

LONG-TERM TEMPORAL TRENDS IN CRIMES AND THE INFLUENCE OF THE COVID PANDEMIC

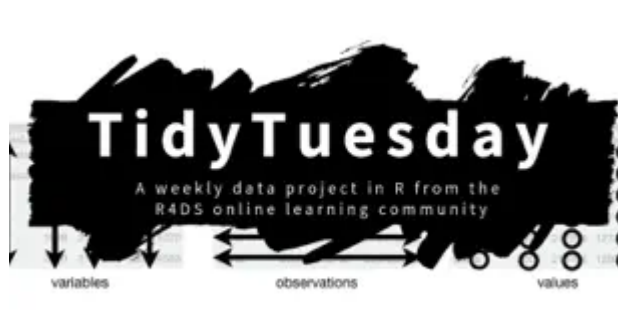


Figure 1: NSC-R Tidy Tuesday

Introduction

Below you find the materials Wim Bernasco prepared for the Tidy Tuesday workshop on May 24th 2022. [Here](#) is the original link to this workshop and the materials. You find links to two small datasets. One (`crime.csv`) contains the number of police recorded residential burglaries, bike thefts and assaults in the Netherlands per month between January 2012 and April 2022. The other (`population.csv`) contains the number of inhabitants of the Netherlands during the same period, per first day of the month. In the workshop he tried to make sense of long-term temporal trends in these crimes, and hopefully also make useful statements about how things changed when the COVID pandemic arrived around February 2020. He worked out eight steps in the analysis (Bernasco, 2022).

Start-up

Create a new folder on the computer where you run R, and store the two datafiles `crime.csv` and `residents.csv` in the folder.

You will normally load libraries here at the top of your script, but in this example I will load libraries at the point where I need their functionality. This shows when and why they are needed.

If your project is reasonably small and all files (data, scripts, and output files) can be stored in a single folder without creating confusion, setting a working folder is a good idea. All reading from and writing to disk will be done from and to this folder.

```
setwd("X:/YYY/ZZZ") # for example: setwd("C:/Users/bernasco/crime_trends_NL")
```

Note: In this script I will use the new `|>` piping symbol. It is equivalent to `%>%` but has two advantages, which are

- (1) it is shorter (2 instead of 3 symbols), and
- (2) it does not require you to load a library, as it has been built into base R

Read both files from disk and assign each a name.

Any name will do, but I suggest "crime" and "residents". Read data using the `read_csv` function in package `readr`. `read_csv` has a few advantages over `read.csv` but the latter will also do.

`read_csv` is in the `readr` library, so we load `readr` first. I added `tidyverse` also already.

```
library(readr)      # read_csv function
library(tidyverse)
```

Note: in the live workshop we stumbled upon an error caused by our assumption that "crimes.csv" and "population.csv" had dates coded with dashes (like in "2022-05-24") but

¹[Franziska Yasrebi-de Kom corrected errors and provided fruitful suggestions for improvement of the first draft.]

my Excel version had written it with slashes (like in "2022/05/24"). Verify that after reading the files, their 'date' column has type 'date', not type 'character'. You can use `glimpse` to verify this.

```
crimes <- read_csv("C:/FilesHarrie/HarrieGithub/customizing/customizing/pdf/crimes.csv")

population <- read_csv("C:/FilesHarrie/HarrieGithub/customizing/customizing/pdf/population.csv")
```

Explore data

Explore the structure of the files. How many rows (observations), how many columns (variables), what are the variable names? What are their types: Are they integers, characters, dates, factors, .? keys? First explore the crime data. How many observations and how many variables (rows and columns)?

```
crimes |> dim()          # dim(crimes)
```

```
[1] 372  3
```

For `glimpse`, `slice-sample` and `count`, we need library `dplyr` library(`dplyr`). Display the names and types of the variables, and show values for the first few observations column-wise. Note: verify that column date has type 'date'

```
crimes |> glimpse()
```

```
Rows: 372
Columns: 3
$ date      <date> 2012-01-01, 2012-02-01, 2012-03-01, 2012-04-01, 2012-05-01,~
$ frequency <dbl> 9787, 8237, 7944, 5955, 6198, 5493, 6168, 6481, 6102, 8178,~
$ crime_type <chr> "burglary", "burglary", "burglary", "burglary", "burglary",~
```

Show the first few cases row-wise

```
crimes |> head()
```

```
# A tibble: 6 x 3
  date      frequency crime_type
  <date>      <dbl> <chr>
1 2012-01-01     9787 burglary
2 2012-02-01     8237 burglary
3 2012-03-01     7944 burglary
4 2012-04-01     5955 burglary
5 2012-05-01     6198 burglary
6 2012-06-01     5493 burglary
```

Show a random sample of 10 cases rather than the first ones. This will usually have more variation in values

```
crimes |> slice_sample(n=10) |> head()
```

```
# A tibble: 6 x 3
  date      frequency crime_type
<date>      <dbl> <chr>
1 2013-11-01    8496 burglary
2 2017-11-01    7126 bike theft
3 2013-03-01    8912 bike theft
4 2017-08-01    3828 assault
5 2013-12-01    9832 burglary
6 2021-05-01    3080 assault
```

Frequency table of crime types

```
crimes |> count(crime_type)
```

```
# A tibble: 3 x 2
  crime_type      n
  <chr>      <int>
1 assault    124
2 bike theft 124
3 burglary   124
```

You will see we have data for 124 months: 10 full years (2012-2021) + 4 months (Jan-April 2022)

Next explore the population data

```
population |> dim()
```

```
[1] 123  2
```

Note: verify that column date has type 'date'.

```
population |> glimpse()
```

```
Rows: 123
Columns: 2
$ date      <date> 2012-01-01, 2012-02-01, 2012-03-01, 2012-04-01, 2012-05-01~
$ population <dbl> 16730348, 16731280, 16735690, 16737631, 16739764, 16742830,~
```

```
population |> head()
```

```
# A tibble: 6 x 2
  date      population
  <date>      <dbl>
1 2012-01-01 16730348
2 2012-02-01 16731280
```

```
3 2012-03-01 16735690
4 2012-04-01 16737631
5 2012-05-01 16739764
6 2012-06-01 16742830
```

```
population |> slice_sample(n=10) |> head()
```

```
# A tibble: 6 x 2
  date      population
<date>      <dbl>
1 2012-05-01 16739764
2 2018-01-01 17181084
3 2014-07-01 16857811
4 2022-03-01 17609246
5 2016-12-01 17079787
6 2015-05-01 16913226
```

Visualize (plot) population and crime development

Visualize (plot) the development of the population size of the Netherlands between January 2012 and April 2022, and do the same for the frequencies of residential burglary, bike theft and assault. The ggplot function is in the ggplot2 library.

```
library(ggplot2)
```

Plot population as a scatterplot.

```
population |>
  ggplot() +
  geom_point(mapping = aes(x = date, y = population))
```

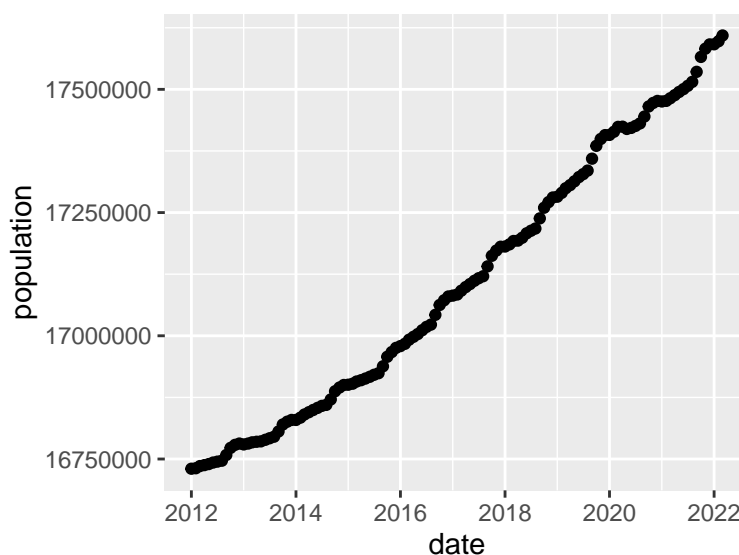


Figure 2: Population grow

```
# or as a line graph
population |>
  ggplot() +
  geom_line(mapping = aes(x = date, y = population))
```

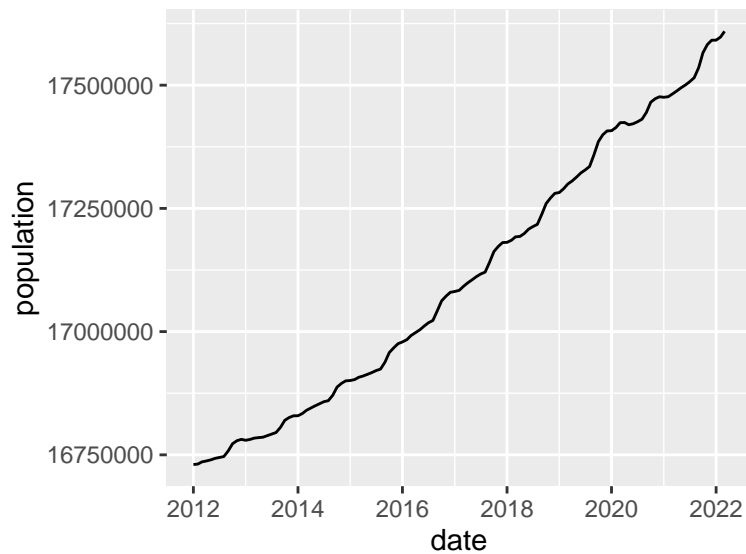


Figure 3: Population grow

Keep it simple first. You can finetune later. Plot burglary frequencies

```
crimes %>%
  filter(crime_type == "burglary") |>
  ggplot() +
  geom_line(mapping = aes(x=date, y=frequency))
```

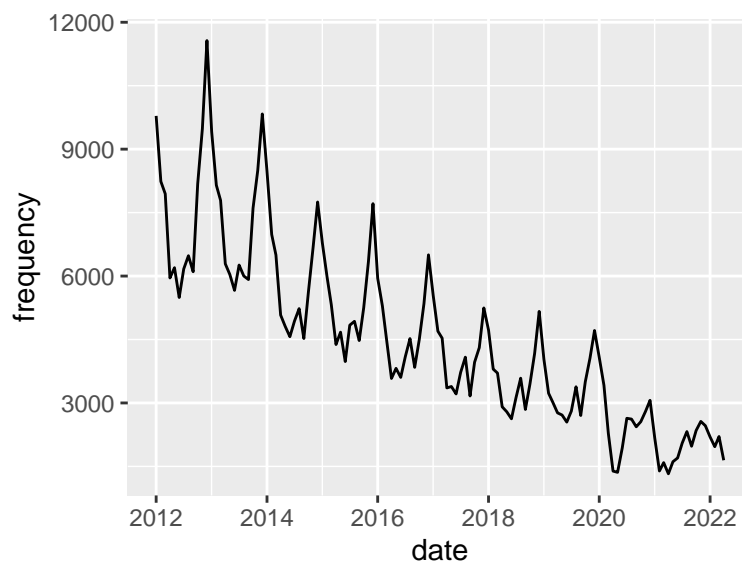


Figure 4: Burglary frequencies

You should see a seasonal pattern with highs in Winter (nov-jan) and lows in summer. Plot bike theft

```
crimes %>%
  filter(crime_type == "bike theft") |>
  ggplot() +
  geom_line(mapping = aes(x = date, y = frequency))
```

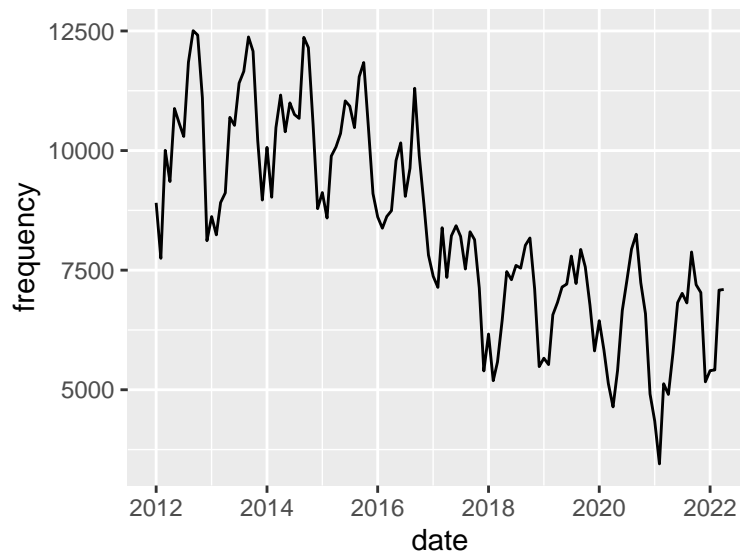


Figure 5: Seasonal pattern

You should also see a seasonal pattern, but with peaks in autumn (sep-nov)

Plot assault

```
crimes %>%
  filter(crime_type == "assault") |>
  ggplot() +
  geom_line(mapping = aes(x = date, y = frequency))
```

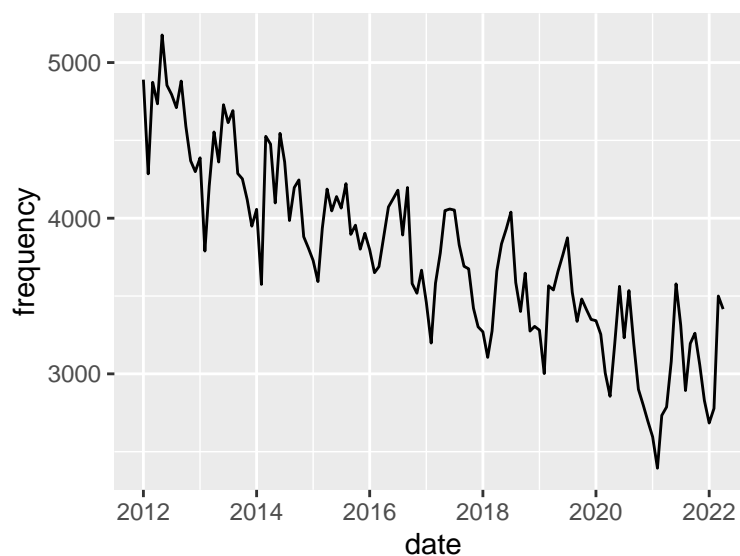


Figure 6: Assault

The seasonal pattern for assault shows peaks in spring (apr-jun). Plot the three crime types

in a single graph and distinguish them by color.

```
crimes |>
  ggplot() +
  geom_line(aes(x = date, y = frequency, color = crime_type))
```

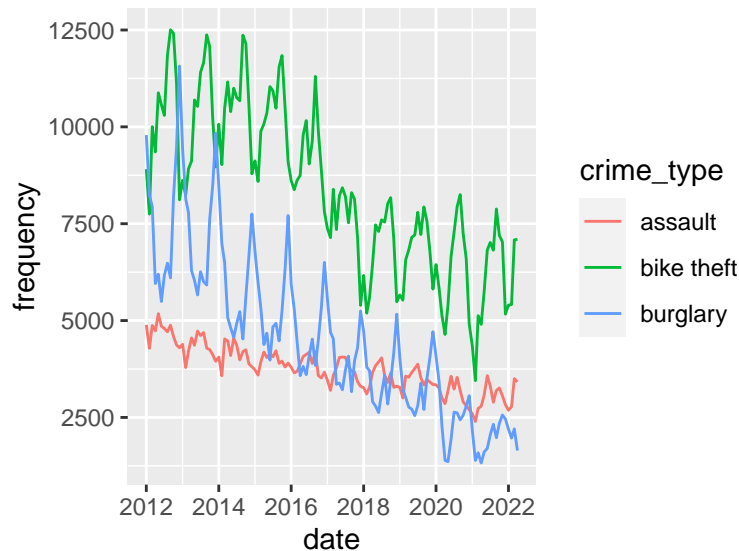


Figure 7: Three crime types

Plot them in separate panels.

```
crimes |>
  ggplot() +
  geom_line(aes(x = date, y = frequency)) +
  # scales = "free_y" allows each panel to have its own
  # Y-axis scaling
  facet_wrap(facets = "crime_type", scales = "free_y", ncol = 1)
```

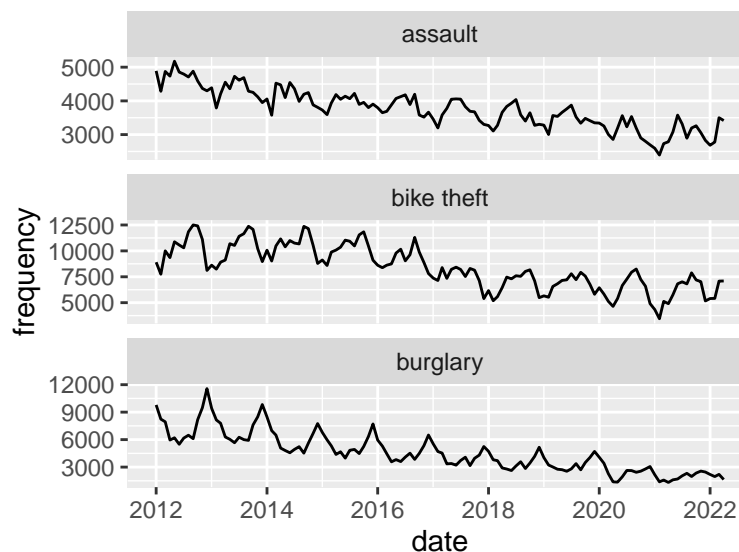


Figure 8: Seperate crime panels

I would not advise it, but you can combine panels and color (if you do, hide the legend to prevent redundancy)

```
crimes |>
  ggplot() +
  geom_line(aes(x = date, y = frequency, color = crime_type)) +
  facet_wrap(facets = "crime_type", scales = "free_y", ncol = 1) +
  theme(legend.position="none")
```

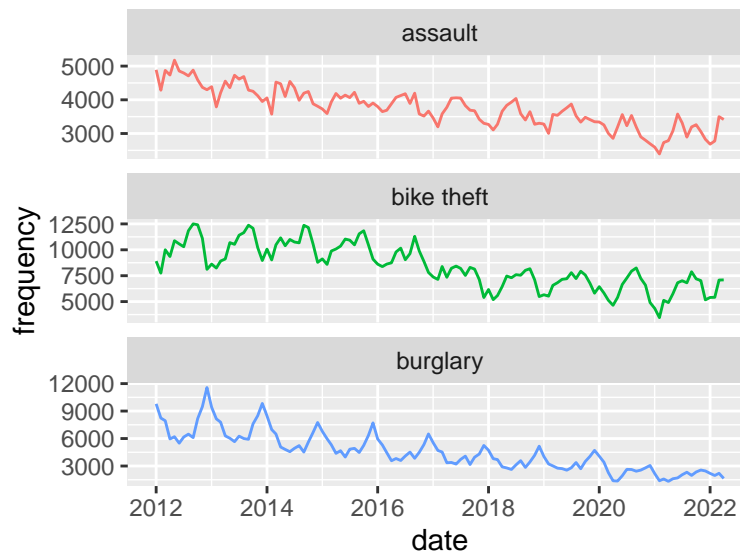


Figure 9: Crime types in color

Do anything else with the data that you find fun or seems

To make sense You can see that there is an overall decline for each crime type (mean_change is always negative) and the decline is more pronounced (mean_change is more negative) during the pandemic, in particular for assault and burglary.

For more rigorous statistical tests, we would need to dive into time series analysis.

References

Bernasco, W. (2022, May). NSC-R Workshops: NSC-R Tidy Tuesday. NSCR. Retrieved from <https://nscrweb.netlify.app/posts/2022-05-24-nsc-r-tidy-tuesday/>