

# R Graphics

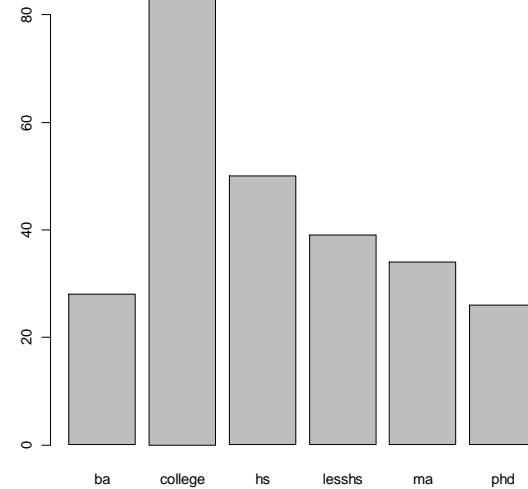
21 Sept 2011

# Three popular graphics systems

- Graphics package (aka “default”)
- Lattice package (aka “trellis” graphics)
- Ggplot2 package (“grammar of graphics”)

# Plot()

- The type of graph depends on the type of data

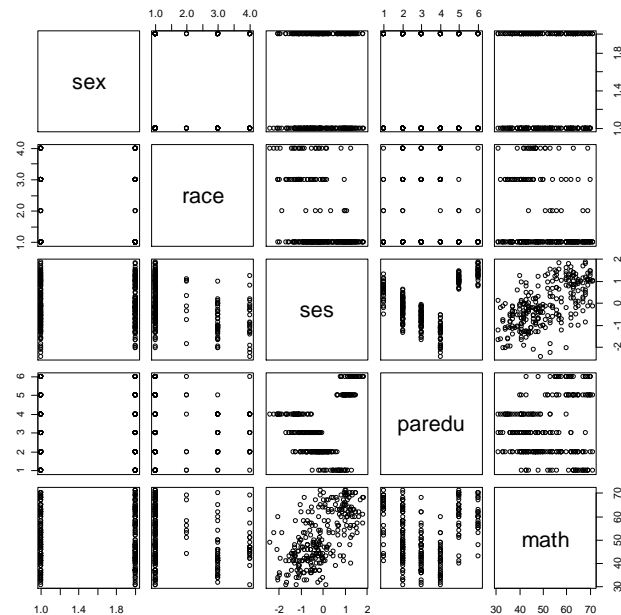


```
# a factor
```

```
plot(nels88$paredu)
```

```
# a data frame
```

```
plot(nels88)
```



# Plot() methods

methods (plot)

```
[1] plot.acf*           plot.data.frame*   plot.decomposed.ts*
[4] plot.default       plot.dendrogram*  plot.density
[7] plot.ecdf          plot.factor*       plot.formula*
[10] plot.hclust*       plot.histogram*    plot.HoltWinters*
[13] plot.isoreg*       plot.lm             plot.medpolish*
[16] plot.mlm           plot.ppr*          plot.prcomp*
[19] plot.princomp*     plot.profile.nls*  plot.spec
[22] plot.spec.coherency plot.spec.phase     plot.stepfun
[25] plot.stl*          plot.table*        plot.ts
[28] plot.tskernel*     plot.TukeyHSD
```

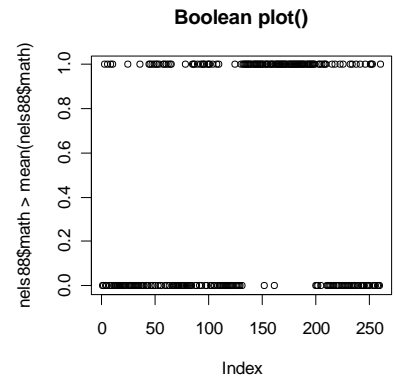
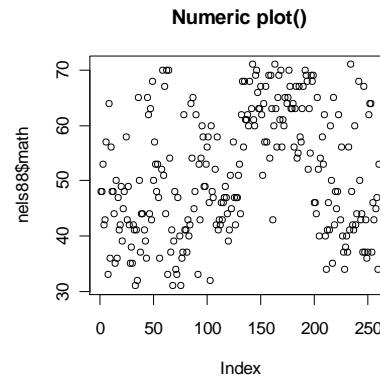
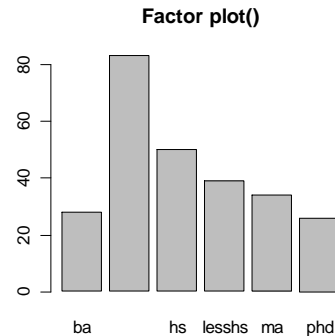
# Univariate plots

```
# for a factor, plot() returns a bar  
chart  
plot(nels88$paredu, main="Factor plot()")
```

```
# ERROR, character data first must be  
converted to a factor  
plot(as.character(nels88$paredu),  
main="Character")
```

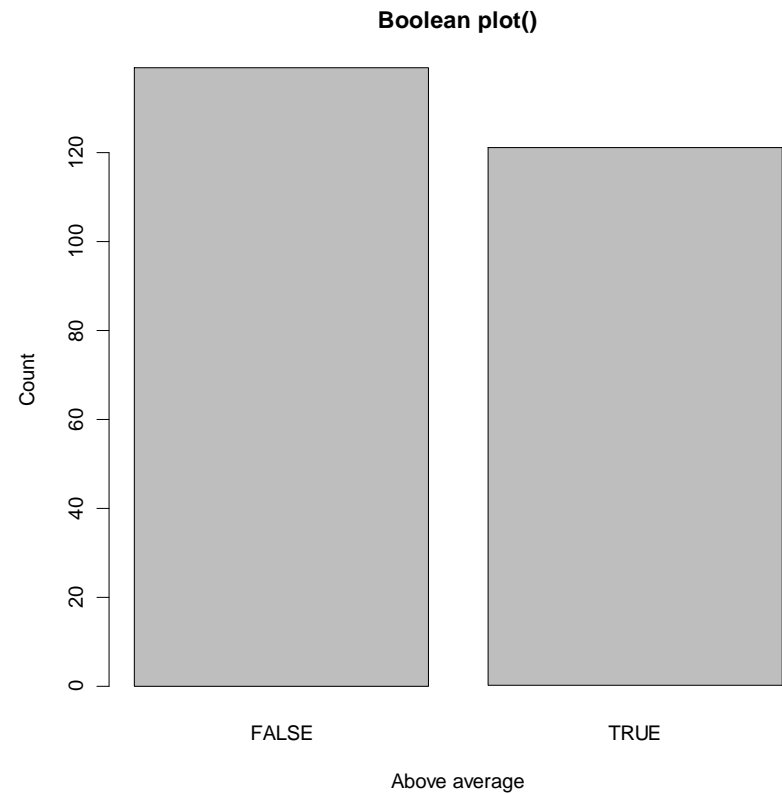
```
# for a numeric vector, a "scatterplot"  
plot(nels88$math, main="Numeric plot()")
```

```
# boolean values coerced to numeric  
plot(nels88$math > mean(nels88$math),  
main="Boolean plot()")
```



# Using coercion to alter type

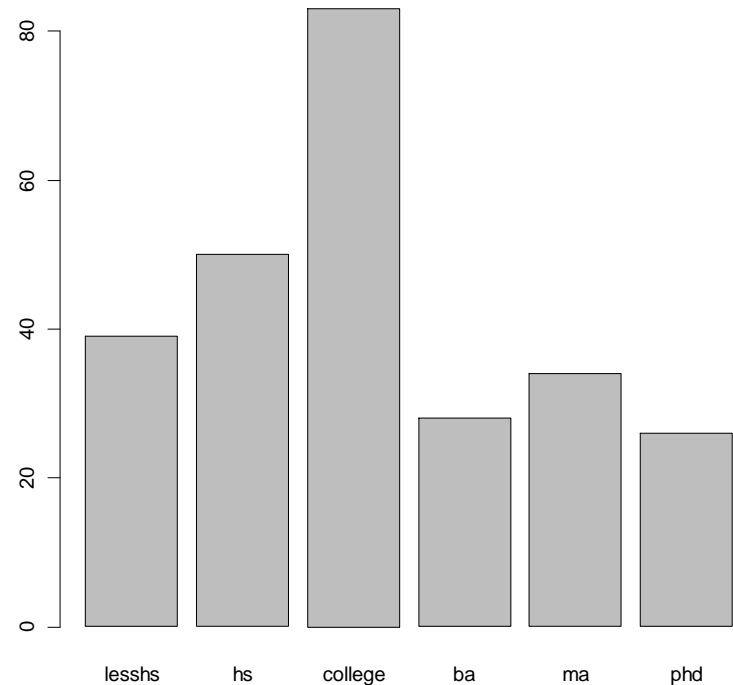
```
plot(factor(nels88$math > mean(
  nels88$math)), main="Boolean
plot()", ylab="Count",
xlab="Above average")
```



# Reordering Factor Levels

```
parentedu <-  
  factor(nels88$parentedu,  
        levels=c("lesshs", "hs",  
                 "college", "ba", "ma",  
                 "phd"),  
        ordered=TRUE)
```

```
plot(parentedu)
```



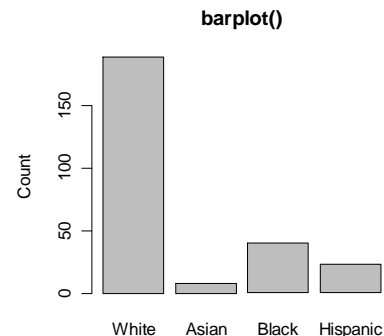
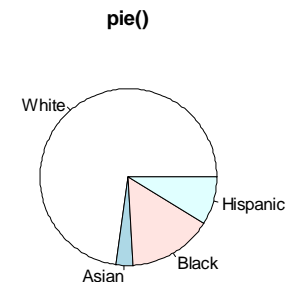
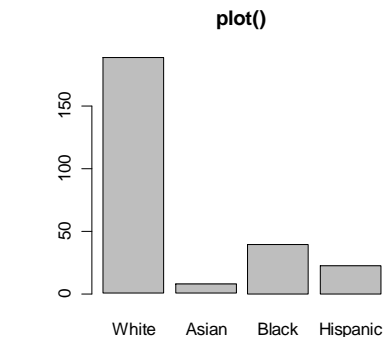
# Other Univariate Plots for Factors

```
plot(nels88$race,  
     main="plot()")
```

```
# note that pie() and barplot()  
# start from tabular data
```

```
pie(table(nels88$race),  
     main="pie()")
```

```
barplot(table(nels88$race),  
         main="barplot()",  
         ylab="Count")
```





# Histograms

```
hist(nels88$ses,  
     probability=TRUE)
```

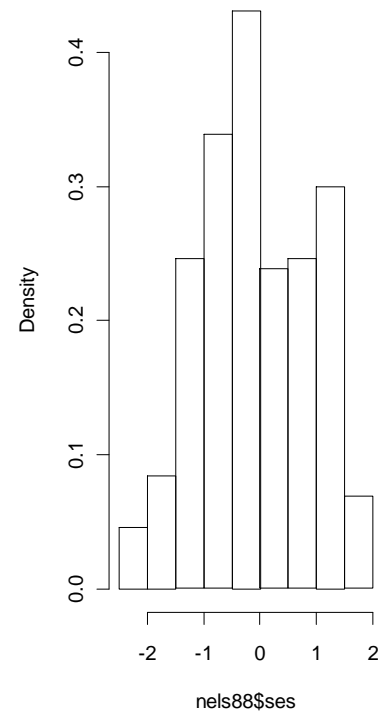
```
a<-hist(nels88$ses,  
        probability=TRUE)
```

```
par(new=TRUE)
```

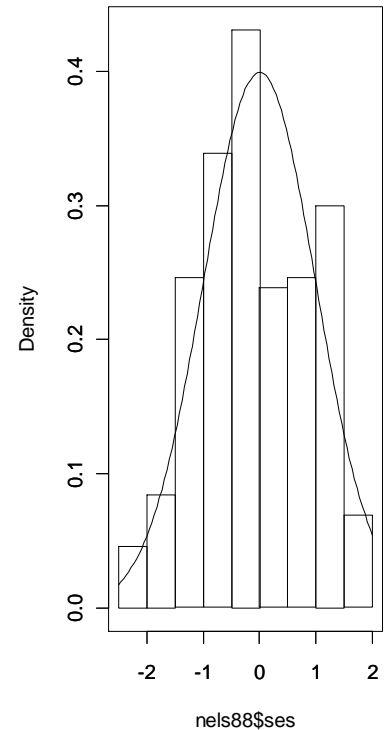
```
curve(dnorm,  
      ylim=c(0,max(a$density)),  
      xlab="", ylab="")
```

```
par(new=FALSE)
```

Histogram of nels88\$ses

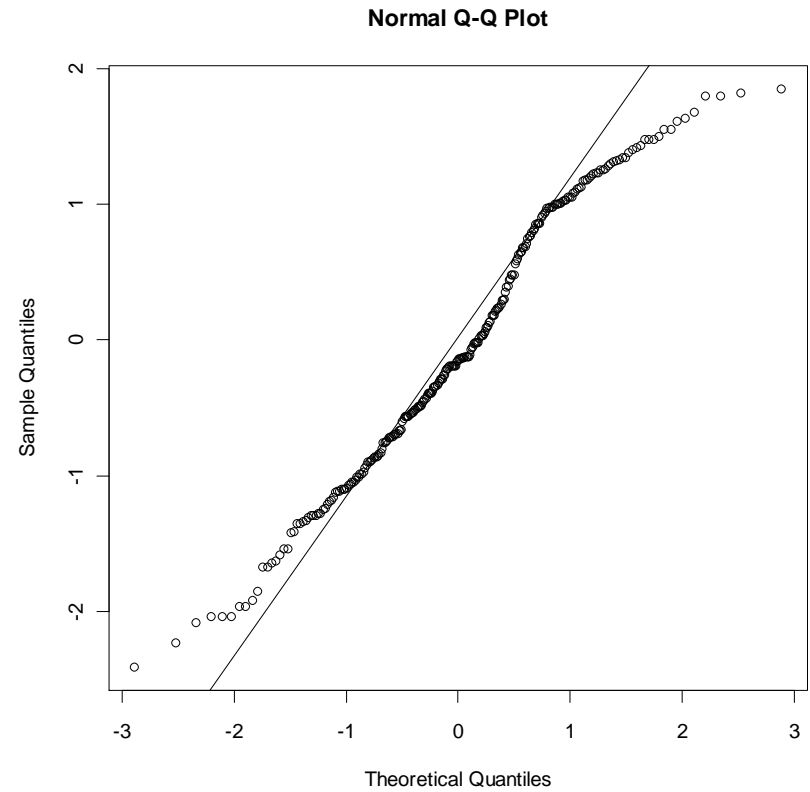


Histogram of nels88\$ses



# Q-Q Plots

```
qqnorm(nels88$ses)  
qqline(nels88$ses)
```



# Bivariate Graphs

- Numeric ~ Factor
- Factor ~ Numeric
- Numeric ~ Numeric
- Factor ~ Factor

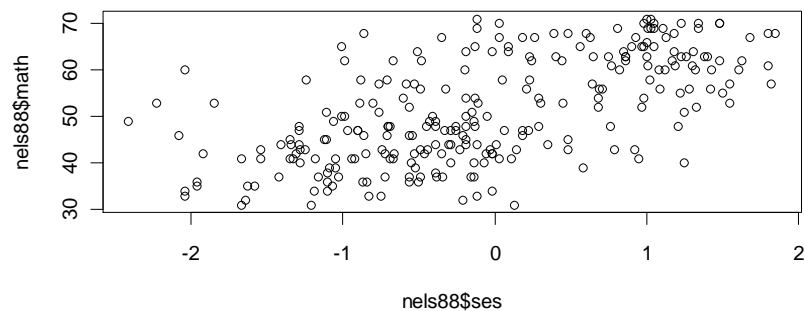
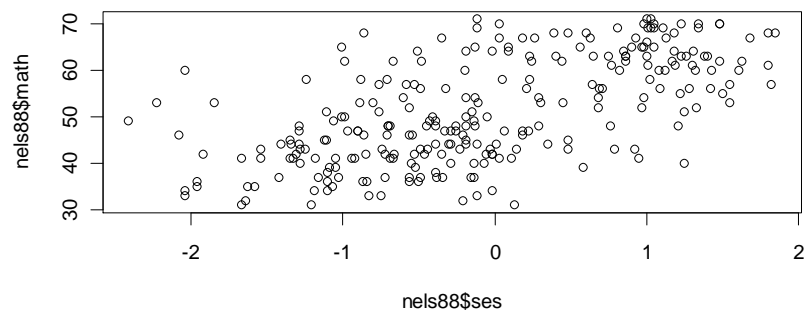
# Scatterplots (Numeric ~ Numeric)

```
# two parameters
```

```
plot(nels88$ses, nels88$math)
```

```
# formula
```

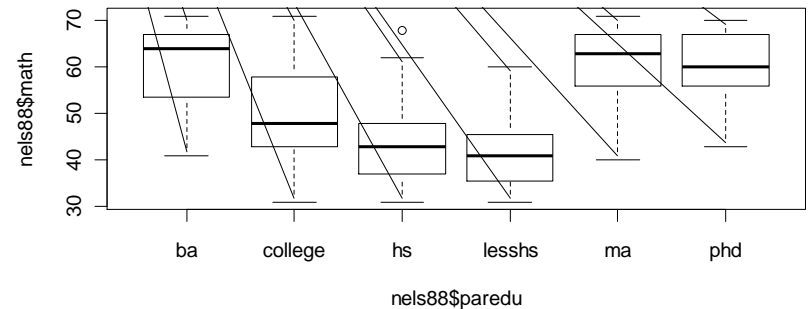
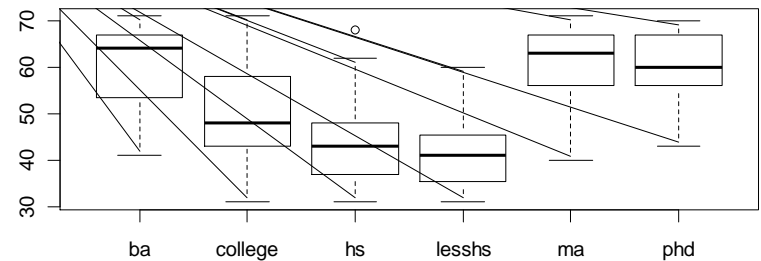
```
plot(nels88$math ~ nels88$ses)
```



# Boxplots (Numeric ~ Factor)

```
# boxplot, two parameters  
plot(nels88$paredu,  
      nels88$math)
```

```
# boxplot, a "formula"  
parameter  
plot(nels88$math ~  
      nels88$paredu)
```



# Factor ~ Numeric

```
# As x=,y= parameters, both  
  vectors are treated as  
  numeric
```

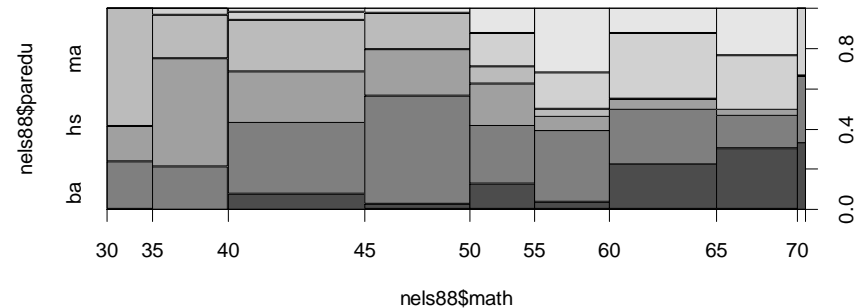
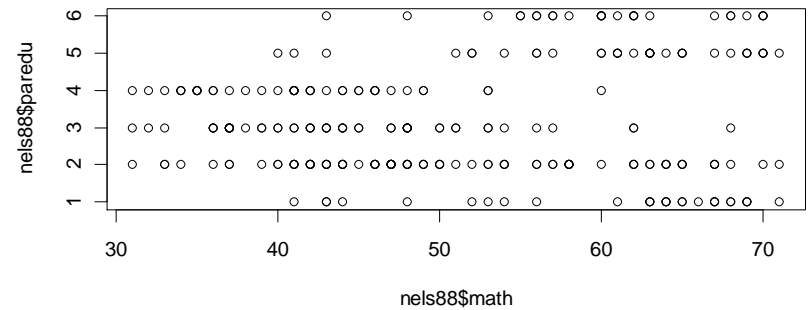
```
# scatterplot
```

```
plot(nels88$math,  
     nels88$paredu)
```

```
# As a formula, both vectors  
  are treated as factors!!!
```

```
# "mosaic", aka "spineplot"
```

```
plot(nels88$paredu ~  
     nels88$math)
```



# Mosaic (Factor ~ Factor)

```
plot(nels88$race,  
     nels88$paredu)
```

```
plot(nels88$paredu ~  
     nels88$race)
```

