

Statystyki bronchoskopii w ramach NFZ - analiza z wykorzystaniem package `nfzapis1`

Mikołaj Kamiński

2 czerwca 2022

1. Wstęp

`nfzapis1` to package R przygotowany przez Centrum e-Zdrowia. Package umożliwia błyskawiczne przetwarzanie danych udostępnionych przez NFZ w ramach API dla statystyk świadczeń szpitalnych.

W tym krótkim materiale przedstawimy Państwu przykładowe wykorzystanie package analizując dane dotyczące bronchoskopii.

2. Wprowadzenie do wykorzystania package

Oryginalna instrukcja API znajduje się pod linkiem: <https://api.nfz.gov.pl/app-stat-api-jgp/> Natomiast package `nfzapis1` <https://github.com/PolishNHF/nfzapis1>

```
library(nfzapis1)
```

Na początek zidentyfikujmy Jednorodną Grupę Pacjentów (JGP) dotyczącą hospitalizacji rozliczanych jako bronchoskopia.

Możemy zastosować bazę danych wgranej do package'u:

```
jgp[[2]] %>%
  filter(str_detect(Nazwa, "Bronchos"))
## # A tibble: 2 x 11
##   Lp. 'Kod grupy' Kod      Nazwa   'Taryfa\nustalona~ 'Wartość punktow~
##   <dbl> <chr>      <chr>      <chr>   <chr>          <chr>
## 1   109 D05      5.51.01.0004005 Bronch~ <NA>          826
## 2   551 PZD03    5.51.01.0018029 Bronch~ <NA>          935
## # ... with 5 more variables: Wartość punktowa - hospitalizacja planowa <dbl>,
## #   Wartość punktowa - "leczenie jednego dnia" <dbl>,
## #   Liczba dni pobytu finansowana grupą - typ umowy hospitalizacja <dbl>,
## #   Wartość punktowa hospitalizacji < 3 dni - typ umowy hospitalizacja/ hospitalizacja planowa <dbl>
## #   Wartość punktowa osobodnia ponad ryczałt finansowany grupą - typ umowy hospitalizacja <dbl>
```

Uzyskujemy dwie JGP: 1) 5.51.01.0004005 "Bronchoskopia *" 2) 5.51.01.0018029 "Bronchoskopia < 18 r.ż.

W tym przykładzie będziemy analizować pierwszą grupę, dotyczącą bronchoskopii u osób dorosłych.

Funkcja `index_tables` pozwala zidentyfikować wszystkie dane dostępne dla danej JGP:

```
index_tables(name='5.51.01.0004005')
## # A tibble: 129 x 4
##   year name full_name link
##   <chr> <chr> <chr> <chr>
## 1 2009 33323130-3636-5254-4450-503030303030 hospitalizati~ https://api.nfz.go~
## 2 2009 33323130-3636-5254-4750-503030303030 hospitalizati~ https://api.nfz.go~
## 3 2009 33323130-3636-3179-4430-353030303030 icd-9-procedu~ https://api.nfz.go~
## 4 2009 33323130-3636-5254-4250-503030303030 hospitalizati~ https://api.nfz.go~
## 5 2009 33323130-3636-5254-4150-503030303030 general-data https://api.nfz.go~
## 6 2009 33323130-3636-5254-4350-503030303030 hospitalizati~ https://api.nfz.go~
## 7 2009 33323130-3636-3330-3030-303030303030 icd-9-procedu~ https://api.nfz.go~
## 8 2009 33323130-3636-5254-4b50-503030303030 hospitalizati~ https://api.nfz.go~
## 9 2009 33323130-3636-5254-4950-503030303030 product-categ~ https://api.nfz.go~
## 10 2009 33323130-3636-5254-4a50-503030303030 histograms https://api.nfz.go~
## # ... with 119 more rows
```

Dane są dostępne dla lat: 2009-2020:

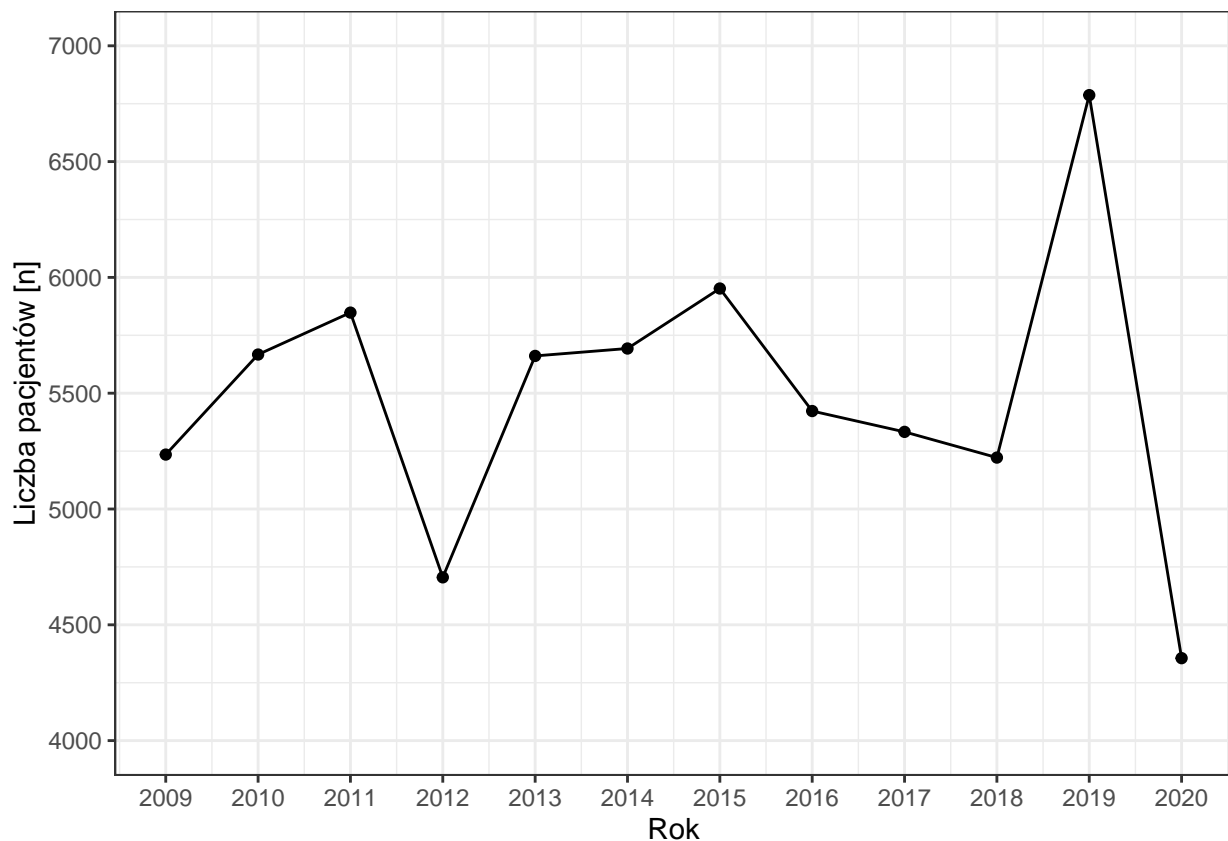
```
index_tables(name='5.51.01.0004005') %>%
  select(year) %>%
  unique()
## New names:
## * ' ' -> ...1
## * ' ' -> ...2
## * ' ' -> ...3
## * ' ' -> ...4
## # A tibble: 12 x 1
##   year
##   <chr>
## 1 2009
## 2 2010
## 3 2011
## 4 2012
## 5 2013
## 6 2014
## 7 2015
## 8 2016
## 9 2017
## 10 2018
## 11 2019
## 12 2020
```

Wykorzystajmy funkcję `basic_data()` by uzyskać liczbę hospitalizacji.

```
b <- basic_data(name = '5.51.01.0004005')
b
## # A tibble: 12 x 5
##   branch 'hospital-types' 'number-of-patie~ 'number-of-hospi~ 'ratio-of-rehosp~
##   <lgl> <lgl> <int> <int> <dbl>
## 1 NA NA 4999 5235 1.05
## 2 NA NA 5416 5667 1.05
## 3 NA NA 5615 5848 1.04
## 4 NA NA 4546 4705 1.04
```

```
## 5 NA      NA      5470      5661      1.03
## 6 NA      NA      5481      5693      1.04
## 7 NA      NA      5689      5952      1.05
## 8 NA      NA      5176      5423      1.05
## 9 NA      NA      4982      5333      1.07
## 10 NA     NA      5010      5222      1.04
## 11 NA     NA      6512      6787      1.04
## 12 NA     NA      4188      4356      1.04
## # ... with 10 more variables: percentage <dbl>, percentage-of-sections <dbl>,
## #   duration-of-hospitalization-mediana <int>,
## #   duration-of-hospitalization-mode <int>,
## #   average-value-of-hospitalization <dbl>,
## #   average-value-of-hospitalization-points <dbl>, average-value-of-drg <dbl>,
## #   average-value-of-drg-points <dbl>, year <int>, jgp <chr>
```

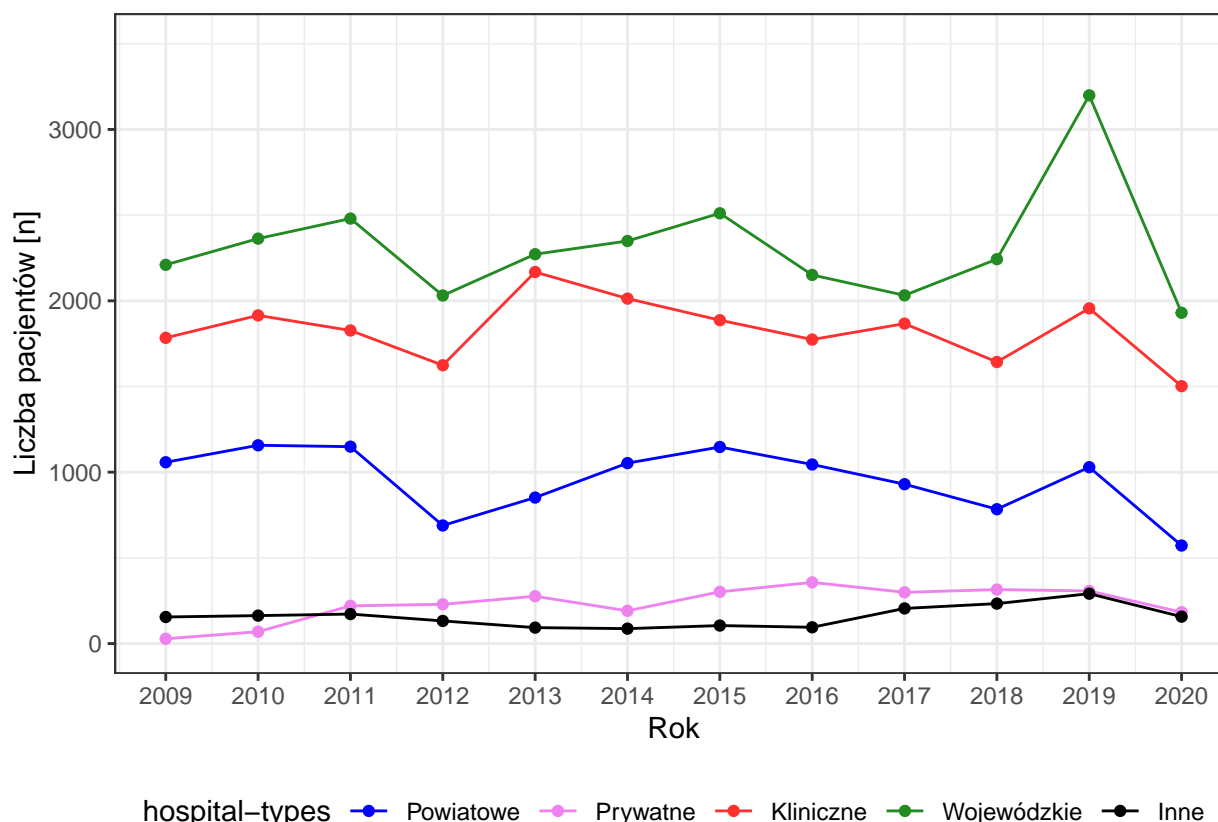
```
b %>% select('number-of-hospitalizations', year) %>%
  ggplot() + geom_line(aes(x=year, y='number-of-hospitalizations')) + geom_point(aes(x=year, y='number-of-hospitalizations')) +
  theme_bw() + xlab('Rok') +
  ylab('Liczba pacjentów [n]') + scale_y_continuous(limits = c(4000, 7000), breaks = seq(4000, 7000, by=500)) +
  scale_x_continuous(limits = c(2009, 2020), breaks = seq(2009, 2020, by=1))
```



Możemy również powtórzyć analizę z podziałem na rodzaje szpitali:

```
b1 <- basic_data(name = '5.51.01.0004005', division = 'hospitalType')
b1 %>% select('number-of-hospitalizations', year, 'hospital-types') %>%
```

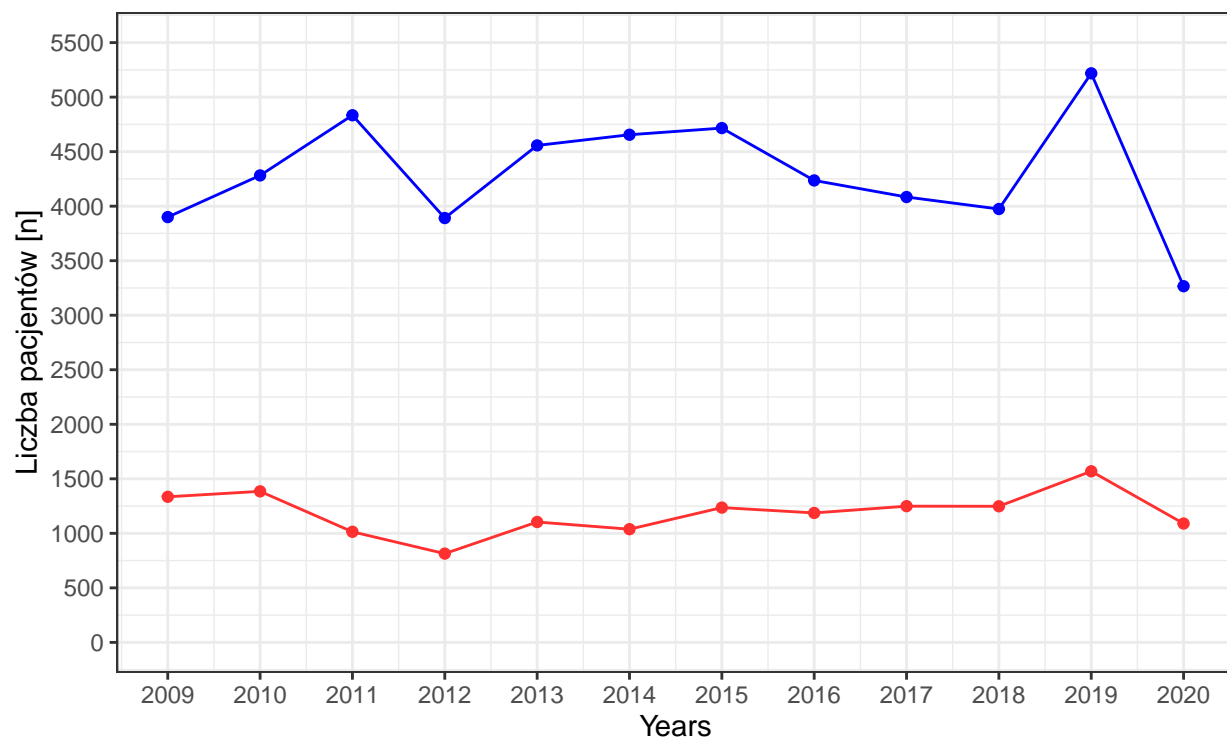
```
ggplot() + geom_line(aes(x=year, y='number-of-hospitalizations', col='hospital-types')) +
geom_point(aes(x=year, y='number-of-hospitalizations', col='hospital-types'))+ theme_bw() +
xlab('Rok') + ylab('Liczba pacjentów [n]') + ylim(0,3500) +
scale_x_continuous(limits = c(2009, 2020), breaks = seq(2009, 2020, by=1))+
scale_color_manual("hospital-types",labels = c('Powiatowe','Prywatne','Kliniczne','Wojewódzkie',"Inne"),
theme(legend.position = "bottom")
```



Funkcja `hospitalization_admission()` pozwala na uzyskanie informacji nt. charakteru przyjęcia pacjenta do szpitala. Poniżej uprościliśmy podział na przyjęcia planowe oraz pilne.

```
adm <- hospitalization_admission(name = '5.51.01.0004005')

adm %>%
  mutate(planned_admission = if_else(grepl("plan", 'type-of-admission-name'), 1, 0)) %>%
  group_by(year, planned_admission) %>% summarize(total = sum(as.numeric('number-of-hospitalizations')))
ggplot() + geom_line(aes(x=year, y = total, col=as.factor(planned_admission))) +
geom_point(aes(x=year, y = total, col=as.factor(planned_admission)))+ theme_bw() + xlab('Years')+
ylab('Liczba pacjentów [n]') + theme(legend.position = "bottom") +
scale_y_continuous(limits = c(0, 5500), breaks = seq(0, 5500, by=500)) +
scale_x_continuous(limits = c(2009, 2020), breaks = seq(2009, 2020, by=1))+
scale_color_manual("planned_admission",labels = c('Pilne','Planowe'), values = c("firebrick1", "blue"),
theme(legend.position = "bottom")
## 'summarise()' has grouped output by 'year'. You can override using the '.groups' argument.
```



Rodzaj przyjęcia szpitala — Pilne — Planowe