Manipulação de Dados - Parte III

Paulo Henrique S. Guimarães

1 Transformação de dados e comandos básicos

1.1 Pacote tidyr

O paconte tidyr é outro pacote do tidyverse focado no manuseio de dados. Seu objetivo é transformar datasets no formato *tidy*.

– "Conjuntos de dados arrumados são todos iguais, mas cada conjunto de dados bagunçado é bagunçado de sua maneira" - Hadley Wickham.

Colocar os dados no formado tidyr (dados arrumados) que possibilita de maneira fácil realizar inúmeras operações mais facilmente dentro do tidyverse.

Figure 1: Pacote tidyr.

Há três regras inter-relacionadas que tornam um conjunto de dados no formato tidy: 1) Cada variável deve ter sua própria coluna; 2) Cada observação deve ter a sua própria linha; 3) Cada valor deve ter sua própria célula.

Esse inter-relacionamento leva a um conjunto ainda mais simples de instruções práticas que são: 1) Coloque cada conjunto de dados em um tibble; 2) Coloque cada variáveis em uma coluna.

Nota: dplyr,ggplot2 e todos os pacotes do tidyverse são projetados para trabalhar com dados 'arrumados'.

1.1.1 Função gather

O conjunto de colunas que serão transformadas de colunas para linhas; O nome da variável (coluna) cujos valores serão as colunas transformadas acima; O nome da variável cujos valores serão os valores correspondentes das colunas transformadas.

```
library(ggplot2)
library(tidyr)

data(economics)
head(economics)
```

```
## # A tibble: 6 x 6
##   date         pce    pop psavert uempmed unemploy
##   <date>     <dbl>  <dbl>   <dbl>   <dbl>    <dbl>
## 1 1967-07-01  507. 198712    12.6     4.5     2944
## 2 1967-08-01  510. 198911    12.6     4.7     2945
## 3 1967-09-01  516. 199113    11.9     4.6     2958
## 4 1967-10-01  512. 199311    12.9     4.9     3143
## 5 1967-11-01  517. 199498    12.8     4.7     3066
## 6 1967-12-01  525. 199657    11.8     4.8     3018
```

```
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
```

```
glimpse(economics)
```

```
## Observations: 574
## Variables: 6
## $ date     <date> 1967-07-01, 1967-08-01, 1967-09-01, 1967-10-01, 1967...
## $ pce      <dbl> 506.7, 509.8, 515.6, 512.2, 517.4, 525.1, 530.9, 533....
## $ pop      <dbl> 198712, 198911, 199113, 199311, 199498, 199657, 19980...
## $ psavert  <dbl> 12.6, 12.6, 11.9, 12.9, 12.8, 11.8, 11.7, 12.3, 11.7,...
## $ uempmed  <dbl> 4.5, 4.7, 4.6, 4.9, 4.7, 4.8, 5.1, 4.5, 4.1, 4.6, 4.4...
## $ unemploy <dbl> 2944, 2945, 2958, 3143, 3066, 3018, 2878, 3001, 2877,...
```

A função **gather** possui três argumentos obrigatórios: *data*, que é o nome do objeto no qual está o banco; *key*, que é o nome que iremos dar à variável que armazenará o nome das variáveis reunidas; e *value*, que é o nome da variável que armazenará os valores das variáveis reunidas.

```
dados1<- economics %>% gather(indicador, valor, -date)
head(dados1)
```

```
## # A tibble: 6 x 3
##   date       indicador valor
##   <date>     <chr>     <dbl>
## 1 1967-07-01 pce        507.
## 2 1967-08-01 pce        510.
## 3 1967-09-01 pce        516.
## 4 1967-10-01 pce        512.
## 5 1967-11-01 pce        517.
## 6 1967-12-01 pce        525.
```

```
dados2<- economics %>% gather(indicador, valor, -date,-pop)
head(dados2)
```

```
## # A tibble: 6 x 4
##   date          pop indicador valor
##   <date>      <dbl> <chr>     <dbl>
## 1 1967-07-01 198712 pce        507.
## 2 1967-08-01 198911 pce        510.
## 3 1967-09-01 199113 pce        516.
## 4 1967-10-01 199311 pce        512.
## 5 1967-11-01 199498 pce        517.
## 6 1967-12-01 199657 pce        525.
```

A função **gather** "empilha" o banco de dados. Com isso, o *dataset* passou do formato *wide* (menos linhas e mais colunas) em um formato *long* (mais linhas e menos colunas). • Em algumas situações, porém, será necessário fazer o contrário: transformar o *dataset* de *long* para *wide*. Para isso, usa-se a função *spread*():

```
data(economics_long)
head(economics_long)
```

```
## # A tibble: 6 x 4
##   date       variable value  value01
##   <date>     <chr>    <dbl>    <dbl>
## 1 1967-07-01 pce       507. 0
## 2 1967-08-01 pce       510. 0.000265
## 3 1967-09-01 pce       516. 0.000762
## 4 1967-10-01 pce       512. 0.000471
## 5 1967-11-01 pce       517. 0.000916
## 6 1967-12-01 pce       525. 0.00157
```

```
economics_long %>%
select(-value01) %>%
spread(variable, value, fill = NA)
```

```
## # A tibble: 574 x 6
##    date         pce    pop psavert uempmed unemploy
##    <date>     <dbl>  <dbl>   <dbl>   <dbl>    <dbl>
##  1 1967-07-01  507. 198712    12.6     4.5     2944
##  2 1967-08-01  510. 198911    12.6     4.7     2945
##  3 1967-09-01  516. 199113    11.9     4.6     2958
##  4 1967-10-01  512. 199311    12.9     4.9     3143
##  5 1967-11-01  517. 199498    12.8     4.7     3066
##  6 1967-12-01  525. 199657    11.8     4.8     3018
##  7 1968-01-01  531. 199808    11.7     5.1     2878
##  8 1968-02-01  534. 199920    12.3     4.5     3001
##  9 1968-03-01  544. 200056    11.7     4.1     2877
## 10 1968-04-01  544  200208    12.3     4.6     2709
## # ... with 564 more rows
```

Outras duas funções importantes deste pacote são: *separate* e *unite*, que separa uma coluna em duas e vice versa.

```r
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------ tidyverse 1.2.1 --

## v tibble  2.1.3     v purrr   0.3.2
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.3     v forcats 0.4.0

## -- Conflicts --------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(tidyr)
library(readxl)

homic <- read_excel("homicidios_uf.xls")
head(homic)
```

```
## # A tibble: 6 x 13
##   Sigla Codigo Estado `2000` `2001` `2002` `2003` `2004` `2005` `2006`
##   <chr> <chr>  <chr>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 AC    12     Acre      108    122    151    135    115    125    155
## 2 AL    27     Alago~    724    836    989   1041   1034   1211   1617
## 3 AM    13     Amazo~    559    478    512    561    523    598    697
## 4 AP    16     Amapá     155    184    181    190    173    196    203
## 5 BA    29     Bahia    1223   1573   1735   2155   2255   2823   3276
## 6 CE    23     Ceará    1229   1298   1443   1560   1538   1692   1793
## # ... with 3 more variables: `2007` <dbl>, `2008` <dbl>, `2009` <dbl>
```

```r
glimpse(homic)
```

```
## Observations: 27
## Variables: 13
## $ Sigla  <chr> "AC", "AL", "AM", "AP", "BA", "CE", "DF", "ES", "GO", "...
## $ Codigo <chr> "12", "27", "13", "16", "29", "23", "53", "32", "52", "...
## $ Estado <chr> "Acre", "Alagoas", "Amazonas", "Amapá", "Bahia", "Ceará...
## $ `2000` <dbl> 108, 724, 559, 155, 1223, 1229, 770, 1449, 1008, 344, 2...
## $ `2001` <dbl> 122, 836, 478, 184, 1573, 1298, 773, 1472, 1087, 536, 2...
## $ `2002` <dbl> 151, 989, 512, 181, 1735, 1443, 744, 1639, 1275, 576, 2...
## $ `2003` <dbl> 135, 1041, 561, 190, 2155, 1560, 856, 1640, 1259, 762, ...
## $ `2004` <dbl> 115, 1034, 523, 173, 2255, 1538, 815, 1630, 1427, 696, ...
## $ `2005` <dbl> 125, 1211, 598, 196, 2823, 1692, 745, 1600, 1398, 903, ...
## $ `2006` <dbl> 155, 1617, 697, 203, 3276, 1793, 769, 1774, 1409, 925, ...
## $ `2007` <dbl> 133, 1839, 711, 171, 3608, 1936, 815, 1885, 1426, 1091,...
## $ `2008` <dbl> 133, 1887, 827, 211, 4750, 2031, 873, 1948, 1754, 1243,...
## $ `2009` <dbl> 149, 1868, 916, 191, 5345, 2166, 1003, 1969, 1784, 1391...
```

```r
# Reune as variáveis de ano espalhadas pela base 'homic'

homic <- gather(homic, Ano, Homicidios, -Sigla, -Codigo,-Estado)
head(homic)
```

```
## # A tibble: 6 x 5
##   Sigla Codigo Estado   Ano   Homicidios
##   <chr> <chr>  <chr>    <chr>      <dbl>
## 1 AC    12     Acre     2000         108
## 2 AL    27     Alagoas  2000         724
## 3 AM    13     Amazonas 2000         559
## 4 AP    16     Amapá    2000         155
## 5 BA    29     Bahia    2000        1223
## 6 CE    23     Ceará    2000        1229
```

```
# ou da forma:

homic <- read_excel("homicidios_uf.xls")
homic <- gather(data = homic, key = Data, value = Homicidios, -Sigla, -Codigo,-Estado)
head(homic)
```

```
## # A tibble: 6 x 5
##   Sigla Codigo Estado   Data  Homicidios
##   <chr> <chr>  <chr>    <chr>      <dbl>
## 1 AC    12     Acre     2000         108
## 2 AL    27     Alagoas  2000         724
## 3 AM    13     Amazonas 2000         559
## 4 AP    16     Amapá    2000         155
## 5 BA    29     Bahia    2000        1223
## 6 CE    23     Ceará    2000        1229
```

```
homic <- spread(homic, Data, Homicidios)
head(homic)
```

```
## # A tibble: 6 x 13
##   Sigla Codigo Estado `2000` `2001` `2002` `2003` `2004` `2005` `2006`
##   <chr> <chr>  <chr>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 AC    12     Acre      108    122    151    135    115    125    155
## 2 AL    27     Alago~    724    836    989   1041   1034   1211   1617
## 3 AM    13     Amazo~    559    478    512    561    523    598    697
## 4 AP    16     Amapá     155    184    181    190    173    196    203
## 5 BA    29     Bahia    1223   1573   1735   2155   2255   2823   3276
## 6 CE    23     Ceará    1229   1298   1443   1560   1538   1692   1793
## # ... with 3 more variables: `2007` <dbl>, `2008` <dbl>, `2009` <dbl>
```

A função **separate()** separa uma coluna em várias outras ao dividir sempre que um caractere separador aparece.

A função **unite()** é o inverso de **separete()**, isto é, combina várias colunas em uma única.

```
z<-tibble(k=c("a,b,c","d,e,f","h,j")) %>%
separate(k,c("um","dois","três"))
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 1 rows [3].
```

```r
df <- tibble(x = c("a",1,2,3), y = c("b", 1,3,5))
w<- df %>% unite("z", x:y, remove = FALSE)
head(w)
```

```
## # A tibble: 4 x 3
##   z     x     y
##   <chr> <chr> <chr>
## 1 a_b   a     b
## 2 1_1   1     1
## 3 2_3   2     3
## 4 3_5   3     5
```

```r
w %>% separate(z, c("x", "y"))
```

```
## # A tibble: 4 x 2
##   x     y
##   <chr> <chr>
## 1 a     b
## 2 1     1
## 3 2     3
## 4 3     5
```

2 Referências

1) Livro: R for Data Science - Hadley Wickham & Garrett Grolemund. Alta Books, 2019.

2) https://www.ufrgs.br/wiki-r/index.php?title=Manipulando_Dados_com_dplyr_e_tidyr