Lesson 05
Data Manipulation

Outline

- Manipulating Data Frames
  - Using `dplyr` package
    - pipe, `tibble`, `select`, `filter`, `mutate`
    - `summarize`, `group_by`, `arrange`
- Other packages (not covered)
  - `data.table`, `purrr`

Data Manipulation – dplyr Package

- `dyplyr` package is today's standard for data manipulation in R
  - Designed for fast manipulation of data frames
    - Matrices and lists are moved to `purrr` package (not covered)
  - Takes advantage of pipes (from `magrittr` package)
    - Easier chaining of functions / operations instead of storing results in temporary variables
  - Simpler syntax when compared to `data.table`, another popular data manipulation package (not covered) designed for manipulation of data frames

Data Manipulation – dplyr Package

- `dyplyr` package is today's standard for data manipulation in R
  - Syntax relies on using the "grammar of data" familiar to SQL users through the use of verbs such as: `select`, `filter`, `group_by`, etc..
- `dplyr` extends `data.frame` into `tibble` object
  - Prints a subset of rows and columns (to fit on screen)
  - I will use `tib` instead of `tbl` for short-hand notation

Data Manipulation – select Function

- The `select` function takes a `data.frame` or `tibble` and lists the specified set of columns
  - Works with traditional, nested arguments or using pipes

```
select(loans_tib, loanType, mthPmt)
loans_tib %>% select(loanType, mthPmt)
```

  - Number of alternative specifications include
    - Vector `loans_tib %>% select(c(loanType, mthPmt))`
    - Variable with quoted columns: `selCols=c('loanType','mthPmt')`
    - Column numbers: `loans_tib %>% select(9, 10)`
    - Using search functions `starts_with`, `ends_with`, `contains`, `matches`
    - `loans_tib %>% select(starts_with('loan'))`
    - Excluding columns with the minus sign
    - `loans_tib %>% select(-ends_with('Name'))`

Data Manipulation – filter Function

- The `filter` function takes a `data frame` or `tibble` and restricts the rows based on a logical criteria
  - Same as the SQL WHERE clause
  - Uses relational operators ==, <, >, <=, >=, !=

```
loans_tib %>% filter(loanType == 'Mortg')
```

  - Also uses logical operators: &, | and !

```
loans_tib %>% filter((intRate > 0.07 & loanType ==
'Mortg') | (loanType == 'Car' & amount < 30000))
```

  - You can also use variables to build the logical expressions

```
loans_tib %>% filter(loanType == selType)
```

  - Combine select and filter functions

Data Manipulation – mutate Function

- `mutate` function modifies the existing or creates new columns in a `data.frame` or a `tibble`

```
loans_tib %>% select(2:3,6:8,10) %>% mutate(totPmt =
loanTerm * mthPmt*12)
```

- Can be immediately used in the same mutate call

```
loans_tib %>% select(2:3, 6:8, 10) %>%
mutate(loanRatio=mthPmt/amount, loanConst=loanRatio*1000)
```

- These changes have to be specifically assigned to the existing (or a new) data.frame / tibble
  - Uses assignment `%<>%` operator from `magrittr` package

```
loans2_tib <- loans_tib
loans2_tib %<>%  select(2:3, 6:8, 10) %>%
  mutate(totPmt=loanTerm*mthPmt) %>%
  mutate(loanRatio=mthPmt/amount, loanConst=loanRatio*1000)
```

Data Manipulation – Other Functions

- `summarize` function results in one or more summaries of the (typically) numerical columns
  - Same as using summary functions in SQL without GROUP BY

```
loans2_tib %>% summarize(mean(mthPmt))
```

- `group_by` function used to partition the data and then apply the `summarize` function on the different groups
  - GROUP BY almost always used with SQL summary queries

```
loans2_tib %>% group_by(loanType) %>%
summarize(AvgMthPmt=mean(mthPmt))
```

- `arrange` function used to sort the data
  - Much easier and more intuitive than order / sort functions

```
… %>% arrange(desc(AvgMthPmt))
```

Summary

- Reviewed a fair amount of data manipulation (a.k.a. munging, wrangling, transforming, …) tools and techniques
- Concentrated on `data frames/tibbles` with `dplyr` package representing a defacto data manipulation standard today
- Mentioned (without covering) a few other data manipulation packages such as `data.table` and `purrr`