dplyr

Jędrzej Rybczyński

7/29/2021

1. Zbiór danych airquality zawiera informacje o warunkach pogodowych w Nowym Yorku od maja do września 1973 roku. Wykonaj poniższy fragment kodu (pamiętaj o odpowiednich bibliotekach) aby przygotować dane do dalszych ćwiczeń.

```
airquality %>%
  select(Temp, Month, Day) %>%
  as_tibble() -> data.set
```

(a) Przekształć dane w postać szeroką. Kluczem powinien być miesiąc, a wartościami temperatura.

```
data.set %>%
  pivot_wider(names_from = 'Month', values_from = 'Temp') -> df.wide
```

(b) Dane szerokie z poprzedniego podpunktu przywróć do postaci długiej.

```
df.wide %>%
  pivot_longer(cols = -Day, names_to = 'Month', values_to = 'Temp')
```

```
## # A tibble: 155 x 3
##      Day Month  Temp
##    <int> <chr> <int>
##  1     1 5        67
##  2     1 6        78
##  3     1 7        84
##  4     1 8        81
##  5     1 9        91
##  6     2 5        72
##  7     2 6        74
##  8     2 7        85
##  9     2 8        81
## 10     2 9        92
## # ... with 145 more rows
```

(c) Połącz zmienne Day oraz Month w nową zmienną Date o formacie %d.%m.

```
data.set %>%
  unite('Date', Day:Month, sep = '.') -> df.Date
```

(d) Podziel uprzednio utworzoną zmienną Date na dwie zmienne: Day oraz Month.

```
df.Date %>%
  separate(Date, into = c('Day', 'Month'), sep = '\\.')
```

```
## # A tibble: 153 x 3
##     Temp Day   Month
##    <int> <chr> <chr>
##  1    67 1     5
##  2    72 2     5
##  3    74 3     5
##  4    62 4     5
##  5    56 5     5
##  6    66 6     5
##  7    65 7     5
##  8    59 8     5
##  9    61 9     5
## 10    69 10    5
## # ... with 143 more rows
```

(e) Wygeneruj pięć braków w danych za pomocą poniższego kodu. Zastąp braki danych (NA) przez Unknown.

```
set.seed(1000)
data.set[sample(nrow(data.set), 5, replace = FALSE), 'Temp'] <- NA
```

```
data.set %>%
  replace_na(list(Temp = 'Unknown', Month = NA, Day = NA)) -> check
```

(f) Zastąp braki w danych za pomocą uzupełniania przez ostatnią zaobserwowaną wartość.

```
data.set %>%
  fill(Temp, .direction = 'downup') -> data.set
```

2. Wszystkie polecenia w tym zadaniu dotyczą zbioru auta2012 z pakietu PogromcyDanych.

(a) Ile cech jest cechami jakościowymi?

```
sum(sapply(auta2012, function(x) is.factor(x)))
```

```
## [1] 14
```

(b) Która marka samochodów jest najpopularniejsza?

```
auta2012 %>%
  group_by(Brand) %>%
  summarise(n = n()) %>%
  arrange(-n) %>%
  top_n(1)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## Selecting by n
```

```
## # A tibble: 1 x 2
##   Brand          n
##   <fct>      <int>
## 1 Volkswagen 22826
```

(c) Ile procent samochodów jest napędzane benzyną?

```
auta2012 %>%
  group_by(Type.of.fuel) %>%
  summarise(n = n()) %>%
  arrange(-n) %>%
  mutate(Perc = round(n / nrow(auta2012) * 100, 2)) %>%
  select(Type.of.fuel, Perc)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 2
##   Type.of.fuel            Perc
##   <fct>                  <dbl>
## 1 olej napedowy (diesel) 55.4
## 2 benzyna                39.4
## 3 benzyna+LPG             5.15
## 4 hybryda                 0.09
## 5 naped elektryczny       0.03
## 6 etanol                  0.01
```

(d) Ile aut od 2000 PLN?

```
auta2012 %>%
  count(Price.in.PLN >= 2000)
```

```
##   Price.in.PLN >= 2000      n
## 1                FALSE   4041
## 2                 TRUE 203561
```

(e) Ile procent aut ma pojemność silnika większą bądź równą 1500 cm3?

```
auta2012 %>%
  filter(Engine.cubic.capacity >= 1500) %>%
  summarise(Perc = (n() / nrow(auta2012)) * 100)
```

```
##       Perc
## 1 76.13173
```

(f) Ile aut zostało zarejestrowanych w Polsce i jest tańsze od 2000 PLN?

```
auta2012 %>%
  filter(Price.in.PLN < 2000, Country.of.current.registration == 'Polska') %>%
  summarise(number = n())
```

```
##   number
## 1   2018
```

(g) Ile procent aut ma pojemność silnika większą od 1500 cm3 i jest dieslem.

```
auta2012 %>%
  filter(Engine.cubic.capacity > 1500,
  Type.of.fuel == 'olej napedowy (diesel)') %>%
  summarise(perc = (n() / nrow(auta2012)) * 100)
```

```
##       perc
## 1 48.10166
```

(h) Wybierz jedynie auta marki Volkswagen. Dla tak wybranych danych utwórz tablicę kontyngencji dla zmiennej Type.of.fuel.

```
auta2012 %>%
  filter(Brand == 'Volkswagen') %>%
  count(Type.of.fuel)
```

```
##             Type.of.fuel     n
## 1                benzyna  6409
## 2             benzyna+LPG   931
## 3                  etanol     2
## 4                hybryda     1
## 5      naped elektryczny     7
## 6 olej napedowy (diesel) 15476
```

(i) Wybierz jedynie auta marki Volkswagen. Dla tak wybranych danych wyznacz średnią cenę i średni przebieg.

```
auta2012 %>%
  filter(Brand == 'Volkswagen') %>%
  summarise(mean.price = mean(Price.in.PLN),
            mean.mileage = mean(Mileage, na.rm = TRUE))
```

```
##   mean.price mean.mileage
## 1   27797.71     158411.1
```

(j) Wyznacz średnią cenę dla każdej marki.

```
auta2012 %>%
  group_by(Brand) %>%
  summarise(mean.price = mean(Price.in.PLN), n = n()) %>%
  arrange(desc(mean.price))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 106 x 3
##    Brand       mean.price     n
##    <fct>            <dbl> <int>
##  1 Bugatti       7200811.     4
##  2 Maybach        999950.     9
##  3 Ferrari        578691.    96
##  4 Lamborghini    537286.    29
##  5 AstonMartin    505360.    35
##  6 Bentley        478484.    39
##  7 Porsche        270814.   937
##  8 Rolls-Royce    204566.    19
##  9 Maserati       201047.    43
## 10 Infiniti       139807.   190
## # ... with 96 more rows
```

(k) Wybierz jedynie auta Toyota Corolla. Dla tak wybranych danych wyznacz pierwszy i trzeci kwartyl ceny.

```
auta2012 %>%
  filter(Brand == 'Toyota', Model == 'Corolla') %>%
  summarise(Q1 = quantile(Price.in.PLN, 0.25),
            Q3 = quantile(Price.in.PLN, 0.75))
```

```
##         Q1       Q3
## 1 11852.95 26815.22
```

(l) Wybierz jedynie auta marki Toyota. Dla tak wybranych danych, dla każdego modelu wyznacz średnią cenę. Wyniki przedstaw posortowane w kolejności malejącej.

```
auta2012 %>%
  filter(Brand == 'Toyota') %>%
  group_by(Model) %>%
  summarise(mean.price = mean(Price.in.PLN), n = n()) %>%
  filter(n >= 50) %>%
  arrange(-mean.price)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 17 x 3
##    Model         mean.price     n
##    <fct>              <dbl> <int>
##  1 Land Cruiser     119977.   327
##  2 Sienna            89788.    54
##  3 Verso             67795.    58
##  4 Hilux             56614.    61
##  5 Prius             56111.    59
```

```
##  6 Camry             46952.   103
##  7 RAV-4             43634.   897
##  8 Auris             40727.   435
##  9 Avensis           33479.  1550
## 10 Corolla Verso     33141.   508
## 11 Avensis Verso     25446.   105
## 12 Corolla           21424.  1191
## 13 Aygo              19987.   154
## 14 Yaris             19649.  1552
## 15 Yaris Verso       13701.    66
## 16 Celica            13087.   146
## 17 Carina             4233.    66
```

(m) Wybierz auta Volkswagen Passat z roku 2006. Dla tak wybranych danych wyznacz średnią cenę. Ile spośród wybranych aut jest tańsze od 35 000 PLN?

```
auta2012 %>%
  filter(Brand == 'Volkswagen', Model == 'Passat', Year == 2006) %>%
  summarise(mean.price = mean(Price.in.PLN),
            cheaper.than.35000 = sum(Price.in.PLN < 35000),
            count = n())
```

```
##   mean.price cheaper.than.35000 count
## 1   36694.95               222   652
```

(n) Wybierz jedynie auta z roku 2007. Dla tak wybranych danych ile mamy aut każdej marki? Przedstaw wyniki w postaci posortowanej (kolejność rosnąca) po wielkości każdej grupy.

```
auta2012 %>%
  filter(Year == 2007) %>%
  count(Brand) %>%
  arrange(n) %>%
  as_tibble()
```

```
## # A tibble: 67 x 2
##    Brand             n
##    <fct>         <int>
##  1 ""                1
##  2 "Brilliance"      1
##  3 "Isuzu"           1
##  4 "Mahindra"        1
##  5 "Mercury"         1
##  6 "Pontiac"         1
##  7 "Santana"         1
##  8 "GMC"             2
##  9 "Lamborghini"     2
## 10 "Lotus"           2
## # ... with 57 more rows
```

3. Zbiór danych Fertility z pakietu AER zawiera informacje na temat zamężnych kobiet w wieku 21-35 lat, które posiadają dwoje lub więcej dzieci (spis z roku 1980 w USA).

(a) Przyjrzyj się danym wykorzystując np. polecenie glimpse().

```
data(Fertility, package = 'AER')
glimpse(Fertility)
```

```
## Rows: 254,654
## Columns: 8
## $ morekids <fct> no, no, no, no, no, no, no, no, no, no, yes, no, no, no, n...
## $ gender1  <fct> male, female, male, male, female, male, female, male, fema...
## $ gender2  <fct> female, male, female, female, female, female, male, male, ...
## $ age      <int> 27, 30, 27, 35, 30, 26, 29, 33, 29, 27, 28, 28, 35, 34, 32...
## $ afam     <fct> no, no, no, yes, no, no, no, no, no, no, no, no, no, no, n...
## $ hispanic <fct> no, no, no, no, no, no, no, no, no, no, no, no, no, no, no...
## $ other    <fct> no, no, no, no, no, no, no, no, no, no, no, no, no, no, no...
## $ work     <int> 0, 30, 0, 0, 22, 40, 0, 52, 0, 0, 0, 52, 52, 52, 8, 7, 0, ...
```

(b) Wybierz wiersze od 35 do 50 i kolumny age oraz work.

```
Fertility %>%
  slice(35:50) %>%
  select(age, work)
```

```
##    age work
## 1   28   20
## 2   33   12
## 3   32    0
## 4   26   52
## 5   32   52
## 6   28    0
## 7   32   40
## 8   35    0
## 9   33    0
## 10  32   42
## 11  29    0
## 12  29   52
## 13  31    0
## 14  30   51
## 15  28    0
## 16  29    0
```

(c) Wybierz ostatni wiersz danych.

```
Fertility %>%
  tail(1)
```

```
##        morekids gender1 gender2 age afam hispanic other work
## 254654      yes  female  female  35   no       no    no    0
```

(d) Ile kobiet miało trzecie dziecko?

```
Fertility %>%
  count(morekids)
```

```
##   morekids      n
## 1       no 157742
## 2      yes  96912
```

(e) Która z kombinacji płci (4 możliwości) dla pierwszej dwójki dzieci jest najpopularniejsza?

```
Fertility %>%
  count(gender1, gender2) %>%
  arrange(n)
```

```
##   gender1 gender2     n
## 1  female  female 60946
## 2  female    male 62724
## 3    male  female 63185
## 4    male    male 67799
```

(f) Wyznacz procent kobiet pracujących 4 tygodnie lub mniej biorąc pod uwagę czynnik rasowy.

```
Fertility %>%
  group_by(afam, hispanic, other) %>%
  summarise(Perc = sum(work < 4) / n())
```

```
## `summarise()` regrouping output by 'afam', 'hispanic' (override with `.groups` argument)
```

```
## # A tibble: 6 x 4
## # Groups:   afam, hispanic [4]
##   afam  hispanic other  Perc
##   <fct> <fct>    <fct> <dbl>
## 1 no    no       no    0.500
## 2 no    no       yes   0.462
## 3 no    yes      no    0.515
## 4 no    yes      yes   0.496
## 5 yes   no       no    0.297
## 6 yes   yes      no    0.449
```

(g) Wyznacz procent kobiet w wieku 22-24 lat, których pierwszym dzieckiem był chłopiec.

```
Fertility %>%
  filter(between(age, 22, 24)) %>%
  summarise(Perc = mean(gender1 == 'male'))
```

```
##        Perc
## 1 0.5036608
```

(h) Dla jakiej rasy proporcja chłopców jako pierwsze dziecko jest najmniejsza. Ile jest takich?

```
Fertility %>%
  group_by(afam, hispanic, other) %>%
  summarise(Perc = mean(gender1 == 'male'),
            n = sum(gender1 == 'male')) %>%
  arrange(Perc)
```

```
## `summarise()` regrouping output by 'afam', 'hispanic' (override with `.groups` argument)
```

```
## # A tibble: 6 x 5
## # Groups:   afam, hispanic [4]
##   afam  hispanic other  Perc      n
##   <fct> <fct>    <fct> <dbl>  <int>
## 1 yes   no       no    0.509   6596
## 2 no    yes      no    0.512   5691
## 3 no    yes      yes   0.513   3891
## 4 no    no       no    0.515 111180
## 5 no    no       yes   0.520   3516
## 6 yes   yes      no    0.561    110
```

(i) Wyznacz procent kobiet posiadających trzecie dziecko z podziałem na płeć dwóch pierwszych dzieci.

```
Fertility %>%
  group_by(gender1, gender2) %>%
  summarise(perc = mean(morekids == 'yes')) %>%
  arrange(perc)
```

```
## `summarise()` regrouping output by 'gender1' (override with `.groups` argument)
```

```
## # A tibble: 4 x 3
## # Groups:   gender1 [2]
##   gender1 gender2  perc
##   <fct>   <fct>   <dbl>
## 1 male    female  0.346
## 2 female  male    0.347
## 3 male    male    0.404
## 4 female  female  0.425
```

4. Zbiór danych Theoph zawiera dane z eksperymentu dotyczącego farmakokinetyki teofiliny. Wykonaj poniższy fragment kodu aby uzyskać obiekt df.

```
df <- tibble::as_tibble(Theoph)
```

(a) Wybierz wszystkie kolumny pomiędzy (włącznie) Subject i Dose.

```
df %>%
  select(Subject:Dose)
```

```
## # A tibble: 132 x 3
##    Subject    Wt  Dose
##    <ord>   <dbl> <dbl>
##  1 1        79.6  4.02
##  2 1        79.6  4.02
##  3 1        79.6  4.02
##  4 1        79.6  4.02
##  5 1        79.6  4.02
##  6 1        79.6  4.02
##  7 1        79.6  4.02
##  8 1        79.6  4.02
##  9 1        79.6  4.02
## 10 1        79.6  4.02
## # ... with 122 more rows
```

(b) Posortuje dane biorąc jako pierwsze kryterium wagę (rosnąco), a jako drugie czas (malejąco).

```
df %>%
  arrange(desc(Wt), Time)
```

```
## # A tibble: 132 x 5
##    Subject    Wt  Dose  Time  conc
##    <ord>   <dbl> <dbl> <dbl> <dbl>
##  1 9        86.4   3.1  0     0
##  2 9        86.4   3.1  0.3   7.37
##  3 9        86.4   3.1  0.63  9.03
##  4 9        86.4   3.1  1.05  7.14
##  5 9        86.4   3.1  2.02  6.33
##  6 9        86.4   3.1  3.53  5.66
##  7 9        86.4   3.1  5.02  5.67
##  8 9        86.4   3.1  7.17  4.24
##  9 9        86.4   3.1  8.8   4.11
## 10 9        86.4   3.1 11.6   3.16
## # ... with 122 more rows
```

(c) Dodaj dodatkową zmienną weight.cat, która opsiuje klasyfikację osób według poniższego schematu:

- **Poniżej 66,8 kg – Welterweight,**
- **66,8 – 72,57 – Light-Middleweight,**
- **72,57 – 76,2 – Middleweight,**
- **Powyżej 76,2 kg – Super-Middleweight.**

```
df %>%
  mutate(weight.cat = ifelse(Wt < 66.8, 'Welterweight',
                             ifelse(Wt <= 72.57, 'Light-Middleweight',
                                    ifelse(Wt <= 76.62, 'Middleweight',
                                           'Super-Middleweight')))) -> df1
df1
```

```
## # A tibble: 132 x 6
##    Subject    Wt  Dose   Time  conc weight.cat
##    <ord>   <dbl> <dbl>  <dbl> <dbl> <chr>
##  1 1        79.6  4.02  0      0.74 Super-Middleweight
##  2 1        79.6  4.02  0.25   2.84 Super-Middleweight
##  3 1        79.6  4.02  0.570  6.57 Super-Middleweight
##  4 1        79.6  4.02  1.12  10.5  Super-Middleweight
##  5 1        79.6  4.02  2.02   9.66 Super-Middleweight
##  6 1        79.6  4.02  3.82   8.58 Super-Middleweight
##  7 1        79.6  4.02  5.1    8.36 Super-Middleweight
##  8 1        79.6  4.02  7.03   7.47 Super-Middleweight
##  9 1        79.6  4.02  9.05   6.89 Super-Middleweight
## 10 1        79.6  4.02 12.1    5.94 Super-Middleweight
## # ... with 122 more rows
```

(d) Pogrupuj dane ze względu na zmienną weight.cat i znajdź średni czas i sumę dawek dla każdej kategorii wagowej.

```
df1 %>%
  mutate(weight.cat = factor(weight.cat,
                             levels =  c('Welterweight',
                                         'Light-Middleweight',
                                         'Middleweight',
                                         'Super-Middleweight'),
                             ordered = TRUE)) %>%
  group_by(weight.cat) %>%
  summarise(mean.time = mean(Time), sum.dose = sum(Dose))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 3
##   weight.cat         mean.time sum.dose
##   <ord>                  <dbl>    <dbl>
## 1 Welterweight            5.88    292.
## 2 Light-Middleweight      5.89    148.
## 3 Middleweight            5.94     48.4
## 4 Super-Middleweight      5.90    122.
```