Data Frames

R's usual data table object.

- Library of example datasets:

```
> data()
Data sets in package 'base':

Formaldehyde    Determination of Formaldehyde
HairEyeColor    Hair and Eye Color of Statistics Students
[ . . . ]

> class(Formaldehyde)
[1] "data.frame"
> help(Formaldehyde)
[ . . . ]
> Formaldehyde
  carb optden
1  0.1  0.086
2  0.3  0.269
3  0.5  0.446
4  0.6  0.538
5  0.7  0.626
6  0.9  0.782
> summary(Formaldehyde)
      carb            optden
 Min.   :0.1000   Min.   :0.0860
 1st Qu.:0.3500   1st Qu.:0.3132
 Median :0.5500   Median :0.4920
 Mean   :0.5167   Mean   :0.4578
 3rd Qu.:0.6750   3rd Qu.:0.6040
 Max.   :0.9000   Max.   :0.7820
>
```


Data Frames

- Building them manually:

```
> people <- data.frame(age=c(24,20,18,20,19),
                       gender=c("M","F","F","M","F"),
                       height=c(61, 55, 52, 57, 57))
> people
  age gender height
1  24      M     61
2  20      F     55
3  18      F     52
4  20      M     57
5  19      F     57
> class(people$gender)
[1] "factor"
> row.names(people)
[1] "1" "2" "3" "4" "5"
> row.names(people) <- c("Stan","Ruojia","Sam",
                         "Ahmed","Felicia")
> people
        age gender height
Stan     24      M     61
Ruojia   20      F     55
Sam      18      F     52
Ahmed    20      M     57
Felicia  19      F     57
>
```


Data Frames: Using Them

- Summarizing:

```
> summary(people)
      age       gender     height
 Min.   :18.0   F:3    Min.   :52.0
 1st Qu.:19.0   M:2    1st Qu.:55.0
 Median :20.0          Median :57.0
 Mean   :20.2          Mean   :56.4
 3rd Qu.:20.0          3rd Qu.:57.0
 Max.   :24.0          Max.   :61.0
>
```

- Selecting parts:

```
> people
        age gender height
Stan     24      M     61
Ruojia   20      F     55
Sam      18      F     52
Ahmed    20      M     57
Felicia  19      F     57
> people$age
[1] 24 20 18 20 19
> people[[2]]
[1] M F F M F
Levels: F M
> people[4,1]
[1] 20
> people[5,]
        age gender height
Felicia  19      F     57
> people[people$age >= 20 & people$gender == "M",
         c("age","height")]
      age height
Stan   24     61
Ahmed  20     57
>
```


Data Frames: Using Them

- Using `attach` and `detach`:

```
> age
Error: Object "age" not found
> attach(people)
> age
[1] 24 20 18 20 19
> gender
[1] M F F M F
Levels: F M
> table(gender)
gender
F M
3 2
> tapply(age, gender, mean)
 F  M
19 22
> detach()
> age
Error: Object "age" not found
>
```

- Building models:

```
> names(people)
[1] "age"    "gender" "height"
> l <- lm(height ~ age + gender, data=people)
> summary(l)
[ . . . ]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.7667    12.4922   2.703    0.114
age           1.1000     0.6545   1.681    0.235
genderM       1.0333     2.7248   0.379    0.741
[ . . . ]
>
```

Getting Data into R

- From the keyboard: `c()` or `scan()`

```
> colour <- c("red", "blue", "yellow", "green")
> preference <- scan()
1: 10 2 8 6
5:
Read 4 items
> colour.type <- factor(scan(what=""))
1: primary primary primary secondary
5:
Read 4 items
> colours <- data.frame(colour=colour,
                        preference=preference,
                        type=colour.type)
> colours
  colour preference      type
1    red         10   primary
2   blue          2   primary
3 yellow          8   primary
4  green          6 secondary
>
```

- Interactively with `fix` or `edit`:

```
> fix(colours)
```

or

```
> new.colours <- edit(colours)
```


Getting Data into R

- From files: using `scan(...)`

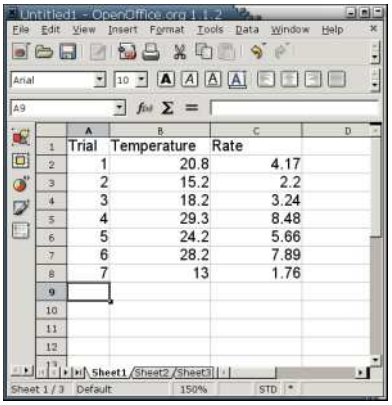
```
> system("cat numbers")
10 15 15
17 21 10.3 -8
> scan("numbers")
Read 7 items
[1] 10.0 15.0 15.0 17.0 21.0 10.3 -8.0
> system("cat strings")
one
string per
line
> scan("strings", sep="\n")
Error in scan("strings", sep = "\n") :
  "scan" expected a real, got "one"
> scan("strings", what="example", sep="\n")
Read 3 items
[1] "one"        "string per" "line"
>
```


Getting Data into R

- From files: using `read.table(...)`

```
> system("cat table")
day high.temp low.temp snowfall
Fri    0         -2       5
Sat    0         -4       2
Sun    1         -4       0
Mon    1         -1       0
> forecast <- read.table("table", header=T)
> forecast
  day high.temp low.temp snowfall
1 Fri         0       -2        5
2 Sat         0       -4        2
3 Sun         1       -4        0
4 Mon         1       -1        0
> class(forecast$day)
[1] "factor"
>
```


Exchanging Data with Spreadsheet



1. Save as a CSV (Comma-Separated Value) file.

```
> system("cat experiment.csv")
"Trial","Temperature","Rate"
1,20.8,4.17
2,15.2,2.2
3,18.2,3.24
4,29.3,8.48
5,24.2,5.66
6,28.2,7.89
7,13,1.76
>
```

Exchanging Data with Spreadsheet

2. Read using `read.csv`

```
> experiment <- read.csv("experiment.csv")
> experiment
  Trial Temperature Rate
1     1        20.8 4.17
2     2        15.2 2.20
3     3        18.2 3.24
4     4        29.3 8.48
5     5        24.2 5.66
6     6        28.2 7.89
7     7        13.0 1.76
>
```

3. Write using `write.table`

```
> experiment$Temperature <- 1.8*experiment$Temperature+32
> write.table(experiment, "newfile.csv", sep=",", col.names=NA)
>
```

See `help(write.table)` for details and examples.

Working with Columns

- adding, removing, and transforming:

```
> people
        age gender height
Stan     24      M     61
Ruojia   20      F     55
Sam      18      F     52
Ahmed    20      M     57
Felicia  19      F     57
> names(people)
[1] "age"    "gender" "height"
> names(people)[3] <- "height.inches"
> people$height.cms <- people$height.inches * 2.54
> people$age <- people$age + 0.5
> people$gender <- NULL
> people
         age height.inches height.cms
Stan    24.5            61     154.94
Ruojia  20.5            55     139.70
Sam     18.5            52     132.08
Ahmed   20.5            57     144.78
Felicia 19.5            57     144.78
>
```


Working with Columns

- acting on several at once:

```
> iris
  Sepal.Len Sepal.Wid Petal.Len Petal.Wid Species
1       5.1       3.5       1.4       0.2  setosa
2       4.9       3.0       1.4       0.2  setosa
[ . . . ]
> iris[,1:4] <- iris[,1:4]*10  # convert cm to mm
> iris
  Sepal.Len Sepal.Wid Petal.Len Petal.Wid Species
1        51        35        14         2  setosa
2        49        30        14         2  setosa
[ . . . ]
> mean(iris[,1:4])
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
    58.43333     30.57333     37.58000     11.99333
> hist(iris[,1:4])
Error in hist.default(iris[, 1:4]) : ‘x’ must be numeric
> par(ask=T); lapply(iris[,1:4], hist)
Hit <Return> to see next plot:
[ . . . ]
>
```