Mathematical Biostatistics Bootcamp: Lecture 11, Plotting

Brian Caffo

Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health
Johns Hopkins University

September 13, 2012

Table of contents

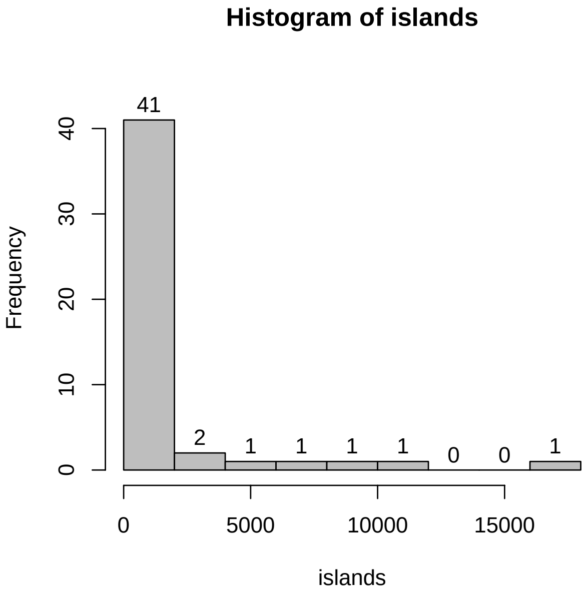
1. Table of contents
2. Histograms
3. Stem and leaf
4. Dotcharts
5. Boxplots
6. KDEs
7. QQ-plots
8. Mosaic plots

Histograms

- Histograms display a sample estimate of the density or mass function by plotting a bar graph of the frequency or proportion of times that a variable takes specific values, or a range of values for continuous data, within a sample

Example

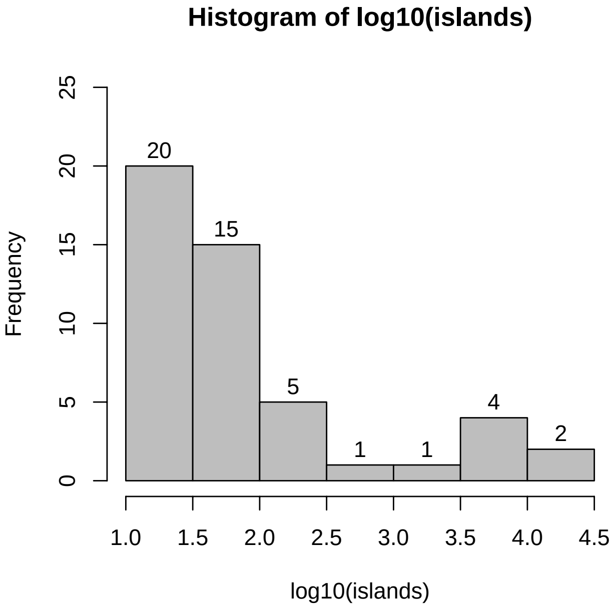
- The data set `islands` in the R package `datasets` contains the areas of all land masses in thousands of square miles
- Load the data set with the command `data(islands)`
- View the data by typing `islands`
- Create a histogram with the command `hist(islands)`
- Do `?hist` for options


Histogram of islands
Frequency
0
10
20
30
40
41
2
1
1
1
1
0
0
1
0
5000
10000
15000
islands

Pros and cons

- Histograms are useful and easy, apply to continuous, discrete and even unordered data
- They use a lot of ink and space to display very little information
- It's difficult to display several at the same time for comparisons

Also, for this data it's probably preferable to consider log base 10 (orders of magnitude), since the raw histogram simply says that most islands are small


Histogram of log10(islands)
Frequency
0
5
10
15
20
25
20
15
5
1
1
4
2
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
log10(islands)

Stem-and-leaf plots

- Stem-and-leaf plots are extremely useful for getting distribution information on the fly
- Read the text about creating them
- They display the complete data set and so waste very little ink
- Two data sets' stem and leaf plots can be shown back-to-back for comparisons
- Created by John Tukey, a leading figure in the development of the statistical sciences and signal processing

Example

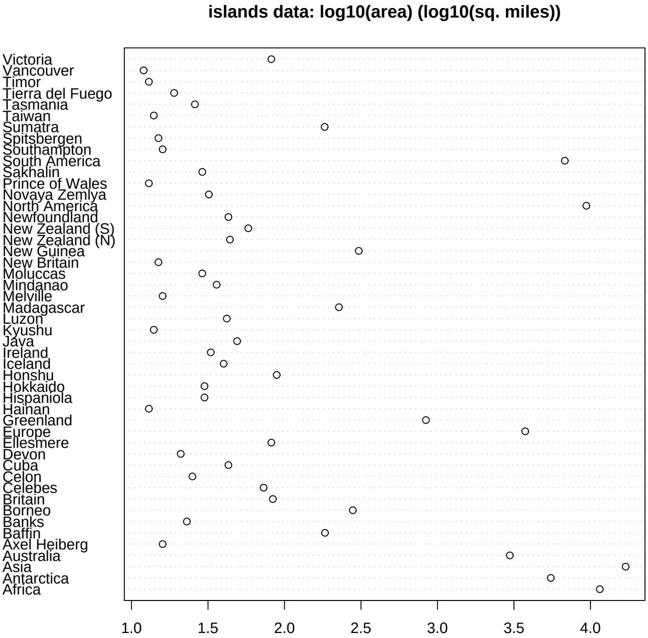
```
> stem(log10(islands))

  The decimal point is at the |

  1 | 1111112222233444
  1 | 5555556666667899999
  2 | 3344
  2 | 59
  3 |
  3 | 5678
  4 | 012
```

Dotcharts

- Dotcharts simply display a data set, one point per dot
- Ordering of the of the dots and labeling of the axes can the display additional information
- Dotcharts show a complete data set and so have high data density
- May be impossible to construct/difficult to interpret for data sets with lots of points


islands data: log10(area) (log10(sq. miles))
Victoria
Vancouver
Timor
Tierra del Fuego
Tasmania
Taiwan
Sumatra
Spitsbergen
Southampton
South America
Sakhalin
Prince of Wales
Novaya Zemlya
North America
Newfoundland
New Zealand (S)
New Zealand (N)
New Guinea
New Britain
Moluccas
Mindanao
Melville
Madagascar
Luzon
Kyushu
Java
Ireland
Iceland
Honshu
Hokkaido
Hispaniola
Hainan
Greenland
Europe
Ellesmere
Devon
Cuba
Celon
Celebes
Britain
Borneo
Banks
Baffin
Axel Heiberg
Australia
Asia
Antarctica
Africa
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Discussion

- Maybe ordering alphabetically isn't the best thing for this data set
- Perhaps grouped by continent, then nations by geography (grouping Pacific islands together)?

Dotplots comparing grouped data

- For data sets in groups, you often want to display density information by group
- If the size of the data permits, it displaying the whole data is preferable
- Add horizontal lines to depict means, medians
- Add vertical lines to depict variation, show confidence intervals interquartile ranges
- Jitter the points to avoid overplotting (`jitter`)

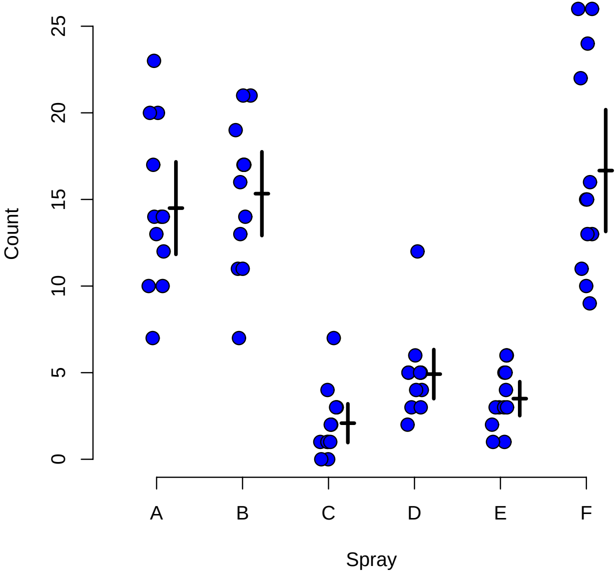
Example

- The InsectSprays dataset contains counts of insect deaths by insecticide type (A, B, C, D, E, F)
- You can obtain the data set with the command

```
data(InsectSprays)
```

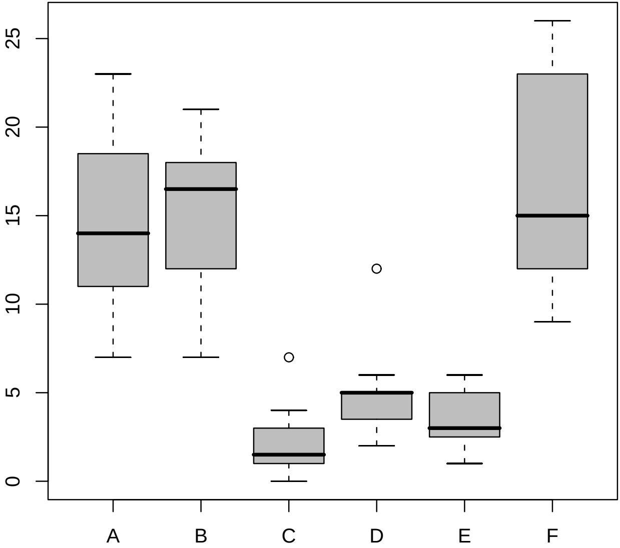
The gist of the code is below

```
attach(InsectSprays)
plot(c(.5, 6.5), range(count))
sprayTypes <- unique(spray)
for (i in 1 : length(sprayTypes)){
  y <- count[spray == sprayTypes[i]]
  n <- sum(spray == sprayTypes[i])
  points(jitter(rep(i, n), amount = .1), y)
  lines(i + c(.12, .28), rep(mean(y), 2), lwd = 3)
  lines(rep(i + .2, 2),
        mean(y) + c(-1.96, 1.96) * sd(y) / sqrt(n)
       )
}
```


Count
0
5
10
15
20
25
A
B
C
D
E
F
Spray

Boxplots

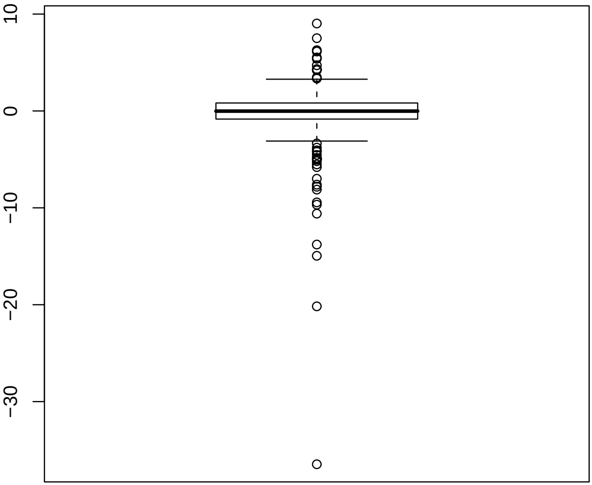
- Boxplots are useful for the same sort of display as the dot chart, but in instances where displaying the whole data set is not possible
- Centerline of the boxes represents the median while the box edges correspond to the quartiles
- Whiskers extend out to a constant times the IQR or the max value
- Sometimes potential outliers are denoted by points beyond the whiskers
- Also invented by Tukey
- Skewness indicated by centerline being near one of the box edges


0
5
10
15
20
25
A
B
C
D
E
F

Boxplots discussion

- Don't use boxplots for small numbers of observations, just plot the data!
- Try logging if some of the boxes are too squished relative to other ones; you can convert the axis to unlogged units (though they will not be equally spaced anymore)
- For data with lots and lots of observations omit the outliers plotting if you get so many of them that you cant see the points
- Example of a bad box plot

```
boxplot(rt(500, 2))
```


10
0
−10
−20
−30

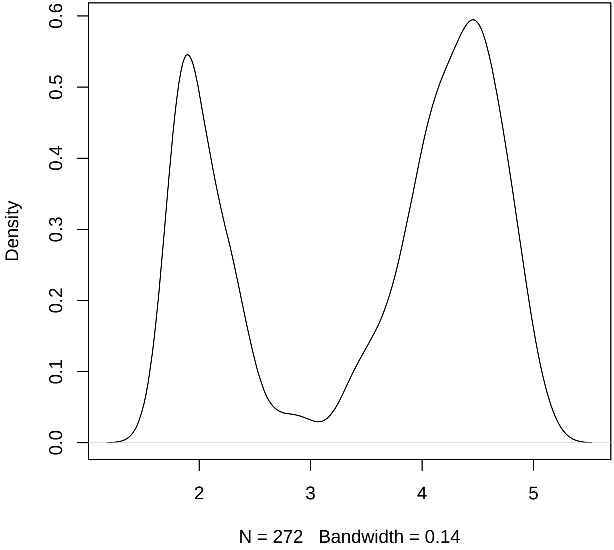
Kernel density estimates

- Kernel density estimates are essentially more modern versions of histograms providing density estimates for continuous data
- Observations are weighted according to a “kernel”, in most cases a Gaussian density
- “Bandwidth” of the kernel effectively plays the role of the bin size for the histogram
  a. Too low of a bandwidth yields a too variable (jagged) measure of the density
  b. Too high of a bandwidth oversmooths
- The R function `density` can be used to create KDEs

Example

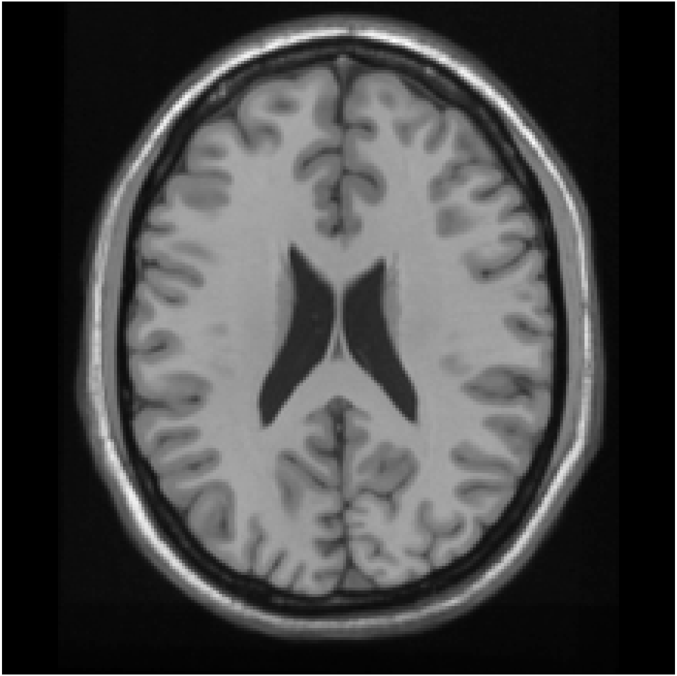
Data is the waiting and eruption times in minutes between eruptions of the Old Faithful Geyser in Yellowstone National park

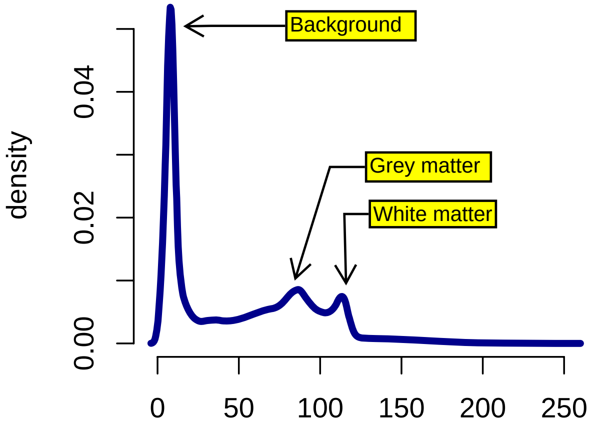
```
data(faithful)
d <- density(faithful$eruptions, bw = "sj")
plot(d)
```


0.6
0.5
0.4
0.3
0.2
0.1
0.0
Density
2
3
4
5
N = 272   Bandwidth = 0.14

Imaging example

- Consider the following image slice (created in R) from a high resolution MRI of a brain
- This is a single (axial) slice of a three-dimensional image
- Consider discarding the location information and plotting a KDE of the intensities

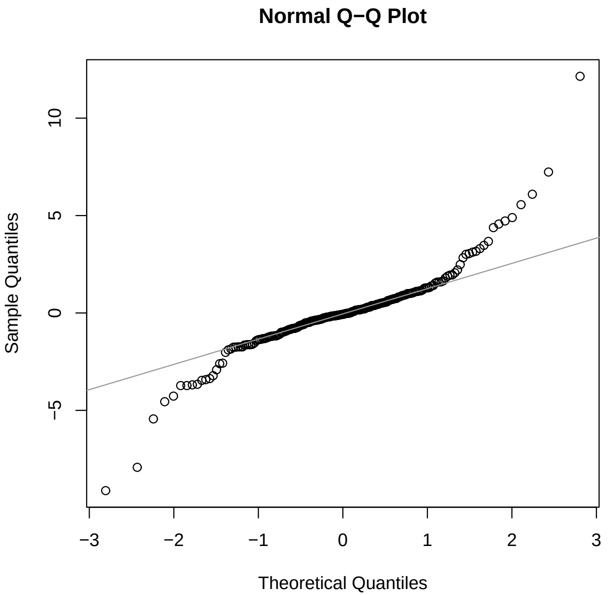


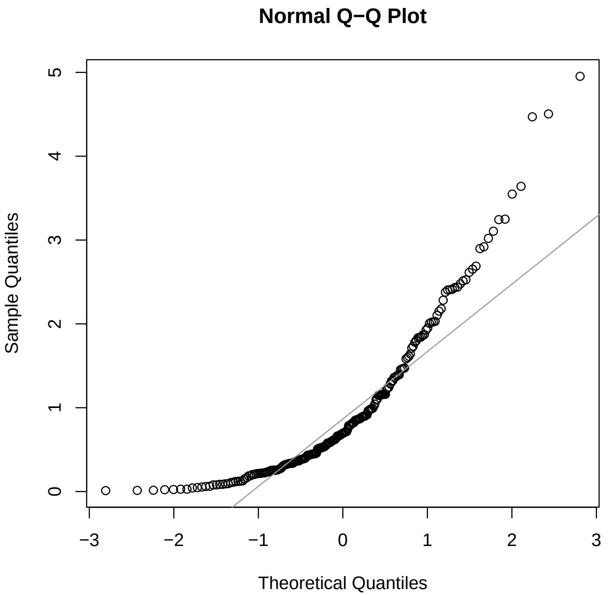

Background
Grey matter
White matter
density
0.00
0.02
0.04
0
50
100
150
200
250

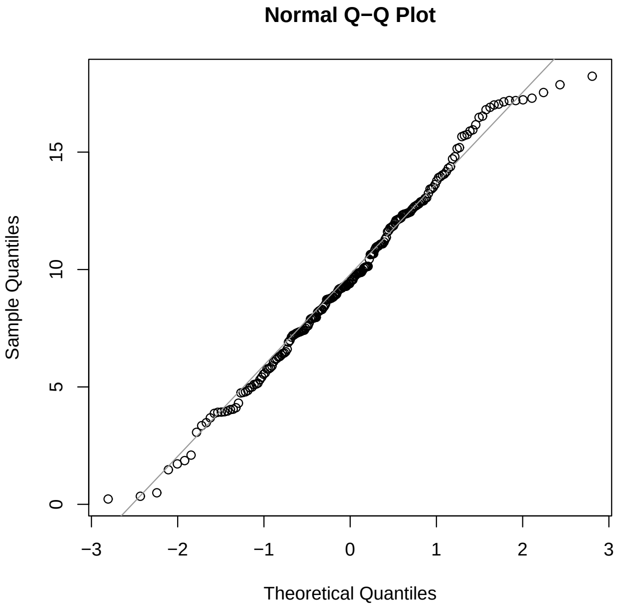
QQ-plots

- QQ-plots (for quantile-quantile) are extremely useful for comparing data to a theoretical distribution
- Plot the empirical quantiles against theoretical quantiles
- Most useful for diagnosing normality

- Let x_p be the p^{th} quantile from a $N(\mu, \sigma^2)$
- Then $P(X \leq x_p) = p$
- Clearly $P(Z \leq \frac{x_p - \mu}{\sigma}) = p$
- Therefore $x_p = \mu + z_p \sigma$ (this should not be news)
- Result, quantiles from a $N(\mu, \sigma^2)$ population should be linearly related to standard normal quantiles
- A normal qq-plot plot the empirical quantiles against the theoretical standard normal quantiles
- In R `qqnorm` for a normal QQ-plot and `qqplot` for a qqplot against an arbitrary distribution


Normal Q−Q Plot
Sample Quantiles
−5
0
5
10
−3
−2
−1
0
1
2
3
Theoretical Quantiles

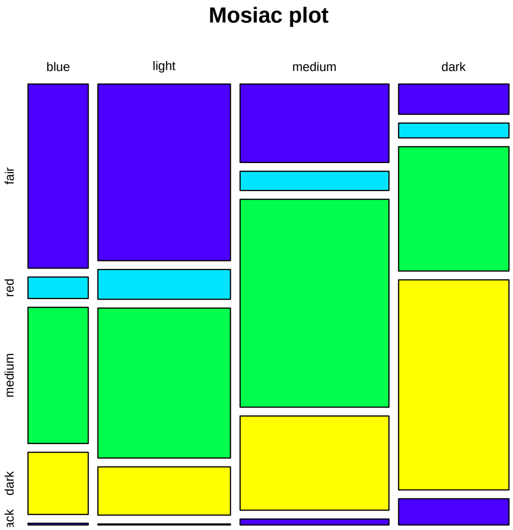

Normal Q−Q Plot
Sample Quantiles
0
1
2
3
4
5
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Normal Q−Q Plot
Sample Quantiles
0
5
10
15
Theoretical Quantiles
−3
−2
−1
0
1
2
3

Mosaic plots

- Mosaic plots are useful for displaying contingency table data
- Consider Fisher's data regarding hair and eye color data for people from Caithness

```
library(MASS)
data(caith)
caith
mosaicplot(caith, color = topo.colors(4),
           main = "Mosiac plot")
       fair red medium dark black
blue    326  38    241  110     3
light   688 116    584  188     4
medium  343  84    909  412    26
dark     98  48    403  681    85
```


Mosiac plot
blue
light
medium
dark
fair
red
medium
dark
ack