Data Cleaning: 'tidyr' Package

Mr. Sachin B.

Introduction

- `tidyr` package specially designed to tidying the data.
- `tidyr` package developed by Hadley Wickham of RStudio

Tidy data satisfies three conditions

1. Each variable forms a column
2. Each observation forms a row
3. Each type of observational unit forms a table

Following data can be consider as messy

1. Column header are values, not variable names.
2. Multiple variables are stored in one column.
3. Variables are stored in both rows and columns.
4. Multiple types of observational units are stored in the same table.
5. A single observational unit is stored in multiple tables.

Required Libraries

```
library(tidyr)
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.0.4
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.4
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

1. Column header are values, not variable names.

```
if(!file.exists("./data")){dir.create("./data")}

fileUrl <- "https://github.com/b-sachin/R-Programming/blob/main/dataset/Student1.csv?raw=true"
download.file(fileUrl,destfile = "./data/Student1.csv",method = "curl",extra = '-L')

Student1 <- read.csv("./data/Student1.csv")
```

Download data

```
print(Student1)
```

Print Dataset

```
##   grade male female
## 1     A    5      3
## 2     B    4      1
## 3     C    8      6
## 4     D    4      5
## 5     E    5      5
```

This dataset actually has three variables: **1.grade 2.gender 3.count**

- The variable `grade` is already a column, so that should remain as it is.
- The Second variable, `gender` is captured by 2nd & 3rd column headings.
- The Third variable, `count` is no. of students for each combination of `grade` & `gender`

```
gather(Student1,gender,count,-grade)
```

`gather()`

```
##    grade gender count
## 1      A   male     5
## 2      B   male     4
## 3      C   male     8
## 4      D   male     4
## 5      E   male     5
## 6      A female     3
## 7      B female     1
## 8      C female     6
## 9      D female     5
## 10     E female     5
```

```
# The data argument, 'Student1' -> gives the name of the original dataset
# The key & value arguments ['gender' & 'count'] -> gives column name for out tidy dataset.
# The final argument, '-grade' -> says that we want to gather all columns EXCEPT the grade column.
```

2. Multiple variables are stored in one column.

```
if(!file.exists("./data")){dir.create("./data")}

fileUrl <- "https://github.com/b-sachin/R-Programming/blob/main/dataset/Student2.csv?raw=true"
download.file(fileUrl,destfile = "./data/Student2.csv",method = "curl",extra = '-L')

Student2 <- read.csv("./data/Student2.csv")
```

Download data

```
print(Student2)
```

Print Dataset

```
##   grade male.1 female.1 male.2 female.2
## 1     A      7        0      5        8
## 2     B      4        0      5        8
## 3     C      7        4      5        6
## 4     D      8        2      8        1
## 5     E      8        4      1        0
```

This dataset has multiple variables stores in each column (**class** & **gender**) [**male-1**]

gather()

```
Student2_gather<-gather(Student2,gender_class,count,-grade)

print(Student2_gather)
```

```
##    grade gender_class count
## 1      A       male.1     7
## 2      B       male.1     4
## 3      C       male.1     7
## 4      D       male.1     8
## 5      E       male.1     8
## 6      A     female.1     0
## 7      B     female.1     0
## 8      C     female.1     4
## 9      D     female.1     2
```

```
## 10     E     female.1     4
## 11     A       male.2     5
## 12     B       male.2     5
## 13     C       male.2     5
## 14     D       male.2     8
## 15     E       male.2     1
## 16     A     female.2     8
## 17     B     female.2     8
## 18     C     female.2     6
## 19     D     female.2     1
## 20     E     female.2     0
```

We still have two different variables **gender** & **class**, stored together in the **gender_class** column

seperate()

Function offers separating one column into multiple column

```
separate(data=Student2_gather,col=gender_class,into=c("gender","class"))
```

```
##    grade gender class count
## 1      A   male     1     7
## 2      B   male     1     4
## 3      C   male     1     7
## 4      D   male     1     8
## 5      E   male     1     8
## 6      A female     1     0
## 7      B female     1     0
## 8      C female     1     4
## 9      D female     1     2
## 10     E female     1     4
## 11     A   male     2     5
## 12     B   male     2     5
## 13     C   male     2     5
## 14     D   male     2     8
## 15     E   male     2     1
## 16     A female     2     8
## 17     B female     2     8
## 18     C female     2     6
## 19     D female     2     1
## 20     E female     2     0
```

```
gather(Student2,gender_class,count,-grade) %>%
  separate(gender_class, c("gender","class")) %>%
  print
```

2 Step process can be shorten with %>% operator (Pipeline)

```
##    grade gender class count
```

```
## 1      A   male     1     7
## 2      B   male     1     4
## 3      C   male     1     7
## 4      D   male     1     8
## 5      E   male     1     8
## 6      A female     1     0
## 7      B female     1     0
## 8      C female     1     4
## 9      D female     1     2
## 10     E female     1     4
## 11     A   male     2     5
## 12     B   male     2     5
## 13     C   male     2     5
## 14     D   male     2     8
## 15     E   male     2     1
## 16     A female     2     8
## 17     B female     2     8
## 18     C female     2     6
## 19     D female     2     1
## 20     E female     2     0
```

3. Variables are stored in both rows & columns

```
if(!file.exists("./data")){dir.create("./data")}

fileUrl <- "https://github.com/b-sachin/R-Programming/blob/main/dataset/Student3.csv?raw=true"
download.file(fileUrl,destfile = "./data/Student3.csv",method = "curl",extra = '-L')

Student3 <- read.csv("./data/Student3.csv",na.strings="")
```

Download data

```
print(Student3)
```

Print Dataset

```
##     name    test class1 class2 class3 class4 class5
## 1  Sally Midterm      A   <NA>      B   <NA>   <NA>
## 2  Sally   Final      C   <NA>      C   <NA>   <NA>
## 3   Jeff Midterm   <NA>      D   <NA>      A   <NA>
## 4   Jeff   Final   <NA>      E   <NA>      C   <NA>
## 5  Rojer Midterm   <NA>      C   <NA>   <NA>      B
## 6  Rojer   Final   <NA>      A   <NA>   <NA>      A
## 7  Karen Midterm   <NA>   <NA>      C      A   <NA>
## 8  Karen   Final   <NA>   <NA>      C      A   <NA>
## 9  Brian Midterm      B   <NA>   <NA>   <NA>      A
## 10 Brian   Final      B   <NA>   <NA>   <NA>      C
```

In above Data Frame, We have midterm & final exam grades for five students, each of whom were enrolled in exactly two of five possible classes.

gather()

```
Student3 %>%
  gather(class,grade,class1:class5,na.rm=TRUE)
```

```
##     name    test  class grade
## 1  Sally Midterm class1     A
## 2  Sally   Final class1     C
## 9  Brian Midterm class1     B
## 10 Brian   Final class1     B
## 13  Jeff Midterm class2     D
## 14  Jeff   Final class2     E
## 15 Rojer Midterm class2     C
## 16 Rojer   Final class2     A
## 21 Sally Midterm class3     B
## 22 Sally   Final class3     C
## 27 Karen Midterm class3     C
## 28 Karen   Final class3     C
## 33  Jeff Midterm class4     A
## 34  Jeff   Final class4     C
## 37 Karen Midterm class4     A
## 38 Karen   Final class4     A
## 45 Rojer Midterm class5     B
## 46 Rojer   Final class5     A
## 49 Brian Midterm class5     A
## 50 Brian   Final class5     C
```

spread()

Function allows us to turn the values of the test column [**midterm** & **final**] into column heads

```
Student3 %>%
  gather(class,grade,class1:class5,na.rm=TRUE) %>%
  spread(test,grade) %>%
  print
```

```
##     name  class Final Midterm
## 1  Brian class1     B       B
## 2  Brian class5     C       A
## 3   Jeff class2     E       D
## 4   Jeff class4     C       A
## 5  Karen class3     C       C
## 6  Karen class4     A       A
## 7  Rojer class2     A       C
## 8  Rojer class5     A       B
## 9  Sally class1     C       A
## 10 Sally class3     C       B
```

readr Package

```
#Test
library(readr)
parse_number("class5")
```

```
## [1] 5
```

```
Student3_Output <- Student3 %>%
                    gather(class,grade,class1:class5,na.rm=TRUE) %>%
                    spread(test,grade) %>%
                    mutate(class=parse_number(class))

print(Student3_Output)
```

```
##     name class Final Midterm
## 1  Brian     1     B       B
## 2  Brian     5     C       A
## 3   Jeff     2     E       D
## 4   Jeff     4     C       A
## 5  Karen     3     C       C
## 6  Karen     4     A       A
## 7  Rojer     2     A       C
## 8  Rojer     5     A       B
## 9  Sally     1     C       A
## 10 Sally     3     C       B
```

4. Multiple types of observational units are stored in the same table.

```
if(!file.exists("./data")){dir.create("./data")}

fileUrl <- "https://github.com/b-sachin/R-Programming/blob/main/dataset/Student4.csv?raw=true"
download.file(fileUrl,destfile = "./data/Student4.csv",method = "curl",extra = '-L')

Student4 <- read.csv("./data/Student4.csv")
```

Download data

```
print(Student4)
```

Print Dataset

```
##    id  name gender class final midterm
## 1  15 Brian      F     1     B       B
## 2  15 Brian      F     5     C       A
```

```
## 3  24  Jeff      M     2     E       D
## 4  24  Jeff      M     4     C       A
## 5  34 Karen      F     3     C       C
## 6  34 Karen      F     4     A       A
## 7  25 Rojer      M     2     A       C
## 8  25 Rojer      M     5     A       B
## 9  13 Sally      F     1     C       A
## 10 13 Sally      F     3     C       B
```

Problem in Data

- At first glance, There doesn't seen to be much of a problem with `Student4`.
- All columns are variables and all rows are observations.
- However, Notice that each `id`,`name` & `gender` is repeated twice.
- This hint that our data contains multiple observational units in a single table.

Solution

1. Break Student4 into 2 separate tables
2. Student_Information -> `id`, `name` & `gender`
3. Grades -> `id`,`class`,`midterm`,`final`

```
Student_Information <- Student4 %>%
                       select(id,name,gender) %>%
                       print
```

Create Student_Information

```
##    id  name gender
## 1  15 Brian      F
## 2  15 Brian      F
## 3  24  Jeff      M
## 4  24  Jeff      M
## 5  34 Karen      F
## 6  34 Karen      F
## 7  25 Rojer      M
## 8  25 Rojer      M
## 9  13 Sally      F
## 10 13 Sally      F
```

It contain 5 duplicate rows

```
Student_Information <- Student4 %>%
                       select(id,name,gender) %>%
                       unique %>%
                       print
```

Create Student_Information with Unique rows

```
##   id  name gender
## 1 15 Brian      F
## 3 24  Jeff      M
## 5 34 Karen      F
## 7 25 Rojer      M
## 9 13 Sally      F
```

Similarly,

```
gradebook <- Student4 %>%
                select(id,class,midterm,final) %>%
                print
```

```
##    id class midterm final
## 1  15     1       B     B
## 2  15     5       A     C
## 3  24     2       D     E
## 4  24     4       A     C
## 5  34     3       C     C
## 6  34     4       A     A
## 7  25     2       C     A
## 8  25     5       B     A
## 9  13     1       A     C
## 10 13     3       B     C
```

5. Single observational unit is stored in multiple tables

```
# Student5.csv download
if(!file.exists("./data")){dir.create("./data")}

fileUrl <- "https://github.com/b-sachin/R-Programming/blob/main/dataset/Student5.csv?raw=true"

download.file(fileUrl,destfile = "./data/Student5.csv",method = "curl",extra = '-L')

Student5 <- read.csv("./data/Student5.csv")


# Student6.csv download
fileUrl <- "https://github.com/b-sachin/R-Programming/blob/main/dataset/Student6.csv?raw=true"

download.file(fileUrl,destfile = "./data/Student6.csv",method = "curl",extra = '-L')

Student6 <- read.csv("./data/Student6.csv")

# Assign Student5.csv & Student6.csv to 'Passed' & 'Failed' Variables respectively
Passed <- read.csv("Student5.csv")
Failed <- read.csv("Student6.csv")

# Print 'Passed & 'Failed'
print(Passed)
```

Download data

```
##    name class Final
## 1 Brian     1     B
## 2 Rojer     2     A
## 3 Rojer     5     A
## 4 Karen     4     A
```

```
print(Failed)
```

```
##    name class Final
## 1 Brian     5     C
## 2 Sally     1     C
## 3 Sally     3     C
## 4  Jeff     2     E
## 5  Jeff     4     C
## 6 Karen     3     C
```

- The name of each dataset actually represents the value of a new variable that we will call 'status'.
- Before joining the two tables together, we'll add a new column to each containing this information.

```
# Add Column 'Status' to Passed & Failed Table Respectively
Passed <- mutate(Passed,status="Passed")
Failed <- mutate(Failed,status="Failed")

# Print 'Passed & 'Failed'
print(Passed)
```

```
##    name class Final status
## 1 Brian     1     B Passed
## 2 Rojer     2     A Passed
## 3 Rojer     5     A Passed
## 4 Karen     4     A Passed
```

```
print(Failed)
```

```
##    name class Final status
## 1 Brian     5     C Failed
## 2 Sally     1     C Failed
## 3 Sally     3     C Failed
## 4  Jeff     2     E Failed
## 5  Jeff     4     C Failed
## 6 Karen     3     C Failed
```

```
bind_rows(Passed,Failed)
```

Combine 2 Data Frames

```
##     name class Final status
## 1  Brian     1     B Passed
## 2  Rojer     2     A Passed
## 3  Rojer     5     A Passed
## 4  Karen     4     A Passed
## 5  Brian     5     C Failed
## 6  Sally     1     C Failed
## 7  Sally     3     C Failed
## 8   Jeff     2     E Failed
## 9   Jeff     4     C Failed
## 10 Karen     3     C Failed
```