Datacamp_Cleaning Data in R_Tidying data

dizhen

2019/4/4

Introduction to tidyr

1. Summary of key tidyr functions

gather() - Gather columns into key-value pairs (wide -> long)

spread() - Spread key-value pairs into columns (long -> wide)

separate() - Separate one column into multiple

unite() - Unite multiple columns into one

```
library("tidyr")
wide_df <- data.frame(col = c("X","Y"), A = c(1,4), B = c(2,5), C= c(3,6))

# Gather the columns of wide_df
long_df<-gather(wide_df, my_key, my_val, -col)
long_df
```

```
##   col my_key my_val
## 1   X      A      1
## 2   Y      A      4
## 3   X      B      2
## 4   Y      B      5
## 5   X      C      3
## 6   Y      C      6
```

```
# Spread the key-value pairs of long_df
spread(long_df, my_key, my_val)
```

```
##   col A B C
## 1   X 1 2 3
## 2   Y 4 5 6
```

```
treatments <- data.frame(patient= c("X","Y","X","Y"), treatment = c("A","A","B","B"), year_mo = c("2010

# Separate year_mo into two columns
treatments_sep<-separate(treatments, year_mo, c("year", "month"))
treatments_sep
```

```
##   patient treatment year month response
## 1       X         A 2010    10        1
## 2       Y         A 2010    10        4
## 3       X         B 2012    08        2
## 4       Y         B 2012    08        5
```

```
# Unite year and month to form year_mo column
unite(treatments_sep, year_mo, year, month)
```

```
##   patient treatment year_mo response
## 1       X         A 2010_10        1
## 2       Y         A 2010_10        4
## 3       X         B 2012_08        2
## 4       Y         B 2012_08        5
```

Practice

```
library(readr)
bmi <- read_csv("data/bmi_clean.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Country = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
# Apply gather() to bmi and save the result as bmi_long
bmi_long <- gather(bmi,year, bmi_val, -Country)

# View the first 8 rows of the result
head(bmi_long, n=8)
```

```
## # A tibble: 8 x 3
##   Country             year  bmi_val
##   <chr>               <chr>   <dbl>
## 1 Afghanistan         Y1980    21.5
## 2 Albania             Y1980    25.2
## 3 Algeria             Y1980    22.3
## 4 Andorra             Y1980    25.7
## 5 Angola              Y1980    20.9
## 6 Antigua and Barbuda Y1980    23.3
## 7 Argentina           Y1980    25.4
## 8 Armenia             Y1980    23.8
```

```
# Apply spread() to bmi_long
bmi_wide <- spread(bmi_long, year, bmi_val)

# View the head of bmi_wide
head(bmi_long)
```

```
## # A tibble: 6 x 3
##   Country             year  bmi_val
##   <chr>               <chr>   <dbl>
## 1 Afghanistan         Y1980    21.5
## 2 Albania             Y1980    25.2
## 3 Algeria             Y1980    22.3
## 4 Andorra             Y1980    25.7
## 5 Angola              Y1980    20.9
## 6 Antigua and Barbuda Y1980    23.3
```

```
head(bmi_wide)
```

```
## # A tibble: 6 x 30
##   Country Y1980 Y1981 Y1982 Y1983 Y1984 Y1985 Y1986 Y1987 Y1988 Y1989 Y1990
##   <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Afghan~  21.5  21.5  21.5  21.4  21.4  21.4  21.4  21.4  21.3  21.3  21.2
## 2 Albania  25.2  25.2  25.3  25.3  25.3  25.3  25.3  25.3  25.3  25.3  25.3
## 3 Algeria  22.3  22.3  22.4  22.5  22.6  22.7  22.8  22.8  22.9  23.0  23.0
## 4 Andorra  25.7  25.7  25.7  25.8  25.8  25.9  25.9  25.9  26.0  26.0  26.1
## 5 Angola   20.9  20.9  20.9  20.9  20.9  20.9  21.0  21.0  21.0  21.1  21.1
## 6 Antigu~  23.3  23.4  23.5  23.5  23.6  23.7  23.8  23.9  24.1  24.2  24.3
## # ... with 18 more variables: Y1991 <dbl>, Y1992 <dbl>, Y1993 <dbl>,
## #   Y1994 <dbl>, Y1995 <dbl>, Y1996 <dbl>, Y1997 <dbl>, Y1998 <dbl>,
## #   Y1999 <dbl>, Y2000 <dbl>, Y2001 <dbl>, Y2002 <dbl>, Y2003 <dbl>,
## #   Y2004 <dbl>, Y2005 <dbl>, Y2006 <dbl>, Y2007 <dbl>, Y2008 <dbl>
```

```
# Apply separate() to bmi_cc
bmi_cc_clean <- separate(bmi_cc, col = Country_ISO, into = c("Country", "ISO"), sep = "/")

# Print the head of the result
head(bmi_cc_clean)


# Apply unite() to bmi_cc_clean
bmi_cc <- unite(bmi_cc_clean, Country_ISO, Country, ISO, sep = "-")

# View the head of the result
head(bmi_cc)
```

```
library(readr)
census <- read_csv("data/census-retail.csv")
```

```
## Parsed with column specification:
## cols(
##   YEAR = col_double(),
##   JAN = col_double(),
##   FEB = col_double(),
##   MAR = col_double(),
##   APR = col_double(),
##   MAY = col_double(),
##   JUN = col_double(),
##   JUL = col_double(),
##   AUG = col_double(),
##   SEP = col_double(),
##   OCT = col_double(),
##   NOV = col_double(),
##   DEC = col_double()
## )
```

```
# View the head of census
head(census)
```

```
## # A tibble: 6 x 13
##    YEAR    JAN    FEB    MAR    APR    MAY    JUN    JUL    AUG    SEP
##   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1  1992 146913 147270 146831 148082 149015 149821 150809 151064 152595
## 2  1993 157525 156292 154774 158996 160624 160171 162832 162491 163285
## 3  1994 167504 169652 172775 173099 172340 174307 174801 177289 178776
## 4  1995 182423 179472 180996 181702 183543 186088 185470 186814 187338
## 5  1996 189167 192269 193993 194712 196210 196127 196229 196215 198843
## 6  1997 202414 204273 204965 203372 201676 204666 207049 207643 208298
## # ... with 3 more variables: OCT <dbl>, NOV <dbl>, DEC <dbl>
```

```
# Gather the month columns
census2 <- gather(census,month,amount,-YEAR)

# Arrange rows by YEAR using dplyr's arrange
library("dplyr")
```

```
##
## 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
```

```
census2_arr <- arrange(census2, YEAR)

# View first 8 rows of census2
head(census2_arr,n=8)
```

```
## # A tibble: 8 x 3
##    YEAR month amount
##   <dbl> <chr>  <dbl>
## 1  1992 JAN   146913
## 2  1992 FEB   147270
## 3  1992 MAR   146831
## 4  1992 APR   148082
## 5  1992 MAY   149015
## 6  1992 JUN   149821
## 7  1992 JUL   150809
## 8  1992 AUG   151064
```

```
# View first 8 rows of census_long
head(census,n=8)
```

```
## # A tibble: 8 x 13
##    YEAR    JAN    FEB    MAR    APR    MAY    JUN    JUL    AUG    SEP
##   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1  1992 146913 147270 146831 148082 149015 149821 150809 151064 152595
## 2  1993 157525 156292 154774 158996 160624 160171 162832 162491 163285
## 3  1994 167504 169652 172775 173099 172340 174307 174801 177289 178776
## 4  1995 182423 179472 180996 181702 183543 186088 185470 186814 187338
## 5  1996 189167 192269 193993 194712 196210 196127 196229 196215 198843
## 6  1997 202414 204273 204965 203372 201676 204666 207049 207643 208298
## 7  1998 209684 209532 210792 213623 214619 216324 214853 213669 215712
## 8  1999 224020 226240 227407 228978 231238 231926 233933 236589 237516
## # ... with 3 more variables: OCT <dbl>, NOV <dbl>, DEC <dbl>
```

```
census_long <- census2_arr
census_long$type <- factor(c(rep("MED", nrow(census_long)/3),rep("HIGH", nrow(census_long)/3),rep("LOW"

# Spread the type column
census_long2 <- spread(census_long,type,amount)

# View first 8 rows of census_long2
head(census_long2,n=8)
```

```
## # A tibble: 8 x 5
##    YEAR month  HIGH   LOW    MED
##   <dbl> <chr> <dbl> <dbl>  <dbl>
## 1  1992 APR      NA    NA 148082
## 2  1992 AUG      NA    NA 151064
## 3  1992 DEC      NA    NA 155504
## 4  1992 FEB      NA    NA 147270
## 5  1992 JAN      NA    NA 146913
## 6  1992 JUL      NA    NA 150809
## 7  1992 JUN      NA    NA 149821
## 8  1992 MAR      NA    NA 146831
```

```
tail(census_long2,n=8)
```

```
## # A tibble: 8 x 5
##    YEAR month  HIGH    LOW   MED
##   <dbl> <chr> <dbl>  <dbl> <dbl>
## 1  2015 JAN      NA 383889    NA
## 2  2015 JUL      NA 395100    NA
## 3  2015 JUN      NA 391955    NA
## 4  2015 MAR      NA 387665    NA
## 5  2015 MAY      NA 392268    NA
## 6  2015 NOV      NA 395160    NA
## 7  2015 OCT      NA 394562    NA
## 8  2015 SEP      NA 394429    NA
```

```r
# View the head of census_long3
head(census_long3)

# Separate the yr_month column into two
census_long4 <- separate(census_long3,yr_month, into = c("year","month"))

# View the first 6 rows of the result
head(census_long4, n = 6)
```