

# Wpływ sytuacji emidemiologicznej na środowisko naturalne

Jakub Koziel, Konstanty Kraszewski, Bartosz Sawicki

11/30/2020

## Wprowadzenie

Wraz z zespołem postanowiliśmy poddać analizie dane dotyczące jakości powietrza i sprawdzić w jaki sposób rozwijająca się epidemia oddziaływała na jego stan. Dokonywaliśmy analizy zbiorów zaproponowanych dotyczących rozwoju COVID-19, a także danych na temat powietrza ze strony <https://aqicn.org/data-platform/covid19>.

Poniżej prezentujemy kilka z najciekawszych wyników naszej eksploracji

## Pobieranie danych

```
curl --compressed -o waqi-covid-2020.csv https://aqicn.org/data-platform/covid19/report/19780-07745f9
curl --compressed -o waqi-covid-2019Q1.csv https://aqicn.org/data-platform/covid19/report/19780-07745f9
curl --compressed -o waqi-covid-2019Q2.csv https://aqicn.org/data-platform/covid19/report/19780-07745f9
curl --compressed -o waqi-covid-2019Q3.csv https://aqicn.org/data-platform/covid19/report/19780-07745f9
curl --compressed -o waqi-covid-2019Q4.csv https://aqicn.org/data-platform/covid19/report/19780-07745f9
```

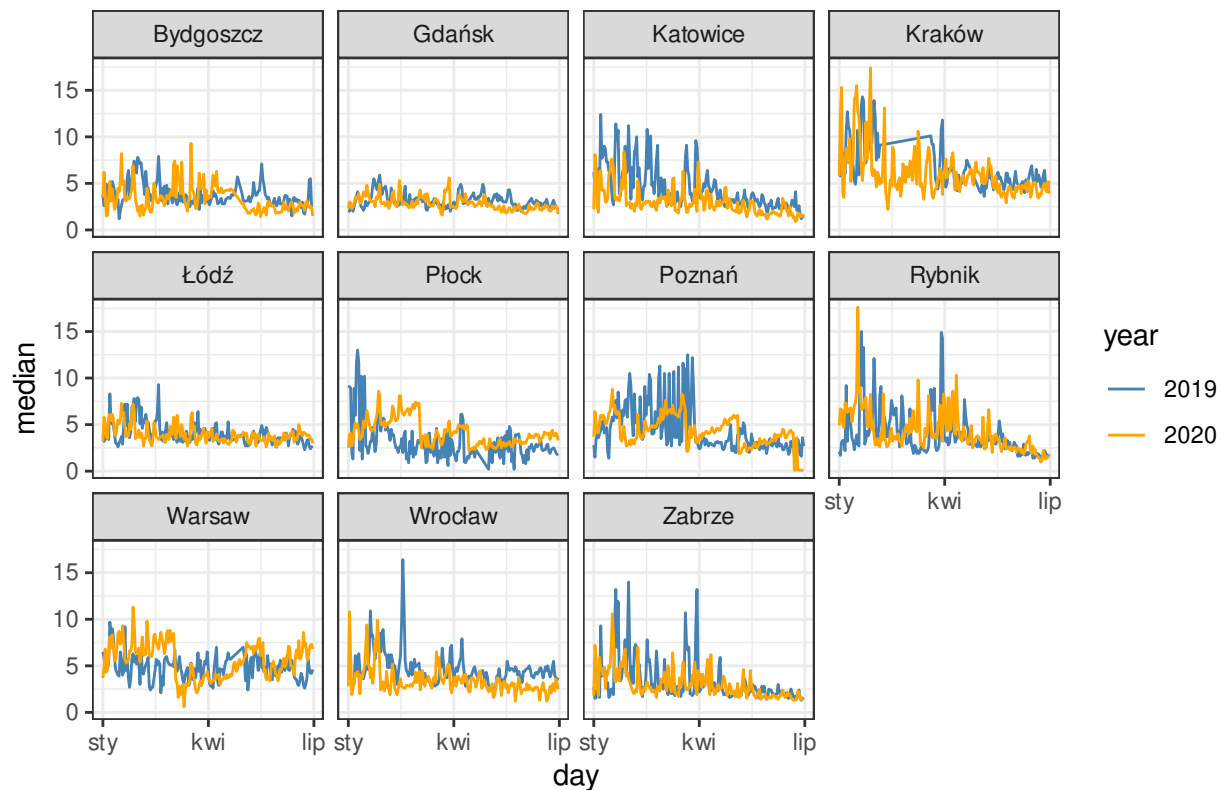
## Wczytywanie danych

```
source("data_upload.R")
```

## Sytuacja w pierwszym półroczu 2020 w Polsce

```
raw_data %>%
  filter(Country == "PL") %>%
  filter(Specie == "co") %>%
  filter(City != "Tarnów", City != "Szczecin", City != "Kielce") %>% #nie mają wszystkich lat
  mutate(year = as.factor(year(Date))) %>%
  mutate(day = as.Date(format(Date, format = "%d-%m"), format = "%d-%m")) %>%
  filter(day < as.Date("01.07.2020", format = "%d.%m.%Y")) %>%
  ggplot(aes(x = day)) +
  geom_line(aes(y = median, group = year, color = year)) +
  facet_wrap(~City) +
  ggtitle("CO concentration in polish cities") +
  scale_color_manual(values = c("steelblue", "orange")) +
  theme_bw()
```

## CO concentration in polish cities



### Wnioski

- w Warszawie i Płocku widać spadek emisji CO.
- w innych miastach nie widać spadku (ogrzewanie domów?).

## Przedstawienie danych na mapie

```
#wspolrzedne dla polskich miast
miasta <- raw_data_2020 %>% filter(Country == c("PL")) %>% distinct(City) %>% select(City)

nrow <- nrow(miasta)
counter <- 1
miasta$lon[counter] <- 0

## Warning: Unknown or uninitialised column: `lon`.
miasta$lat[counter] <- 0

## Warning: Unknown or uninitialised column: `lat`.

while (counter <= nrow){
  CityName <- gsub(' ', '%20', miasta$City[counter]) #remove space for URLs
  CountryCode <- "PL"
  url <- paste(
    "http://nominatim.openstreetmap.org/search?city=",
    CityName
    , "&countrycodes="
    , CountryCode
  )
}
```

```

    , "&limit=9&format=json"
    , sep="")
x <- fromJSON(url)
if(is.vector(x)){
  miasta$lon[counter] <- x[[1]]$lon
  miasta$lat[counter] <- x[[1]]$lat
}
counter <- counter + 1
}

##przygotowanie danych

no2_2019 <- raw_data_2019 %>% filter(Country == c("PL")) %>% filter(Specie == "no2") %>% group_by(City)
no2_2020 <- raw_data_2020 %>% filter(Country == c("PL")) %>% filter(Specie == "no2") %>% group_by(City)

df <- merge(miasta, no2_2019)
df <- merge(df, no2_2020)

df[2,3] <- 54.2143405

## rysowanie na mapie

# funkcja do rysowania barplotow
mapbars <- function(x, xllc = 0, yllc = 0, barwidth=1, maxheight=10){
  # calculate how long each bar needs to be
  bars <- (x/max(x)) * maxheight
  # get some quick colors
  col <- c("steelblue","orange")

  for(i in 1:length(x)){
    # figure out x- and y coordinates for the corners
    leftx <- xllc + ((i-1) * barwidth)
    rightx <- leftx + barwidth
    bottomy <- yllc
    topy <- yllc + bars[i]
    # draw the bar
    polygon(x=c(leftx, rightx, rightx, leftx, leftx),
            y=c(bottomy, bottomy, topy, topy, bottomy),
            col=col[i])
  }
}

data("wrld_simpl")
Poland <- subset(wrld_simpl, NAME=="Poland")

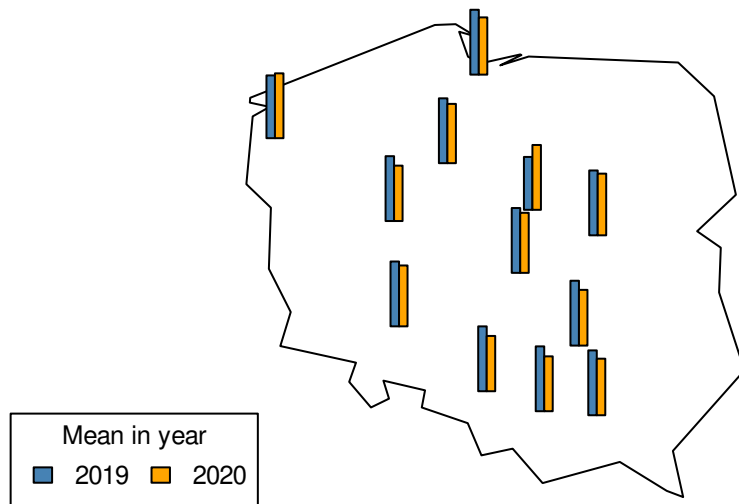
plot(Poland, axes=FALSE)

for(i in 1:nrow(df)){
  if(df[i, 1]=="Katowice") next
  if(df[i, 1]=="Rybnik") next

  mapbars(x = c(df[i, 4], df[i, 5]), xllc=as.numeric(df[i, 2]), yllc=as.numeric(df[i, 3]), barwidth=.1)
}

```

```
legend("bottomleft", inset=.02, title="Mean in year",
      c("2019", "2020"), fill=c("steelblue", "orange"), horiz=TRUE, cex=0.8)
```

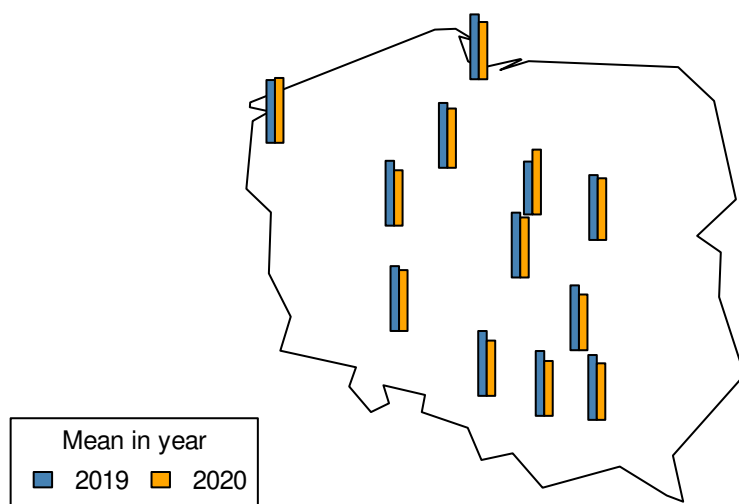


## Wnioski

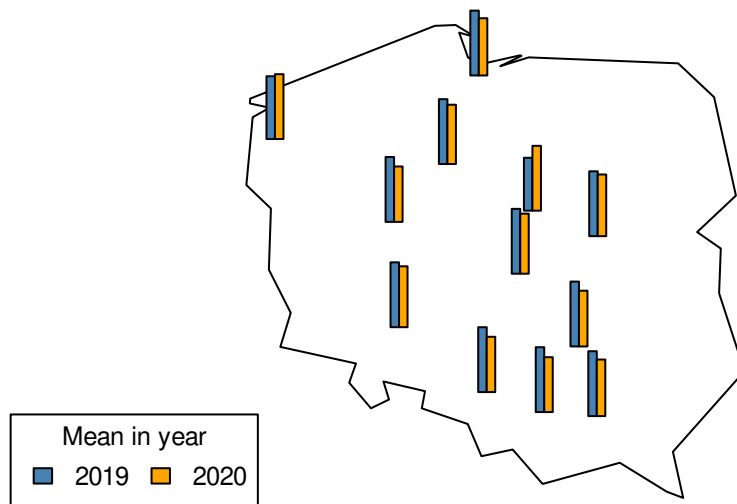
Na mapie w poszczególnych miastach zostały osadzone barploty, w których kolorem zgodnie z legendą oznaczone zostało średnie zanieczyszczenie NO<sub>2</sub> w danym roku w pierwszym jego kwartale (informacja ta pojawi się na finalnej mapie). Z tej mapy wynika, że średni poziom tego zanieczyszczenia dla pierwszego kwartału roku 2020 jest niższy niż w roku ubiegłym. Możemy zatem wywnioskować, że jakość powietrza pod względem zanieczyszczenia NO<sub>2</sub> wzrosła, a sytuacja epidemiologiczna miała na to pozytywny wpływ. Przed pojawieniem się na plakacie, mapa prawdopodobnie zostanie poddana obróbce graficznej, aby pokazać jaką wielkość reprezentuje dany słupek. Analizie w analogiczny sposób mogą też zostać poddane inne zanieczyszczenia.

Przykłady map wygenerowane dla innych zanieczyszczeń.

## PM10 pollution in Poland



## CO pollution in Poland



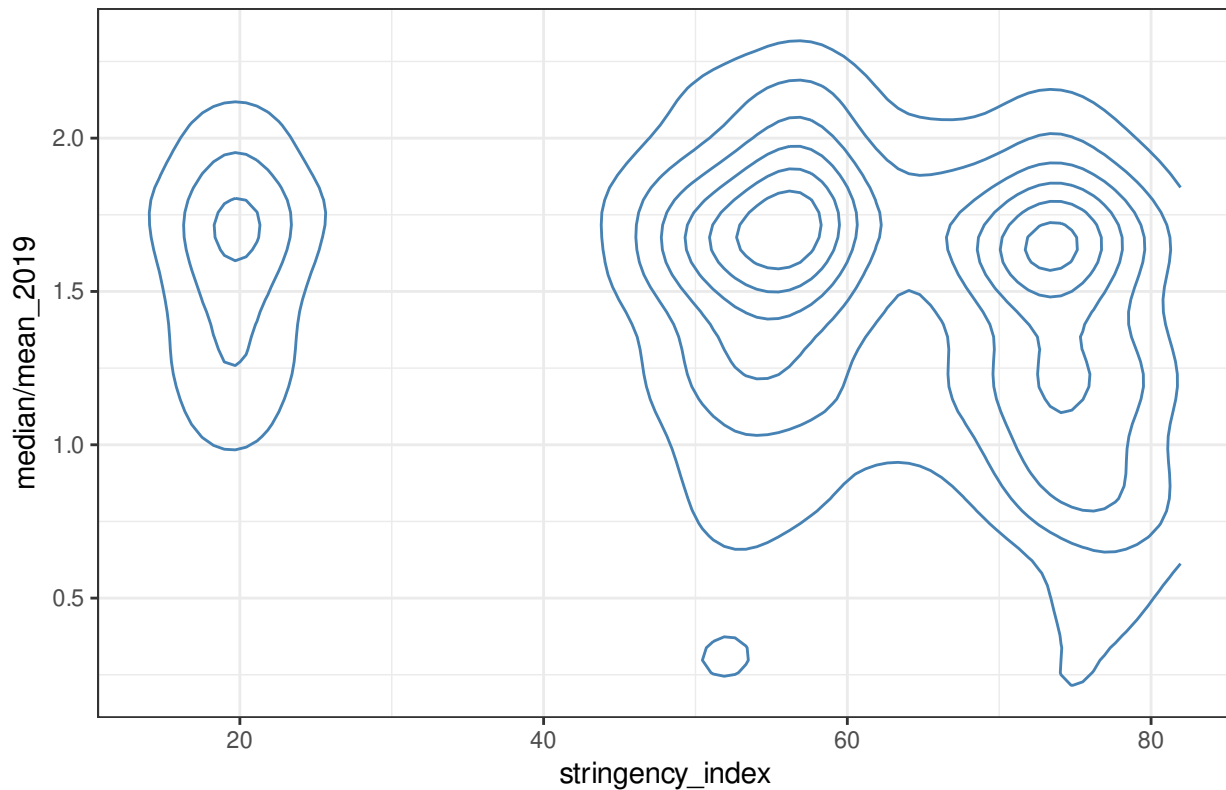
### Wnioski

Przyglądając się teraz mapom wygenerowanym dla innych zanieczyszczeń, możemy uzyskać pełniejszy obraz sytuacji, a wysnuwane wnioski będą podobne jak dla omówionego wyżej NO<sub>2</sub>.

## Jakość powietrza w odniesieniu do średniej w miastach w 2019 roku

```
raw_data_2020 %>%
  filter(Specie == "aqi") %>%
  inner_join(means_2019, by = "City") %>%
  inner_join(iso_codes, by = c("Country" = "Alpha-2 code")) %>%
  inner_join(covid_df, by = c("Date" = "Date", "Alpha-3 code" = "iso_code")) %>%
  ggplot(aes(y = median/mean_2019, x = stringency_index)) +
  geom_density2d(color = "steelblue") +
  ggtitle("Air Quality Index in comparison to 2019 mean by stringency index") +
  theme_bw()
```

## Air Quality Index in comparison to 2019 mean by stringency index



### Wnioski

- większość miast odnotowała poprawę jakości powietrza.
- nie widać dużego związku jakości powietrza z wprowadzanymi ograniczeniami

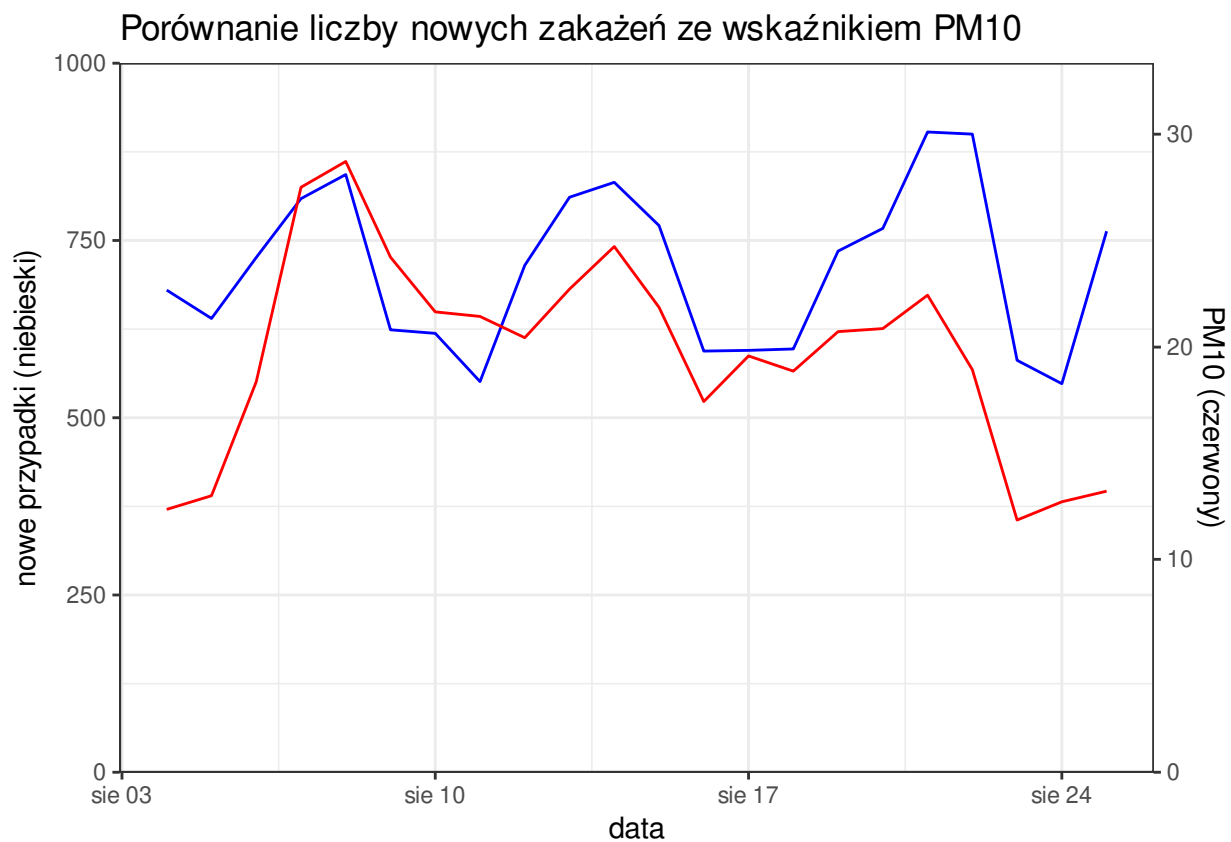
## Porównanie liczby nowych zakażeń ze wskaźnikiem PM10

```
aqcin <- read_csv("waqi-covid-2020.csv", skip = 4)
```

```
## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   Country = col_character(),
##   City = col_character(),
##   Specie = col_character(),
##   count = col_double(),
##   min = col_double(),
##   max = col_double(),
##   median = col_double(),
##   variance = col_double()
## )
```

```
aqcin %>%
  filter(Country=="PL", Specie=="pm10") %>%
  group_by(Date) %>%
  summarise(pm10 = mean(median)) -> data
data$Date = as.Date(data$Date)
```

```
data %>%
  inner_join(filter(covid_df, location=="Poland"),
             by = c("Date" = "Date")) %>%
  select(Date, pm10, new_cases) -> data
ggplot(data, aes(x = Date)) +
  geom_line(aes(y = new_cases), color = "blue") +
  geom_line(aes(y = pm10 * 30), color = "red") +
  scale_y_continuous(name = "nowe przypadki (niebieski)",
                     sec.axis = sec_axis( trans=~./30,
                                           name="PM10 (czerwony)"),
                     limits = c(0, 1000),
                     expand = expansion(mult = c(0, 0))
  ) +
  xlim(as.Date(c("2020-08-04", "2020-08-25"))) +
  labs(title = "Porównanie liczby nowych zakażeń ze wskaźnikiem PM10",
       x = "data") +
  theme_bw()
```



## Wnioski

- dwie łamane przyjmują bardzo podobne kształty
- wskaźnik PM10 wydaje się “wyprzedzać” liczbę nowych zakażeń mniej więcej o jeden dzień
- być może wartości te są ze sobą powiązane