API-231 / GIS-PubPol
Meeting 20 ("Russian-Ukrainian War" Walk Through)

Yuri M. Zhukov
Visiting Associate Professor of Public Policy
Harvard Kennedy School

April 11, 2024

Figure 1: News = data about who did what to whom, when and where

Who	Did what	To whom	When	Where	Source
Russia	rocket strike	Ukraine	2/24/2022	Kyiv	CNN
Russia	rocket strike	Ukraine	2/24/2022	Kharkiv	CNN

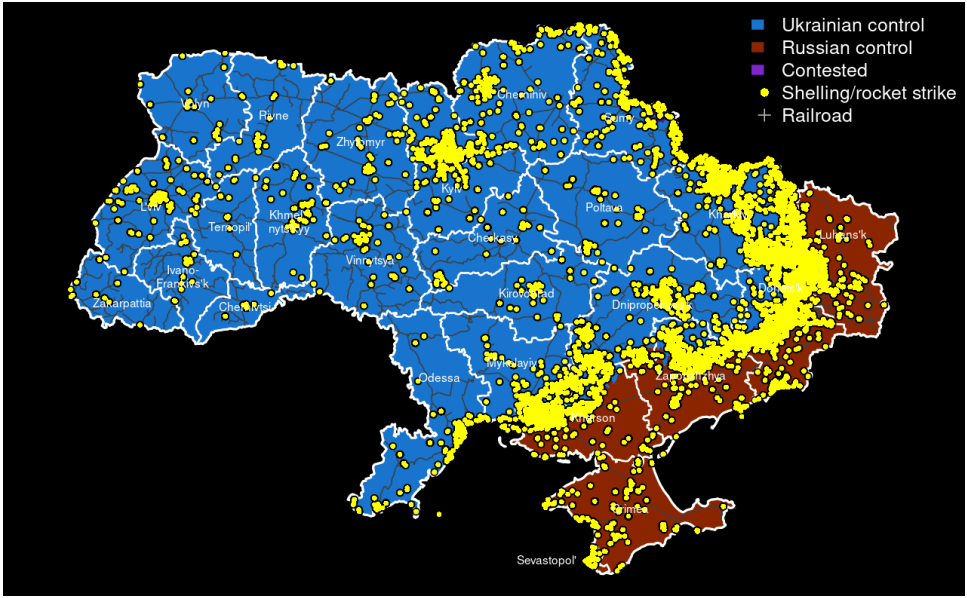


Figure 2: Near-real time event, territorial control data on Russian invasion of Ukraine

Overview

Motivation: during Russia's 2022 invasion of Ukraine...

1. Russia has become a hermetically-sealed information environment
   a) media required to stick to MoD press releases
   b) cannot use word "war" when describing "special military operation"
   c) up to 100K ruble fine for publicly "discrediting" Russian army
   d) up to 15-year sentence for "knowingly false information" about war
   e) last independent media shut down (e.g. TV Rain, Echo of Moscow)
   f) Facebook, Twitter, Instagram, VPNs blocked
2. Ukrainian media more free, but vulnerable
   a) TV news sometimes broadcasts from basements, bomb shelters
   b) Russia has targeted TV towers, cut electricity, cell service
   c) all national TV channels merged onto one platform under martial law
   d) radio silence on Ukrainian casualties, ongoing operations

Solution: use machine learning, remote sensing to track events on the ground

Tracking the War in Near-Real Time

What are "event data"?

1. Incident-level data on "who did what to whom, when and where"
   a) "who": initiator of action (subject)
   b) "did what": description of action/tactic (verb)
   c) "to whom": target of action (object)
   d) "when": time/date of event
   e) "where": location of event
2. Types of events (examples we have used in this class)
   a) political violence
   b) bike crashes in NYC
   c) crimes in DC
   d) 311 calls about flooding in New Orleans
3. Sources of data
   a) media/open sources (including social media)
   b) government records/archives
   c) remote sensing

VIINA: Violent Incident Information from News Articles

1. Near-real time event data on Russian invasion of Ukraine (updated daily)
   a) based on news reports from Ukrainian and Russian media, geocoded and classified with Bidirectional Encoder Representations from Transformers (BERT)
   b) each event is accompanied by full source info, text and URLs
2. Data on territorial control at municipality level (updated daily)
   a) based on vectorized georeferenced maps (e.g. DeepState, Wikipedia)
   b) "boosted" by VIINA event data on changes in control

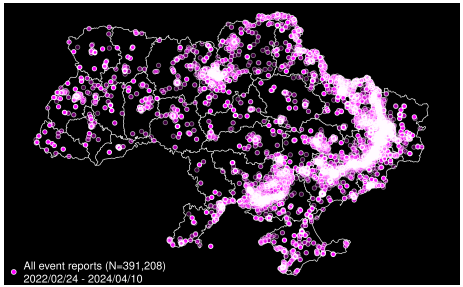


Figure 3: Events

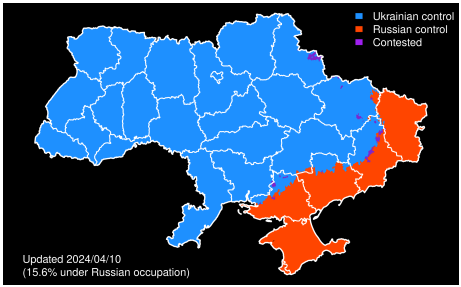


Figure 4: Control

Problem: 1 news report $\neq$ 1 unique event

1. We can characterize each event as a unique configuration of
   `[subject]-[verb]-[object]-[time]-[location]`
   (i.e. who did what to whom, when and where)
2. We can learn about these events from news reports
   a) one report of one event ("A attacked B")
   b) multiple reports of one event ("A attacked B", "B attacked by A")
   c) multiple events in one report ("A attacked B, C attacked A")
   $\vdots$
   d) N reports of an unknown $\#$ of events

Question: do these news reports refer to the same event?

Date	Source	English translation
2022/02/25	Interfax.ua	Russian forces purposefully shelling residential buildings in Kharkiv – head of regional administration Synyehubov
2022/02/25	24tv.ua	Shell hits residential building in Kharkiv: casualties possible – frightening photos

Who	Did what	To whom	When	Where
	✓	✓	✓	✓

Question: do these news reports refer to the same event?

Date	Source	English translation
2022/08/10	liveuamap	In Kharkiv as a result of Russian shelling one person is wounded
2022/08/15	24tv.ua	Woman, wounded during shelling of Saltivka in Kharkiv, died in hospital

Who	Did what	To whom	When	Where
	✓			✓

Coreference resolution (CR)

Process of resolving multiple references to same physical object or event

1. Why are duplicates a problem?
   a) duplicates are a threat to causal inference
   b) over-reporting of events may be correlated with unobservables (e.g. media presence, perceived "newsworthiness" or novelty)
   c) duplicates make it harder to assess ground truth about violence
   d) this problem affects both machine-coded and hand-coded data
2. Applications to event data:
   a) remove exact textual duplicates ("bare minimum")
   b) "1 per day" filter (if two reported events are of the same type, and were reported in same location on same day, then they are references to the same event)
   c) MELTT spatio-temporal filter (match based on co-occurrence in space and time)
   d) model-based methods (e.g. convolutional neural networks, transformers)

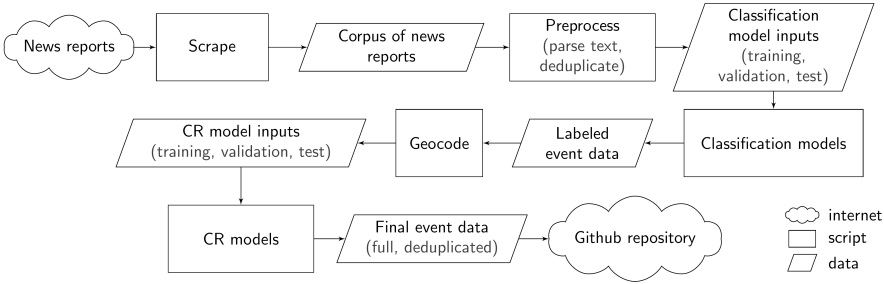


Figure 5: VIINA workflow

VIINA turns online news articles & social media into geocoded event data by:

1. Scraping online news & social media, preprocessing the raw text
2. Classifying the news reports by actor and tactics with large language models
3. Assigning geo coordinates based on locations mentioned in reports
4. Identifying (but not removing) likely duplicate events

Other Near-Real Time Data Sources

Historical weather & climate raster data (partial list)

Dataset	ACLED	GDELT	ICEWS	VIINA
Data on Violence?	✓	✓	✓	✓
Data on Control?				✓
Fully Automated?		✓	✓	✓
Text Descriptions?	✓			✓
Source URLs?		✓		✓
Events in 1st Year	40,448	778,350	27,858	113,446
Unique Event Locations	2,430	1,762	581	9,771
Update Frequency	1 week	Daily	1-2 months	Daily
Event Types	24	50	105	23
Sources	97	8,887	126	30
English-Only?	No	Yes	Yes	No
Ukrainian Sources (%)	74.2	10.1	4.2	92.5
Russian Sources (%)	14	11.8	12	7.5
Unknown Sources (%)	0	0	47.3	0

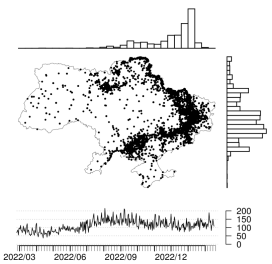


Figure 6: ACLED

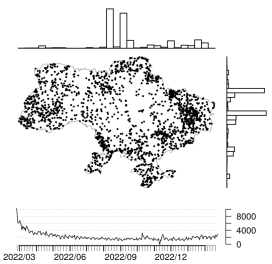


Figure 7: GDELT

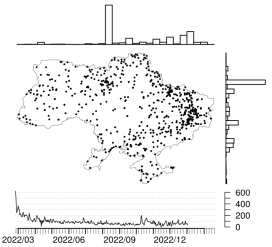


Figure 8: ICEWS

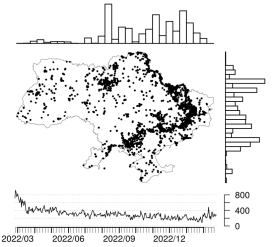


Figure 9: VIINA

Near-real time remote sensing data (partial list)

Type	**Source/link**	**Spatial resolution**	**Frequency**	**Free?**
Fire anomalies	`FIRMS`	Points	Daily	✓
Night lights	`VIIRS`	Raster	Nightly	✓
Vegetation	`NDVI`	Raster	2 weeks	✓
Meteorological events	`NASA Worldview`	Raster	Daily	✓
Reflectance (photos)	`NASA Worldview`	Raster	Hourly/Daily	✓

What kinds of events can remote sensing capture that media cannot? (+ vice versa)

Vignettes

Overview of lab exercise

1. How much of Ukraine's territory does Russia occupy?
2. Compare media reports to remote sensing data on fire anomalies.

We will work with a (very large) dataset on territorial control in Ukraine

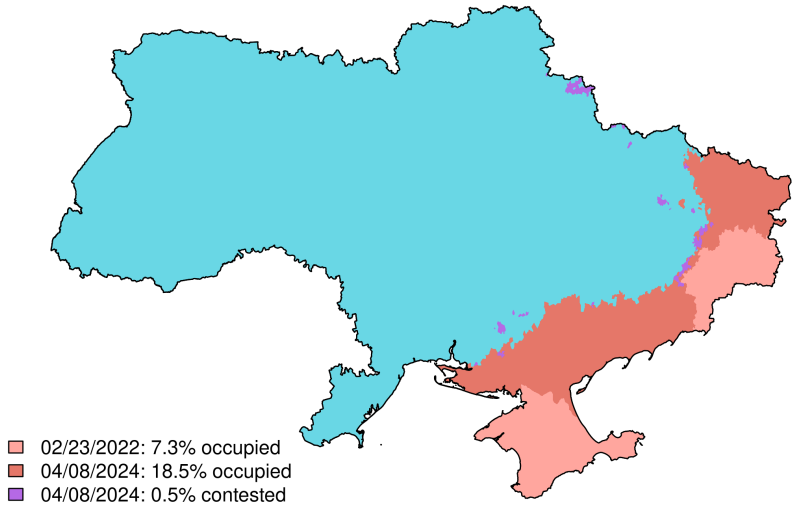


Figure 10: Vignette 1

We will then integrate NASA's data on active fires with VIINA event reports

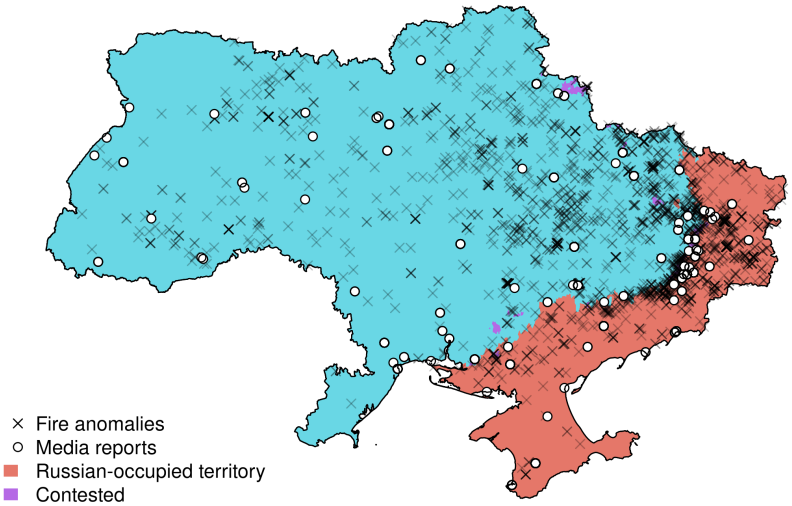


Figure 11: Vignette 2 / Step 1

. . . and identify locations that may be overlooked media vs. fires data

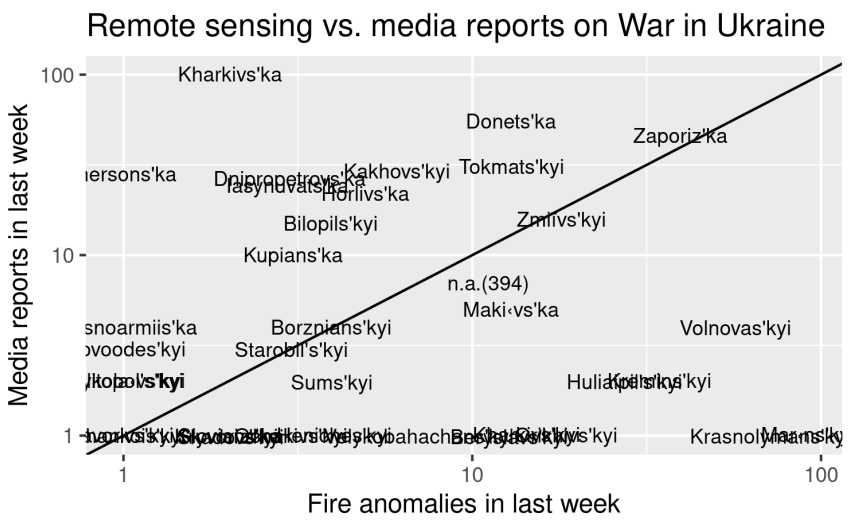
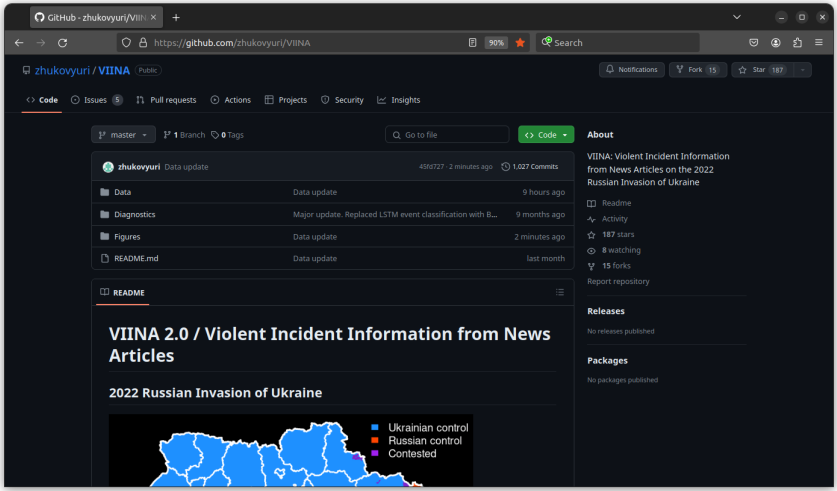
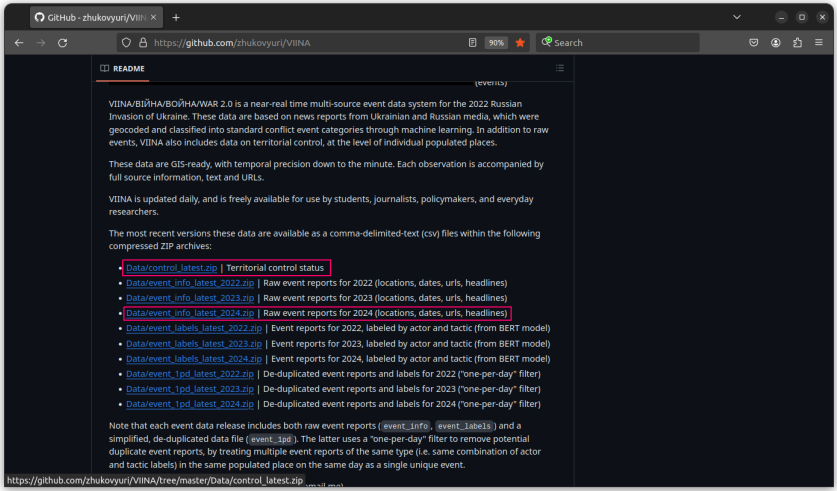


Figure 12: Vignette 2 / Step 2

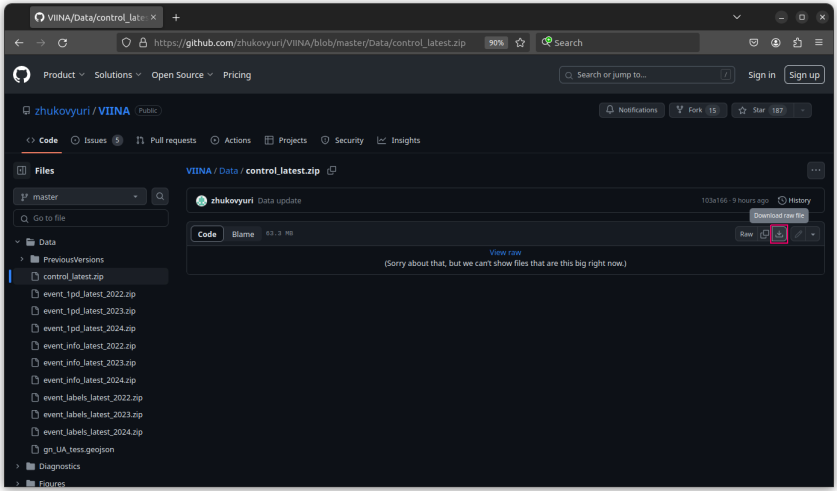
We can obtain **data on territorial control and media reports** from `github.com/zhukovyuri/VIINA`



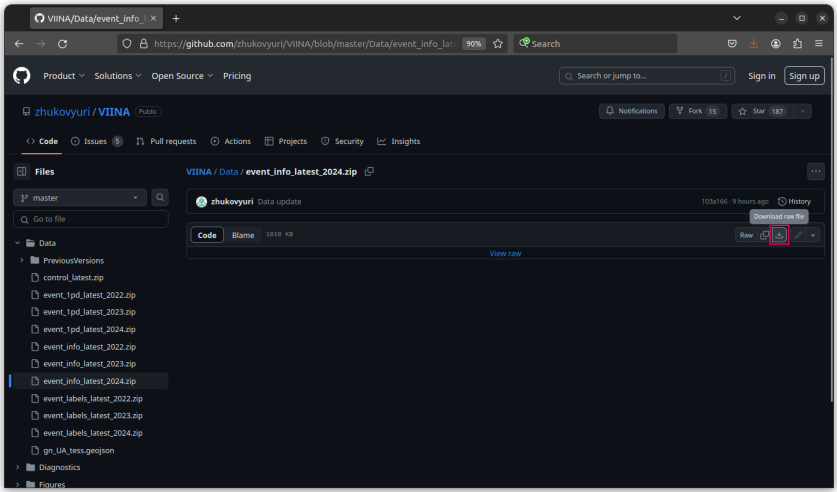
There are several datasets here. The ones we need are `control_latest` and `event_info_latest_2024`



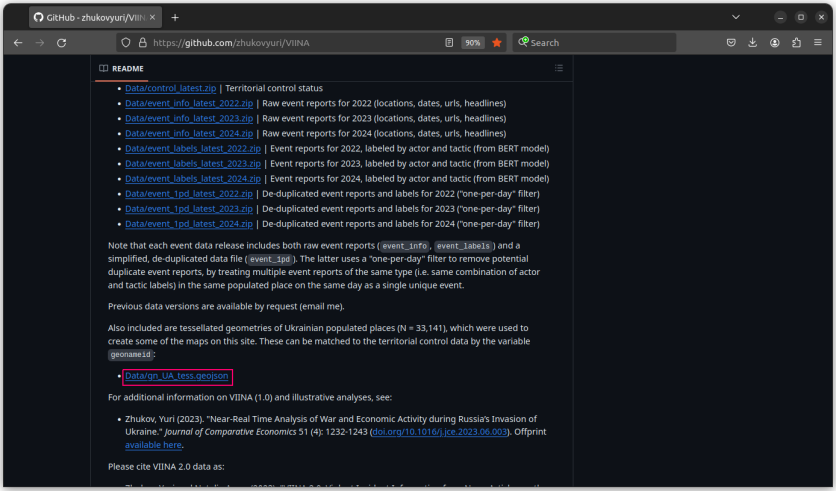
Go to `control_latest.zip` and download the file by clicking on the "Download raw file" button



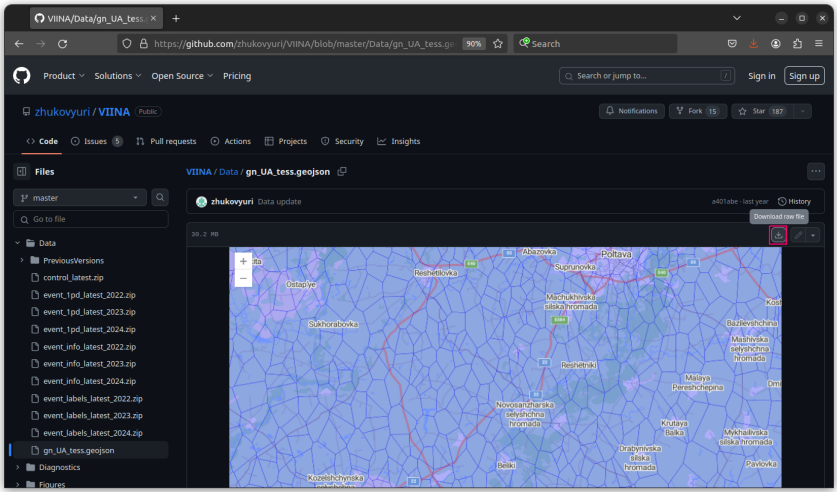
Do the same thing for `event_info_latest_2024.zip`



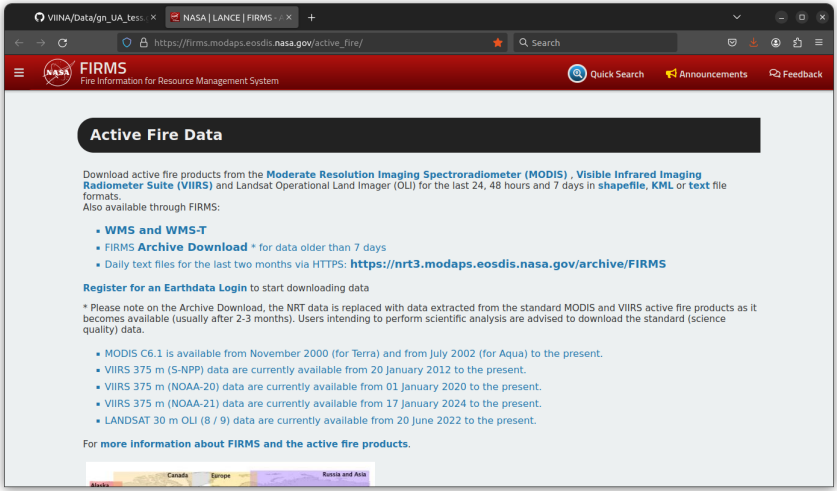
While we're here, let's also grab the GIS boundaries for Ukrainian populated places, `gn_UA_tess.geojson`



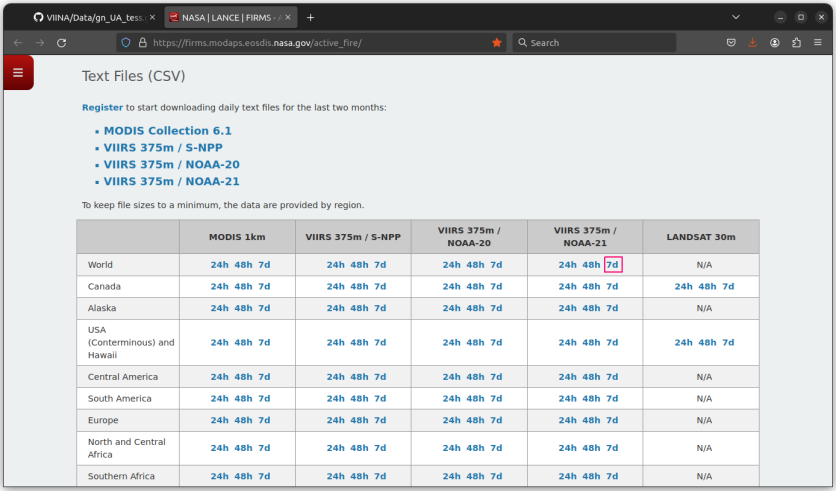
We can also get this file through the "Download raw file" link



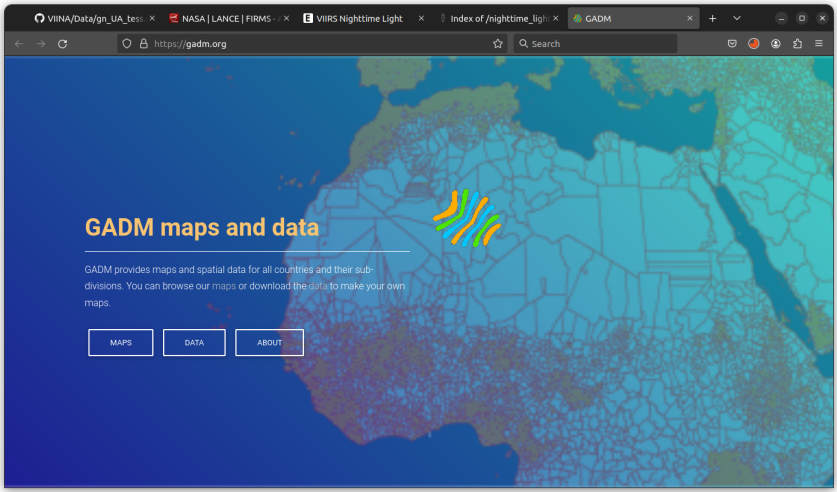
Let's now get the **FIRMS Active Fires data** from `firms.modaps.eosdis.nasa.gov/active_fire/`



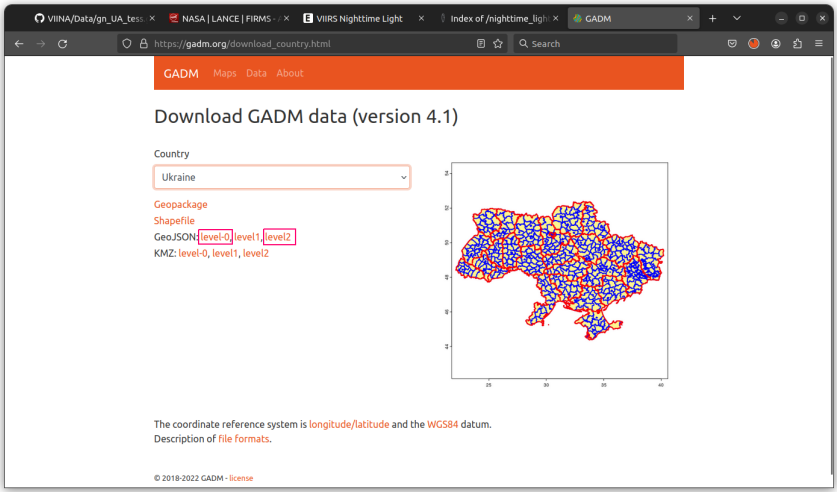
Scroll down to "Text Files (CSV)" and download the latest weekly (7d) data for "World" from "VIIRS 375m/NOAA-21"



We will use **country-level and district-level boundaries** data from `gadm.org`



Download the `level-0` and `level-2` files for Ukraine, in `GeoJSON` format



Here is the full list of data sources and links:

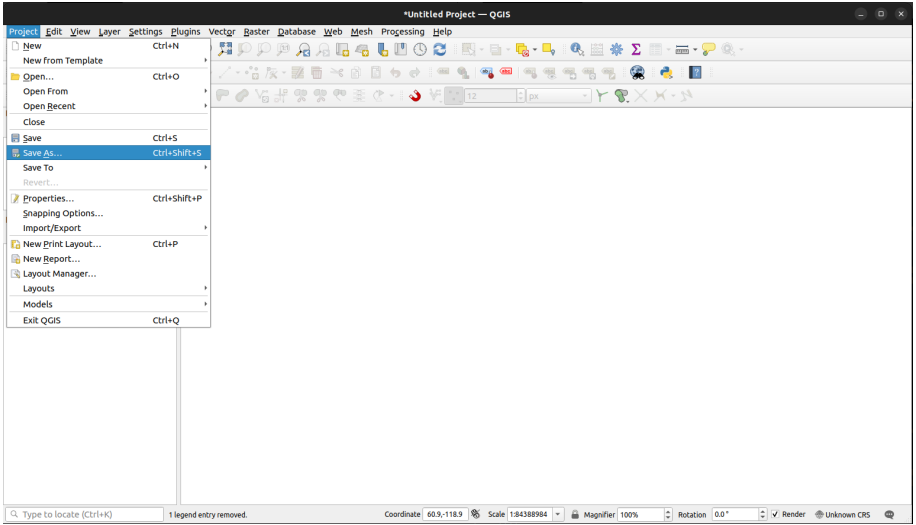
Category	Type	Format	Data source
Territorial control	Vector (polygons)	`.csv`, `.geojson`	VIINA
Media event reports	Vector (points)	`.csv`	VIINA
Active fires	Vector (points)	`.csv`	NASA FIRMS
Administrative borders	Vector (polygons)	`.geojson`	GADM

These are all in the `WT04.zip` file posted on Canvas.

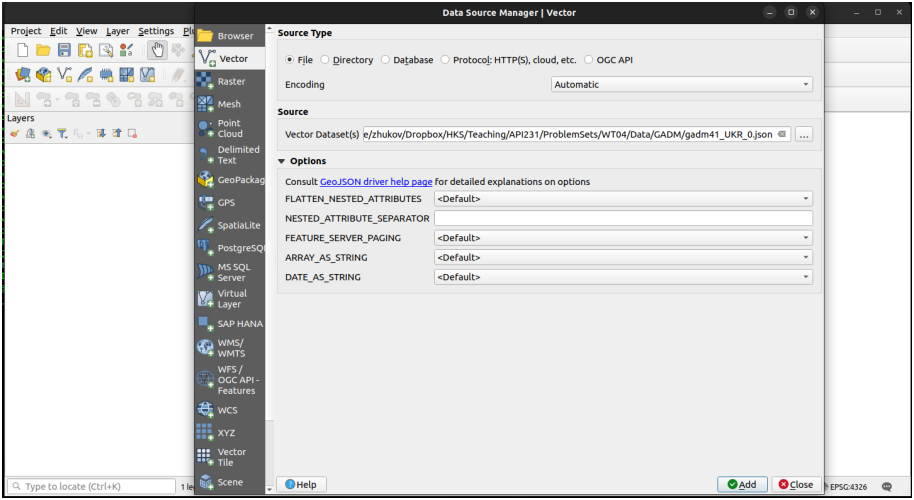
How much of Ukraine's territory does Russia occupy?

Always save your progress!

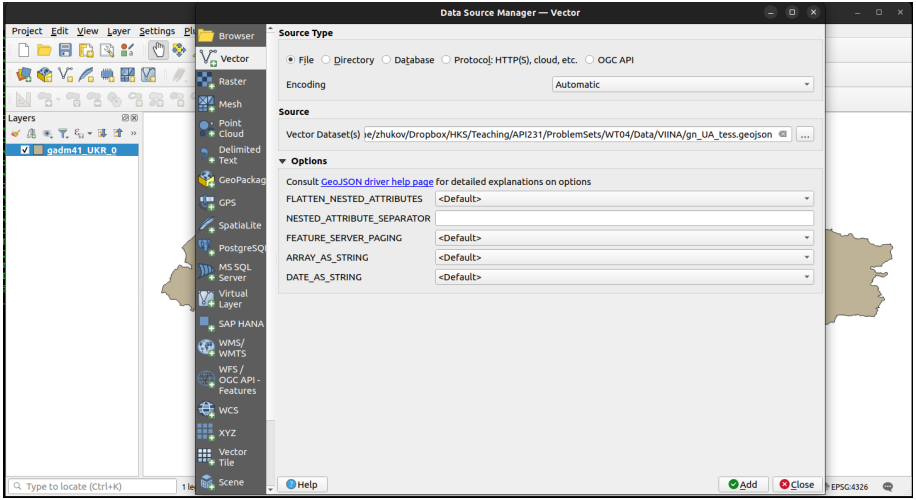
Go to `Project` → `Save As...`



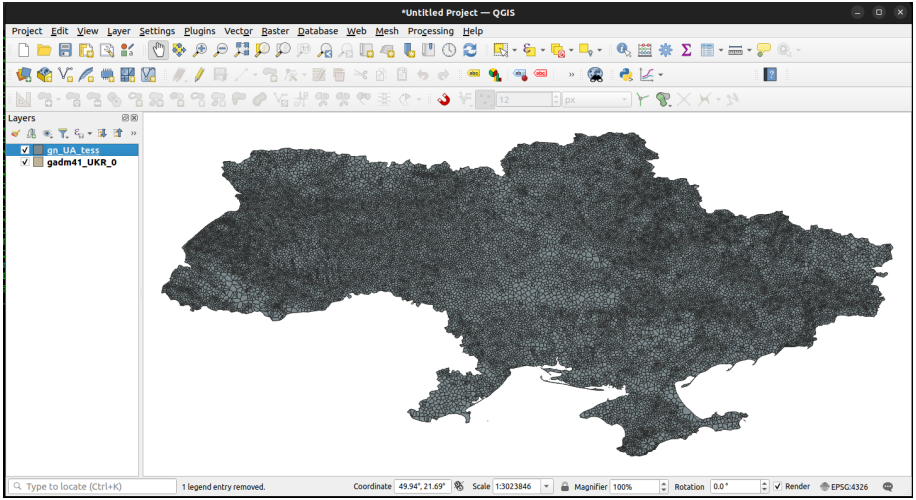
Vignette 1. Load *Ukraine's national borders* (`Layer` → `Add Layer` → `Add Vector Layer`). `gadm41_UKR_0.json` file in `Data/GADM`



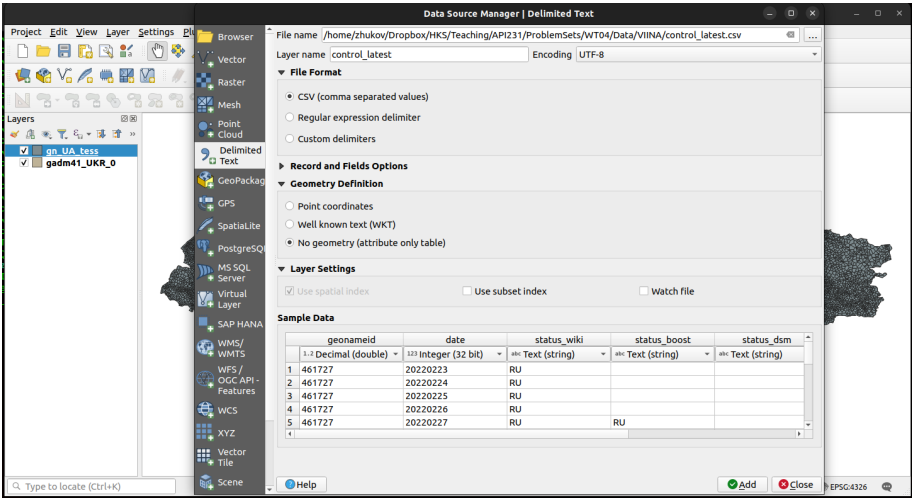
Also load the *populated place borders* (`Layer` → `Add Layer` → `Add Vector Layer`). `gn_UA_tess.geojson` file in `Data/VIINA`



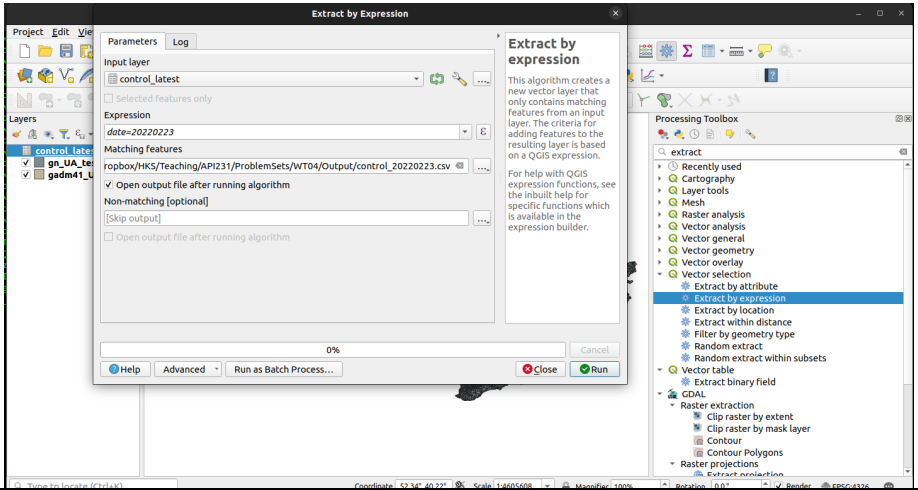
There are 33,141 populated places in Ukraine. This is the level at which territorial control is measured



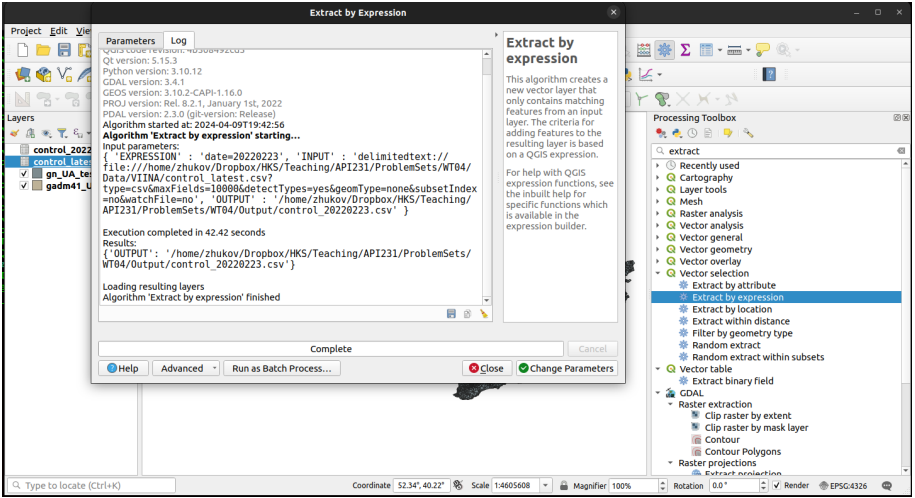
Load the *territorial control data* (`control_latest.csv`) as a delimited text file with no geometries. This is a HUGE table (> 25M rows), of which we'll be using only a small part. Take a note of how the `date` field is formatted (`YYYYMMDD`)



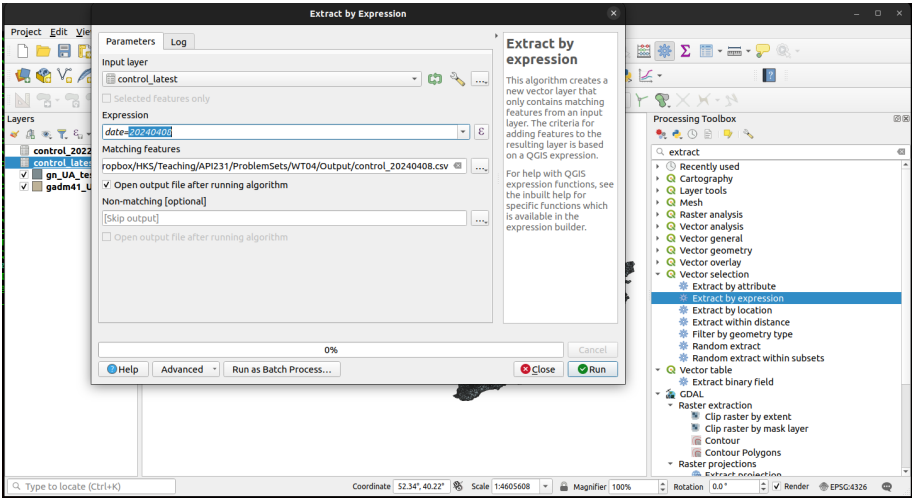
Let's take a subset of this file by date, starting with the day before the full scale invasion (23 Feb 2024). Go to the "*Extract by Expression*" tool in "Processing Toolbox" → "Vector selection". Set `Input layer`: `control_latest` and set `Expression`: `date=20220223`. Save as `control_20220223.csv`



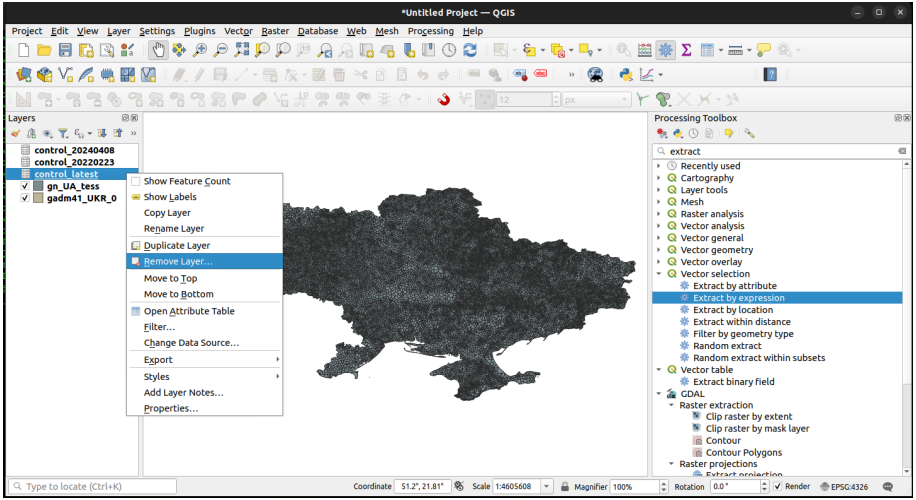
This will take a few minutes to run due to the file size.



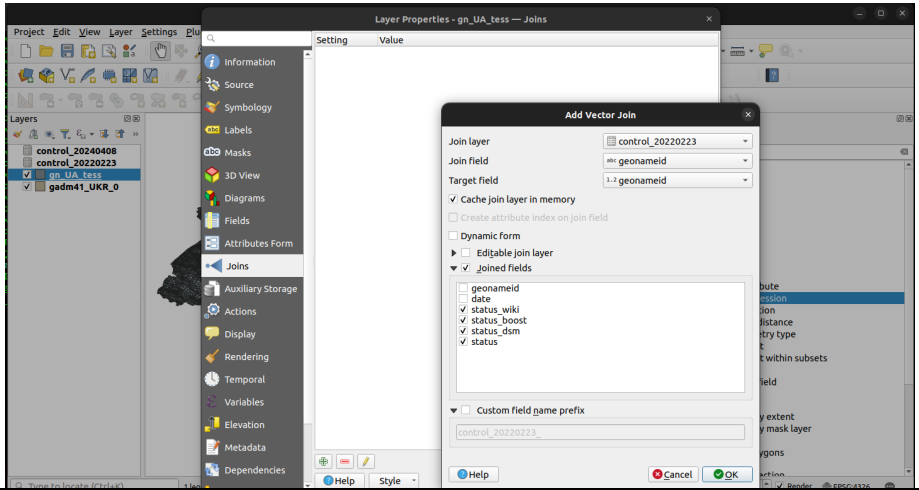
Repeat this process for the most recent date (8 Apr 2024). Set `Input layer`: `control_latest` and `Expression`: `date=20240408`. Save as `control_20240408.csv`



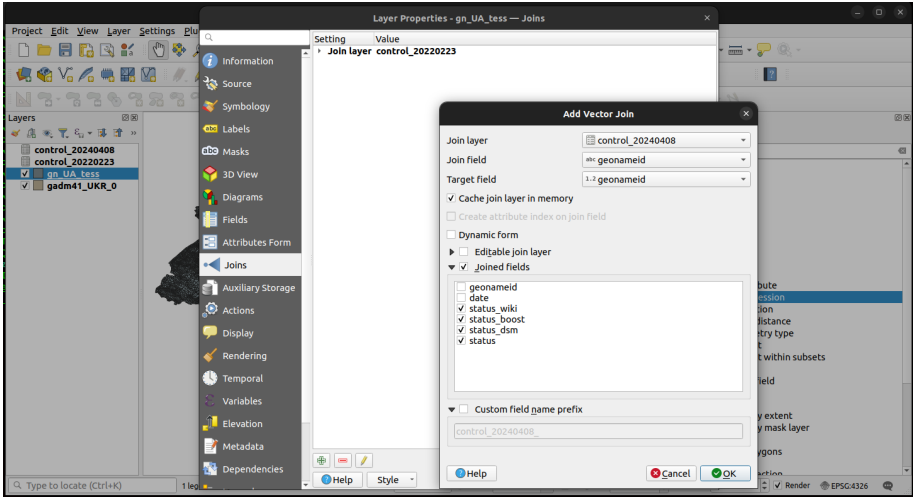
Once both subset tables are loaded, you can remove the original `control_latest` from memory



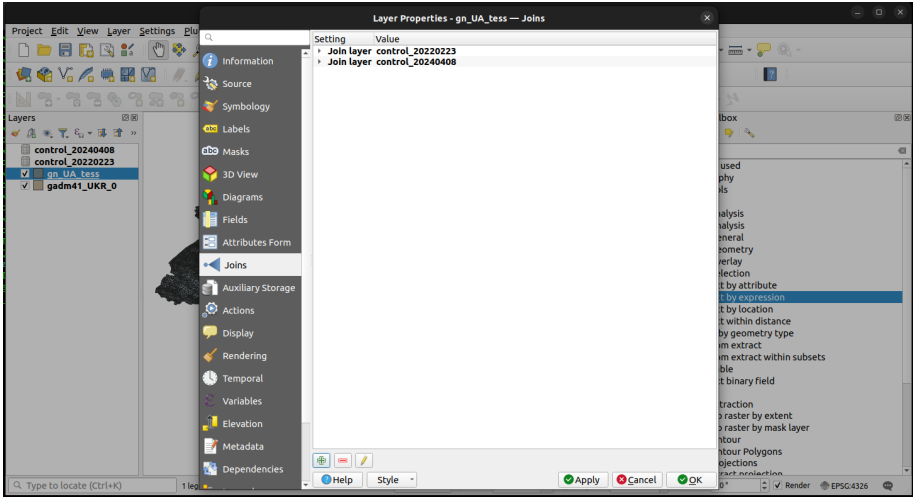
Let's now join the two subset tables to the populated place geometries. Go to the "Joins" tab in layer "Properties" for `gn_UA_tess`, and add a new join with `Join layer` = `control_20220223` and `geonameid` as the `Join field` and `Target field`. Select the four `status*` fields as `Joined fields`



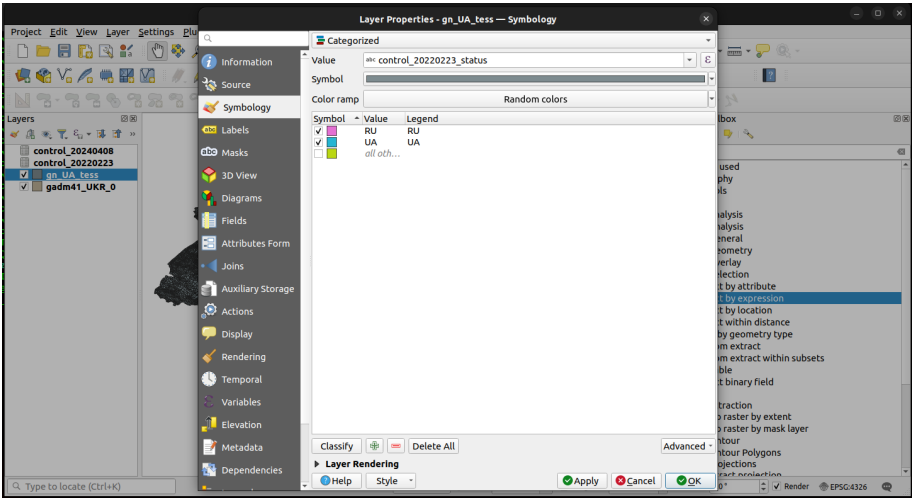
Add a second join with `control_20240408` as `Join layer`



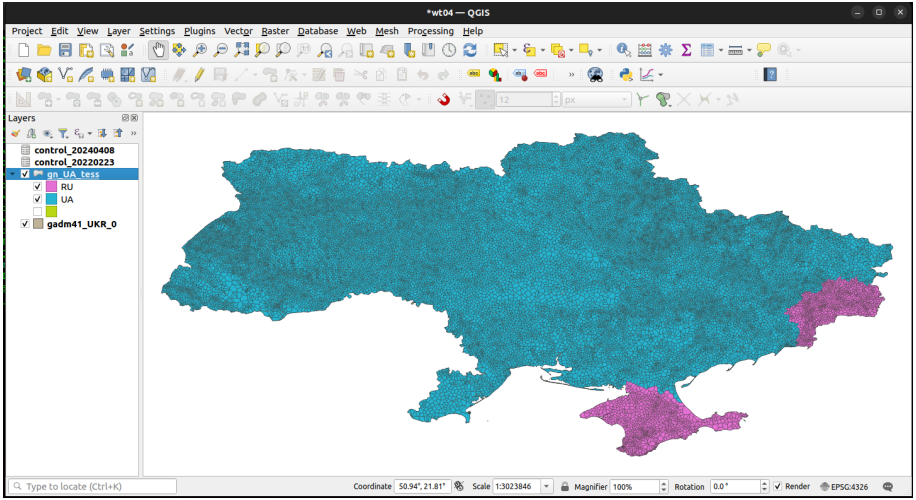
The two join layers should now appear in the "Joins" tab



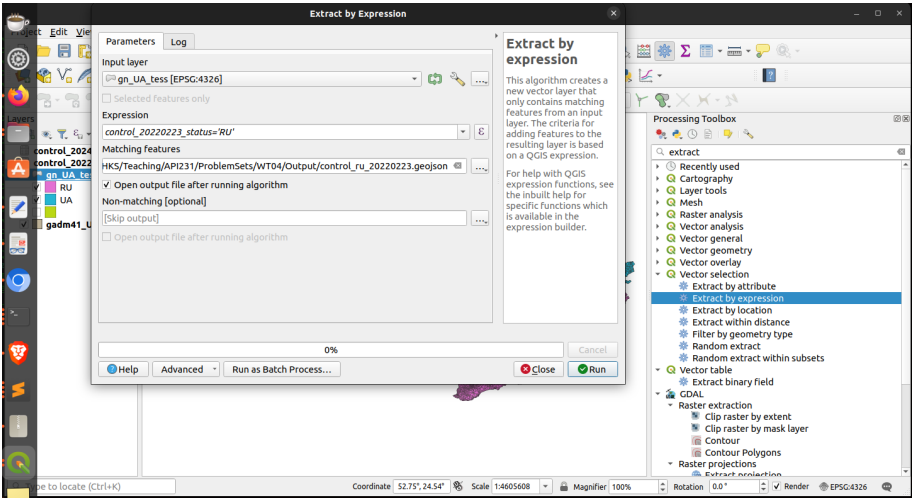
Let's visualize the control status, to make sure everything is right. Change to symbology to `Categorized` with `Value` = `control_20220223_status` and click `Classify` and `OK`



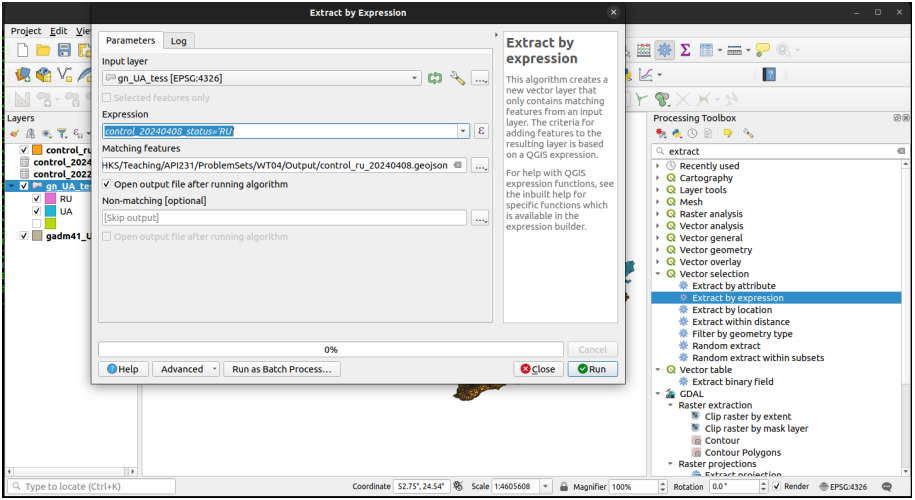
This looks about right. Now we just need to extract these occupied areas and calculate how much of Ukraine's territory they represent



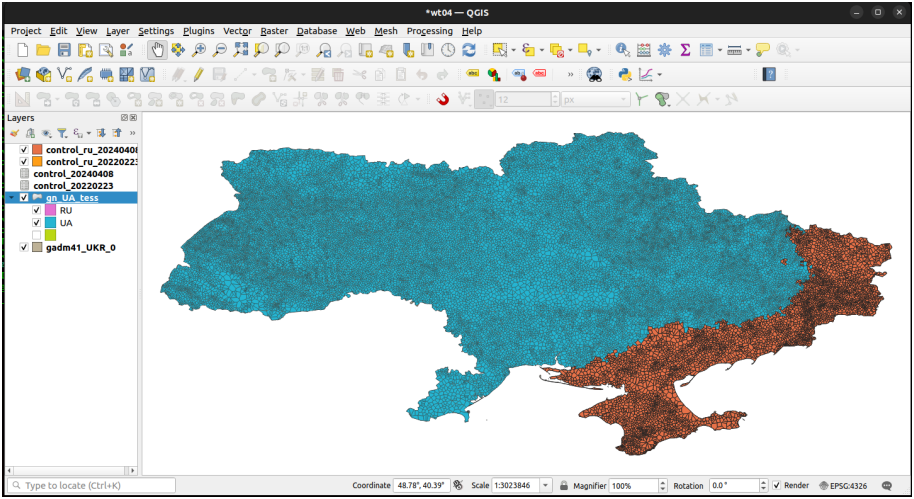
Go back to the "Extract by Expression" tool, with `Input layer`: `gn_UA_tess` and `Expression`: `control_20220223_status='RU'`. Save the output as `control_ru_20220223.geojson`



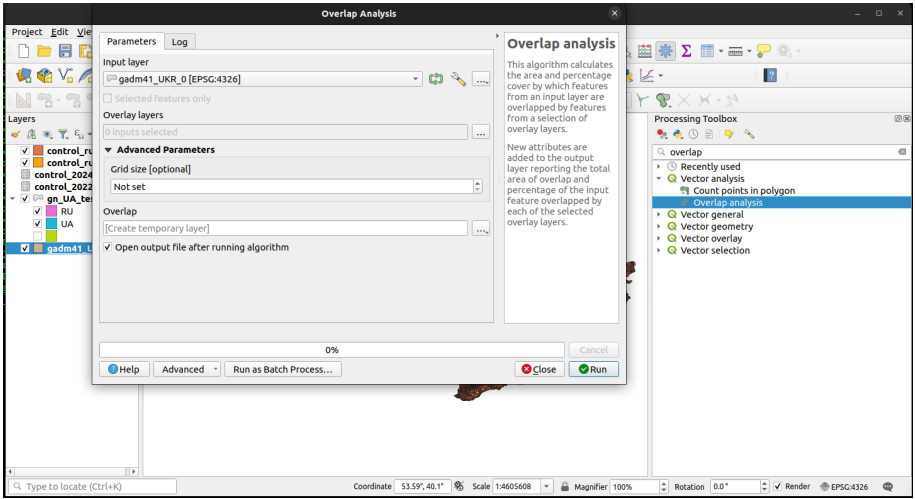
Repeat for the latest date, with `Expression`: `control_20240408_status='RU'`. Save the output as `control_ru_20240408.geojson`



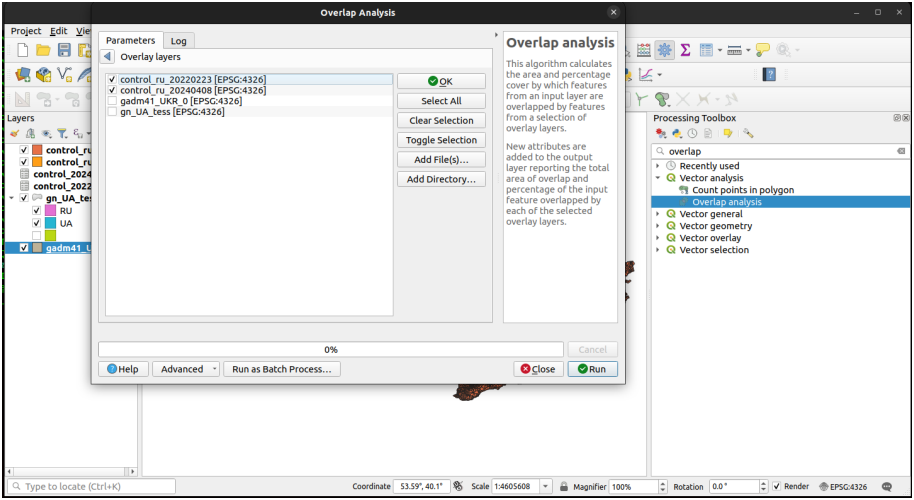
The extracted areas should appear in the project window.



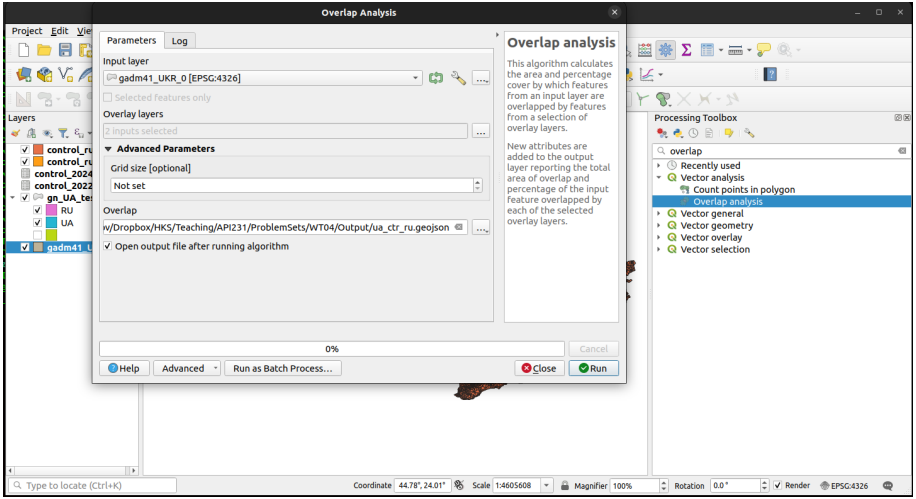
Go to the "Overlap Analysis" tool in "Processing Toolbox" → "Vector analysis". Set `Input layer`: `gadm41_UKR_0` (country-level borders). Click on the `...` button next to `Overlay layers`



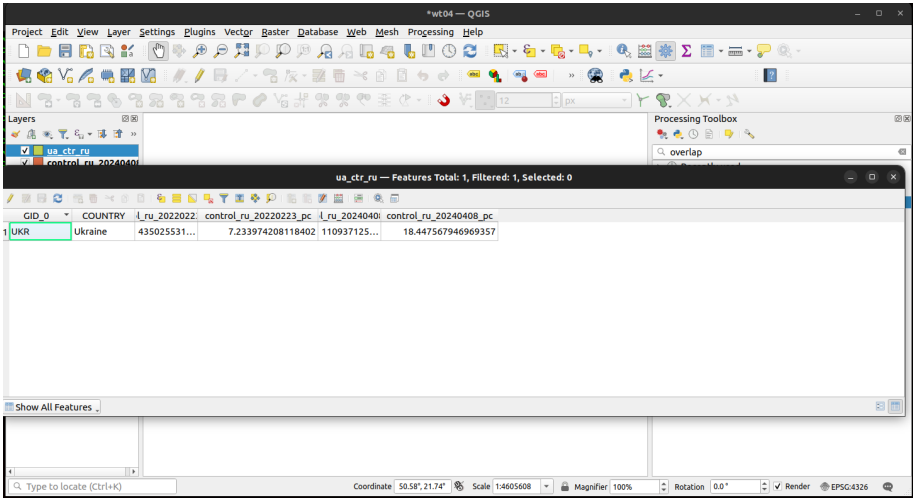
Select `control_ru_20220223` and `control_ru_20240408` as `Overlay layers`. Click `OK`



Save the output as `ua_ctr_ru.geojson`

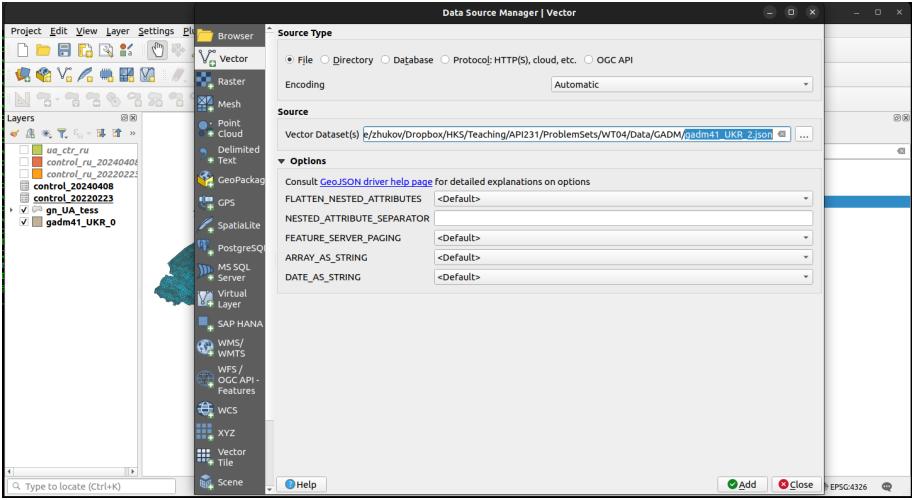


Open the attribute table for the newly-created `ua_ctr_ru` layer. The `*_pc` variables indicate that Russia occupied 7.23% of Ukraine's territory on 23 Feb 2024 and 18.45% on 8 Apr 2024

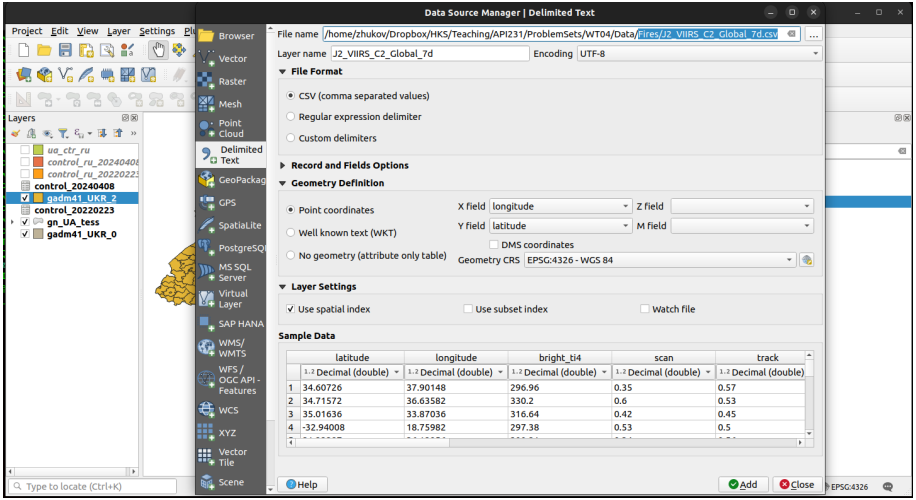


Comparing media event reports to remote sensing data on fires

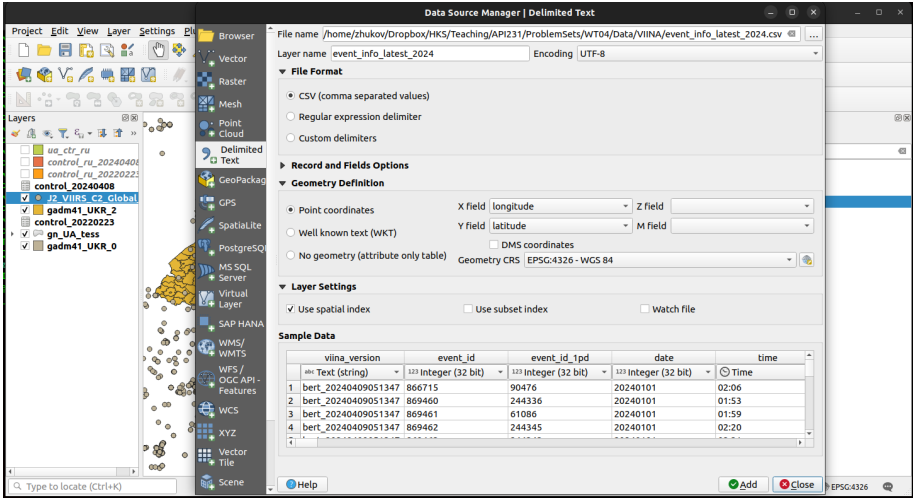
Vignette 2! Let's load *Ukrainian district borders*: `gadm41_UKR_2.json` from `Data/GADM`. These will be our spatial units of analysis



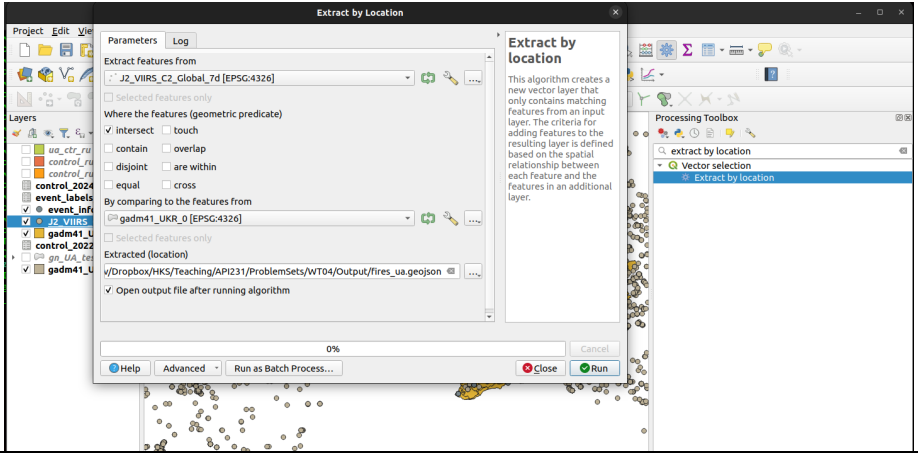
Load *active fires data* as delimited text: `J2_VIIRS_C2_Global_7d.csv`. Make sure the `X` and `Y` fields are properly specified, check box next to ✓ `Use spatial index`



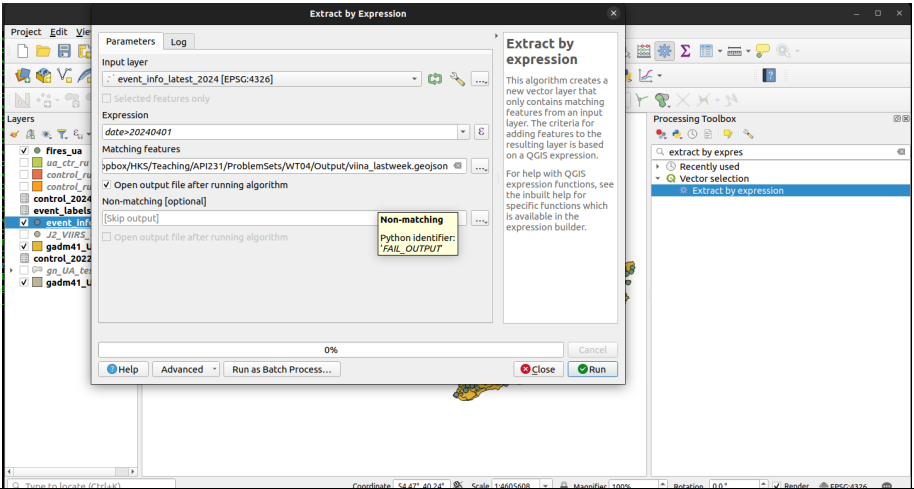
Load *media event reports* as delimited text: `event_info_latest_2024.csv`. Here, too, specify the `X` and `Y` fields and check the box next to ✓ `Use spatial index`



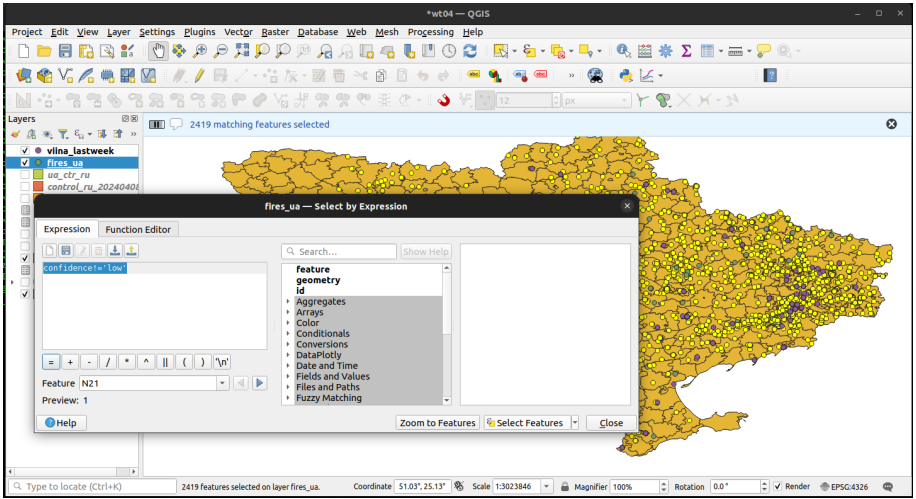
Let's extract just the fires inside of Ukraine. Go to the "Extract by Location" tool in "Processing Toolbox" → "Vector selection". Set `Extract features from`: `J2_VIIRS_C2_Global_7d`, check ✓ `Intersect`, `By comparing to the features from`: `gadm41_UKR_0`. Save the extracted features as `fires_ua.geojson`



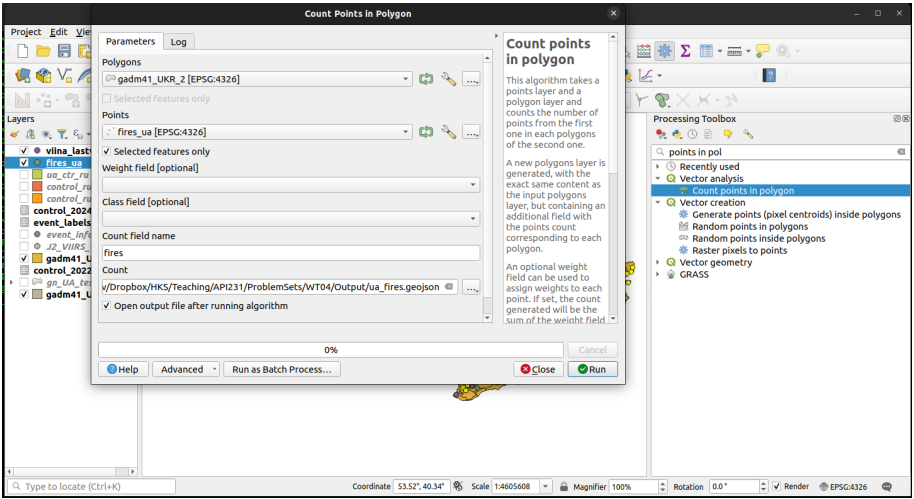
Let's also extract the media reports for the same time period as the fires (last week). Go to the "Extract by Expression" tool and set `Input layer`: `event_info_latest_2024`, and `Expression`: `date>20240401`. Save the extracted features as `viina_lastweek.geojson`



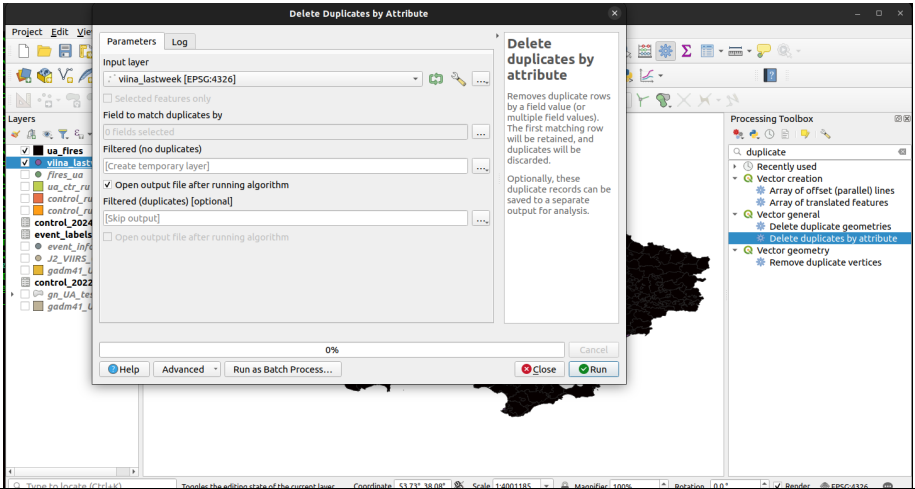
Before counting the fires, let's remove the "low confidence" fire anomalies. With `fires_ua` as the active layer, go to `Select by Expression`, with `Expression`: `confidence!='low'` (`!=` means "$\neq$"). This should select about 2419 features



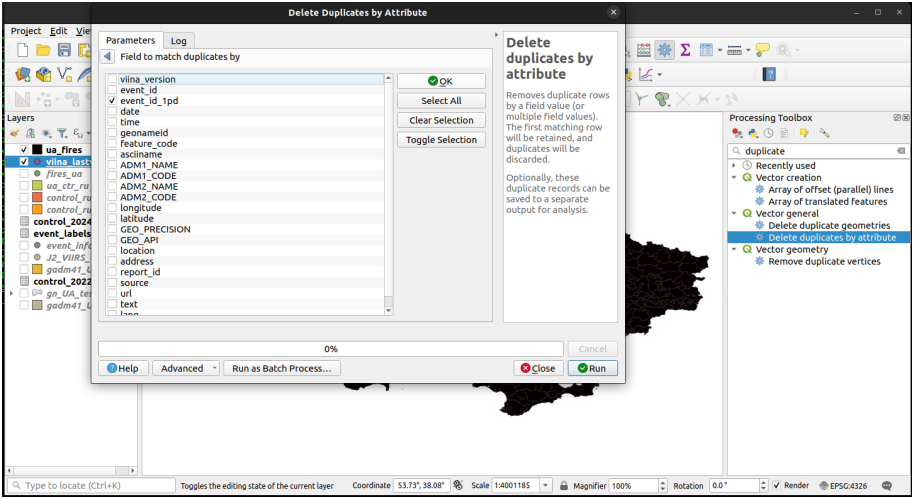
Open the "Count Points in Polygon" tool, set `Polygons`: `gadm41_UKR_2`, `Points`: `fires_ua`, check the box ✓ `Selected features only`, name the count field `fires` and save the output as `ua_fires.geojson`



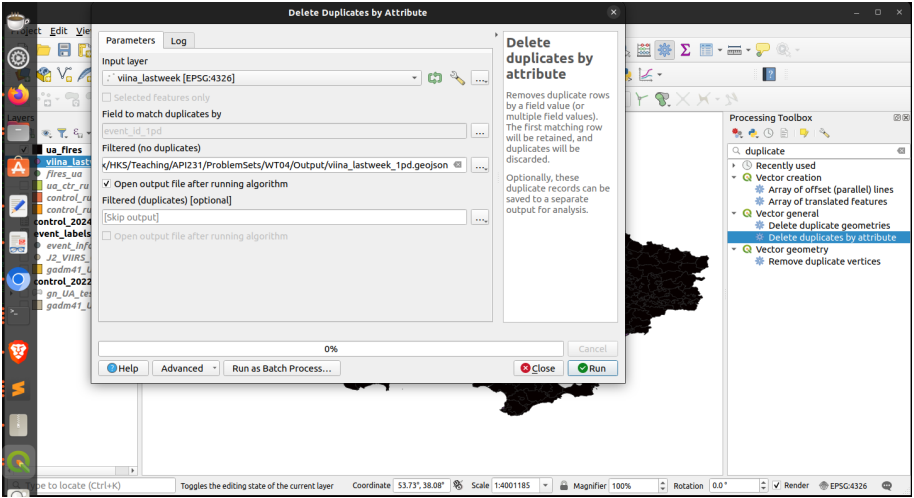
Before counting the media reports, let's remove the obvious duplicates. Open the "*Delete Duplicates by Attribute*" tool in "Processing Toolbox" → "Vector general". Set `Input layer`: `viina_lastweek`. Click the `...` button next to `Field to match duplicates by`



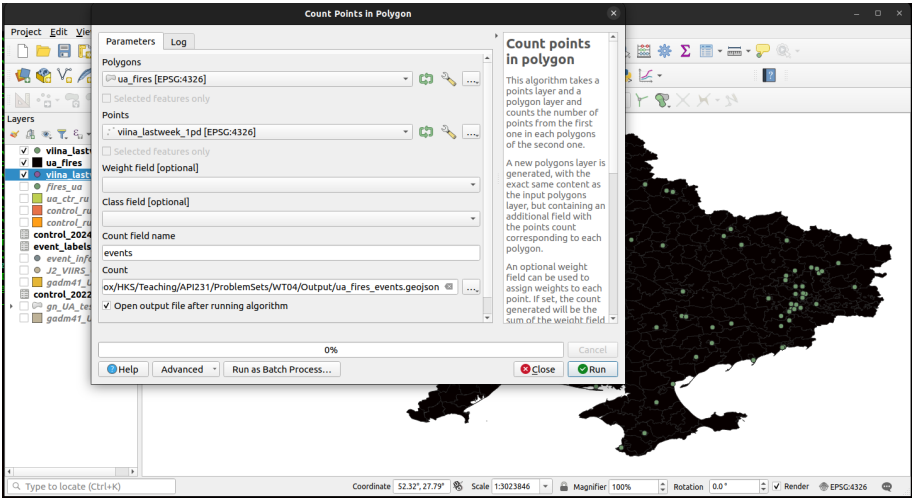
Select `event_id_1pd` as the field, click `OK`



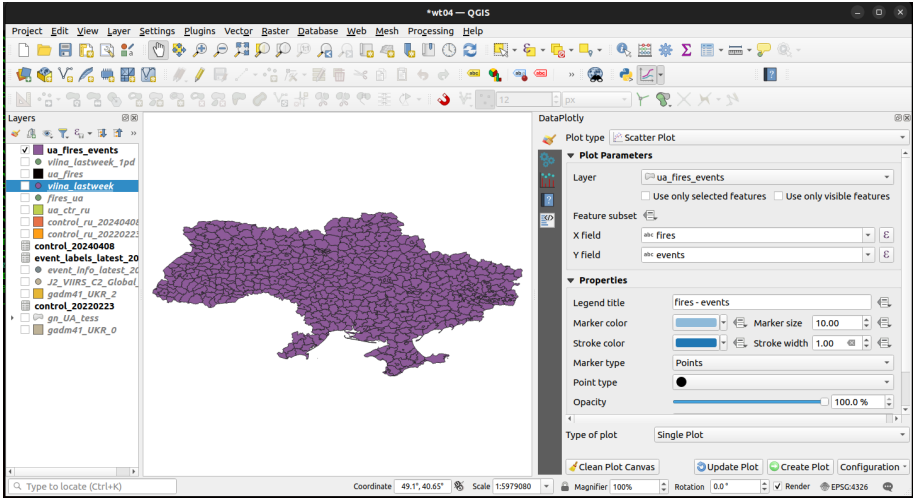
Save the deduplicated output as `viina_lastweek_1pd.geojson`



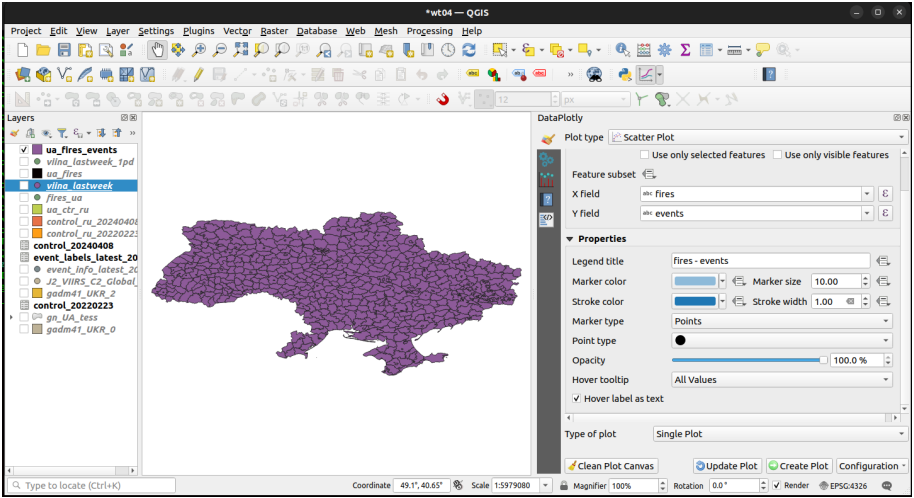
Go back to "Count points in polygon". Set `Polygons`: `ua_fires`, `Points`: `viina_lastweek_1pd`. Name the count field `events` and save as `ua_fires_events.geojson`



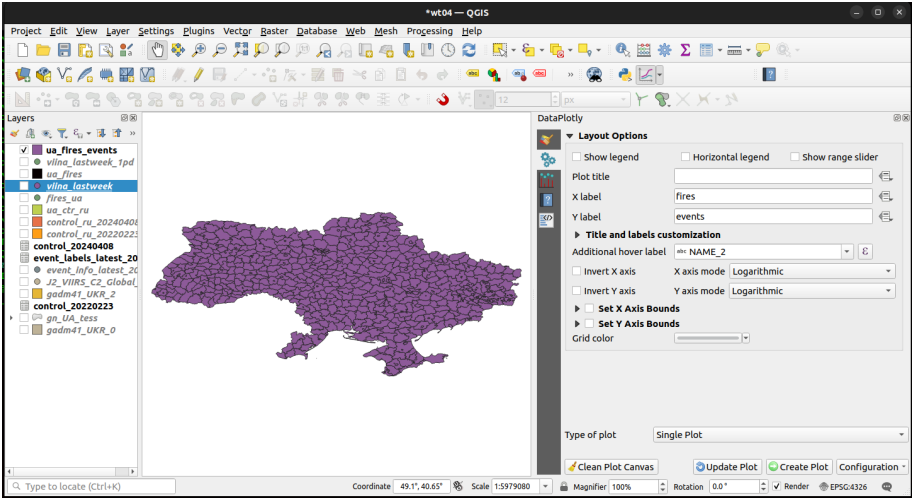
Open the Plotly tool. Set `Plot type` = `Scatter Plot`, Layer = `ua_fires_events`, `X field` = `fires`, `Y field` = `events`



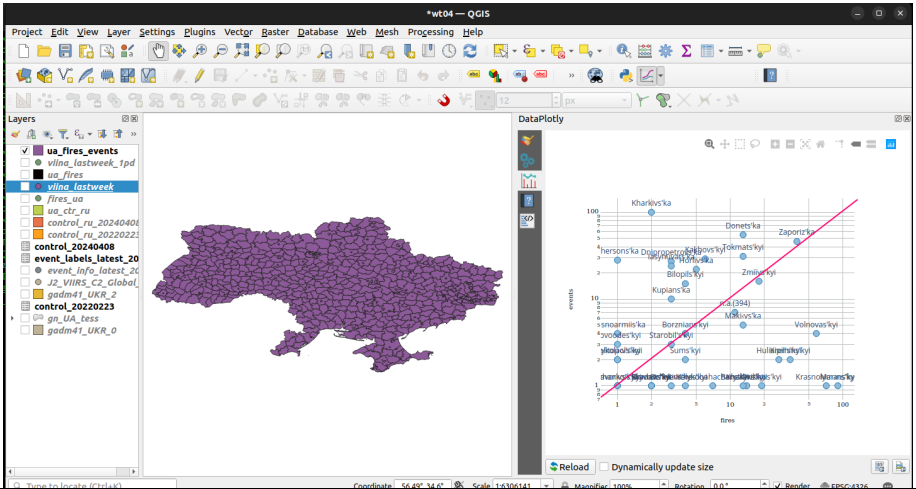
Check the box next to ✓ `Hover label as text`



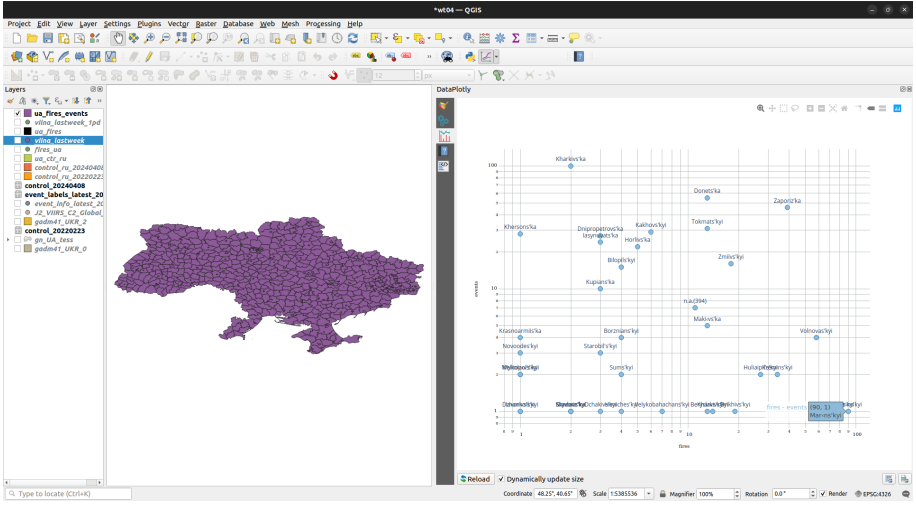
In "Layout Options", uncheck the box next to ☐ `Show legend`. Change the `X` and `Y axis mode` to `Logarithmic`



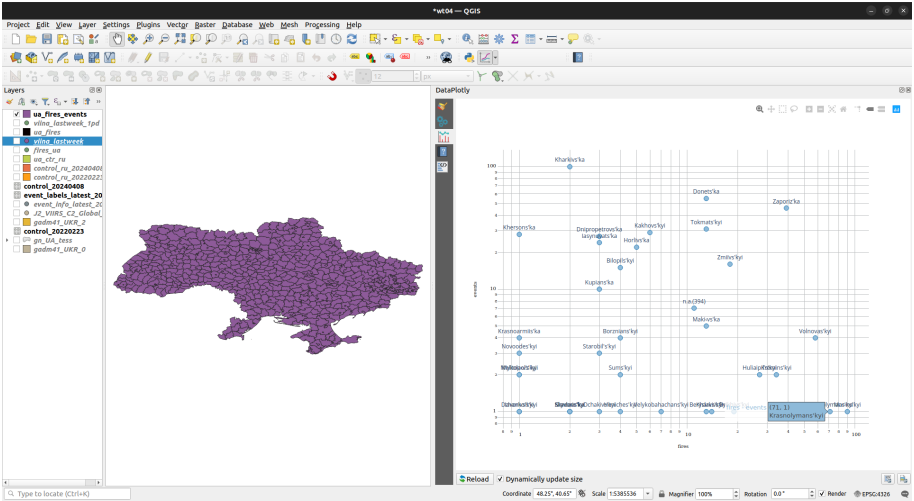
You can think of the points falling along the (imaginary) red line here as locations where both active fires data and media reports captured the same number of incidents. Points below (above) this line are ones where the fires data caught more (fewer) incidents than media reports



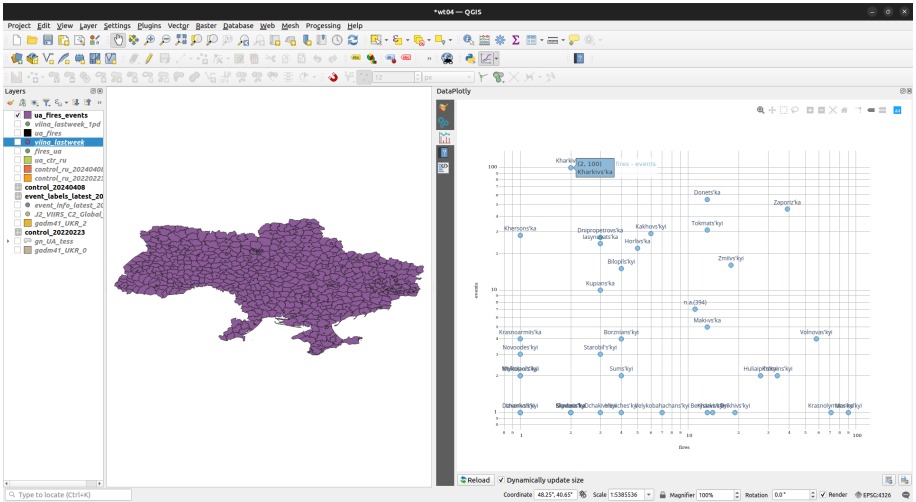
Locations with more active fires than media reports include places like Mar'inka...



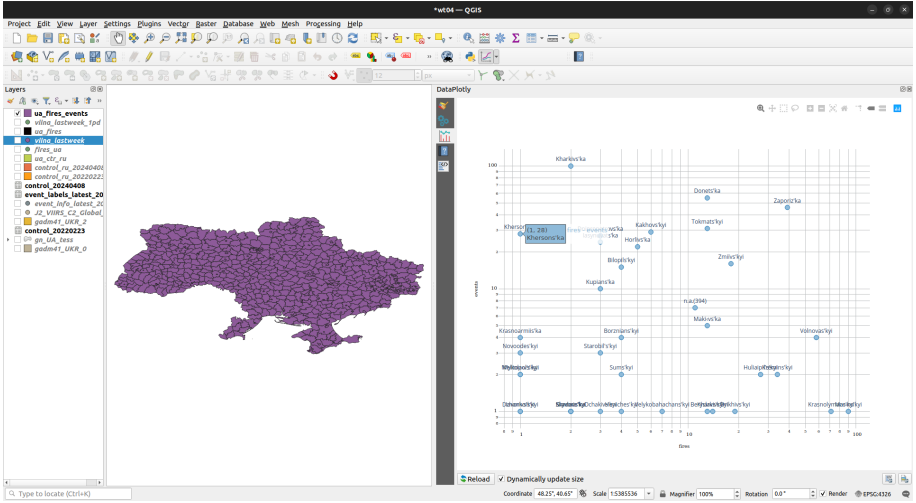
. . . and Lyman (in former Krasnolymans'kyy disrict). These are mostly-destroyed front line towns that are hard for journalists to access



Places where media reports capture more incidents include Kharkiv (Ukraine's second largest city, under Ukrainian control)...



. . . and Kherson (another province capital near the front line, but under Ukrainian control). Both of these places are far easier for journalists to access than the "no man's land" towns in the lower-right corner



You can perform all these steps in R
(see replication code `wt04_demo.R` in `WT04.zip`)

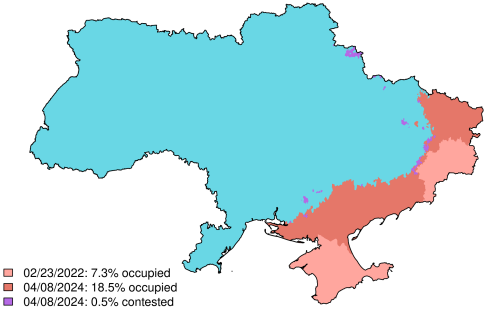


Figure 13: Vignette 1

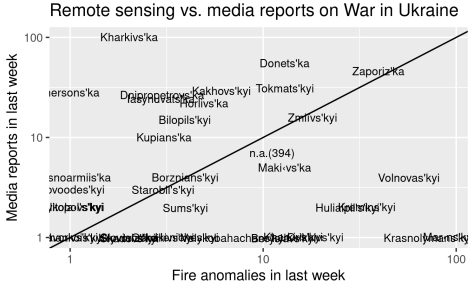


Figure 14: Vignette 2