Simple models

Deepayan Sarkar

Indian Statistical Institute, Delhi

Data structures for data analysis

Basic types of data

- Atomic Vectors
  - Numeric
  - Categorical (factor)
  - Character
  - Logical
- Lists: vectors with arbitrary components

Basic types of data: examples

```
> month.name # built-in

 [1] "January"   "February"  "March"     "April"
 [5] "May"       "June"      "July"      "August"
 [9] "September" "October"   "November"  "December"

> x <- rnorm(10)
> x

 [1]  0.1804841  0.8820482  0.9350085  0.2864500  0.3395899
 [6] -0.4924313  0.5290983 -0.5975911  1.4143346 -0.8129160
```

Basic types of data: examples

```
> month.name # built-in

 [1] "January"   "February"  "March"     "April"
 [5] "May"       "June"      "July"      "August"
 [9] "September" "October"   "November"  "December"

> x <- rnorm(10)
> x

 [1]  0.1804841  0.8820482  0.9350085  0.2864500  0.3395899
 [6] -0.4924313  0.5290983 -0.5975911  1.4143346 -0.8129160

> str(x) # useful function

 num [1:10] 0.18 0.882 0.935 0.286 0.34 ...

> str(month.name)

 chr [1:12] "January" "February" "March" ...
```

Basic types of data: examples

```
> m <- sample(1:12, 30, rep = TRUE)
> m

 [1] 12  8  1 11  8 11  8  3  1 12  1  3  2  4  1  4  4  7
[19]  1  5  9  4  4  2  2  2  9  8  4  8

> mf <- factor(m, levels = 1:12, labels = month.name)
> mf

 [1] December  August    January   November  August
 [6] November  August    March     January   December
[11] January   March     February  April     January
[16] April     April     July      January   May
[21] September April     April     February  February
[26] February  September August    April     August
12 Levels: January February March April May June ... December

> str(m)

 int [1:30] 12 8 1 11 8 11 8 3 1 12 ...

> str(mf)

 Factor w/ 12 levels "January","February",..: 12 8 1 11 8 11 8 3 1 12 ...
```

Basic types of data: examples

```
> ml <- list(m = m, mf = mf)
> str(ml)
```

```
List of 2
 $ m : int [1:30] 12 8 1 11 8 11 8 3 1 12 ...
 $ mf: Factor w/ 12 levels "January","February",..: 12 8 1 11 8 11 8 3 1 12
```

Basic types of data: examples

```
> ml <- list(m = m, mf = mf)
> str(ml)

List of 2
 $ m : int [1:30] 12 8 1 11 8 11 8 3 1 12 ...
 $ mf: Factor w/ 12 levels "January","February",..: 12 8 1 11 8 11 8 3 1 12

> ml$m

 [1] 12  8  1 11  8 11  8  3  1 12  1  3  2  4  1  4  4  7
[19]  1  5  9  4  4  2  2  2  9  8  4  8
```

Basic types of data: examples

```
> ml <- list(m = m, mf = mf)
> str(ml)

List of 2
 $ m : int [1:30] 12 8 1 11 8 11 8 3 1 12 ...
 $ mf: Factor w/ 12 levels "January","February",..: 12 8 1 11 8 11 8 3 1 12

> ml$m

 [1] 12  8  1 11  8 11  8  3  1 12  1  3  2  4  1  4  4  7
[19]  1  5  9  4  4  2  2  2  9  8  4  8

> ml[["mf"]]

 [1] December  August    January   November  August
 [6] November  August    March     January   December
[11] January   March     February  April     January
[16] April     April     July      January   May
[21] September April     April     February  February
[26] February  September August    April     August
12 Levels: January February March April May June ... December
```

Most common structures for statistical data

Vectors, matrices / arrays: vectors with dimension

```
> VADeaths

      Rural Male Rural Female Urban Male Urban Female
50-54       11.7          8.7       15.4          8.4
55-59       18.1         11.7       24.3         13.6
60-64       26.9         20.3       37.0         19.3
65-69       41.0         30.9       54.6         35.1
70-74       66.0         54.3       71.1         50.0

> dim(VADeaths)

[1] 5 4
```

Most common structures for statistical data

Data frames: lists that also behave like a matrix

```
> str(iris)

'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1

> head(iris)

  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

Data input / output

- Statistical data are usually structured like a spreadsheet (e.g., Excel)
- Typical approach: read data from spreadsheet file into data frame
- Easiest route:
    - R itself cannot read Excel files directly
    - Save as CSV file from Excel
    - Read with `read.csv()` or `read.table()` (more flexible)
- Alternative option:
    - Use "Import Dataset" menu item in R Studio (requires add-on package)

- Data frames can be exported as a spreadsheet file using `write.csv()` or `write.table()`

```
> data(Cars93, package = "MASS")
> write.csv(Cars93, file = "cars93.csv") # export
> cars <- read.csv("cars93.csv") # import
```

Basic statistical problems

Steps in a typical data analysis problem

- Formulate purpose of the analysis, e.g.,
  - prediction
  - testing / identifying important variables

Steps in a typical data analysis problem

- Formulate purpose of the analysis, e.g.,
  - prediction
  - testing / identifying important variables
- Build model
- Check and refine model
- Use model for further insight

Types of data

- Categorical
- Numeric (continuous)

Types of data

- Categorical
- Numeric (continuous)

- Also discrete numeric (e.g., count data)

Data roles

- Response or outcome variable
- Predictors or explanatory variable

Predictor	Response	Problem type
Categorical	Numeric	t-test, ANOVA (testing)
Numeric	Numeric	Regression (prediction, testing)
Categorical	Categerical	Test of independence (testing)
Either	Categerical	Classification (prediction)

"Regression" often refers to the general class of problems with a continuous response.

Examples

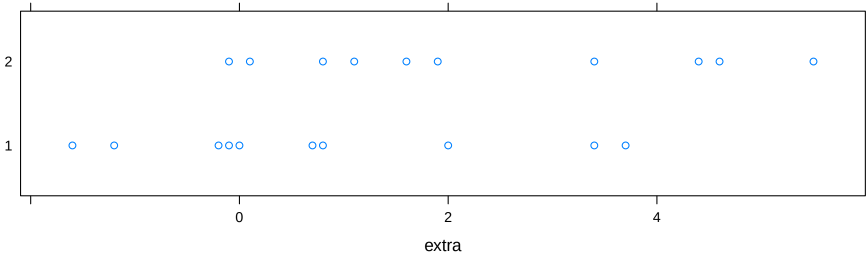
Example: sleep data

Amount of extra sleep (in hours) after taking three sleep-inducing drugs

```
> sleep
   extra group ID
1    0.7     1  1
2   -1.6     1  2
3   -0.2     1  3
4   -1.2     1  4
5   -0.1     1  5
6    3.4     1  6
7    3.7     1  7
8    0.8     1  8
9    0.0     1  9
10   2.0     1 10
11   1.9     2  1
12   0.8     2  2
13   1.1     2  3
14   0.1     2  4
15  -0.1     2  5
16   4.4     2  6
17   5.5     2  7
18   1.6     2  8
```

First step: plot the data

```
> library(lattice)
> stripplot(group ~ extra, data = sleep)
```



Possible questions:

- Do the drugs work?
- Is one of the drugs more effective than the other?

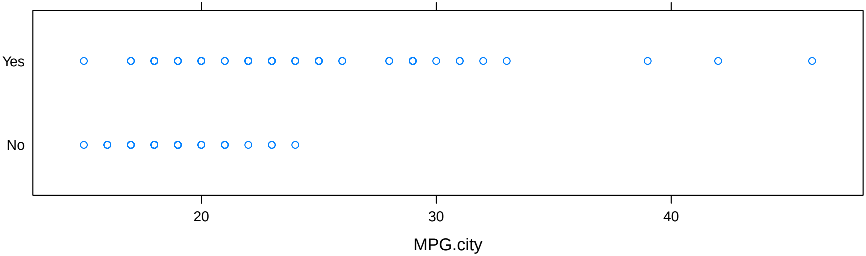
Example: 1993 passenger car models

```
> data(Cars93, package = "MASS")
> str(Cars93)
```

```
'data.frame':   93 obs. of  27 variables:
 $ Manufacturer      : Factor w/ 32 levels "Acura","Audi",..: 1 1 2 2 3 4 4
 $ Model             : Factor w/ 93 levels "100","190E","240",..: 49 56 9 1
 $ Type              : Factor w/ 6 levels "Compact","Large",..: 4 3 1 3 3 3
 $ Min.Price         : num  12.9 29.2 25.9 30.8 23.7 14.2 19.9 22.6 26.3 33
 $ Price             : num  15.9 33.9 29.1 37.7 30 15.7 20.8 23.7 26.3 34.7
 $ Max.Price         : num  18.8 38.7 32.3 44.6 36.2 17.3 21.7 24.9 26.3 36
 $ MPG.city          : int  25 18 20 19 22 22 19 16 19 16 ...
 $ MPG.highway       : int  31 25 26 26 30 31 28 25 27 25 ...
 $ AirBags           : Factor w/ 3 levels "Driver & Passenger",..: 3 1 2 1
 $ DriveTrain        : Factor w/ 3 levels "4WD","Front",..: 2 2 2 2 3 2 2 3
 $ Cylinders         : Factor w/ 6 levels "3","4","5","6",..: 2 4 4 4 2 2 4
 $ EngineSize        : num  1.8 3.2 2.8 2.8 3.5 2.2 3.8 5.7 3.8 4.9 ...
 $ Horsepower        : int  140 200 172 172 208 110 170 180 170 200 ...
 $ RPM               : int  6300 5500 5500 5500 5700 5200 4800 4000 4800 41
 $ Rev.per.mile      : int  2890 2335 2280 2535 2545 2565 1570 1320 1690 15
 $ Man.trans.avail   : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 1 1 1 1 .
 $ Fuel.tank.capacity: num  13.2 18 16.9 21.1 21.1 16.4 18 23 18.8 18 ...
 $ Passengers        : int  5 5 5 6 4 6 6 6 5 6 ...
```

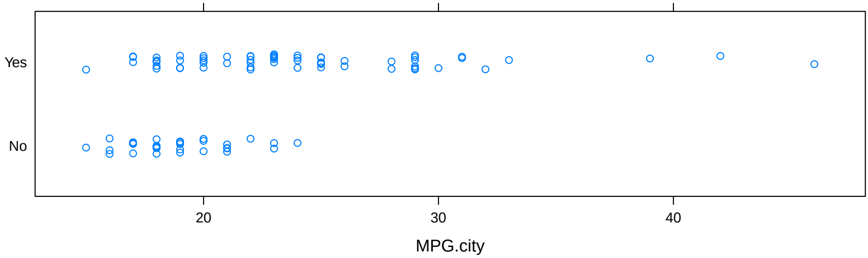
Plot

```
> stripplot(Man.trans.avail ~ MPG.city, data = Cars93)
```



- Are manual transmission cars more fuel efficient?

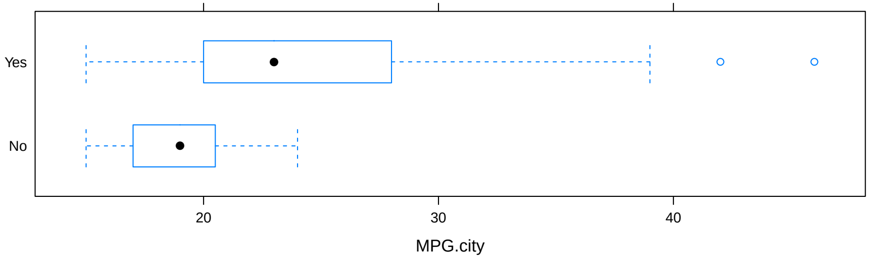
```
> stripplot(Man.trans.avail ~ MPG.city, data = Cars93, jitter = TRUE)
```



- Are manual transmission cars more fuel efficient?

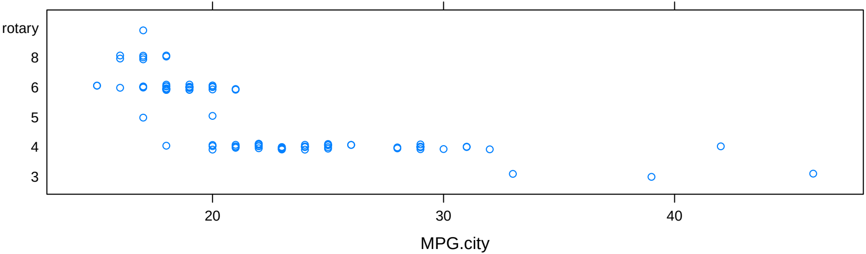
Plot

```
> bwplot(Man.trans.avail ~ MPG.city, data = Cars93)
```



- Are manual transmission cars more fuel efficient?

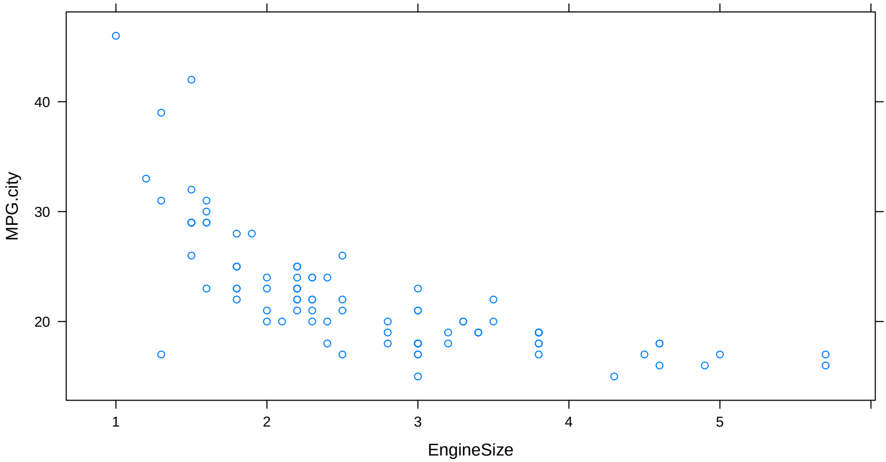
```
> stripplot(Cylinders ~ MPG.city, data = Cars93, jitter = TRUE)
```



- Does fuel efficiency depend on number of cylinders?

Plot

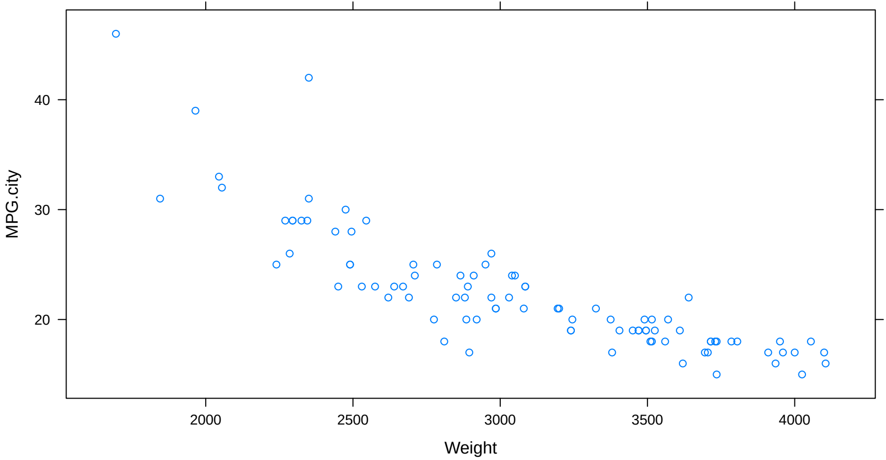
```
> xyplot(MPG.city ~ EngineSize, data = Cars93)
```



- Does fuel efficiency depend on engine size?

Plot

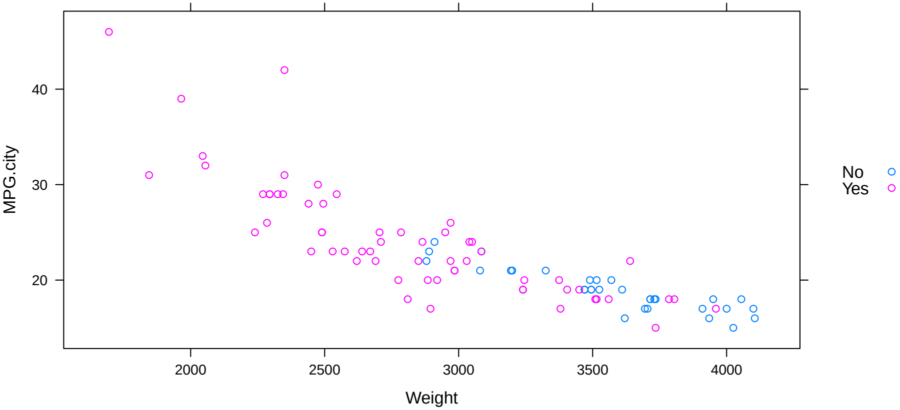
```
> xyplot(MPG.city ~ Weight, data = Cars93)
```



- Does fuel efficiency depend on weight?

Plot

```
> xyplot(MPG.city ~ Weight, data = Cars93, groups = Man.trans.avail,
+        auto.key = list(space = "right"))
```



- How does dependence on weight vary with manual transmission?

Fitting models in R (demo)

- Two-sample comparison (categorical vs categorical)
  - Test of independence
  - χ^2-test
  - Permutation test
- Two-sample comparisons:
  - Nonparametric (rank-sum test)
  - Two-sample t-test
  - Permutation test?
- Multi-sample comparisons: ANOVA
- Regression