Input Example Code

Examples

- Mannheim wireless data
- Enron email summary information
  (Mark's example lab)
- Ad hoc longitudinal data
  Name, age, SSN
    v11, v12, v13, ., v15, ...
    v21, v22, ., v24, v25

  with variable number of observations per person.

Data

# timestamp=2006-02-11 22:14:37
# usec=250
# minReadings=110
t=1139692477303;id=00:02:2D:21:0F:
33;pos=0.0,0.05,0.0;degree=130.5;00:14:bf:b1:97:8a=-43,2437000000,3;00:0f:a3:39:e1:c0=-52,2462000000,3;00:14:bf:
3b:c7:c6=-62,2432000000,3;00:14:bf:b1:97:81=-58,2422000000,3;00:14:bf:b1:97:8d=-62,2442000000,3;00:14:bf:b1:97:90=-57,2427000000,3;00:0f:a
3:39:e0:4b=-79,2462000000,3;00:0f:a3:39:e2:10=-88,2437000000,3;00:0f:a3:39:dd:cd=-64,2412000000,3;02:64:fb:
68:52:e6=-87,2447000000,1;02:00:42:55:31:00=-85,2457000000,1
t=1139692477555;id=00:02:2D:21:0F:
33;pos=0.0,0.05,0.0;degree=130.5;00:14:bf:b1:97:8a=-43,2437000000,3;00:14:bf:b1:97:8a=-43,2437000000,3;00:0f:a3:39:e1:c0=-52,2462000000,3;0
0:14:bf:b1:97:90=-57,2427000000,3;00:14:bf:b1:97:8d=-64,2442000000,3;00:0f:a3:39:e0:4b=-77,2462000000,3;00:0f:a3:39:dd:cd=-62,2412000000,3;
02:00:42:55:31:00=-85,2457000000,1;02:64:fb:68:52:e6=-88,2447000000,1
t=1139692477807;id=00:02:2D:21:0F:
33;pos=0.0,0.05,0.0;degree=130.5;00:0f:a3:39:e1:c0=-51,2462000000,3;00:14:bf:b1:97:90=-49,2427000000,3;00:14:bf:
3b:c7:c6=-62,2432000000,3;00:14:bf:b1:97:8a=-44,2437000000,3;00:14:bf:b1:97:81=-68,2422000000,3;00:0f:a3:39:e0:4b=-75,2462000000,3;00:0f:
a3:39:dd:cd=-66,2412000000,3;00:0f:a3:39:e2:10=-90,2437000000,3;02:00:42:55:31:00=-87,2457000000,1
t=1139692478059;id=00:02:2D:21:0F:
33;pos=0.0,0.05,0.0;degree=130.5;00:0f:a3:39:e1:c0=-52,2462000000,3;00:14:bf:b1:97:90=-58,2427000000,3;00:14:bf:b1:97:8a=-39,2437000000,3;0
0:14:bf:b1:97:8d=-63,2442000000,3;00:0f:a3:39:dd:cd=-66,2412000000,3;00:14:bf:
3b:c7:c6=-68,2432000000,3;00:0f:a3:39:e0:4b=-76,2462000000,3;00:0f:a3:39:e2:10=-88,2437000000,3;02:00:42:55:31:00=-85,2457000000,1;02:64
:fb:68:52:e6=-88,2447000000,1
t=1139692478311;id=00:02:2D:21:0F:
33;pos=0.0,0.05,0.0;degree=130.5;00:14:bf:b1:97:8a=-42,2437000000,3;00:0f:a3:39:e1:c0=-51,2462000000,3;00:14:bf:b1:97:81=-60,2422000000,3;00
:14:bf:
3b:c7:c6=-59,2432000000,3;00:14:bf:b1:97:8d=-64,2442000000,3;00:0f:a3:39:dd:cd=-61,2412000000,3;00:0f:a3:39:e0:4b=-76,2462000000,3;00:0f:a
3:39:e2:10=-90,2437000000,3
t=1139692478563;id=00:02:2D:21:0F:
33;pos=0.0,0.05,0.0;degree=130.5;00:14:bf:b1:97:90=-57,2427000000,3;00:0f:a3:39:e1:c0=-51,2462000000,3;00:14:bf:b1:97:8a=-43,2437000000,3;00
:14:bf:
3b:c7:c6=-62,2432000000,3;00:14:bf:b1:97:8d=-63,2442000000,3;00:0f:a3:39:dd:cd=-62,2412000000,3;00:0f:a3:39:e0:4b=-76,2462000000,3;00:14:
bf:b1:97:81=-57,2422000000,3;00:0f:a3:39:e2:10=-89,2437000000,3;02:00:42:55:31:00=-85,2457000000,1;02:64:fb:68:52:e6=-87,2447000000,1
t=1139692478819;id=00:02:2D:21:0F:
33;pos=0.0,0.05,0.0;degree=130.5;00:14:bf:b1:97:8a=-44,2437000000,3;00:0f:a3:39:e1:c0=-50,2462000000,3;00:14:bf:
3b:c7:c6=-62,2432000000,3;00:14:bf:b1:97:90=-56,2427000000,3;00:0f:a3:39:dd:cd=-61,2412000000,3;00:14:bf:b1:97:8d=-63,2442000000,3;00:0f:a
3:39:e0:4b=-76,2462000000,3;00:0f:a3:39:e2:10=-91,2437000000,3;02:00:42:55:31:00=-87,2457000000,1;02:64:fb:68:52:e6=-89,2447000000,1

Data

- Each record has a
  - time stamp (t=1139692477303)
  - own MAC address (id=00:02:2D:21:0F:33)
  - position (pos=0.0,0.5,0.0)
  - orientation (degree=130.5)
  - for each base station it can detect, measures
    - MAC address of the base station
    - signal strength
    - channel frequency
    - whether it is an access point or ad hoc network device.

- First off, txt = readLines(file)
- Remove the lines that start with #
  - Use regular expression
    txt = txt[- grep("^#", txt)]
  - or txt = txt[substring(txt, 1, 1) != "#"]
- Now split each line at the ; to get the tokens
  - records = strsplit(txt, ";")

- Consider simple record

  t=1139692477303;id=00:02:2D:21:0F:33;pos=0.0,0.05,0.0;degree=130.5;
  00:14:bf:b1:97:8a=-43,2437000000,3;00:0f:a3:39:e1:c0=-52,2462000000,3

- So 1 position, detected 2 other wireless devices

- Consider if we stacked the wireless device entries

  00:14:bf:b1:97:8a=-43,2437000000,3
  00:0f:a3:39:e1:c0=-52,2462000000,3

- Starting to look a lot more usual

- Let's turn this into a character matrix
  (turn the values into numbers later in one operation for all of the records)

00:14:bf:b1:97:8a=-43,2437000000,3
00:0f:a3:39:e1:c0=-52,2462000000,3

For each line, split the "words" separate by = and ,

w = strsplit(txt, "=|,") - regular expression

Turn into a 2 by 4 matrix with

MAC

matrix(unlist(w), , 4, byrow = TRUE)

- Write a function to do this
  (handle case where no wireless base stations)
- Use this function & lapply() on all the records to generate a list of * by 4 matrices.
  All different number of rows.

```
function(x) {
  macs = x[-(1:4)]   # drop the handheld device
  if(length(macs) == 0)
     return(matrix(NA, 0, 4))
  matrix(unlist(strsplit(macs, "=|,")), , 4, byrow = TRUE)
}
```

- Now, if we could "stack" these all together into a single matrix of 4 columns we'd <u>almost</u> be finished

- The rbind() (row bind) function is useful
  rbind(m1, m2, m3)

- But we have a list of matrices, not 2.
  So we can't say rbind(list.of.matrices)

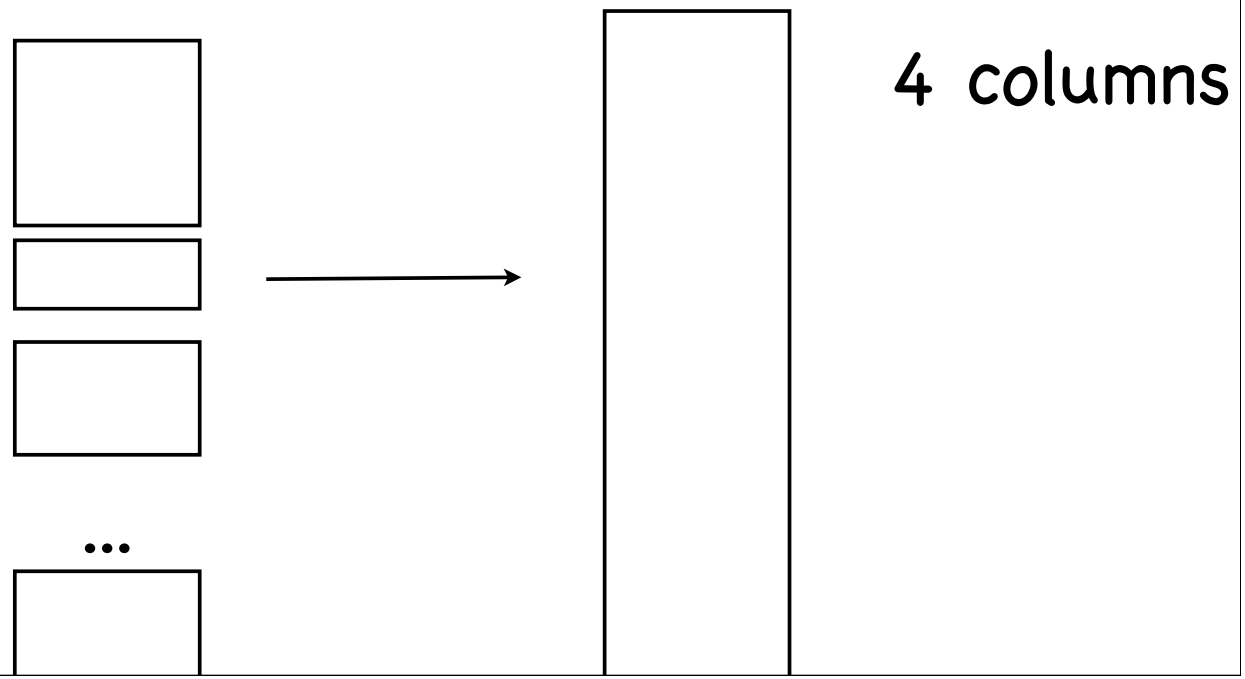
- We want R to do
  rbind(list.of.matrices[[1]], list.of.matrices[[2]], ...)

- Could loop, either concatenating or pre-allocating the matrix and inserting into particular rows

- So we use do.call(), a powerful function for these cases
- Give the name of the function to call and a list of the arguments.

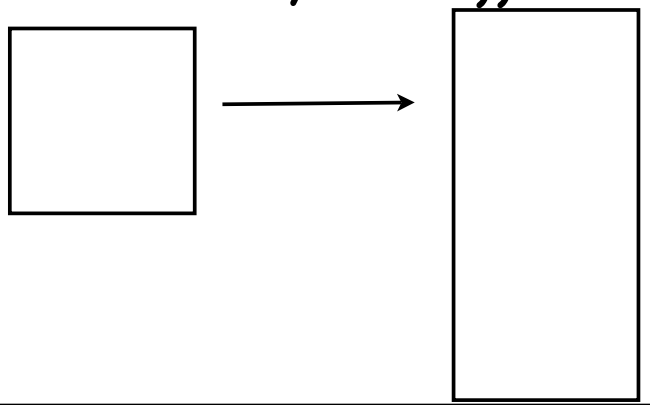
  Unravels 2nd argument as we wanted above.
- do.call("rbind", list.of.matrices)

and as.data.frame() and convert columns 2:4 to numbers



- We need to put the corresponding information for the hand-held device (position, degree, time) with each row in the data.frame.
- Each record now maps to multiple rows of data frame
- Record 1 corresponds to nrow(list.of.matrices[[1]]) rows
  2 . ... [[2]]
- Suppose we have a simple data.frame() of time, pos, degree in hand.held
  - hand.held[c(1, 1),] would match the corresponding 2 rows

- Usually, rep() repeats a vector a number of times
  rep(c(1, 2), 3) => c(1, 2, 1, 2, 1, 2)
- But rep(vector, vector) repeats element-wise
  rep(c(1, 2), c(3, 2)) => c(1, 1, 1, 2, 2)
- i = rep(1:nrow(hand.held),
  sapply(list.of.matrices, nrow))
- hand.held[i]



- Got to do checks at each step of the way
  - compute summary statistics of number of elements, number of records, etc. and plot, etc.

Additional Input Exercises

- State of the Union speeches
  - Break the data into speeches, words, president, year
  - EDA
  - Multi-dimensional scaling to see "proximity" over time and/or party affiliation

NASA geographic & atmospheric data

- Data Expo from JSM '06 organized by Paul Murrell
- http://stat-computing.org/dataexpo/2006/
- Many files in different directories in slightly awkward format.

```
          VARIABLE : Mean high cloud amount (%)
           FILENAME : ISCCPMonthly_avg.nc
           FILEPATH : /usr/local/fer_data/data/
           SUBSET   : 24 by 24 points (LONGITUDE-LATITUDE)
           TIME     : 16-JAN-1995 00:00
            113.8W 111.2W 108.8W 106.2W 103.8W 101.2W 98.8W  ........
              27     28     29     30     31     32     33    .........
36.2N / 51:  26.00  23.00  23.00  17.00  19.50  17.00  16.00  ..........
33.8N / 50:  20.00  20.00  18.50  16.50  18.00  15.00  15.00  ...........
31.2N / 49:  16.00  16.00  14.00  12.50   ..............................
```

- Explore "important features of the data"!