STAT 545A
Class meetings #5 and #6
Monday, September 23, 2013
Wednesday, September 25, 2013

Dr. Jennifer (Jenny) Bryan

Department of Statistics and Michael Smith Laboratories

You should be able to name files following the convention, create your Rpubs and Gists following the convention, and give us Markdown-ready links by now.

- most common mistake: leaving a space between the square bracket part and the parenthesis part

```
here is a [link that works](www.google.com)
here is a [broken link] (www.google.com)
```

NO!

Consider observations of one quantitative variable X ... possibly in the presence of one or two categorical variables Y and Z, that take on a small number of values

X might be ... life expectancy in Gapminder

Y, Z might be ... country or continent or year

"Using fancy tools like neural nets, boosting, and support vector machines without understanding basic statistics is like doing brain surgery before knowing how to use a band-aid."

-- Larry Wasserman in preface of "All of Statistics"

X = life expectancy

Y, Z = country,continent and/or year

What would you most like to know about the observed distribution of the X's (ignore Y, Z)?

Now focus on the possible relationship between X and Y, Z. What would you most like to know?

<make a list>

What would you most like to know about the observed distribution of the X's (ignore Y, Z)?

stuff people say
a "typical" value the "middle" of the data the value that's "most likely"

formal terms
location central location central tendency

relevant parameters
mean or average or expectation median mode

be careful with your words and thoughts

never forget the distinction between the underlying truth ("the true mean") and the current observation ("the observed average")

What would you most like to know about the observed distribution of the X's (ignore Y, Z)?

concepts
spread variability dispersion

relevant parameters
standard deviation, variance median absolute deviation interquartile range range

we can compute these from the observed data

average

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

variance

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y}_n)^2$$

we do so to make inference about these

mean or expectation

$$E(Y) = \sum_y y p_Y(y) \text{ for discrete rv Y}$$

$$E(Y) = \int y f_Y(y)\,dy \text{ for continuous rv Y}$$

variance

$$V(Y) = E(Y-\mu)^2$$

we can compute these from the observed data

average

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

variance

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y}_n)^2$$

we do so to make inference about these

mean or expectation

$$E(Y) = \sum_y y p_Y(y) \text{ for discrete rv Y}$$

$$E(Y) = \int y f_Y(y)\, dy \text{ for continuous rv Y}$$

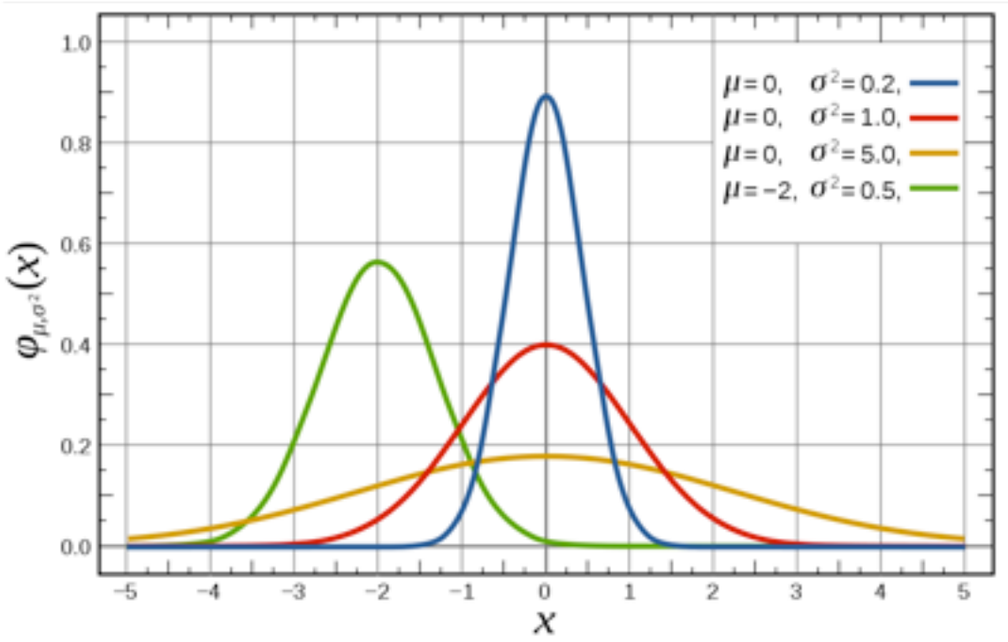
variance

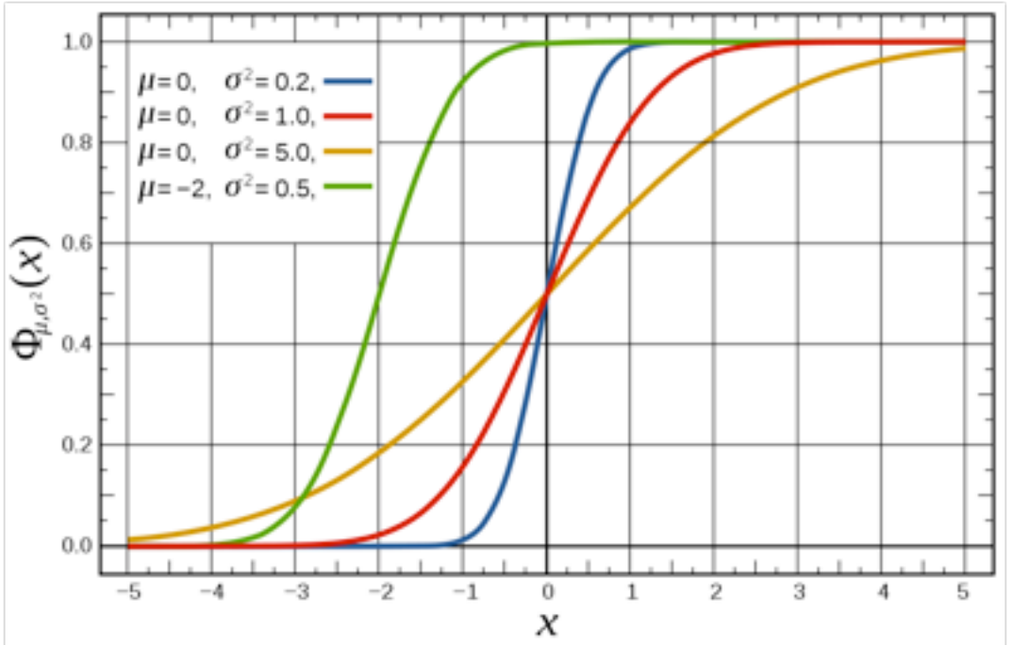
$$V(Y) = E(Y-\mu)^2$$

density or prob. mass function

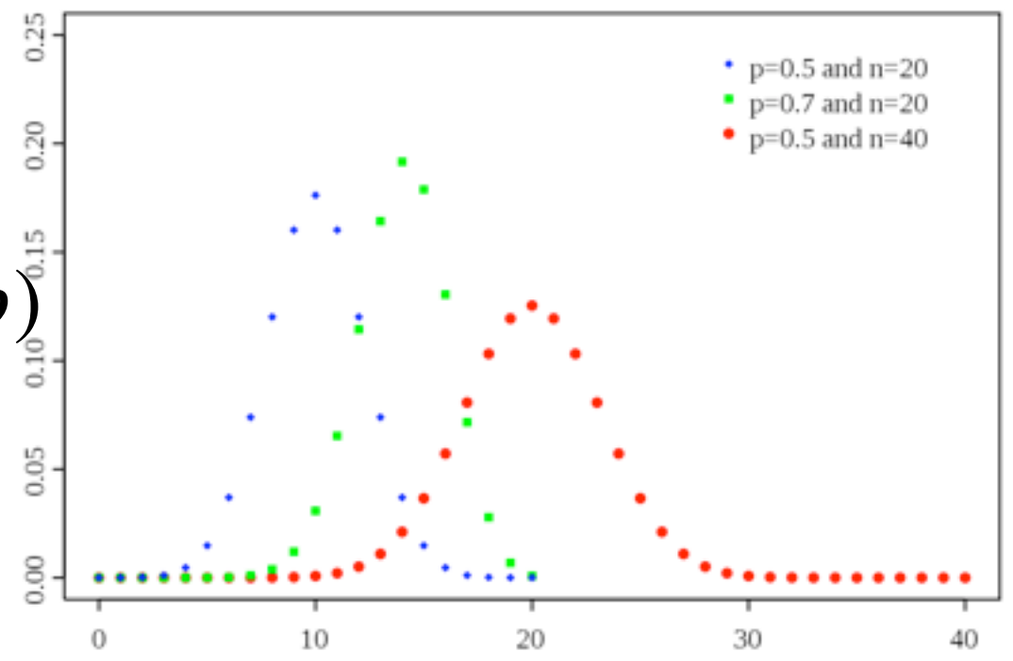
CDF

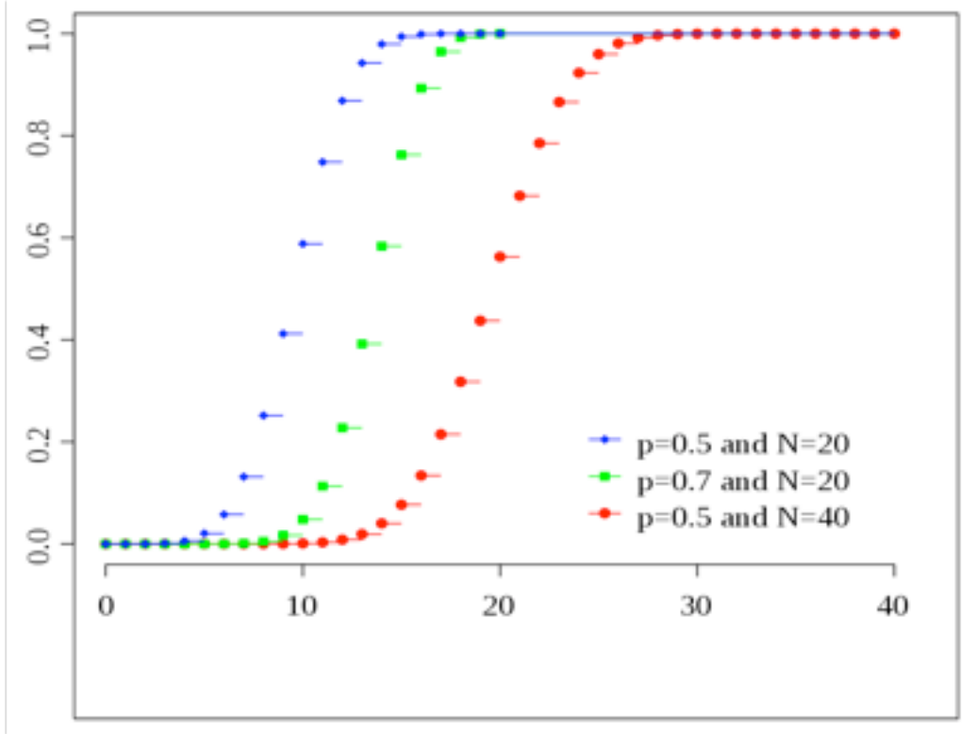
$N(\mu,\sigma^2)$





$Binom(n,p)$





sources of images on previous page

http://en.wikipedia.org/wiki/File:Normal_Distribution_PDF.svg

http://en.wikipedia.org/wiki/File:Normal_Distribution_CDF.svg

http://en.wikipedia.org/wiki/File:Binomial_distribution_pmf.svg

http://en.wikipedia.org/wiki/File:Binomial_distribution_cdf.svg

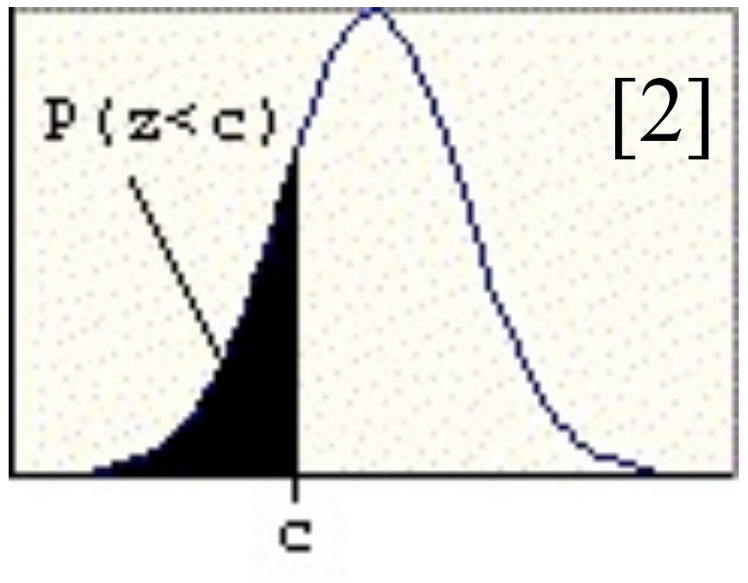
"cumulative distribution function (CDF)"

$$F_Y(a) = P(Y \leq a) = \int_{-\infty}^{a} f_Y(y)dy \text{ (for a continuous Y)}$$

$$F_Y(a) = P(Y \leq a) = \sum_{y_i \leq a} p_Y(y_i) \text{ (for a discrete Y)}$$

yes, we really do distinguish the density function (continuous rv) from the CDF with the deceptively subtle lowercase "*f*" vs. uppercase "*F*"

how to get a probability from a density



[2]

$$[1]\ P(a < Y < b) = \int_a^b f_Y(y)\,dy$$

$$[2]\ P(Y \leq a) = \int_{-\infty}^{a} f_Y(y)\,dy$$

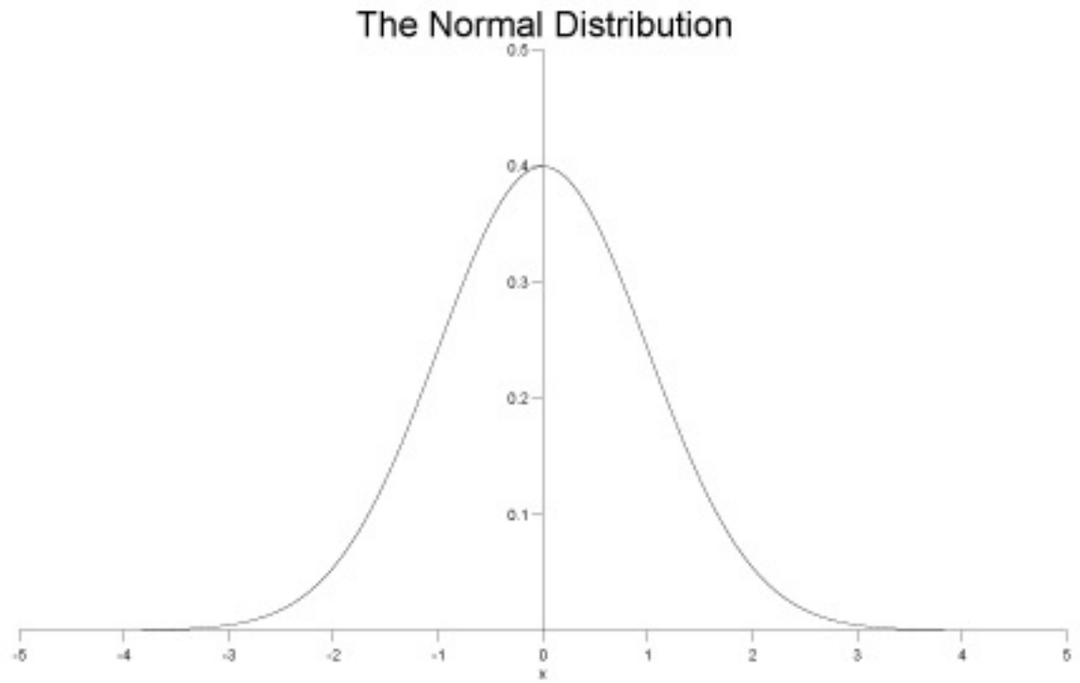
$$[3]\ P(Y \geq a) = \int_a^{\infty} f_Y(y)\,dy$$

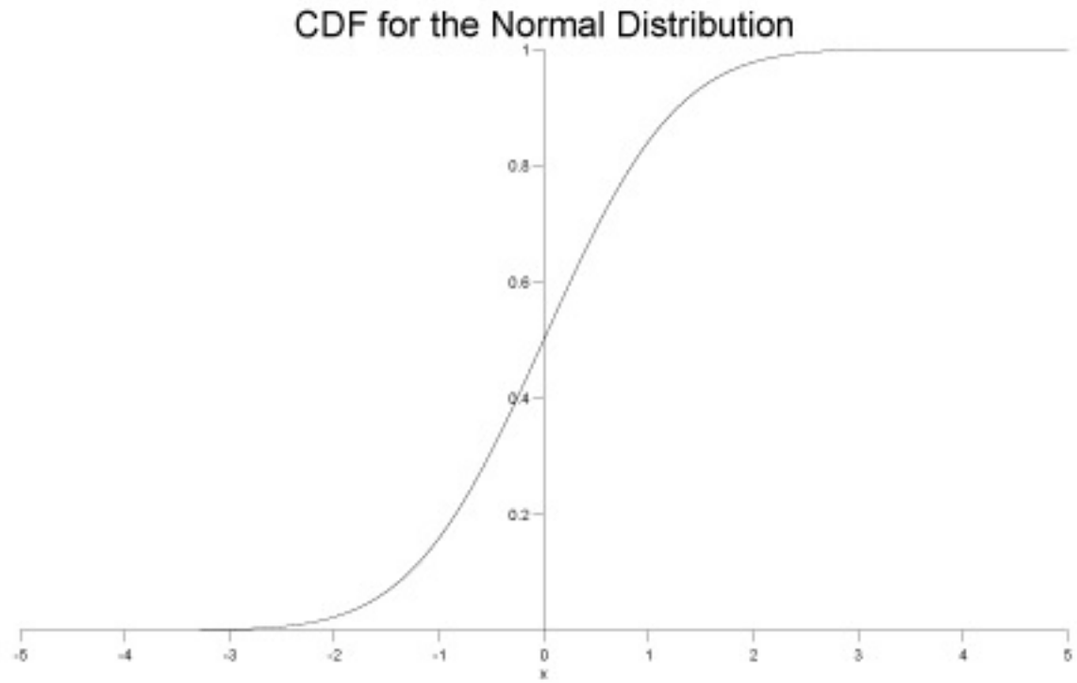
$$[4]\ P(|Y| \geq a) = \int_{-\infty}^{-a} f_Y(y)\,dy + \int_a^{\infty} f_Y(y)\,dy$$

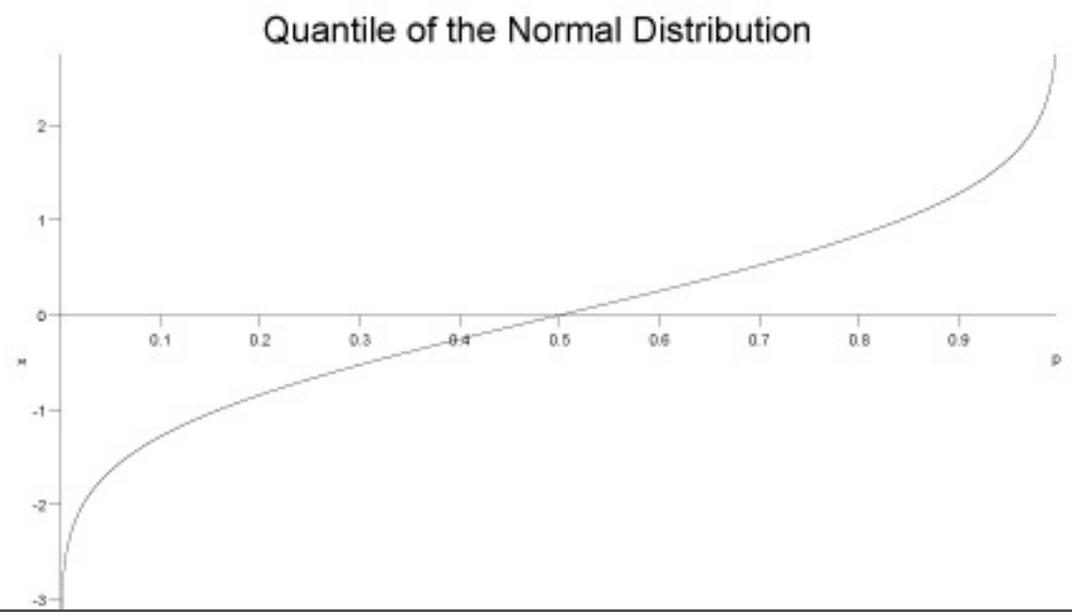
"cumulative distribution function"

inverse CDF, quantile function

$$F_Y^{-1}(q) = \text{smallest } y \text{ such that } F_Y(y) > q$$







inverse CDF, quantile function

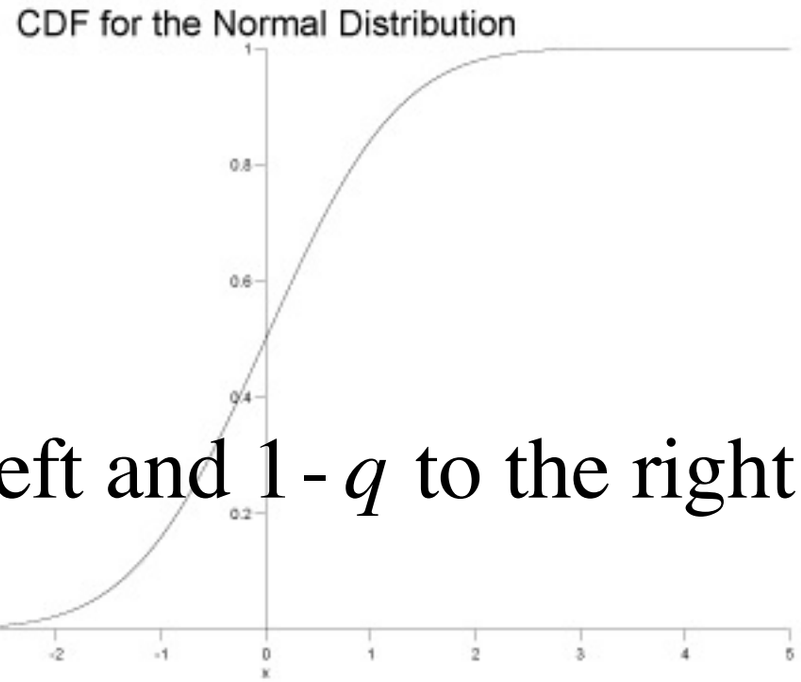
$$F_Y^{-1}(q) = \text{smallest } y \text{ such that } F_Y(y) > q$$

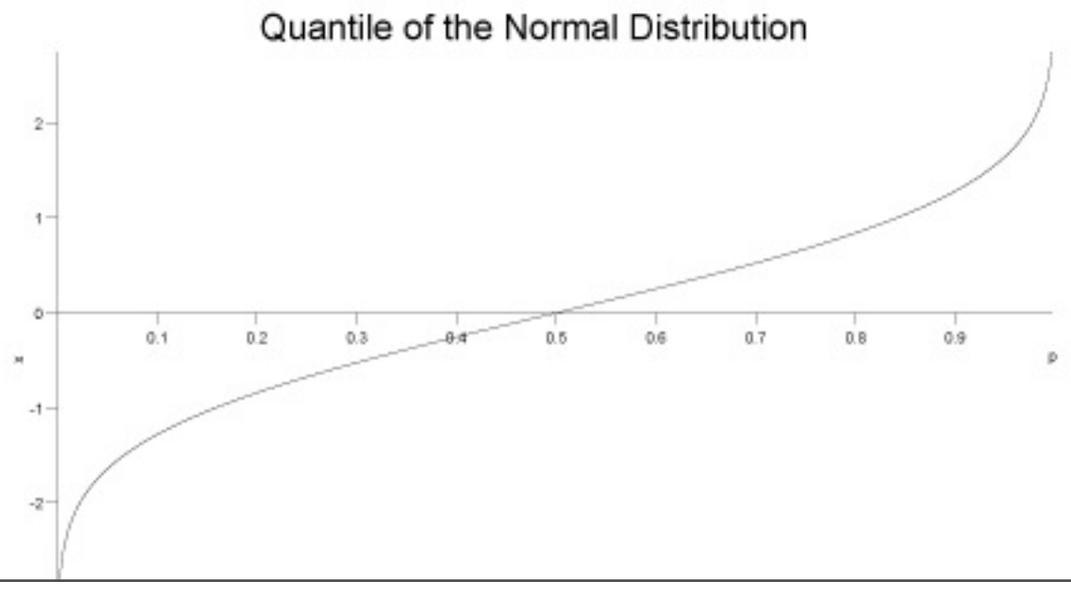
$F_Y^{-1}(0.5)$ = "the median"

$F_Y^{-1}(0.25)$ = "the first quartile"

$F_Y^{-1}(q)$ = value that traps probability q to the left and $1-q$ to the right

IQR = $F^{-1}(0.75) - F^{-1}(0.25)$

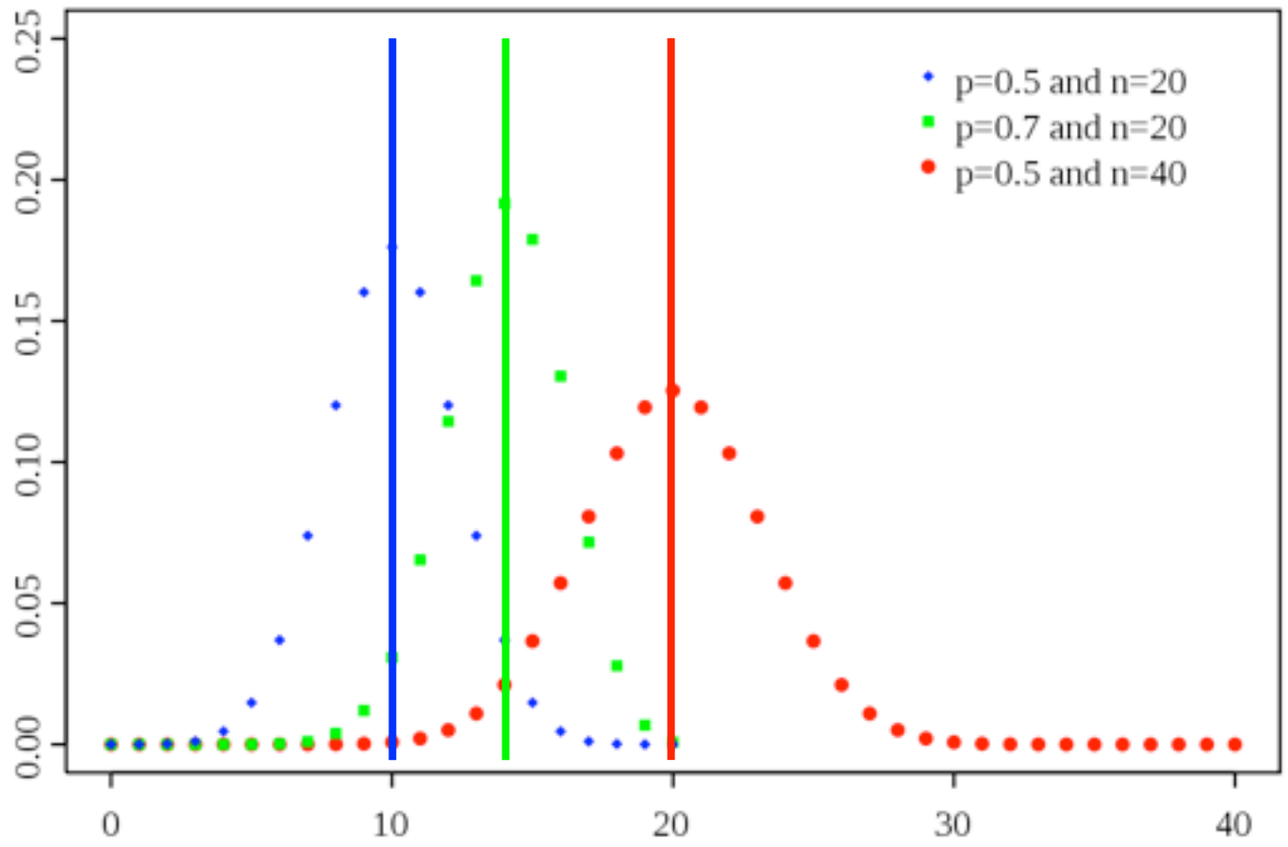




expectation, expected value, the mean

$$E(Y) = \sum_y y p_Y(y) \text{ for discrete rv Y}$$

$$E(Y) = \int y f_Y(y) dy \text{ for continuous rv Y}$$



binomial example:

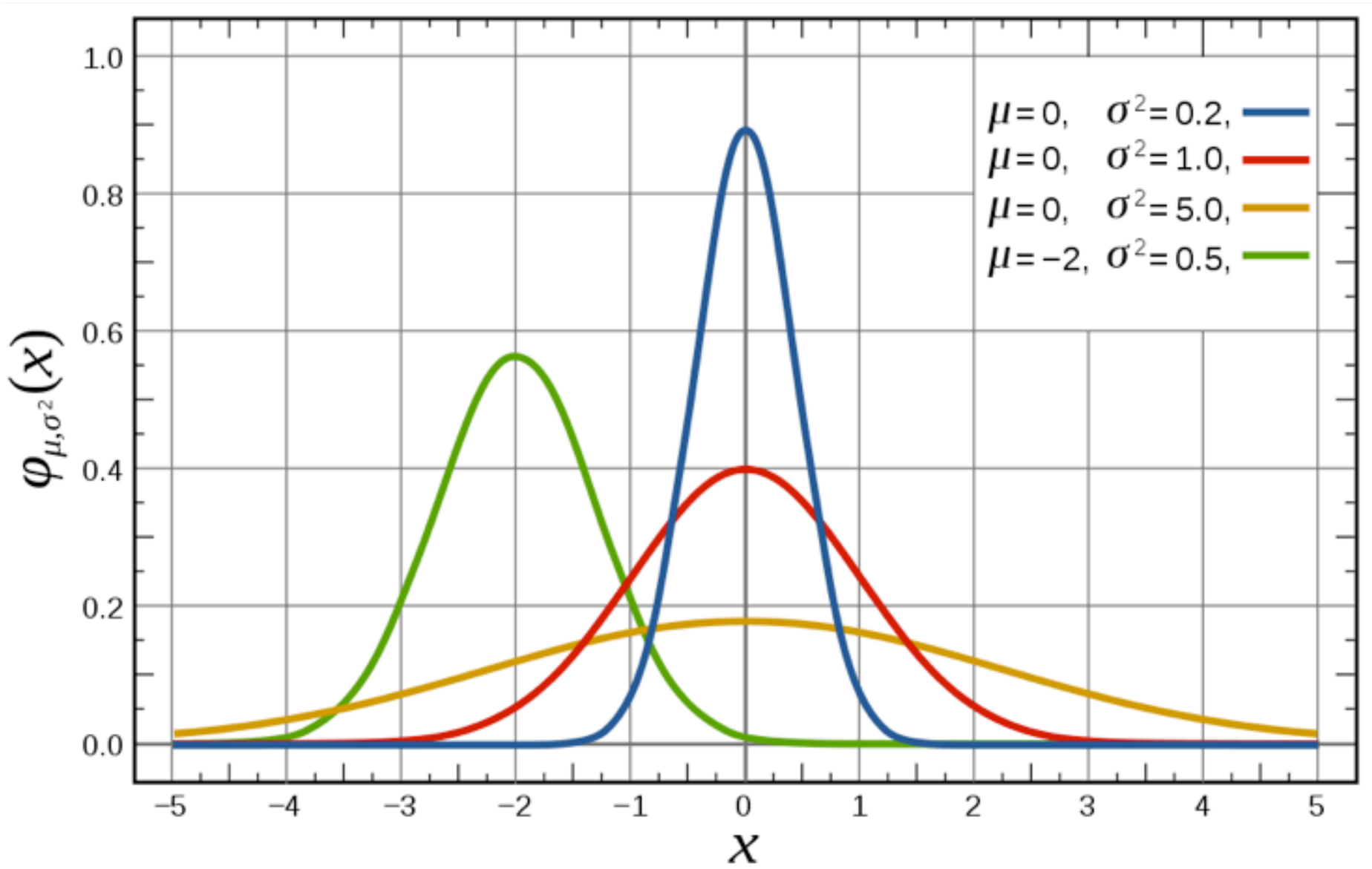
$$Y \sim Binom(n, p)$$

$$E(Y) = np$$

the mean is a measure of "location"
often is one of the "obvious" parameters (e.g. normal)
or is easily computed from them (e.g. binomial)

variance

standard deviation = √variance



normal as example; bigger σ^2 ↔ bigger "spread"

Key concepts -- less ‘tidy’

- Unimodal? If not, how many modes? Where?
- Symmetric? If not, what’s the shape? Which tail is long?
- Extremes ... what’s the max and min?
- If considering Y, is the distribution of X meaningfully different ... in location, spread, shape, etc. ... for different values of Y?

Summaries computed from observed data are *empirical versions* of those "key" concepts

I.e. the average of a sample is an estimate -- and merely an estimate -- of the true mean

Clear statistical thinkers make a *big distinction* between these concepts, though we often speak casually about it

In this exploratory data analysis class we will be fairly relaxed but don't ever forget these distinctions are real

Numerical summaries, esp. location

```
> summary(gDat$lifeExp)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.60   48.20   60.71   59.47   70.85   82.60

> fivenum(gDat$lifeExp)
[1] 23.5990 48.1850 60.7125 70.8460 82.6030

> mean(gDat$lifeExp)
[1] 59.47444

> median(gDat$lifeExp)
[1] 60.7125
```

Numerical summaries, esp. spread

```
> var(gDat$lifeExp)
[1] 166.8517

> sd(gDat$lifeExp)
[1] 12.91711

> mad(gDat$lifeExp)
[1] 16.10104

> IQR(gDat$lifeExp)
[1] 22.6475
```

Numerical summaries, esp. extremes

```
> min(gDat$lifeExp)
[1] 23.599

> max(gDat$lifeExp)
[1] 82.603

> quantile(gDat$lifeExp, probs = c(0.05, 0.95))
     5%     95%
38.4924 77.4370

> range(gDat$lifeExp)
[1] 23.599 82.603

> which.min(gDat$lifeExp)
[1] 1293

> gDat[which.min(gDat$lifeExp), ]
     country year     pop continent lifeExp gdpPercap
1293  Rwanda 1992 7290203    Africa  23.599  737.0686

> which.max(gDat$lifeExp)
[1] 804

> gDat[which.max(gDat$lifeExp), ]
    country year       pop continent lifeExp gdpPercap
804   Japan 2007 127467972      Asia  82.603  31656.07
```

go over to R and try those functions out

especially try them out in a data aggregation setting, probably using functions from plyr

http://www.stat.ubc.ca/~jenny/STAT545A/block04_dataAggregation.html

but who wants to look at tables of numbers all day?

begin visual exploration ... first with lattice, later with ggplot2

we did the stripplot() part of this tutorial:
http://www.stat.ubc.ca/~jenny/STAT545A/block07_univariatePlotsLattice.html

STAT 545A
Class meeting #6
Wednesday, September 25, 2013

Dr. Jennifer (Jenny) Bryan

Department of Statistics and Michael Smith Laboratories

picking up where we left off ....

let's talk about data agg hw

then come back here for formulas

then switch between here and R to visualizations of a quantitative variable besides stripplot

Digression: R's formula syntax

http://cran.r-project.org/doc/manuals/R-intro.html#Formulae-for-statistical-models

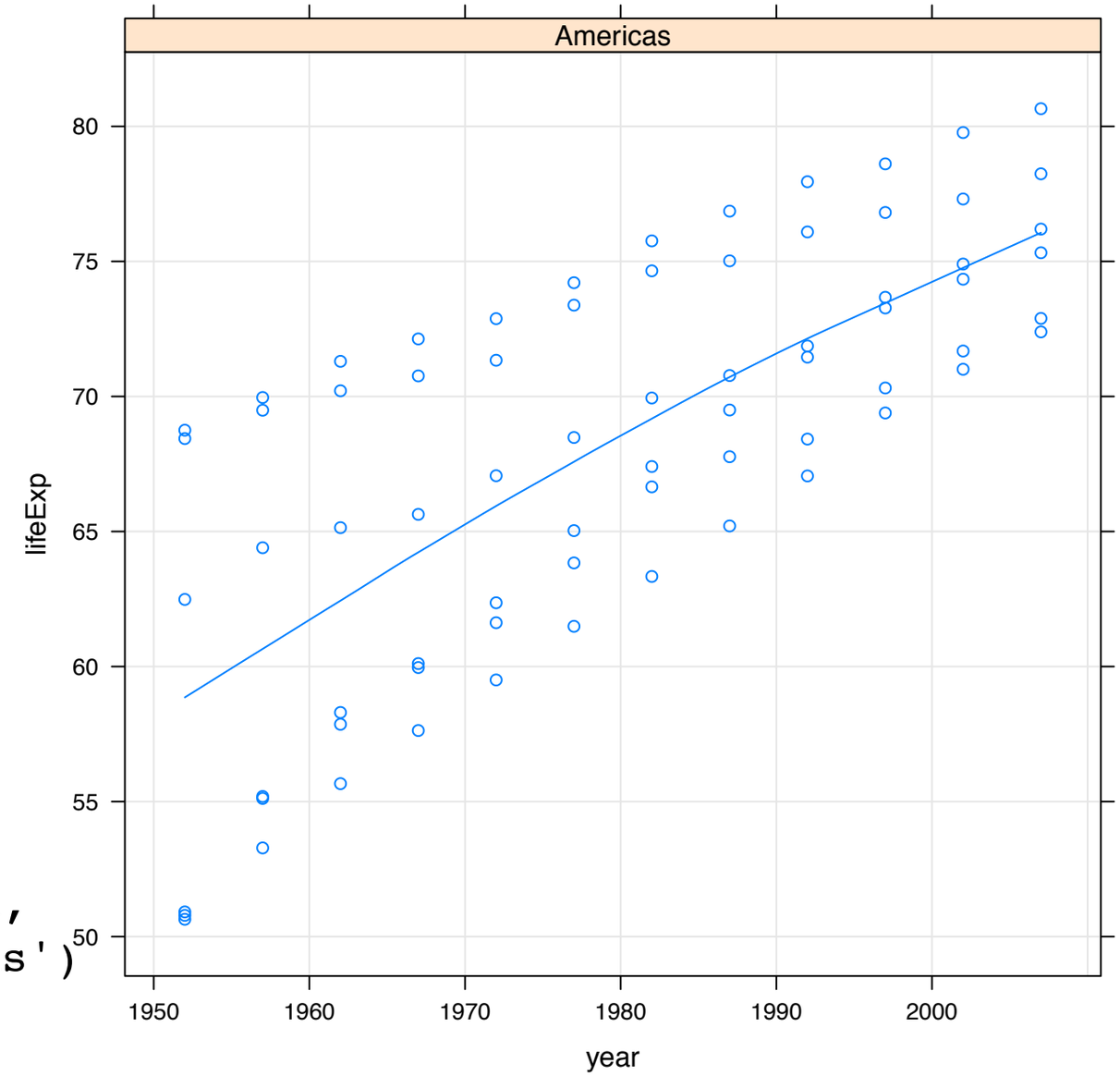
```
y ~ x
```

"y twiddle x"

In modelling functions, says y is response or dependent variable and x is the predictor or covariate or independent variable. More generally, the right-hand side can be much more complicated.

simple linear regression example

x and y are quantitative



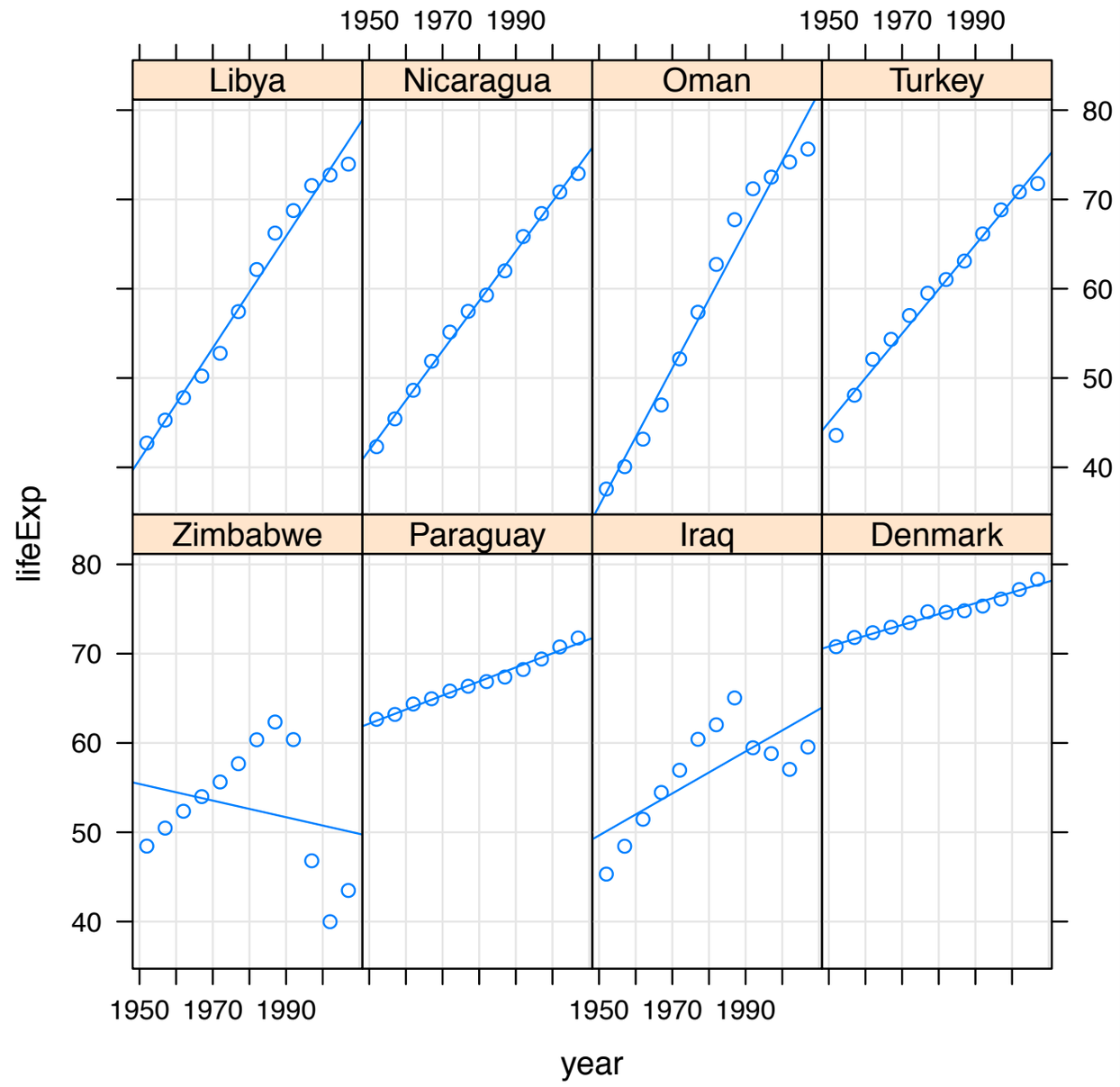

```
> jFit <- lm(lifeExp ~ I(year - 1950), gDat,
+            subset = continent == 'Americas')
> summary(jFit)

<snip, snip>

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    59.03624    1.20834  48.857  < 2e-16 ***
I(year - 1950)  0.30944    0.03535   8.753 7.52e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.178 on 70 degrees of freedom
Multiple R-squared: 0.5225, Adjusted R-squared: 0.5157
F-statistic: 76.61 on 1 and 70 DF,  p-value: 7.524e-13
```

y ~ x

"y twiddle x"

In many plotting functions, esp. lattice, this says to plot y against x.

```
xyplot(lifeExp ~ year | country, zDat,
       layout = c(4,2), type = c('p','g','r'))
```



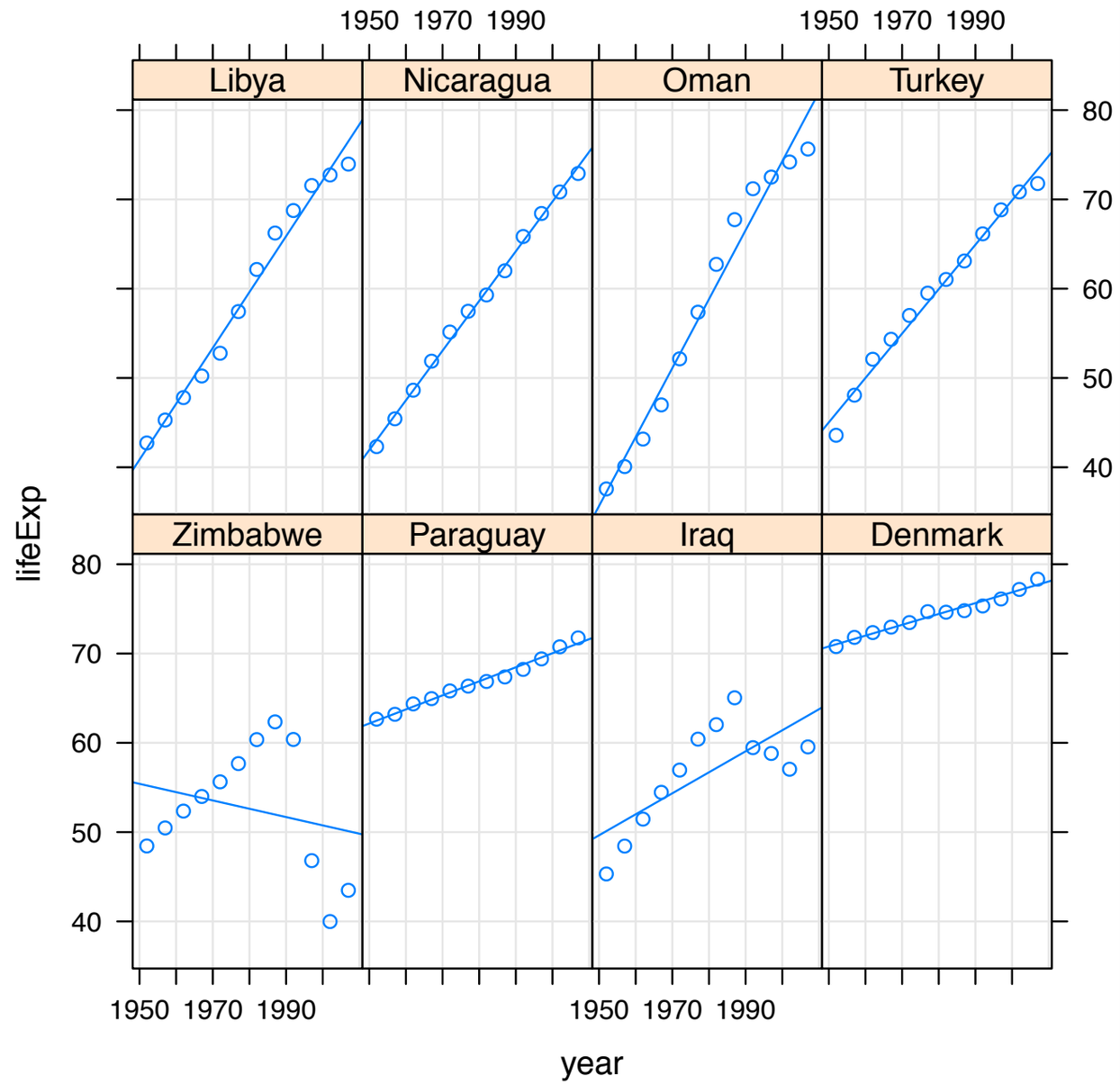
scatterplot example

x and y are quantitative

y ~ x | z

In many lattice plotting functions, this says to plot y against x for every level of z (assumed to be categorical). Evokes conditional probability, "given z", etc.

```
xyplot(lifeExp ~ year | country, zDat,
       layout = c(4,2), type = c('p','g','r'))
```



scatterplot example

x and y are quantitative

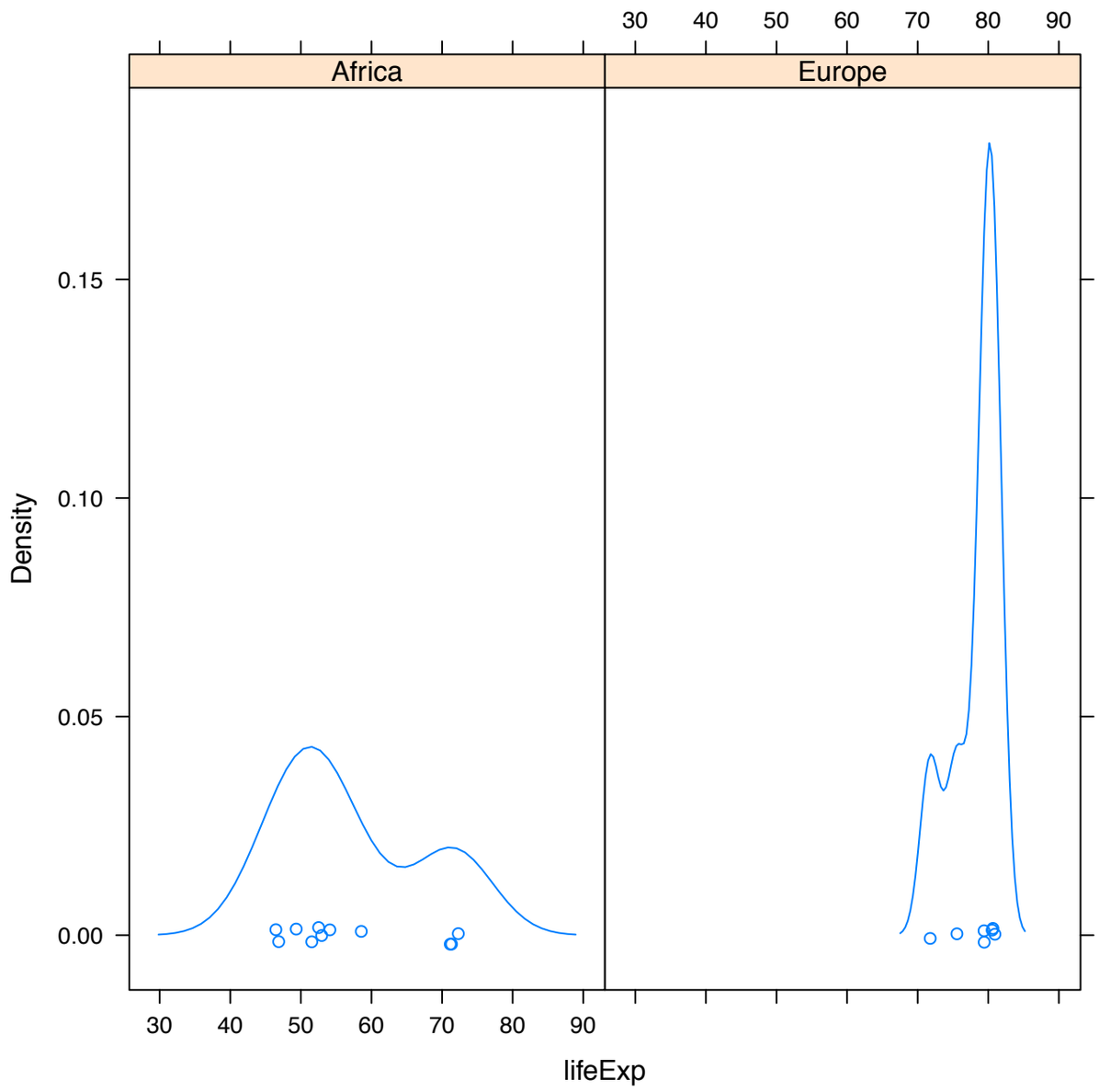
z is categorical

Inference example: is there a difference in distribution of Y given X = x?

```
y ~ x
```

two-groups testing example

y is quantitative and x is the binary variable that specifies the two groups




```
> t.test(lifeExp ~ continent, tinyDat)

    Welch Two Sample t-test

data:  lifeExp by continent
t = -6.5267, df = 13.291, p-value = 1.727e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -28.35922 -14.27766
sample estimates:
mean in group Africa mean in group Europe
            57.01227             78.33071
```

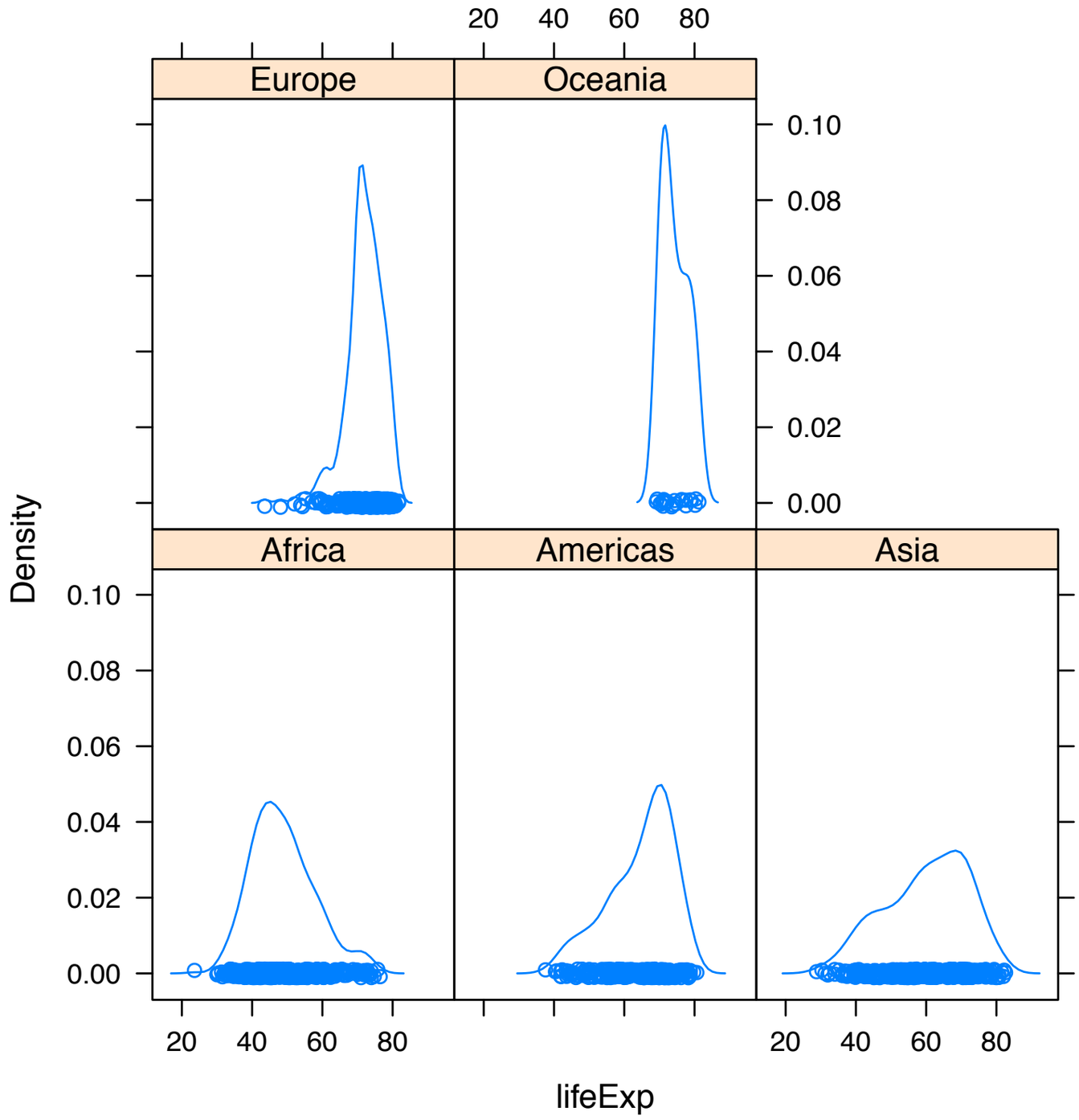
```
y ~ x | z + k
y ~ x | z * k
```

In lattice plotting functions, this says to plot y against x for every combination of levels of z and k (both assumed to be categorical). Note that the "+" and "*" do the same thing in this context, **which is not true in, e.g., a linear model**.

So, the formula syntax in lattice (and that of plyr) is **inspired** by the model formula syntax -- and deliberately so -- but it is not exactly the same.

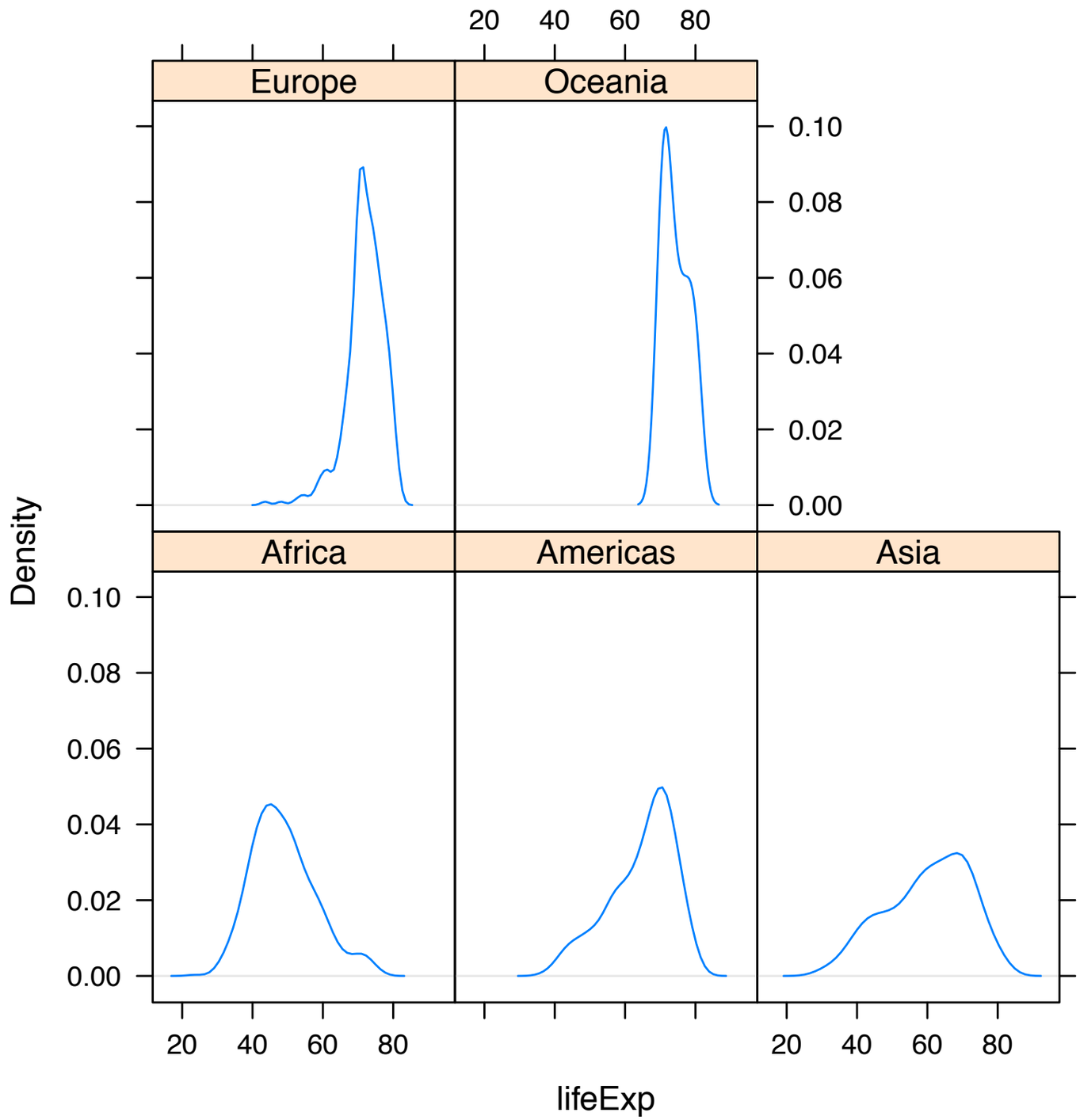
watch my formulas in the following graphing examples to see more ways to use the formula interface

end digression

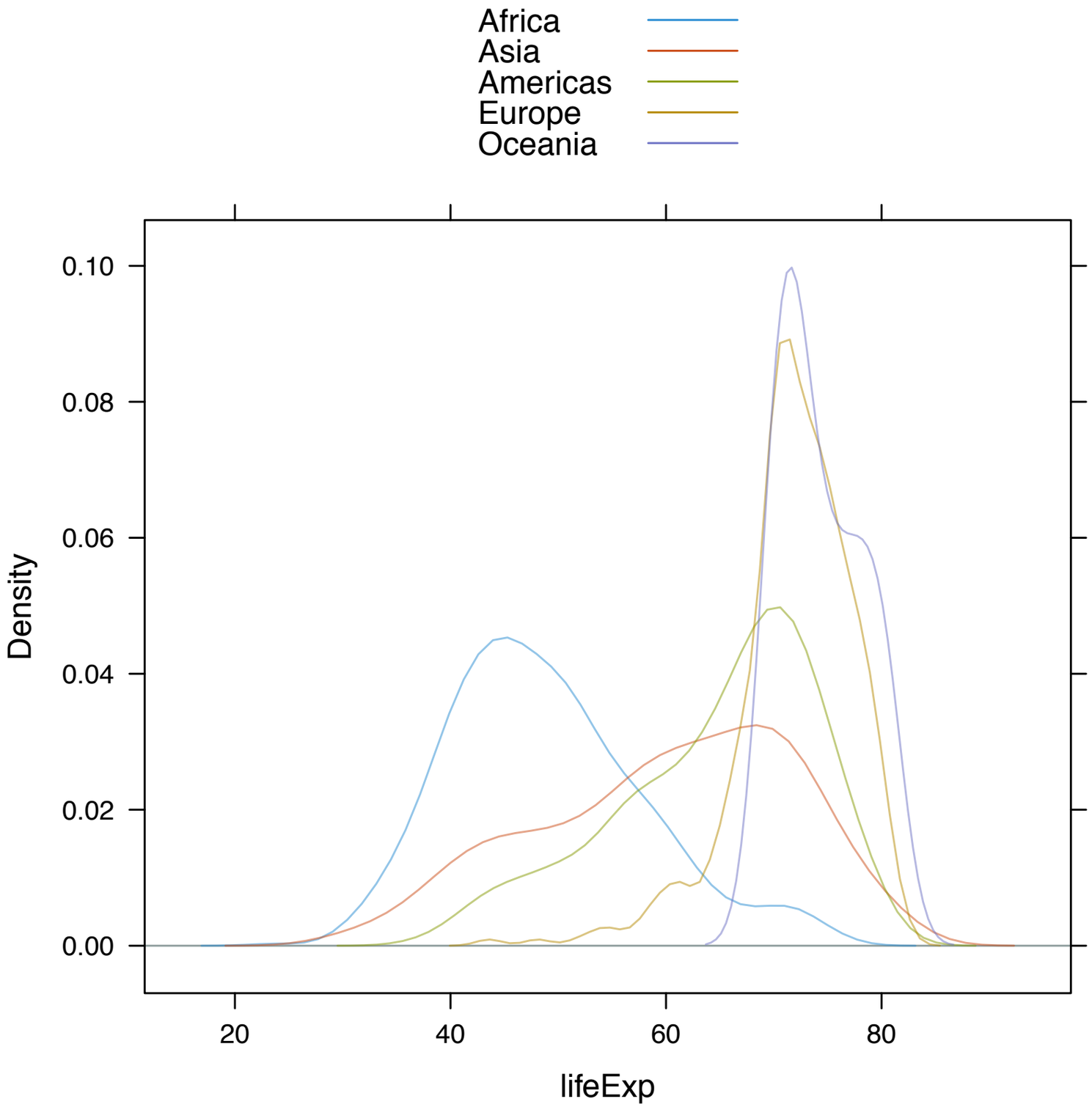



```
densityplot(~ lifeExp | continent, gDat)
```

```
densityplot(~ lifeExp | continent, gDat,
            plot.points = FALSE, ref = TRUE)
```



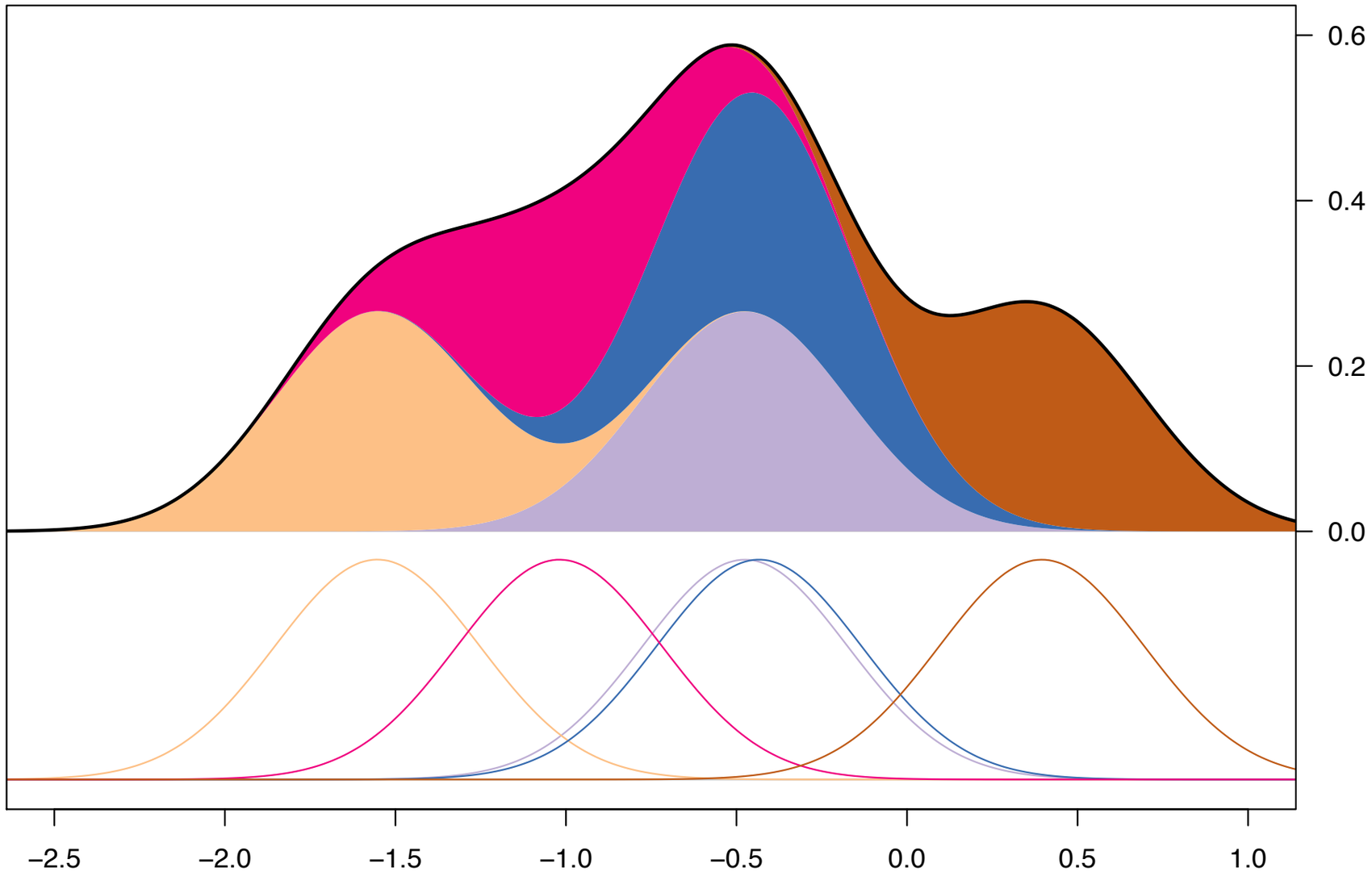
```
densityplot(~ lifeExp, gDat,
            groups = reorder(continent, lifeExp), auto.key = TRUE,
            plot.points = FALSE, ref = TRUE)
```



ability to superpose to facilitate direct visual comparison is big advantage of densityplot over histogram

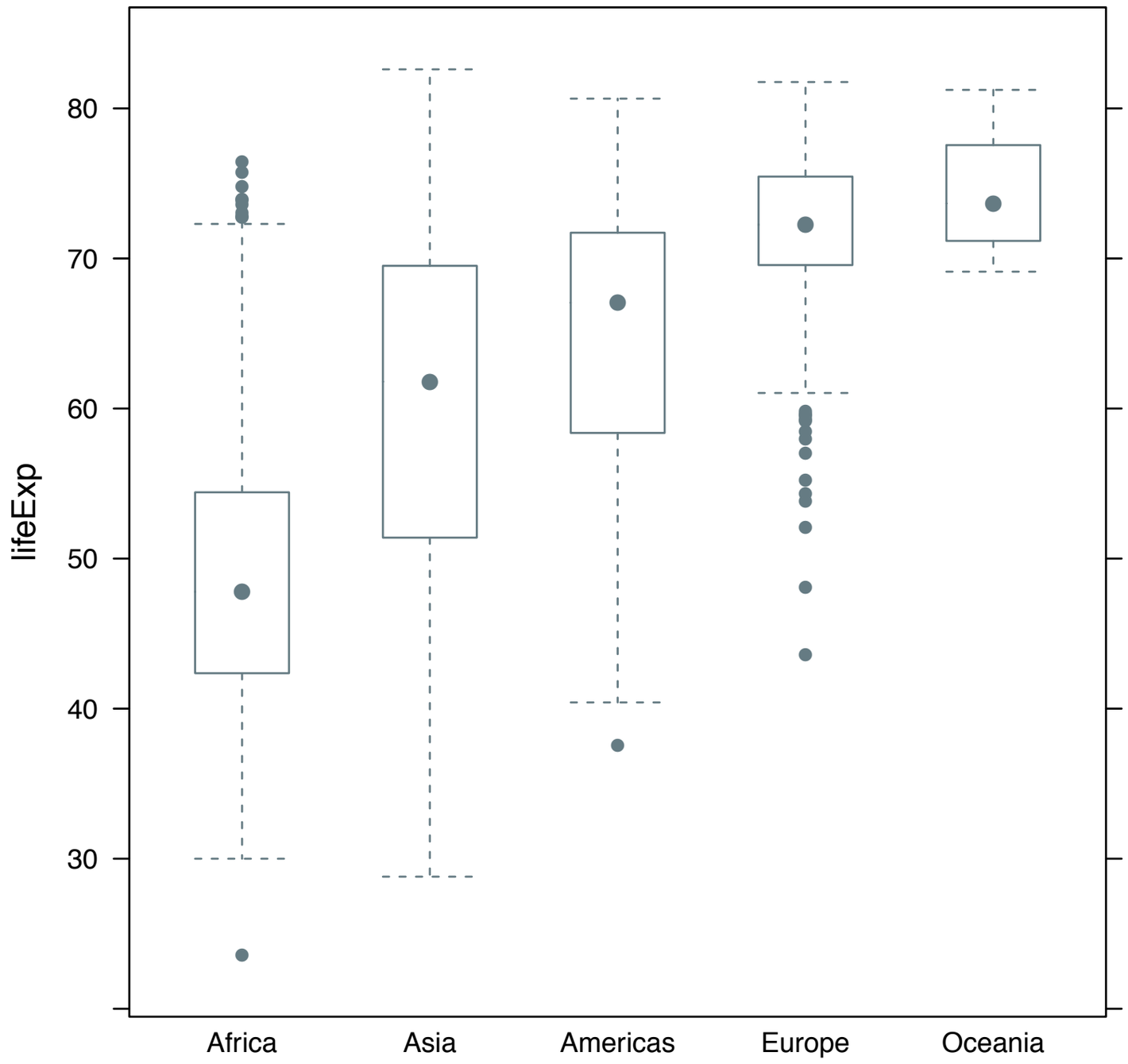
using reorder() again so that order in key better matches order of the distributions

Illustration of kernel density estimation



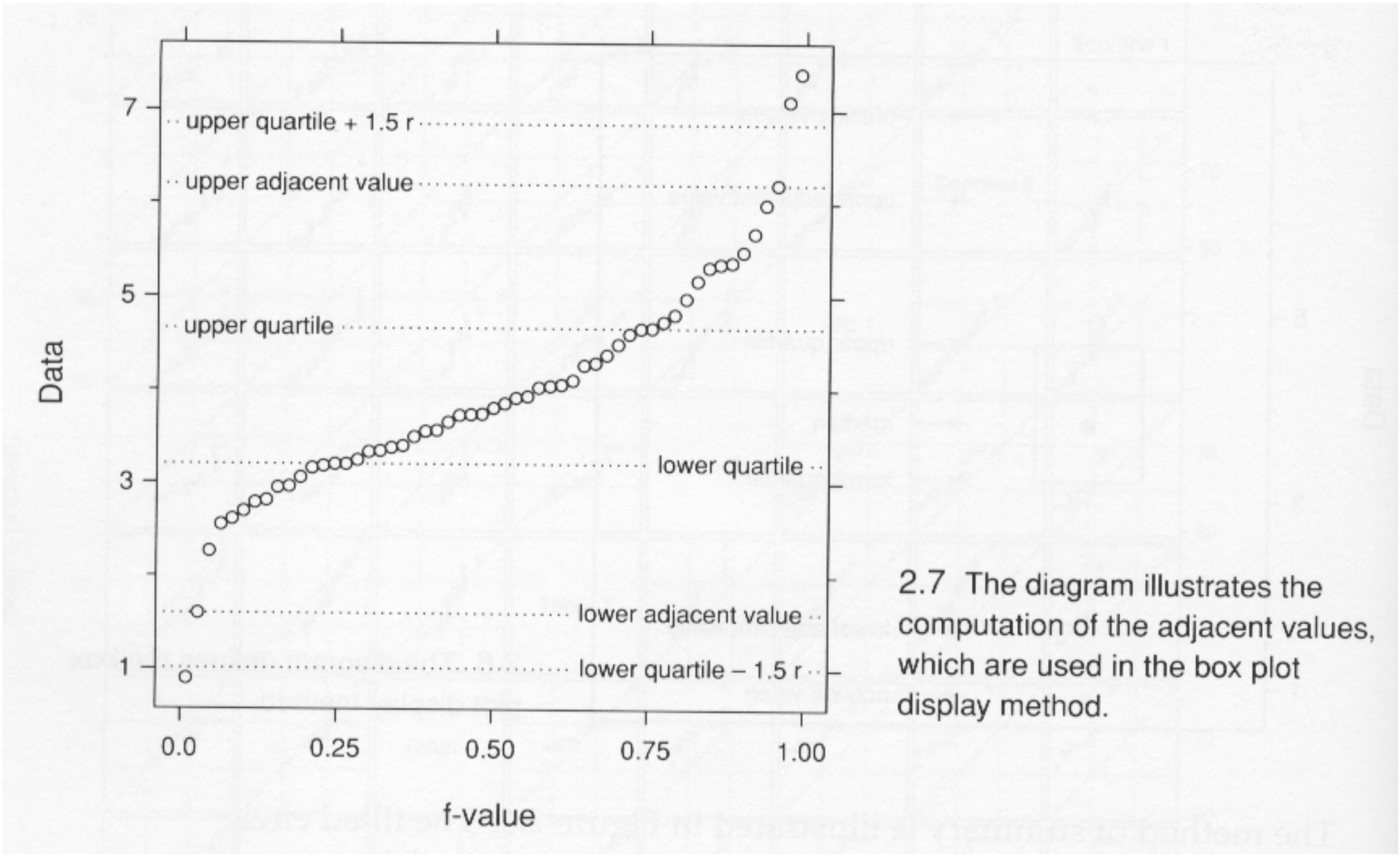
Produced from code at the R graph gallery

boxplot or `box and whiskers' plot (hence 'bwplot()')




```
bwplot(lifeExp ~ reorder(continent, lifeExp), gDat)
```

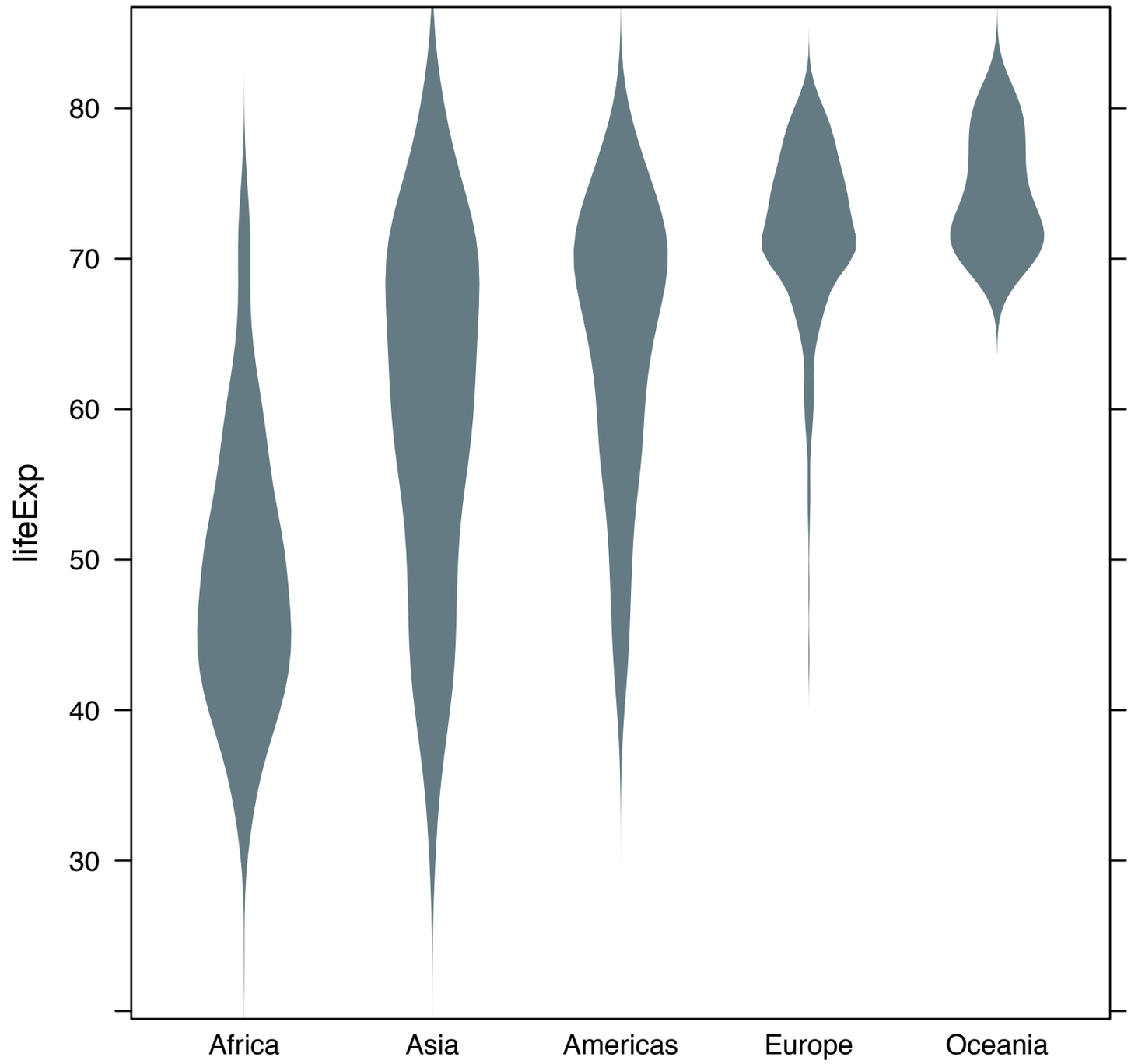
Where boxplots come from



2.7 The diagram illustrates the computation of the adjacent values, which are used in the box plot display method.

Figure from Visualizing Data by Cleveland.

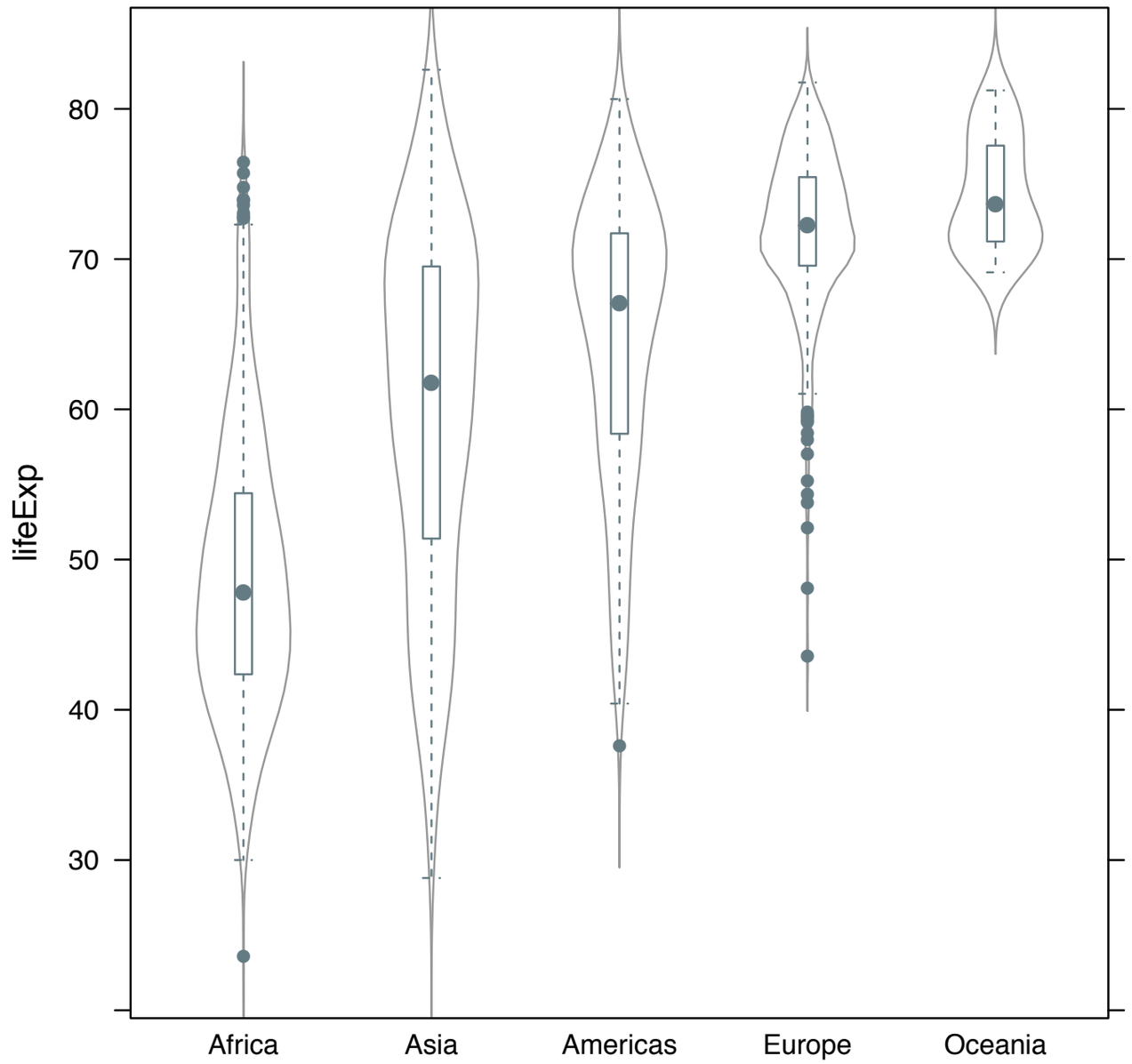
violin plot



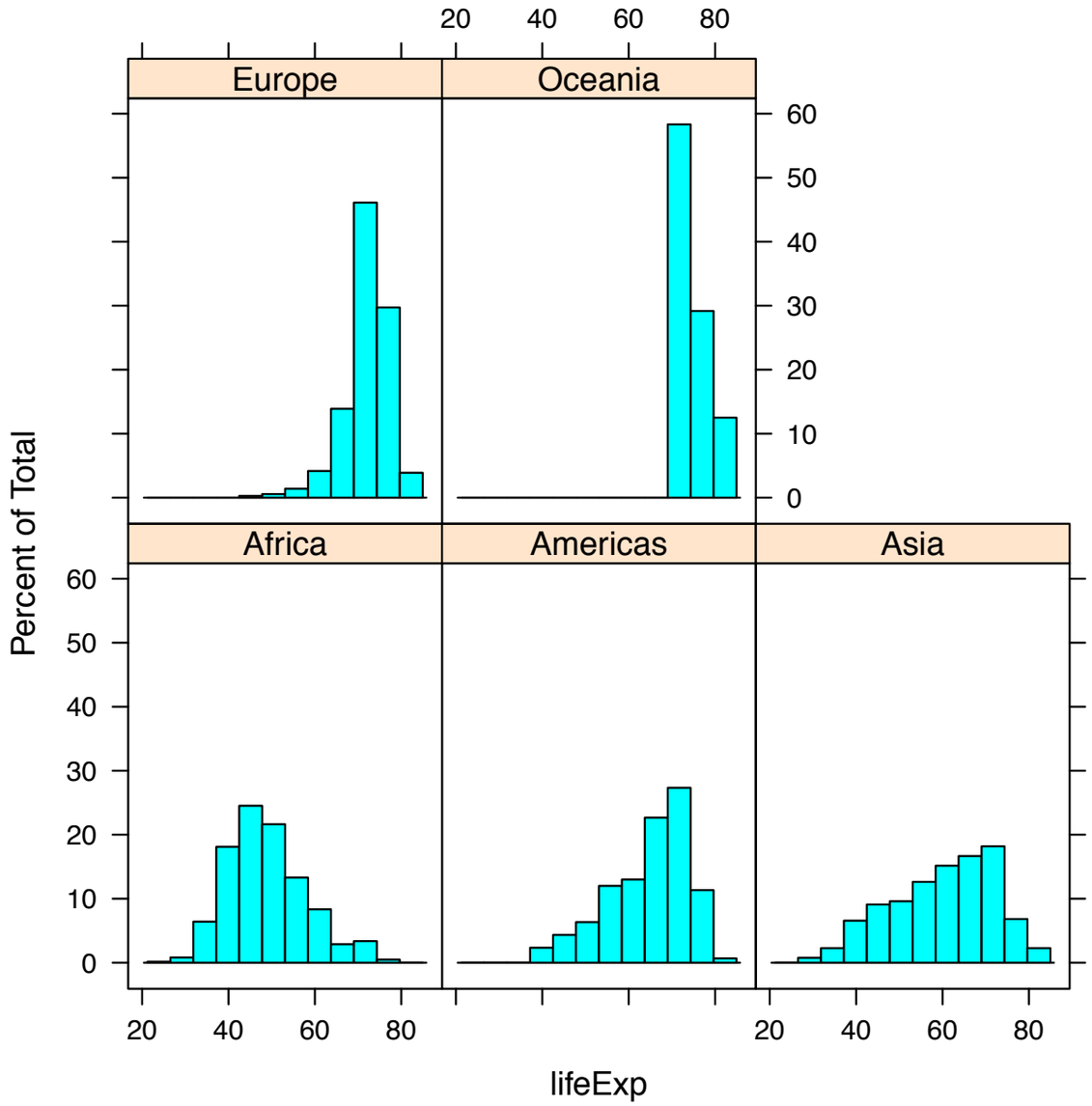

```
bwplot(lifeExp ~ reorder(continent, lifeExp), gDat,
       panel = panel.violin)
```

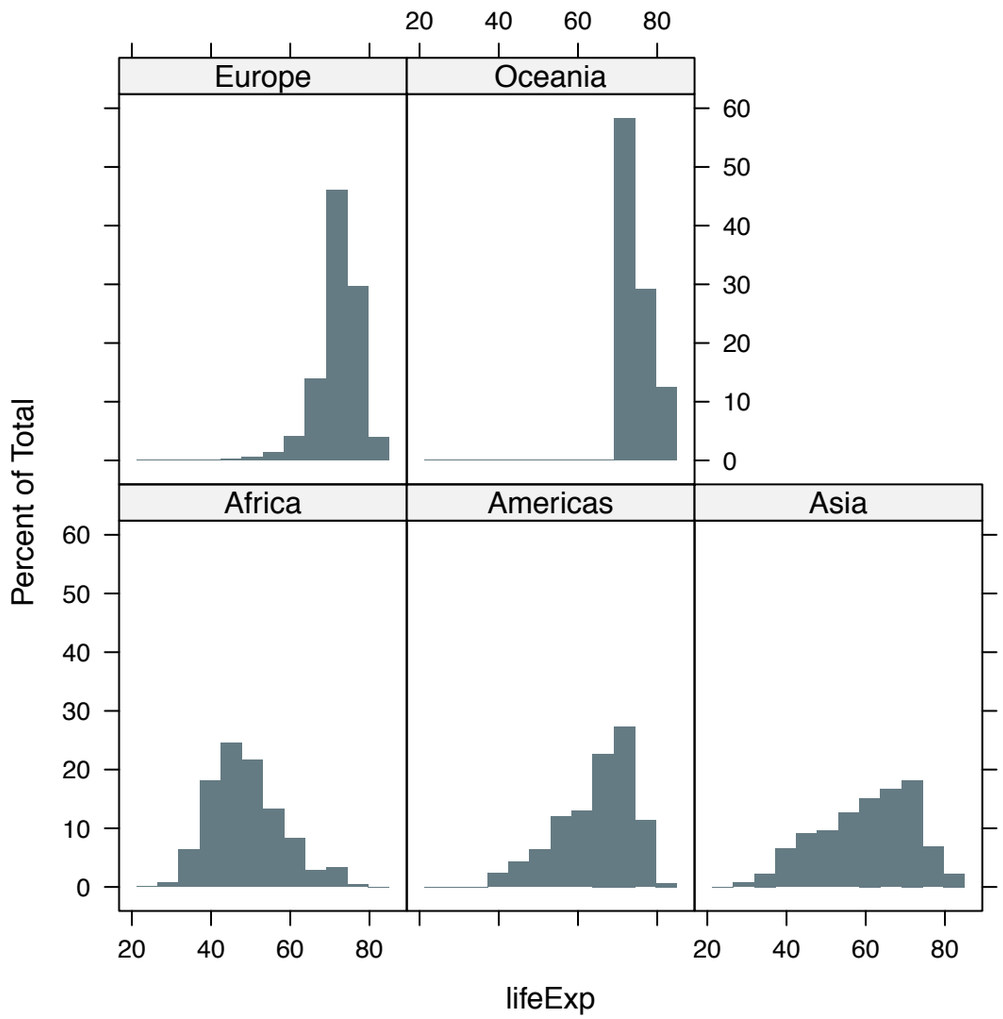
Note: I will talk explicitly about panel functions elsewhere.

violin plot + boxplot



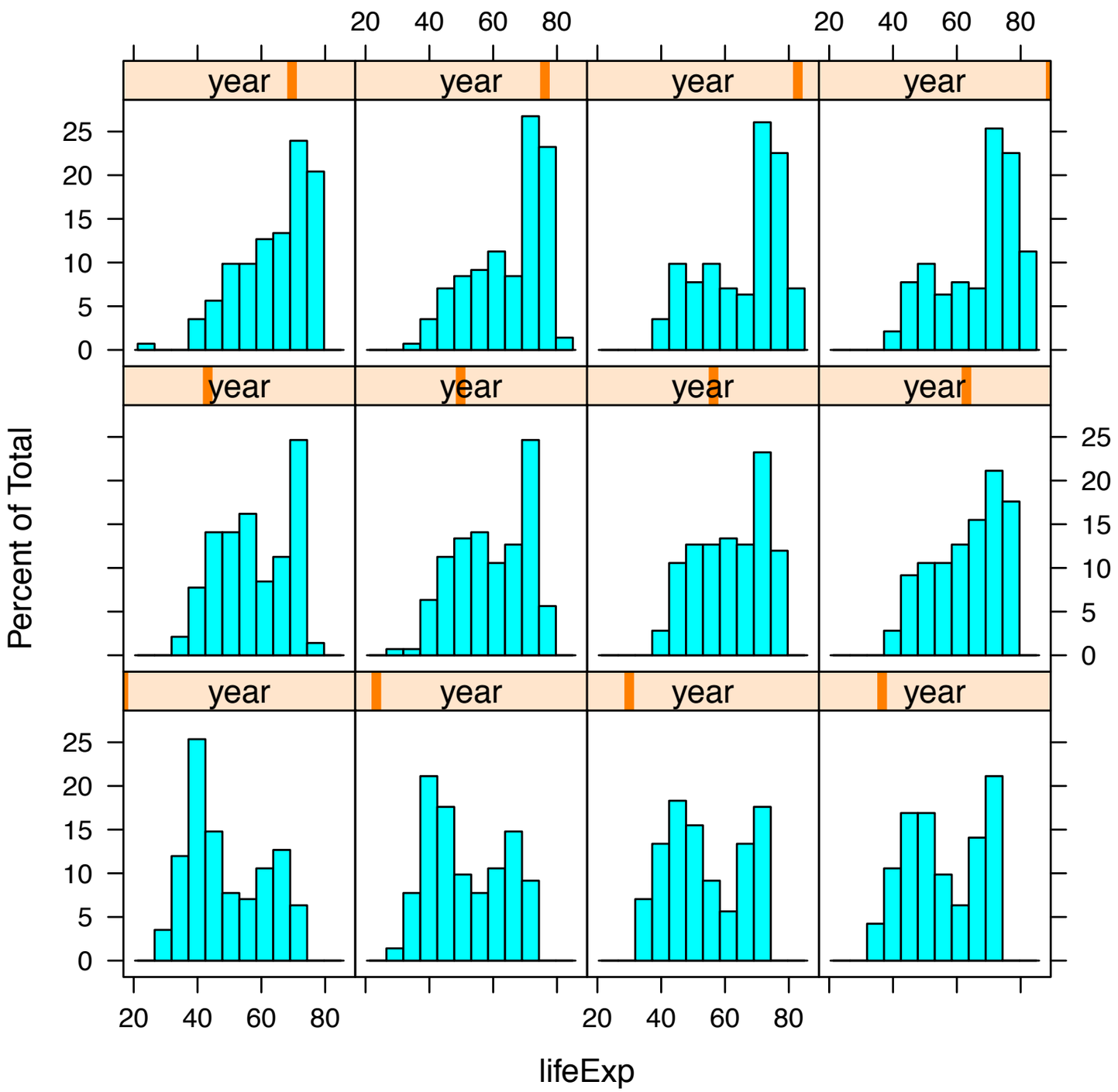

```
bwplot(lifeExp ~ reorder(continent, lifeExp), gDat,
       panel = function(..., box.ratio) {
         panel.violin(..., col = "transparent", border = "grey60",
                      varwidth = FALSE, box.ratio = box.ratio)
         panel.bwplot(..., fill = NULL, box.ratio = .1)
       })
```

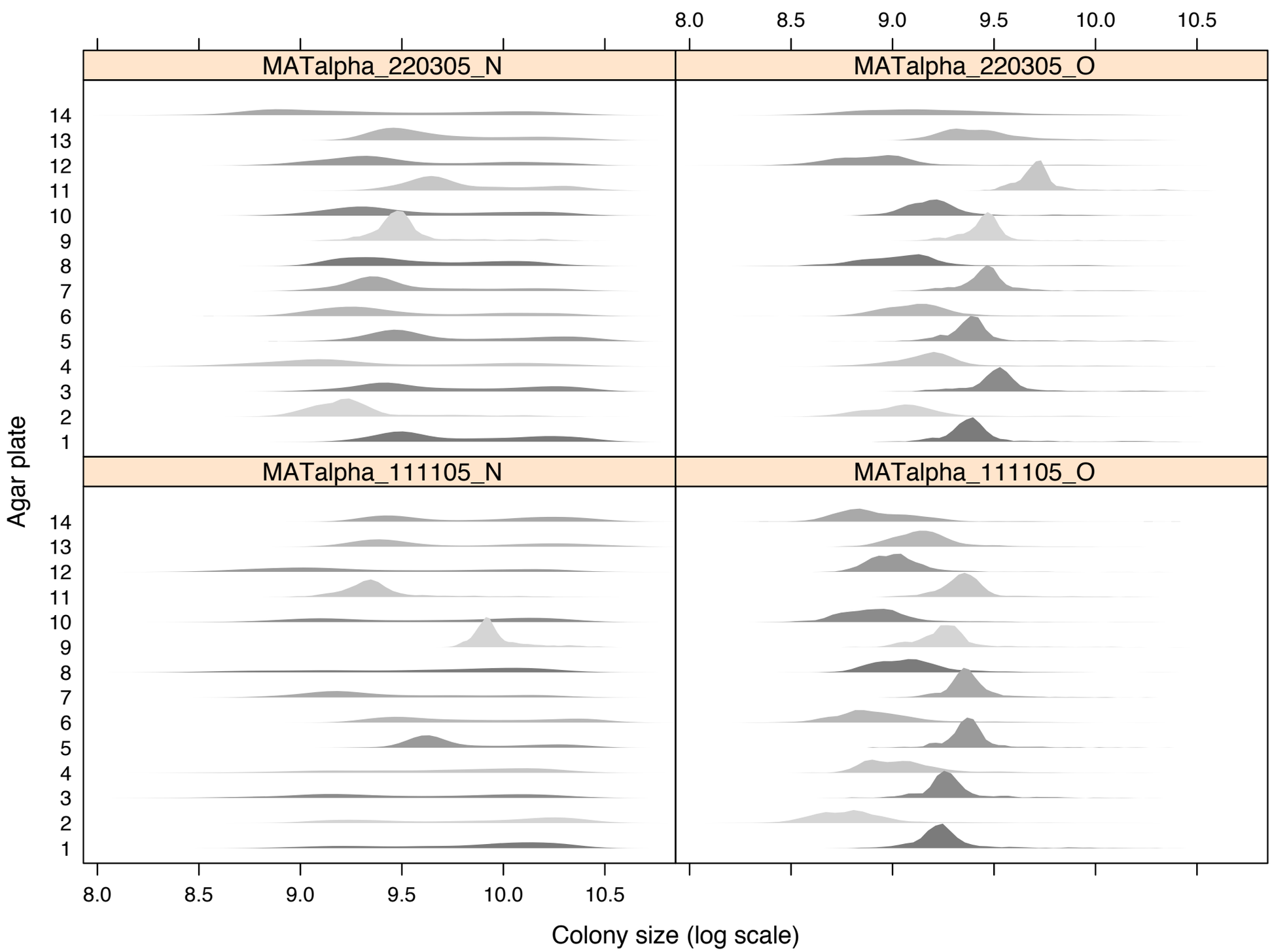





```
histogram(~ lifeExp | continent, gDat)
```

```
histogram(~ lifeExp | year, gDat)
```





I was so disappointed that y ~ x and y ~ x | z didn't work for densityplot, that I implemented that.

Why do I like densityplot better than histogram?

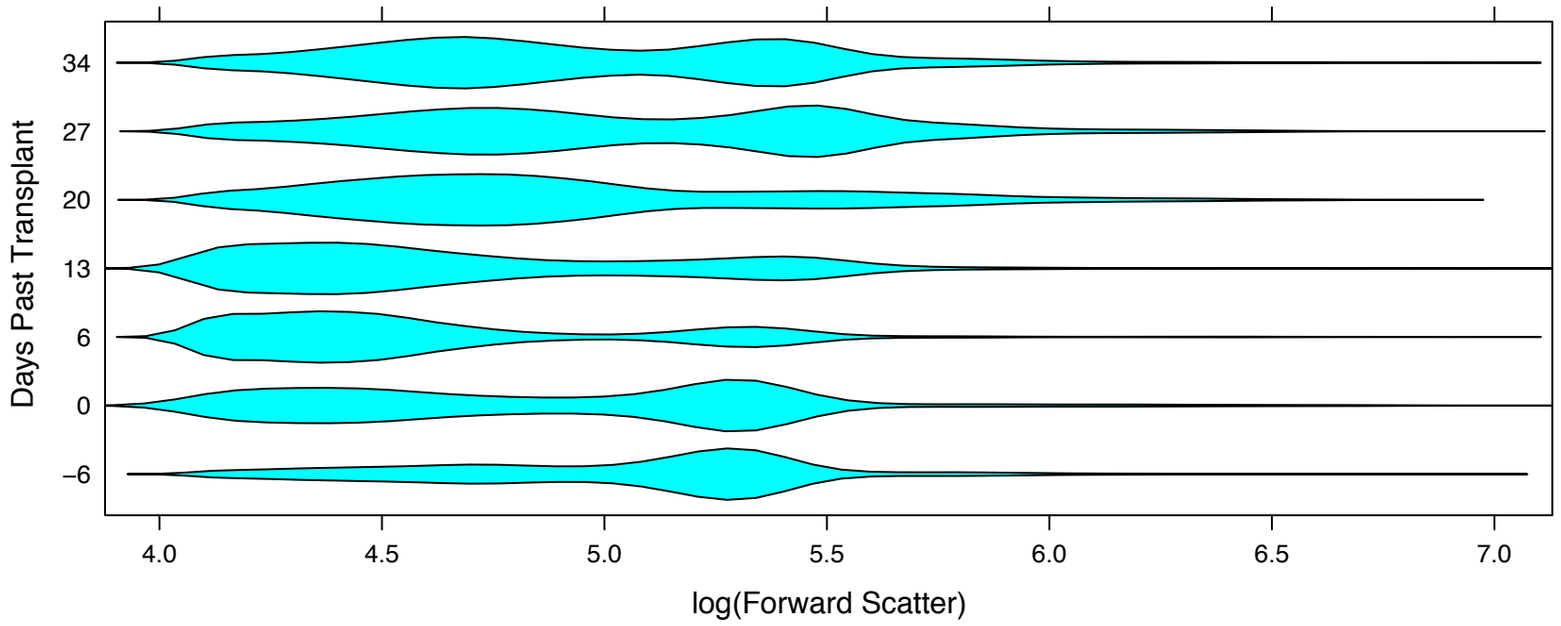
less sensitive (at least visually) to arbitrary choice of tuning parameter (bandwidth for densityplot, bin boundaries for histogram)

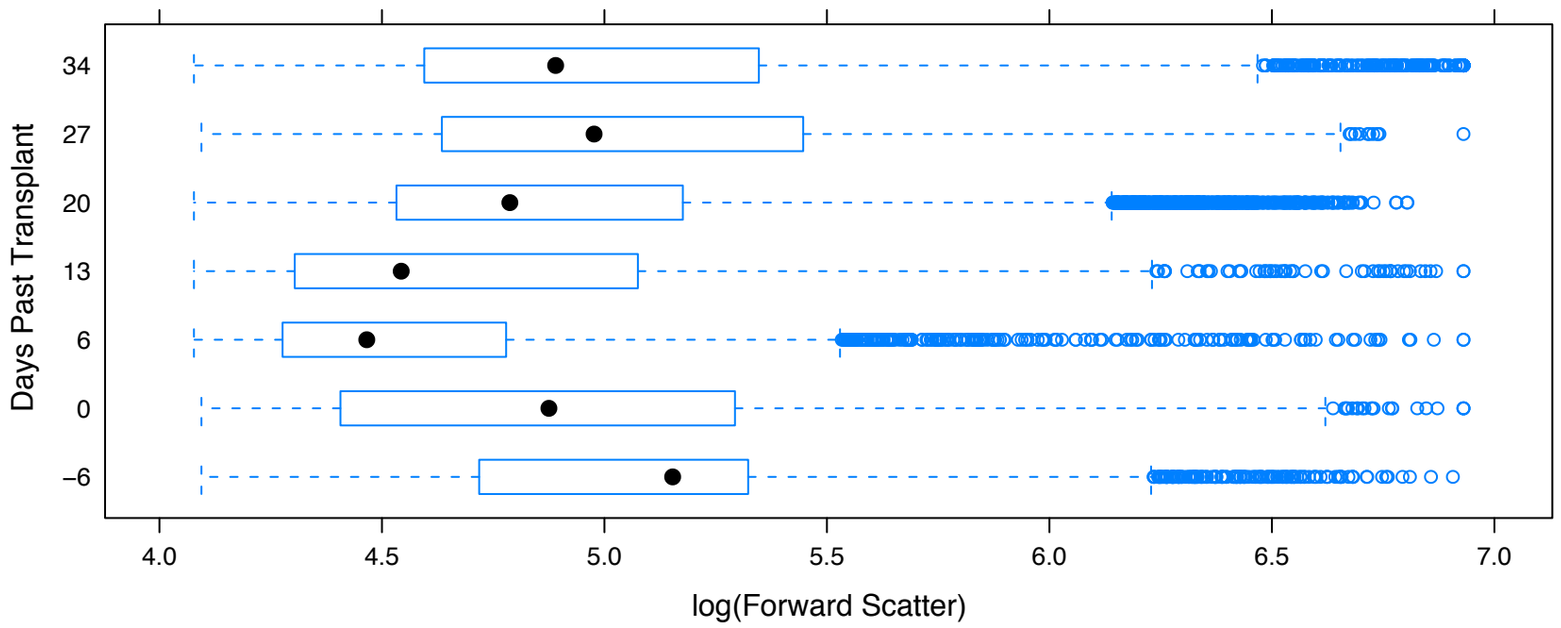
ability to superpose

natural to include raw observed data in a rug

Why do I like violinplot and my version of densityplot better than boxplot? ability to spot bimodality

Where boxplots fail





gvhd10 package:latticeExtra R Documentation

Flow cytometry data from five samples from a patient

Description:

Flow cytometry data from blood samples taken from a Leukemia patient before and after allogenic bone marrow transplant. The data spans five visits.

Usage:

```
data(gvhd10)
```

Format:

A data frame with 113896 observations on the following 8 variables.

'FSC.H' forward scatter height values

'SSC.H' side scatter height values

'FL1.H' intensity (height) in the FL1 channel

'FL2.H' intensity (height) in the FL2 channel

'FL3.H' intensity (height) in the FL3 channel

'FL2.A' intensity (area) in the FL2 channel

'FL4.H' intensity (height) in the FL4 channel

'Days' a factor with levels '-6' '0' '6' '13' '20' '27' '34'

Violin plot > boxplot?

Figure 3.13

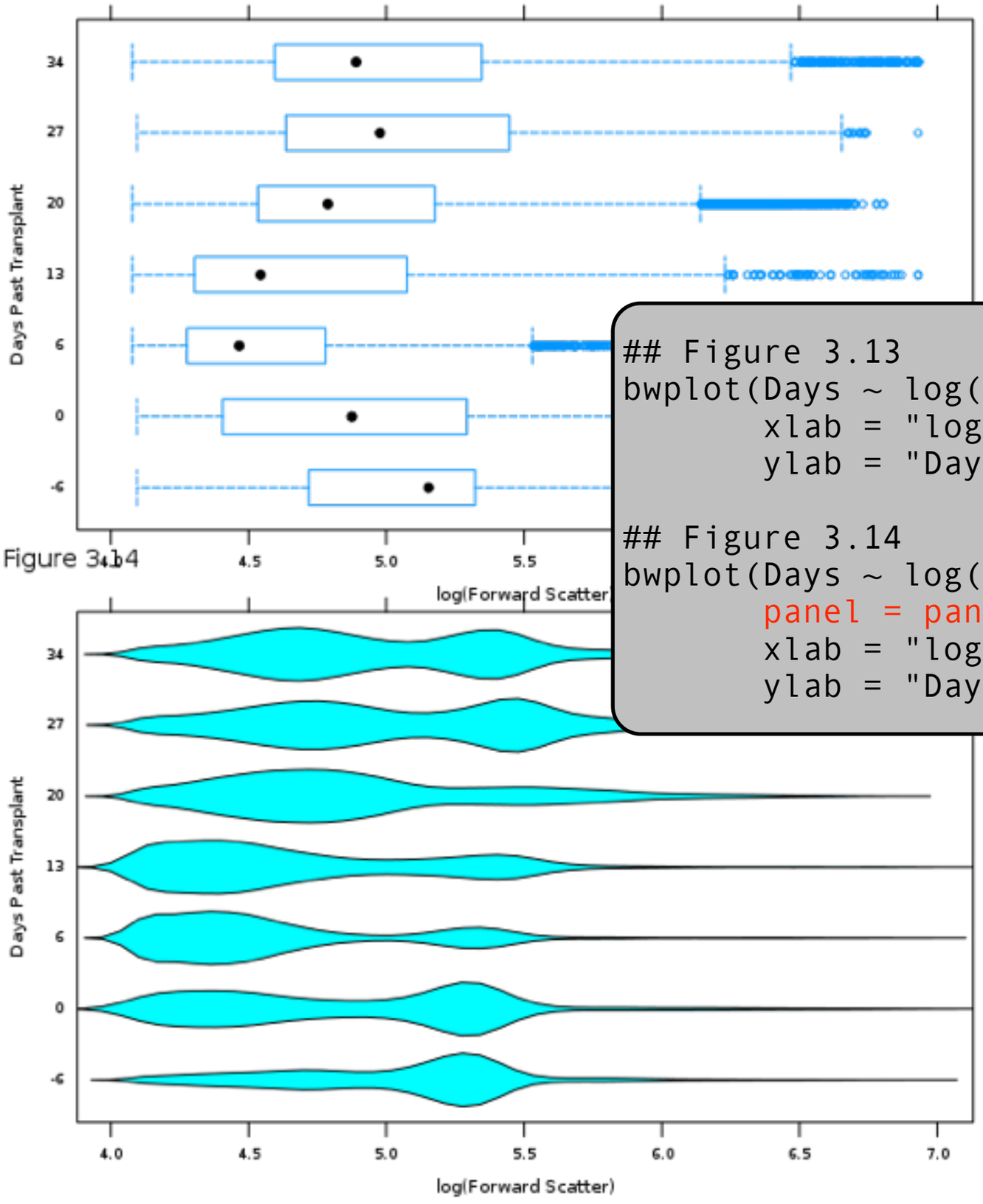


Figure 3.14


```
## Figure 3.13
bwplot(Days ~ log(FSC.H), data = gvhd10,
       xlab = "log(Forward Scatter)",
       ylab = "Days Past Transplant")

## Figure 3.14
bwplot(Days ~ log(FSC.H), gvhd10,
       panel = panel.violin, box.ratio = 3,
       xlab = "log(Forward Scatter)",
       ylab = "Days Past Transplant")
```

what about “empirical cumulative distribution plots” or ECDF plots?

Personally, I don’t have much use for them.

What is the empirical cumulative distribution (ecdf)?

$$\hat{F}_n(x) = \frac{\#\, x_i\text{'s} \;\leq\; x}{n}$$

$$\hat{F}_n(x) = \frac{1}{n}\sum_i I(x_i \leq x)$$

A step function that increases by 1/n at every observed value of X. The NPMLE of F.

histogram vs. densityplot

not a huge difference in what you can see/learn

Figure 3.4

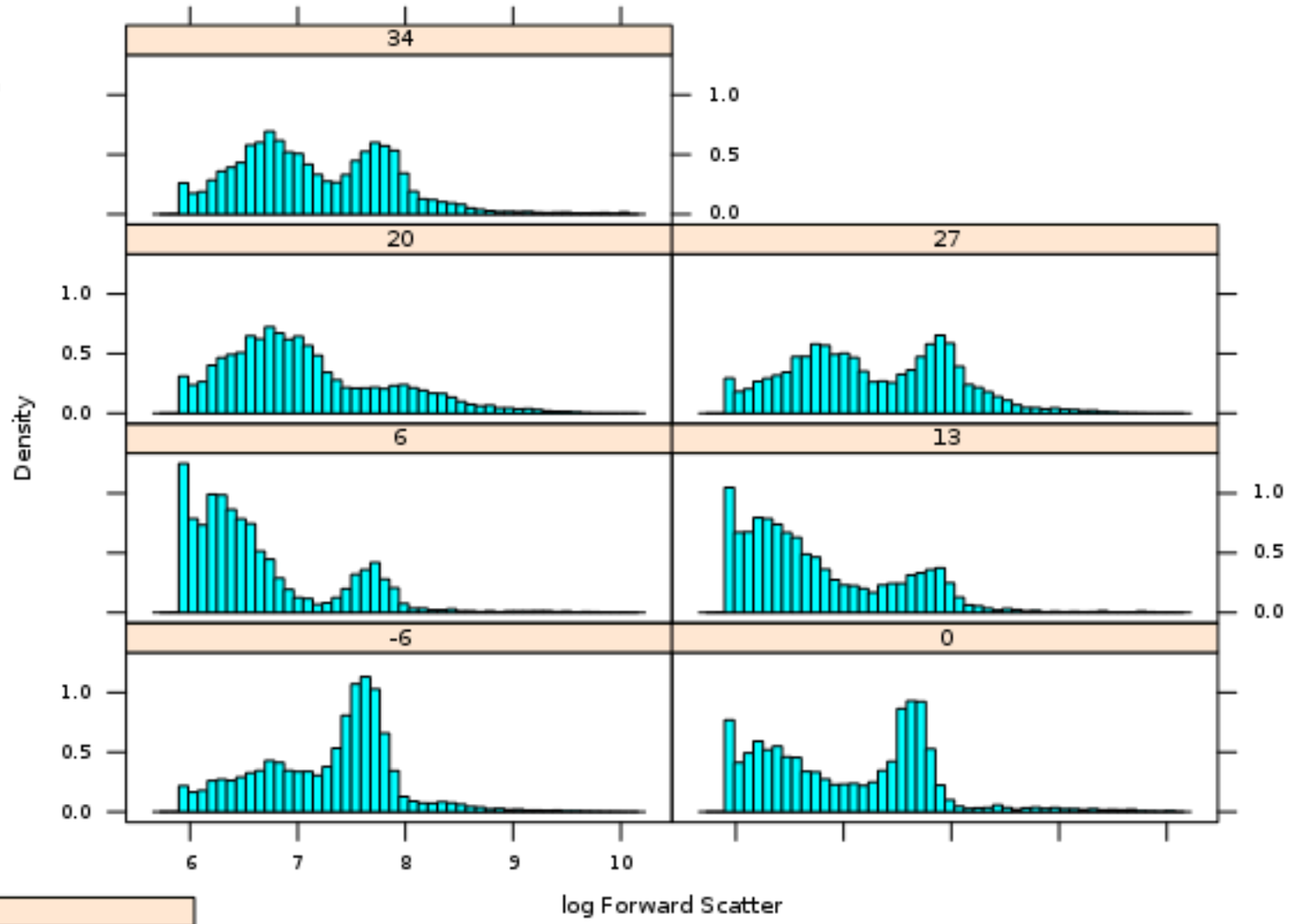
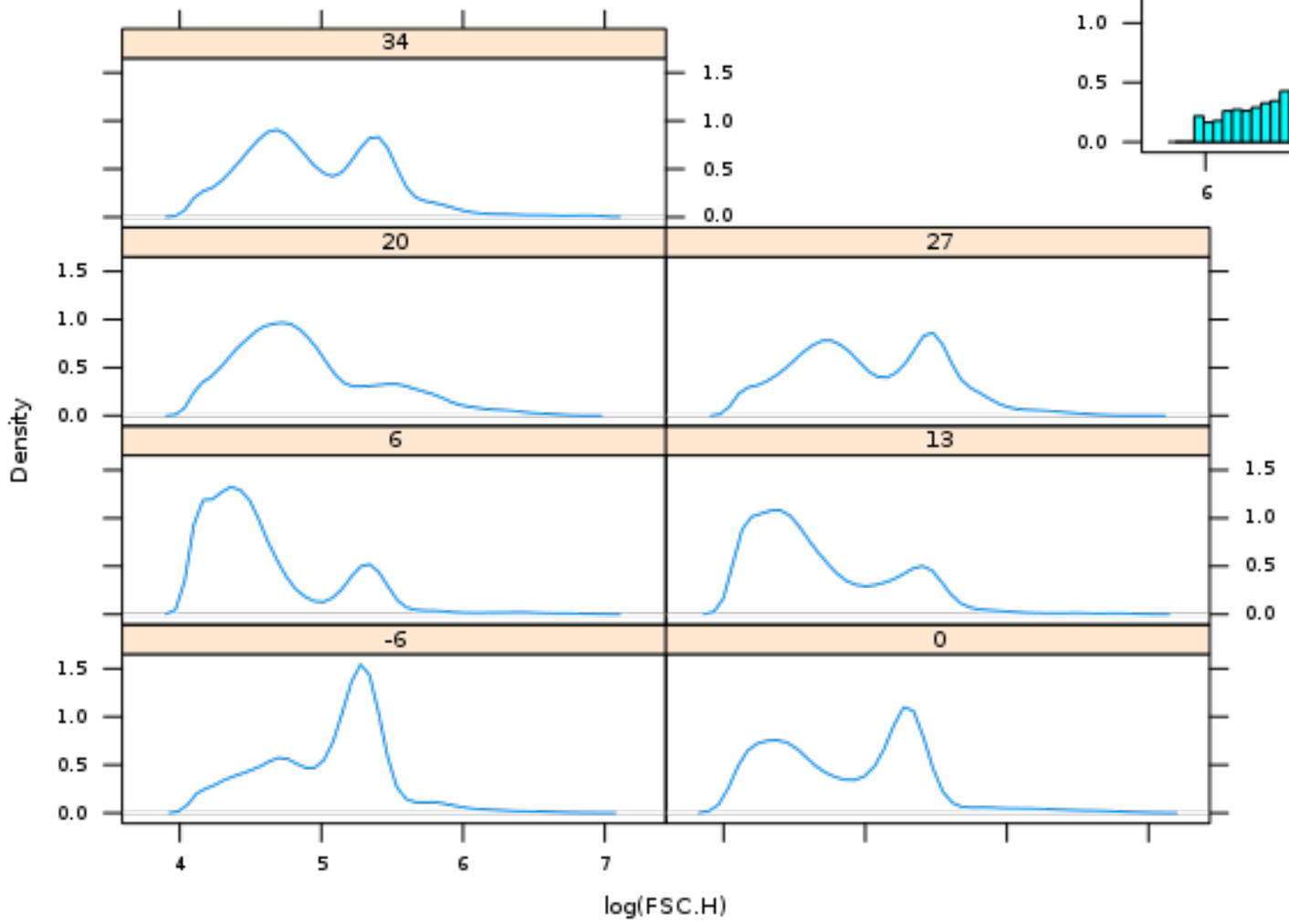


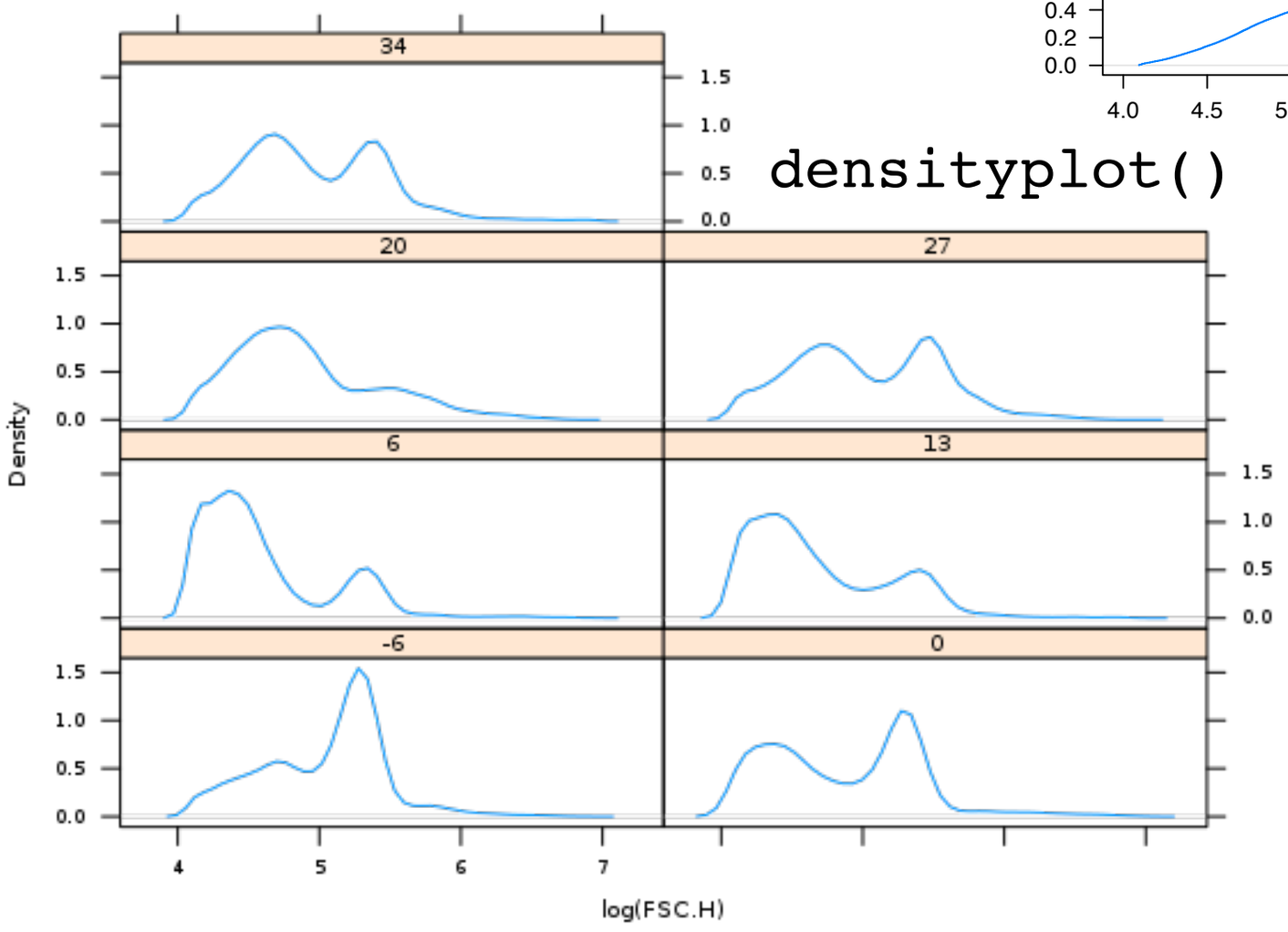
Figure 3.3

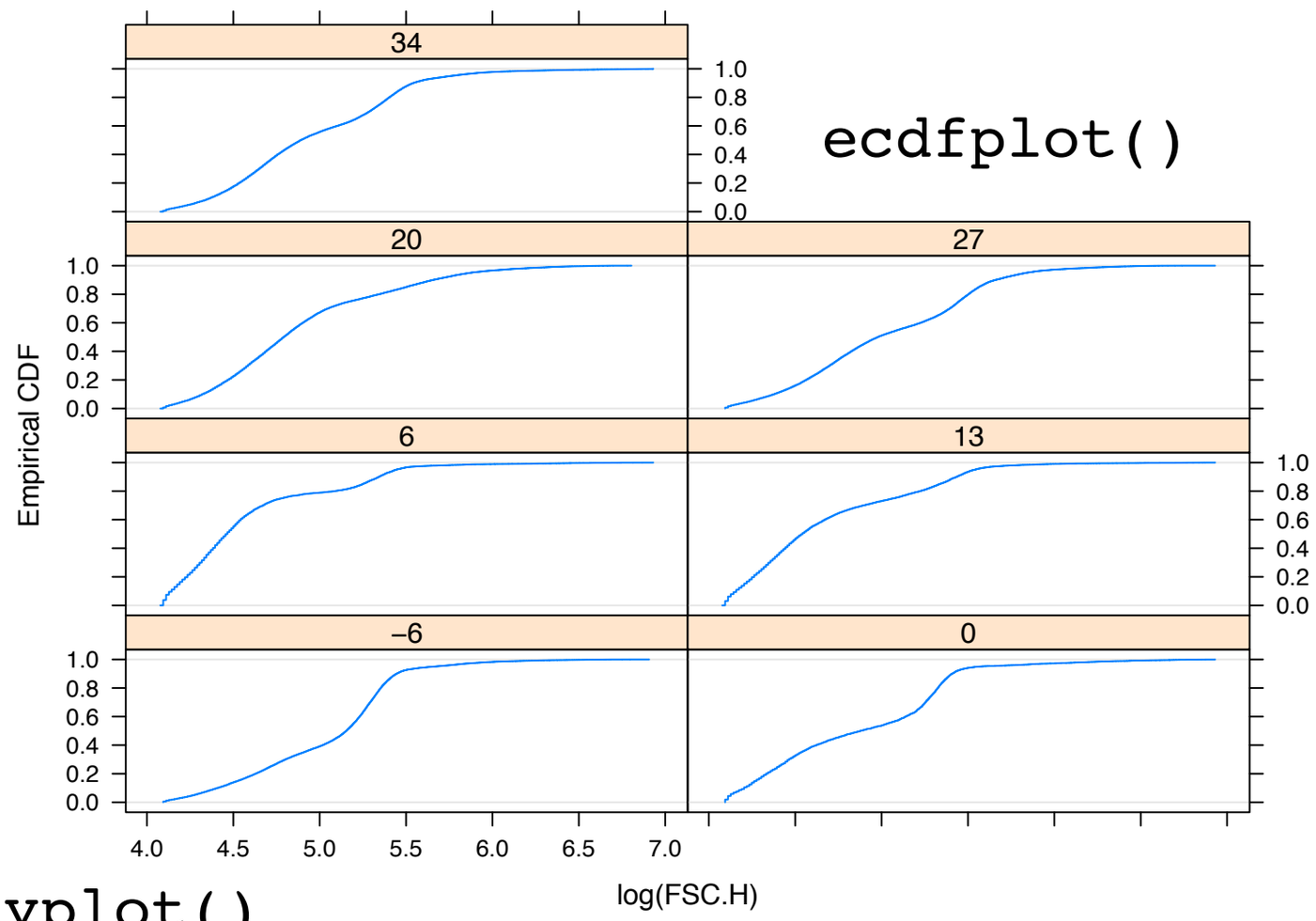


ecdfplot vs. densityplot

very different view of the data!

Figure 3.3





I cannot 'read' ecdfplots ... can you spot bimodality? What's the mean? Which distribution has greater spread?

For medium-to-large datasets,
main data visualizations driven by

1. the density f
   a. histogram
   b. kernel density estimate
2. the CDF F
   1. box-and-whisker plot
   2. empirical cumulative distribution function

See Ch. 3 of Sarkar

Main data visualizations driven by

1. the density f
   a. histogram (`histogram`)
   b. kernel density estimate (`densityplot`)
2. the CDF F
   1. box-and-whisker plot (`bwplot`)
   2. empirical cumulative distribution function (`ecdfplot`)

If `densityplot` and `bwplot` had a child ...
you might get a violin plot.

See Ch. 3 of Sarkar

functions from `lattice` or `latticeExtra`

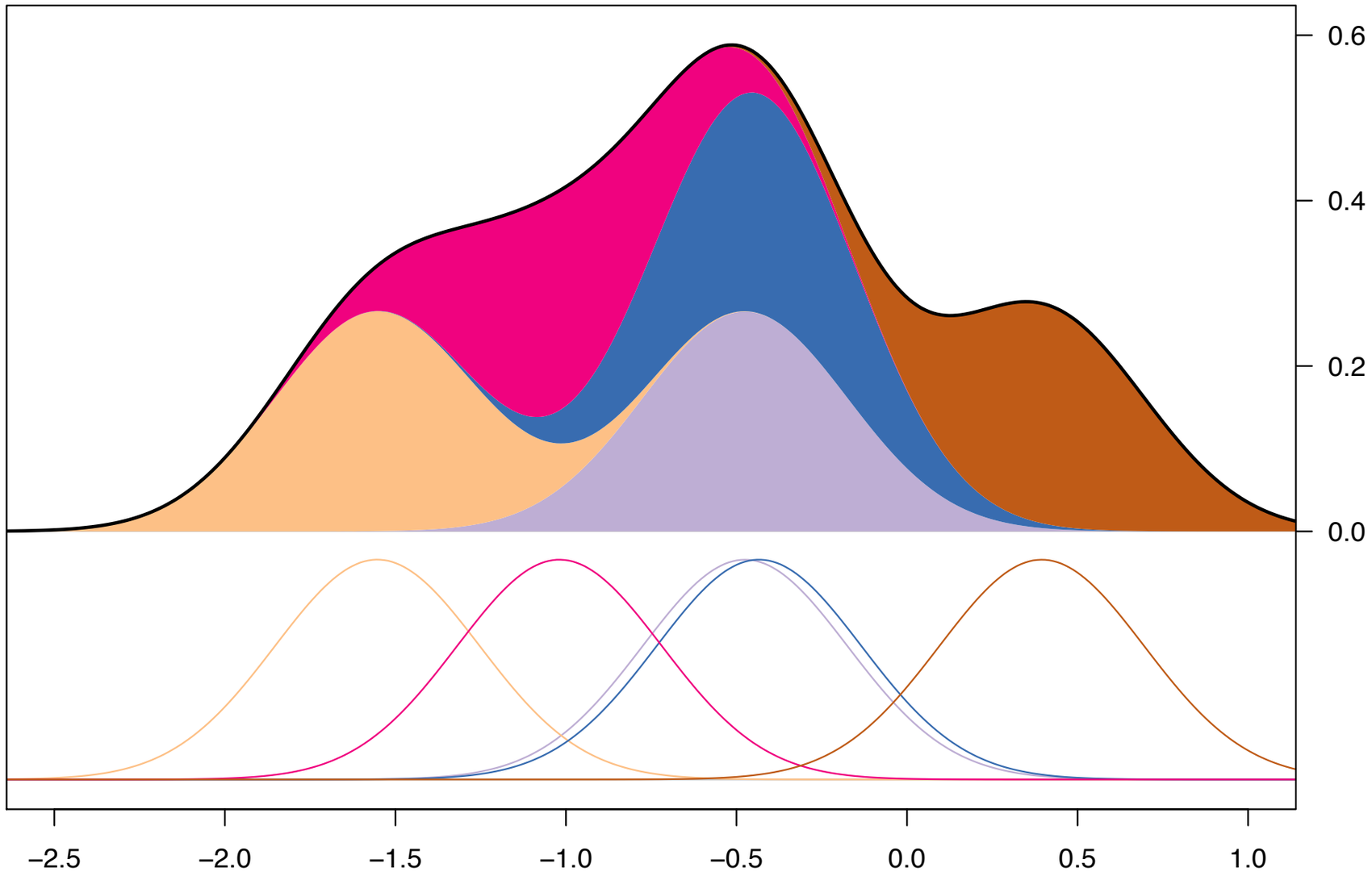
✓=possible/ sensible	`~ x`	`y ~ x`	`~ x | y`	`~ x , groups = y`
`stripplot`		✓		
`bwplot`		✓		
`histogram`	✓		✓	
`densityplot`	✓	*	✓	✓
`ecdfplot`	✓		✓	✓

* I've actually extended densityplot to work here, for personal use. See other page.

Visualizing dist'n of X (given Y = y)

- I favor smooth histograms = density estimates. Path of least resistance is `densityplot`.
- Observed data, if sample small enough, can be overlaid via points or rug.
- In small datasets, strip plot is good, especially with summary statistic, such as median, overlaid.
- Boxplots and, in some very special cases, ecdf plots, seem useful. I like violin plots.
- Honestly, hard to find advantage of histograms, given all the other options.

Illustration of kernel density estimation



Produced from code at the R graph gallery

brief introduction to kernel density estimation

based on Camila Souza's presentation in STAT 545A (2008)

Here's something that IS available via STATSnetBase (and seems to have been a source for this material in the first place!):

Chapter 8, Density Estimation: Erupting Geysers and Star Clusters from
A Handbook of Statistical Analyses Using R, Second Edition
Torsten Hothorn and Brian S . Everitt
Chapman and Hall/CRC 2009
Pages 139–159
Print ISBN: 978-1-4200-7933-3
eBook ISBN: 978-1-4200-7934-0
DOI: 10.1201/9781420079340.ch8

JB succeeded in getting as PDF!
Maybe this link will work for you (?):
http://www.crcnetbase.com/doi/pdfplus/10.1201/9781420079340.ch8
otherwise get on STATSnetBASE yourself and search and click ....

Histogram

Well-established, widely-practiced method of density estimation.

Basic principle: count the number of observations in an interval of size h, called a *bin*. Formally bin B_j is:

$$B_j = [x_0 + (j-1)h, x_0 + jh], j = 1,2,...k$$

The histogram density estimate is:

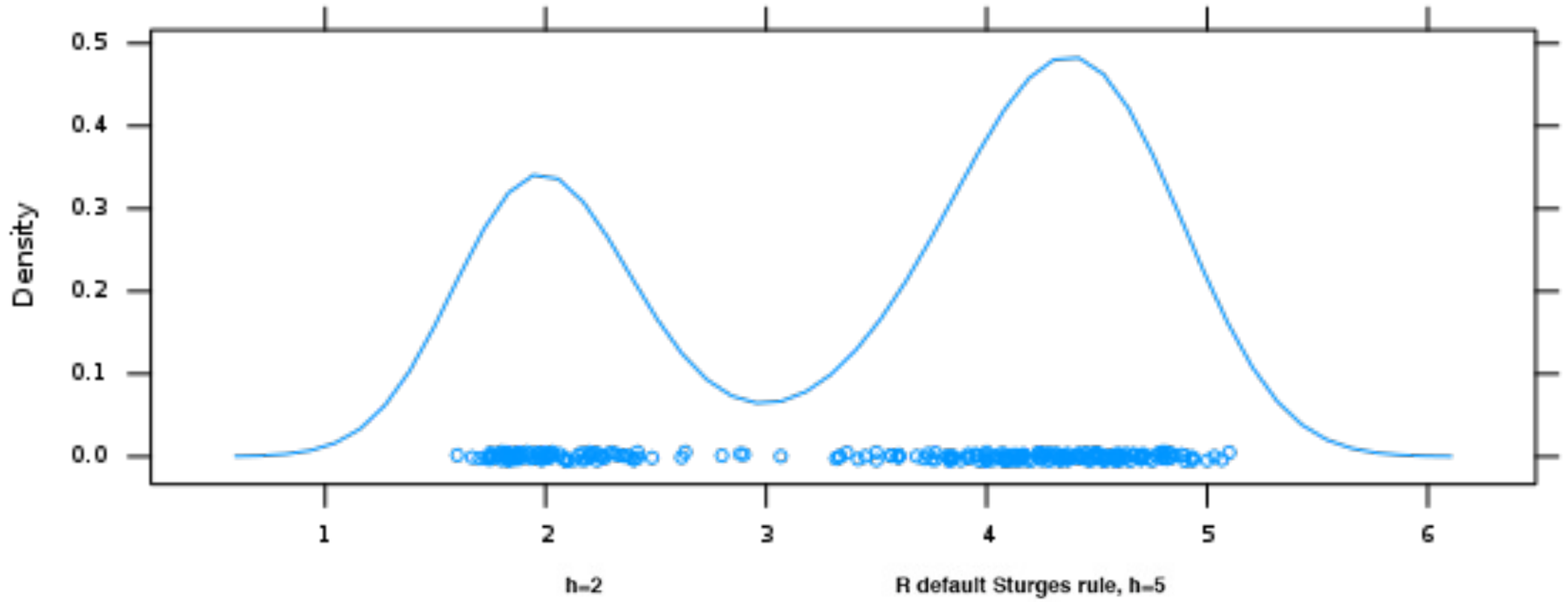
$$\hat{f}_h(x) = \frac{1}{nh} \sum_i \sum_j I(x_i \in B_j) I(x \in B_j)$$

Histogram

Crucial 'tuning' parameter for histogram density estimation: the bins (or bin widths or number of bins)

hist() base R	$k = 1 + \log_2 n$
truehist() MASS	$h = 3.5\hat{\sigma}n^{-1/3}$
histogram() lattice	$k = \text{round}(1 + \log_2 n)$

`faithful` data set



bin selection has a huge impact on the result!

Naive estimator, uniform kernel estimator

Remember definition of the density f:

$$f(x) = \lim_{h \to 0} \frac{1}{2h} P(x - h < X < x + h)$$

Therefore, for small h, a crude estimator of f is:

$$\hat{f}_h(x) = \frac{1}{2nh} [\# \, x_i \in (x - h, x + h)]$$

Naive estimator, uniform kernel estimator

Therefore, for small h, a crude estimator of f is:

$$\hat{f}_h(x) = \frac{1}{2nh}[\#\, x_i \in (x-h, x+h)]$$

Define a weight function:

$$w(x) = \frac{1}{2} \text{ if } |x| < 1 \text{ and } 0 \text{ otherwise}$$

And re-write the crude / naive estimator as:

$$\hat{f}_h(x) = \frac{1}{n}\sum_i \frac{1}{h} w\left(\frac{x - x_i}{h}\right)$$

Naive estimator, uniform kernel estimator

And re-write the crude / naive estimator as:

$$\hat{f}_h(x) = \frac{1}{n}\sum_i \frac{1}{h} w\left(\frac{x - x_i}{h}\right)$$

In plain English, place a box of width 2h and height $(2nh)^{-1}$ on each observation. Density estimate at any point x is the sum of these boxes.

Moving beyond a uniform (or rectangular) kernel

Let's replace the weight function with another function K that satisfies the following:

$$K(x) \geq 0$$

$$\int K(x)dx = 1$$

So K is a probability density function and, usually, is symmetric.

Tabela 1: *Kernels*

Kernel	$Kern(u)$
Uniform	$\frac{1}{2}I(\|u\| \leq 1)$
Triangle	$(1 - \|u\|)I(\|u\| \leq 1)$
Epanechnikov	$0.75(1 - u^2)I(\|u\| \leq 1)$
Gaussian	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}u^2)$

In general, the kernel estimator is given by:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_i K\left(\frac{x - x_i}{h} \right)$$

Tuning parameter h is called the *bandwidth*

In general, the kernel estimator is given by:

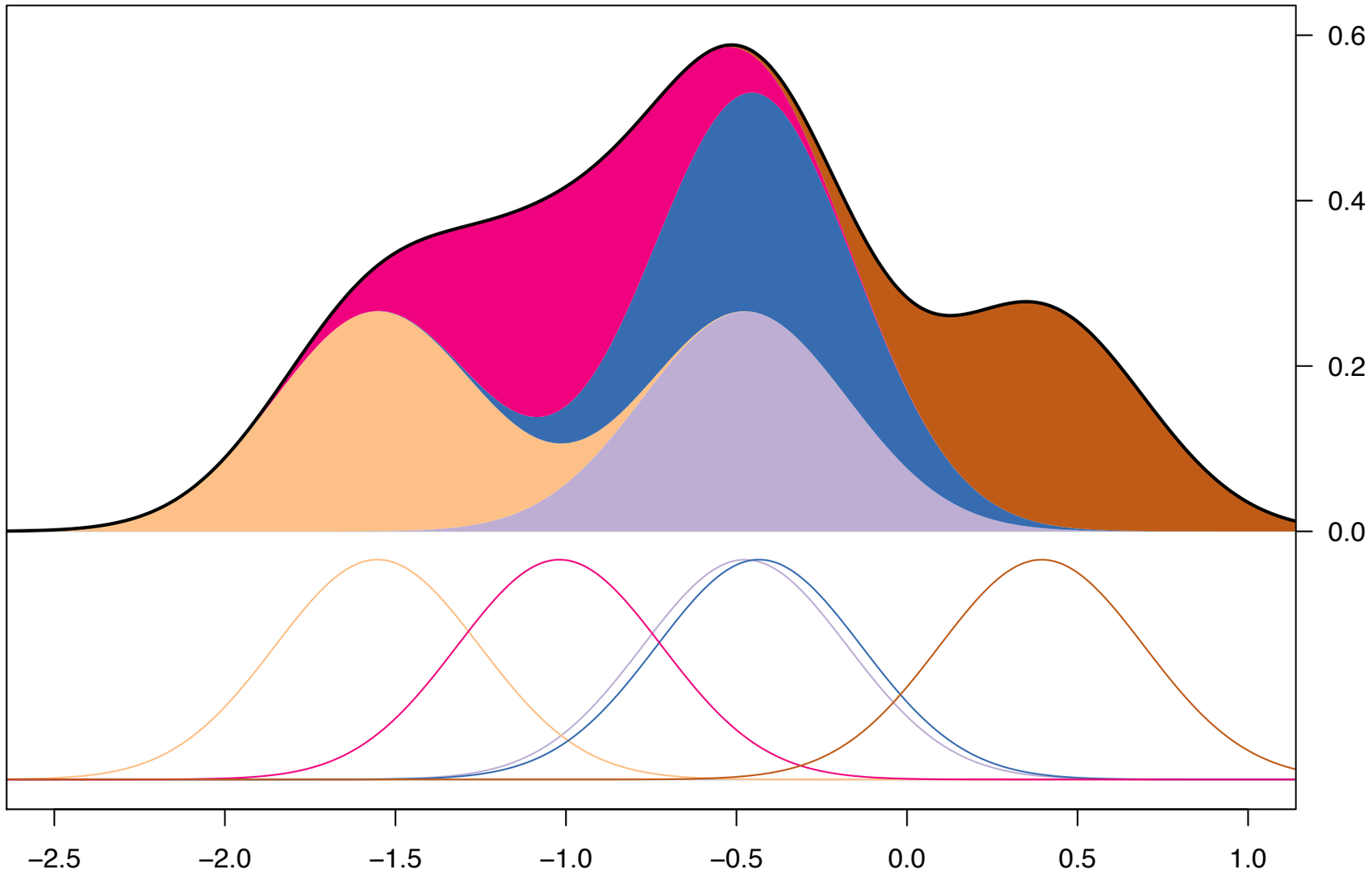
$$\hat{f}_h(x) = \frac{1}{nh} \sum_i K\left(\frac{x - x_i}{h} \right)$$

Tuning parameter h is called the *bandwidth*

Instead of a sum of boxes, the kernel estimator is a sum of 'bumps'. K determines the shape of the bumps and h determines their width.

Elsewhere, we can talk about how to choose h with cross-validation.

Illustration of kernel density estimation



Produced from code at the R graph gallery

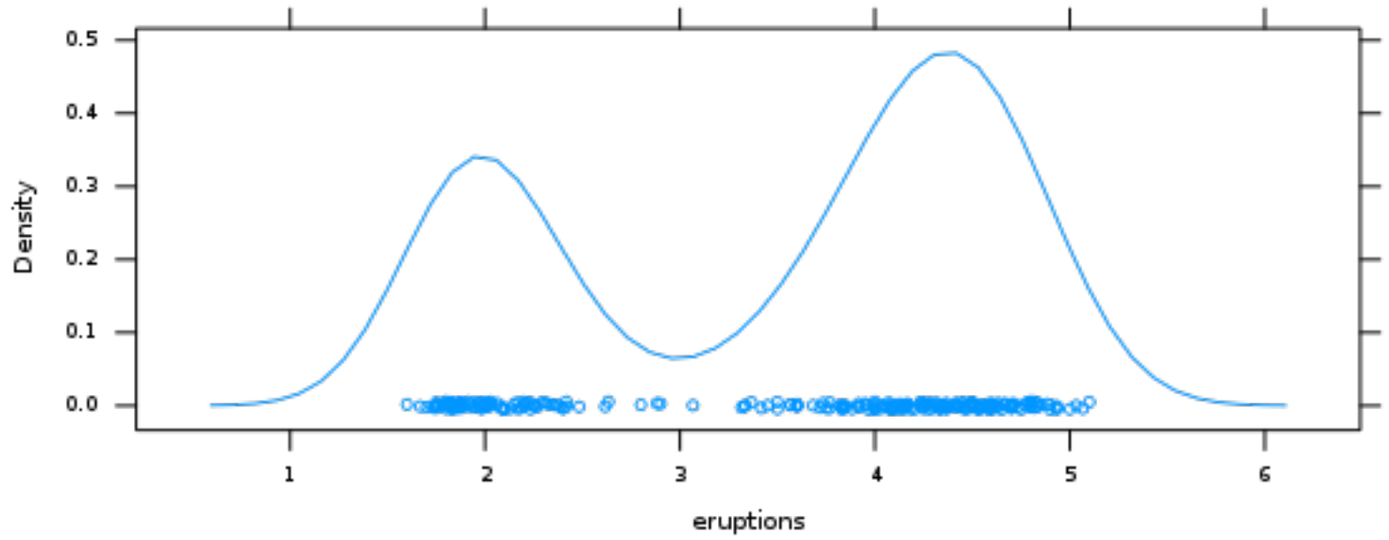
```
density                  package:stats                  R Documentation

Kernel Density Estimation

Description:

     The (S3) generic function 'density' computes kernel density
     estimates.  Its default method does so with the given kernel and
     bandwidth for univariate observations.

Usage:

     density(x, ...)
     ## Default S3 method:
     density(x, bw = "nrd0", adjust = 1,
             kernel = c("gaussian", "epanechnikov", "rectangular",
                        "triangular", "biweight",
                        "cosine", "optcosine"),
             weights = NULL, window = kernel, width,
             give.Rkern = FALSE,
             n = 512, from, to, cut = 3, na.rm = FALSE, ...)

Arguments:

       x: the data from which the estimate is to be computed.

      bw: the smoothing bandwidth to be used.  The kernels are scaled
          such that this is the standard deviation of the smoothing
          kernel.  (Note this differs from the reference books cited
          below, and from S-PLUS.)

          'bw' can also be a character string giving a rule to choose
          the bandwidth.  See 'bw.nrd'.
          The default, '"nrd0"', has remained the default for
          historical and compatibility reasons, rather than as a
          general recommendation, where e.g., '"SJ"' would rather fit,
          see also V&R (2002).

          The specified (or computed) value of 'bw' is multiplied by
          'adjust'.

  adjust: the bandwidth used is actually 'adjust*bw'.  This makes it
          easy to specify values like 'half the default' bandwidth.

kernel, window: a character string giving the smoothing kernel to be
          used. This must be one of '"gaussian"', '"rectangular"',
          '"triangular"', '"epanechnikov"', '"biweight"', '"cosine"' or
          '"optcosine"', with default '"gaussian"', and may be
          abbreviated to a unique prefix (single letter).

          '"cosine"' is smoother than '"optcosine"', which is the usual
          'cosine' kernel in the literature and almost MSE-efficient.
          However, '"cosine"' is the version used by S.

      n: the number of equally spaced points at which the density is
         to be estimated.  When 'n > 512', it is rounded up to a power
         of 2 during the calculations (as 'fft' is used) and the final
         result is interpolated by 'approx'.  So it almost always
         makes sense to specify 'n' as a power of two.
```

density() is the workhorse function that powers densityplot()

important arguments highlighted

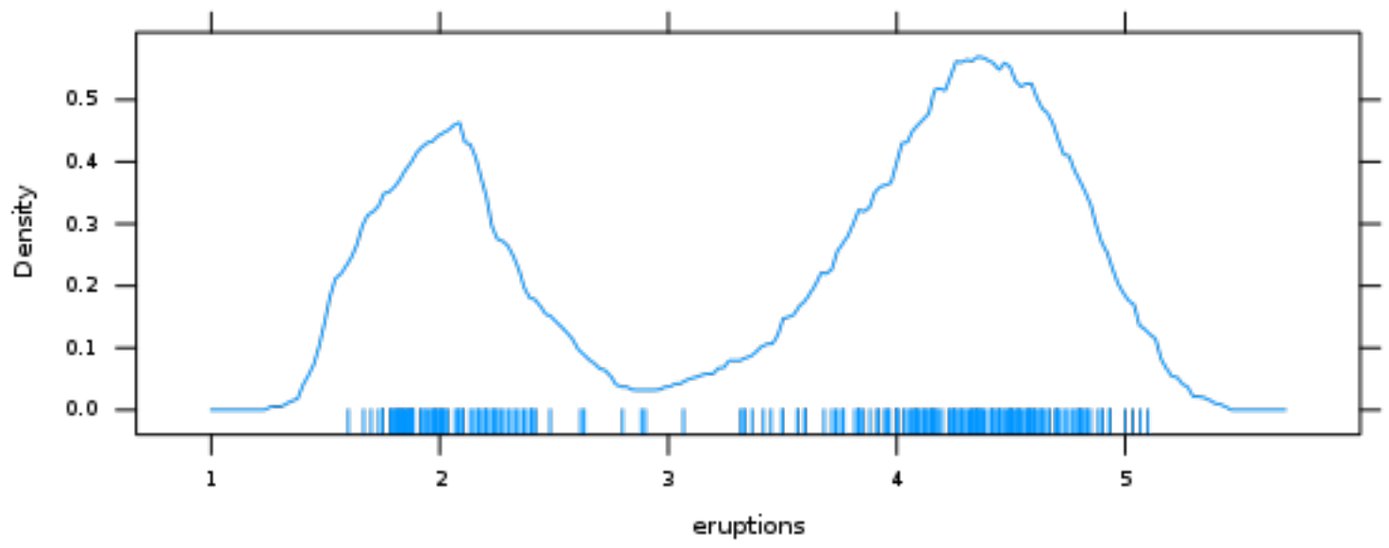
don’t confuse yourself: you probably think ‘n’ means the size of your sample but density() has different ideas!

Figure 3.1

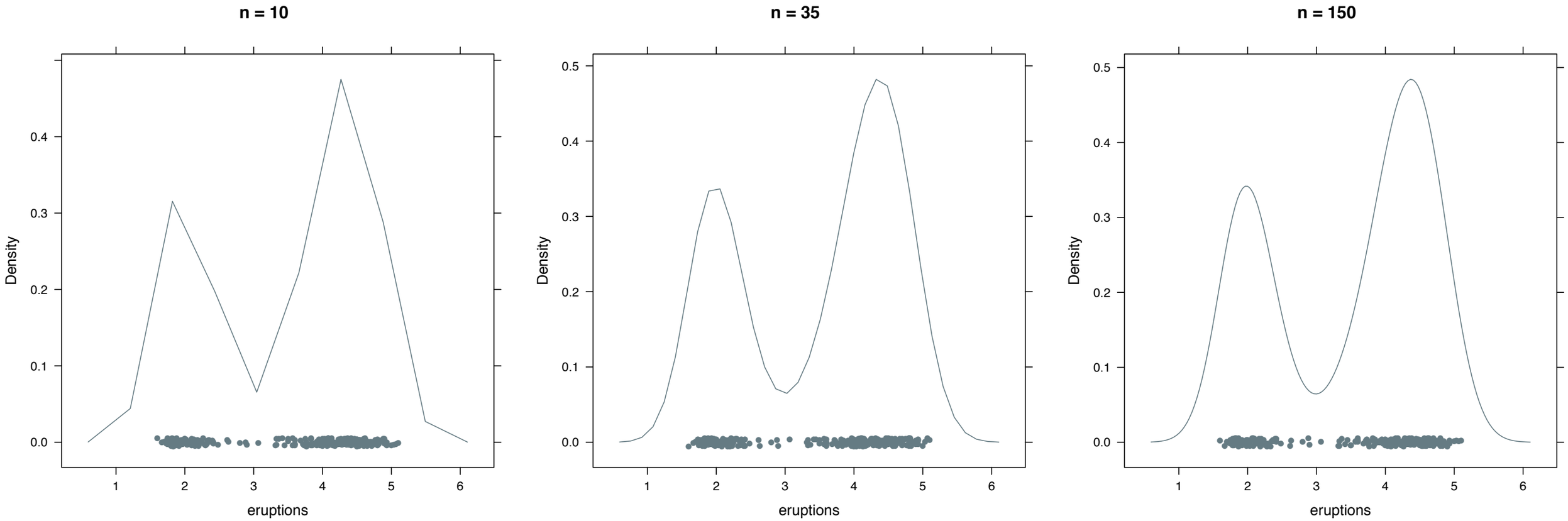



```
densityplot(~ eruptions, data = faithful)
densityplot(~ eruptions, data = faithful,
            kernel = "rect", bw = 0.2,
            plot.points = "rug", n = 200)
```

Figure 3.2



`densityplot` allows the user to specify the arguments to `density`, e.g. the kernel, bandwidth




```
densityplot(~ eruptions, data = faithful,
            n = 35,
            main = "n = 35")
```

```
density                  package:stats                  R Documentation

Kernel Density Estimation

<snip, snip>

      n: the number of equally spaced points at which the density is
         to be estimated.  When 'n > 512', it is rounded up to a power
         of 2 during the calculations (as 'fft' is used) and the final
         result is interpolated by 'approx'.  So it almost always
         makes sense to specify 'n' as a power of two.
```

Practical usage tip: if the line drawing of the KDE has hard edges, try specifying a higher value of n.

Practical usage tip: if the actual KDE is wigglier than you'd like, try changing the bandwidth.

Preferred method is via "adjust"

See the tutorial for live examples.

Other recommended sources:

Härdle, W. (1990) Smoothing Techniques With Implementation in S, Springer-Verlag, 1990. Sadly, not available via SpringerLink.

Silverman, B.W. (1986) Density Estimation for Statistics and Data Analysis, Chapman & Hall, 1986. Sadly, not available via STATSnetBASE.