Fundamentals of R: Manipulating Dataframes

“selecting rows and columns”
and More !

Dataframe Topics

- What are Dataframes?
- Selecting Rows at Random.
- Using Logical Conditions to Select Rows.
- Dealing with Missing Data.
- Using Row Names Instead of Row Numbers.
- Creating Dataframes from Other Object Types.
- Eliminating Duplicate Rows.
- Using the **`match()`** Function.
- Merging Two Dataframes.
- Adding Margins.
- Summarizing the Contents.

Dataframes

- A **dataframe** is an object with rows and columns.
- **Columns** represent ***variables***.
- **Rows** represent ***observations***.
- **Response variables** must all be in one column (stacked).

Columns represent variables

```
> head(worms)
       Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1     Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
2  Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
3   Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
4     Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5 Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
6         Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
```

Rows represent observations, or cases

Importing Data

- Can use **`read.table()`** or **`read.csv()`** functions.
- **`attach()`** makes the variables accessible by name.
- **`names()`** returns the variable (columns) names.

```
> worms <- read.table("c:\\temp\\worms.txt",header=T)

> attach(worms)

> names(worms)
[1] "Field.Name"    "Area"          "Slope"         "Vegetation"
[5] "Soil.pH"       "Damp"          "Worm.density"
```

Subscript of first value on the line “Soil.pH” in vector of variable names

Subscripts and Indices

- Key to working with dataframes is to understand working with **subscripts** (sometimes called ***indices***):
- ***Rows*** referred to with left subscript, ***columns*** with right:

```
# fetches value of Soil.pH (column 5) in row 3:
> worms[3,5]
[1] 4.3

# use colon to generate a series of subscripts:
> worms[14:19,7]
[1] 0 6 8 4 5 1

# to fetch Area and Slope from rows from rows 1 to 5:
> worms[1:5,2:3]
  Area Slope
1  3.6    11
2  5.1     2
3  2.8     3
4  2.4     5
5  3.8     0
```

Subscripts and Indices

- To select **all entries** in a:
  - ***row***, syntax is ‘number comma blank’
  - ***column***, syntax is ‘blank comma number’.

```
# to select all entries in row 3:
> worms[3,]
     Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
3 Nursery.Field  2.8     3  Grassland     4.3 FALSE            2

# to select all entries in column 3:
> worms[,3]
 [1] 11  2  3  5  0  2  3  0  0  4 10  1  2  6  0  0  8  2  1 10
```

Subscripts and Indices

- Note that these two commands *create objects of different classes*:

```
> class(worms[3,])
[1] "data.frame"

> class(worms[,3])
[1] "integer"
```

Subscripts and Indices

- Use subscripts to create **sets** of rows and/or columns:

```
> worms[,c(1,5)]
          Field.Name Soil.pH
1        Nashs.Field     4.1
2     Silwood.Bottom     5.2
3      Nursery.Field     4.3
4        Rush.Meadow     4.9
5    Gunness.Thicket     4.2
6           Oak.Mead     3.9
7       Church.Field     4.2
8            Ashurst     4.8
9        The.Orchard     5.7
10     Rookery.Slope     5.0
11       Garden.Wood     5.2
12      North.Gravel     4.1
13      South.Gravel     4.0
14 Observatory.Ridge     3.8
```

Selecting Rows at Random

- Use **`sample()`** function, default **`replace = F`** to select a unique 8 of the 20 rows at random:

```
> worms[sample(1:20,8),]
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
8             Ashurst  2.1     0     Arable     4.8 FALSE            4
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
```

Sorting Dataframes

- Use **`order()`** function to sort by rows on basis of values in one column:

Ties based on Slope = 0 in their original order

```
> worms[order(Slope),]
```

	Field.Name	Area	Slope	Vegetation	Soil.pH	Damp	Worm.density
5	Gunness.Thicket	3.8	0	Scrub	4.2	FALSE	6
8	Ashurst	2.1	0	Arable	4.8	FALSE	4
9	The.Orchard	1.9	0	Orchard	5.7	FALSE	9
15	Pond.Field	4.1	0	Meadow	5.0	TRUE	6
16	Water.Meadow	3.9	0	Meadow	4.9	TRUE	8
12	North.Gravel	3.3	1	Grassland	4.1	FALSE	1
19	Gravel.Pit	2.9	1	Grassland	3.5	FALSE	1
2	Silwood.Bottom	5.1	2	Arable	5.2	FALSE	7
6	Oak.Mead	3.1	2	Grassland	3.9	FALSE	2
13	South.Gravel	3.7	2	Grassland	4.0	FALSE	2
18	Pound.Hill	4.4	2	Arable	4.5	FALSE	5
3	Nursery.Field	2.8	3	Grassland	4.3	FALSE	2

Sorting Dataframes

- Use **`rev()`** function outside the **`order()`** function to reverse the order:

```
> worms[rev(order(Slope)),]
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
```


Ties based on Slope also in original reversed order

Sorting Dataframes by Two Variables

- Separate series of variable names by commas within the **`order()`** function:

```
> worms[order(Vegetation,Worm.density),]
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
8             Ashurst  2.1     0     Arable     4.8 FALSE            4
18         Pound.Hill  4.4     2     Arable     4.5 FALSE            5
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
19         Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
```

Sorting Dataframes by Three Variables

- Separate three variable names by commas within the **`order()`** function:

```
> worms[order(Vegetation,Worm.density,Soil.pH),]
         Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
```

Sorting Dataframes with Select Columns

- Sort dataframe with select columns:

```
> worms[order(Vegetation,Worm.density),c(4,7,5,3)]
   Vegetation Worm.density Soil.pH Slope
8      Arable             4     4.8     0
18     Arable             5     4.5     2
2      Arable             7     5.2     2
14  Grassland             0     3.8     6
12  Grassland             1     4.1     1
19  Grassland             1     3.5     1
3   Grassland             2     4.3     3
6   Grassland             2     3.9     2
13  Grassland             2     4.0     2
7   Grassland             3     4.2     3
1   Grassland             4     4.1    11
10  Grassland             7     5.0     4
4      Meadow             5     4.9     5
```

Using Logical Conditions to Select Rows

- Select rows with damp fields.
- Syntax for logical subscripts is: [‘which rows’, blank].

T is not enclosed in quotes (e.g. ‘T’)

```
> worms[Damp==T,]
      Field.Name Area Slope Vegetation Soil.pH Damp Worm.density
4    Rush.Meadow  2.4     5     Meadow     4.9 TRUE            5
10 Rookery.Slope  1.5     4  Grassland     5.0 TRUE            7
15    Pond.Field  4.1     0     Meadow     5.0 TRUE            6
16  Water.Meadow  3.9     0     Meadow     4.9 TRUE            8
17     Cheapside  2.2     8      Scrub     4.7 TRUE            4
20     Farm.Wood  0.8    10      Scrub     5.1 TRUE            3
```

Using Logical Conditions to Select Rows

- Want rows where **`Worm.density`** is higher than the median and **`Soil.pH`** was less than 5.2:

```
> worms[Worm.density > median(Worm.density)& Soil.pH < 5.2,]
        Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
4      Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5  Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
10   Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
15      Pond.Field  4.1     0     Meadow     5.0  TRUE            6
16    Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
18      Pound.Hill  4.4     2     Arable     4.5 FALSE            5
```

Using Logical Conditions to Select Rows

- Only want columns that contain numbers:

```
> worms[,sapply(worms,is.numeric)]
   Area Slope Soil.pH Worm.density
1   3.6    11     4.1            4
2   5.1     2     5.2            7
3   2.8     3     4.3            2
4   2.4     5     4.9            5
5   3.8     0     4.2            6
6   3.1     2     3.9            2
7   3.5     3     4.2            3
8   2.1     0     4.8            4
9   1.9     0     5.7            9
10  1.5     4     5.0            7
11  2.9    10     5.2            8
12  3.3     1     4.1            1
13  3.7     2     4.0            2
14  1.8     6     3.8            0
15  4.1     0     5.0            6
```

Using Logical Conditions to Select Rows

- Or perhaps only want columns that contain factors:

```
> worms[,sapply(worms,is.factor)]
          Field.Name Vegetation
1        Nashs.Field  Grassland
2     Silwood.Bottom     Arable
3      Nursery.Field  Grassland
4        Rush.Meadow     Meadow
5    Gunness.Thicket      Scrub
6           Oak.Mead  Grassland
7       Church.Field  Grassland
8            Ashurst     Arable
9        The.Orchard    Orchard
10     Rookery.Slope  Grassland
11       Garden.Wood      Scrub
12      North.Gravel  Grassland
13      South.Gravel  Grassland
14 Observatory.Ridge  Grassland
15        Pond.Field     Meadow
```

Use Negative Subscripts to Drop Rows

- To drop a row or rows, use **negative subscripts**:

```
> worms[-(6:15),]
        Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1      Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
2   Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
3    Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
4      Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5  Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
16    Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
17       Cheapside  2.2     8      Scrub     4.7  TRUE            4
18      Pound.Hill  4.4     2     Arable     4.5 FALSE            5
19      Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
20       Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

Use Logical Symbol ! NOT to Exclude Rows

- Select all rows that are **not** Grasslands (the logical symbol **!** means NOT):

```
> worms[!(Vegetation=="Grassland"),]
        Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
2   Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
4      Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5  Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
8          Ashurst  2.1     0     Arable     4.8 FALSE            4
9      The.Orchard  1.9     0    Orchard     5.7 FALSE            9
11     Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
15      Pond.Field  4.1     0     Meadow     5.0  TRUE            6
16    Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
17       Cheapside  2.2     8      Scrub     4.7  TRUE            4
18      Pound.Hill  4.4     2     Arable     4.5 FALSE            5
20       Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

Use `which()` Function to Exclude Rows

- Can use **`which()`** function coupled with minus sign.
- These statements are all equivalent:

```
> worms[!Damp==F,]
> worms[Damp==T,]
> worms[-which(Damp==F),]
      Field.Name Area Slope Vegetation Soil.pH Damp Worm.density
4    Rush.Meadow  2.4     5     Meadow     4.9 TRUE            5
10 Rookery.Slope  1.5     4  Grassland     5.0 TRUE            7
15    Pond.Field  4.1     0     Meadow     5.0 TRUE            6
16  Water.Meadow  3.9     0     Meadow     4.9 TRUE            8
17     Cheapside  2.2     8      Scrub     4.7 TRUE            4
20     Farm.Wood  0.8    10      Scrub     5.1 TRUE            3
```

Omitting Rows with Missing Data

- Read in sister **`worms.missing.txt`** dataframe (with missing data):

```
> data <- read.table("c:\\temp\\worms.missing.txt",header=T)
> data
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
2     Silwood.Bottom  5.1    NA     Arable     5.2 FALSE            7
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
7       Church.Field  3.5     3  Grassland      NA    NA           NA
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
```

Omitting Rows with Missing Data

- Omit rows with missing data:

```
> na.omit(data)
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE             4
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE             2
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE             5
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE             6
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE             2
8            Ashurst  2.1     0     Arable     4.8 FALSE             4
9        The.Orchard  1.9     0    Orchard     5.7 FALSE             9
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE             7
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE             8
12      North.Gravel  3.3     1  Grassland     4.1 FALSE             1
13      South.Gravel  3.7     2  Grassland     4.0 FALSE             2
```

Checking for Missing Data

- To check for missing data across rows:

```
> complete.cases(data)
 [1]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
[11]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
```

- What you want is to check individual cells and variables:

```
> apply(apply(data,2,is.na),2,sum)
  Field.Name         Area        Slope   Vegetation      Soil.pH         Damp Worm.density
           0            1            1            0            1            1            1

> apply(data,2,is.na)
      Field.Name  Area Slope Vegetation Soil.pH  Damp Worm.density
 [1,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
 [2,]      FALSE FALSE  TRUE      FALSE   FALSE FALSE        FALSE
 [3,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
 [4,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
 [5,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
 [6,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
```

Omitting Rows with Missing Data

- Equivalent command to **`na.omit(data)`**:

```
> data[complete.cases(data),]
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
```

Using `order()` and `!duplicated()` to Eliminate Pseudoreplication

- Want one record for each vegetation type which also has the greatest worm density:

```
> new <- worms[rev(order(Worm.density)),]

> !duplicated(new$Vegetation)
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
[11] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

> new[!duplicated(new$Vegetation),]
       Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
9     The.Orchard  1.9     0    Orchard     5.7 FALSE            9
16   Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
11    Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
10  Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
2  Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
```

Complex Ordering with Mixed Directions

- Want vegetation sorted a to z but worm density high to low:

```
> worms[order(Vegetation,-Worm.density),]
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
```

Complex Ordering with Mixed Directions

- Want vegetation sorted z to a but worm density high to low:

```
> worms[order(-rank(Vegetation),-Worm.density),]
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
11       Garden.Wood  2.9    10       Scrub     5.2 FALSE            8
5    Gunness.Thicket  3.8     0       Scrub     4.2 FALSE            6
17         Cheapside  2.2     8       Scrub     4.7  TRUE            4
20         Farm.Wood  0.8    10       Scrub     5.1  TRUE            3
9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
```

Selecting Columns by Logical Operations

- Want columns with ‘S’ (upper-case) in name:

```
> names(worms)
[1] "Field.Name"   "Area"          "Slope"         "Vegetation"
[5] "Soil.pH"      "Damp"          "Worm.density"
> grep("S",names(worms))
[1] 3 5
> worms[,grep("S",names(worms))]
   Slope Soil.pH
1     11     4.1
2      2     5.2
3      3     4.3
4      5     4.9
5      0     4.2
6      2     3.9
7      3     4.2
8      0     4.8
9      0     5.7
10     4     5.0
```

Using Row Names Instead of Row Numbers

- Want 1st column (names of fields) to be row names:

```
> detach(worms)
> worms <- read.table("c:\\temp\\worms.txt", header=T, row.names=1)
> worms
                  Area Slope Vegetation Soil.pH  Damp Worm.density
Nashs.Field        3.6    11  Grassland     4.1 FALSE            4
Silwood.Bottom     5.1     2     Arable     5.2 FALSE            7
Nursery.Field      2.8     3  Grassland     4.3 FALSE            2
Rush.Meadow        2.4     5     Meadow     4.9  TRUE            5
Gunness.Thicket    3.8     0      Scrub     4.2 FALSE            6
Oak.Mead           3.1     2  Grassland     3.9 FALSE            2
Church.Field       3.5     3  Grassland     4.2 FALSE            3
Ashurst            2.1     0     Arable     4.8 FALSE            4
The.Orchard        1.9     0    Orchard     5.7 FALSE            9
Rookery.Slope      1.5     4  Grassland     5.0  TRUE            7
Garden.Wood        2.9    10      Scrub     5.2 FALSE            8
North.Gravel       3.3     1  Grassland     4.1 FALSE            1
South.Gravel       3.7     2  Grassland     4.0 FALSE            2
```

Creating Dataframes From Other Object Types

- Bind (and coerce) three vectors into dataframe:

```
> x <- runif(10)
> y <- letters[1:10]
> z <- sample (c(rep(T,5), rep(F,5)))
> new <- data.frame(y,z,x); new
   y     z          x
1  a  TRUE 0.65006333
2  b  TRUE 0.42238742
3  c FALSE 0.01974397
4  d  TRUE 0.49122287
5  e FALSE 0.51144160
6  f FALSE 0.04860806
7  g FALSE 0.27856008
8  h FALSE 0.69728709
9  i  TRUE 0.24273183
10 j  TRUE 0.05636418
```

Creating Dataframes From Other Object Types

- Create Poisson count in table and convert to dataframe:

```
> y <- rpois(1500,1.5)
> table(y); y
y
  0   1   2   3   4   5   6   7
313 515 387 186  68  22   8   1
> as.data.frame(table(y))
  y Freq
1 0  313
2 1  515
3 2  387
4 3  186
5 4   68
6 5   22
7 6    8
8 7    1
```

Eliminating Duplicate Rows

- Can use **`unique()`** function to strip out duplicate rows:

```
> dups <- read.table("c:\\temp\\dups.txt",header=T); dups
  cow dog cat bat
1   1   2   3   1
2   1   2   2   1
3   3   2   1   1
4   4   4   2   1
5   3   2   1   1
6   6   1   2   5
7   1   2   3   2
> unique(dups)
  cow dog cat bat
1   1   2   3   1
2   1   2   2   1
3   3   2   1   1
4   4   4   2   1
6   6   1   2   5
7   1   2   3   2
```

Using the `match()` Function

- **`worms`** contains five unique vegetation types:

```
> unique(worms$Vegetation)
[1] Grassland Arable    Meadow    Scrub     Orchard
Levels: Arable Grassland Meadow Orchard Scrub
```

- We want to know appropriate herbicides to use in each of 20 fields:

```
> herbicides <- read.table("c:\\temp\\herbicides.txt",header=T)
> herbicides
        Type Herbicide
1   Woodland  Fusilade
2    Conifer  Weedwipe
3     Arable  Twinspan
4       Hill  Weedwipe
5    Bracken  Fusilade
6      Scrub  Weedwipe
7  Grassland  Allclear
```

Using the `match()` Function

- Notice two vectors in **`match()`**

```
> herbicides$Herbicide[match(worms$Vegetation,herbicides$Type)]
 [1] Allclear Twinspan Allclear Propinol Weedwipe Allclear Allclear
 [8] Twinspan Fusilade Allclear Weedwipe Allclear Allclear Allclear
[15] Propinol Propinol Weedwipe Twinspan Allclear Weedwipe
Levels: Allclear Fusilade Propinol Twinspan Vanquish Weedwipe
```

Could add this information as a new column in **`worms`**

```
> worms$hb <- herbicides$Herbicide[match(worms$Vegetation,herbicides$Type)]
> worms
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density       hb
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4 Allclear
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7 Twinspan
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2 Allclear
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5 Propinol
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6 Weedwipe
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2 Allclear
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3 Allclear
```

Using the `match()` Function

- Or create a new dataframe **`recs`** with all fields:

```
> recs <- data.frame(worms,hb=herbicides$Herbicide[match(worms$Vegetation,herbicides$Type)])
> recs
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density       hb     hb.1
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4 Allclear Allclear
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7 Twinspan Twinspan
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2 Allclear Allclear
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5 Propinol Propinol
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6 Weedwipe Weedwipe
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2 Allclear Allclear
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3 Allclear Allclear
8            Ashurst  2.1     0     Arable     4.8 FALSE            4 Twinspan Twinspan
9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9 Fusilade Fusilade
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7 Allclear Allclear
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8 Weedwipe Weedwipe
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1 Allclear Allclear
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2 Allclear Allclear
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0 Allclear Allclear
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6 Propinol Propinol
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8 Propinol Propinol
17         Cheapside  2.2     8      Scrub     4.7  TRUE            4 Weedwipe Weedwipe
18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5 Twinspan Twinspan
```

Merging Two Dataframes

- Two dataframes sharing variables genus and species:

```
> (lifeforms <- read.table("c:\\temp\\lifeforms.txt",header=T))
  Genus     species lifeform
1  Acer platanoides     tree
2  Acer    palmatum     tree
3 Ajuga     reptans     herb
4 Conyza sumatrensis   annual
5 Lamium       album     herb

> (flowering <- read.table("c:\\temp\\fltimes.txt",header=T))
     Genus      species flowering
1     Acer   platanoides      May
2    Ajuga       reptans     June
3 Brassica         napus    April
4 Chamerion angustifolium     July
5   Conyza     bilbaoana   August
6   Lamium         album  January
```

Merging Two Dataframes

- With identical variable names, can use **`merge()`**:

```
> merge(flowering,lifeforms)
   Genus     species flowering lifeform
1   Acer platanoides       May     tree
2  Ajuga      reptans      June     herb
3 Lamium        album   January     herb
```

- Important point is that merged dataframe only contains those rows which had *complete* entries in both dataframes.

Merging Two Dataframes

- If want to include all species with NA's inserted when flowering times or lifeforms are unknown, use **`all=T`**.

```
> (both <- merge(flowering,lifeforms,all=T))
      Genus        species flowering lifeform
1      Acer   platanoides       May     tree
2      Acer      palmatum      <NA>     tree
3     Ajuga        reptans      June     herb
4  Brassica          napus     April     <NA>
5 Chamerion angustifolium      July     <NA>
6    Conyza     bilbaoana    August     <NA>
7    Conyza   sumatrensis      <NA>   annual
8    Lamium          album   January     herb
```

Merging Two Dataframes with Synonym Variables

- ‘Seeds’ dataframe with ***same variables*** but with ***different names*** (e.g. synonyms).
  - Variable Genus is ‘name1’ and variable species is ‘name2’.

```
> (seeds <- read.table("c:\\temp\\seedwts.txt",header=T))
     name1         name2 seed
1     Acer   platanoides 32.0
2  Lamium         album 12.0
3   Ajuga        reptans  4.0
4 Chamerion angustifolium  1.5
5  Conyza     bilbaoana  0.5
6 Brassica         napus  7.0
7     Acer      palmatum 21.0
8  Conyza   sumatrensis  0.6
```

Merging Two Dataframes with Synonym Variables

- Let's do a simple **`merge()`**

```
> merge(both,seeds)
       Genus       species flowering lifeform     name1         name2 seed
1       Acer   platanoides       May     tree      Acer   platanoides 32.0
2       Acer      palmatum      <NA>     tree      Acer   platanoides 32.0
3      Ajuga        reptans      June     herb      Acer   platanoides 32.0
4   Brassica          napus     April     <NA>      Acer   platanoides 32.0
5  Chamerion angustifolium      July     <NA>      Acer   platanoides 32.0
6     Conyza     bilbaoana    August     <NA>      Acer   platanoides 32.0
7     Conyza   sumatrensis      <NA>   annual      Acer   platanoides 32.0
8     Lamium         album   January     herb      Acer   platanoides 32.0
9       Acer   platanoides       May     tree    Lamium         album 12.0
10      Acer      palmatum      <NA>     tree    Lamium         album 12.0
11     Ajuga        reptans      June     herb    Lamium         album 12.0
12  Brassica          napus     April     <NA>    Lamium         album 12.0
13 Chamerion angustifolium      July     <NA>    Lamium         album 12.0
14    Conyza     bilbaoana    August     <NA>    Lamium         album 12.0
 .
 .
```

Merging Two Dataframes with Synonym Variables

- Need to inform **`merge()`** of synonyms:

```
> merge(both,seeds,by.x=c("Genus","species"),by.y=c("name1","name2"))
      Genus        species flowering lifeform seed
1      Acer       palmatum      <NA>     tree 21.0
2      Acer    platanoides       May     tree 32.0
3     Ajuga        reptans      June     herb  4.0
4  Brassica          napus     April     <NA>  7.0
5 Chamerion angustifolium      July     <NA>  1.5
6    Conyza      bilbaoana    August     <NA>  0.5
7    Conyza    sumatrensis      <NA>   annual  0.6
8    Lamium          album   January     herb 12.0 .
 .
```

Adding Margins

- Dataframe showing sales by season and by person:

```
> frame <- read.table("c:\\temp\\sales.txt",header=T); frame
              name spring summer autumn winter
1       Jane.Smith     14     18     11     12
2     Robert.Jones     17     18     10     13
3      Dick.Rogers     12     16      9     14
4 William.Edwards     15     14     11     10
5      Janet.Jones     11     17     11     16

> people <- rowMeans(frame[,2:5]); people
[1] 13.75 14.50 12.75 12.50 13.75

> people <- people-mean(people); people
[1]  0.30  1.05 -0.70 -0.95  0.30
```

Want to add column showing peoples' departure from mean

Adding Margins: New Column

- Dataframe showing sales by season and by person:

```
> ## Add new column using cbind():

> (new.frame <- cbind(frame,people))
              name spring summer autumn winter people
1       Jane.Smith     14     18     11     12   0.30
2     Robert.Jones     17     18     10     13   1.05
3      Dick.Rogers     12     16      9     14  -0.70
4 William.Edwards     15     14     11     10  -0.95
5      Janet.Jones     11     17     11     16   0.30
```

Robert Jones is most effective sales person

Adding Margins: New Row

- Calculate column means the same way:

```
> seasons <- colMeans(frame[,2:5]); seasons
spring summer autumn winter
  13.8   16.6   10.4   13.0

> seasons <- seasons-mean(seasons); seasons
spring summer autumn winter
  0.35   3.15  -3.05  -0.45

> ## Can't use rbind so we make a copy
> ## of one of the rows of the new dataframe:
> new.row <- new.frame[1,]; new.row
        name spring summer autumn winter people
1 Jane.Smith     14     18     11     12    0.3
```

Adding Margins: New Row

- And then we edit the new row:

```
> new.row[1] <- "seasonal effects"; new.row # edit Jane.Smith
               name spring summer autumn winter people
1 seasonal effects     14     18     11     12    0.3

> new.row[2:5] <- seasons; new.row #edit row to show mean diffs
               name spring summer autumn winter people
1 seasonal effects   0.35   3.15  -3.05  -0.45    0.3

> new.row[6] <- 0; new.row # initialize gm to be 0
               name spring summer autumn winter people
1 seasonal effects   0.35   3.15  -3.05  -0.45      0
```

Adding Margins: New Row

- Now we can use **`rbind()`** to add row to bottom:

```
> (new.frame <- rbind(new.frame,new.row))
                name spring summer autumn winter people
1         Jane.Smith  14.00  18.00  11.00  12.00   0.30
2       Robert.Jones  17.00  18.00  10.00  13.00   1.05
3        Dick.Rogers  12.00  16.00   9.00  14.00  -0.70
4    William.Edwards  15.00  14.00  11.00  10.00  -0.95
5        Janet.Jones  11.00  17.00  11.00  16.00   0.30
6 seasonal effects     0.35   3.15  -3.05  -0.45   0.00
```

Adding Margins: New Row

- Replace counts of sales by departures from overall mean sale per person per season:

```
> gm <- mean(unlist(new.frame[1:5,2:5])); gm
[1] 13.45

> gm <- rep(gm,4); gm
[1] 13.45 13.45 13.45 13.45

> new.frame[1:5,2:5] <- sweep(new.frame[1:5,2:5],2,gm); new.frame
                name spring summer autumn winter people
1        Jane.Smith   0.55   4.55  -2.45  -1.45   0.30
2      Robert.Jones   3.55   4.55  -3.45  -0.45   1.05
3       Dick.Rogers  -1.45   2.55  -4.45   0.55  -0.70
4  William.Edwards   1.55   0.55  -2.45  -3.45  -0.95
5       Janet.Jones  -2.45   3.55  -2.45   2.55   0.30
6 seasonal effects   0.35   3.15  -3.05  -0.45   0.00
```

`worms` (again): Summarizing Contents of Dataframes

```
> worms
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

Summarizing the Contents of Dataframes

- **`summary()`**

```
> summary(worms)
   Field.Name          Area           Slope          Vegetation     Soil.pH         Damp          Worm.density
 Ashurst     : 1   Min.   :0.800   Min.   : 0.00   Arable    :3   Min.   :3.500   Mode :logical   Min.   :0.00
 Cheapside   : 1   1st Qu.:2.175   1st Qu.: 0.75   Grassland:9    1st Qu.:4.100   FALSE:14        1st Qu.:2.00
 Church.Field: 1   Median :3.000   Median : 2.00   Meadow    :3   Median :4.600   TRUE :6         Median :4.00
 Farm.Wood   : 1   Mean   :2.990   Mean   : 3.50   Orchard   :1   Mean   :4.555   NA's :0         Mean   :4.35
 Garden.Wood : 1   3rd Qu.:3.725   3rd Qu.: 5.25   Scrub     :4   3rd Qu.:5.000                   3rd Qu.:6.25
 Gravel.Pit  : 1   Max.   :5.100   Max.   :11.00                  Max.   :5.700                   Max.   :9.00
 (Other)     :14
```

Values of continuous variable are summarized under six headings: one parametric (the **`mean`**) and five non-parametric (**`maximum`**, (**`minimum`**,**`median`**, 25th percentile or **`1st quartile`**, and 75th percentile or **`3rd quartile`**). Levels of categorical variables are counted. Note that field names are not listed in full because they are unique to each row; six of them are named, then R says ‘14 others’.

Summarizing the Contents of Dataframes

- **aggregate()**

```
> aggregate(worms[,c(2,3,5,7)],by=list(veg=Vegetation),mean)
        veg     Area    Slope  Soil.pH Worm.density
1    Arable 3.866667 1.333333 4.833333     5.333333
2 Grassland 2.911111 3.666667 4.100000     2.444444
3    Meadow 3.466667 1.666667 4.933333     6.333333
4   Orchard 1.900000 0.000000 5.700000     9.000000
5     Scrub 2.425000 7.000000 4.800000     5.250000
```

aggregate() is used like **tapply()** to apply a function (**mean()** in this case) to the levels of a specified categorical variable (**Vegetation** in this case) for a specified range of variables (**Area**, **Slope**, **Soil.ph** and **Worm.density** are defined using their subscripts as a column index in **worms[,c(2,3,5,7)]**.

Summarizing the Contents of Dataframes

- **aggregate()**

```
> aggregate(worms[,c(2,3,5,7)],by=list(veg=Vegetation,d=Damp),mean)
        veg     d     Area    Slope  Soil.pH Worm.density
1    Arable FALSE 3.866667 1.333333 4.833333     5.333333
2 Grassland FALSE 3.087500 3.625000 3.987500     1.875000
3   Orchard FALSE 1.900000 0.000000 5.700000     9.000000
4     Scrub FALSE 3.350000 5.000000 4.700000     7.000000
5 Grassland  TRUE 1.500000 4.000000 5.000000     7.000000
6    Meadow  TRUE 3.466667 1.666667 4.933333     6.333333
7     Scrub  TRUE 1.500000 9.000000 4.900000     3.500000
```

The **by** argument needs to be a list even if, as here, we have only one classifying factor. Here are the aggregated summaries for **Vegetation** and **Damp**. Note that this summary is unbalanced because there were no damp arable or orchard sites and no dry meadows.

Summarizing the Contents of Dataframes

```
> by(worms,Vegetation,mean)
Vegetation: Arable
  Field.Name         Area        Slope   Vegetation      Soil.pH         Damp Worm.density
          NA     3.866667     1.333333           NA     4.833333     0.000000     5.333333
------------------------------------------------------------------------------------------
Vegetation: Grassland
  Field.Name         Area        Slope   Vegetation      Soil.pH         Damp Worm.density
          NA    2.9111111    3.6666667           NA    4.1000000    0.1111111    2.4444444
------------------------------------------------------------------------------------------
Vegetation: Meadow
  Field.Name         Area        Slope   Vegetation      Soil.pH         Damp Worm.density
          NA     3.466667     1.666667           NA     4.933333     1.000000     6.333333
------------------------------------------------------------------------------------------
Vegetation: Orchard
  Field.Name         Area        Slope   Vegetation      Soil.pH         Damp Worm.density
          NA          1.9          0.0           NA          5.7          0.0          9.0
------------------------------------------------------------------------------------------
Vegetation: Scrub
  Field.Name         Area        Slope   Vegetation      Soil.pH         Damp Worm.density
          NA        2.425        7.000           NA        4.800        0.500        5.250
```