MODULE II
DATA EXPLORATION

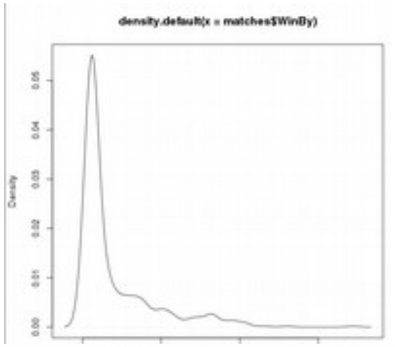
INFO 590: Applied Data Science
Summer 2017

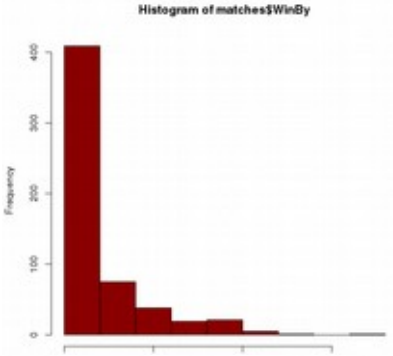
INTRO

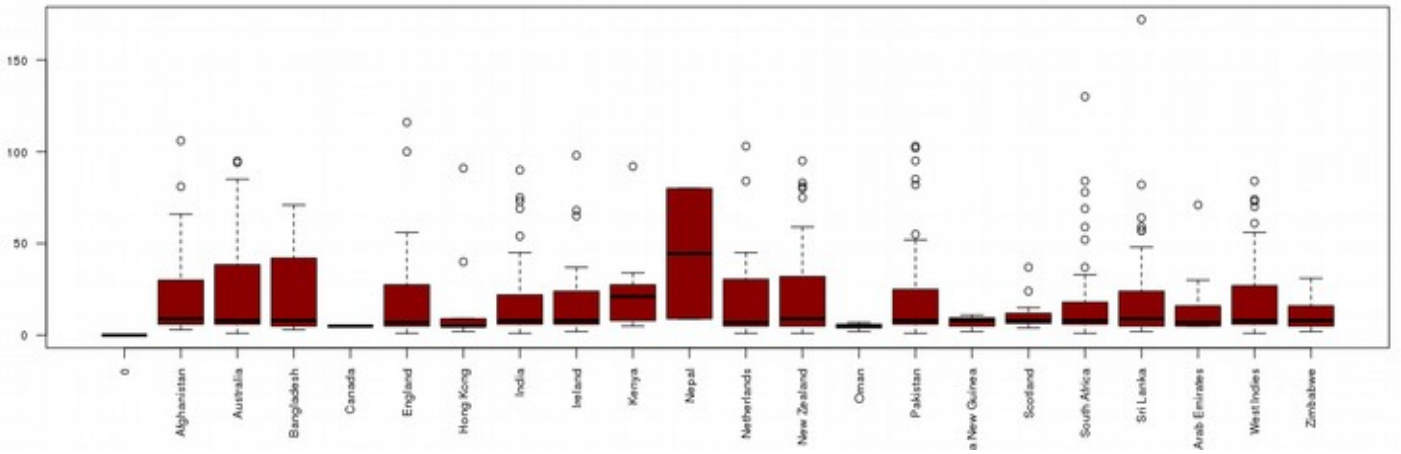
DATA EXPLORATION

```
'data.frame':   569 obs. of  13 variables:
 $ ID           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Team1        : Factor w/ 21 levels "Afghanistan",..: 21 2 2 13 19 18 2 12 6 3 ...
 $ Team2        : Factor w/ 21 levels "Afghanistan",..: 7 18 20 14 16 12 5 16 2 21 ...
 $ PlayerOfMatch: Factor w/ 269 levels "0","A Bagai",..: 52 243 130 188 27 66 244 115 14 164 ...
 $ Date         : Factor w/ 445 levels "2005-02-17","2005-06-13",..: 320 27 214 127 358 76 130 423 236 351 ...
 $ CoinFlipWin  : Factor w/ 21 levels "Afghanistan",..: 21 2 20 14 19 12 2 11 2 21 ...
 $ CoinDec      : Factor w/ 2 levels "bat","field": 1 2 1 2 1 1 1 2 2 1 ...
 $ Location     : Factor w/ 79 levels "0","Abu Dhabi",..: 1 15 12 38 29 23 1 2 20 45 ...
 $ Stadium      : Factor w/ 94 levels "Adelaide Oval",..: 28 52 10 67 31 62 1 70 63 69 ...
 $ NeutralVenue : int  0 1 0 0 0 0 0 1 0 0 ...
 $ Winner       : Factor w/ 22 levels "0","Afghanistan",..: 22 3 21 13 20 13 6 17 6 22 ...
 $ WinType      : Factor w/ 3 levels "0","runs","wickets": 2 3 2 2 2 2 3 2 2 2 ...
 $ WinBy        : int  10 10 27 39 9 3 1 7 27 31 ...
> 
```

```
> summary(matches$CoinDec)
  bat field
  289   280
> 
```







READINGS

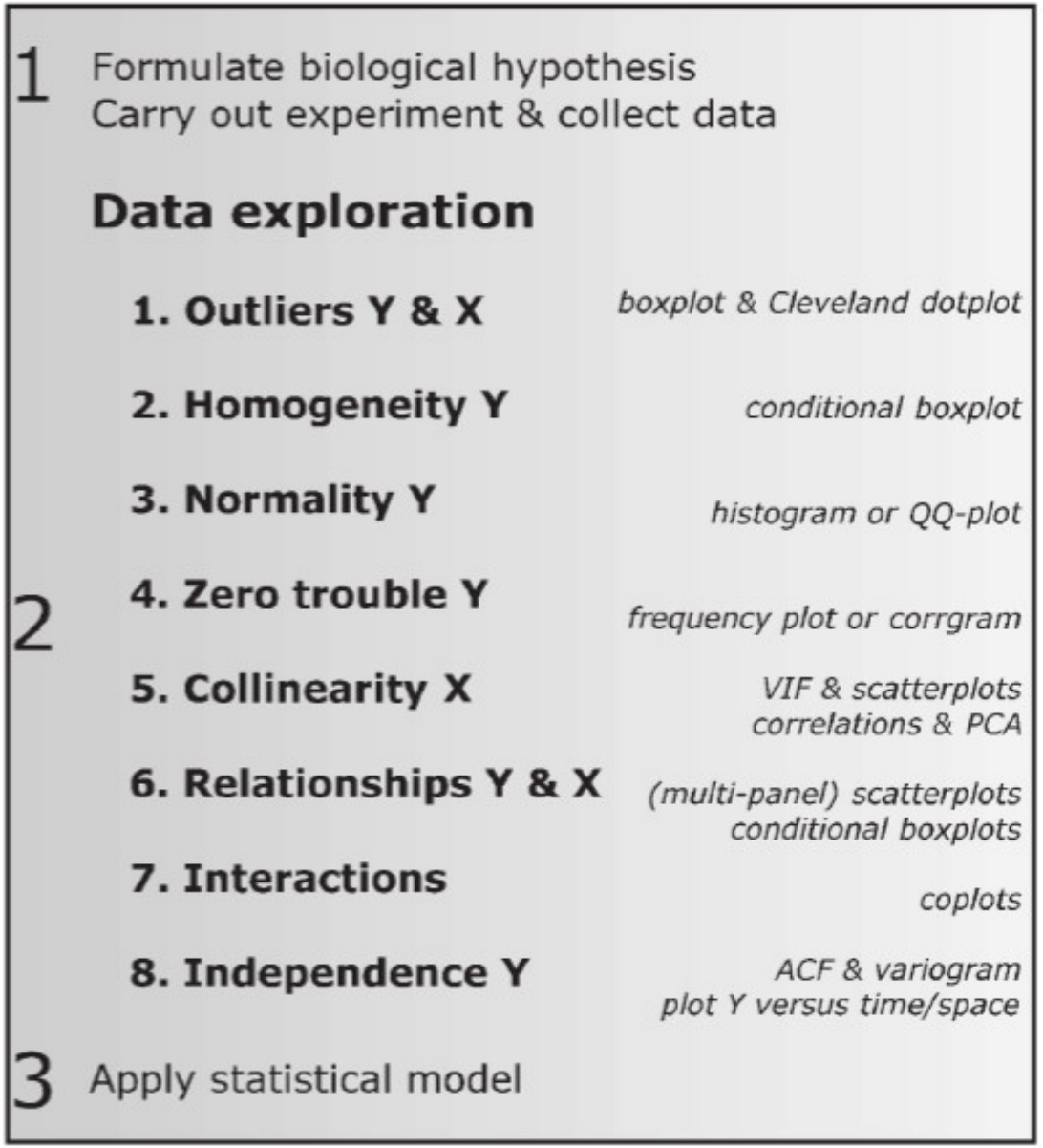


Fig. 1. Protocol for data exploration.

READINGS

Rcommander

a graphical interface for R

R commander (Rcmdr)

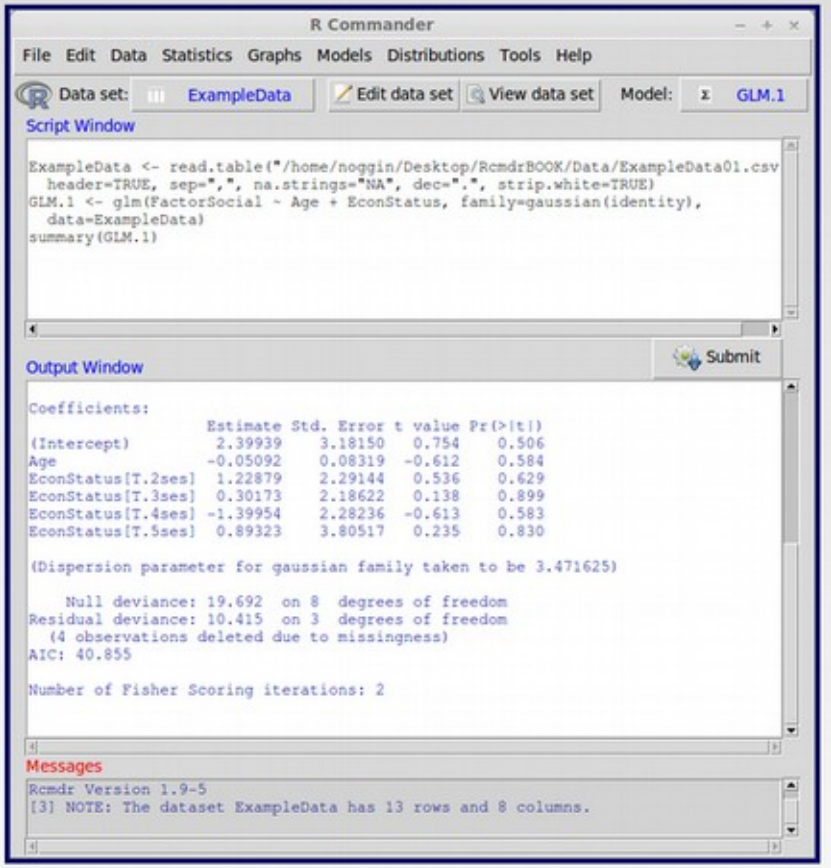
R provides a powerful and comprehensive system for analysing data and when used in conjunction with the R-commander (a graphical user interface, commonly known as Rcmdr) it also provides one that is easy and intuitive to use. Basically, R provides the engine that carries out the analyses and Rcmdr provides a convenient way for users to input commands. The Rcmdr program enables analysts to access a selection of commonly-used R commands using a simple interface that should be familiar to most computer users. It also serves the important role of helping users to implement R commands and develop their knowledge and expertise in using the command line ··· an important skill for those wishing to exploit the full power of the program.

READINGS

Information about installing R can be found at the website of the R homepage http://www.r-project.org/ which provides lots of information about the R project and also directs users to one of the CRAN sites (the Comprehensive R Archive Network) that have been set up on many servers across the world in order for users to download the software. CRAN provides all files necessary to install R on a number of different computing platforms (Linux, MacOS X and Windows) along with detailed information about installation and also offers manuals and contributed documentation in a number of langauges and for a number of specific disciplines.

Definitive information about the Rcmdr can be found at it's author's (John Fox) webpage:

http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/



TUTORIAL

WALK THROUGH: CRICKET DATA

CRICKET DATA EXPLORATION WALKTHROUGH

INFO 590: Applied Data Science
Summer 2017

LOADING DATA

LOADING DATA

```
setwd("~/Documents/ADS-SU17/")
matches <- read.csv("match-data-full.csv")
balls <- read.csv("ball-data-full.csv")
```

Get or Set Working Directory

Description

getwd returns an absolute filepath representing the current working directory of the R process; setwd(dir) is used to set the working directory to dir.

Usage

```
getwd()
setwd(dir)
```

Data Input

Description

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           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", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

STRUCTURE

What do we have here?

```
> str(matches)
'data.frame':   569 obs. of  13 variables:
 $ ID           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Team1        : Factor w/ 21 levels "Afghanistan",..: 21 2 2 13 19 18 2 12 6 3 ...
 $ Team2        : Factor w/ 21 levels "Afghanistan",..: 7 18 20 14 16 12 5 16 2 21 ...
 $ PlayerOfMatch: Factor w/ 269 levels "0","A Bagai",..: 52 243 130 188 27 66 244 115 14 164 ...
 $ Date         : Factor w/ 445 levels "2005-02-17","2005-06-13",..: 320 27 214 127 358 76 130 423 236 351 ...
 $ CoinFlipWin  : Factor w/ 21 levels "Afghanistan",..: 21 2 20 14 19 12 2 11 2 21 ...
 $ CoinDec      : Factor w/ 2 levels "bat","field": 1 2 1 2 1 1 1 2 2 1 ...
 $ Location     : Factor w/ 79 levels "0","Abu Dhabi",..: 1 15 12 38 29 23 1 2 20 45 ...
 $ Stadium      : Factor w/ 94 levels "Adelaide Oval",..: 28 52 10 67 31 62 1 70 63 69 ...
 $ NeutralVenue : int  0 1 0 0 0 0 0 1 0 0 ...
 $ Winner       : Factor w/ 22 levels "0","Afghanistan",..: 22 3 21 13 20 13 6 17 6 22 ...
 $ WinType      : Factor w/ 3 levels "0","runs","wickets": 2 3 2 2 2 2 3 2 2 2 ...
 $ WinBy        : int  10 10 27 39 9 3 1 7 27 31 ...
> str(balls)
'data.frame':   131321 obs. of  11 variables:
 $ GID        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Batting    : Factor w/ 22 levels "Afghanistan",..: 22 22 22 22 22 22 22 22 22 22 ...
 $ Over       : int  0 0 0 0 0 0 1 1 1 1 ...
 $ Ball       : int  1 2 3 4 5 6 1 2 3 4 ...
 $ Bowler     : Factor w/ 671 levels "A Bhattarai",..: 79 79 79 79 79 79 545 545 545 545 ...
 $ Batsman    : Factor w/ 897 levels "A Bagai","A Balbirnie",..: 284 284 284 284 284 284 284 151 151 284 ...
 $ NonStriiker: Factor w/ 891 levels "A Bagai","A Balbirnie",..: 151 151 151 151 151 151 151 284 284 151 ...
 $ TotalRuns  : int  0 0 2 2 0 3 1 0 1 1 ...
 $ BatterRuns : int  0 0 2 2 0 3 0 0 1 0 ...
 $ WicketType : Factor w/ 9 levels "0","bowled","caught",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ PlayerOut  : Factor w/ 817 levels "0","A Bagai",..: 1 1 1 1 1 1 1 1 1 1 ...
>
```

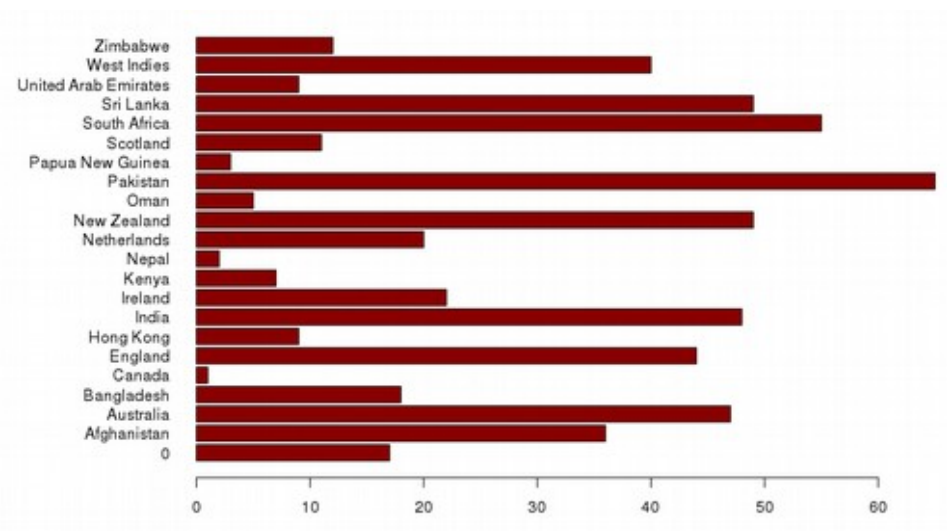
str(x) shows us what “types” of data we have, int, str, factor, etc. and how much data we have

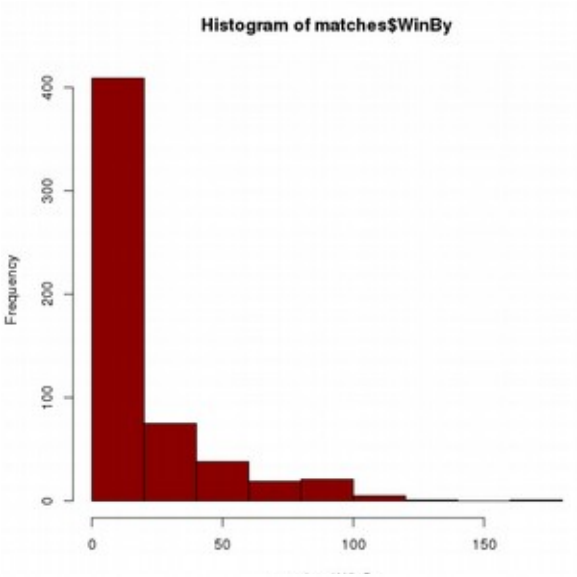
int = Integer (counting numbers: 1,2,3,4,5)
Factor = Bounded list (e.g., U.S. states because there are only 50)
Other types include string (text, unbounded) and float / double (decimals)

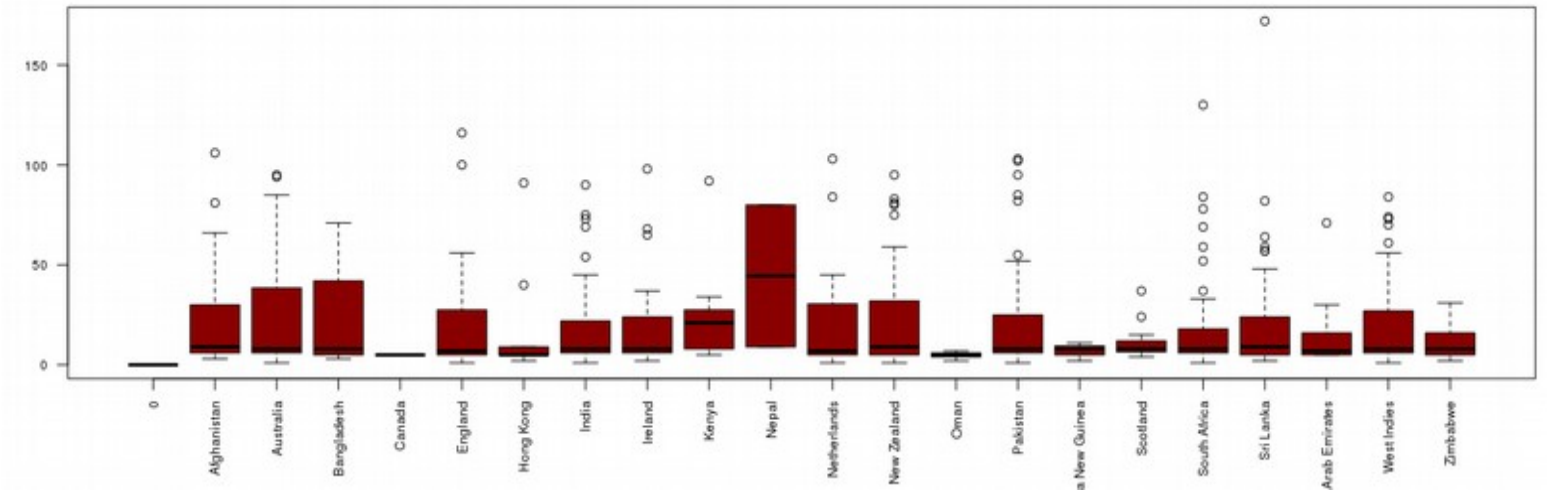
SHAPE

Simple charts

```
 8  hist(matches$WinBy,col="darkred")
 9  barplot(table(matches$Winner),cex.lab=.75,las=1,mar=c(1,3,1,1),col="darkred",horiz = T)
10  boxplot(matches$WinBy~matches$Winner,cex.lab=.5,las=3,col="darkred",mar=c(3,1,1,1)
```



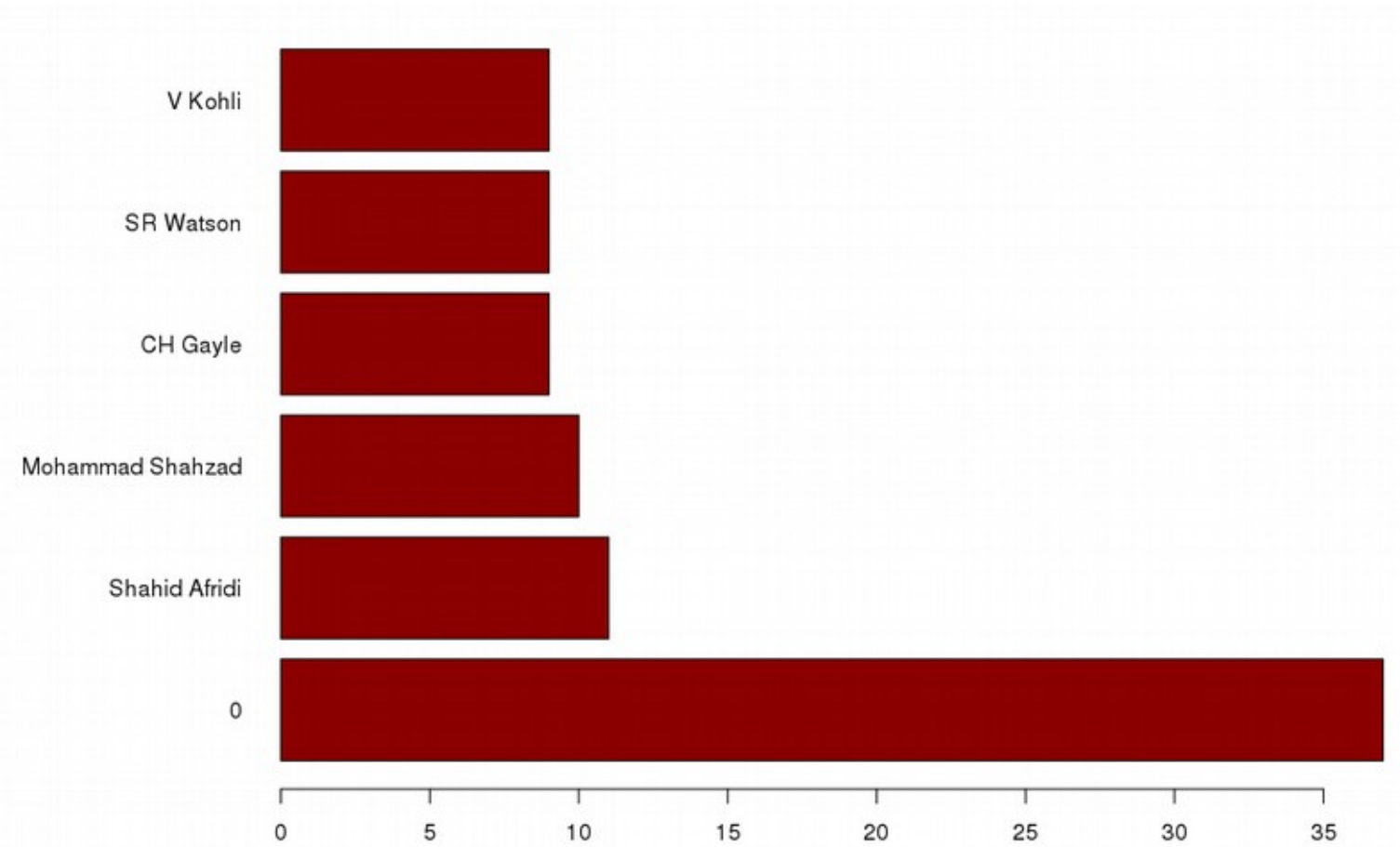




SHAPE

More complex charts...

```
14  barplot(sort(table(matches$PlayerOfMatch),decreasing =T)[seq(1,6)],
15          cex.lab=.75,las=1,col="darkred",horiz = T)
16
17
```



SUMMARY

summary(x)

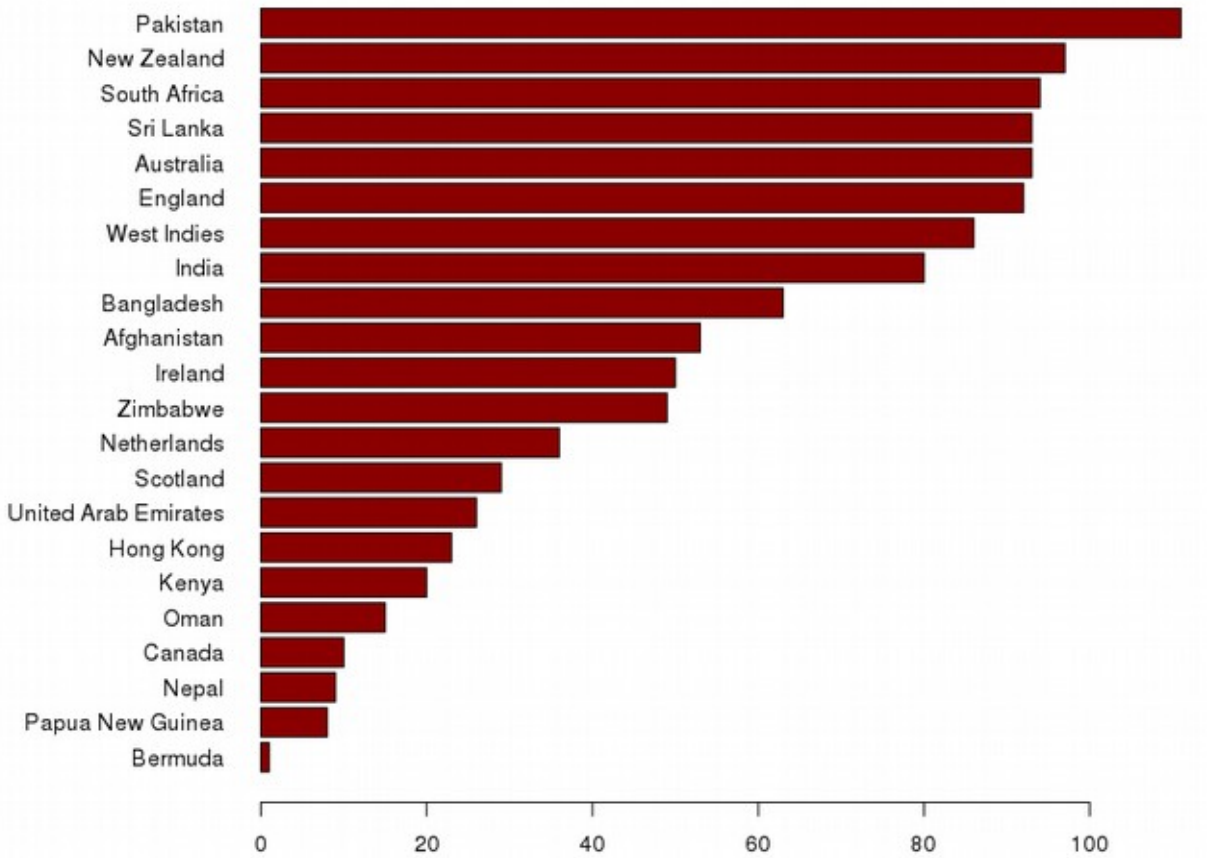
```
> summary(matches)
       ID                Team1                Team2                PlayerOfMatch          Date          CoinFlipWin  CoinDec            Location
 Min.   :   1   Australia   : 63   Pakistan    : 86   0               : 37   2017-01-20:  4   Pakistan    : 59   bat  :289   0             :114
 1st Qu.:143   England     : 60   Sri Lanka   : 57   Shahid Afridi   : 11   2007-09-12:  3   New Zealand : 52   field:280   Mirpur        : 36
 Median :285   New Zealand : 58   West Indies : 55   Mohammad Shahzad: 10   2007-09-14:  3   Australia   : 47               Colombo       : 27
 Mean   :285   South Africa: 48   South Africa: 46   CH Gayle        :  9   2007-09-16:  3   England     : 47               Abu Dhabi     : 24
 3rd Qu.:427   Afghanistan : 45   New Zealand : 39   SR Watson       :  9   2007-09-18:  3   West Indies : 46               Johannesburg: 24
 Max.   :569   India       : 42   India       : 38   V Kohli         :  9   2007-09-20:  3   South Africa: 42               London        : 24
               (Other)     :253   (Other)     :248   (Other)         :484   (Other)   :550   (Other)     :276               (Other)       :320
                                  Stadium     NeutralVenue            Winner       WinType       WinBy
 Dubai International Cricket Stadium: 43   Min.   :0.0000   Pakistan    : 65   0       : 17   Min.   :  0.00
 Shere Bangla National Stadium      : 36   1st Qu.:0.0000   South Africa: 55   runs    :288   1st Qu.:  5.00
 R Premadasa Stadium                : 26   Median :0.0000   New Zealand : 49   wickets:264   Median :  8.00
 New Wanderers Stadium              : 24   Mean   :0.4569   Sri Lanka   : 49                 Mean   : 19.41
 Sheikh Zayed Stadium               : 24   3rd Qu.:1.0000   India       : 48                 3rd Qu.: 25.00
 Kensington Oval, Bridgetown        : 17   Max.   :1.0000   Australia   : 47                 Max.   :172.00
 (Other)                            :399                    (Other)     :256
> |
```

```
> summary(balls)
      GID                Batting           Over             Ball             Bowler            Batsman           NonStriiker
 Min.   :  1.0   Pakistan    :13249   Min.   : 0.000   Min.   : 1.000   Shahid Afridi  :  2132   BB McCullum    :  1629   TM Dilshan     :  1755
 1st Qu.:144.0   New Zealand :11015   1st Qu.: 4.000   1st Qu.: 2.000   SL Malinga     :  1513   TM Dilshan     :  1553   BB McCullum    :  1580
 Median :287.0   Sri Lanka   :10842   Median : 9.000   Median : 4.000   Saeed Ajmal    :  1457   Mohammad Hafeez:  1430   Shoaib Malik   :  1450
 Mean   :286.5   South Africa:10732   Mean   : 9.039   Mean   : 3.612   Sohail Tanvir  :  1264   MJ Guptill     :  1429   Umar Akmal     :  1446
 3rd Qu.:429.0   Australia   :10620   3rd Qu.:14.000   3rd Qu.: 5.000   KMDN Kulasekara:  1259   Shoaib Malik   :  1421   Mohammad Hafeez:  1388
 Max.   :569.0   England     :10618   Max.   :19.000   Max.   :11.000   Umar Gul       :  1224   Umar Akmal     :  1411   MJ Guptill     :  1343
                 (Other)     :64245                                     (Other)        :122472   (Other)        :122448   (Other)        :122359
   TotalRuns       BatterRuns       WicketType                PlayerOut
 Min.   :0.000   Min.   :0.000   0         :124090   0               :124090
 1st Qu.:0.000   1st Qu.:0.000   caught    :  4059   Shahid Afridi   :    75
 Median :1.000   Median :1.000   bowled    :  1481   Mohammad Hafeez:    71
 Mean   :1.233   Mean   :1.163   run out   :   681   TM Dilshan      :    65
 3rd Qu.:1.000   3rd Qu.:1.000   lbw       :   528   Umar Akmal      :    64
 Max.   :8.000   Max.   :7.000   stumped   :   277   DA Warner       :    62
                                 (Other)   :   205   (Other)         :  6894
> |
```

...

Making sense of strange features...

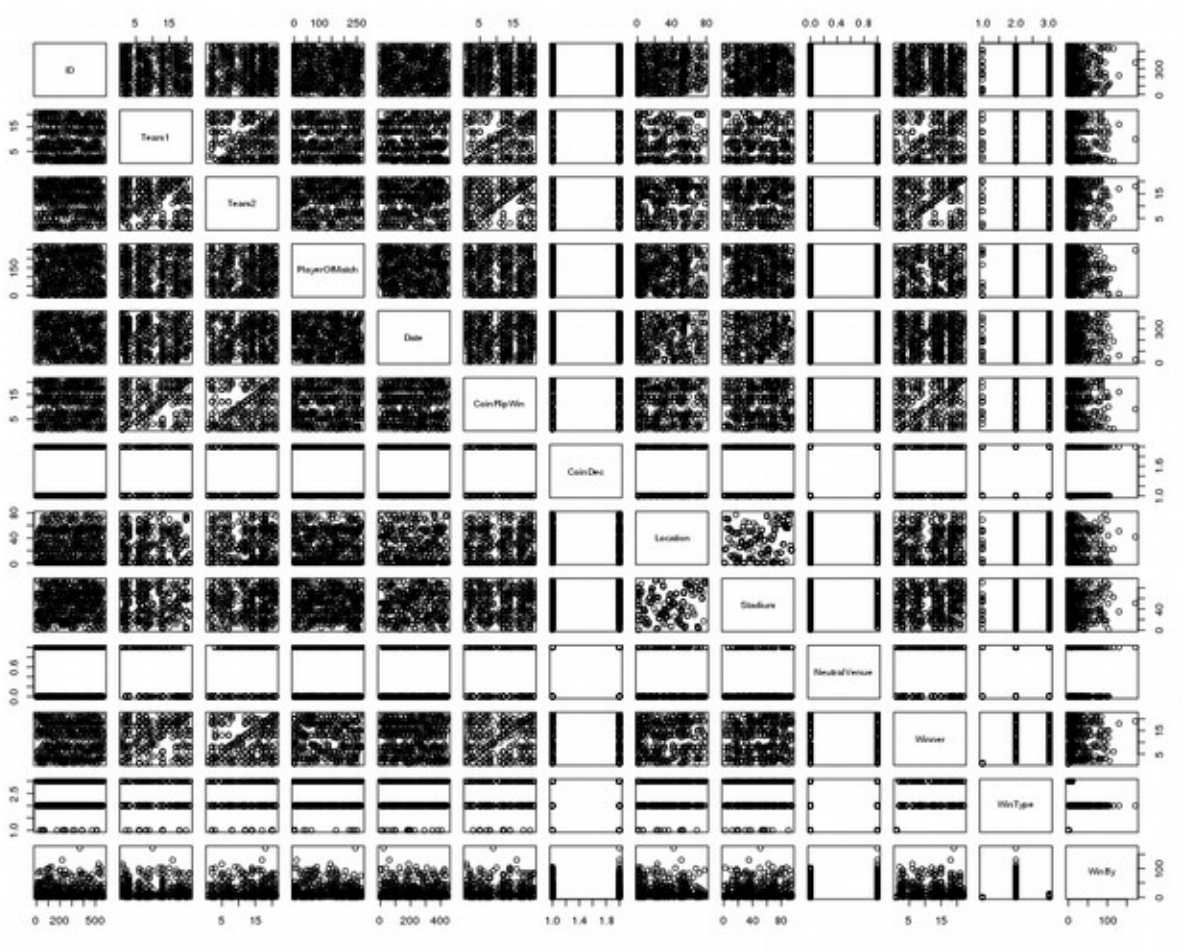
```
23
24 par(mar=c(10,10,4,2))
25 total.matches <- table(as.factor(c(as.character(matches$Team1),as.character(matches$Team2))))
26 barplot(sort(total.matches),cex.lab=.75,las=1,col="darkred",horiz=T)
27
```



plot(x)

Try this on something else...

```
28  plot(matches)
```



RECAP

Where to start
str(x)
summary(x)

Exploratory chart functions
barplot(x)
boxplot(x)
hist(x)
plot(x)

What are you looking for?
correlations
limits
general “feel” for the data