

Introduction to R for Data Management and Analysis

Marcel Ramos, MPH

Session 4

Announcements

- Additional topics to cover
 - Reshaping your data
- Piping operator `magrittr::%>%` or `|>` (new; R > 4.2)
 - Takes the LHS as input to the RHS
 - Readable
 - Allows easy command chaining

Outline for today

- Review exercises
- Using dplyr to combine data manipulations
- Reshaping data
- Plotting in base R
- Exploratory Data Analysis
- Intro to ggplot2
- Saving graphics

But first, a quote...

The data may not contain the answer. The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data.
-John Tukey

Review

- Exercises 1 - 3

Using the *nycflights13* dataset

```
library(nycflights13); library(dplyr)
flights |> group_by(carrier) |>
  summarise(avg_depdelay = mean(dep_delay, na.rm = TRUE),
            count = n()) |> left_join(airlines) |>
  arrange(avg_depdelay) |> head()
```

```
## # A tibble: 6 x 4
##   carrier avg_depdelay count name
##   <chr>      <dbl> <int> <chr>
## 1 US          3.78  20536 US Airways Inc.
## 2 HA          4.90   342 Hawaiian Airlines Inc.
## 3 AS          5.80   714 Alaska Airlines Inc.
## 4 AA          8.59  32729 American Airlines Inc.
## 5 DL          9.26  48110 Delta Air Lines Inc.
## 6 MQ         10.6  26397 Envoy Air
```

Reshaping data

- Useful to prepare data for visualizations
- long vs wide
- long format - multiple observations per row (survival data)
- wide format - a single observation per row

Reshaping using *pivot_wider*

```
library(tidyr); library(tidycensus)
head(us_rent_income)
```

```
## # A tibble: 6 x 5
##   GEOID NAME      variable estimate   moe
##   <chr> <chr>    <chr>         <dbl> <dbl>
## 1 01      Alabama income      24476   136
## 2 01      Alabama rent         747     3
## 3 02      Alaska  income      32940   508
## 4 02      Alaska  rent         1200    13
## 5 04      Arizona income      27517   148
## 6 04      Arizona rent         972     4
```


Reshaped us_rent_income

```
us_rent_income |>  
  pivot_wider(names_from = variable,  
              values_from = c(estimate, moe)) |> head(4)
```

```
## # A tibble: 4 x 6
```

```
##   GEOID NAME      estimate_income estimate_rent moe_income m  
##   <chr> <chr>          <dbl>          <dbl>      <dbl>  
## 1 01     Alabama      24476          747        136  
## 2 02     Alaska       32940         1200        508  
## 3 04     Arizona       27517          972        148  
## 4 05     Arkansas      23789          709        165
```

Reshaping using *pivot_longer*

```
head(relig_income)
```

```
## # A tibble: 6 x 11
##   religion ' <$10k' '$10-20k' '$20-30k' '$30-40k' '$40-50k'
##   <chr>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 Agnostic      27        34        60        81        76
## 2 Atheist       12        27        37        52        35
## 3 Buddhist     27        21        30        34        33
## 4 Catholic    418       617       732       670       638
## 5 Don't kn~     15        14        15        11        10
## 6 Evangelic~  575      869     1064      982     881
## # ... with 3 more variables: '$100-150k' <dbl>, '>150k' <dbl>
## #   'Don't know/refused' <dbl>
```

Long dataset

```
relig_income |> pivot_longer(-religion, names_to = "income",  
  values_to = "count") |> head()
```

```
## # A tibble: 6 x 3  
##   religion income    count  
##   <chr>    <chr>    <dbl>  
## 1 Agnostic <$10k      27  
## 2 Agnostic $10-20k    34  
## 3 Agnostic $20-30k    60  
## 4 Agnostic $30-40k    81  
## 5 Agnostic $40-50k    76  
## 6 Agnostic $50-75k   137
```

- -religion - don't include religion when reshaping
- names_to - create an income variable out of the columns
- values_to - cell values are counts

group_by operations

- Allow users to group different levels of categories of 1 or more variables
- Efficient summarization

Using group_by (1)

```
relig_income |> pivot_longer(-religion,  
  names_to = "income", values_to = "count") |>  
  group_by(income) |> summarise(totals = sum(count))
```

```
## # A tibble: 10 x 2
```

##	income	totals
##	<chr>	<dbl>
##	1 <\$10k	1930
##	2 >150k	2608
##	3 \$10-20k	2781
##	4 \$100-150k	3197
##	5 \$20-30k	3357
##	6 \$30-40k	3302
##	7 \$40-50k	3085
##	8 \$50-75k	5185
##	9 \$75-100k	3990

Using group_by (2)

```
relig_income |> pivot_longer(-religion,  
  names_to = "income", values_to = "count") |>  
  group_by(religion) |> summarise(totals = sum(count))
```

```
## # A tibble: 18 x 2
```

##	religion	totals
##	<chr>	<dbl>
##	1 Agnostic	826
##	2 Atheist	515
##	3 Buddhist	411
##	4 Catholic	8054
##	5 Don't know/refused	272
##	6 Evangelical Prot	9472
##	7 Hindu	257
##	8 Historically Black Prot	1995
##	9 Jehovah's Witness	215

Plotting and Graphing

- Exploratory Data Analysis
- Base R graphics
- Intro ggplot2
- Saving graphics

Plotting systems in R

- 'Base' graphics
- lattice
- ggplot2

Exploratory Data Analysis

- Informal representation data
- Looking for patterns, outliers, etc.
- Get familiar with your data!

Types of graphs

- Histogram
- Scatterplot
 - Scatterplot matrix
- Boxplots / dotplots (ggplot2)
- Violin plots (ggplot2)
- Q-Q plots
- Mosaic plots
- and many more!

ggplot2 - Grammar of Graphics

- Different syntax
 - Slight learning curve
- Plots are built in layers
- Operations add layers to the plot

Saving outputs

- Common formats for saving plots:
 - PDF
 - SVG
 - PNG/TIFF
- but there are more
- ggsave

Output sandwich

- Start with a function pdf, png, jpeg, etc.



- End in `dev.off()` for closing the graphics window

Saving plots in ggplot2

- `ggplot2` graphics require a `print` (or a call) before it gets rendered in the file.
- `ggsave` - added to make it easier to save plotting objects

Recommended resources

- Fundamentals of Data Visualization
 - Claus O. Wilke
- R Graphics Cookbook
 - Winston Chang