NSC-R workshop: open data

Samuel Langton, Netherlands Institute for the Study of Crime and Law Enforcement

14/09/2021

Key links

- GitHub (https://github.com/langtonhugh/opendata_nscr)
- Rijkswaterstaat (https://www.rijkswaterstaat.nl/apps/geoservices/geodata/dmc/nwb-wegen/geogegevens/shapefile)
- UK police data (https://data.police.uk/)
- Transport for London API (https://api.tfl.gov.uk/) and documentation (http://content.tfl.gov.uk/example-api-requests.pdf)
- CBS (https://www.cbs.nl/en-gb/onze-diensten/open-data/statline-as-open-data/quick-start-guide)
- Drimble (https://drimble.nl/112/amsterdam/index_p)
- Open Street Map book chapter (https://osf.io/a96y7)

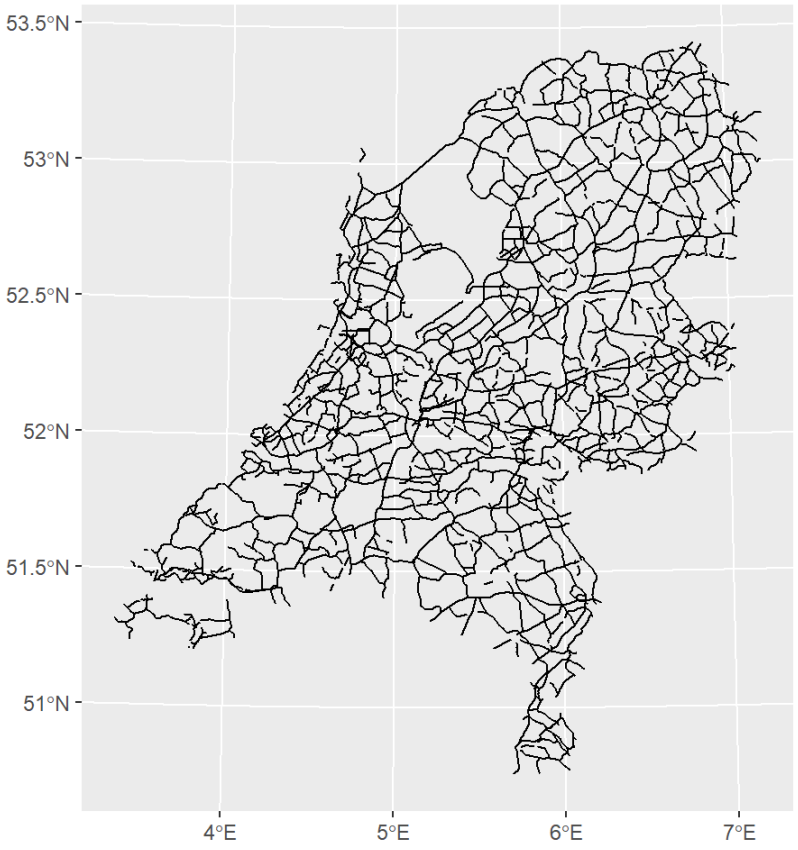
Setup

```
# Load packages.
library(readr)
library(dplyr)
library(janitor)
library(purrr)
library(haven)
library(forcats)
library(stringr)
library(sf)
library(ggplot2)
library(rvest)
library(jsonlite)
library(cbsodataR)
```

Direct download

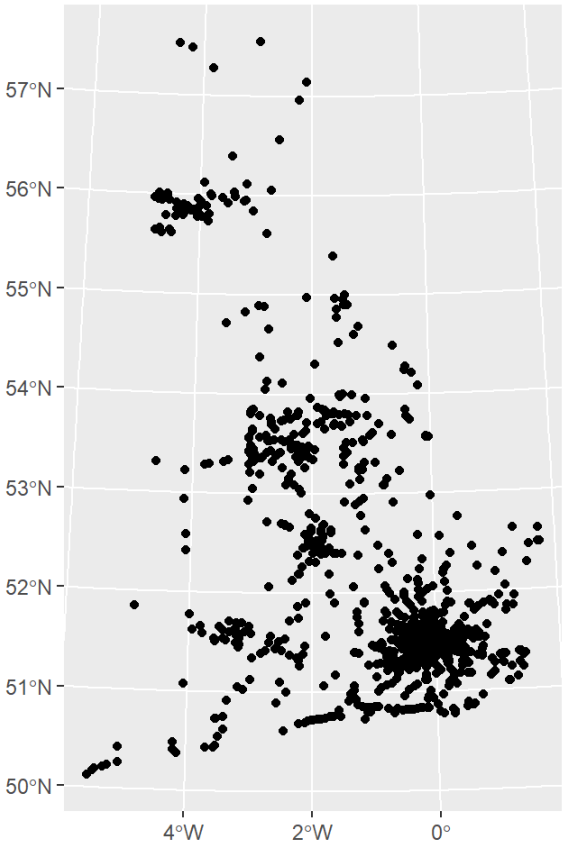
Download file

```
# Direct download.
download.file(url = "https://www.rijkswaterstaat.nl/apps/geoservices/geodata/dmc/nwb-wegen/geogegevens/
              shapefile/NWB-light/01-12-2020.zip",
              destfile = "data/nwn_light.zip")

# Unzip.
unzip(zipfile = "data/nwn_light.zip", exdir = "data/nwn_light")

# Load.
nwb_sf <- st_read("data/nwn_light/01-12-2020/NWB-Light/nwb-light.shp")

# Plot.
ggplot(data = nwb_sf) +
  geom_sf()
```



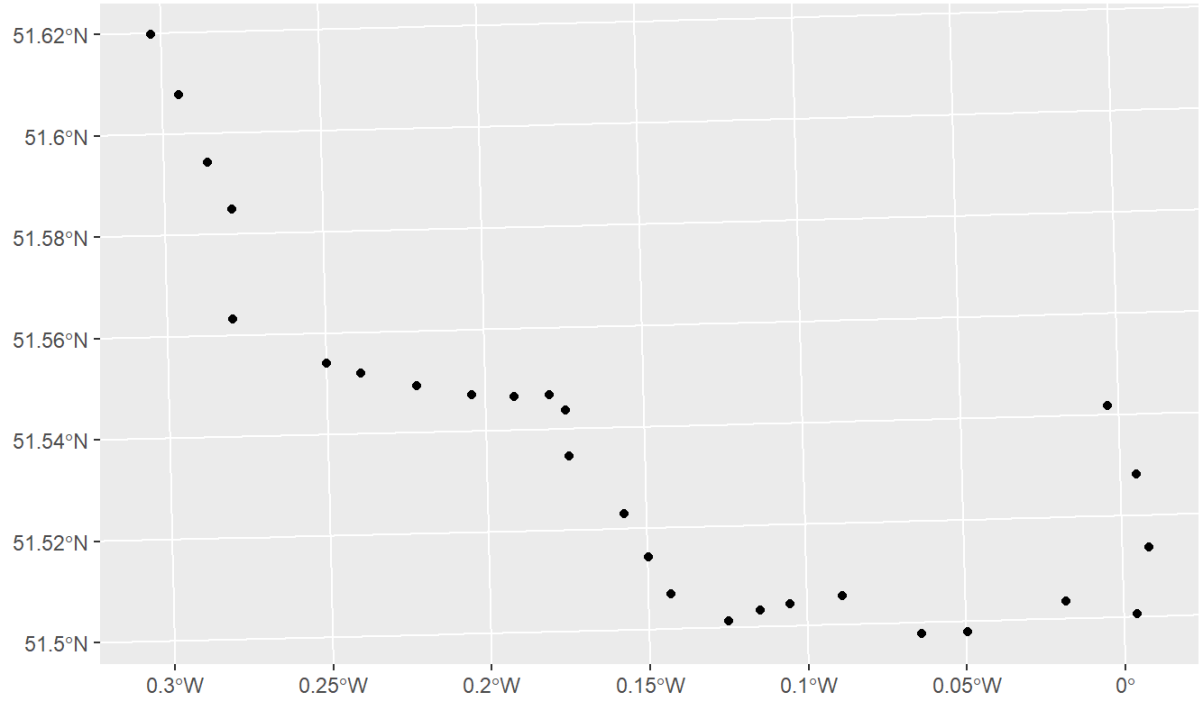
Repository

```
# CSV from Github.
btp_df <- read_csv("https://github.com/langtonhugh/osm_crim/raw/master/data/2020-01-btp-street.csv")

# Make spatial.
btp_sf <- btp_df %>%
  st_as_sf(coords = c(x = "Longitude", y = "Latitude"), crs = 4326) %>%
  st_transform(27700)

# Plot.
ggplot(data = btp_sf) +
  geom_sf()
```

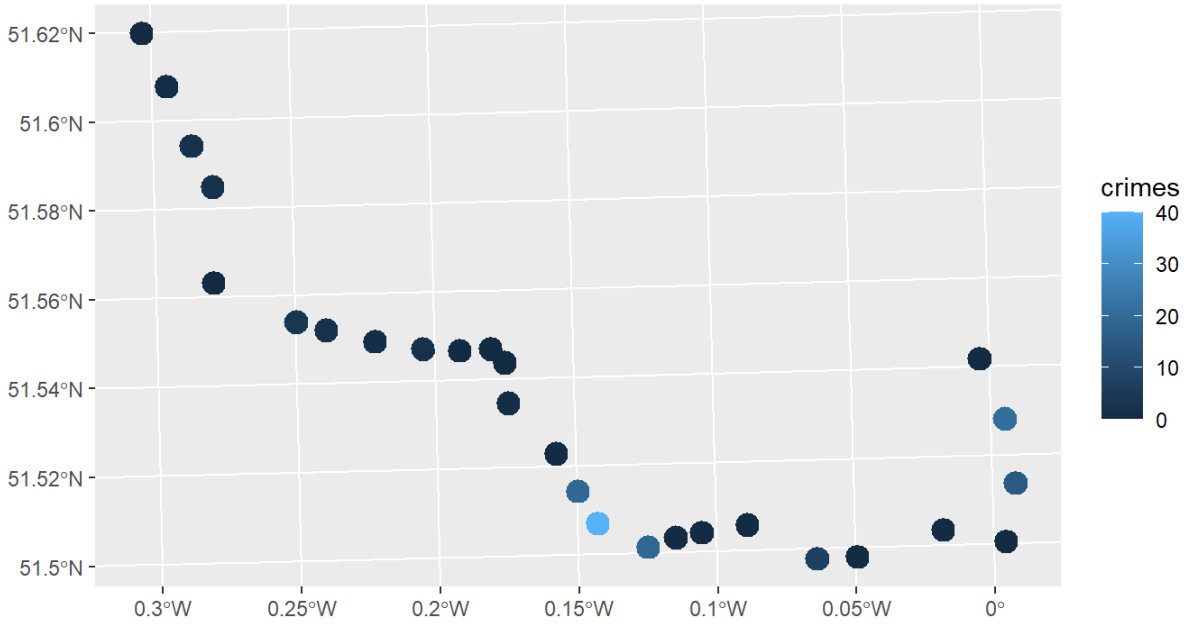


API

```
# Direct call to Transport for London API.
api_call <- fromJSON(readLines("https://api.tfl.gov.uk/line/jubilee/stoppoints"))

# Make spatial.
tfl_jub_sf <- api_call %>%
  select(commonName, lat, lon) %>%
  st_as_sf(coords = c(x = "lon", y = "lat"), crs = 4326) %>%
  st_transform(27700)

# Plot.
ggplot(data = tfl_jub_sf) +
  geom_sf()
```

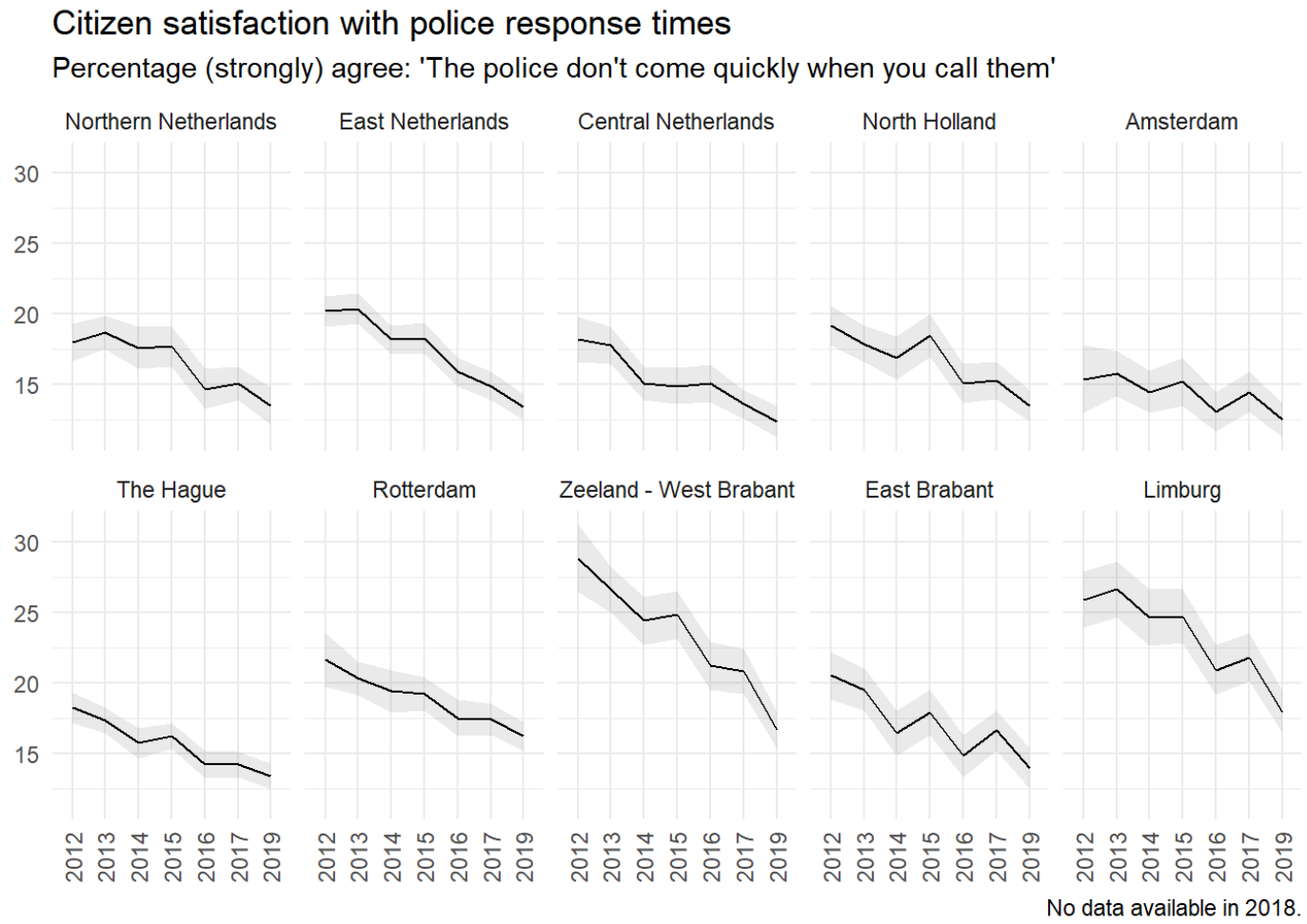



```
# Buffer.
tfl_buff_sf <- tfl_jub_sf %>%
  st_buffer(dist = 50)

# Aggregate.
tfl_jub_sf <- tfl_buff_sf %>%
  mutate(crimes = lengths(st_intersects(tfl_buff_sf, btp_sf)))

# Plot.
ggplot(data = tfl_jub_sf) +
  geom_sf(mapping = aes(colour = crimes), size = 4)
```


51.62°N
51.6°N
51.58°N
51.56°N
51.54°N
51.52°N
51.5°N
0.3°W
0.25°W
0.2°W
0.15°W
0.1°W
0.05°W
0°
crimes
40
30
20
10
0

API wrapper

```
# TOC for CBS open data package.
cbs_get_catalogs()

# Search.
politie_df <- cbs_search("Politie")

# Get one.
citpol_df <- cbs_get_data("81928NED", Periods = has_substring("JJ"))

# Get nationwide stats for satisfaction with response times (WIP!).
citpol_pd_df <- citpol_df %>%
  clean_names() %>%                                              # Clean var names
  zap_label() %>%                                                # Remove SPSS names
  mutate(regio_s = trimws(regio_s),                              # Remove whitespace
         pd      = str_detect(regio_s, "RE")) %>%                # ID regions
  filter(pd == TRUE) %>%                                         # Filter only regions
  rename(pol_quick_call = komt_niet_snel_als_je_ze_roept_71) %>%  # Rename var
  select(marges, regio_s, perioden, pol_quick_call) %>%          # Subset vars needed
  group_split(marges) %>%                                        # Split stats up
  bind_cols() %>%                                                # Stick back
  clean_names() %>%                                              # Clean var names
  rename(regio = regio_s_2,                                      # Renames
         perioden = perioden_3,
         pol_quick_call_est = pol_quick_call_8,
         pol_quick_call_ci = pol_quick_call_4) %>%
  select(regio, perioden, pol_quick_call_est, pol_quick_call_ci) %>% # Subset new vars
  mutate(perioden   = str_remove_all(perioden, "JJ00"),           # Keep only year
         regio_naam = fct_recode(regio,                           # Recode to names
                                 `Northern Netherlands`   = "RE01",
                                 `East Netherlands`       = "RE02",
                                 `Central Netherlands`    = "RE03",
                                 `North Holland`          = "RE04",
                                 `Amsterdam`              = "RE05",
                                 `The Hague`              = "RE06",
                                 `Rotterdam`              = "RE07",
                                 `Zeeland - West Brabant` = "RE08",
                                 `East Brabant`           = "RE09",
                                 `Limburg`                = "RE10"))

# Plot.
ggplot(data = citpol_pd_df,) +
  geom_ribbon(mapping = aes(x = perioden, group  = regio_naam,
                            ymax = pol_quick_call_est+pol_quick_call_ci,
                            ymin = pol_quick_call_est-pol_quick_call_ci),
              alpha = 0.1) +
  geom_line(mapping = aes(x = perioden, y = pol_quick_call_est, group  = regio_naam)) +
  facet_wrap(~regio_naam, nrow = 2) +
  labs(x = NULL,
       y = NULL,
       title = "Citizen satisfaction with police response times",
       subtitle = "Percentage (strongly) agree: 'The police don't come quickly when you call them'",
       caption = "No data available in 2018.") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



Scraping

```
# Scraping (P2000 via Drimble).
drimble_scrape <- read_html("https://drimble.nl/112/amsterdam/index_p") %>%   # read in page
  html_nodes(".content") %>%        # grab the content table
  html_table() %>%                  # convert to table
  pluck(1) %>%                      # grab relevant info
  row_to_names(row_number = 1) %>%  # create variable name
  clean_names()                     # clean variable names

# Show.
drimble_scrape
```

```
FALSE # A tibble: 23 x 3
FALSE    datum     tijd  x112_melding
FALSE    <chr>     <chr> <chr>
FALSE  1 14-09-20~ 11:43 Ambulance met hoge urgentie naar Bart de Ligtstraat, Amsterd~
FALSE  2 14-09-20~ 11:42 Ambulance met hoge urgentie naar Hogeweg, AmsterdamA1 13104 ~
FALSE  3 14-09-20~ 11:41 Begeleidend vervoer door een (zorg)ambulance naar Meibergdre~
FALSE  4 14-09-20~ 11:40 Ambulance met hoge urgentie naar Valeriusplein, AmsterdamA1 ~
FALSE  5 14-09-20~ 11:30 Ambulance met normale urgentie naar Kinderdijkstraat, Amster~
FALSE  6 14-09-20~ 11:23 Begeleidend vervoer door een (zorg)ambulance naar Wisseloord~
FALSE  7 14-09-20~ 11:19 Ambulance met hoge urgentie naar Buitenveldertselaan, Amster~
FALSE  8 14-09-20~ 11:05 Ambulance met normale urgentie naar Latexweg, AmsterdamA2 13~
FALSE  9 14-09-20~ 10:55 Ambulance met hoge urgentie naar Raphael Lemkinstraat, Amste~
FALSE 10 14-09-20~ 10:53 Ambulance met hoge urgentie naar Kabelweg, AmsterdamA1 13114~
FALSE # ... with 13 more rows
```