Data Manipulation

Michael Malick

Updated: 15 May 2013

Order a Vector or Dataframe

It is often useful to order a vector or dataframe

```
x <- c(10, 11, 9, 6, 1, 13)

sort(x)
sort(x, decreasing = TRUE)

order(x) # gives indices
x[order(x)]

iris[order(iris$Sepal.Length), ]
iris[order(iris$Species, iris$Sepal.Length), ]
```

Add and Remove Columns

You can easily add or remove a column from a dataframe

```
# Add Columns
iris$random <- rnorm(150)
iris$index  <- 1:150


# Remove Columns
iris$random <- NULL
iris <- iris[, -6]
```

Missing Values

- R uses the `NA` symbol to represent missing values
- When reading in data files R automatically replaces blank cells with `NA`
- `NA` values are handled differently than regular data points

```
x <- c(2, 3, 4, NA)
mean(x)
mean(x, na.rm = TRUE)
```

Missing Values

You can test whether a value is NA and remove it if it is

```
x <- c(2, 3, 4, NA, 5, NA, 6)

is.na(x)
sum(is.na(x))

!is.na(x)

x <- x[!is.na(x)]
```

Combine Datasets: `rbind()`

Datasets are often scattered across files and need to be combined

```
head(beaver1)
head(beaver2)

beaver1$beaver <- 1
beaver2$beaver <- 2

beaver <- rbind(beaver1, beaver2)
```

Merge Two Dataframes: merge ( )

Datasets can be easily merged keeping only unique data

```
grade <- data.frame(name = c("Mike", "Erin", "Joe"),
                    grade = c("A", "C", "B"))

perc <- data.frame(name = c("Mike", "Erin", "Joe"),
                   percent = c(97, 77, 88))

merge(grade, perc)
```

Merge Two Dataframes: merge ( )

Datasets can be easily merged keeping only unique data

```
grade <- data.frame(name = c("Mike", "Erin", "Joe"),
                    grade = c("A", "C", "B"))

perc <- data.frame(first = c("Mike", "Erin", "Joe"),
                   percent = c(97, 77, 88))

merge(grade, perc)
merge(grade, perc, by.x = "name", by.y = "first")
```

Dataset Organization

There are two primary ways to organize data

Wide

Year	Site1	Site2	Site3
1960	240,000	142,236	332,867
1961	60,000	45,972	47,049
1962	133,800	208,086	194,910
1963	38,081	373,412	127,154

↔

Long

Year	Site	Count
1960	Site1	240,000
1961	Site1	60,000
1962	Site1	133,800
1963	Site1	38,081
1960	Site2	142,236
1961	Site2	45,972
1962	Site2	208,086
1963	Site2	373,412
1960	Site3	332,867
1961	Site3	47,049
1962	Site3	194,910
1963	Site3	127,154

Reshape2 Package

"Reshape lets you flexibly restructure and aggregate data using just two function: melt and cast"

```
install.packages("reshape2")
library(reshape2)
```

- `melt()`: go from "wide" format to "long" format
- `dcast()`: go from "long" format to "wide" format
  - Also used to aggregate data

Reshape Example #1

```
mm <- data.frame(year = 1960:1963,
    site1 = c(10, 13, 9, 20),
    site2 = c(30, 11, 18, 24),
    site3 = c(40, 44, 49, 20))

melt(mm)

mm.long <- melt(mm, id.vars = "year")

mm.wide <- dcast(mm.long, year ~ variable,
    value.var = "value")
```

Reshape Example #2

```
head(airquality)

melt(airquality)

airquality.long <- melt(airquality,
   id.vars = c("Month", "Day"))

melt(airquality,
   id.vars = c("Month", "Day"),
   measure.vars = "Ozone")

dcast(airquality.long, Month + Day ~
   variable, value.var = "value")
```

You Try...

1. Order the `mtcars` dataset by increasing `mpg`
2. Add a column called “index” to the `Orange` dataset that gives the row number for each record
3. Remove the column in the `Orange` dataset you just created