GEOG 5680
Introduction to R
08: Data manipulation with dplyr

Simon Brewer

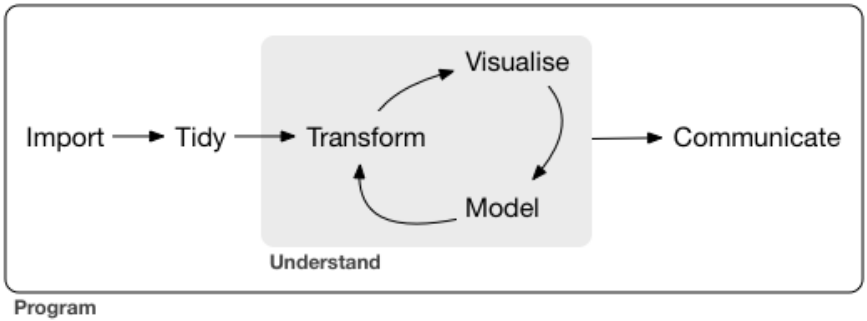
Geography Department
University of Utah
Salt Lake City, Utah 84112

simon.brewer@geog.utah.edu

May 04, 2020

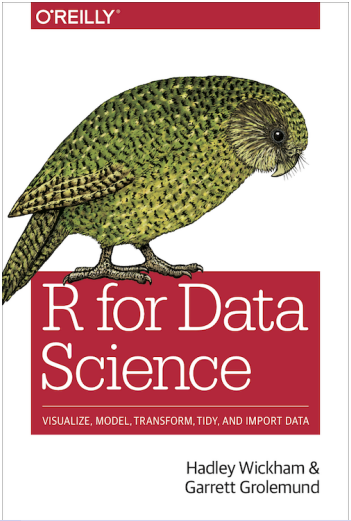
Tidyverse

- Set of packages developed by Hadley Wickham and RStudio for data science
- Designed to cover the main steps of data analysis: data import, manipulation, transformation, visualization and modeling



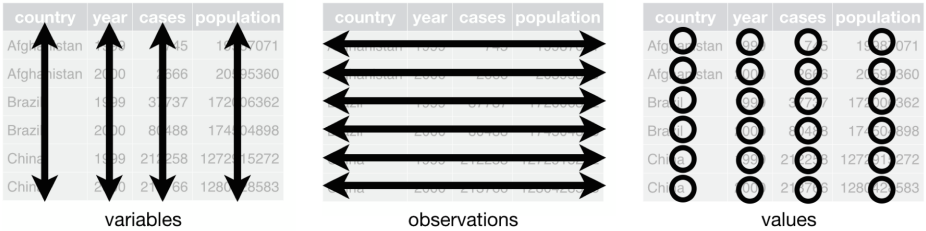
Tidyverse

- **ggplot2**: visualization
- **dplyr**: manipulation
- **tidyr**: tidying
- **purrr**: functional programming
- **tibble**: improved data frames
- **stringr**: string manipulation
- **forcat**: factor manipulation



Data structure semantics

- Data manipulation of data frames (or tibbles)
- Data frame consists of *observations* (rows) and *variables* (columns)
- Each combination of observations and variables has a *value*
- Each variable is one of R's modes (numeric, factor, etc.)



Split-apply-combine

- Common approach to data analysis
- "break up a big problem into manageable pieces, operate on each piece independently and then put all the pieces back together"
  - Group-wise ranking or standardization
  - Creating data summaries or marginal means
  - Fitting models to individual panels of data
- Analogous to the map-reduce strategy in parallel processing

dplyr

- The **dplyr** package contains tools for this approach
- Set of functions that can be combined to provide full workflow
  - `filter`: subset or remove observations (rows)
  - `select`: subset or remove variables (columns)
  - `mutate`: modify or creaate new variable
  - `summarize`: aggregate multiple values (e.g. mean or sum)
  - `group_by`: divide dataset according to one or more factor(s)
  - `arrange`: order the observations

dplyr

```
filter(gap, lifeExp < 29)

##       country year     pop continent lifeExp gdpPercap
## 1 Afghanistan 1952 8425333      Asia  28.801  779.4453
## 2      Rwanda 1992 7290203    Africa  23.599  737.0686

filter(gap, country == "Rwanda", year > 1979)

##   country year     pop continent lifeExp gdpPercap
## 1  Rwanda 1982 5507565    Africa  46.218  881.5706
## 2  Rwanda 1987 6349365    Africa  44.020  847.9912
## 3  Rwanda 1992 7290203    Africa  23.599  737.0686
## 4  Rwanda 1997 7212583    Africa  36.087  589.9445
## 5  Rwanda 2002 7852401    Africa  43.413  785.6538
## 6  Rwanda 2007 8860588    Africa  46.242  863.0885
```

dplyr

```
select(gap, year, lifeExp)
```

```
##      year  lifeExp
## 1    1952 28.80100
## 2    1957 30.33200
## 3    1962 31.99700
## 4    1967 34.02000
## 5    1972 36.08800
## 6    1977 38.43800
## 7    1982 39.85400
## 8    1987 40.82200
## 9    1992 41.67400
## 10   1997 41.76300
## 11   2002 42.12900
## 12   2007 43.82800
## 13   1952 55.23000
## 14   1957 59.28000
## 15   1962 64.82000
## 16   1967 66.22000
## 17   1972 67.69000
## 18   1977 68.93000
```

The pipe

- Based on the concept of Unix pipes that allow commands to be chained together
- The **magrittr** package introduced a similar concept to R
- Allows the output of one function to be passed to the next
- Syntax is '%>%'
- Works with many base R functions, but integrates well with **dplyr**

dplyr

```
gap %>% filter(country == "Rwanda", year > 1979)

##   country year     pop continent lifeExp gdpPercap
## 1  Rwanda 1982 5507565    Africa  46.218  881.5706
## 2  Rwanda 1987 6349365    Africa  44.020  847.9912
## 3  Rwanda 1992 7290203    Africa  23.599  737.0686
## 4  Rwanda 1997 7212583    Africa  36.087  589.9445
## 5  Rwanda 2002 7852401    Africa  43.413  785.6538
## 6  Rwanda 2007 8860588    Africa  46.242  863.0885

gap %>%
  filter(country == "Rwanda", year > 1979) %>%
  select(country, pop)

##   country     pop
## 1  Rwanda 5507565
## 2  Rwanda 6349365
## 3  Rwanda 7290203
## 4  Rwanda 7212583
## 5  Rwanda 7852401
## 6  Rwanda 8860588
```

Split-apply-combine with **dplyr**

Step 1: split using group_by

```
gap %>%
  group_by(continent)

## # A tibble: 1,704 x 6
## # Groups:   continent [5]
##    country      year      pop continent lifeExp gdpPercap
##    <chr>       <int>    <dbl> <chr>       <dbl>     <dbl>
##  1 Afghanistan  1952  8425333 Asia         28.8      779.
##  2 Afghanistan  1957  9240934 Asia         30.3      821.
##  3 Afghanistan  1962 10267083 Asia         32.0      853.
##  4 Afghanistan  1967 11537966 Asia         34.0      836.
##  5 Afghanistan  1972 13079460 Asia         36.1      740.
##  6 Afghanistan  1977 14880372 Asia         38.4      786.
##  7 Afghanistan  1982 12881816 Asia         39.9      978.
##  8 Afghanistan  1987 13867957 Asia         40.8      852.
##  9 Afghanistan  1992 16317921 Asia         41.7      649.
## 10 Afghanistan  1997 22227415 Asia         41.8      635.
## # ... with 1,694 more rows
```

Split-apply-combine with **dplyr**

Steps 2 and 3: apply a function (here summarize the mean life expectancy) and combine into new data frame:

```
gap %>%
  group_by(continent) %>%
  summarize(meanLifeExp = mean(lifeExp))

## # A tibble: 5 x 2
##   continent meanLifeExp
##   <chr>           <dbl>
## 1 Africa           48.9
## 2 Americas         64.7
## 3 Asia             60.1
## 4 Europe           71.9
## 5 Oceania          74.3
```

Split-apply-combine with **dplyr**

Note that you can use multiple factors for grouping:

```
gap %>%
  group_by(continent, year) %>%
  summarize(meanLifeExp = mean(lifeExp))

## # A tibble: 60 x 3
## # Groups:   continent [5]
##    continent  year meanLifeExp
##    <chr>     <int>       <dbl>
##  1 Africa     1952        39.1
##  2 Africa     1957        41.3
##  3 Africa     1962        43.3
##  4 Africa     1967        45.3
##  5 Africa     1972        47.5
##  6 Africa     1977        49.6
##  7 Africa     1982        51.6
##  8 Africa     1987        53.3
##  9 Africa     1992        53.6
## 10 Africa     1997        53.6
## # ... with 50 more rows
```

Split-apply-combine with **dplyr**

Results can be piped to other functions (e.g. `ggplot`):

```
gap %>%
  group_by(continent, year) %>%
  summarize(meanLifeExp = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = meanLifeExp, col = continent)) + geom_line()
```