Change Default Location of Packages

Do this only once on ITAP machines

Create a directory to store packages on your “H” drive

```
H:/R_LIBS
```

Create and edit R startup file to change the default location of packages

- Create a new file named “.Rprofile” under R startup directory

  ```
  H:/My Documents
  ```

- Enter this line to “.Rprofile”

  ```
  .libPaths(c("H:/R_LIBS",.libPaths()))
  ```

Start R, install and load packages as usual
Now packages will be installed into and loaded from your custom directory

Other useful functions

count()

arrange()

summarise()

colwise()

We will use Barley data in the lattice package to demonstrate the usage of these functions

Barley Data

Load data

```
> library(lattice)
> ?barley
> barley
> head(barley)
> tail(barley)
> summary(barley)
```

Yield for 10 varieties of barley at 6 sites in each of two years

120 records

4 variables: yield, variety, year, site

Function count()

Count the number of occurrences

```
count(df, vars, wt_var)
```

Number of observations for each site

```
> count(df=barley, vars="site")
```

Number of observations for each site and year combination

```
> count(df=barley, vars=c("site", "year"))
```

Number of observations for each site, again but with weight

```
> tmp = count(df=barley, vars=c("site", "year"))
> tmp
> count(df=tmp, vars="site", wt_var="freq")
```

Function arrange()

Order a data frame by its columns

```
arrange(df, ...)
```

Order by one column: by yield from largest to smallest

```
> arrange(df=barley, -yield)
```

Order by multiple columns: first by year and site, then by yield from largest to smallest

```
> arrange(df=barley, year, site, -yield)
```

Function summarise()

Summarise a data frame

```
summarise(.data, ...)
```

Summarise the whole data frame

```
> summarise(.data=barley,
    max=max(yield), min=min(yield)
)
```

Group-wise summaries

```
> ddply(
    .data        = barley,
    .variables   = c("year", "site"),
    .fun         = summarise,
    max          = max(yield),
    min          = min(yield)
)
```

Function colwise()

Column-wise function

```
colwise(.fun, .cols)
```

Turn a function that operates on a vector into a function that operates column-wise on a data frame

Add a column to Barley data

```
> barley$noise = rnorm(nrow(barley))
```

Compute the mean for both yield and noise

```
> colwise(.fun=mean,
    .cols=c("yield","noise")
)
> colwise(.fun=mean,
    .cols=c("yield","noise")
)(barley)
```

Carrying Out Split-Apply-Combine

Split and combine are taken care of by plyr

Analyst needs only think about applying methods

Goal: compute the five number summary of yield at each site in each year

Yield at one site in one year is a working unit

Subset data at one site in one year

```
> unit = subset(barley,
    subset=(site=="University Farm" & year==1931)
)
```

Apply the Analysis

Compute the five number summary

```
> result = quantile(unit$yield)
```

Make it a function

```
> five.num = function(data) {
    quantile(data$yield)
}
> result = five.num(unit)
```

Yields at Every Site in Every year: User plyr Functions

Use ddply()

```
> results.plyr = ddply(
    .data        = barley,
    .variables   = c("site","year"),
    .fun         = five.num
)
```

Yields at Every Site in Every year: User Base R Functions

Split to pieces

```
> pieces = split(
    x = barley,
    f = list(barley$site, barley$year)
)
```

Initialize results

```
> results = list()
```

Apply to pieces

```
> for(i in seq_along(pieces)) {
    piece              = pieces[[i]]
    results[[i]]       = five.num(piece)
}
```

Combine pieces

```
> results = do.call("rbind", results)
> results = as.data.frame(results)
```

Yields at Every Site in Every year: User Base R Functions Cont.

Not done yet, need proper labels

Obtain the names of pieces

```
> groups = names(pieces)
```

Split the names by dot

```
> groups = strsplit(groups, split="\\.")
```

Make the names a data.frame

```
> groups       = do.call("rbind", groups)
> groups       = as.data.frame(groups)
> names(groups) = c("site", "year")
```

Merge with the five number summary data.frame

```
> results.r = cbind(groups, results)
```

Pop-up Quiz No. 1

Find the difference between yields in 1931 and 1932 for each variety at each site, and order the result by site and difference

```
> head(result)
            variety           site difference
1           Glabron Grand Rapids  -14.70000
2  Wisconsin No. 38 Grand Rapids  -13.80000
3          Svansota Grand Rapids  -13.03334
4           No. 457 Grand Rapids  -12.70000
5         Manchuria Grand Rapids  -10.83334
6             Trebi Grand Rapids   -9.13334
```

Pop-up Quiz No. 1 Cont.

Write your own function to compute the difference for each unit

```
> find.diff = function(unit) {
    c(difference = diff(unit$yield))
}
```

Let plyr take care of the rest

```
> result = ddply(
    .data        = barley,
    .variables   = c("variety", "site"),
    .fun         = find.diff
)
```

Order the result

```
> result = arrange(result, site, difference)
```

Pop-up Quiz No. 1 Cont.

Or, use function summarise()

```
> result = ddply(
    .data        = barley,
    .variables   = c("variety", "site"),
    .fun         = summarise,
    difference   = diff(yield)
)
```

Order the result

```
> result = arrange(result, site, difference)
```

Pop-up Quiz No. 2

Find the varieties with the largest or smallest yield at each site in each year

Min and max in the same row

```
> head(result)
   year            site max.yield      max.variety min.yield min.variety
1  1932    Grand Rapids  32.23333            Velvet  14.43333     Glabron
2  1932          Duluth  31.36667          Peatland  22.23333    Svansota
3  1932 University Farm  38.00000 Wisconsin No. 38  25.56667     No. 462
4  1932          Morris  47.16667 Wisconsin No. 38  34.36666   Manchuria
5  1932       Crookston  41.83333             Trebi  20.63333    Svansota
6  1932          Waseca  58.16667 Wisconsin No. 38  33.46667   Manchuria
```

Min and max in different rows

```
> head(result)
   year            site type    yield           variety
1  1932    Grand Rapids  min 14.43333           Glabron
2  1932    Grand Rapids  max 32.23333            Velvet
3  1932          Duluth  min 22.23333          Svansota
4  1932          Duluth  max 31.36667          Peatland
5  1932 University Farm  min 25.56667           No. 462
6  1932 University Farm  max 38.00000 Wisconsin No. 38
```

Pop-up Quiz No. 2 Cont.

Min and max in the same row

```
> result = ddply(
    .data        = barley,
    .variables   = c("year", "site"),
    .fun         = summarise,
    max.yield    = yield[which.max(yield)],
    max.variety  = variety[which.max(yield)],
    min.yield    = yield[which.min(yield)],
    min.variety  = variety[which.min(yield)]
)
```

Pop-up Quiz No. 2 Cont.

Min and max in different rows

```
> result = ddply(
    .data        = barley,
    .variables   = c("year", "site"),
    .fun         = summarise,
    type         = c("min", "max"),
    yield        = range(yield),
    variety      = variety[c(which.min(yield),which.max(yield))]
)
```