MAD-CB

Data Frames com Colunas de Listas

Fonte

Vem do blog post "Take Your Data Frames to the Next Level" de Kiefer Smith do blog "Real Data: Adventures in Data Science"; url: `https://realdataweb.wordpress.com/2017/03/31/take-your-data-frames-to-the-next-level/`

Operações com Colunas das Listas

- Podemos estender a estrutura de data.frame utilizando funções de dplyr
  - para criar colunas que são listas
- Uso de 2 novas funções
  - `group_by()` – agrupar a data frame por uma variável de classe `factor`
  - `nest()` – criar novas colunas por grupo com as outras como ítens na lista

Dataset Simples e Tradicional - Íris

- Data frame que demonstra as caracteristicas da flor íris
- Um de mais tradicionais na historia de estatística
- Elaborado pelo Ronald Fisher
  - Aquele de ANOVA e o Exact Test

Criar Colunas de Listas Agrupadas por Especie

```
irisnin <- iris %>%
           group_by(Species) %>%
           nest()
```

Resultado de Agrupamento

```
irisnin
```

```
## # A tibble: 3 × 2
##      Species              data
##       <fctr>            <list>
## 1     setosa <tibble [50 × 4]>
## 2 versicolor <tibble [50 × 4]>
## 3  virginica <tibble [50 × 4]>
```

```
head(irisnin$data[[1]])
```

```
## # A tibble: 6 × 4
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
##          <dbl>       <dbl>        <dbl>       <dbl>
## 1          5.1         3.5          1.4         0.2
## 2          4.9         3.0          1.4         0.2
## 3          4.7         3.2          1.3         0.2
## 4          4.6         3.1          1.5         0.2
## 5          5.0         3.6          1.4         0.2
## 6          5.4         3.9          1.7         0.4
```

Usar Funções purrr::map() para Fazer Calculos sem Loops – Média das Variáveis

```
mediasiris <- map(irisnin$data, colMeans)
mediasiris
```

```
## [[1]]
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        5.006        3.428        1.462        0.246
##
## [[2]]
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        5.936        2.770        4.260        1.326
##
## [[3]]
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        6.588        2.974        5.552        2.026
```

Reverter os Dados para o Estado Original para Trabalhar com os Resultados

```
head(unnest(irisnin))
```

```
## # A tibble: 6 × 5
##   Species Sepal.Length Sepal.Width Petal.Length Petal.Width
##    <fctr>        <dbl>       <dbl>        <dbl>       <dbl>
## 1  setosa          5.1         3.5          1.4         0.2
## 2  setosa          4.9         3.0          1.4         0.2
## 3  setosa          4.7         3.2          1.3         0.2
## 4  setosa          4.6         3.1          1.5         0.2
## 5  setosa          5.0         3.6          1.4         0.2
## 6  setosa          5.4         3.9          1.7         0.4
```

Gapminder Dataset

- Caracteristicas dos paises

```
library(gapminder)
glimpse(gapminder)
```

```
## Observations: 1,704
## Variables: 6
## $ country   <fctr> Afghanistan, Afghanistan, Afghanistan, Afghanistan,...
## $ continent <fctr> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asi...
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992...
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.8...
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 1488...
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 78...
```

Agrupar os Dados de gapminder por continent

```
contninho <- gapminder %>%
             select(-year) %>%
             group_by(continent, country)

glimpse(contninho)
```

```
## Observations: 1,704
## Variables: 5
## $ country   <fctr> Afghanistan, Afghanistan, Afghanistan, Afghanistan,...
## $ continent <fctr> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asi...
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.8...
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 1488...
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 78...
```

```
head(contninho)
```

```
## Source: local data frame [6 x 5]
## Groups: continent, country [1]
##
##       country continent lifeExp      pop gdpPercap
##        <fctr>    <fctr>   <dbl>    <int>     <dbl>
## 1 Afghanistan      Asia  28.801  8425333  779.4453
## 2 Afghanistan      Asia  30.332  9240934  820.8530
```

Para Evitar um Loop, Fazer um nest()

```
contninho <- contninho %>% nest()
glimpse(contninho)
```

```
## Observations: 142
## Variables: 3
## $ continent <fctr> Asia, Europe, Africa, Africa, Americas, Oceania, Eu...
## $ country   <fctr> Afghanistan, Albania, Algeria, Angola, Argentina, A...
## $ data      <list> [<c("28.801", "30.332", "31.997", "34.020", "36.088...
```

```
head(contninho)
```

```
## # A tibble: 6 × 3
##   continent     country              data
##      <fctr>      <fctr>            <list>
## 1      Asia Afghanistan <tibble [12 × 3]>
## 2    Europe     Albania <tibble [12 × 3]>
## 3    Africa     Algeria <tibble [12 × 3]>
## 4    Africa      Angola <tibble [12 × 3]>
## 5  Americas   Argentina <tibble [12 × 3]>
## 6   Oceania   Australia <tibble [12 × 3]>
```

Calcular Médias para as Variáveis Numéricas

```
mediasgap <- map(contninho$data, colMeans)
mediasgap[[1]]
```

```
##         lifeExp             pop       gdpPercap
##        37.47883 15823715.41667       802.67460
```