# Data Reshaping

Data Wrangling in R

# Reshaping Data

In this module, we will show you how to:

1. Reshaping data from wide (fat) to long (tall)

2. Reshaping data from long (tall) to wide (fat)

3. Merging Data/Joins

4. Perform operations by a grouping variable

# Setup

We will show you how to do each operation in base R then show you how to use the `dplyr` or `tidyr` package to do the same operation (if applicable).

See the "Data Wrangling Cheat Sheet using `dplyr` and `tidyr`":

- https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

# What is wide/long data?

See http://www.cookbook-r.com/Manipulating_data/Converting_data_between_wide_and_long_format/

- Wide - multiple columns per observation

    - e.g. visit1, visit2, visit3

```
# A tibble: 2 x 4
     id visit1 visit2 visit3
  <int>  <dbl>  <dbl>  <dbl>
1     1     10      4      3
2     2      5      6     NA
```

- Long - multiple rows per observation

```
# A tibble: 5 x 3
     id visit value
  <dbl> <int> <dbl>
1     1     1    10
2     1     2     4
3     1     3     3
4     2     1     5
5     2     2     6
```

More accurately, data is wide or long **with respect** to certain variables.

# Data used: Charm City Circulator

http://sisbid.github.io/Data-Wrangling/data/Charm_City_Circulator_Ridership.csv

```
circ = read_csv(
  paste0("http://sisbid.github.io/Data-Wrangling/",
         "data/Charm_City_Circulator_Ridership.csv"))
head(circ, 2)
```

```
# A tibble: 2 x 15
  day     date     orangeBoardings orangeAlightings orangeAverage purpleBoardin
  <chr>   <chr>              <dbl>            <dbl>         <dbl>          <db
1 Monday  01/11/…              877             1027           952
2 Tuesday 01/12/…              777              815           796
# … with 9 more variables: purpleAlightings <dbl>, purpleAverage <dbl>,
#   greenBoardings <dbl>, greenAlightings <dbl>, greenAverage <dbl>,
#   bannerBoardings <dbl>, bannerAlightings <dbl>, bannerAverage <dbl>,
#   daily <dbl>
```

```
class(circ$date)
```

```
[1] "character"
```

# Creating a Date class from a character date

```r
library(lubridate) # great for dates!

sum(is.na(circ$date))
```

```
[1] 0
```

```r
sum( circ$date == "")
```

```
[1] 0
```

```r
circ = mutate(circ, date = mdy(date))
sum( is.na(circ$date) ) # all converted correctly
```

```
[1] 0
```

```r
head(circ$date, 3)
```

```
[1] "2010-01-11" "2010-01-12" "2010-01-13"
```

```r
class(circ$date)
```

```
[1] "Date"
```

# Reshaping data from wide (fat) to long (tall): base R

The `reshape` command exists. It is a **confusing** function. Don't use it.

# tidyr package

`tidyr` allows you to "tidy" your data. We will be talking about:

- `gather` - make multiple columns into variables, (wide to long)

- `spread` - make a variable into multiple columns, (long to wide)

- `separate` - string into multiple columns

- `unite` - multiple columns into one string

- All the "join" functions for mergin are in `dplyr`

# Reshaping data from wide (fat) to long (tall): tidyr

`tidyr::gather` - puts column data into rows.

We want the column names into "`var`" variable in the output dataset and the value in "`number`" variable. We then describe which columns we want to "gather:"

```
long = gather(circ, key = "var", value = "number",
              -day, -date, -daily)
head(long, 4)
```

```
# A tibble: 4 x 5
  day       date         daily var              number
  <chr>     <date>        <dbl> <chr>             <dbl>
1 Monday    2010-01-11     952  orangeBoardings     877
2 Tuesday   2010-01-12     796  orangeBoardings     777
3 Wednesday 2010-01-13    1212. orangeBoardings    1203
4 Thursday  2010-01-14    1214. orangeBoardings    1194
```

# Reshaping data from wide (fat) to long (tall): tidyr

- Could be explicit on what we want to gather

```
long = gather(circ, key = "var", value = "number",
              starts_with("orange"), starts_with("purple"),
              starts_with("green"), starts_with("banner"))
long
```

```
# A tibble: 13,752 x 5
   day       date        daily var               number
   <chr>     <date>      <dbl> <chr>               <dbl>
 1 Monday    2010-01-11   952  orangeBoardings       877
 2 Tuesday   2010-01-12   796  orangeBoardings       777
 3 Wednesday 2010-01-13  1212. orangeBoardings      1203
 4 Thursday  2010-01-14  1214. orangeBoardings      1194
 5 Friday    2010-01-15  1644  orangeBoardings      1645
 6 Saturday  2010-01-16  1490. orangeBoardings      1457
 7 Sunday    2010-01-17   888. orangeBoardings       839
 8 Monday    2010-01-18  1000. orangeBoardings       999
 9 Tuesday   2010-01-19  1035  orangeBoardings      1023
10 Wednesday 2010-01-20  1396. orangeBoardings      1375
# ... with 13,742 more rows
```

# Reshaping data from wide (fat) to long (tall): tidyr

```
long %>% count(var)
```

```
# A tibble: 12 x 2
     var                     n
     <chr>              <int>
 1 bannerAlightings   1146
 2 bannerAverage      1146
 3 bannerBoardings    1146
 4 greenAlightings    1146
 5 greenAverage       1146
 6 greenBoardings     1146
 7 orangeAlightings   1146
 8 orangeAverage      1146
 9 orangeBoardings    1146
10 purpleAlightings   1146
11 purpleAverage      1146
12 purpleBoardings    1146
```

# Making a separator

We will use `str_replace` from `stringr` to put _ in the names

```
long = long %>% mutate(
  var = var %>%
    str_replace("Board", "_Board") %>%
    str_replace("Alight", "_Alight") %>%
    str_replace("Average", "_Average")
)
long %>% count(var)
```

```
# A tibble: 12 x 2
   var                  n
   <chr>            <int>
 1 banner_Alightings  1146
 2 banner_Average     1146
 3 banner_Boardings   1146
 4 green_Alightings   1146
 5 green_Average      1146
 6 green_Boardings    1146
 7 orange_Alightings  1146
 8 orange_Average     1146
 9 orange_Boardings   1146
10 purple_Alightings  1146
11 purple_Average     1146
12 purple_Boardings   1146
```

# Reshaping data from wide (fat) to long (tall): tidyr

Now each `var` is boardings, averages, or alightings. We want to separate these so we can have these by route. Remember "." is special character:

```
long = separate(long, var, into = c("route", "type"), sep = "_")
head(long, 2)
```

```
# A tibble: 2 x 6
  day      date        daily route  type        number
  <chr>    <date>      <dbl> <chr>   <chr>        <dbl>
1 Monday   2010-01-11    952 orange  Boardings      877
2 Tuesday  2010-01-12    796 orange  Boardings      777
```

```
unique(long$route)
```

```
[1] "orange" "purple" "green"  "banner"
```

```
unique(long$type)
```

```
[1] "Boardings"  "Alightings" "Average"
```

# Re-uniting all the routes

If we had the opposite problem, we could use the `unite` function:

```
reunited = long %>%
  unite(col = var, route, type, sep = "_")
reunited %>% select(day, var) %>% head(3) %>% print
```

```
# A tibble: 3 x 2
  day       var
  <chr>     <chr>
1 Monday    orange_Boardings
2 Tuesday   orange_Boardings
3 Wednesday orange_Boardings
```

We could also use `paste/paste0`.

# Reshaping data from long (tall) to wide (fat): tidyr

In `tidyr`, the `spread` function spreads rows into columns. Now we have a long data set, but we want to separate the Average, Alightings and Boardings into different columns:

```
# have to remove missing days
wide = long %>% filter(!is.na(date))
wide = wide %>% spread(type, number)
head(wide)
```

```
# A tibble: 6 x 7
  day    date       daily route  Alightings Average Boardings
  <chr>  <date>     <dbl> <chr>       <dbl>   <dbl>     <dbl>
1 Friday 2010-01-15 1644  banner         NA      NA        NA
2 Friday 2010-01-15 1644  green          NA      NA        NA
3 Friday 2010-01-15 1644  orange       1643    1644      1645
4 Friday 2010-01-15 1644  purple         NA      NA        NA
5 Friday 2010-01-22 1394. banner         NA      NA        NA
6 Friday 2010-01-22 1394. green          NA      NA        NA
```

# Pivoting Functions

`pivot_longer` and `pivot_wider` are new (as of late 2019) `tidyr` functions.

See link below:

https://tidyr.tidyverse.org/dev/articles/pivot.html

# Pivoting Functions

```r
long2 = circ %>%
  rename_all(function(var) {
    var %>%
    str_replace("Board", "_Board") %>%
    str_replace("Alight", "_Alight") %>%
    str_replace("Average", "_Average")
  })
longer =long2 %>% pivot_longer(
  cols = matches("orange|purple|green|banner"),
  names_to = c("route", "type"),
  names_sep = "_"
)
head(longer)
```

```
# A tibble: 6 x 6
  day     date        daily route  type        value
  <chr>   <date>      <dbl> <chr>  <chr>        <dbl>
1 Monday  2010-01-11    952 orange Boardings      877
2 Monday  2010-01-11    952 orange Alightings    1027
3 Monday  2010-01-11    952 orange Average        952
4 Monday  2010-01-11    952 purple Boardings       NA
5 Monday  2010-01-11    952 purple Alightings      NA
6 Monday  2010-01-11    952 purple Average         NA
```

# Pivoting Functions

```
longer %>%
  filter(!is.na(value)) %>% # keep where there is data
  pivot_wider(
    names_from = type,
    values_from = value
  )
```

```
# A tibble: 2,884 x 7
   day       date        daily route  Boardings Alightings Average
   <chr>     <date>      <dbl> <chr>      <dbl>      <dbl>   <dbl>
 1 Monday    2010-01-11   952  orange       877       1027     952
 2 Tuesday   2010-01-12   796  orange       777        815     796
 3 Wednesday 2010-01-13  1212. orange      1203       1220    1212.
 4 Thursday  2010-01-14  1214. orange      1194       1233    1214.
 5 Friday    2010-01-15  1644  orange      1645       1643    1644
 6 Saturday  2010-01-16  1490. orange      1457       1524    1490.
 7 Sunday    2010-01-17   888. orange       839        938     888.
 8 Monday    2010-01-18  1000. orange       999       1000    1000.
 9 Tuesday   2010-01-19  1035  orange      1023       1047    1035
10 Wednesday 2010-01-20  1396. orange      1375       1416    1396.
# … with 2,874 more rows
```