



Serbia  
Digital  
Week



eUprava

KANCELARIJA  
ZA IT I eUPRAVU



WORLD BANK GROUP



British Embassy  
Belgrade



UKaid

from the British people



UN  
DP

Empowered lives.  
Resilient nations.

# VIZUELIZACIJA PODATAKA

Dejan Vujičić

Fakultet tehničkih nauka u Čačku,

6. april 2019. godine

# UVOZ PODATAKA

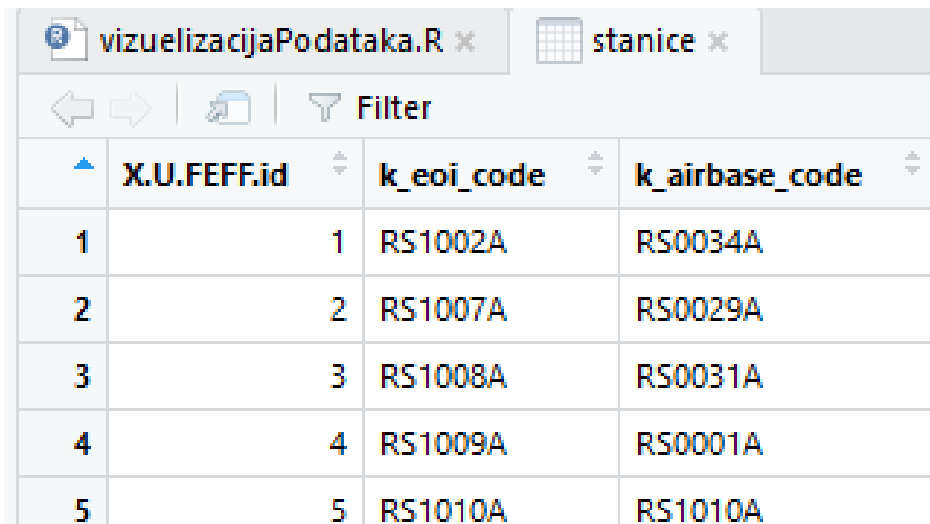
- Skup podataka o zagađenosti vazduha je isti kao i u prethodnoj oblasti, obrada podataka
- Dodatno, koristićemo drugi skup podataka o stanicama
- Podrazumeva se da su u radnom okruženju ostali objekti *'podaci'* i *'cacak'*
- Uvoz druge datoteke:

```
stanice = read.csv("station.csv", header =  
TRUE, encoding="UTF-8")
```



# PREIMENOVANJE KOLONE

- Pošto kolona *'id'* sadrži dodatne karaktere na početku, usled UTF-8 učitavanja, treba preimenovati tu kolonu:



The screenshot shows a data visualization interface with a table. The table has four columns: an index column, 'X.U.FEFFF.id', 'k\_eoi\_code', and 'k\_airbase\_code'. The data rows are as follows:

	X.U.FEFFF.id	k_eoi_code	k_airbase_code
1	1	RS1002A	RS0034A
2	2	RS1007A	RS0029A
3	3	RS1008A	RS0031A
4	4	RS1009A	RS0001A
5	5	RS1010A	RS1010A

- `names(stanice)[1] = "station_id"`



# SPAJANJE TABELA

- Pošto želimo da prikazemo merna mesta na mapi, potrebno je spojiti dve tabele, po nazivu kolone kao jedinstvenom identifikatoru:
- `podaci = merge(podaci, stanice, by="station_id")`



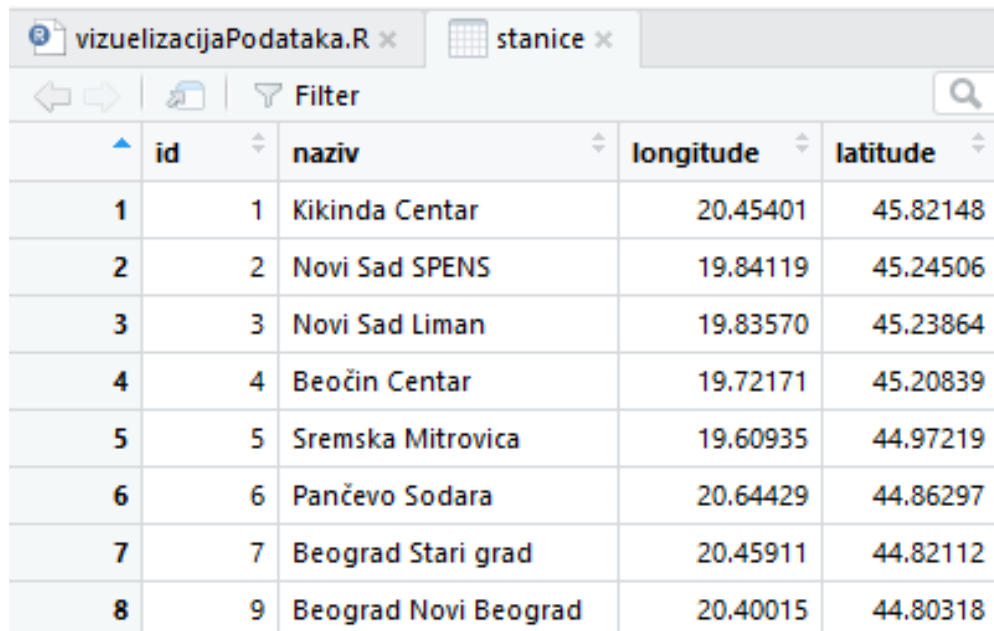
# ODREĐIVANJE JEDINSTVENIH VREDNOSTI

- Za dalju analizu potrebne su nam jedinstvene vrednosti za ID stanice, geografsku širinu i dužinu i naziv stanice
- `id = unique(podaci$station_id)`
- `latitude = unique(podaci$latitude)`
- `longitude = unique(podaci$longitude)`
- `naziv = unique(podaci$k_name)`



# KREIRANJE TABELE

- Sada možemo kreirati tabelu od ovih jedinstvenih parametara
- `stanice = data.frame(id, naziv, longitude, latitude)`



	id	naziv	longitude	latitude
1	1	Kikinda Centar	20.45401	45.82148
2	2	Novi Sad SPENS	19.84119	45.24506
3	3	Novi Sad Liman	19.83570	45.23864
4	4	Beočin Centar	19.72171	45.20839
5	5	Sremska Mitrovica	19.60935	44.97219
6	6	Pančevo Sodara	20.64429	44.86297
7	7	Beograd Stari grad	20.45911	44.82112
8	9	Beograd Novi Beograd	20.40015	44.80318



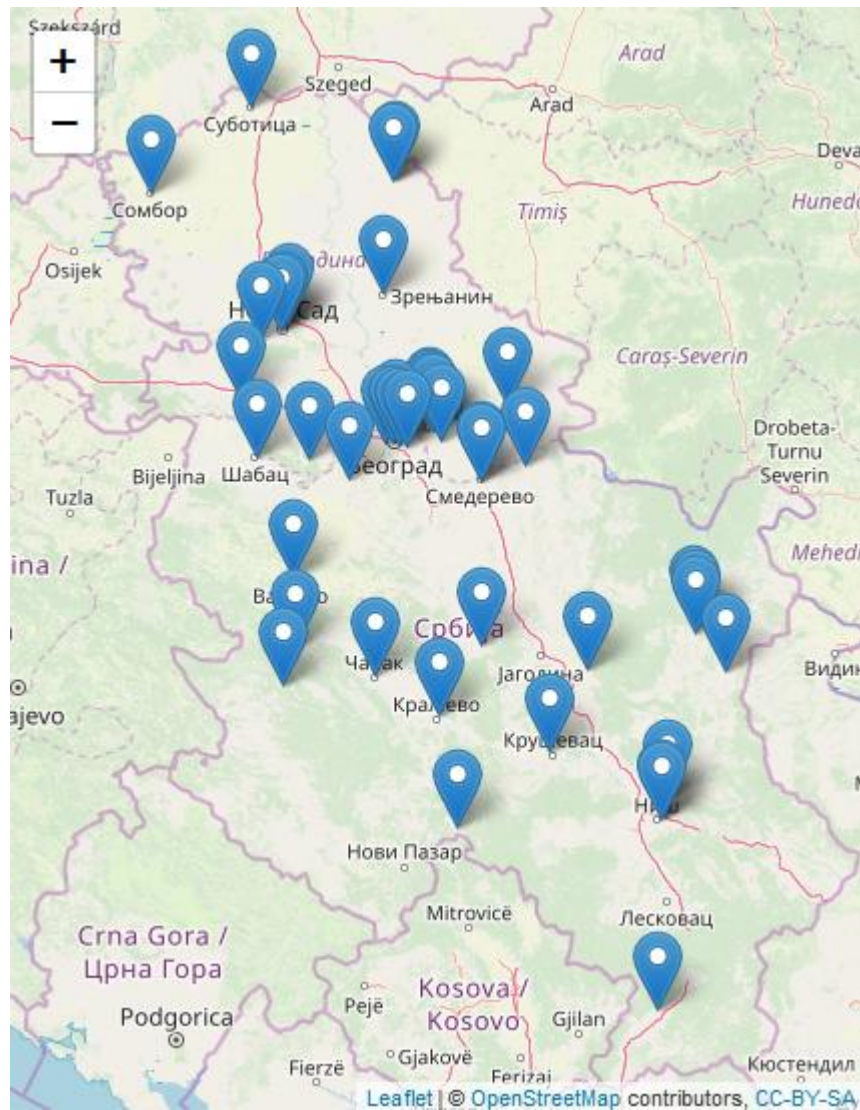
# PRIKAZ NA MAPI

- Za prikaz stanica na mapi korišćićemo biblioteku *'leaflet'*
- Mapa se kreira pozivom funkcije *addTiles()* uz poziv *addMarkers()*

```
library(leaflet)
m = leaflet() %>%
  addTiles() %>% # dodavanje pozadine
  addMarkers(lng=longitude,
             lat=latitude,
             popup=as.character(paste(id,naziv)))
m # prikaz mape
```



# PRIKAZ NA MAPI





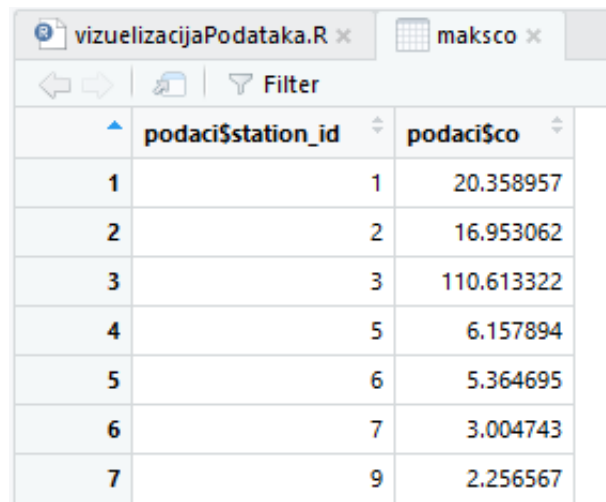
# PRIKAZ NA MAPI

- Klikom na marker prikazuju se informacije o ID stanice i nazivu mernog mesta



# PRIKAZ PODATAKA OD INTERESA

- Pored ovoga, možemo prikazati neke korisne podatke na markerima, npr. najveću vrednost parametra CO po stanicama
- Potrebno je kreirati tabelu koja uzima ove maksimalne vrednosti i povezuje ih sa ID stanica
- `maksco = aggregate(podaci$co ~ podaci$station_id, podaci, max)`



The screenshot shows a R Shiny application window with two tabs: 'vizuelizacijaPodataka.R' and 'maksco'. The 'maksco' tab is active, displaying a data table with two columns: 'podaci\$station\_id' and 'podaci\$co'. The table contains 7 rows of data, showing the maximum CO value for each station ID.

	podaci\$station_id	podaci\$co
1	1	20.358957
2	2	16.953062
3	3	110.613322
4	5	6.157894
5	6	5.364695
6	7	3.004743
7	9	2.256567



## PRIKAZ PODATAKA OD INTERESA

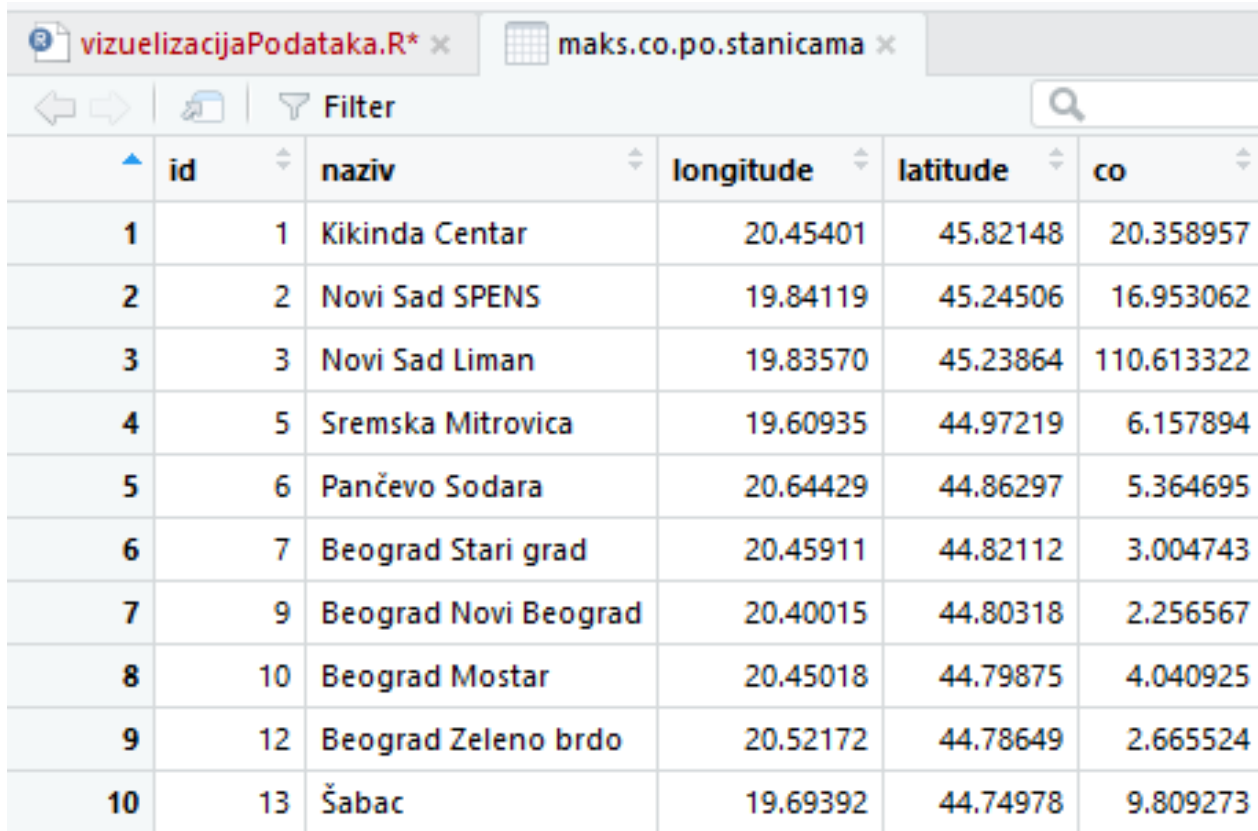
- Zatim treba kreirati novu tabelu koja će imati ID stanice, naziv, geografske koordinate i maksimalnu vrednost CO
- Ovo se može uraditi korišćenjem SQL upita i biblioteke *'sqldf'*

```
library(sqldf)
```

```
maks.co.po.stanicama = sqldf(  
  'SELECT id, naziv, longitude,  
latitude, podaci$co AS co  
  FROM stanice JOIN maksco  
  ON id=podaci$station_id')
```



# PRIKAZ PODATAKA OD INTERESA



The image shows a screenshot of a web application interface for data visualization. The browser tabs at the top are 'vizuelizacijaPodataka.R\*' and 'maks.co.po.stanicama'. The interface includes navigation arrows, a filter icon, and a search box. Below the navigation is a table with 7 columns: an index column, 'id', 'naziv', 'longituda', 'latituda', and 'co'. The table contains 10 rows of data representing different stations.

	id	naziv	longituda	latituda	co
1	1	Kikinda Centar	20.45401	45.82148	20.358957
2	2	Novi Sad SPENS	19.84119	45.24506	16.953062
3	3	Novi Sad Liman	19.83570	45.23864	110.613322
4	5	Sremska Mitrovica	19.60935	44.97219	6.157894
5	6	Pančevo Sodara	20.64429	44.86297	5.364695
6	7	Beograd Stari grad	20.45911	44.82112	3.004743
7	9	Beograd Novi Beograd	20.40015	44.80318	2.256567
8	10	Beograd Mostar	20.45018	44.79875	4.040925
9	12	Beograd Zeleno brdo	20.52172	44.78649	2.665524
10	13	Šabac	19.69392	44.74978	9.809273



## PRIKAZ PODATAKA OD INTERESA

```
m = leaflet() %>%
```

```
  addTiles() %>%
```

```
  addMarkers(lng=maks.co.po.stanicama$longitu  
de,
```

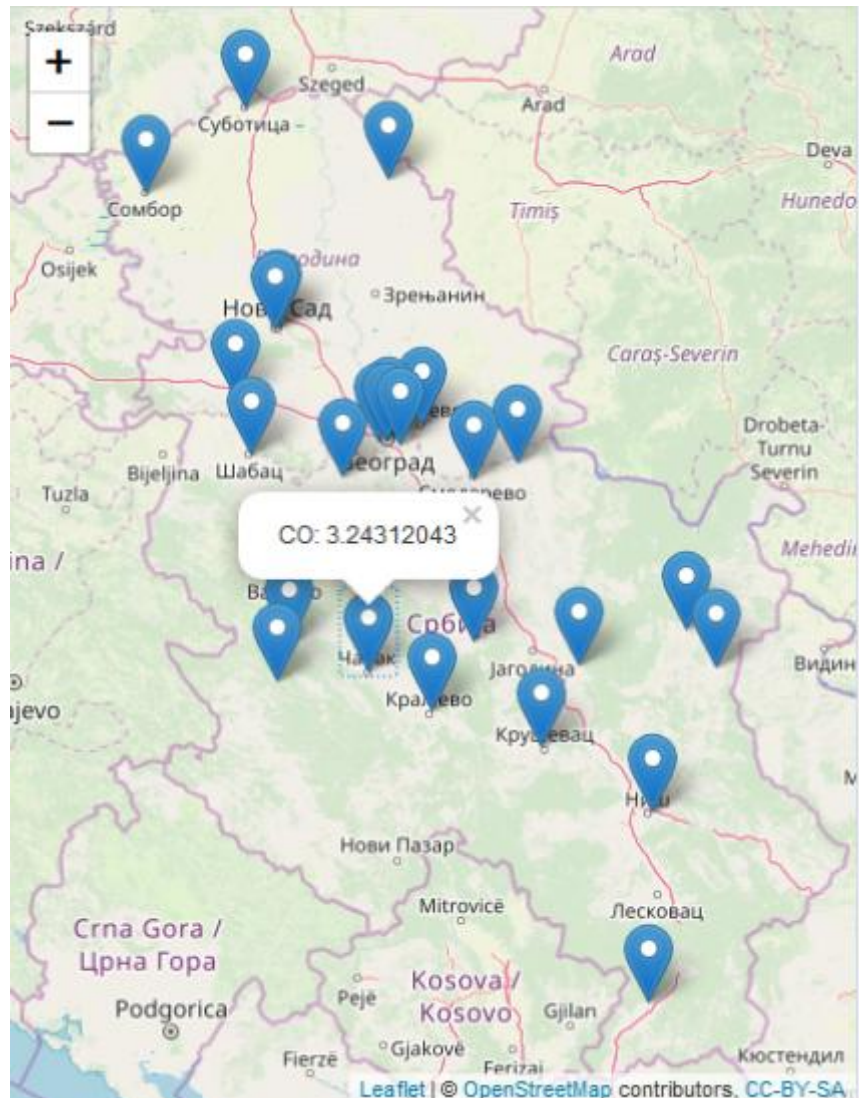
```
lat=maks.co.po.stanicama$latitude,
```

```
popup=as.character(paste("CO:",maks.co.po.s  
tanicama$co)))
```

```
m # prikaz mape
```



# PRIKAZ PODATAKA OD INTERESA



# PRIKAZ PODATAKA OD INTERESA

- Sada bi valjalo i vizuelno predstaviti markere prema bojama, u zavisnosti od graničnih vrednosti parametra CO
- Kreiramo pomoćnu funkciju za postavljanje boje:

```
boja <- function(tabela) {  
  sapply(tabela$co, function(co) {  
    if(co <= 5) {  
      "green"  
    } else if(co <= 10) {  
      "orange"  
    } else {  
      "red"  
    } })  
}
```



# PRIKAZ PODATAKA OD INTERESA

- Sada koristimo *leaflet* API za postavljanje drugačijih markera:

```
icons <- awesomeIcons(  
  icon = 'ios-close',  
  iconColor = 'black',  
  library = 'ion',  
  markerColor = boja(maks.co.po.stanicama)  
)
```

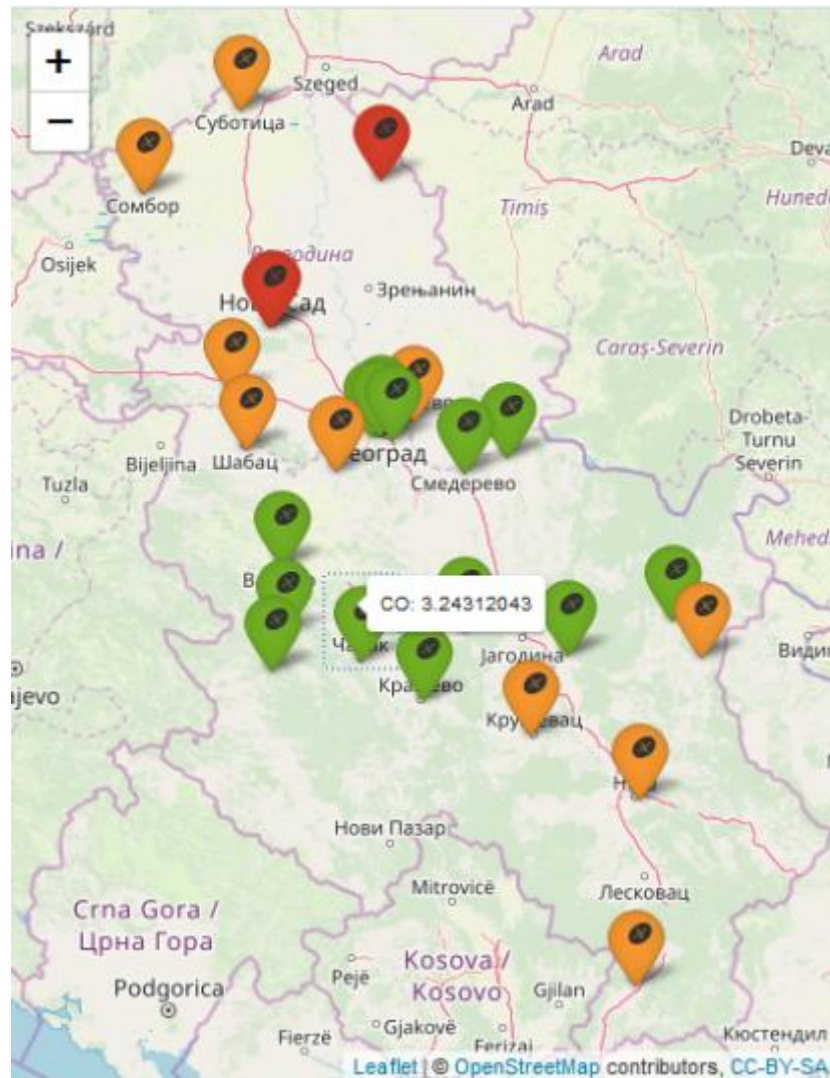
- I kreiramo novu mapu:

```
leaflet(maks.co.po.stanicama) %>% addTiles() %>%  
  addAwesomeMarkers(~longitude, ~latitude,  
  icon=icons, label=~as.character(paste("CO:",co)))
```





# PRIKAZ PODATAKA OD INTERESA



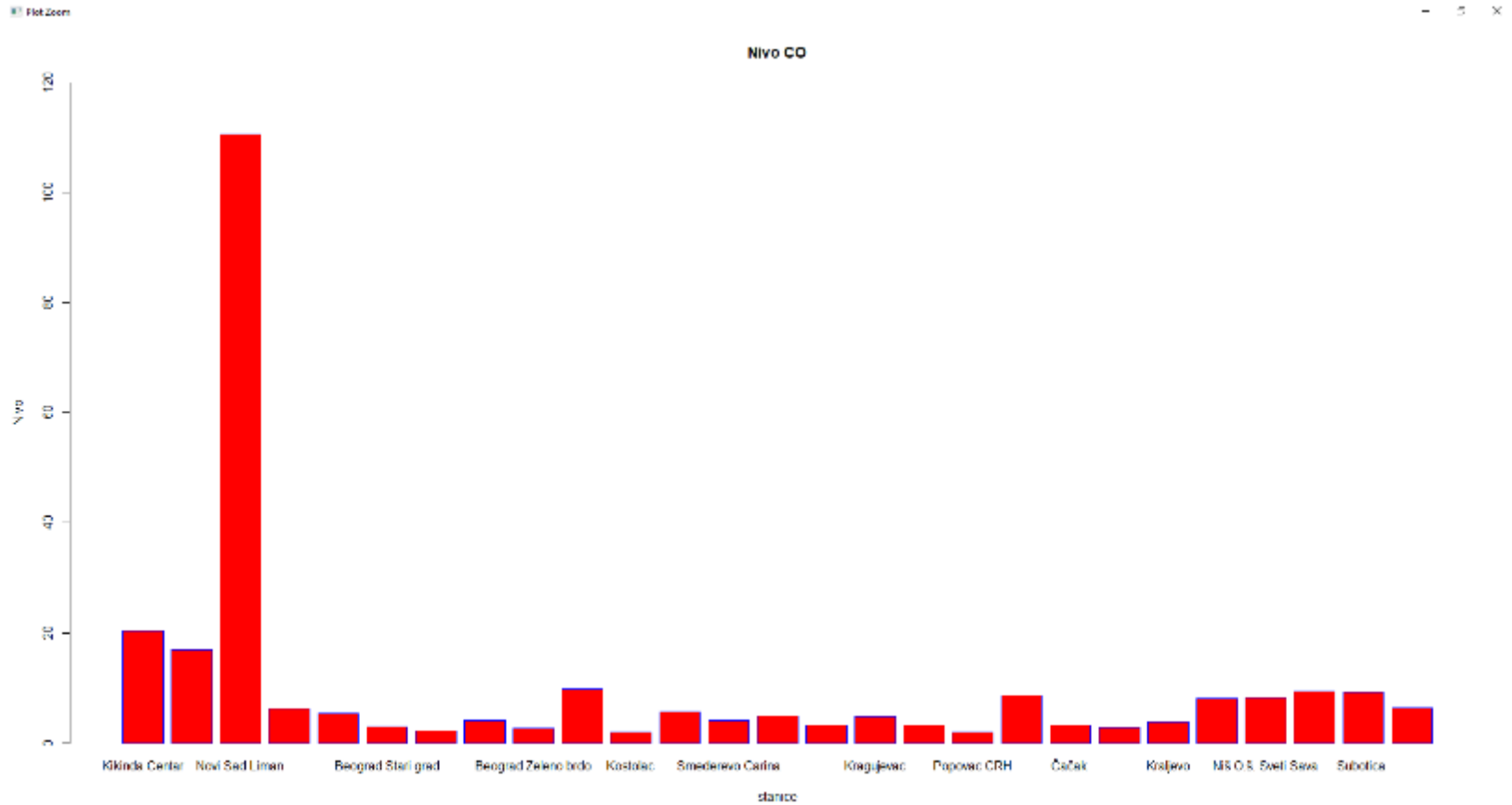
# HISTOGRAM

- Osnovni histogram u R-u se kreira pozivom funkcije *barplot()*
- Prosleđuje se tabela iz koje se prikazuju podaci, a mogu se definisati i naziv histograma, nazivi  $x$  i  $y$  osa, granice osa, itd.

```
barplot(maks.co.po.stanicama$co,  
        main = "Nivo CO",  
        xlab = "stanice",  
        ylab = "Nivo",  
        ylim = c(0,120),  
        names.arg = maks.co.po.stanicama$naziv,  
        border = "blue",  
        col = "red")
```



# HISTOGRAM



# GGPLOT2

- Biblioteka *ggplot2* se koristi za napredno grafičko prikazivanje podataka
- Postoji mnoštvo opcija i mogućnosti za prilagođavanje prikaza
- Ovde možemo prikazati, npr. vrednosti svih ekoloških parametara u Čačku tokom jednog dana merenja

```
dan = subset(cacak, subset = cacak$date ==  
"2019-03-01")
```



# GGPLOT2

- Crtamo grafik koji prikazuje sve parametre merenja, a možemo ga i sačuvati kao sliku

```
library(ggplot2)
```

```
ggplot(dan, aes(x = time, y = Vrednost, color =  
Parametar)) +
```

```
  geom_point(aes(y = no2, col="no2"),size = 5) +
```

```
  geom_point(aes(y = nox, col="nox"),size = 5) +
```

```
  geom_point(aes(y = no, col="no"),size = 5) +
```

```
  geom_point(aes(y = co, col="co"),size = 5) +
```

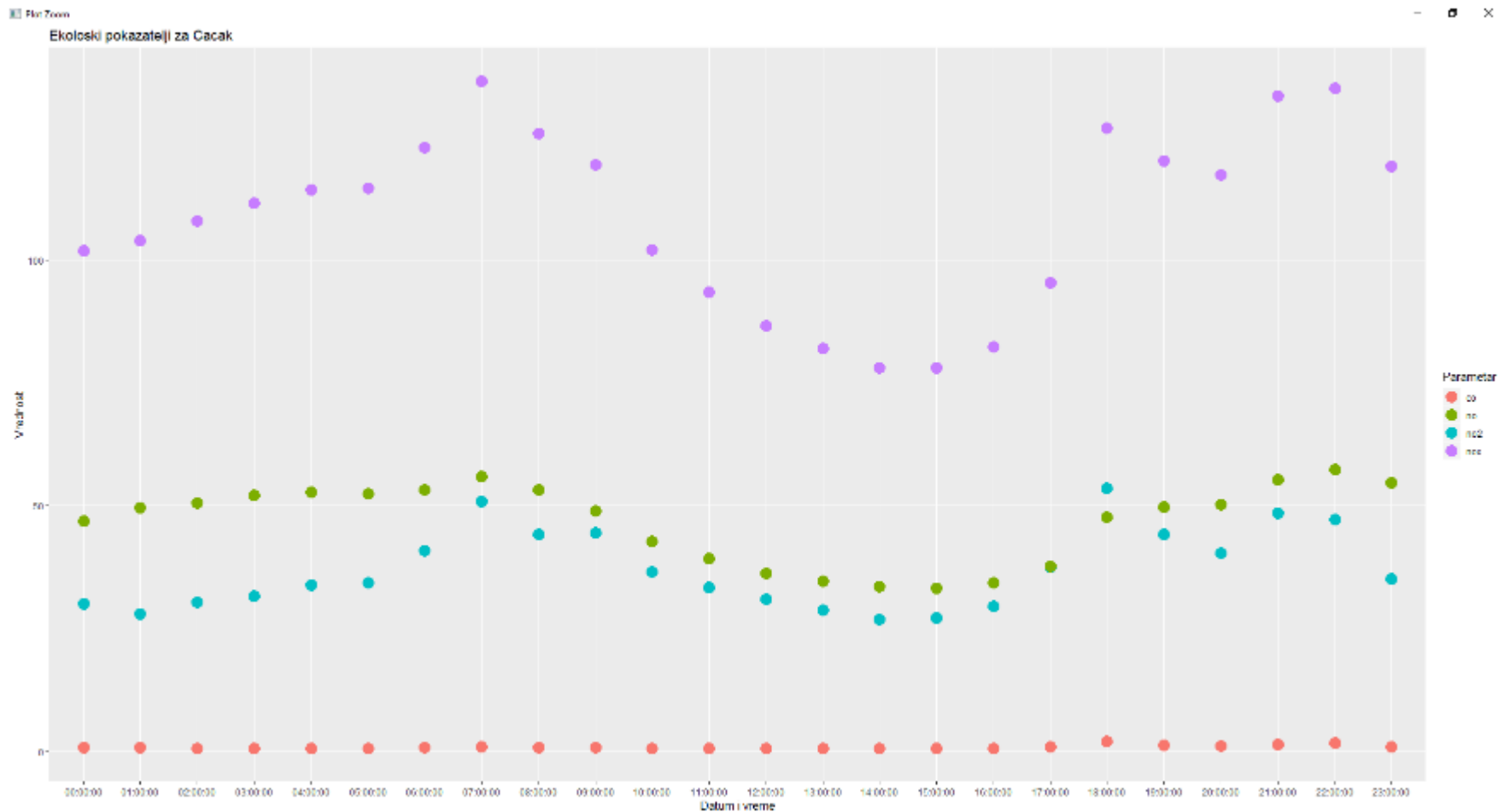
```
  labs(title = "Ekoloski pokazatelji za Cacak",
```

```
        x = "Datum i vreme")
```

```
ggsave("grafik.png")
```



# GGPLOT2



Hvala na pažnji!

