithm and compute the frequency or density of points falling within each bin.
- **Density Plots** typically use a kernel density estimator[1] and provide information on density and the overall shape of the distribution.
- Both techniques can obscure data and change shape based on arbitrary bins or kernels.

[1] There are numerous algorithms and options for degree of smoothing


Histogram of Positive Affect
count
0
2
4
6
8
10
15
20
25
30
35
40
45
50
PANAS Positive Affect

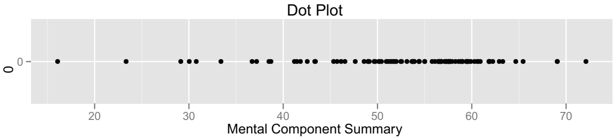

Histogram and Density Plot of PA
density
0.00
0.02
0.04
0.06
10
20
30
40
PANAS Positive Affect 6 Months
Count
0
2
4
6
8

Dot Plots

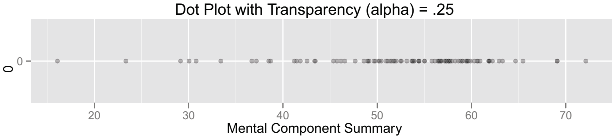
- Dot plots present an excellent method for visualizing distributions and density, but...

Dot Plots

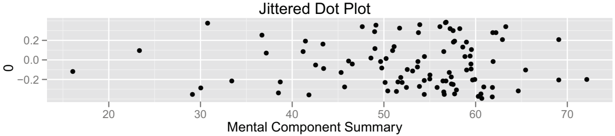
- Dot plots present an excellent method for visualizing distributions and density, but...
- Overplotting can make it difficult or impossible to see how many points fall in particular areas
- Common solutions:
  - Make points partially transparent
  - colour based on density
  - jitter points



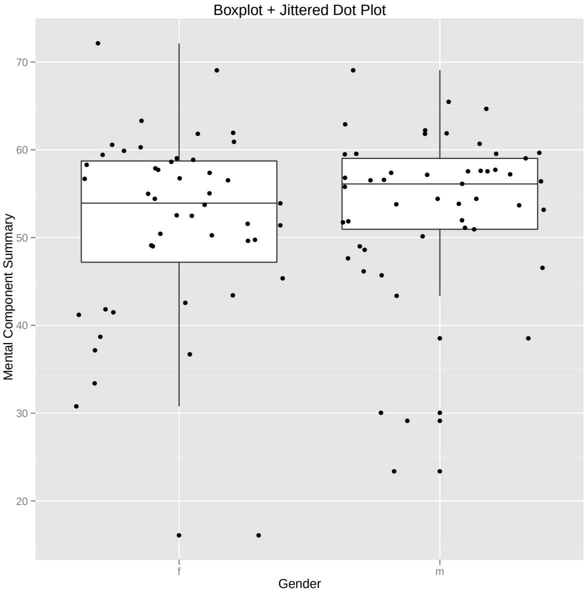
Dot Plot
0
0
20
30
40
50
60
70
Mental Component Summary



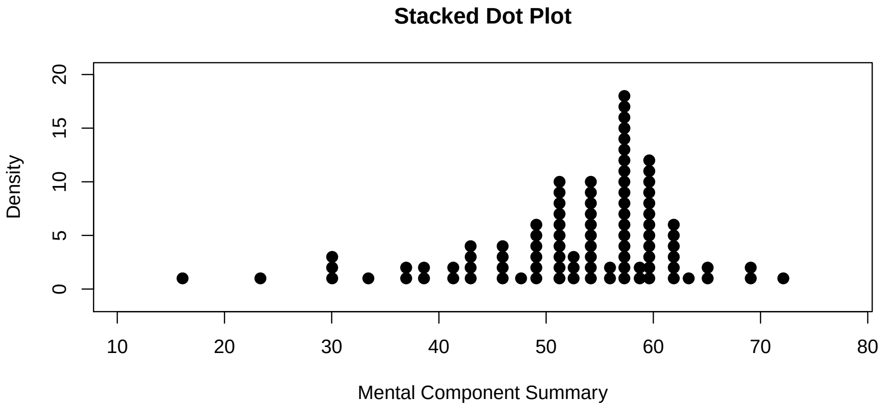
Dot Plot with Transparency (alpha) = .25
0
0
20
30
40
50
60
70
Mental Component Summary



Jittered Dot Plot
0
0.2
0.0
−0.2
20
30
40
50
60
70
Mental Component Summary


Boxplot + Jittered Dot Plot
Mental Component Summary
70
60
50
40
30
20
f
m
Gender

Stacked Dot Plot



Stacked Dot Plots

- Novel solution: displace orthogonally to the graphing plane (stack)
- I implemented an algorithm by Leland Wilkinson and extended by Dang & Wilkinson [7, 2] in `R`
  1. Determine dot size (radius)
  2. Find each dot's "neighbor"—what other dots would overlap *at that dot size*
  3. Start with the dot with the most neighbors (overlap) and create a "stack" (orthogonally displace)
  4. Continue in like manner until all dots are assigned to unique stacks
  5. Center each stack at *its* median (or mean)

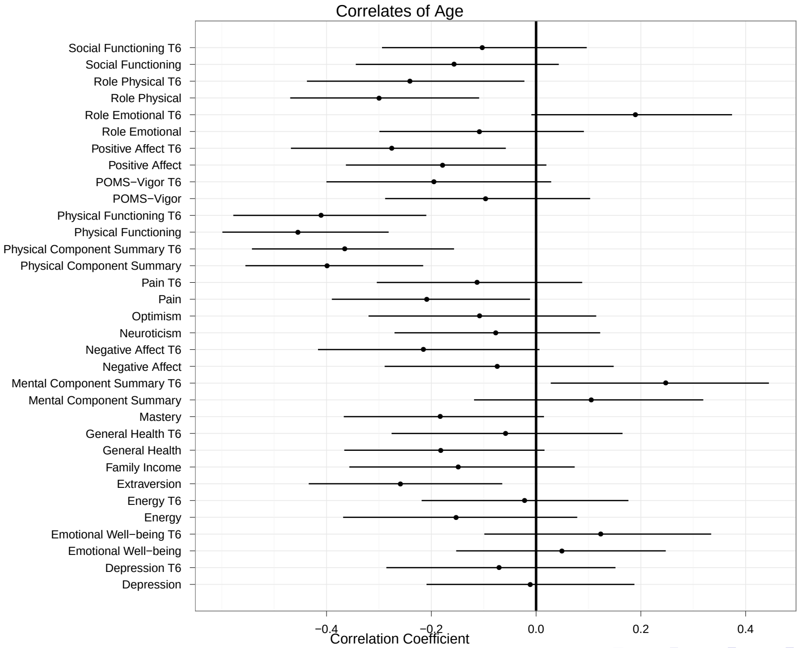
Implications

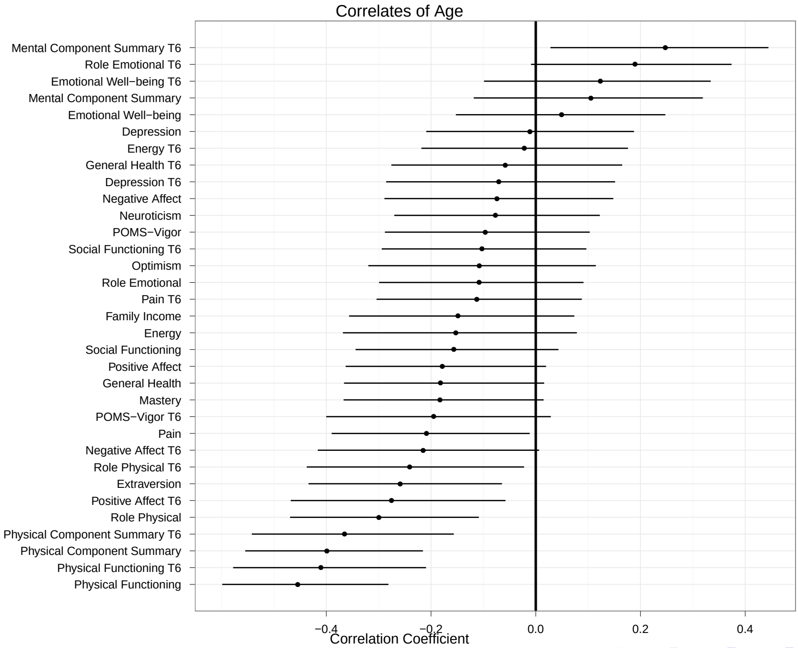
The algorithm and implementation suggest:

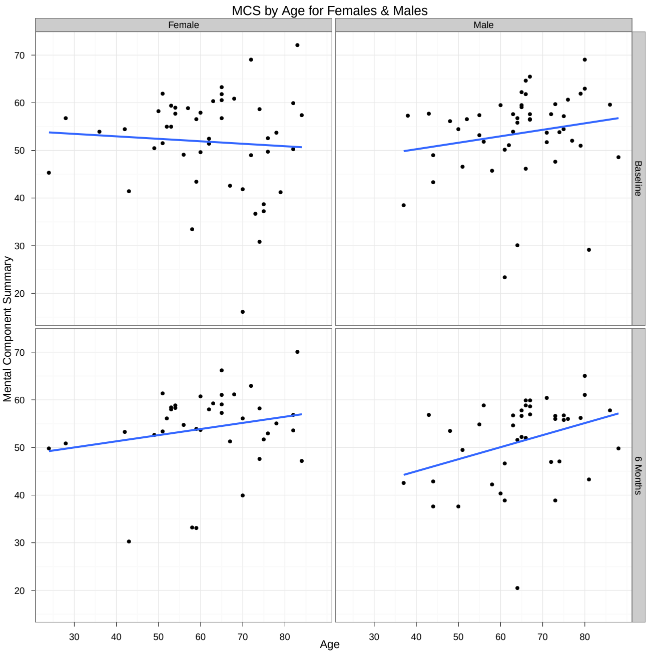
- Dot size determines the shape & degree of smoothing
- Using default dot size ($n^{-\frac{1}{2}}$), graph closely matches actual data (low bias) at the cost of high variance [7].
- Difficult with 2+ dimensions
- Additional information can easily be encoded via aesthetics such as colour or shape (e.g., squares, stars, triangles).

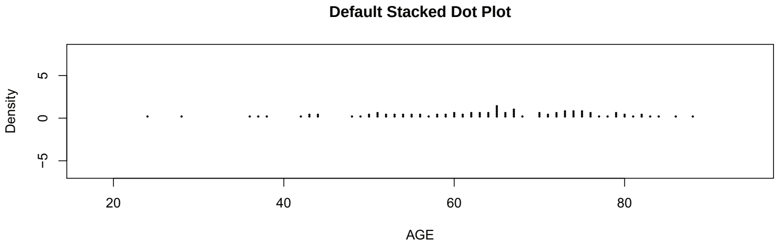
Stacked dot plots present an ideal way to display the actual data and their distribution, particularly with relatively small sample sizes ($n \leq 200$).

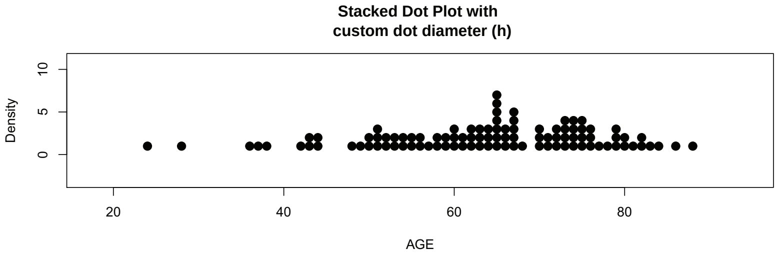
Table of Contents

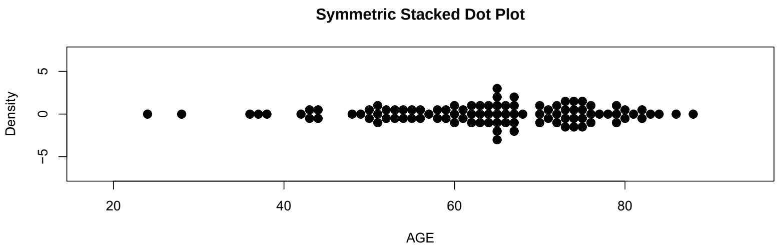

Correlates of Age
Social Functioning T6
Social Functioning
Role Physical T6
Role Physical
Role Emotional T6
Role Emotional
Positive Affect T6
Positive Affect
POMS−Vigor T6
POMS−Vigor
Physical Functioning T6
Physical Functioning
Physical Component Summary T6
Physical Component Summary
Pain T6
Pain
Optimism
Neuroticism
Negative Affect T6
Negative Affect
Mental Component Summary T6
Mental Component Summary
Mastery
General Health T6
General Health
Family Income
Extraversion
Energy T6
Energy
Emotional Well−being T6
Emotional Well−being
Depression T6
Depression
−0.4
−0.2
0.0
0.2
0.4
Correlation Coefficient


Correlates of Age
Mental Component Summary T6
Role Emotional T6
Emotional Well−being T6
Mental Component Summary
Emotional Well−being
Depression
Energy T6
General Health T6
Depression T6
Negative Affect
Neuroticism
POMS−Vigor
Social Functioning T6
Optimism
Role Emotional
Pain T6
Family Income
Energy
Social Functioning
Positive Affect
General Health
Mastery
POMS−Vigor T6
Pain
Negative Affect T6
Role Physical T6
Extraversion
Positive Affect T6
Role Physical
Physical Component Summary T6
Physical Component Summary
Physical Functioning T6
Physical Functioning
−0.4
−0.2
0.0
0.2
0.4
Correlation Coefficient

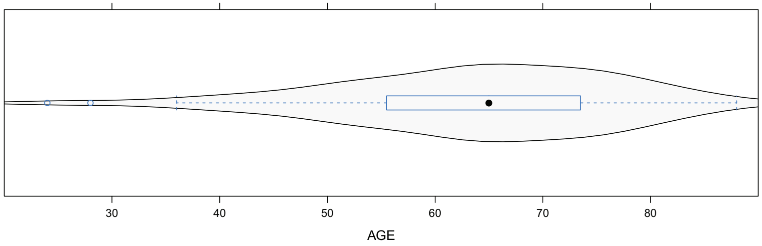

MCS by Age for Females & Males
Female
Male
Baseline
6 Months
Mental Component Summary
Age
70
60
50
40
30
20
30
40
50
60
70
80


Default Stacked Dot Plot
Density
5
0
−5
20
40
60
80
AGE


Stacked Dot Plot with
custom dot diameter (h)
Density
10
5
0
20
40
60
80
AGE



Symmetric Stacked Dot Plot
Density
5
0
−5
20
40
60
80
AGE



30
40
50
60
70
80
AGE

Syntax I

The syntax to create these plots is quite straightforward. The function doing the work is called "StackedDots", and it takes two main arguments:

1. the name of your dataset (e.g., "ocmel")
2. a "map" linking variables in your dataset to a particular dimension of the graph (e.g., color)

Here you can see the code for the first graph:

```
StackedDots(data = ocmel, map = link(x = AGE))
```

Syntax II

- Manually increasing the dot size using the h parameter

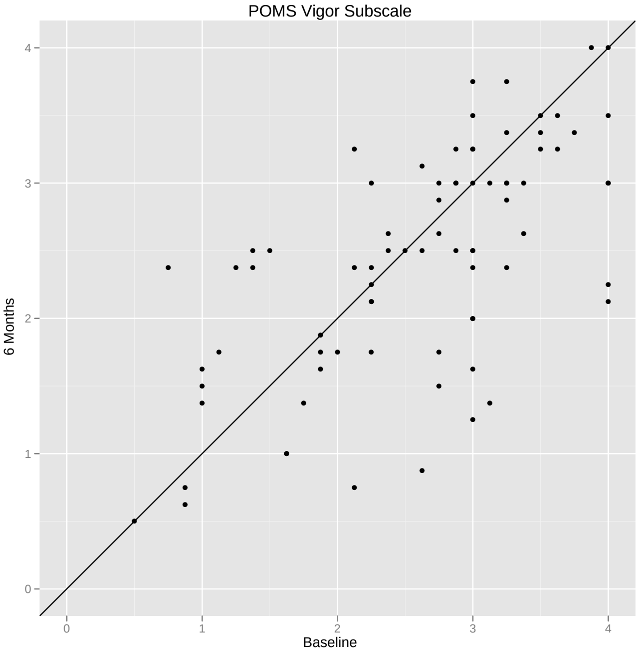
```
StackedDots(data = ocmel, map = link(x = AGE, h = 1))
```

- Adding the optional argument, `symm = TRUE` to make it symmetric

```
StackedDots(data = ocmel, map = link(x = AGE, h = 1), symm
= TRUE)
```

- In many cases, argument names are not needed—these two calls are equivalent

```
StackedDots(data = ocmel, map = link(x = AGE, h = 1))
StackedDots(ocmel, map = link(AGE, 1))
```


POMS Vigor Subscale
6 Months
Baseline
0
1
2
3
4
0
1
2
3
4

R Syntax

- Depression at baseline does not appear normally distributed

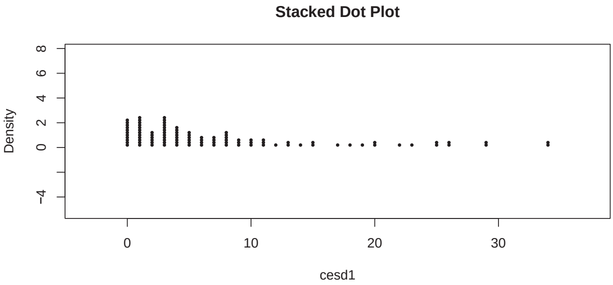
```
StackedDots(ocmel, map = link(cesd1))
```

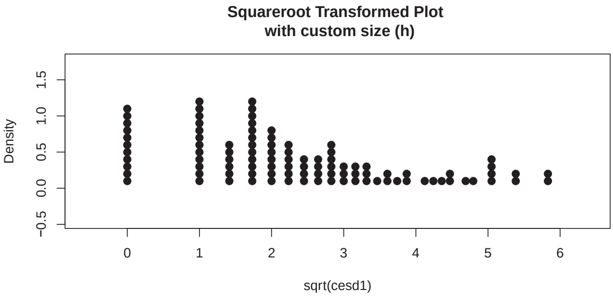
- `StackedDots` is designed to make using transformations easy. I demonstrate `sqrt`, but any function would work (e.g., logarithm).

```
StackedDots(ocmel, link(sqrt(cesd1), h = .1))
```

- The final graph colors the dots by gender, this is achieved by mapping the gender variable to color

```
StackedDots(ocmel, link(sqrt(cesd6), .1, color = GENDER))
```


Stacked Dot Plot
Density
−4
0
2
4
6
8
0
10
20
30
cesd1


Squareroot Transformed Plot
with custom size (h)
Density
−0.5
0.0
0.5
1.0
1.5
0
1
2
3
4
5
6
sqrt(cesd1)

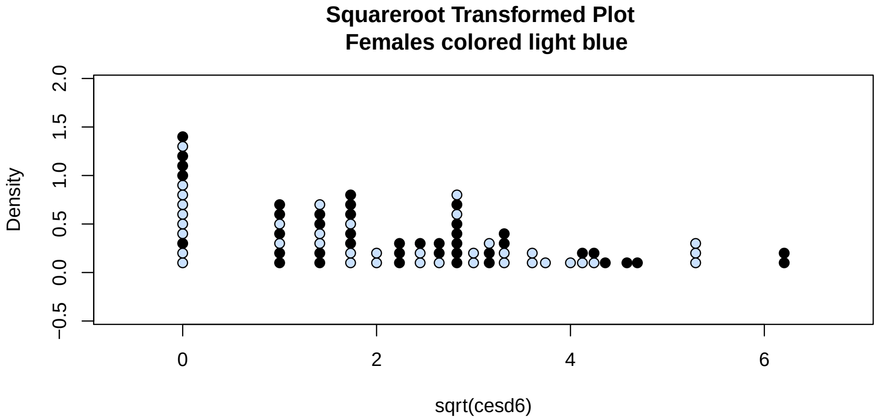

Squareroot Transformed Plot
Females colored light blue
Density
−0.5
0.0
0.5
1.0
1.5
2.0
0
2
4
6
sqrt(cesd6)

Table of Contents

Future Directions

- Allow additional variables to be encoded through different shapes (e.g., squares, triangles)
- Add argument to include kernel density estimate plots with dots
- Integrate better with existing graphical packages (e.g., add to margins to show distribution in scatterplot)
- Write a more efficient implementation of the algorithm
- Expand to bivariate distribution (3D)
- Extend to deal with lines as well as dots

Notes and Acknowledgements

- All graphs were created in the `R` [3] environment using a combination of `lattice` [4], `ggplot2` [5], and `Jmisc` [6].
- The posters were surveyed from Franz Tower, floors A - 8; Franz Middle, floors A - 3; and Franz Old, floors A - 3. I defined "graph" rather loosely as any figure or picture conveying quantitative information. Finally, as I was the only person collecting and checking the data, it is likely that there are some mistakes.

Bibliography I

W. S. Cleveland.

The Elements of Graphing Data.

Hobart Press, Summit, New Jersey, revised edition, 1994.

T. N. Dang, L. Wilkinson, and A. Anand.

Stacking graphic elements to avoid over-plotting.

IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, 16(6):1044–1052, NOVEMBER/DECEMBER 2010.

Bibliography II

R Development Core Team.
R: A Language and Environment for Statistical Computing.
R Foundation for Statistical Computing, Vienna, Austria, 2011.
ISBN 3-900051-07-0.

D. Sarkar.
Lattice: Multivariate Data Visualization with R.
Springer, New York, 2008.
ISBN 978-0-387-75968-5.

Bibliography III

H. Wickham.

ggplot2: elegant graphics for data analysis.

Springer New York, 2009.

J. Wiley.

Jmisc: A collection of miscellaneous functions, 2011.

R package version 0.2-0.

L. Wilkinson.

Dot plots.

The American Statistician, 53(3):276–281, August 1999.