Stat Overview

Paul E. Johnson[1] [2]

[1]Department of Political Science

[2]Center for Research Methods and Data Analysis, University of Kansas

2019

Outline

Outline

Check your packages

- The base install of R (R Core Team, 2017) loads the stats module. See?

```
sessionInfo()
```

```
R version 3.6.0 (2019-04-26)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 19.04

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/atlas/libblas.so.3.10.3
LAPACK: /usr/lib/x86_64-linux-gnu/atlas/liblapack.so.3.10.3

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
     LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8
     LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8
    LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```

Check your packages ...

```
loaded via a namespace (and not attached):
[1] compiler_3.6.0 tools_3.6.0
```

- The presence of “stats” means that functions like these are available.
  - mean
  - lm
- See about the stats package

```
help(package="stats")
```

For anything else, run library

- Specialized stat procedures are generally provided in separate packages
- Possibly most famous stat-oriented packages are associated with stats textbooks

MASS Venables, William and Ripley, Brian, *Modern Applied Statistics with S*

car Fox, John, *Applied Regression Analysis and Generalized Linear Models* and *Companion to Applied Regression*

nlme Pinheiro, Jose and Bates, *Douglas Mixed-Effects Models in S and S-PLUS.*

Outline

variable types: numeric versus factors

- Numbers can be logged, squared, added, etc
- Factors cannot be logged, squared

religion	
label	R's internal integer for record keeping
Catholic	1
Protestant	2
Jewish	3
Muslim	4

- In R, categorical variables are called factors (see functions `factor()`, `ordered()`, `levels()`)
- Many functions will "promote" character variables to factors automatically

R functions adapt to data

- Most R statistical procedures try to "do the right thing" if we use a factor variable
- Regression Example. *As we all know* regression coefficients are only defined for numeric predictors. However, factor predictors can be included).
    - Suppose $sex \in \{Male, Female\}$.
- Including sex as a predictor in a regression will cause R to
    - Notice sex is not numeric. It is an unordered factor.
    - R will create a "dummy variable" named `sexFemale` (Male=0, Female=1). (Also known as an "indicator variable", "binary variable", "dichotomous variable")

R functions adapt to data ...

- If the predictor were $rel \in \{Cath, Prot, Jewi, Musl, Hind\}$, a regression routine would typically create 4 "dummies", relProt, relJewi, relMusl, relHind, the last 4 columns here.

religion	numeric score	relCath	relProt	relJewi	relMusl	relHind
Cath	1	1	0	0	0	0
Prot	2	0	1	0	0	0
Jewi	3	0	0	1	0	0
Musl	4	0	0	0	1	0
Hind	5	0	0	0	0	1

  - However, user can adjust the regression formula to request estimation of both sexFemale and sexMale or all 5 levels of religion.

R functions adapt to data

- Example 2. The plot function responds differently to inputs

```
plot(y ~ x)
```

will make

- a scatterplot if y and x are numeric
- a box plot if y is numeric and x is a factor
- a bar plot if both are factors

Output

You only get what you know how to ask for (Paul Johnson, 2016)

- Most procedures return minimal output. This is silent, unless there is an error message

```
m1 <- lm(mydv ~ x1 + x2 + x3 + x4, data =
    wonderful)
```

- m1 is an “object”, waiting to be quizzed and investigated.

For Functions within R's recommended packages

- we can be fairly sure that functions like `print()`, `summary()`, `plot()`, `coef()` will work as expected
- Almost always, `summary()` will create a new object which can be further explored
- If you download additional packages, *all bets are off*.

Cultural Norms versus Coding Requirements

- R is an open, flexible culture
  - opinion leaders
  - mutual expectations
  - shorthand symbolic references
- R allows creation of new symbols and functions
- Until now, the most respected voices have been authors coming out of the ATT S tradition
  - They are focused on re-usability of function names across contexts.
  - `summary()` is supposed to work on any kind of object, and change understandably across contexts
  - `anova()` is supposed to be a general purpose way to compare fitted statistical objects
- These are recommended practices, but not universally followed.

Open Source/Free Software

If it breaks, you get to keep all the pieces (Anon)

- R is an engine congenial to the addition of contributed packages
- Perhaps R is the "lingua franca of statistics" ("Data Analysts Captivated by R's Power", *New York Times*, January 6, 2009), but it is not a corporation.
- r-help email list, Stackexchange forums

Fish follow glittering objects

- Some of the most enticing R packages are also the most frustrating
- Fancy model appears, our dissertation advisor says “use that”, and it doesn’t work.
- Download the package source code, to find out how they did it
  - Bad news: The code a complicated tangle we have no hope of learning from it.

Outline

Some example data I made a long time ago

I have a file in "data" called "gss-subset2.Rda". If you don't have it, it can be downloaded:

http://pj.freefaculty.org/guides/stat/DataSets/GSS/gss-subset2.Rda

- Lets check the workspace before loading

```
(ls.old <- ls())
```

```
[1] "opts.orig" "par.orig"   "pjmar"      "tdir"
```

- This is an RData structure, it can drop any number of objects into my workspace

- Check workspace after loading

```
(ls.new <- ls())
```

Some example data I made a long time ago ...

```
[1] "dat"        "ls.old"     "opts.orig" "par.orig"   "pjmar"      "tdir"
```

```
setdiff(ls.new, ls.old)
```

```
[1] "dat"    "ls.old"
```

- `setdiff()` is a handy function, it goes along with the R functions `union()` and `intersect()`

Outline

summary()

```
summary(dat)
```

```
      hrs1                      wrkslf                   marital
 Min.    : 1.00   SELF-EMPLOYED: 508   MARRIED      :2170
 1st Qu.:38.00   SOMEONE ELSE :3799   WIDOWED      : 366
 Median :40.00   NA's          : 203   DIVORCED     : 732
 Mean   :42.08                         SEPARATED    : 156
 3rd Qu.:50.00                         NEVER MARRIED:1080
 Max.   :89.00                         NA's         :   6
 NA's   :1771
     sex            race           educ
 MALE  :2003   WHITE:3284   Min.   : 0.00
 FEMALE:2507   BLACK: 634   1st Qu.:12.00
               OTHER: 592   Median :13.00
                            Mean   :13.29
                            3rd Qu.:16.00
                            Max.   :20.00
                            NA's   :11
               partyid          age
 INDEPENDENT        :997   Min.   :18.00
 NOT STR DEMOCRAT   :736   1st Qu.:34.00
 STRONG DEMOCRAT    :700   Median :46.00
 NOT STR REPUBLICAN:637   Mean   :47.14
 IND,NEAR DEM       :527   3rd Qu.:59.00
```

summary() ...

```
(Other)          :887   Max.   :89.00
NA's             : 26   NA's   :18
                      polint                            news
VERY INTERESTED       : 278   EVERYDAY             : 945
FAIRLY INTERESTED     : 361   FEW TIMES A WEEK : 611
SOMEWHAT INTERESTED   : 427   ONCE A WEEK          : 418
NOT VERY INTERESTED   : 234   LESS THAN ONCE WK: 404
NOT AT ALL INTERESTED: 216    NEVER                : 350
NA's                  :2994   NA's                 :1782

        newsfrom                      income06
TV           : 916   REFUSED             : 442
Newspapers   : 446   $40000 TO 49999 : 394
The Internet: 253    $60000 TO 74999 : 360
Radio        : 131   $50000 TO 59999 : 332
Family       :  33   $75000 TO $89999: 284
(Other)      :  80   (Other)             :2503
NA's         :2651   NA's                : 195
   realinc           gunlaw          owngun
Min.   :   275   FAVOR :1568    YES     : 664
1st Qu.: 11702   OPPOSE: 395    NO      :1307
Median : 24782   NA's  :2547    REFUSED:   30
Mean   : 32694                  NA's    :2509
3rd Qu.: 45433
Max.   :139981
NA's   :637
```

summary() ...

```
                  vote00                               pres00
VOTED                :1826   GORE                   :  813
DID NOT VOTE         : 715   BUSH                   :  903
INELIGIBLE           : 389   NADER                  :   26
REFUSED TO ANSWER:     0   OTHER (SPECIFY):   19
NA's                 :1580   DIDNT VOTE             :    9
                             NA's                   :2740

                  vote04                               pres04
VOTED                :3037   KERRY                  :1434
DID NOT VOTE         :1089   BUSH                   :1446
INELIGIBLE           : 335   NADER                  :   47
REFUSED TO ANSWER:     0   OTHER (SPECIFY):    0
NA's                 :   49   DIDNT VOTE             :   17
                             NA's                   :1566

                 sexfreq                                   polviews
NOT AT ALL           : 595   MODERATE                   :1683
2-3 PER WEEK         : 430   CONSERVATIVE               : 685
2-3 TIMES A MONTH: 361   SLGHTLY CONSERVATIVE: 618
WEEKLY               : 343   LIBERAL                    : 524
ONCE A MONTH         : 265   SLIGHTLY LIBERAL           : 517
(Other)              : 339   (Other)                    : 306
NA's                 :2177   NA's                       : 177
     sei                              mhgvthlt
Min.   :17.10   Definitely should       : 488
```

summary() ...

```
 1st Qu.:32.80   Probably should        : 617
 Median :42.20   Probably shouldnt be   : 200
 Mean   :49.41   Definitely shouldnt be:  77
 3rd Qu.:64.10   NA's                   :3128
 Max.   :97.20
 NA's   :268
                 mhgvthme                     lessreg
 Definitely should      : 311   STRONGLY IN FAVOR: 251
 Probably should        : 701   IN FAVOR         : 518
 Probably shouldnt be   : 278   NEITHER          : 406
 Definitely shouldnt be:  97   AGAINST          : 236
 NA's                   :3123   STRONGLY AGAINST :  69
                                NA's             :3030


    numwomen            nummen
 Min.   :  0.000   Min.   :  0.000
 1st Qu.:  0.000   1st Qu.:  0.000
 Median :  0.000   Median :  1.000
 Mean   :  9.479   Mean   :  9.286
 3rd Qu.:  4.000   3rd Qu.:  3.000
 Max.   :997.000   Max.   :997.000
 NA's   :2204      NA's   :2214
                 sexsex5                   evstray
 EXCLUSIVELY MALE     :1059   YES            : 350
 BOTH MALE AND FEMALE:  40   NO             :1414
 EXCLUSIVELY FEMALE  : 893   NEVER MARRIED: 623
```

summary() ...

```
NA's                  :2518   NA's           :2123
```

rockchalk::summarize()

- summary() has been that way since, well, forever
  - output is text, not an object with numbers we can re-use
  - no diversity values (variance, skewness, kurtosis)
  - I prefer to separate the numeric and factor variables, and to alphabetize the output
  - entropy is a diversity measure for discrete sets.
  - normedEntropy range
    - 0 (all scores observed in one category)
    - 1 (all outcomes equally likely)

Mean, Variance, etc

- There are functions in stats package for basic descriptive statistics

purpose	R function
sample average	mean(x)
sample variance	var(x)
sample standard deviation	sd(x)
range	range(x)
minimum	min(x)
maximum	max(x)
quantiles (range values)	quantile(x)

But there's a "gotcha" I need to warn you about

- Observe

```
mean(dat$age)
```

```
[1] NA
```

The age variable is average is missing in GSS. WTF?

- The range does not exist either? And no maximum?

```
range(dat$age)
```

```
[1] NA NA
```

```
max(dat$age)
```

```
[1] NA
```

Missing values

- The symbol **NA** is used for "missing data" in R vectors and data frames
- At least quantile throws us a warning

```
quantile(dat$age)
```

```
Error in quantile.default(dat$age) :
missing values and NaN's not allowed if 'na.rm' is FALSE
```

- passive-aggressive approach to missings in R

```
mean(dat$age, na.rm = TRUE)
```

```
[1] 47.14159
```

```
range(dat$age, na.rm = TRUE)
```

```
[1] 18 89
```

Missing values ...

```
quantile(dat$age, na.rm = TRUE)
```

```
 0%  25%  50%  75% 100%
 18   34   46   59   89
```

- Some functions will automatically ignore missings (plot(), lm()). Simple stats will not. Grrrr!

Outline

Sometimes, a Cross Tabulation is the best you can do

The Iron Laws of Crosstabs. 3 rules for a happy life.

1. IV on top, DV on left
2. Convert to percentages (or proportions) on the columns
3. Compare the across, find if columns are distributed differently

The FX Network is	Column Percentages Respondent Sex	
	male	female
really infantile	25%	60%
OK	50%	18%
really great	25%	22%
N	343	288

Here's a Table I Typed By hand

	Does Respondent Own a Gun?		
Stance on Gun Registration	Yes	No	Refused To Say
Favor	70.7%	84.9	62.9
Oppose	29.3	15.1	38.0
Number of Cases	656	1128	27

Found 2 typographical errors when reviewing against real numbers below.

R base tools for tables can be made to work

```
t1 <- table(dat$gunlaw, dat$owngun)
t1
```

```
         YES   NO REFUSED
FAVOR    464 1085      17
OPPOSE   192  193      10
```

```
prop.table(t1, 2)
```

```
             YES        NO   REFUSED
FAVOR  0.7073171 0.8489828 0.6296296
OPPOSE 0.2926829 0.1510172 0.3703704
```

```
addmargins(t1)
```

```
         YES   NO REFUSED  Sum
FAVOR    464 1085      17 1566
OPPOSE   192  193      10  395
Sum      656 1278      27 1961
```

package gmodels introduced SPSS style CrossTable function

```
library(gmodels)
CrossTable(dat$gunlaw, dat$owngun)
```

```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  1961


             | dat$owngun
  dat$gunlaw |       YES |        NO |   REFUSED | Row Total |
-------------|-----------|-----------|-----------|-----------|
       FAVOR |       464 |      1085 |        17 |      1566 |
             |     6.841 |     4.067 |     0.965 |           |
             |     0.296 |     0.693 |     0.011 |     0.799 |
```

package gmodels introduced SPSS style CrossTable function ...

```
             |      0.707 |      0.849 |      0.630 |            |
             |      0.237 |      0.553 |      0.009 |            |
-------------|------------|------------|------------|------------|
      OPPOSE |        192 |        193 |         10 |        395 |
             |     27.121 |     16.123 |      3.826 |            |
             |      0.486 |      0.489 |      0.025 |      0.201 |
             |      0.293 |      0.151 |      0.370 |            |
             |      0.098 |      0.098 |      0.005 |            |
-------------|------------|------------|------------|------------|
Column Total |        656 |       1278 |         27 |       1961 |
             |      0.335 |      0.652 |      0.014 |            |
-------------|------------|------------|------------|------------|
```

rockchalk has pctable

- While CrossTable was a welcome invention, it did not boil down to the sort of table that I required of my students.
- We explored alternatives, some of which are very nice (packages memisc, vcd, and descr).
- But, now, feast your eyes on this:

```
library(rockchalk)
pctable(gunlaw ~ owngun, data = dat, rvlab =
    "Stance on Gun Registration", cvlab = "Does
    Respondent Own a Gun?")
```

rockchalk has pctable ...

```
Count (column %)
                        Does Respondent Own a Gun?
Stance on Gun Registration YES        NO          REFUSED
                     FAVOR  464(70.7%) 1085(84.9%) 17(63%)
                     OPPOSE 192(29.3%) 193(15.1%)  10(37%)
                     Sum    656        1278        27
                        Does Respondent Own a Gun?
Stance on Gun Registration Sum
                     FAVOR  1566
                     OPPOSE 395
                     Sum    1961
```

- which can be wrestled into a nice looking table, either in html or LaTeX. Here's the LaTeX

tabsum	YES	NO	REFUSED	Sum
FAVOR	464(70.7%)	1085(84.9%)	17(63%)	1566
OPPOSE	192(29.3%)	193(15.1%)	10(37%)	395
Sum	656	1278	27	1961

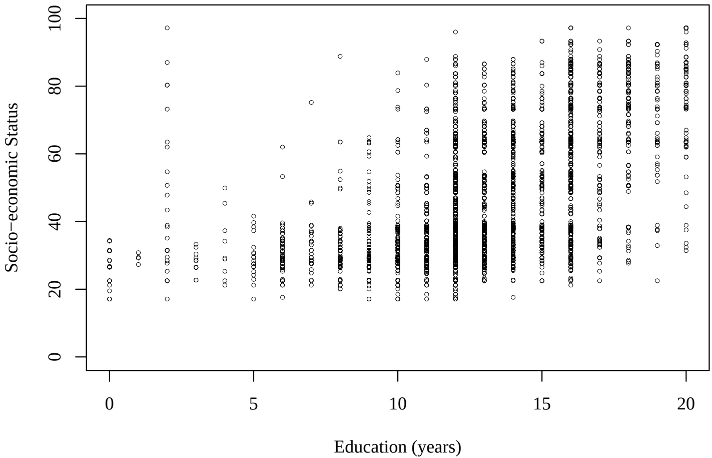
Outline

Scatterplot: 2 numeric variables

- Socio-economic status and education

```
plot(sei ~ educ, data = dat, cex = 0.5, lwd =
   0.2, main = "",
 xlab =  "Education (years)", ylab =
    "Socio-economic Status", ylim = c(0, 100))
```

Scatterplot: 2 numeric variables ...

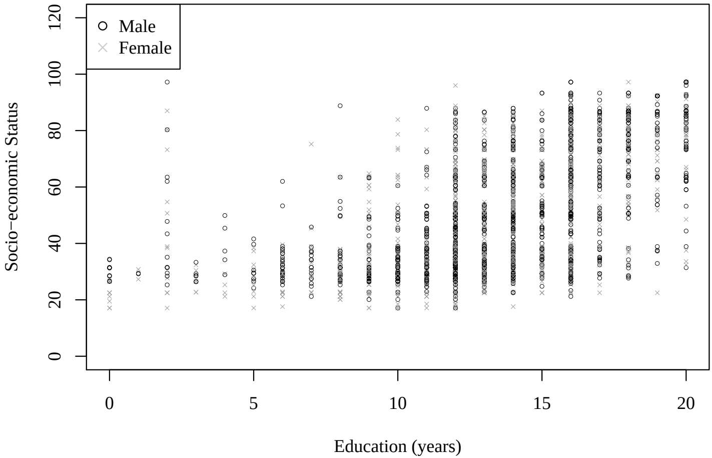


- Color Coded Men and Women

Scatterplot: 2 numeric variables ...

```
plot(sei ~ educ, data = dat,  main = "", xlab =
    "Education (years)",
      ylab = "Socio-economic Status", ylim =
          c(0,120), type = "n")
sexcolor <- ifelse(dat$sex == "MALE", "black",
    "gray60")
sexpch <- ifelse(dat$sex == "MALE", 1, 4)
points(sei ~ educ, data = dat, cex = 0.5, lwd =
    0.2,
      col = sexcolor, pch = sexpch)
legend("topleft", legend = c("Male","Female"),
      col = c("black","gray80"), pch = c(1,4))
```

Scatterplot: 2 numeric variables ...

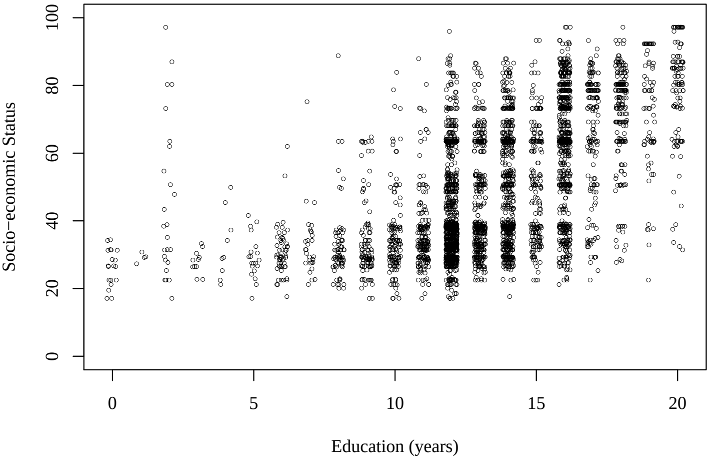


"Piled up observations" Problem

- I made the symbols light to give a hint: There are lots of repeated scores.
- The most common quick fix for this is to "jitter" the observations so they don't overlap quite so much.

```
plot(jitter(sei) ~ jitter(educ), data = dat, cex
    = 0.5, lwd = 0.2, main = "", xlab =
    "Education (years)", ylab = "Socio-economic
    Status", ylim = c(0, 100))
```

"Piled up observations" Problem ...



Lately, People are looking at smarter plot types

- CRAN package "hexbin"
- You should install "hexbin", then run

```
library(hexbin)
help(package = "hexbin")
example(hexbin)
vignette("hexabon_binning")
```

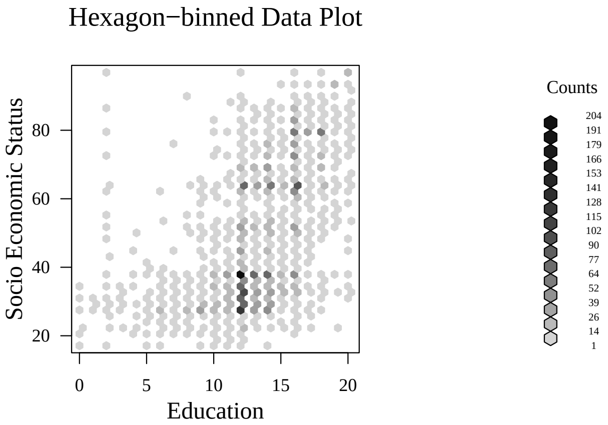
- Following usage is in classic R style
  - An object "hbin" is created (class = hexbin)
  - Then a plot method is used (which responds to common R style arguments xlab, ylab, etc)

Lately, People are looking at smarter plot types ...

```
library(hexbin)
hbin <- hexbin(dat$educ, dat$sei, xbins = 40)
plot(hbin, xlab = "Education", ylab = "Socio
    Economic Status",
      main =  "Hexagon-binned Data Plot",
      lcex = 0.6)
```

- I had some difficulty understanding how that worked, believe answer is in “ `?gplot.hexbin` ” (maybe you also run ” `methods(class = ”hexbin”)` ” to retrace my steps
- Creates six sided shapes, counts observations within
- plot method draws color-coded hexagons

Lately, People are looking at smarter plot types ...



The R lattice package implements "Trellis" plots

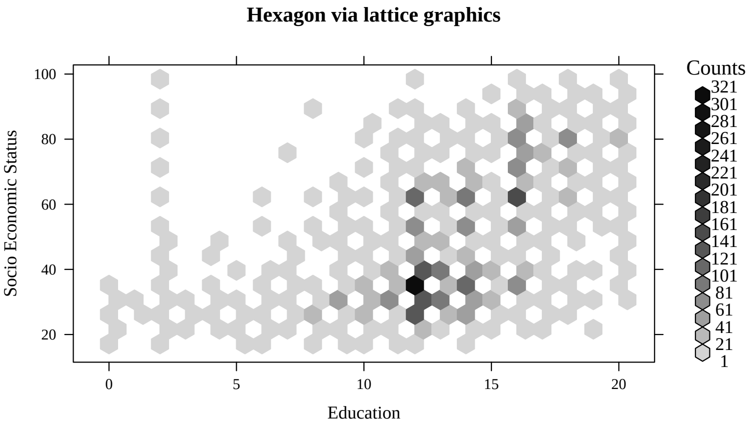
- The lattice package is a huge accomplishment by U. Wisc. PhD Deepayan Sarkar (see Sarkar, Deepayan (2008) *Lattice: Multivariate Data Visualization with R*).
- To get the flavor of it, run

```
library(lattice)
example(xyplot)
?xyplot
```

- The hexbin package includes a function that calls lattice tools, hexbinplot

```
hexbinplot(sei ~ educ, dat,
        xlab = "Education", ylab = "Socio
            Economic Status",
        main =  "Hexagon via lattice graphics")
```

The R lattice package implements ”Trellis” plots ...

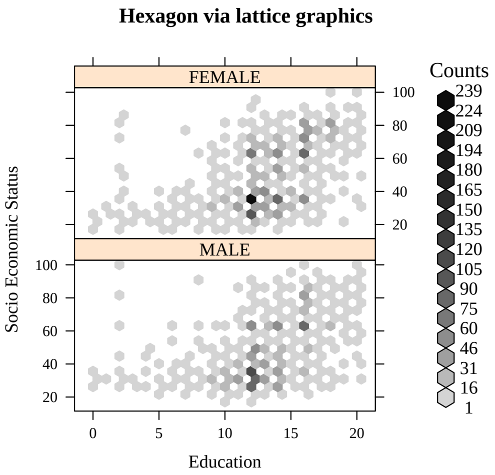


Compare Male and Female

- Lattice graphs are intended to "scale up" to display many sub-groups compactly.
    - Syntax uses bar "|" to indicate grouping variable
    - Elaborate framework for specifying style details of panels inside xyplot
- plot method draws color-coded hexagons

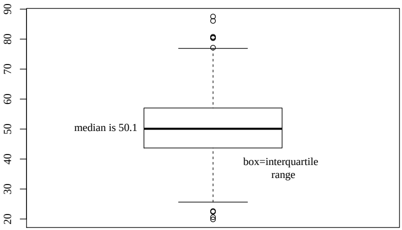
```
hexbinplot(sei ~ educ | sex, dat,
        xlab = "Education", ylab = "Socio
           Economic Status",
        main =  "Hexagon via lattice graphics")
```

Compare Male and Female ...



Boxplot: Like a Histogram Turned on its Side

- A boxplot is John Tukey's invention
- Dark line at Median
- Box has 25% of cases above and below (IQ range)
- "Whiskers" default to reach out 1.5*interquartile range
- Dots represent extreme cases.

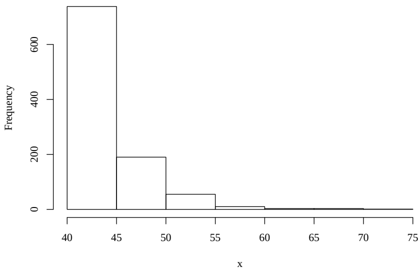


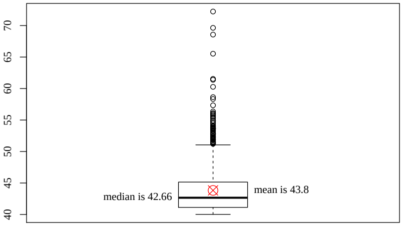

```
boxplot(x)
text(0.8, median(x),
    paste("median is",
    round(median(x), 2)),
    pos=2)
text(1.2, 37,
    paste("box=interquartile
    \n range"))
```

Boxplot: Like a Histogram Turned on its Side ...

This variable is symmetric, with mean near median of 50.

Boxplot: For a Nonsymmetric Variable






```
hist(x, main="")
```

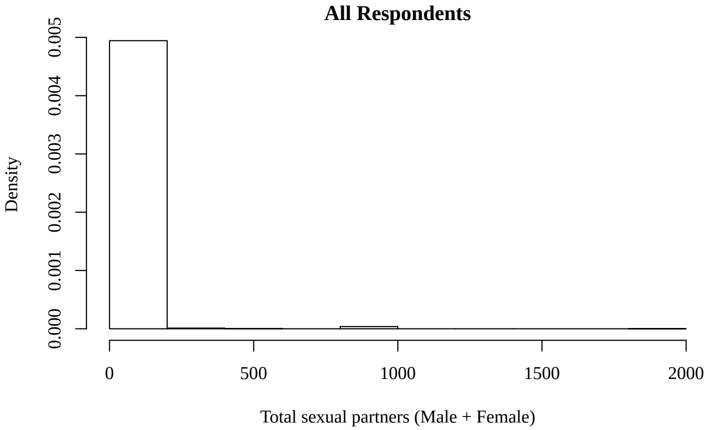
```
boxplot(x, xlim=c(0,2))
text(0.8, median(x),
    paste("median is",
    round(median(x), 2)),
    pos=2)
text(1.2, mean(x),
    paste("mean is",
    round(mean(x),1)), pos=4)
```

Boxplot Case Study in GSS Data

- A histogram can display only one group of respondents
- Boxplot can offer more compact multi-group view.
- GSS has questions about the total number of sexual partners that a person has had in their lifetimes, both male and female (what self-respecting 13 year old boy is not interested in that?)

```
dat$totnum <- dat$nummen + dat$numwomen
hist(dat$totnum, prob=TRUE, xlab="Total sexual
   partners (Male + Female)", main = "All
   Respondents")
```

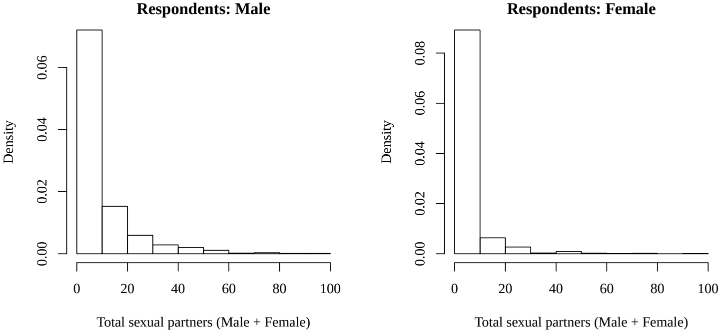
Boxplot Case Study in GSS Data ...



I concluded we'd better exclude respondents with more than 99 partners

Boxplot Case Study in GSS Data ...

Histograms for Number of Sexual Partners(GSS 2006)



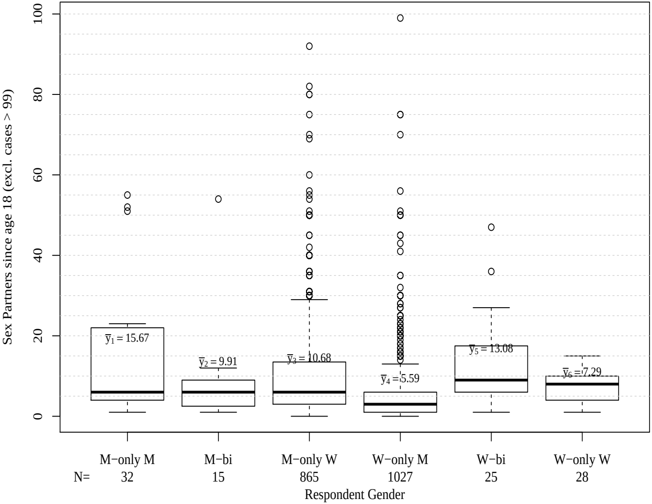
Boxplot Case Study in GSS Data ...

Use a Box Plot Instead

Boxplot Case Study in GSS Data ...

I spent about 1 million hours on this in 2007, so I insist you look

Boxplot Case Study in GSS Data ...



Barplot

- Barplot: graphic presentation of a tabulation
- Horizontal: discrete variable
- Vertical: Any numeric value (summary score ,mean, proportion, count)
- Problem: The width of the bar has no “substantive” meaning (Unlike a histogram, where the width $\times$ height represents the area)

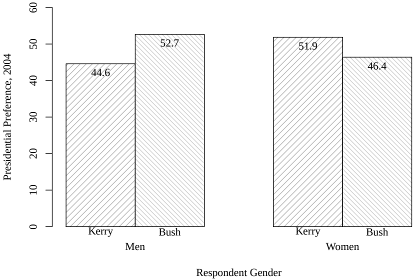
Barplot

- In R, we are asked to assemble a barplot in 2 steps
  1. Create a table that includes the values we intend to plot
     1. Usually table(), or
     2. prop.table(table()), or
     3. Any other matrix-making function, like memisc::genTable.
  2. Run the barplot() function to create the graphic

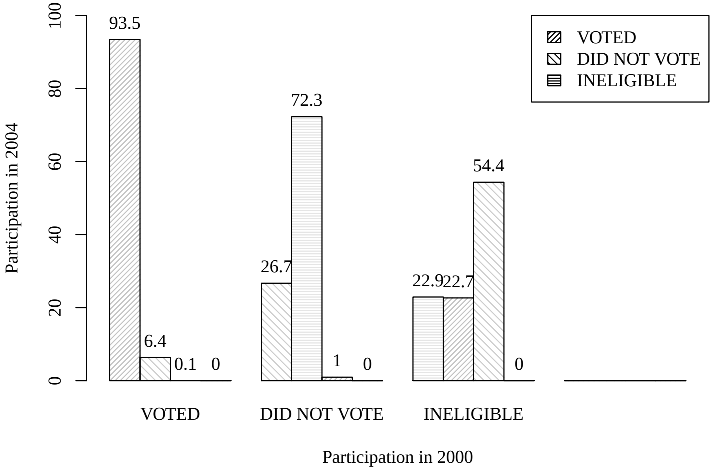
Gender Gap in 2004

Presidential Choice	Respondent Gender Male	Female
Kerry	45%	52
Bush	53	47
Nader	2	1
Didn't Vote*	1	1
Number of Cases	1137	1487

* Respondent voted, but did not cast vote in Presidential contest



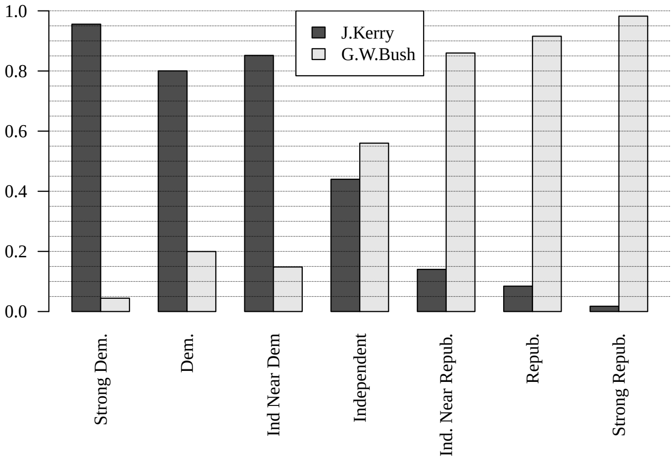
Voter Participation Dynamics



Voter Participation Dynamics ...

```
par(xpd=TRUE)
ptvote <- 100*prop.table(table(dat$vote04,
    dat$vote00),2)
mycolors <- c("gray76", "gray80", "gray90")
bpbeside <- barplot(ptvote, ylim=c(0,100), beside
    = TRUE, col = mycolors, density =
    c(30,20,40), angle = c(45,-45,0), xlab =
    "Participation in 2000", ylab =
    "Participation in 2004")
legend("topright", legend =
    levels(factor(dat$vote04)), col = mycolors,
    density = c(30,20,40), angle = c(45,-45,0))
text(as.vector(bpbeside), as.vector(ptvote),
    labels=round(as.vector(ptvote),1),pos=3)
```

Barplot: Partisanship in 2004



Barplot: Partisanship in 2004 ...

```
opar <- par(no.readonly = TRUE)
newmar <- par("mar") + c(3, 0,0,0)
par(mar = newmar)
##From the 2010 midterm notes
dat$partyid[dat$partyid %in%
    levels(dat$partyid)[8]] <- NA
dat$partyid <- factor(dat$partyid)
levels(dat$partyid) <- c("Strong Dem.", "Dem.",
    "Ind Near Dem", "Independent", "Ind. Near
    Repub.", "Repub.", "Strong Repub.")
dat$pres04[dat$pres04 %in%
    levels(dat$pres04)[3:10]]<-NA
dat$pres04 <- factor(dat$pres04)
t1 <- with(dat, prop.table(table(pres04,
    partyid), 2))
barplot(t1, beside = TRUE, las = 2, ylim = c(0,1))
abline(h=seq(0.05,1,by=0.05), lty=4, lwd=0.2)
```

Barplot: Partisanship in 2004 ...

```
legend("top", legend=c("J.Kerry","G.W.Bush"),
    fill=gray.colors(2), bg="white")
par <- opar
```

A German Student Helped me Figure this out

- Is was not truly interested in bar plots, but a young student from Germany was
- I learned a great deal, and you will too, if you step through these examples:
  http://pj.freefaculty.org/R/WorkingExamples/plot-barplot-1.R
  http://pj.freefaculty.org/R/WorkingExamples/plot-barplot-2.R
  - There are "html" output files there too
- These help not only with barplots, but also with the problem of "writing outside the plot region"

A German Student Helped me Figure this out

NB: Many Other Types of Plots

- "spinogram" is a barplot that scales the widths of the bars according to the numbers of observations
- dot plot replaces the "big boxy bars" with smaller dots to mark the tops of the bars.
- pie charts are awful, every reasonable person would agree they should never be used for anything. (my definition of reasonable is based on your answer: "do you hate pie charts?").

Outline

t.test

- Does GSS report different SES for men and women?
- $H_0 : \mu_{men} = \mu_{women}$

```
t.test(sei ~ sex, data = dat)
```

```
  Welch Two Sample t-test

data:  sei by sex
t = 0.39224, df = 4015.9, p-value = 0.6949
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.9520256  1.4282301
sample estimates:
  mean in group MALE mean in group FEMALE
            49.54071             49.30261
```

- In 2002 (or so), R Core decided to use “Welch’s unequal variance correction” for this

```
t.test(sei ~ sex, data = dat, var.equal = TRUE)
```

t.test ...

```
  Two Sample t-test

data:  sei by sex
t = 0.39354, df = 4240, p-value = 0.6939
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.9480687  1.4242732
sample estimates:
  mean in group MALE mean in group FEMALE
            49.54071             49.30261
```

t.test

- I suppose the expected value of age is smaller than 46
- NULL $H_0 : \mu_{age} \geq 46$ Alternative $H_A : \mu_{age} < 46$

```
t.test(dat$age, mu = 46, alternative = "less")
```

```
  One Sample t-test

data:  dat$age
t = 4.5289, df = 4491, p-value = 1
alternative hypothesis: true mean is less than 46
95 percent confidence interval:
     -Inf 47.55629
sample estimates:
mean of x
 47.14159
```

Outline

KU

Easy access to random number generators

- R provides a family of random number generators
- When we find new methods, it is easiest to understand them if we make up some data, so we know what we are supposed to get
- Simulation offers a "low barrier to entry" for people who want to learn more about statistical distributions

What is that Gamma thing?

- I'll create 4 variables with the same expected values
- Which should have roughly the same means in a sample of 500

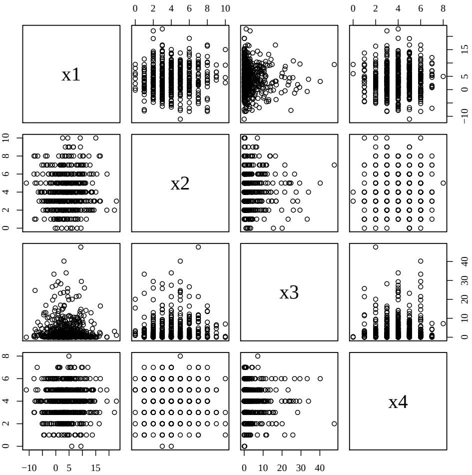
```
set.seed(234234)
N <- 500
dat2 <- data.frame(x1 = rnorm(N, m = 4, sd = 5),
                   x2 = rpois(N, lambda = 4),
                   x3 = rgamma(N, shape = 0.4,
                      scale = 10),
                   x4 = rbinom(N, size = 8, prob
                      = 0.5))
rockchalk::summarize(dat2)
```

What is that Gamma thing? ...

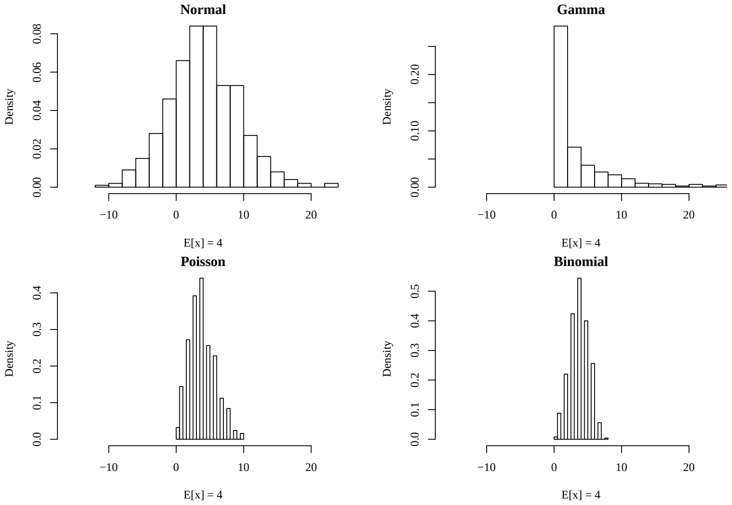
```
Numeric variables
              x1         x2         x3         x4
min        -11.064       0          0          0
med          3.983       4          1.428      4
max         22.772      10         47.616      8
mean         4.101       4.052      3.903      3.968
sd           5.163       1.999      6.273      1.453
skewness     0.203       0.450      2.954     -0.074
kurtosis     0.346      -0.084     10.911     -0.410
nobs       500        500        500        500
nmissing     0          0          0          0
```

I Cannot See Too Much in the Scatterplot Matrix

```
pairs(dat2, lwd
    = 0.8)
```



Compare Histograms



Compare Histograms ...

```
par(mfcol=c(2,2))
hist(dat2$x1, main = "Normal", prob = TRUE,
    breaks = 20,  xlab = paste("E[x] = 4"), xlim
    = c(-16,24))
hist(dat2$x2, main = "Poisson", prob = TRUE,
    breaks = 20, xlab = paste("E[x] = 4"), xlim =
    c(-16,24))
hist(dat2$x3, main = "Gamma", prob = TRUE, breaks
    = 20, xlab = paste("E[x] = 4"), xlim =
    c(-16,24))
hist(dat2$x4, main = "Binomial", prob = TRUE,
    breaks = 20, xlab = paste("E[x] = 4"), xlim =
    c(-16,24))
```

Outline

The R Experience is What You Make of It

- If you are completely inexperienced, hooray!
  - It seems certain to me that R is the best statistical & programming learning environment the planet Earth has ever known
  - R is
    - open to experimentation
    - invention of new tools
  - And yet R is disciplined and structured
- If you are experienced with other statistical software, hooray!
  - You will experience the same trauma and struggle that I did
  - Look for similarities, but don't assume they will exist
  - Other stat packs are gradually adapting to be more like R, I expect the differences will not be so start for the students in the future
    - Stata and SAS now have facilities similar to R factors, for example.

References

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Session

```
sessionInfo()
```

```
R version 3.6.0 (2019-04-26)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 19.04

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/atlas/libblas.so.3.10.3
LAPACK: /usr/lib/x86_64-linux-gnu/atlas/liblapack.so.3.10.3

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
      LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8
      LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8
     LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
```

Session ...

```
 [1] memisc_0.99.17.2  MASS_7.3-51.4     hexbin_1.27.3     tables_0.8.8
         Hmisc_4.2-0
 [6] ggplot2_3.2.0     Formula_1.2-3     survival_2.44-1.1
      lattice_0.20-38   rockchalk_1.8.144
[11] gmodels_2.18.1

loaded via a namespace (and not attached):
 [1] jsonlite_1.6        splines_3.6.0       carData_3.0-2
      gtools_3.8.1
 [5] assertthat_0.2.1    stats4_3.6.0        latticeExtra_0.6-28
      cellranger_1.1.0
 [9] pbivnorm_0.6.0      pillar_1.4.2        backports_1.1.4
      glue_1.3.1
[13] digest_0.6.20       RColorBrewer_1.1-2  checkmate_1.9.3
     minqa_1.2.4
[17] colorspace_1.4-1    htmltools_0.3.6     Matrix_1.2-17
     plyr_1.8.4
[21] pkgconfig_2.0.2     haven_2.1.0         purrr_0.3.2
     xtable_1.8-4
[25] scales_1.0.0        gdata_2.18.0        openxlsx_4.1.0
     rio_0.5.16
[29] lme4_1.1-21         htmlTable_1.13.1    tibble_2.1.3
     car_3.0-2
[33] withr_2.1.2         repr_1.0.1          nnet_7.3-12
     lazyeval_0.2.2
```

Session ...

```
[37] mnormt_1.5-5        readxl_1.3.1        magrittr_1.5
     crayon_1.3.4
[41] kutils_1.69         nlme_3.1-140        forcats_0.4.0
     foreign_0.8-71
[45] tools_3.6.0         data.table_1.12.2   hms_0.4.2
     stringr_1.4.0
[49] munsell_0.5.0       cluster_2.0.9       zip_2.0.2
     compiler_3.6.0
[53] rlang_0.4.0         grid_3.6.0          nloptr_1.2.1
     rstudioapi_0.10
[57] htmlwidgets_1.3     lavaan_0.6-3        base64enc_0.1-3
     boot_1.3-22
[61] gtable_0.3.0        abind_1.4-5         curl_3.3
     R6_2.4.0
[65] gridExtra_2.3       knitr_1.22          dplyr_0.8.3
     stringi_1.4.3
[69] Rcpp_1.0.1          rpart_4.1-15        acepack_1.4.1
     tidyselect_0.2.5
[73] xfun_0.7
```