Finding Tito - Streets and squares dedicated to Tito in the former Yugoslavia

Giorgio Comai (OBC Transeuropa / Dublin City University, IICRR)

2017-09-02

Contents

Chapter 1

Introduction

Click here to see a summary of results and relevant maps

1.1 What is this all about?

This document represents an effort to find and place on a map all the names of streets and squares that are currently dedicated to Tito in the former Yugoslavia. It advances by trial and error, and suggestions for improvement are welcome.

1. First, relevant data have been extracted from OpenStreetMap. However, the resulting dataset proved to be incomplete in some areas.
2. OpenStreetMap data have been used to find common variants of street names dedicated to Tito in each of the countries of the region
3. OpenStreetMap data have been used to extract all names of villages and towns of the region
4. Google Maps has been queried with all combinations of the above, basically "asking" Google Maps if it knew of any "Tito Street", repeating the query for all inhabited locations with more than 1000 inhabitants recorded by OpenStreetMap (tens of thousands of queries in total).
5. The results have been plotted on a map

1.2 How to access this document

Data collection methods and sources are outlined in writing and data processing is presented in both a human and machine readable format. This allows for the process to be as transparent as possible, since all sources are explicitly mentioned, and all the code used for data processing and graph creation is fully explorable.

Each section introduces data that is then processed to generate composite data and graphs in the ensuing parts. All tables and graphs are available for download in multiple formats.

This document can best be explored from in its online version, currently available at https://giorgiocomai.eu/FindingTito.

A table of contents is by default always visible on the left part of the screen. Clicking on the left or right part of the screen allows to "turn pages", thus opening the previous/next section. By hovering with the mouse on the top part of the screen, it is possible to access additional customisation options, including the possibility to hide the table of contents, and to change font size and colour. There is also an edit button, which allows to suggest changes to the author (feedback is welcome also by email).

This document has been created in the R programming language using the bookdown package. The full source code of this document, including all text and code used for data processing and data visualisation is currently available on GitHub. This document is published under a Creative Commons license (CC-BY-4.0). OpenStreetMap and Google Maps data are covered by their respective licenses.

Chapter 2

Extracting data from OpenStreetMap

2.1 Installing OpenStreetMap packages

Filtering and extracting contents from OpenStreetMap dump files is facilitated by dedicated utilities. In this project, Osmconvert and Osmfilter have been used to filter relevant data.

```
# install osmfilter
if (file.exists("osmfilter")==FALSE) {
  system(command = "wget -O - http://m.m.i24.cc/osmfilter.c |cc -x c - -O3 -o osmfilter")
}

# install osmconvert
if (file.exists(file.path("osmconvert64"))==FALSE) {
  download.file(url = "http://m.m.i24.cc/osmconvert64", destfile = file.path("osmconvert64"))
  system(command = "chmod +x osmconvert64")
}
```

2.2 Download OpenStreetMap data

It is possible to download all available data for selected countries from online repositories.

```
dir.create(path = "data", showWarnings = FALSE)
dir.create(path = file.path("data", "pbf"), showWarnings = FALSE)


for (i in countries) {
  if (file.exists(file.path("data", "pbf", paste0(i, "-latest.osm.pbf")))==FALSE) {
    download.file(url = paste0("http://download.geofabrik.de/europe/", i, "-latest.osm.pbf"),
                  destfile = file.path("data", "pbf", paste0(i, "-latest.osm.pbf")))
  }
}
```

2.3 Keep only street names

Through the above-mentioned utilities, all names of street found in the region have been extracted from the dump and converted to .cvs format. This has been accomplished by filtering only the names of objects

recorded under the key `highway` in OpenStreetMap, which according to OpenStreetMap's wiki and in spite of its name, "is the main key used for identifying any kind of road, street or path."

At this stage, only streets have been included in the filter, but other OpenStreetMap categories may include relevant data, e.g. `Tag:place=square`, or `Tag:leisure=park`.

```
# convert to a format that can be read by osmfilter, and remove author data to reduce file size
dir.create(path = file.path("data", "o5m"), showWarnings = FALSE)

for (i in countries) {
  if (file.exists(file.path("data", "o5m", paste0(i, "-latest.o5m")))==FALSE) {
    system(paste0('./osmconvert64 data/pbf/', i,
                  '-latest.osm.pbf --drop-version --out-o5m -o=data/o5m/', i,
                  '-latest.o5m'))
  }
}

# filter only streets
dir.create(path = file.path("data", "o5m-streets"), showWarnings = FALSE)

for (i in countries) {
  if (file.exists(file.path("data", "o5m-streets", paste0(i, "-streets.o5m")))==FALSE) {
    system(paste0('./osmfilter data/o5m/', i, '-latest.o5m --keep="highway=*" --drop-version > ', 'data/
  }
}

# export to csv only street type, name, and lon/lat

dir.create(path = file.path("data", "csv-streets"), showWarnings = FALSE)

for (i in countries) {
  if (file.exists(file.path("data", "csv-streets", paste0(i, "-streets.csv")))==FALSE) {
    system(paste0('./osmconvert64 data/o5m-streets/', i,
                  '-streets.o5m --all-to-nodes --csv="@id @lat @lon highway name" > data/csv-streets/',
                  '-streets.csv', " --csv-separator='; '"))
  }
}
```

2.4 Keep only street names including reference to Tito

Once all street names have been extracted, only street names inlcuding reference to Tito have been kept for further analysis.

Criteria for filters:

- ends with 'Tito'
- ends with 'Tita'
- includes 'Titov'
- includes ' ' (and accordingly, also all ' ' etc., but excluding " *')

```
StreetsSummary <- data_frame(Country = countries, TotalStreets = as.integer(NA), TitoStreets = as.integ

FindTito <- "Tito$|Tita$|Titov|  |(?<! )  "
tito_all <- data_frame()
```

```
for (i in countries) {
  # Import from csv
  temp_streets <- read_delim(file = file.path("data", "csv-streets", paste0(i, "-streets.csv")),
                             delim = "; ",
                             col_names = FALSE,
                             locale = locale(decimal_mark = "."),
                             trim_ws = TRUE)
  # Store total number of streets for each country
  StreetsSummary$TotalStreets[StreetsSummary$Country==i] <- nrow(temp_streets)
  # Filter only Tito's streets
  temp_tito <- temp_streets %>%
    filter(is.na(X5)==FALSE) %>%
    filter(stringr::str_detect(string = X5, pattern = stringr::regex(pattern = FindTito, ignore_case = T
    transmute(lat = X2, lon = X3, streetname = X5)  %>%
    # remove duplicate location with which have similar coordinates (same first decimal)
    mutate(lonShort = format(lon, digit = 3), latShort = format(lat, digit = 3)) %>%
    distinct(lonShort, latShort, .keep_all = TRUE) %>%
    select(-lonShort, -latShort) %>%
    mutate(country = i)
  # Store number of Tito's streets in given country
  StreetsSummary$TitoStreets[StreetsSummary$Country==i] <- nrow(temp_tito)
  # Merge table for each country
  tito_all <- bind_rows(tito_all, temp_tito)
}

tito_all <- tito_all %>% distinct()

# tito_all <- tito_all %>%
#   mutate(lonShort = format(lon, digit = 4), latShort = format(lat, digit = 4)) %>%
#   distinct(lonShort, latShort, .keep_all = TRUE) %>%
#   select(-lonShort, -latShort)

ExportData(data = tito_all, filename = "OSM_tito_all")
```

The data are available as a spreadsheet in .csv, .xlsx, and as a data frame in R's .rds format.

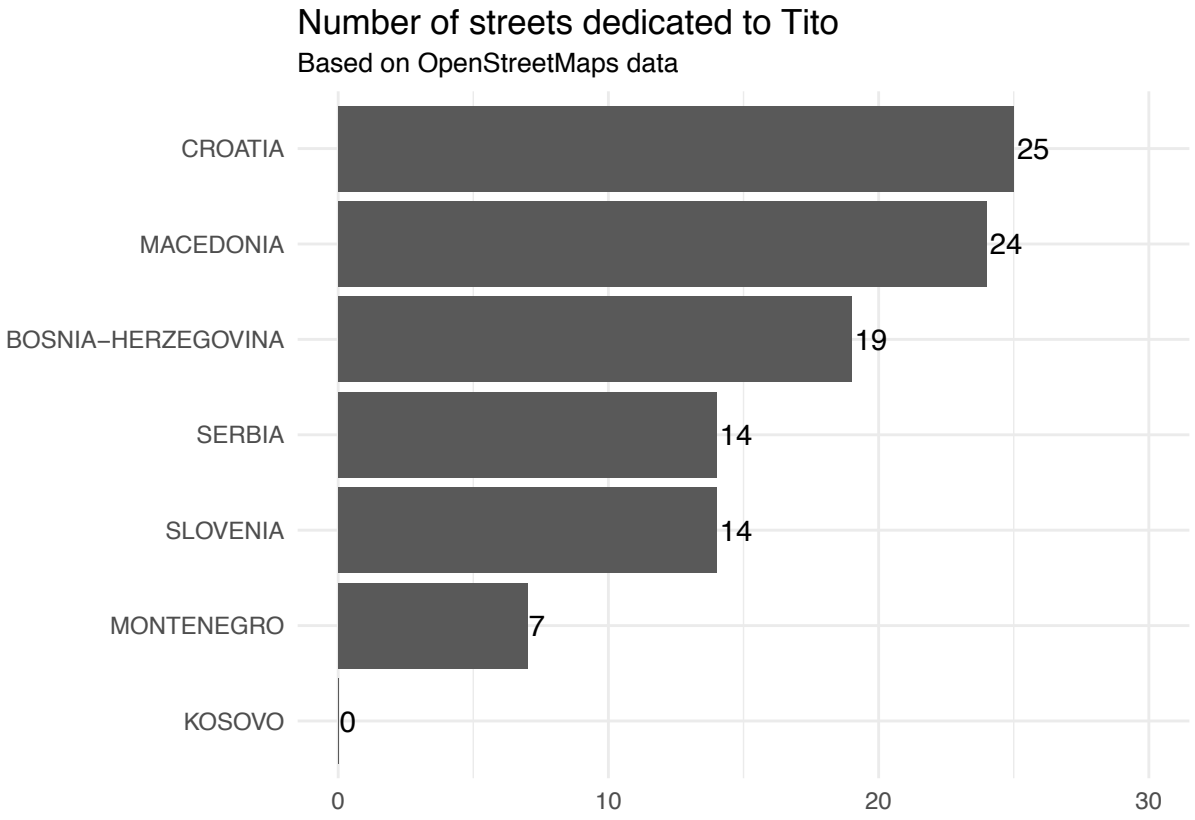
2.5 Preliminary results based on OpenStreetMap

As appears from the following sections, these data include substantial inaccuracies. They are reported here as preliminary results based on OpenStreetMap. See the following sections for more accurate data.

2.5.1 Number of streets dedicated to Tito

```
StreetsSummary %>%
  arrange(TitoStreets) %>%
  mutate(Country = toupper(Country)) %>%
  mutate(Country = forcats::fct_inorder(Country)) %>%
  ggplot(aes(x = Country, y = TitoStreets, label = TitoStreets)) +
  geom_col() +
  scale_y_continuous(name = "", limits = c(0, 30)) +
```

```
  scale_x_discrete(name = "") +
  geom_text(hjust = -0.1) +
  coord_flip() +
  theme_minimal() +
  labs(title = "Number of streets dedicated to Tito",
       subtitle = "Based on OpenStreetMaps data")
```

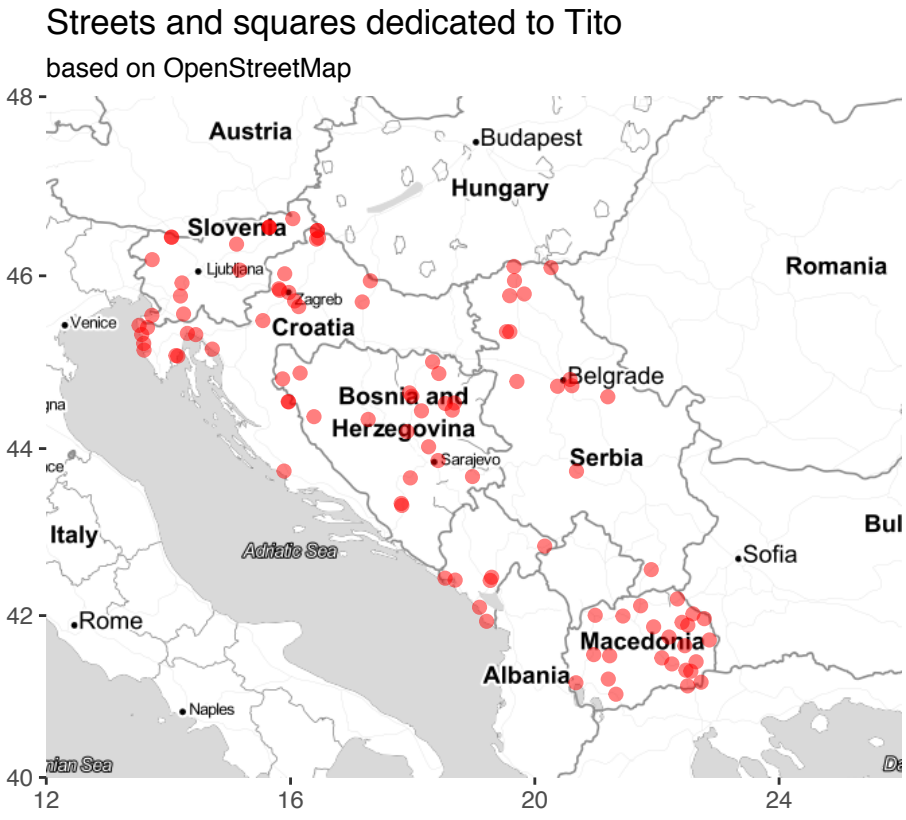



```
ShowTable(data = StreetsSummary %>% arrange(desc(TitoStreets)))
```

Country	TotalStreets	TitoStreets
croatia	3873902	25
macedonia	739505	24
bosnia-herzegovina	2077207	19
slovenia	3231464	14
serbia	2423391	14
montenegro	713605	7
kosovo	672647	0

2.6 Putting streets dedicated to Tito (according to OpenStreetMap) on a map

```
# Setting the coordinates
wb <- c(left = 12, bottom = 40, right = 26, top = 48)

# Preparing the canvas, and forcing cache
if (file.exists(file.path("temp", "mapTonerLite.rds"))==FALSE) {
  saveRDS(object = ggmap::get_stamenmap(bbox = wb, zoom = 6, maptype = "toner-lite") %>% ggmap(), file =
}

mapTonerLite <- readRDS(file = file.path("temp", "mapTonerLite.rds"))

OSM_TitoTonerLight <-  mapTonerLite + geom_point(data=tito_all, aes(x=lon, y=lat), color="red", size=2,
  labs(x = '', y = '') +
  labs(title = "Streets and squares dedicated to Tito",
       subtitle = paste("based on OpenStreetMap"))

OSM_TitoTonerLight
```

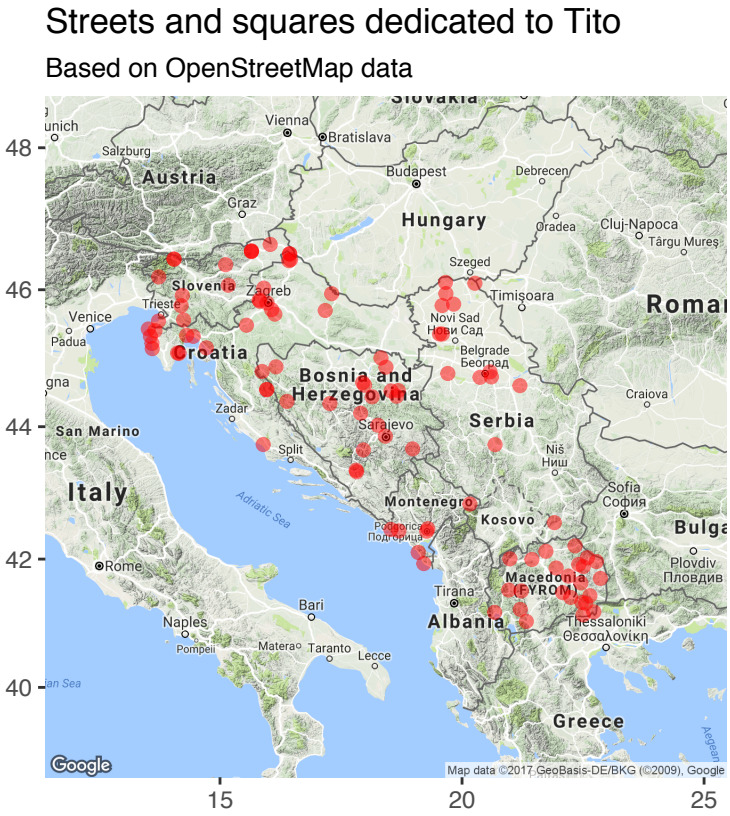



```
ExportGraph(graph = OSM_TitoTonerLight, filename = "OSM_TitoTonerLight")
```

This image is available for download in .png, .svg, and as an object in R's .rds format.

```
# Preparing the canvas
dir.create(path = "temp", showWarnings = FALSE)
```

```
## forcing cache to prevent reloading map from Google Maps each time
if (file.exists(file.path("temp", "mapG.rds"))==FALSE) {
  saveRDS(object = get_googlemap("Sarajevo", scale = 2, zoom = 6), file = file.path("temp", "mapG.rds"))
}

mapG <- readRDS(file = file.path("temp", "mapG.rds"))

ggmap(mapG) +
  geom_point(data=tito_all, aes(x=lon, y=lat), color="red", size=2, alpha=0.5) +
  labs(x = '', y = '') +
  labs(title = "Streets and squares dedicated to Tito",
       subtitle = paste("Based on OpenStreetMap data"))
```




```
ExportGraph(filename = "OSM_TitoGmaps")
```

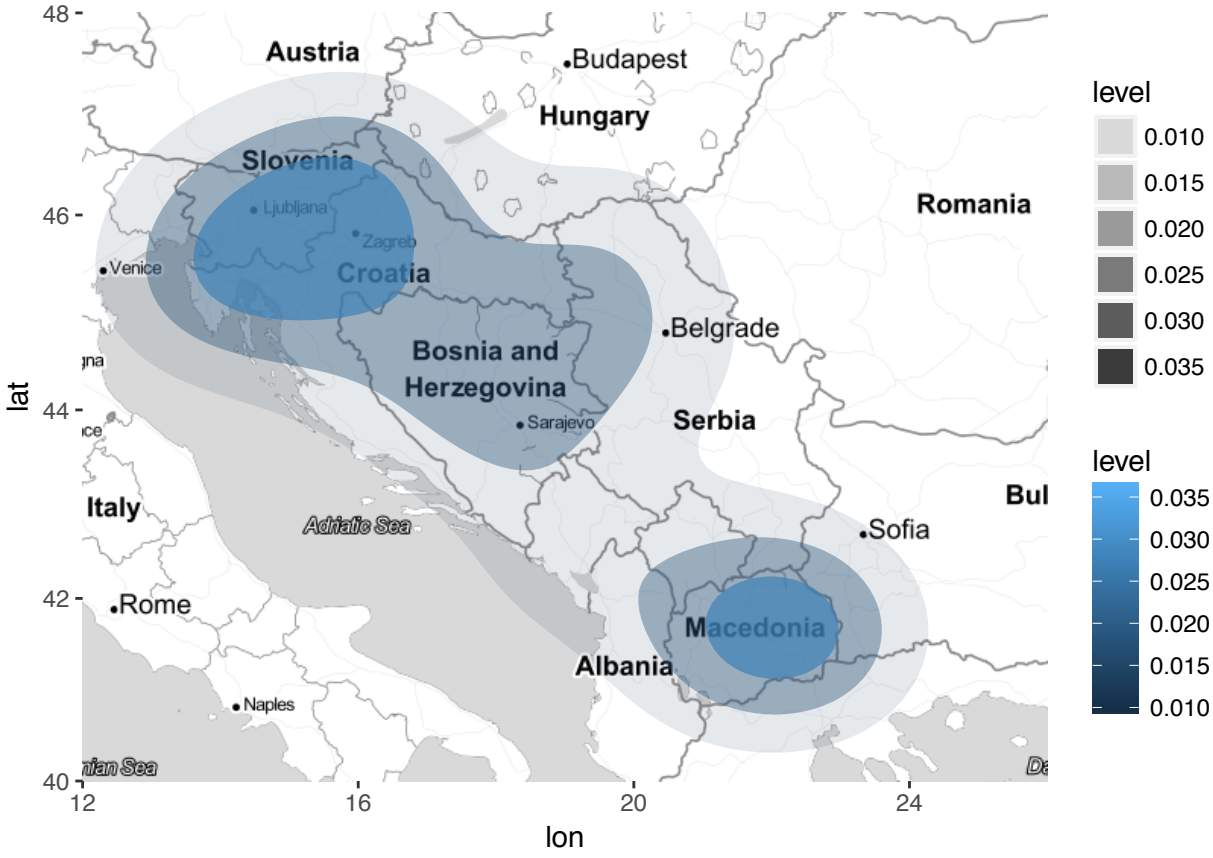
This image is available for download in .png, .svg, and as an object in R's .rds format.

```
# # Preparing the canvas
# mapWatercolor <- get_stamenmap(bbox = wb, zoom = 7, maptype = "watercolor") %>% ggmap()
# # Adding the dots
# mapWatercolor + geom_point(data=tito_all, aes(x=lon, y=lat), color="red", size=2, alpha=0.5) +
#   labs(x = '', y = '') + ggtitle("Streets dedicated to Tito")
# ggsave("TitoWatercolor.png")
```

2.7 Alternative density maps

Testing alternative visualisations.

```
mapTonerLite +
  stat_density2d(
    aes(x = lon, y = lat, fill = ..level..,  alpha = ..level..),
    size = 2, bins = 4, data = tito_all,
    geom = "polygon"
)
```



2.8 Summary of all relevant street names found on OpenStreetMap

The following table presents all versions of street names including reference to Tito found in each of the countries included in the analysis.

```
tito_all_streetname <- tito_all %>% select(country, streetname) %>% group_by(country, streetname) %>% c

ShowTable(data = tito_all_streetname)
```

country	streetname	n
bosnia-herzegovina	Maršala Tita	11
bosnia-herzegovina	Titova	4

country	streetname	n
bosnia-herzegovina	Titova ili Put Oficirske Škole	1
bosnia-herzegovina	Trg maršala Tita	1
bosnia-herzegovina	Ul Maršala Tita	1
bosnia-herzegovina	ul. Maršala Tita	1
croatia	Maršala Tita	6
croatia	Ulica Maršala Tita	4
croatia	Obala Maršala Tita	2
croatia	Titov trg	2
croatia	Hodaliste marsala tita	1
croatia	Josipa Broza Tita	1
croatia	Obala Josipa Broza Tita	1
croatia	Obala m. Tita	1
croatia	Poljana maršala Tita	1
croatia	Trg J. B. Tita	1
croatia	Trg Josipa Broza Tita	1
croatia	Trg maršala Tita	1
croatia	Trg Maršala Tita	1
croatia	Ulica Josipa Broza Tita	1
croatia	Ulica Josipa Broza-Tita	1
macedonia		14
macedonia	bul. Marsal Tito	1
macedonia	Marsal Tito	1
macedonia	Marshal Tito	1
macedonia	ul. Marshal Tito	1
macedonia		1
macedonia		1
macedonia		1
macedonia		1
macedonia	.	1
macedonia		1
montenegro	Marsala Tita	2
montenegro	Gjergj Kastrioti - Skënderbeu / Maršal Tito	1
montenegro	Josipa Broza Tita	1
montenegro	Maršala Tita	1
montenegro	Titove Korenice	1
montenegro	trg Maršala Tita	1
serbia	Maršala Tita	5
serbia		4
serbia	Aleja Maršala Tita	1
serbia	Marsala Tita	1
serbia	Ulica Marsala Tita	1
serbia		1
serbia		1
slovenia	Titova ulica	3
slovenia	Cesta maršala Tita	2
slovenia	Titova cesta	2
slovenia	Trg maršala Tita	2
slovenia	Titov most	1
slovenia	Titov trg	1
slovenia	Titov trg / Piazza Tito	1
slovenia	Titova - Nasipna	1
slovenia	Ulica Josipa Broza-Tita	1

```
ExportData(data = tito_all_streetname, filename = "tito_all_streetname")
```

The data are available as a spreadsheet in .csv, .xlsx, and as a data frame in R's .rds format.

Chapter 3

Extracting data from Google Maps

Google Maps does not allow to extract all the street names for a given country, or all streets with a given name in a country. While there may be more efficient ways (suggestions welcome), we proceed by extracting all the names of villages, towns and cities in relevant region from OpenStreetMap, and then query Google Maps for Tito street (or similar) in each of them.

3.1 Extracting all places OpenStreetMap

We use previously downloaded OpenStreetMap dumps with different filters.

```
# filter only places
dir.create(path = file.path("data", "o5m-places"), showWarnings = FALSE)

for (i in countries) {
  if (file.exists(file.path("data", "o5m-places", paste0(i, "-places.o5m")))==FALSE) {
    system(paste0('./osmfilter data/o5m/', i, '-latest.o5m --keep="place=*" --drop-version > ', 'data/o5
  }
}

# export to csv only street type, name, and lon/lat

dir.create(path = file.path("data", "csv-places"), showWarnings = FALSE)

for (i in countries) {
  if (file.exists(file.path("data", "csv-places", paste0(i, "-places.csv")))==FALSE) {
    system(paste0('./osmconvert64 data/o5m-places/', i, '-places.o5m --all-to-nodes --csv="@id @lat @lo
  }
}

all_places <- data_frame()
for (i in countries) {
  # Import from csv
  places <- read_delim(file = file.path("data", "csv-places", paste0(i, "-places.csv")), delim = "; ", c
  places <- cbind(places, i)
  all_places <- bind_rows(all_places, places)
}
colnames(all_places)  <- c("id", "lat", "lon", "place", "name", "country")
```

```
all_places <- all_places %>% filter(is.na(name)==FALSE)

ExportData(data = all_places, "all_places")
```

The data are available as a spreadsheet in .csv, .xlsx, and as a data frame in R's .rds format.

This filter provides a list of 43826 place names; testing mutiple street names for each of them would require a large (and costly) number of queries to the Google Api. We can therefore filter the data in order to include only place names that are tagged as city, town, suburb, or village. This should include all inhabited locations with more than 1000 residents, and exclude places tagged as "locality", "isolated_dwelling", and "hamlet", which are expected to be mostly irrelevant.

```
# http://wiki.openstreetmap.org/wiki/Tag:place%3Dvillage

all_places_over1000 <- all_places %>% filter(is.na(name)==FALSE) %>% filter(place == "city" | place ==
```

This more restrictive filter provides a sizable, but somewhat more managable dataset of 25881 place names.

The data are available as a spreadsheet in .csv, .xlsx, and as a data frame in R's .rds format.

3.2 What are potential street names that should be queried?

By simply querying "tito" for all place names emerging from the filter, we would likely still receive meaningful results. However, querying for potential street names should give more accurate results. We can base a list of potential street names in each country on previously extracted OpenStreetMaps data.

```
OSM_tito_all <- ImportData("OSM_tito_all")

for (i in unique(OSM_tito_all$country)) {
  ShowTable(
    OSM_tito_all %>% filter(country==i) %>% select(streetname, country) %>% count(streetname, country, s
  )
}
```

streetname	n	country
Titova ulica	3	slovenia
Cesta maršala Tita	2	slovenia
Titova cesta	2	slovenia
Trg maršala Tita	2	slovenia
Titov most	1	slovenia
Titov trg	1	slovenia
Titov trg / Piazza Tito	1	slovenia
Titova - Nasipna	1	slovenia
Ulica Josipa Broza-Tita	1	slovenia

streetname	n	country
Maršala Tita	6	croatia
Ulica Maršala Tita	4	croatia
Obala Maršala Tita	2	croatia
Titov trg	2	croatia
Hodaliste marsala tita	1	croatia
Josipa Broza Tita	1	croatia

streetname	n	country
Obala Josipa Broza Tita	1	croatia
Obala m. Tita	1	croatia
Poljana maršala Tita	1	croatia
Trg J. B. Tita	1	croatia
Trg Josipa Broza Tita	1	croatia
Trg maršala Tita	1	croatia
Trg Maršala Tita	1	croatia
Ulica Josipa Broza Tita	1	croatia
Ulica Josipa Broza-Tita	1	croatia

streetname	n	country
Maršala Tita	11	bosnia-herzegovina
Titova	4	bosnia-herzegovina
Titova ili Put Oficirske Škole	1	bosnia-herzegovina
Trg maršala Tita	1	bosnia-herzegovina
Ul Maršala Tita	1	bosnia-herzegovina
ul. Maršala Tita	1	bosnia-herzegovina

streetname	n	country
Maršala Tita	5	serbia
	4	serbia
Aleja Maršala Tita	1	serbia
Marsala Tita	1	serbia
Ulica Marsala Tita	1	serbia
	1	serbia
	1	serbia

streetname	n	country
Marsala Tita	2	montenegro
Gjergj Kastrioti - Skënderbeu / Maršal Tito	1	montenegro
Josipa Broza Tita	1	montenegro
Maršala Tita	1	montenegro
Titove Korenice	1	montenegro
trg Maršala Tita	1	montenegro

streetname	n	country
	14	macedonia
bul. Marsal Tito	1	macedonia
Marsal Tito	1	macedonia
Marshal Tito	1	macedonia
ul. Marshal Tito	1	macedonia
	1	macedonia
	1	macedonia
	1	macedonia
	1	macedonia

streetname	n	country
.	1	macedonia
	1	macedonia

Considering that if Google Maps does not find exact matches, it offers a similar result (and accordingly deals with transliteration when needed), querying for 'Titov' and 'Maršala Tita' should provide an almost complete set of cases.

Shortcomings of this approach:

- if there are towns/villages with same name, in the same country, but in different region, only one is counted (Google decides which)
- if there is more than one street in the same village with similar name (say, both a "Marshal Tito street" and a "Marshal Tito Boulevard"), then only one is counted.

3.3 Finding "titov"" on Google Maps

```
 all_places_over1000 <- all_places_over1000 %>% filter(is.na(name)==FALSE) %>%  distinct(name, country,

titovQuery <- paste("titov", all_places_over1000$name, all_places_over1000$country, sep = ", ")
```

This is the kind of queries that will be made:

```
ShowTable(head(data_frame(Query = titovQuery)))
```

Query
titov, Ljubljana, slovenia
titov, Banovci, slovenia
titov, Postojna, slovenia
titov, Piran, slovenia
titov, Izola, slovenia
titov, Kranj, slovenia

Google Maps API has a daily quota of 2500 free queries per day. We can either make 2500 queries per day (it would take more than a week for checking only "Titov" streets) or pay the 0.50 USD/per 1000 queries fee. In this case, querying for all "Titov" streets should cost less than 10 USD.

```
### if using API, uncomment this section

# saveRDS(object = "<API>", file = "GoogleApiKey.rds")
# register_google(key = readRDS("GoogleApiKey.rds"), account_type = "premium", day_limit = 50000)

## this just prepares a properly structured data frame
# titovResults <- cbind(geocode("Titov, Sarajevo, Bosnia and Herzegovina", output='more', messaging=TRU

# if (file.exists(file.path("data", "titovResults.rds")==FALSE)) {
#   for (i in seq_along(titovQuery)) {
#     temp <- try(geocode(location = titovQuery[i], output='more', messaging=TRUE))
#     Sys.sleep(time = 1.5) #wait in order to stay within API limits
#     if (is.na(temp$lon)==FALSE) {
#       temp <- cbind(temp, Query = titovQuery[i])
```

```
#       titovResults <- bind_rows(titovResults, temp)
#       # saves the results as the process goes (just in case)
#       ExportData(data = titovResults, filename = "titovResults", xlsx = FALSE)
#     }
#   }
# }
```

```
### This makes only the number of queries allowed in a given day, then it stops.
### If you re-run this another day it will pick up from where it left.



dir.create(path = "temp", showWarnings = FALSE)
# do nothing if already no free queries available
if (geocodeQueryCheck()>1) {
  if (file.exists(file.path("data", "titovResults.rds"))==FALSE) {
    #this simply aims to prepare a properly structured data frame
    titovResults <- cbind(geocode("Titov, Sarajevo, Bosnia and Herzegovina", output='more'),
                          Query = "Titov, Sarajevo, Bosnia and Herzegovina", QueryId = 0)
    start <- sum(titovProgressId, 1)
    stop <- sum(titovProgressId, geocodeQueryCheck())
    if (stop>length(titovQuery)) {
      stop <- length(titovQuery)
    }
    temp <- data_frame(lon = "")
    for (i in start:stop) {
      temp <- geocode(location = titovQuery[i], output='more', messaging=FALSE)
      Sys.sleep(time = 1.5) #wait in order to stay within API limits
      if (is.na(temp$lon)==FALSE) {
        temp <- cbind(temp, Query = titovQuery[i], QueryId = i)
        titovResults <- bind_rows(titovResults, temp)
        # saves the results as the process goes (just in case)
        ExportData(data = titovResults, filename = "titovResults", xlsx = FALSE, showDataLink = FALSE)
      }
      saveRDS(object = i, file = file.path("temp", "titovProgressId.rds"))
    }
  } else {
    # If this script has already been run, start from where it was last interrupted due to query limit
    titovResults <- ImportData(filename = "titovResults")
    titovProgressId <- readRDS(file = file.path("temp", "titovProgressId.rds"))
    if (titovProgressId<length(titovQuery)) {
      start <- sum(titovProgressId, 1)
      stop <- sum(titovProgressId, geocodeQueryCheck())
      if (stop>length(titovQuery)) {
        stop <- length(titovQuery)
      }
      temp <- data_frame(lon = "")
      for (i in start:stop) {
        # If it receives an "over_query_limit" warning then skip
        if (temp$lon!="OVER_QUERY_LIMIT") {
          # makes sure over quota is properly dealt with: if over quota, just skips
          temp <- tryCatch(expr = geocode(location = titovQuery[i], output='more', messaging=FALSE), wa
            msg <- conditionMessage(w)
            if (grepl(pattern = "OVER_QUERY_LIMIT", x = msg) == TRUE) {
```

```
            return(data_frame(lon = "OVER_QUERY_LIMIT", lat = "OVER_QUERY_LIMIT"))
          } else if (grepl(pattern = "ZERO_RESULTS", x = msg) == TRUE) {
            return(data_frame(lon = "ZERO_RESULTS", lat = "ZERO_RESULTS"))
          } else {
            return(data_frame(lon = msg, lat = msg))
          }
        })
        if (temp$lon=="OVER_QUERY_LIMIT") {
          # do nothing really
        } else {
          Sys.sleep(time = 1.5) #wait in order to stay within API limits
          if (is.na(temp$lon)==FALSE & temp$lon!="ZERO_RESULTS") {
            temp <- cbind(temp, Query = titovQuery[i], QueryId = i)
            titovResults <- bind_rows(titovResults, temp)
            # saves the results as the process goes, so it can be stopped anytime and nothing is lost
            ExportData(data = titovResults, filename = "titovResults", xlsx = FALSE, showDataLink = F
          }
          saveRDS(object = i, file = file.path("temp", "titovProgressId.rds"))
        }
      }
    }
  }
}
```

3.4 Querying separately for 'Marshal tito' (' ')

Given the "popularity" of Maršala Tita/ , even if many have been already captured by querying for Titov, the same process is now repeated for Maršala Tita (' ' in Macedonia). Here are some examples of the queries that will be made.

```
marsalaTitaQuery <- paste("Maršala Tita", all_places_over1000$name, all_places_over1000$country, sep = "
marsalaTitaQuery[grepl(pattern = ", macedonia", x = marsalaTitaQuery)] <- gsub(pattern = "Maršala Tita,

head(x = marsalaTitaQuery)
```

[1] "Maršala Tita, Ljubljana, slovenia" "Maršala Tita, Banovci, slovenia"
[3] "Maršala Tita, Postojna, slovenia" "Maršala Tita, Piran, slovenia"
[5] "Maršala Tita, Izola, slovenia" "Maršala Tita, Kranj, slovenia"

```
head(x = marsalaTitaQuery[grepl(pattern = ", macedonia", x = marsalaTitaQuery)])
```

[1] " , , macedonia" " , , macedonia"
[3] " , , macedonia" " , , macedonia" [5] " , , macedonia" " , , macedonia"

```
dir.create(path = "temp", showWarnings = FALSE)
# do nothing if already no free queries available
if (geocodeQueryCheck()>1) {
  if (file.exists(file.path("data", "marsalaTitaResults.rds"))==FALSE) {
    #this simply aims to prepare a properly structured data frame
    marsalaTitaResults <- cbind(geocode("Maršala Tita, Sarajevo, Bosnia and Herzegovina", output='more'
    start <- sum(1)
    stop <- geocodeQueryCheck()
```

```
  if (stop>length(marsalaTitaQuery)) {
    stop <- length(marsalaTitaQuery)
  }
  temp <- data_frame(lon = "")
  for (i in start:stop) {
    temp <- geocode(location = marsalaTitaQuery[i], output='more', messaging=FALSE)
    Sys.sleep(time = 1.5) #wait in order to stay within API limits
    if (is.na(temp$lon)==FALSE) {
      temp <- cbind(temp, Query = marsalaTitaQuery[i], QueryId = i)
      marsalaTitaResults <- bind_rows(marsalaTitaResults, temp)
      # saves the results as the process goes (just in case)
      ExportData(data = marsalaTitaResults, filename = "marsalaTitaResults", xlsx = FALSE, showDataLi
    }
    saveRDS(object = i, file = file.path("temp", "marsalaTitaProgressId.rds"))
  }
} else {
  # If this script has already been run, start from where it was last interrupted due to query limit
  marsalaTitaResults <- ImportData(filename = "marsalaTitaResults")
  marsalaTitaProgressId <- readRDS(file = file.path("temp", "marsalaTitaProgressId.rds"))
  if (marsalaTitaProgressId<length(marsalaTitaQuery)) {
    start <- sum(marsalaTitaProgressId, 1)
    stop <- sum(marsalaTitaProgressId, geocodeQueryCheck())
    if (stop>length(marsalaTitaQuery)) {
      stop <- length(marsalaTitaQuery)
    }
    temp <- data_frame(lon = "")
    if (start!=stop) {
      for (i in start:stop) {
        # If it receives an "over_quey limit" warning then skip
        if (temp$lon!="OVER_QUERY_LIMIT") {
          # makes sure over quota is properly dealt with: if over quota, just skips
          temp <- tryCatch(expr = geocode(location = marsalaTitaQuery[i], output='more', messaging=FA
            msg <- conditionMessage(w)
            if (grepl(pattern = "OVER_QUERY_LIMIT", x = msg) == TRUE) {
              return(data_frame(lon = "OVER_QUERY_LIMIT", lat = "OVER_QUERY_LIMIT"))
            } else if (grepl(pattern = "ZERO_RESULTS", x = msg) == TRUE) {
              return(data_frame(lon = "ZERO_RESULTS", lat = "ZERO_RESULTS"))
            } else {
              return(data_frame(lon = msg, lat = msg))
            }
          })
          if (temp$lon=="OVER_QUERY_LIMIT") {
            # do nothing really
          } else {
            Sys.sleep(time = 1.5) #wait in order to stay within API limits
            if (is.na(temp$lon)==FALSE & temp$lon!="ZERO_RESULTS") {
              temp <- cbind(temp, Query = marsalaTitaQuery[i], QueryId = i)
              marsalaTitaResults <- bind_rows(marsalaTitaResults, temp)
              # saves the results as the process goes, so it can be stopped anytime and nothing is lo
              ExportData(data = marsalaTitaResults, filename = "marsalaTitaResults", xlsx = FALSE, sh
            }
            saveRDS(object = i, file = file.path("temp", "marsalaTitaProgressId.rds"))
          }
```

```
          }
        }
      }
    }
  }
}
```

3.5 Polishing the results

Removing results included multiple times, and results that are not streets or squares.

```
titovResults <- ImportData(filename = "titovResults")
marsalaTitaResults <- ImportData(filename = "marsalaTitaResults" )

TitoGmapsResults <- bind_rows(titovResults, marsalaTitaResults) %>%
  filter(type=="route", country != "Italy") %>% # exclude non-YU and non streets/squares
  filter(grepl(pattern = "Tit|T ", x = route)) %>% # remove most non-tito
  filter(!grepl(pattern = "Strozzi|Brezova", x = route)) %>%  # remove remaining non-Tito
  distinct(address, .keep_all = TRUE) %>% # remove those with same address
  distinct(locality, route, .keep_all = TRUE) %>% #remove same locality, same street name
  distinct(lon, lat, route)

ExportData(data = TitoGmapsResults, filename = "TitoGmapsResults")
```

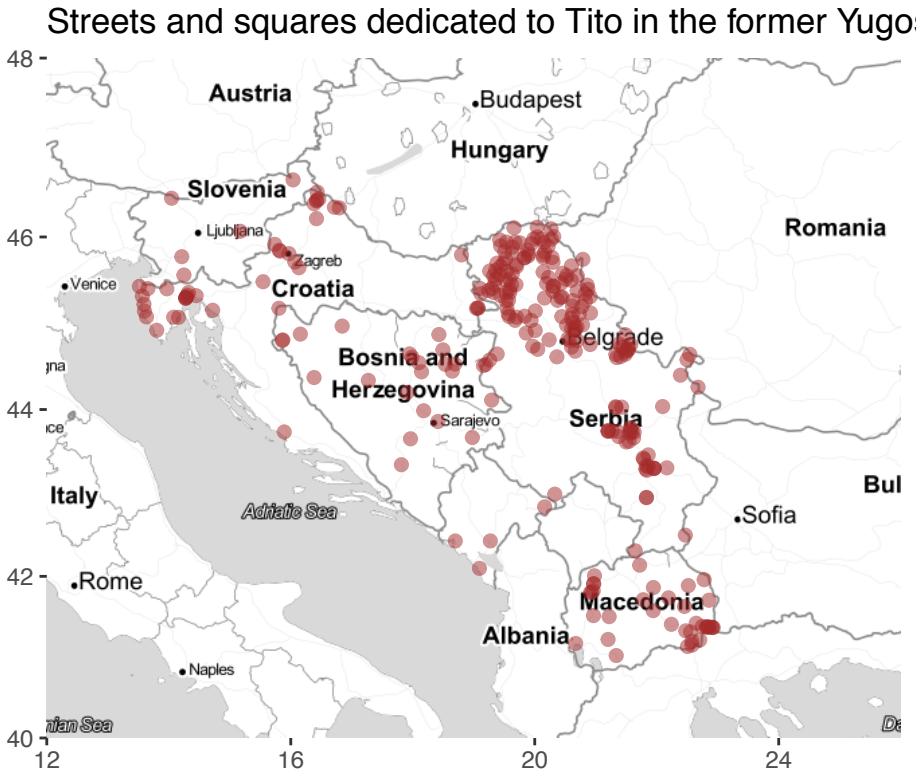
The data are available as a spreadsheet in .csv, .xlsx, and as a data frame in R's .rds format.

Chapter 4

Where is Tito?

4.1 See the results on a map

```
TitoGmapsResults <- ImportData(filename = "TitoGmapsResults")
# Preparing the canvas, and forcing cache
wb <- c(left = 12, bottom = 40, right = 26, top = 48)
if (file.exists(file.path("temp", "mapTonerLite.rds"))==FALSE) {
  saveRDS(object = ggmap::get_stamenmap(bbox = wb, zoom = 6, maptype = "toner-lite") %>% ggmap(), file =
}

mapTonerLite <- readRDS(file = file.path("temp", "mapTonerLite.rds"))

TitoGmapsResults_point_gg <-
  mapTonerLite + geom_point(data=TitoGmapsResults, aes(x=lon, y=lat), color="brown", size=2, alpha=0.5)
  labs(x = '', y = '') + labs(title = "Streets and squares dedicated to Tito in the former Yugoslavia",

TitoGmapsResults_point_gg
```



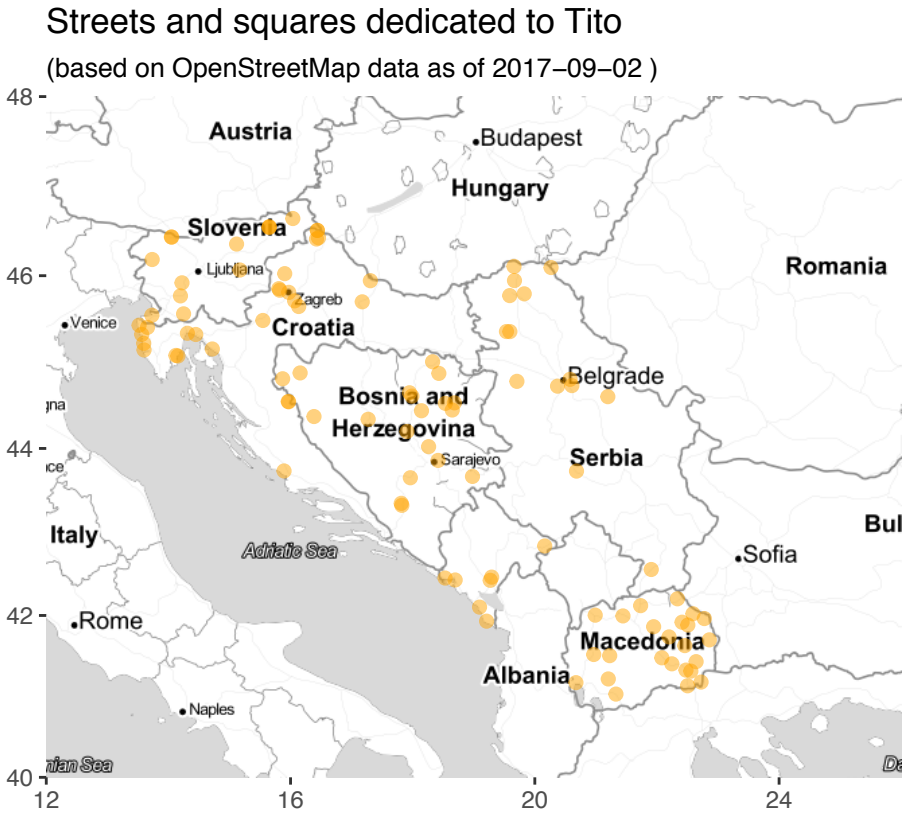
Source: Google Maps; https://giorgiocomai.eu/FindingTito

```
ExportGraph(graph = TitoGmapsResults_point_gg, filename = "TitoGmapsResults_point_gg")
```

This image is available for download in .png, .svg, and as an object in R's .rds format.

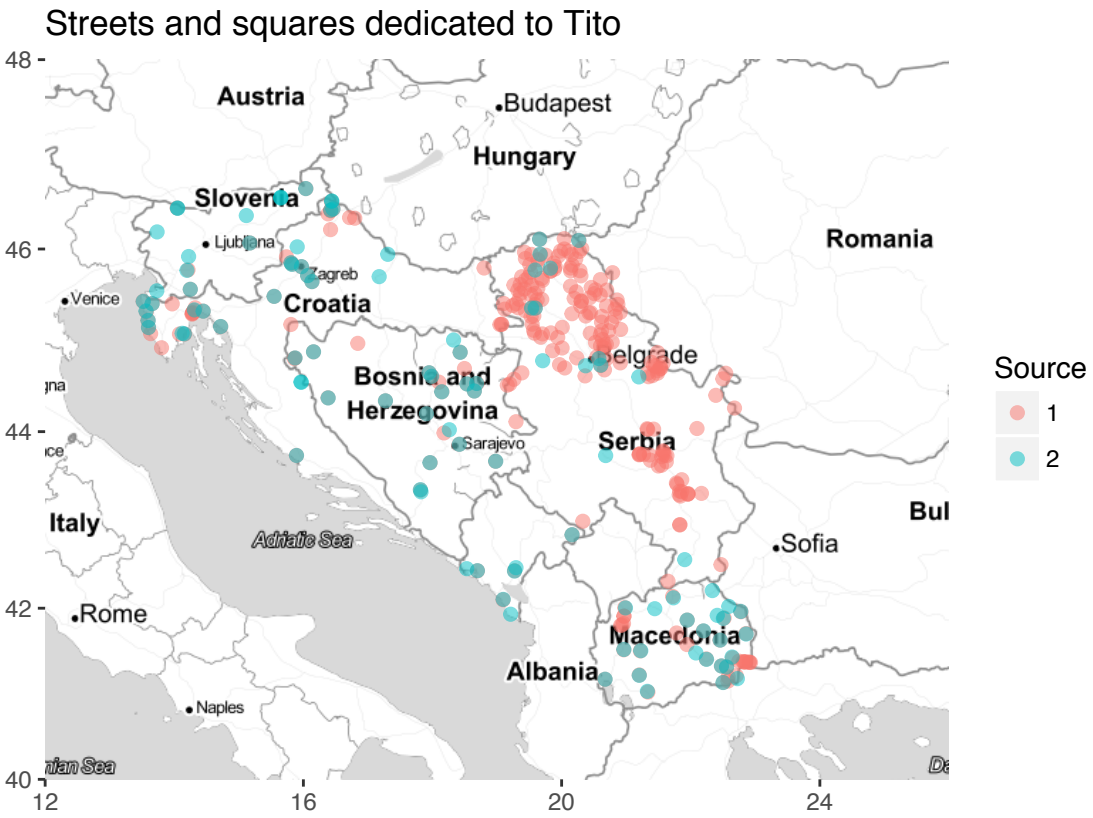
Comparing with OpenStreetMaps results

```
mapTonerLite + geom_point(data=ImportData("OSM_tito_all"), aes(x=lon, y=lat), color="orange", size=2, a
  labs(x = '', y = '') + labs(title = "Streets and squares dedicated to Tito", subtitle = paste("(based
```



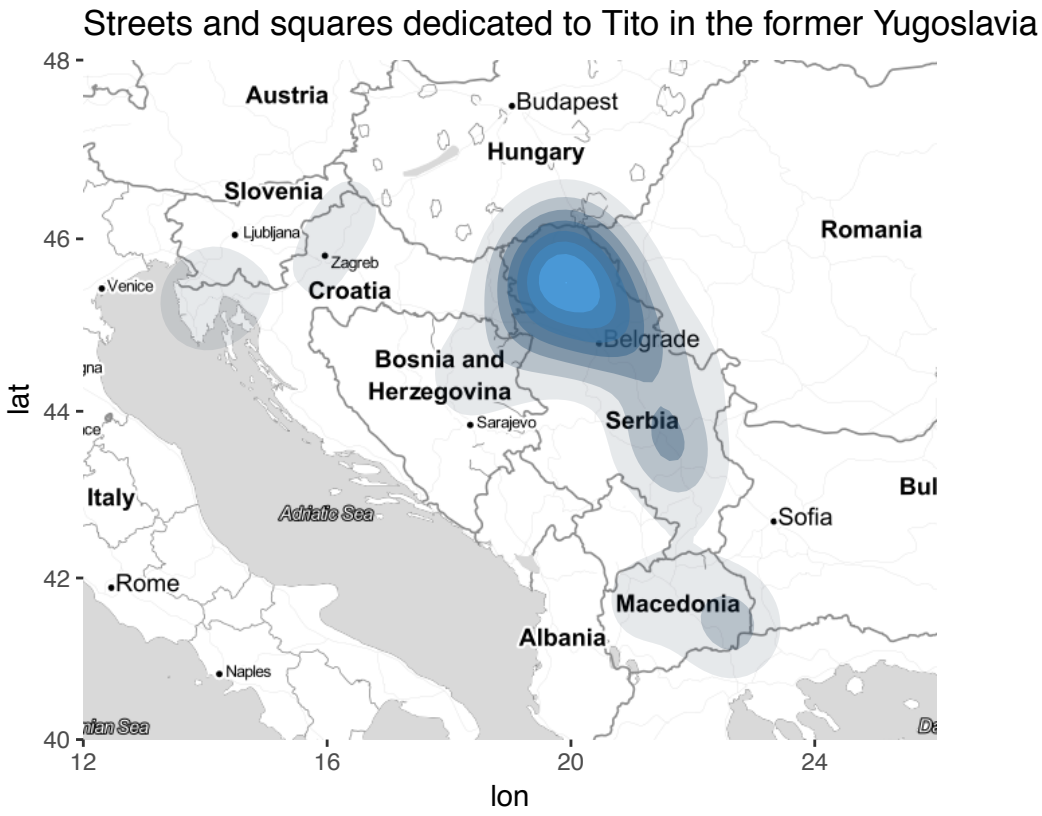
On the same map:

```
OSM_gmaps_Tito <- bind_rows(TitoGmapsResults, ImportData("OSM_tito_all") %>% select(lon, lat), .id = "So

mapTonerLite + geom_point(data=OSM_gmaps_Tito, aes(x=lon, y=lat, color = Source), size=2, alpha=0.5) +
  labs(x = '', y = '') + labs(title = "Streets and squares dedicated to Tito")
```



4.2 Density visualisations

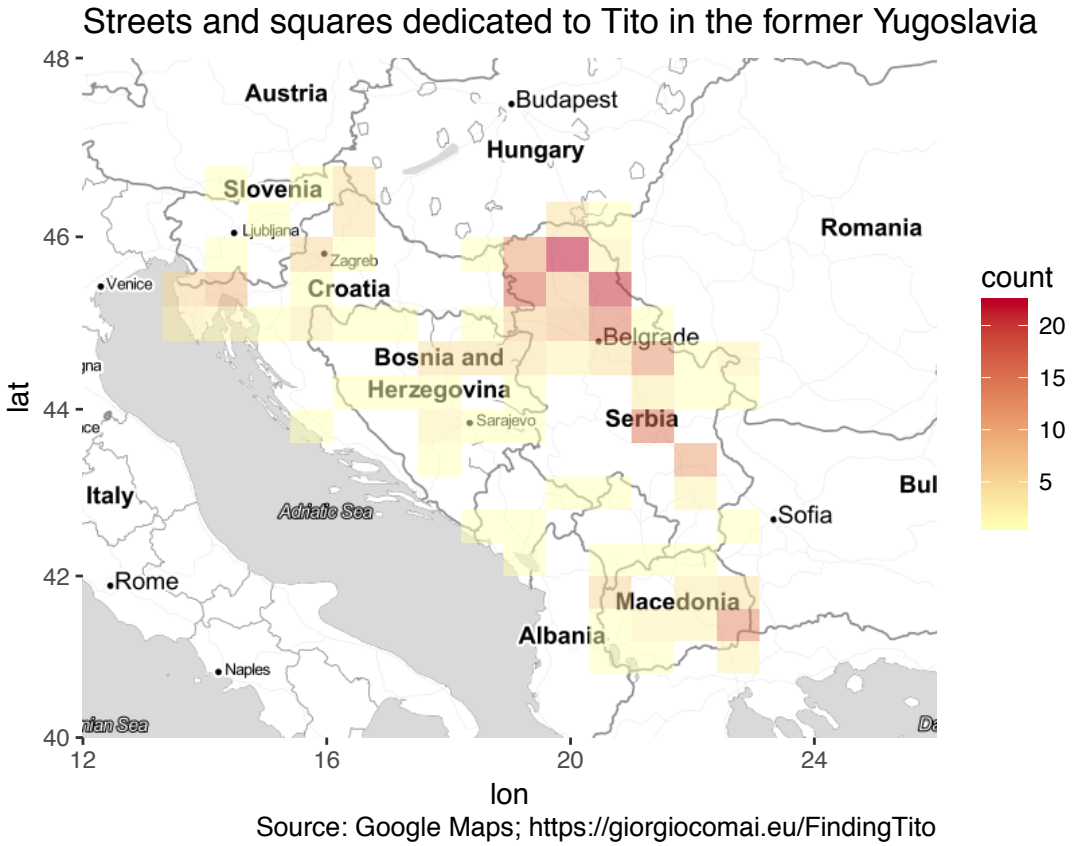
```
TitoGmaps_density_gg <- mapTonerLite +
  stat_density2d(
    aes(x = lon, y = lat, fill = ..level..,  alpha = ..level..),
    size = 2, bins = 8, data = TitoGmapsResults,
    geom = "polygon"
) + labs(title = "Streets and squares dedicated to Tito in the former Yugoslavia", caption = "Source: G

TitoGmaps_density_gg
```




```
ExportGraph(graph = TitoGmaps_density_gg, filename = "TitoGmaps_density_gg")
```

This image is available for download in .png, .svg, and as an object in R's .rds format.

```
TitoGmaps_density_sq_gg <-
  mapTonerLite + stat_bin2d(
    aes(x = lon, y = lat),
    size = .5, bins = 20, alpha = 1/2,
    data = TitoGmapsResults) +
  scale_fill_gradient(low = "#ffffb2", high = "#bd0026") +
  labs(title = "Streets and squares dedicated to Tito in the former Yugoslavia", caption = "Source: Goo

TitoGmaps_density_sq_gg
```




```
ExportGraph(graph = TitoGmaps_density_sq_gg, filename = "TitoGmaps_density_sq_gg")
```

This image is available for download in .png, .svg, and as an object in R's .rds format.

4.3 By country / Administrative level

At the sub-country level, Google Maps records two levels of administrative sub-divisions. Unfortunately, not all streets are recorded with the respective sub-division, and the categorisation does not seem to be consistent.

N.B.: names of countries and administrative units are here presented exactly as they are outputted by the Google APIs. The author is thus not responsible for the uneven trasliteration, or for the names chosen to refer to countries or administrative units.

```
TitoGmapsMore <- bind_rows(ImportData(filename = "titovResults"), ImportData("marsalaTitaResults")) %>%
  filter(type=="route", country != "Italy") %>% # exclude non-YU and non streets/squares
  filter(grepl(pattern = "Tit|T ", x = route)) %>% # remove most non-tito
  filter(!grepl(pattern = "Strozzi|Brezova", x = route)) %>%  # remove remaining non-Tito
  distinct(address, .keep_all = TRUE) %>% # remove those with same address
  distinct(locality, route, .keep_all = TRUE) %>% #remove same locality, same street name
  distinct(lon, lat, route, administrative_area_level_1, administrative_area_level_2, country)

TitoGmapsByCountry <- bind_rows(TitoGmapsMore %>% count(country, sort = TRUE),
                                data_frame(country = "Kosovo", n = 0))

# manually adding Kosovo = 0
```

```
ShowTable(data = TitoGmapsByCountry, caption = "Streets and squares dedicated to Tito in the former Yug
```

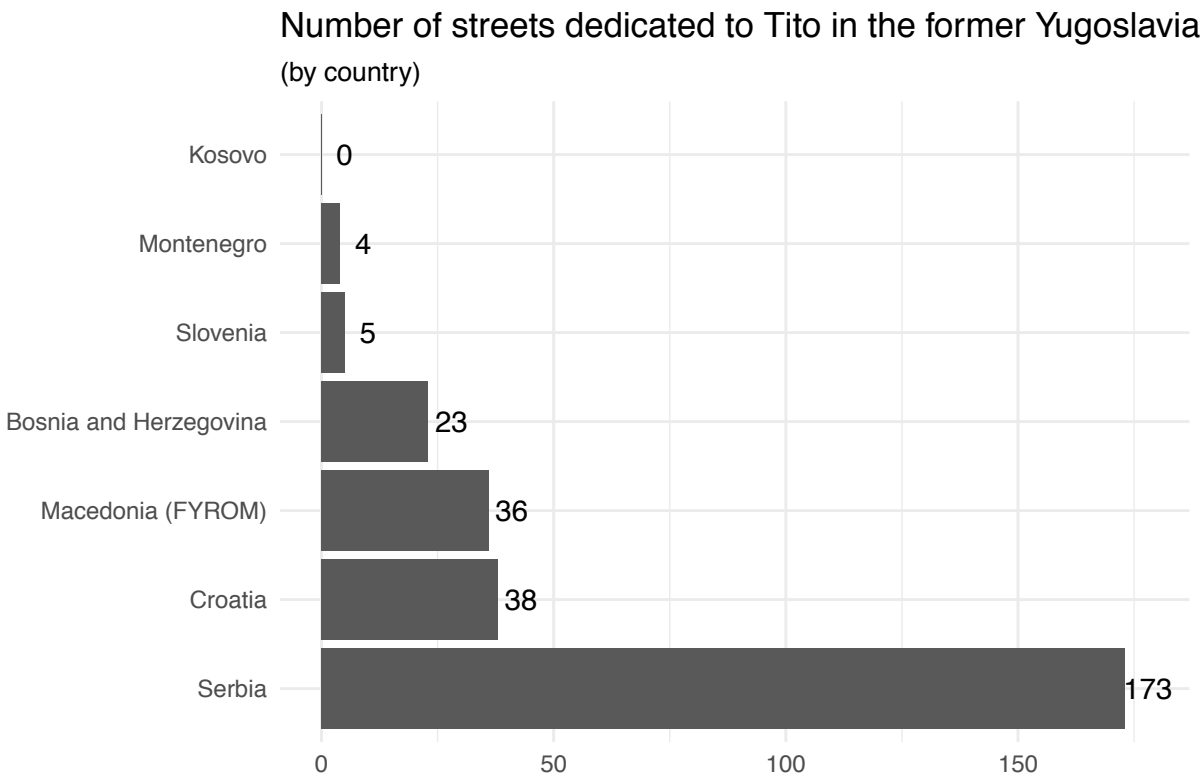
Table 4.1: Streets and squares dedicated to Tito in the former Yugoslavia, by country, according to Google Maps/#FindingTito

country	n
Serbia	173
Croatia	38
Macedonia (FYROM)	36
Bosnia and Herzegovina	23
Slovenia	5
Montenegro	4
Kosovo	0

```
ExportData(data = TitoGmapsByCountry, filename = "TitoGmapsByCountry")
```

The data are available as a spreadsheet in .csv, .xlsx, and as a data frame in R's .rds format.

```
TitoGmapsByCountry_gg <- ImportData(filename = "TitoGmapsByCountry") %>% mutate(country = forcats::fct_
  ggplot(mapping = aes(x = country, y = n, label = n)) +
  geom_col() +
  geom_text(nudge_y = 5) +
  scale_x_discrete(name = "") +
  scale_y_continuous(name = "") +
  theme_minimal() +
  coord_flip() +
  labs(title = "Number of streets dedicated to Tito in the former Yugoslavia", subtitle = "(by country)

TitoGmapsByCountry_gg
```



Source: Google Maps; https://giorgiocomai.eu/FindingTito

```
ExportGraph(graph = TitoGmapsByCountry_gg, filename = "TitoGmapsByCountry_gg")
```

This image is available for download in .png, .svg, and as an object in R's .rds format.

```
ShowTable(TitoGmapsMore %>% count(administrative_area_level_1, country, sort = TRUE))
```

administrative_area_level_1	country	n
Vojvodina	Serbia	112
NA	Serbia	61
Federacija Bosne i Hercegovine	Bosnia and Herzegovina	21
Istarska županija	Croatia	10
Primorsko-goranska županija	Croatia	8
Međimurska županija	Croatia	7
Municipality of Novo Selo	Macedonia (FYROM)	6
Gostivar	Macedonia (FYROM)	3
Vukovarsko-srijemska županija	Croatia	3
Zagrebačka županija	Croatia	3
NA	Croatia	3
Municipality of Bogdanci	Macedonia (FYROM)	2
Republika Srpska	Bosnia and Herzegovina	2
Strumitsa	Macedonia (FYROM)	2
Valandovo	Macedonia (FYROM)	2
NA	Macedonia (FYROM)	2
Delchevo	Macedonia (FYROM)	1
Dojran	Macedonia (FYROM)	1
Glavni grad Podgorica	Montenegro	1

administrative_area_level_1	country	n
Gornja Radgona	Slovenia	1
Ilirska Bistrica	Slovenia	1
Jesenice	Slovenia	1
Karlovačka županija	Croatia	1
Laško	Slovenia	1
Municipality of Berovo	Macedonia (FYROM)	1
Municipality of Bitola	Macedonia (FYROM)	1
Municipality of Bogovinje	Macedonia (FYROM)	1
Municipality of Demir Hisar	Macedonia (FYROM)	1
Municipality of Demir Kapija	Macedonia (FYROM)	1
Municipality of Gevgelija	Macedonia (FYROM)	1
Municipality of Gradsko	Macedonia (FYROM)	1
Municipality of Kichevo	Macedonia (FYROM)	1
Municipality of Kumanovo	Macedonia (FYROM)	1
Municipality of Makedonski Brod	Macedonia (FYROM)	1
Municipality of Shtip	Macedonia (FYROM)	1
Municipality of Struga	Macedonia (FYROM)	1
Municipality of Veles	Macedonia (FYROM)	1
Municipality of Vinica	Macedonia (FYROM)	1
Opština Rožaje	Montenegro	1
Opština Tivat	Montenegro	1
Osječko-baranjska županija	Croatia	1
Postojna	Slovenia	1
Radovish	Macedonia (FYROM)	1
Šibensko-kninska županija	Croatia	1
Sveti Nikole	Macedonia (FYROM)	1
Tetovo	Macedonia (FYROM)	1
Varaždinska županija	Croatia	1
NA	Montenegro	1

```
ExportData(data = TitoGmapsMore %>% count(administrative_area_level_1, country, sort = TRUE), filename =
```

The data are available as a spreadsheet in .csv, .xlsx, and as a data frame in R's .rds format.

```
ShowTable(TitoGmapsMore %>% count(administrative_area_level_2, country, sort = TRUE))
```

administrative_area_level_2	country	n
NA	Macedonia (FYROM)	36
Srednjobanatski okrug	Serbia	26
Nišavski okrug	Serbia	19
Južnobački okrug	Serbia	18
Severnobanatski okrug	Serbia	16
Južnobanatski okrug	Serbia	15
Braničevski okrug	Serbia	12
Severnobački okrug	Serbia	11
Sremski krug	Serbia	11
Zapadnobački okrug	Serbia	11
Grad Beograd	Serbia	6
Rasinski Okrug	Serbia	6
NA	Croatia	6
Borski okrug	Serbia	5
Tuzlanski kanton	Bosnia and Herzegovina	5

administrative_area_level_2	country	n
Zeničko-dobojski kanton	Bosnia and Herzegovina	5
NA	Serbia	5
NA	Slovenia	5
Mačvanski okrug	Serbia	4
Unsko-sanski kanton	Bosnia and Herzegovina	4
NA	Montenegro	4
Općina Nijemci	Croatia	3
Pomoravski okrug	Serbia	3
Hercegovačko-neretvanski kanton	Bosnia and Herzegovina	2
Jablanički okrug	Serbia	2
Općina Lovran	Croatia	2
Općina Opatija	Croatia	2
Općina Velika Gorica	Croatia	2
Pčinjski Okrug	Serbia	2
NA	Bosnia and Herzegovina	2
Bosansko-podrinjski kanton	Bosnia and Herzegovina	1
Kanton 10	Bosnia and Herzegovina	1
Kanton Sarajevo	Bosnia and Herzegovina	1
Matulji	Croatia	1
Općina Buje	Croatia	1
Općina Buzet	Croatia	1
Općina Čakovec	Croatia	1
Općina Crikvenica	Croatia	1
Općina Donji Vidovec	Croatia	1
Općina Fažana	Croatia	1
Općina Karlovac	Croatia	1
Općina Kneževi Vinogradi	Croatia	1
Općina Labin	Croatia	1
Općina Marija Gorica	Croatia	1
Općina Mursko Središće	Croatia	1
Općina Nedelišće	Croatia	1
Općina Novigrad	Croatia	1
Općina Poreč	Croatia	1
Općina Raša	Croatia	1
Općina Rijeka	Croatia	1
Općina Rovinj	Croatia	1
Općina Šibenik	Croatia	1
Općina Sveta Marija	Croatia	1
Općina Umag	Croatia	1
Općina Varaždinske Toplice	Croatia	1
Općina Vrsar	Croatia	1
Raški okrug	Serbia	1
Srednjobosanski kanton	Bosnia and Herzegovina	1
Zeničko-dobojska županija	Bosnia and Herzegovina	1

```
ExportData(data = TitoGmapsMore %>% count(administrative_area_level_2, country, sort = TRUE), filename =
```

The data are available as a spreadsheet in .csv, .xlsx, and as a data frame in R's .rds format.

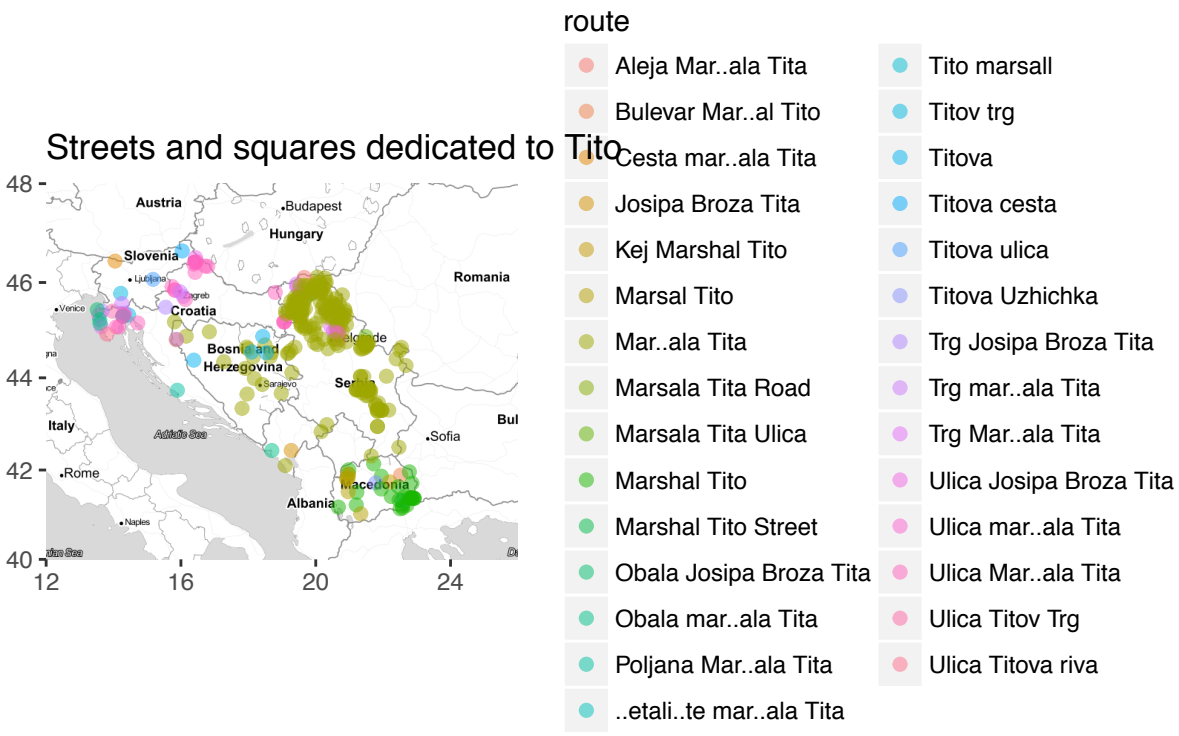
4.4 By street name

This is colourful, but not very useful, since it mostly points at the different ways in which Google Maps stores street names.

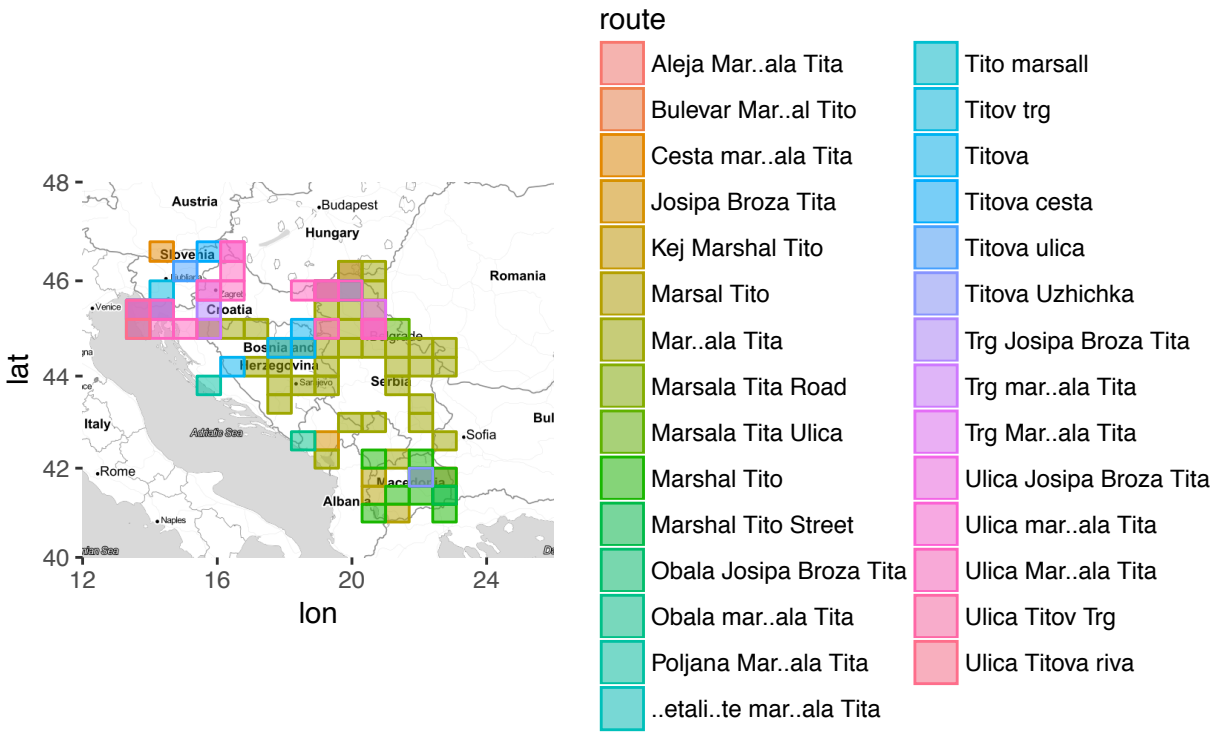
```
ShowTable(TitoGmapsResults %>% count(route, sort = TRUE))
```

route	n
Maršala Tita	179
Marshal Tito	25
Ulica Maršala Tita	25
Marsal Tito	7
Trg Maršala Tita	6
Trg maršala Tita	5
Titova	4
Obala maršala Tita	3
Josipa Broza Tita	2
Titov trg	2
Trg Josipa Broza Tita	2
Ulica Josipa Broza Tita	2
Aleja Maršala Tita	1
Bulevar Maršal Tito	1
Cesta maršala Tita	1
Kej Marshal Tito	1
Marsala Tita Road	1
Marsala Tita Ulica	1
Marshal Tito Street	1
Obala Josipa Broza Tita	1
Poljana Maršala Tita	1
Šetalište maršala Tita	1
Tito marsall	1
Titova cesta	1
Titova ulica	1
Titova Uzhichka	1
Ulica maršala Tita	1
Ulica Titov Trg	1
Ulica Titova riva	1

```
mapTonerLite +
  geom_point(data=TitoGmapsResults, aes(x=lon, y=lat, color = route), size=2, alpha=0.5) +
  labs(x = '', y = '') + labs(title = "Streets and squares dedicated to Tito")
```

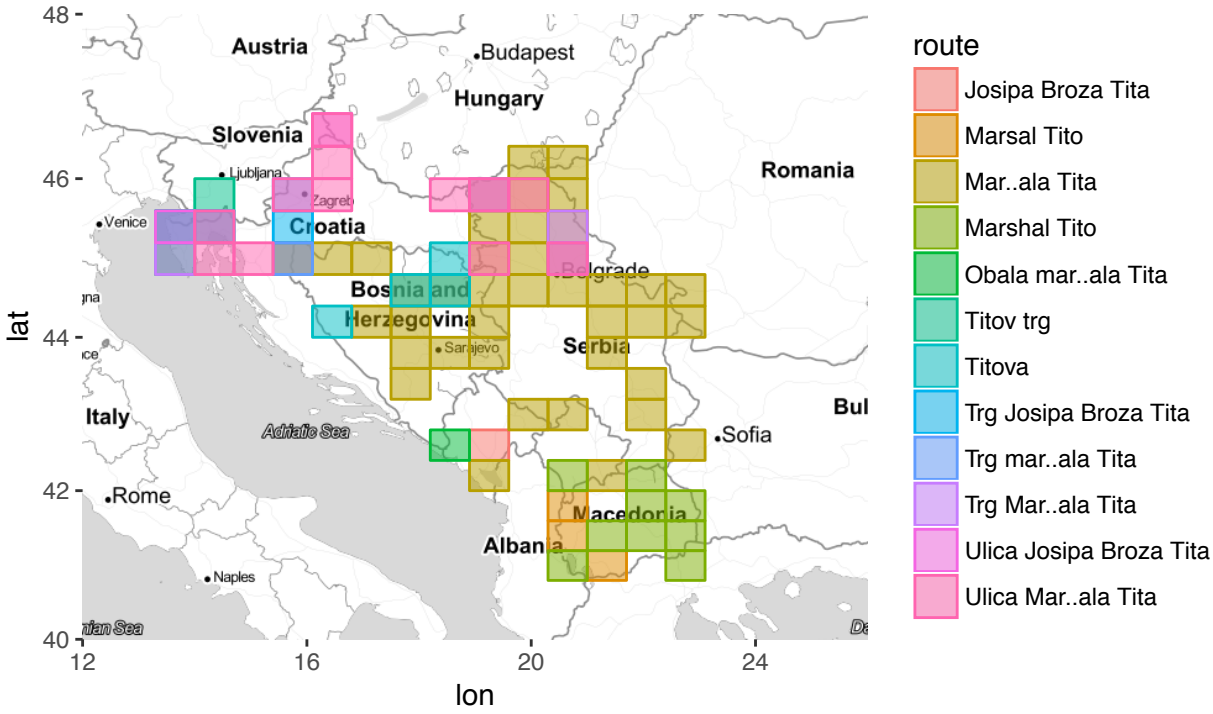



```
mapTonerLite + stat_bin2d(
  aes(x = lon, y = lat, fill = route, colour = route),
  size = .5, bins = 20, alpha = 1/2,
  data = TitoGmapsResults)
```



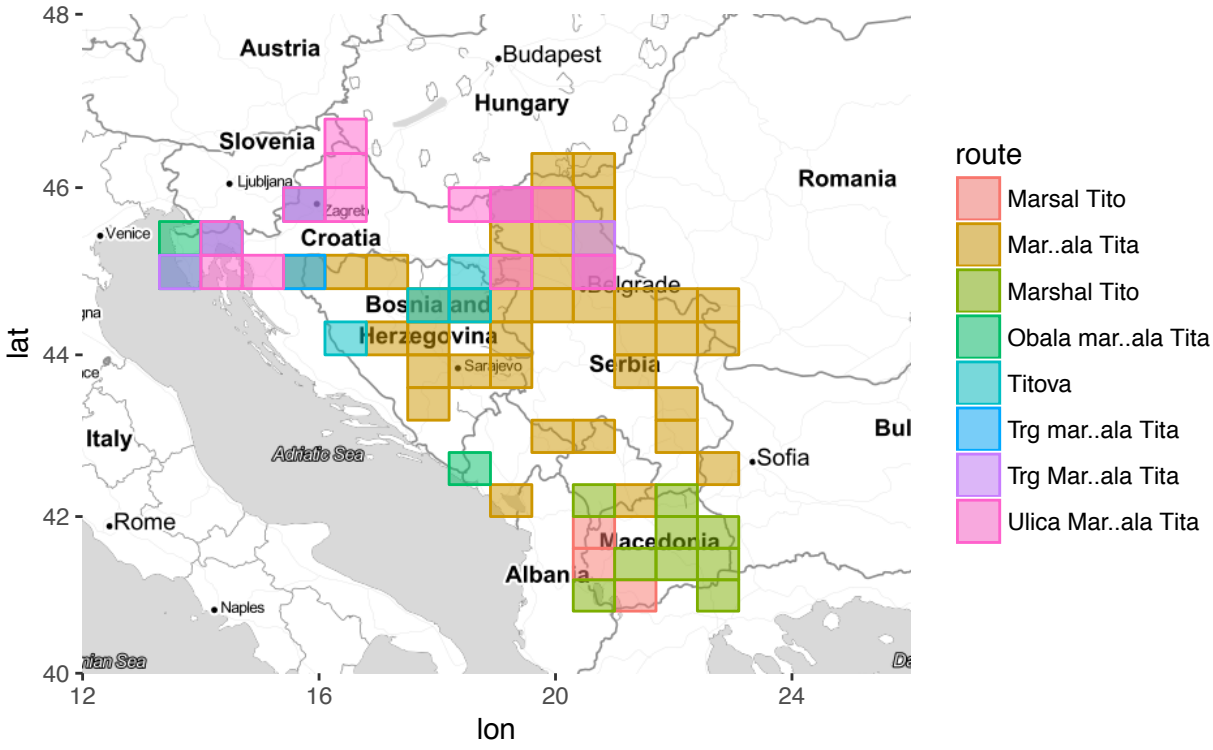
Keeping only street names found more than once.

```
mapTonerLite + stat_bin2d(
  aes(x = lon, y = lat, fill = route, colour = route),
  size = .5, bins = 20, alpha = 1/2,
  data = left_join(TitoGmapsResults, TitoGmapsResults %>% count(route)) %>% filter(n>1))
```

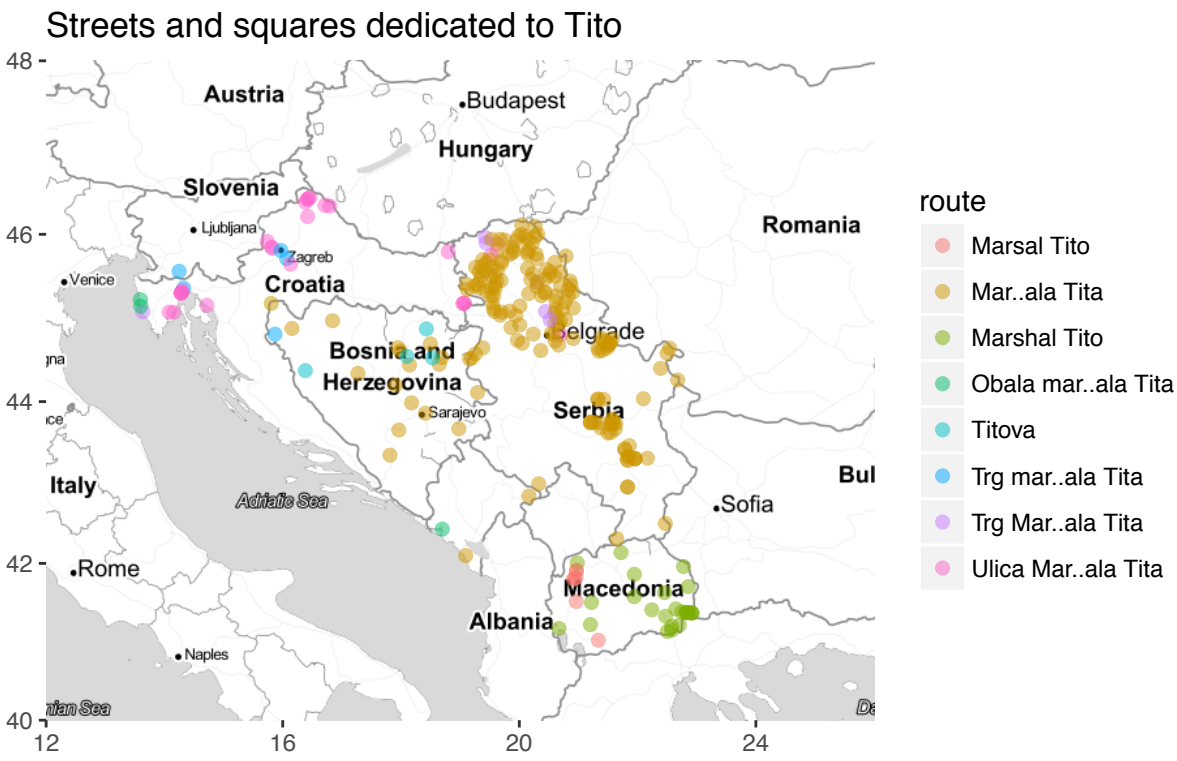


Or more than twice.

```
mapTonerLite + stat_bin2d(
  aes(x = lon, y = lat, fill = route, colour = route),
  size = .5, bins = 20, alpha = 1/2,
  data = left_join(TitoGmapsResults, TitoGmapsResults %>% count(route)) %>% filter(n>2))
```




```
mapTonerLite +
  geom_point(data=left_join(TitoGmapsResults, TitoGmapsResults %>% count(route)) %>%
               filter(n>2),
             aes(x=lon, y=lat, color = route), size=2, alpha=0.5) +
  labs(x = '', y = '') +
  labs(title = "Streets and squares dedicated to Tito")
```


Streets and squares dedicated to Tito
Austria
Budapest
Hungary
Slovenia
Ljubljana
Zagreb
Romania
Venice
Croatia
Belgrade
Bosnia and
Herzegovina
Sarajevo
Serbia
Italy
Adriatic Sea
Sofia
Rome
Macedonia
Albania
Naples
48
46
44
42
40
12
16
20
24
route
Marsal Tito
Mar..ala Tita
Marshal Tito
Obala mar..ala Tita
Titova
Trg mar..ala Tita
Trg Mar..ala Tita
Ulica Mar..ala Tita

Chapter 5

Summary of results

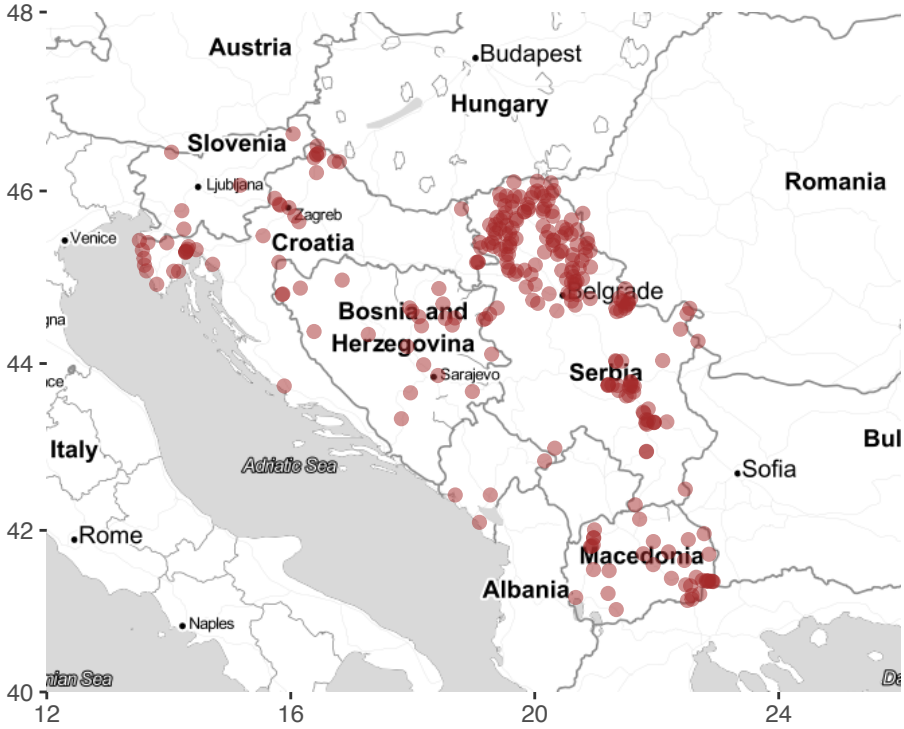
In the former Yugoslavia, streets and squares dedicated to Tito are commonplace in some areas (in particular, Istria and Vojvodina) but almost absent in others. The results presented below refer to data found by systematically querying the Google Maps API; OpenStreetMap data proved to be incomplete and partially misleading, as they would not have allowed to identify the area that seems to have the higher density of streets and squares dedicated to Tito, i.e. Vojvodina. Indeed, Vojvodina alone has more than a hundred instances of streets or squares dedicated to Tito.

Also data found via Google Maps may be incomplete, or not completely up to date. Indeed, as it appears also exploring the results on Google Maps there are some instances of streets dedicated to Tito found by OpenStreetMapa that are not found by Google Maps. For consistency, the maps below include only Google Maps data (as of August 2017): they likely offer a meaningful picture of the presence and distribution of streets and squares dedicated to Tito in the former Yugoslavia.

All results can be viewed as an overlay on Google Maps, or in the maps below.

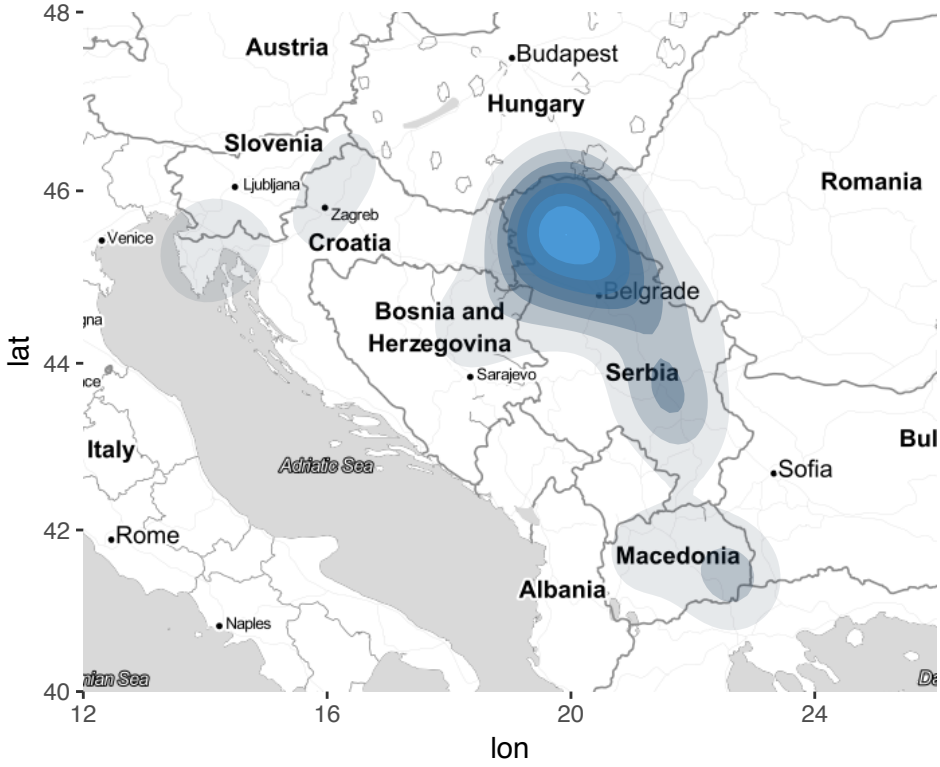
5.1 Tito streets on a map

Streets and squares dedicated to Tito in the former Yugoslavia

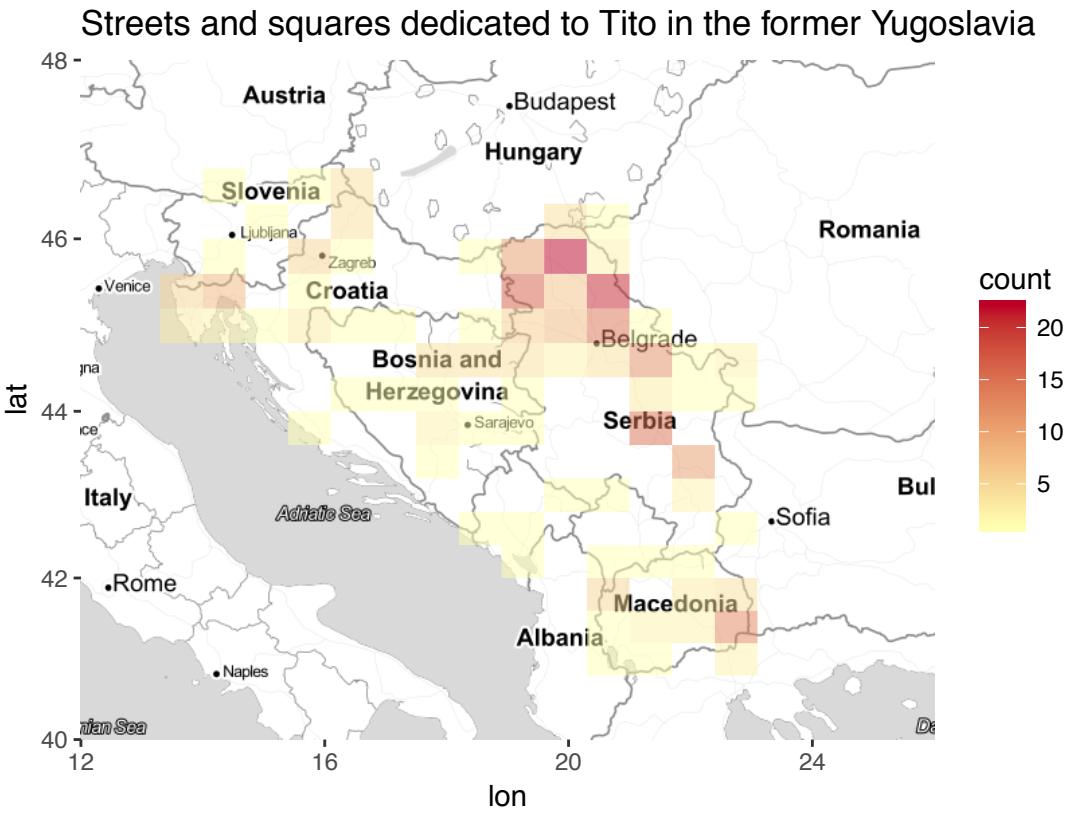


Source: Google Maps; https://giorgiocomai.eu/FindingTito

5.2 Density of Tito streets

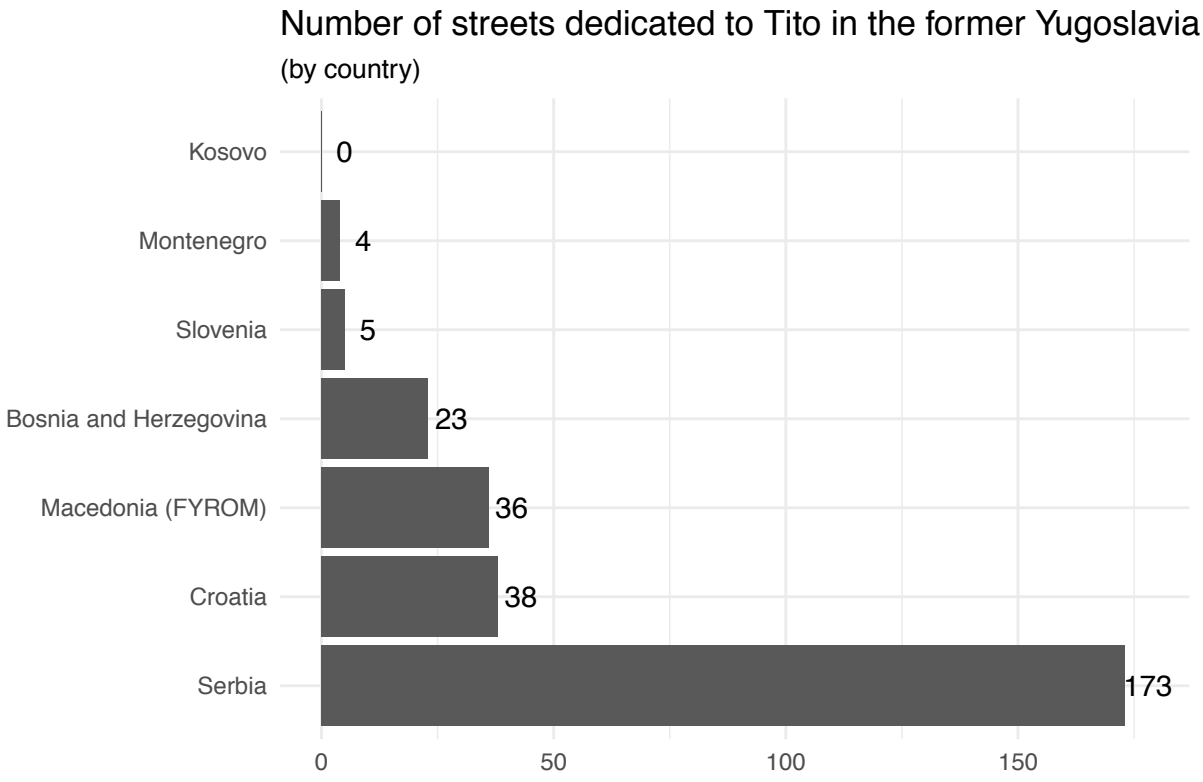


Source: Google Maps; https://giorgiocomai.eu/FindingTito



Source: Google Maps; https://giorgiocomai.eu/FindingTito

5.3 Tito streets by country



Source: Google Maps; https://giorgiocomai.eu/FindingTito