R2

Ashish Mahabal
AyBi199, Caltech
24 May 2011
Running R

1. Create a separate sub-directory, say work, to handle particular problem.

   $ mkdir work
   $ cd work

2. Start the R program with the command

   $ R

3. At this point R commands may be issued (see...

4. To quit the R program the command is

   > q()

You can optionally save data.
Dealing with tables/objects (R frame)

- \( X = \text{read.table(”foo”, header=TRUE)} \)
- objects()
- objects(X)
- names(X)
- X
- Name1
- X$Name1
- attach(X)
- Name1
Assignments

- \texttt{x <- c(10.4, 5.6, 3.1, 6.4, 21.7)}
- \texttt{assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))}
- \texttt{c(10.4, 5.6, 3.1, 6.4, 21.7) -> x} \hspace{1cm} (!)
- \texttt{1/x} \hspace{1cm} \# 0.09, 0.17, 0.32, 0.15, 0.04
- \texttt{y <- c(x,0,x)} \hspace{1cm} \# 10.4,..,21.7,0,10.4,..,21.7
- \texttt{v <- rep(x) + y + 1} \hspace{1cm} \# x repeated 2.2 times, 1 11
- \texttt{var = sum((x-mean(x))^2)/(length(x)-1)}
Lists (and data.frames=restricted)

• Lst <- list(name="Fred", wife="Mary", no.children=3, child.ages=c(4,7,9))
• Always numbered
  • Lst[[4]] # returns  [1] 4 7 9
  • Lst[[4]][2] # returns 7
  • Lst[4][2] # returns Null
Reading from files

- `HousePrice <- read.table("houses.data", header=TRUE)`
- `read.table(file, header = FALSE, sep = "\", quote = "\"\",
  dec = ".", row.names, col.names, as.is = !stringsAsFactors,
  na.strings = "NA", colClasses = NA, nrows = -1, skip = 0, check.names =
  TRUE, fill = !blank.lines.skip, strip.white = FALSE, blank.lines.skip = TRUE,
  comment.char = "#", allowEscapes = FALSE, flush = FALSE,
  stringsAsFactors = default.stringsAsFactors(), fileEncoding = "",
  encoding = "unknown")`

`read.csv, read.csv2, read.delim, read.delim2`
Accessing built-in datasets

- `data()`
- `data(AirPassengers)`
- `?AirPassengers`
- `new <- edit(AirPassengers)`
- `x <- array(c(AirPassengers[1:144]), dim=c(12,12))`
- `pairs(x)`
• ? To get help
• # for comments
• -> or <- (or =) for assignments
• == for comparison

• Concatenate with c # x = c(a,b)
• seq to create sequence # z = seq(-30,30)
• is.na(x) to look for notavailability
• summary(x)
• data(x)
• read.table(file)
• save(file)
• source(file)  # input
• sink(file)    # output
Common routines

- `length(a)`  # length of vector
- `max(a)`     # similarly min, mean
- `sort(a)`, or `sort(a, decreasing=T)`
- `unique(a)`
- `duplicated(a)`  # returns a logical array!
Example

- `a=rnorm(100,mean=5,sd=1)`
- `b=rnorm(200,mean=5,sd=1)`
- `c=c(100*a,100*b)`
- `length(c)`
- `c`
• $d = \text{as.integer(c)}$

• $d$

• $\text{mean}(d)$; $\text{max}(d)$; $\text{min}(d)$; $\text{sd}(d)$

```
> d = c(1, 2, 3, 4, 5)
> mean(d)
[1] 3
> max(d)
[1] 5
> min(d)
[1] 1
> sd(d)
[1] 1.414214
```
• unique(d)
• duplicated(d)
• plot(duplicated(d))  # coercion
• duplicated(unique(d))
• five(d)      # what's the five-of-d
• ?five       # nothing? Do you know five?
• ??five      # at least something that has five?
• fivenum(d)  # ah, so it's fivenum. I'll take that
• ?fivenum    # what is it BTW?
... minimum, lower-hinge, median, upper-hinge, maximum ...
• `fivenum(x, na.rm = TRUE)`  # options!
• g=d
• g[duplicated(d)]<-NA
• `fivenum(d)`
  – [1] 264.0 425.5 483.5 561.5 742.0
• `fivenum(g,na.rm = TRUE)`
  – [1] 264.0 419.5 492.0 571.5 742.0
• boxplot.stats(x, coef = 1.5, do.conf = TRUE, do.out = TRUE)
• quantile, range, bxp
• which selects indices:
• a=which(d<500)  # returns indices
• a=d[which(d<500)]  # returns the elements
• which.max(d)       # index of max element
• `match(x,y)`  # elements of x in y, else NA
• `merge(a,b)`  # using common columns/rows

```r
> t1=read.table('table1',header=TRUE)
> t1
   id  ra  dec
1 101 200.1 33.1
2 102 199.3 -13.3
3 103 200.2 19.1
> t2=read.table('table2',header=TRUE)
> t2
   id mag
1 101 17
2 102 18
3 104 19
```

```r
> merge(t1,t2)
   id  ra  dec mag
1 101 200.1 33.1 17
2 102 199.3 -13.3 18
```
choose(n, k)  # combinations of k from n
choose(5, 3) returns 10
sample(x, size)  # resamples size elements from x (with the option of replacement)

> cards = paste(c("C","D","H","S"), rep(1:13,times=4), sep="")
> cards
[1] "C1" "D2" "H3" "S4" "C5" "D6" "H7" "S8" "C9" "D10" "H11" "S12"
[13] "C13" "D1" "H2" "S3" "C4" "D5" "H6" "S7" "C8" "D9" "H10" "S11"
[25] "C12" "D13" "H1" "S2" "C3" "D4" "H5" "S6" "C7" "D8" "H9" "S10"
[37] "C11" "D12" "H13" "S1" "C2" "D3" "H4" "S5" "C6" "D7" "H8" "S9"
[49] "C10" "D11" "H12" "S13"
• > sample(cards,5)
  [1] "S9" "H4" "D5" "C3" "C9"
• > sample(cards,2)
  [1] "S1" "S11"

> sample(cards, 60, replace=TRUE)
• Conj(4+3i)
• fft(x)
• solve(a)  # matrix inverse of a
• solve(a,b)  # solves $a \times x = b$ for $x$
Graphics in R
R graphics, comparison

• Base graphics (Ross Ihaka, S graphics driver)
  – Can not be modified
• Grid graphics (Paul Murrell, 2000)
  – Low level, but no support for statistical graphics
• Lattice package (Sarkar, 2008)
  – Based on grid. Detailed. Lacks a formal model
• ggplot2 (Hedly, 2005)
  – Layered, based on grammar of graphics, low level,
    static plots, extensions easy, flow like that of analysis
• ggobi, rggobi (Cook, Swayne, Wickham 2007,8)
  – Interactive, grammar of graphics

http://cran.r-project.org/web/views/Graphics.html (online list)
Grammar of graphics (Wilkinson, 2005)

- Mapping *(mappings)* from data *(data)* to aesthetic attributes of geometric objects *(geoms)*.
- Including statistical transformations *(stats)* in specific coordinate systems
- Faceting *(facet)* to allow easy subsetting (synonyms: conditioning, latticing, trellising)
- *scale, coord*
Hadley Wickham’s ggplot2
http://had.co.nz/ggplot2/

- install.packages("ggplot2")
- library(ggplot2)
Diamonds

http://www.diamondse.info/

- help(diamonds)
- data(diamonds)
- dsmall <- diamonds[sample(nrow(diamonds), 100), ]
- dsmall
- help(ggsave)

Aesthetics: color, size, shape

```r
ggsave(filename="diamond_smooth.png", plot=last_plot(), height=3, width=5, dpi=100)
```
<table>
<thead>
<tr>
<th>carat</th>
<th>cut</th>
<th>color</th>
<th>clarity</th>
<th>depth</th>
<th>table</th>
<th>price</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>35183</td>
<td>0.31</td>
<td>Ideal</td>
<td>G</td>
<td>IF</td>
<td>62.0</td>
<td>54.0</td>
<td>891</td>
<td>4.36</td>
<td>4.38</td>
</tr>
<tr>
<td>205</td>
<td>0.98</td>
<td>Fair</td>
<td>H</td>
<td>SI2</td>
<td>67.9</td>
<td>60.0</td>
<td>2777</td>
<td>6.05</td>
<td>5.97</td>
</tr>
<tr>
<td>52175</td>
<td>0.70</td>
<td>Premium</td>
<td>E</td>
<td>SI2</td>
<td>62.8</td>
<td>59.0</td>
<td>2475</td>
<td>5.69</td>
<td>5.64</td>
</tr>
<tr>
<td>39992</td>
<td>0.41</td>
<td>Ideal</td>
<td>F</td>
<td>VS2</td>
<td>61.1</td>
<td>56.0</td>
<td>1107</td>
<td>4.83</td>
<td>4.80</td>
</tr>
<tr>
<td>2776</td>
<td>0.90</td>
<td>Good</td>
<td>G</td>
<td>SI2</td>
<td>61.8</td>
<td>56.0</td>
<td>3253</td>
<td>6.11</td>
<td>6.16</td>
</tr>
<tr>
<td>22560</td>
<td>1.43</td>
<td>Premium</td>
<td>G</td>
<td>VS2</td>
<td>62.2</td>
<td>58.0</td>
<td>10609</td>
<td>7.18</td>
<td>7.15</td>
</tr>
<tr>
<td>22586</td>
<td>1.35</td>
<td>Ideal</td>
<td>H</td>
<td>VVS1</td>
<td>61.9</td>
<td>57.0</td>
<td>10639</td>
<td>7.06</td>
<td>7.09</td>
</tr>
<tr>
<td>44761</td>
<td>0.52</td>
<td>Very Good</td>
<td>E</td>
<td>VS2</td>
<td>62.8</td>
<td>55.0</td>
<td>1621</td>
<td>5.11</td>
<td>5.15</td>
</tr>
<tr>
<td>47749</td>
<td>0.54</td>
<td>Ideal</td>
<td>D</td>
<td>VS2</td>
<td>61.3</td>
<td>56.0</td>
<td>1892</td>
<td>5.22</td>
<td>5.26</td>
</tr>
<tr>
<td>34734</td>
<td>0.37</td>
<td>Ideal</td>
<td>D</td>
<td>VS2</td>
<td>61.0</td>
<td>56.0</td>
<td>876</td>
<td>4.65</td>
<td>4.70</td>
</tr>
<tr>
<td>4107</td>
<td>0.78</td>
<td>Ideal</td>
<td>F</td>
<td>VS1</td>
<td>61.6</td>
<td>56.0</td>
<td>3537</td>
<td>5.90</td>
<td>5.92</td>
</tr>
</tbody>
</table>
Shape is the aesthetic here and a categorical variable
qplot(carat, price, data=dsmall, colour=color)

color is the aesthetic here and a categorical variable
qplot(carat, price, data=diamonds, geom=c("smooth","point"))

An example of a statistical plot
qplot(log(carat), log(price), data=diamonds, geom=c("smooth","point"))

se=False to turn confidence level off
qplot(carat, price, data=dsmall, geom=c("smooth","point"), span=1)
qplot(carat, price, data=dsmall, geom=c("smooth","point"), span=0.2)

Span can be varied between 0 (very wiggly) and 1 (not wiggly)
qplot(carat, x*y*z, data=diamonds)

Easy to use multiple variables. Outliers by volume
qplot(carat, price, data=diamonds, alpha=1/(1/500))

Transparency (scale 0 to 1): 1/1 => not transparent
qplot(color, price / carat, data = diamonds, geom = "jitter")
```r
qplot(color, price / carat, data = diamonds, geom = "jitter", alpha=I(1/50))
```
qplot(color, price / carat, data = diamonds, geom = "boxplot")
qplot(carat, data=diamonds, geom="density", colour = color)

Combining aesthetic and geom
Fuel efficiency data
http://fueleconomy.gov

- data(mpg)
- help(mpg)
- 234 rows, 11 vars
- 1999-2008 popular cars

<table>
<thead>
<tr>
<th>manufacturer</th>
<th>model</th>
<th>displ</th>
<th>year</th>
<th>cyl</th>
<th>trans</th>
<th>drv</th>
<th>cty</th>
<th>hwy</th>
<th>fl</th>
</tr>
</thead>
<tbody>
<tr>
<td>chevrolet</td>
<td>k1500 tahoe 4wd</td>
<td>5.3</td>
<td>2008</td>
<td>8</td>
<td>auto(14)</td>
<td>4</td>
<td>14</td>
<td>19</td>
<td>r</td>
</tr>
<tr>
<td>chevrolet</td>
<td>corvette</td>
<td>7.0</td>
<td>2008</td>
<td>8</td>
<td>manual(m6)</td>
<td>r</td>
<td>15</td>
<td>24</td>
<td>p</td>
</tr>
<tr>
<td>audi</td>
<td>a4 quattro</td>
<td>2.0</td>
<td>2008</td>
<td>4</td>
<td>auto(s6)</td>
<td>4</td>
<td>19</td>
<td>27</td>
<td>p</td>
</tr>
<tr>
<td>dodge</td>
<td>ram 1500 pickup 4wd</td>
<td>4.7</td>
<td>2008</td>
<td>8</td>
<td>auto(l5)</td>
<td>4</td>
<td>13</td>
<td>17</td>
<td>r</td>
</tr>
<tr>
<td>toyota</td>
<td>camry solara</td>
<td>2.4</td>
<td>2008</td>
<td>4</td>
<td>manual(m5)</td>
<td>f</td>
<td>21</td>
<td>31</td>
<td>r</td>
</tr>
<tr>
<td>subaru</td>
<td>forester awd</td>
<td>2.5</td>
<td>2008</td>
<td>4</td>
<td>manual(m5)</td>
<td>4</td>
<td>20</td>
<td>27</td>
<td>r</td>
</tr>
<tr>
<td>honda</td>
<td>civic</td>
<td>1.6</td>
<td>1999</td>
<td>4</td>
<td>manual(m5)</td>
<td>f</td>
<td>28</td>
<td>33</td>
<td>r</td>
</tr>
<tr>
<td>chevrolet</td>
<td>c1500 suburban 2wd</td>
<td>5.3</td>
<td>2008</td>
<td>8</td>
<td>auto(l4)</td>
<td>r</td>
<td>14</td>
<td>20</td>
<td>r</td>
</tr>
<tr>
<td>chevrolet</td>
<td>c1500 suburban 2wd</td>
<td>5.3</td>
<td>2008</td>
<td>8</td>
<td>auto(l4)</td>
<td>r</td>
<td>11</td>
<td>15</td>
<td>e</td>
</tr>
<tr>
<td>dodge</td>
<td>dakota pickup 4wd</td>
<td>4.7</td>
<td>2008</td>
<td>8</td>
<td>auto(l5)</td>
<td>4</td>
<td>14</td>
<td>19</td>
<td>r</td>
</tr>
</tbody>
</table>

Treehugger.com
qplot(displ, hwy, data = mpg, colour = cyl)

qplot(displ, hwy, data = mpg, colour = factor(cyl))
aesthetics: color; geom: point

The geom could as easily have been “line” or even point and line!
qplot(displ, hwy, data=mpg, facets = . ~ year)
qplot(displ, hwy, data=mpg, facets = . ~ year) + geom_smooth()
Anatomy of a plot

- scale transformations
- stat transformations
  - After scaling so that log/lin make sense
- train scales
  - Superset across different layers
- map data to aesthetics
- render geoms
What is a plot (made of)?

- A *dataset*
- Set of *mappings* from variables to aesthetics
- At least one layer, each made of
  - A geometric object (*geom*)
  - A statistical transformation (*stat*)
  - A position adjustment
  - Perhaps a dataset and aesthetic mappings
- One *scale* per mapping
- A *coordinate* system
- *Faceting* specification
Example of scale continuous and discrete variables
p <- qplot(displ, hwy, data = mpg, colour = factor(cyl))

Things to do with an object

• print() – automatically unless in a loop
• ggsave() – to write out an image
• summary() – summary about the makeup
• save() – to save the object
summary(p)

• data: manufacturer, model, displ, year, cyl, trans, drv, cty, hwy, fl, class [234x11]
• mapping: colour = factor(cyl), x = displ, y = hwy
• scales: colour, x, y
• faceting: facet_grid(. ~ ., FALSE)

• geom_point:
• stat_identity:
• position_identity: (width = NULL, height = NULL)
• > # Save plot object to disk
  • > save(p, file = "plot.rdata")
• > # Load from disk
  • > load("plot.rdata")
• > # Save png to disk
  • > ggsave("plot.png", width = 5, height = 5)
Adding layers

- p <- ggplot(diamonds, aes(carat, price, colour = cut))
- p <- p + layer(geom = "point") #stat/pos defaults
- layer(geom, geom_params, stat, stat_params, data, mapping, position)
- p <- ggplot(diamonds, aes(x = carat))
- p <- p + layer(
  geom = "bar",
  geom_params = list(fill = "steelblue"),
  stat = "bin",
  stat_params = list(binwidth = 2) )
- p
Every geom associated with a (default) stat

• `geom_histogram(binwidth = 2, fill = "steelblue")`

And every stat with a (default) geom

stat: bin
geom: bar
28819 movies from IMDB

24 variables including Length of movie. Linear and log plots
qplot(length, data=movies, geom="histogram", binwidth=1, xlim=c(80,100))
Blazar lightcurve example

MJD (actual - 53600)

Mag

Flux (x10^-9)

Type
- CRTS
- PQ

Judy Mou
Some more links ... 

- [http://vostat.org](http://vostat.org)
- R GUI (R commander [http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/](http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/))
- [http://scc.stat.ucla.edu/mini-courses](http://scc.stat.ucla.edu/mini-courses)
- [http://cran.r-project.org](http://cran.r-project.org)
- [http://www.r-project.org](http://www.r-project.org)
Homework

• Apply various graphing techniques learnt to the astronomy dataset
  • http://avyakta.caltech.edu:8080/datasets/dataset2.Rframe
  • Flux ratios aka Colors (u-g, g-r, r-I, i-z) and spectroscopic class

• num umg gmr rmi imz specClass
  • 1  1.80  0.35  0.42  0.29  3
  • 2  1.22  0.01  -0.05  -0.06  1
colors and classification

- `d2 = read.table("dataset2.Rframe", header=TRUE)`
- `objects()`
- `names(d2)`
- `umg`
- `d2$umg`
- `attach(d2)`
- `umg`
- `plot(umg, gmr)`
- `pairs(d2)`

Use ggplot2 methods instead of the above