R Resource

R for Data Science - <https://r4ds.had.co.nz/>

* Tidy data chapter 12 – gather(), spread()
* Relational data chapter 13—merge/join <https://r4ds.had.co.nz/relational-data.html#outer-join>

Correspondence Project Repository - <https://github.com/judgelord/correspondence>

Date Formats in R - <https://www.r-bloggers.com/date-formats-in-r/>

The tidyverse – some of the packages included <https://www.tidyverse.org/packages/>

**Useful Code Bits**

***Rename* a column** <https://stackoverflow.com/questions/7531868/how-to-rename-a-single-column-in-a-data-frame>

colnames(df)[colnames(df) == 'oldName'] <- 'newName'

**Remove duplicated rows using distinct()** <https://stackoverflow.com/questions/13967063/remove-duplicated-rows>

**Remove rows where specified columns are duplicated:**

library(dplyr)

dat %>% distinct(a, .keep\_all = TRUE)

a b

1 1 A

2 2 B

**Remove rows that are complete duplicates of other rows:**

dat %>% distinct

a b

1 1 A

2 2 B

3 1 C

4 2 D

**Add precision/decimals to rounded off values**

Use format(value, nsmall = *number of decimals you want*)

>format(120, nsmall = 4)

[1] "120.0000"

**Manual Imputation**

replace NA values in data$lat with MEDIAN

data %<>% mutate(lat = ifelse(is.na(lat), median(lat[!is.na(lat)]), lat))

**Concatenate with paste() and paste0() –** specify separator with paste(), default is a space

**>**paste("$", 99)

[1] "$ 99"

>paste0("$", 99)

[1] "$99"

**Gsub Example**

Create new variable from old variable. This example takes the 3 letters after a space from the old variable to create a new variable.

data$new <- gsub(".\* (\\w{3}$)", "\\1", flights$old)

**Long Data to Wide Data (example) – spread() function**

Data is in long, tidy form. Reduced to 3 variables: State, Price Index, YearQuarter

> head(data)

# A tibble: 6 x 3

State `Price Index` YearQuarter

<chr> <dbl> <dbl>

1 CA 18.3 1975

2 CA 18.8 1975.25

3 CA 19.4 1976.0

4 CA 20.1 1976.25

5 CA 21.0 1976.50

6 CA 22.1 1976.75

> newdata <- spread(data, State, 'Price Index')

> head(newdata)

# A tibble: 6 x 6

YearQuarter CA DC MA MI NY

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1975 18.3 23.4 21.7 41.3 27.4

2 1975. 18.8 24.5 21.0 42.8 26.4

3 1976. 19.4 24.5 21.5 43.6 27.1

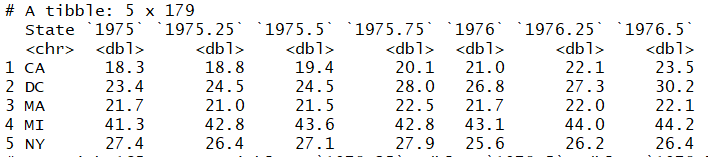
4 1976. 20.1 28.0 22.5 42.8 27.9

5 1976 21.0 26.8 21.7 43.1 25.6

6 1976. 22.1 27.3 22.0 44.0 26.2

> newdata2 <- spread(data, YearQuarter, 'Price Index')

> head(newdata2)



**Wide to Long – gather() \*\*\*\*\*\***

> head(newdata)

# A tibble: 6 x 6

YearQuarter CA DC MA MI NY

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1975 18.3 23.4 21.7 41.3 27.4

2 1975. 18.8 24.5 21.0 42.8 26.4

3 1976. 19.4 24.5 21.5 43.6 27.1

4 1976. 20.1 28.0 22.5 42.8 27.9

5 1976 21.0 26.8 21.7 43.1 25.6

6 1976. 22.1 27.3 22.0 44.0 26.2

> newdata %>% gather("CA","DC","MA","MI", "NY", key="State", value = "Price\_Index")

# A tibble: 890 x 3

YearQuarter State Price\_Index

<dbl> <chr> <dbl>

1 1975 CA 18.3

2 1975. CA 18.8

3 1976. CA 19.4

4 1976. CA 20.1

5 1976 CA 21.0

6 1976. CA 22.1

7 1976. CA 23.5

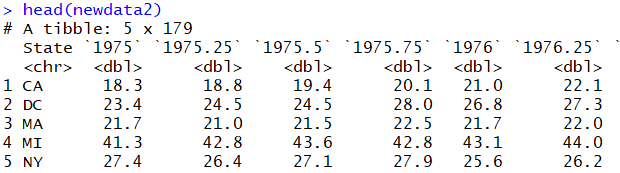
8 1977. CA 24.4

9 1977 CA 25.5

10 1977. CA 27.7

# ... with 880 more rows

OR formatted other way



> yq <- as.character(seq(1975,2019.25, by=0.25))

> newdata2 %>% gather(yq, key="Year\_Quarter", value = "Price\_Index")

# A tibble: 890 x 3

State Year\_Quarter Price\_Index

<chr> <chr> <dbl>

1 CA 1975 18.3

2 DC 1975 23.4

3 MA 1975 21.7

4 MI 1975 41.3

5 NY 1975 27.4

6 CA 1975.25 18.8

7 DC 1975.25 24.5

8 MA 1975.25 21.0

9 MI 1975.25 42.8

10 NY 1975.25 26.4

# ... with 880 more rows

**Merging / Joins**

left\_join(flights, airports, by = c("dest" = "faa"))

* + Matches flights$dest with aiports$faa. Dataframe will have same # observations as flights, but will add variables from airport where observations match
  + right\_join can always be rewritten with left\_join (i.e. right\_join(x,y) and left\_join(y,z))

**Correspondence Project Code examples**

**Using mutate() for changing column values**

|  |
| --- |
| d %<>% |
|  | group\_by(agency, ID, DATE, FROM, first\_name, last\_name) %>% mutate(n = n()) %>% |
|  | mutate(ERROR = ifelse(n >1 & (bioname == "ROGERS, Mike Dennis" | bioname == "ROGERS, Mike"), "FOIA 2 Mike Rogers's", ERROR)) %>% # 2 different members with name Mike Rogers |
|  | mutate(ERROR = ifelse(n >1 & (bioname == "JOHNSON, Timothy Peter (Tim)" | bioname == "JOHNSON, Timothy V."), "FOIA 2 Tim Johns", ERROR)) %>% |
|  | mutate(ERROR = ifelse(grepl("(^| )Biden(,| |$)", FROM)& DATE > as.Date('2009-01-19'), "Joe is VP", ERROR)) %>% |
|  | mutate(ERROR = ifelse((grepl("Eleanor|Holmes", FROM)&grepl("Norton", FROM))|(grepl("Eleanor", FROM)&grepl("Holmes", FROM)), "Non-voting DC Rep", ERROR)) %>% |
|  | mutate(ERROR = ifelse(grepl("^White House$", FROM, ignore.case=T), "White House", ERROR)) %>% |
|  | mutate(ERROR = ifelse(grepl("^Miscellaneous$", FROM, ignore.case=T), "Miscellaneous", ERROR)) |

**Remove blank spaces**

data$FROM <- gsub("^ |^ | $| $", "", data$FROM) # removes extra spaces

|  |
| --- |
|  |
|  | data %<>% select(ID, DATE, FROM, SUBJECT, everything()) |
|  |  |

**Select and order columns – use everything() if keeping all columns**