The ggplot2 Implementation of the Grammar of Graphics

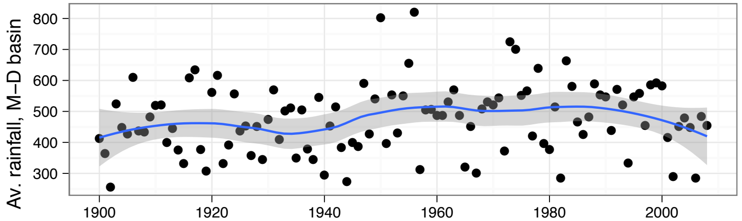
J H Maindonald

Centre for Mathematics and Its Applications
Australian National University.

Each type of plot is a different "geometry"

- As specific examples, there is a "geometry" (geom) for
  - scatterplot: geom_point()
  - histogram: geom_hist()
  - density plot: geom_density()
  - 2-dimensional density estimate: geom_density2d()
- Different geometries can be overlaid.

```
library(DAAG); library(ggplot2)
## Default loess smooth, with SE bands
quickplot(Year, mdbRain, data=bomregions,
          geom=c("point","smooth"),
          span=0.5, se=TRUE,
          xlab="",
          ylab="Av. rainfall, M-D basin")
```



Other arguments include: size, color, shape
But specify color=I("red"), not color="red"

Step by Step Graphical Construction

The function quickplot() generates a sequence of calls that create the different plot components:

```
ggplot(bomregions,
        aes(x=Year, y=mdbRain)) +
  geom_point() +
  geom_smooth(span=0.5, se=TRUE) +
  scale_x_continuous("") +
  scale_y_continuous(
          "Av. rainfall, M-D basin")
## NB: x and y are "aesthetics"
```

NB: Addition ("+") of graphical components

quickplot() plus ...

Another possibility: Use quickplot() to create a ggplot object, then add to it.

```
quickplot(Year, mdbRain, data=bomregions,
          geom="point",
          xlab="",
          ylab="Av. rainfall, M-D basin") +
  geom_smooth(span=0.5, se=TRUE)
```

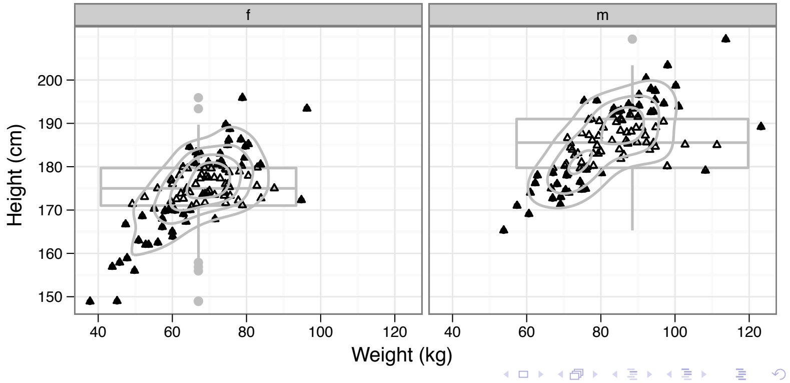
Quantile Curve Estimates

```
library(splines)
library(quantreg)
quickplot(Year, mdbRain, data=bomregions,
          geom=c("point", "quantile"),
          formula = y ~ ns(x,5),
          quantiles=c(0.2,0.5,0.8) )
```

Overlaying Madness? (Code gives a first draft version)

```
## Scatterplot + boxplot + density contours
qplot(wt, ht, data=ais,
      geom=c("boxplot", "point", "density2d"),
      facets = . ~ sex)
```

Code used for the graph as shown is on the next slide.



Overlaying with More Finesse (Graph is on Previous Slide)

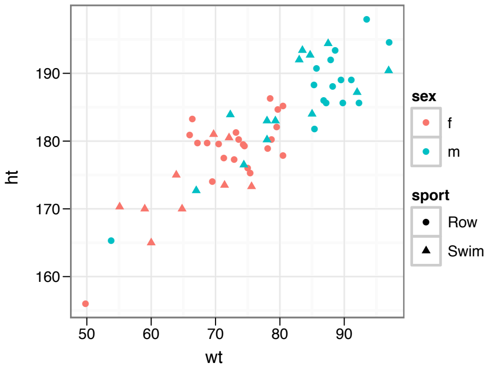
Set axis labels, show boxplot outline in gray, show contour lines in gray (default is blue), etc.

```
quickplot(wt, ht, xlab="Weight (kg)",
          ylab="Height (cm)", data=ais,
          facets = . ~ sex) +
  geom_boxplot(outlier.size=1.75,
               outlier.colour="gray",
               color="gray") +
  geom_point(shape=2, size=1) +
  geom_density2d(color="gray")
```

NB: `facets = row.var ~ col.var`

`row.var` indexes rows (of panels), `col.var` indexes columns; if needed, use "." as placeholder.

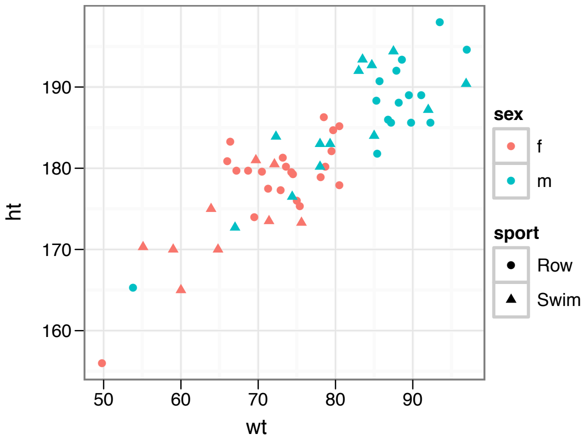
Distinguish sexes by color, sports by shape



```
quickplot(wt, ht,
  data=aisBS,
  geom="point",
  color=sex,
  shape=sport)
```

Distinguish sexes by color, sports by shape

```
quickplot(wt, ht, data=aisBS,  geom="point",
  color=sex, shape=sport)
```



Here., color and shape are aesthetics.

Distinguish between settings and aesthetic mappings

	Use of quickplot()	Plots based on ggplot()
Settings	size=I(3) or cex=3	size=3
Aesthetic mappings	size=sport	aes(size=sport)

The argument size=3 to quickplot() changes the point size, but adds an extraneous key. The same happens if the argument mapping=aes(size=3) is supplied to ggplot() or to geom_point() or to another such function.

Synonyms: cex for size, type for geom, color for colour (sic!) [at least in calls to quickplot()]

Control of Defaults

```
## Set theme_bw() defaults
## (black gridlines; white background)
## Also set base text size to 8pt
old <- theme_set(theme_bw(base_size=8))
## Reduce default size for geom_point(),
## from 2 to 1.5
update_geom_defaults("point", aes(size=1.5))
theme_set(old)     # Restore earlier settings
```

The default theme is theme_gray(), so called because its white gridlines overlay a very light gray background.