Collecting data to monitor housing markets in rural areas: a web scraping approach tested in the Övre Norrland region (Sweden)

Carlos Tapia, Nordregio

2023-07-21

1 Introduction

This document describes a process to collect and analyse real-time data on property transactions in Sweden. The data collected in this study can be useful to analyse the performance of local housing markets, particularly in areas where tensions are expected to emerge as a consequence of external shocks.

The analysis focuses on the on the Swedish northernmost Counties of Norrbotten and Västerbotten. Both of these NUTS-3 regions are included in the Upper Norrland - Övre Norrland NUTS-2 region, which is one of the Living Labs included in the GRANULAR Project. These two remote and mostly rural regions in Sweden are expected to attract population as a consequence of ongoing processes of green re-industrialization in the area. Housing markets in both regions are hence likely to suffer important changes in the years to come.

The work presented here has been produced with R and R Studio. To render this site we used Rmarkdown, an open-source scientific and technical publishing system based on Markdown, a lightweight markup language for creating formatted text using a plain-text editor. This format allows to embed text and code in a self-contained html document, ensuring transparency and replicability of research.

2 Data scrap

The process starts with data collection. To the best of our knowledge, in Sweden there is no official database of real estate prices and registrations of title including geocoded information. This prevents a detailed analysis of housing markets in areas outside the big urban centres.

In response to this scarcity of information, we scrap data from the Hemnet portal, the largest digital marketplace for housing in Sweden. Every year, around 200,000 homes are published on Hemnet, which represents 90 percent of the homes sold annually in the country ("Hemnet" 2023). Hence, the Hemnet database provides access to a representative sample of housing registrations in Sweden.

For each item in the database, the Hemnet portal provides information on: property location (municipality, neighborhood, street and house number), type of property (different types of apartments and houses), property size (including indoor and outdoor area), property facilities and amenities (balcony, garden, etc.), settlement date (day, month and year), and price information (settlement price and price variation during the bidding process).

The following chunk of code shows how this information was scraped hemnet portal using the `ralger` library (Ihaddaden 2021)[1]. Before scraping the website, we confirmed that the `robot.txt` file does not forbid such operation. Moreover, the scraping instruction includes the `askRobot = T` argument, which instructs the function to ask the `robots.txt` if scraping is allowed. Additionally, the instruction is placed in a loop that waits a couple of seconds between each request, to avoid overloading the server. The ralger vignete provides more information on how to use this simple and effective scraping tool.

```
if(exists("hemnet_data") == F) {

# Define base link to the hemnet data for the Övra Norrland region (Norrbotten and Vasterbotten)
  base_link <- "https://www.hemnet.se/salda/bostader?location_ids%5B%5D=17763&location_ids%5B%5D=17764"

  # Define the element ID in the website to scrap
  my_nodes <- c(
    ".sold-property-listing__heading", # location
    ".sold-property-listing__location", # detailed location
    ".svg-icon__fallback-text", # Property type
    ".sold-property-listing__area", # size
    ".sold-property-listing__land-area", # land area
    ".hcl-label--sold-at", # settlement date
    ".sold-property-listing__fee", # Estimated monthly fee paid to housing association
    ".sold-property-listing__price-per-m2", # Price in SEK per sqm
    ".sold-results__normal-hit" # All data
  )

  # Define the column names in the table with the scraped data
  names <- c(
    "address",
    "location",
    "type",
    "size",
    "land_area",
    "date",
    "fee",
    "seksqm",
    "allinfo"
  )

  # hemnet gives access to 50 pages on each query:
  links <- paste0(base_link,"&page=", 1:50)

  # The following loop scraps the information for the i html links create above, waiting a couple of se
conds between each request, to avoid overloading the server.
  for (i in 1:length(links)) {
    if(i==1){
      # On i = 1, create a base df and check that the robot.txt does not forbid scraping this web page
      hemnet_data <- try(tidy_scrap(link = links[i], nodes = my_nodes, colnames = names, clean = T, ask
Robot = T))
    } else {
      hemnet_data_i <- try(tidy_scrap(link = links[i], nodes = my_nodes, colnames = names, clean = T))
      hemnet_data <- rbind(hemnet_data,hemnet_data_i)
    }
    Sys.sleep(2)
    print(paste0("Page: ", i ))
    print(Sys.time())
  }
}

knitr::kable(head(hemnet_data[,-9],2), booktabs = TRUE) %>%
  kable_styling(
    bootstrap_options = c("stripped","hover","condensed","responsive")
  )
```

address	location	type	size	land_area	date	fee	seksqm
Hermelinvägen 108	Hermelinvägen 108 <td> LägenhetLägenhet Mariedal, Umeå kommun </td>	Lägenhet	79 m² 3 rum	497 m² tomt	Såld 7 juli 2023	4 238 kr/mån	30 316 kr/m²
Odalvägen 4	Odalvägen 4 <td> VillaVilla Södra Sunderbyn, Luleå kommun </td>	Villa	127 <td> + 30 m² 5 rum </td>	755 m² tomt	Såld 6 juli 2023	4 214 kr/mån	14 154 kr/m²

3 Data cleaning

The scraping tool is very effective in retrieving the raw information on property registrations and dwelling characteristics. However, the downloaded data is unformatted. The data needs to go through intense wrangling before it can be used for any practical purpose. The following chunk shows an ad-hoc (and non-optimised) approach to data wrangling. This is done using a combination of libraries, most notably the `dplyr` package (Wickham et al. 2023) in the `tidyverse` collection (Wickham et al. 2019).

```
if(exists("hemnet_clean") == F) {

# Clean the data
hemnet_clean <- as.data.frame(hemnet_data) %>%
  tidyr::separate(location, c(do.call(paste0, expand.grid(letters, 1:3))), sep = "          ") %>%
  dplyr::select(d1,e1) %>%
  rename(neigbrhd=d1,muni=e1) %>%
  cbind(.,hemnet_data) %>%
  mutate(
    across(
      .cols = everything(),
      .fns = ~ str_replace_all(
        string = ..1,
        pattern = "[\n]",
        replacement = ""
      )
```

```
    )
  ) %>%
  mutate(address=str_trim(address)) %>%
  dplyr::select(-location) %>%
  mutate(neigbrhd = str_replace_all(neigbrhd,"[[:blank:]]", "")) %>%
  mutate(neigbrhd = str_replace_all(neigbrhd,",", "")) %>%
  mutate(muni = str_replace_all(muni,"kommun", "")) %>%
  mutate(muni = str_replace_all(muni,"[[:blank:]]", "")) %>%
  mutate(muni = str_replace_all(muni,"[\n]", "")) %>%
  mutate(addr_mun = paste0(address,", ",muni)) %>%
  mutate(addr_ngh_mun = paste0(address,", ", neigbrhd,", ", muni)) %>%
  mutate(allinfo = str_replace_all(allinfo,"[\n]", "")) %>%
  mutate(soldprcsek = str_match(allinfo, "Slutpris\\s*(.*?)\\s*kr")[,2]) %>%
  mutate(soldprcsek = str_replace_all(soldprcsek,"[[:blank:]]", "")) %>%
  mutate(price_sek = as.numeric(soldprcsek)) %>%
  mutate(pricechng_per = str_match(allinfo, "kr \\s*(.*?)\\s*%")[,2]) %>%
  mutate(sek_sqm = str_replace_all(seksqm,"kr/m²", "")) %>%
  mutate(sek_sqm = str_replace_all(sek_sqm,"[[:blank:]]", "")) %>%
  mutate(sek_sqm = as.numeric(sek_sqm)) %>%
  mutate(soldate = str_replace_all(date,"Såld","")) %>%
  mutate(soldate = str_replace_all(soldate,"[[:blank:]]", "")) %>%
  mutate(soldate = str_trim(soldate)) %>%
  mutate(soldate = str_replace_all(soldate,"januari","Jan")) %>%
  mutate(soldate = str_replace_all(soldate,"februari","Feb")) %>%
  mutate(soldate = str_replace_all(soldate,"mars","Mar")) %>%
  mutate(soldate = str_replace_all(soldate,"maj","May")) %>%
  mutate(soldate = str_replace_all(soldate,"juni","Jun")) %>%
  mutate(soldate = str_replace_all(soldate,"juli","jul")) %>%
  mutate(soldate = str_replace_all(soldate,"augusti","Aug")) %>%
  mutate(soldate = str_replace_all(soldate,"oktober","Oct")) %>%
  mutate(sold_date = as.Date(soldate, "%d%b%Y", )) %>%
  mutate(size = str_replace_all(size,"[[:blank:]]", "")) %>%
  mutate(size_sqm = sub("m².*", "", size) ) %>%
  mutate(room_n = str_match(size, "m²\\s*(.*?)\\s*rum")[,2])
}

knitr::kable(head(hemnet_clean[,-10],2), booktabs = TRUE) %>%
  kable_styling(
    bootstrap_options = c("stripped","hover","condensed","responsive")
  )
```

neigbrhd	muni	address	type	size	land_area	date	fee	seksqm	addr_mun	addr_ngh_mun	soldprcsek	price_sek	pricechng_per	sek_sqm	soldate	sold_date	size_sqm	ro
Mariedal	Umeå	Hermelinvägen 108	Lägenhet	79m²3rum	497 m² tomt	Såld 7 juli 2023	4 238 kr/mån	30 316 kr/m²	Hermelinvägen 108, Umeå	Hermelinvägen 108, Mariedal, Umeå	2395000	2395000	+9	30316	7jul2023	2023-07-07	79	3
SödraSunderbyn	Luleå	Odalvägen 4	Villa	127+30m²5rum	755 m² tomt	Såld 6 juli 2023	4 214 kr/mån	14 154 kr/m²	Odalvägen 4, Luleå	Odalvägen 4, SödraSunderbyn, Luleå	3550000	3550000	+2	14154	6jul2023	2023-07-06	127+30	5

Now the table has a much nicer structure. However, it was noticed that the normalized settlement price provided by the Hemnet portal (measured in SEK per square metre) has some inconsistencies. To avoid issues with the interpolation of the property prices that follows, we decided to re-calculate the average price information using the original size and price data provided by the portal. In a real-case scenario, this procedure should be re-assessed and validated.

As a final data cleaning step, we also remove the outliers (observations with z-scores greater than 3), as these points typically reflect atypical property values. These can be motivated by properties with exceptional characteristics on both sides if the price/quality spectrum, or are otherwise caused by typing errors. In either case, extreme property values may reduce the reliability of the interpolation step that follows.

```
# Clean the data and remove outliers
hemnet_clean2 <- hemnet_clean %>%
  mutate(room_n = str_match(size, "m²\\s*(.*?)\\s*rum")[,2]) %>%
  # Recalculate the information on settlement price per sqm:
  mutate(size_sqm_cl = gsub("\\+.*","",size_sqm)) %>%
  mutate(size_sqm_cl = gsub(",",".", size_sqm_cl)) %>%
  mutate(size_sqm_cl = as.numeric(size_sqm_cl)) %>%
  mutate(price_sek = as.numeric(price_sek)) %>%
  mutate(sek_sqm_calc = price_sek/size_sqm_cl) %>%
  mutate(z_scores = abs((sek_sqm_calc-mean(sek_sqm_calc, na.rm=T))/sd(sek_sqm_calc,na.rm=T))) %>%
  filter(z_scores<3)


knitr::kable(head(hemnet_clean2[,-10],2), booktabs = TRUE) %>%
  kable_styling(
    bootstrap_options = c("stripped","hover","condensed","responsive")
  )
```


neigbrhd	muni	address	type	size	land_area	date	fee	seksqm	addr_mun	addr_ngh_mun	soldprcsek	price_sek	pricechng_per	sek_sqm	soldate	sold_date	size_sqm	ro
Mariedal	Umeå	Hermelinvägen 108	Lägenhet	79m²3rum	497 m² tomt	Såld 7 juli 2023	4 238 kr/mån	30 316 kr/m²	Hermelinvägen 108, Umeå	Hermelinvägen 108, Mariedal, Umeå	2395000	2395000	+9	30316	7jul2023	2023-07-07	79	3
SödraSunderbyn	Luleå	Odalvägen 4	Villa	127+30m²5rum	755 m² tomt	Såld 6 juli 2023	4 214 kr/mån	14 154 kr/m²	Odalvägen 4, Luleå	Odalvägen 4, SödraSunderbyn, Luleå	3550000	3550000	+2	14154	6jul2023	2023-07-06	127+30	5

4 Geocoding

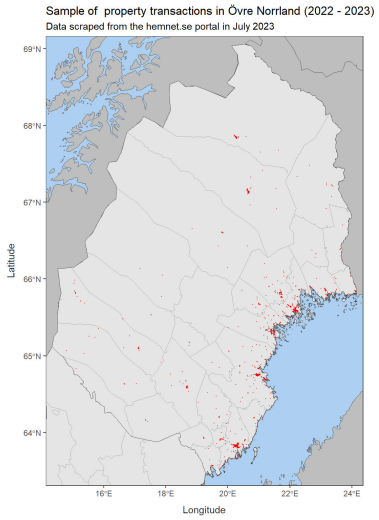
Once the dataset is clean, the following step focuses on geocoding the individual properties. This is performed by making use of the inverse-geocoding tool by OpenStreetMap Nominatim. This system can be accessed through the `tmaptools` library (Tennekes 2021). As on the previous step, we place the geocoding instruction inside a loop and add a time lag to respect the maximum number of queries per second allowed under the Nominatim Usage Policy. Once the loop ends, we add a coordinate reference system to the spatial entity that was created. For this, we use the `sf` package E. Pebesma and Bivand (2023) .

```
if(exists("geocodedOSM") == F) {

  # Geocode loop
  # Geocodes a location to coordinates, using the OpenStreetMap Nominatim (https://operations.osmfounda
tion.org/policies/nominatim/)
  for (i in 1:nrow(hemnet_clean)){
     if (i==1) {
      # Create the base table
      geocodedOSM <- try(geocode_OSM(hemnet_clean$addr_mun[i], details = TRUE,
                                     as.data.frame = TRUE, keep.unfound = T  ))
    } else {
      # Collect more points and feed the table
      new_df <- try(geocode_OSM(hemnet_clean$addr_mun[i], details = TRUE,
                                as.data.frame = TRUE, keep.unfound = T  ))
      geocodedOSM <- bind_rows(geocodedOSM,new_df)
    }
    print(paste0("Point: ", i ))
    print(Sys.time())
    Sys.sleep(1) # Respect the maximum number of queries per second to avoid overloading the server, ac
cording to the Nominatim Usage Policy (aka Geocoding Policy)
    # https://operations.osmfoundation.org/policies/nominatim/
  }
}

if(exists("geocoded_sf") == F) {

# Clean data and add crs:
geocoded_sf <- geocodedOSM %>%
  inner_join(hemnet_clean2, by= c("query" = "addr_mun" )) %>%
  dplyr::select(place_id,lat,lon,display_name,type.y,size_sqm,room_n,sold_date,price_sek,sek_sqm,sek_sq
m_calc,z_scores,pricechng_per,sold_date) %>%
  rename(address=display_name,type=type.y) %>%
  drop_na() %>%
  distinct() %>%
 st_as_sf(coords = c("lon", "lat"), crs=4326)
}
```

The following chunk shows how to produce a map with all the property registrations of title in the region (red dots) during the observed period (between 2022-11-10 and 2023-07-07). Alongside the property transactions, we also use some background layers retrieved from GISCO using the `giscoR` package (Hernangómez 2023). The map is plotted with the `ggplot2` library (Wickham 2016).

```
if(exists("interp_NN_ID_df_nuts2") == F) {

  # Download layers from GISCO

  # Countries
  countries <-
    gisco_get_countries(
      year = "2016",
      epsg = "3035",
      resolution = "03"
    )

  # All Swedish NUTS-2 regions
  nuts2_SE <- gisco_get_nuts(
    year = "2016",
    # epsg = "3035",
    epsg = "4326",
    resolution = "01",
    nuts_level = "2",
    country = "Sweden",
    spatialtype = "RG"
  )

  # Övre Norrland
  nuts2_SE33 <- gisco_get_nuts(
    year = "2016",
    # epsg = "3035",
    epsg = "4326",
    resolution = "01",
    nuts_level = "2",
    country = "Sweden",
    spatialtype = "RG",
    nuts_id = "SE33"
  )

}
```

```
# Plot the map
ggplot() +
  geom_sf(data = countries, color = "gray30", fill = "gray90") +
  geom_sf(data = lau_SE, fill = "transparent", size = 0.1, color="darkgray") +
  geom_sf(data = countries, color = "gray30", fill = "transparent") +
  geom_sf(data = nuts2_SE, color = "gray30", size = .5, fill = "transparent") +
  geom_sf(data = geocoded_sf, color = "red", size=0.01) +
  geom_sf(data = countries[which(countries$CNTR_ID != "SE"),], color = "gray30", fill = "gray") +
  coord_sf(xlim = c(14.12598, 24.36674), ylim = c(63.30829, 69.15997), crs = st_crs(4326), expand = FAL
SE)+
  labs(
    x = "Longitude",
    y = "Latitude",
    title = paste0("Sample of ", nrow(geom_sf)," property transactions in Övre Norrland (",min(lubridat
e::year(geocoded_sf$sold_date)),"- ", max(lubridate::year(geocoded_sf$sold_date)),")"),
    subtitle = "Data scraped from the hemnet.se portal in July 2023") +
  theme_bw() +
  theme(
    panel.background = element_rect(fill = "#ADCFF1", color = NA),
    plot.caption = element_text(face="italic", hjust = 1, size = rel(.75), color = "#939184"),
    axis.title.x = element_text(margin = unit(c(.3, 0, 0, 0), "cm"), color = "grey20"),
    axis.title.y = element_text(margin = unit(c(0, .3, 0, 0), "cm"), color = "grey20"),
    panel.grid.major = element_blank()
    # legend.position = ""
  )
```



5 Interpolation

In order to produce a map of housing prices, the points including the observed transaction values should be interpolated to generate an areal representation of housing prices (e.g. as a grid, a surface or a raster). In order to do this, standard practice is to fit a series of statistical models to estimate the values for unsampled locations in a regular grid that is then transformed to the desired format. A typical workflow includes fitting alternative models, compare model accuracy using cross validation or equivalent method, and finally predict the values for the unsampled points.

Frequently used interpolation models include voronoi polygons, nearest neighbour (NN) classifyers, with direct or inverse distance weighting, and various forms of kriging. Each method has its own set of advantages and disadvantages. Here, we shall demonstrate the use of two of such models, namely inverse distance weighted interpolation and ordinary kriging.

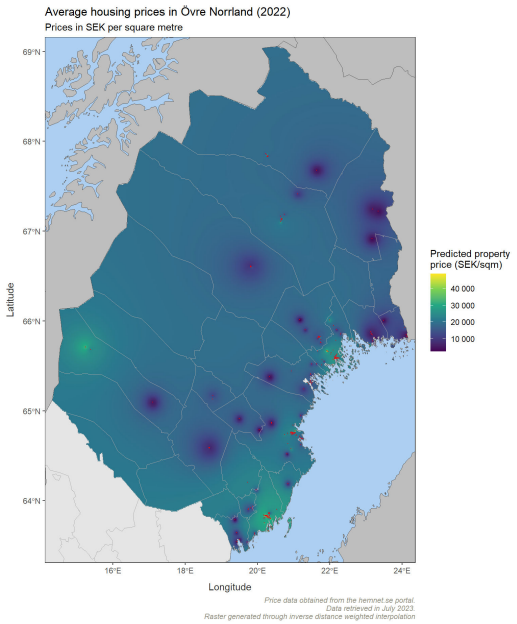
The chunk below applies the inverse distance weighted interpolation method on a subset of residential properties (flats, houses and terraced houses) in the Övre Norrland region. We use the `gstat` package by (Edzer J. Pebesma 2004; Gräler, Pebesma, and Heuvelink 2016) and its dependence `sp` (Edzer J. Pebesma and Bivand 2005; Bivand, Pebesma, and Gómez-Rubio 2008) to fit the model, alongside the `raster` package (Hijmans 2023) to do the interpolation. We fitted the model on all the sampled points for 2022, including n=520 geolocated transactions.

```
if(exists("interp_NN_ID") == F) {

  # Focus on residential properties
  geocoded_sf <- geocoded_sf[which(geocoded_sf$type %in% c("Villa", "Lägenhet", "Radhus")) ,]

  # Create a grid template to use as reference
  grd_template <- expand.grid(
    X = seq(from = 14.32598, to = 24.16674, by = 0.01),
    Y = seq(from = 63.40829, to = 69.05997, by = 0.01)
  )

  # Add crs
  alt_grd_template_raster <- grd_template %>%
    raster::rasterFromXYZ(crs = 4326)

  # Inverse distance weighted interpolation

  fit_NN_ID <- gstat::gstat( # using package {gstat}
    formula = sek_sqm_calc ~ 1,     # The column `sek_sqm` is what we are interested in
    data = as(geocoded_sf[which(lubridate::year(geocoded_sf$sold_date) %in% c(2022)),], "Spatial") # us
ing {sf} and converting to {sp}, which is expected
  )

  # Interpolate points
  interp_NN_ID <- raster::interpolate(alt_grd_template_raster, fit_NN_ID)

}
```

```
## [inverse distance weighted interpolation]
```

Once the interpolation is done, we are ready to visualise the result on a map. The following map shows all property registrations of title in the region in 2022 (red dots), and the interpolated raster with the predicted average property prices in SEK per square meter.

```
# Mask the interpolated raster
interp_NN_ID_masked_NUTS2 <- mask(interp_NN_ID, nuts2_SE33)

# Create a plotable version of the interpolated raster
interp_NN_ID_df_nuts2 <- raster::as.data.frame(interp_NN_ID_masked_NUTS2, xy=T) %>% na.omit()

# Plot the map
ggplot() +
  geom_sf(data = countries, color = "gray30", fill = "gray90") +
  geom_raster(data=interp_NN_ID_df_nuts2, aes(x = x, y = y, fill = var1.pred)) +
  scale_fill_viridis_c(labels = scales::label_number()) +
  geom_sf(data = lau_SE, fill = "transparent", size = 0.1, color="darkgray") +
  geom_sf(data = countries, color = "gray30", fill = "transparent") +
  geom_sf(data = geocoded_sf[which(lubridate::year(geocoded_sf$sold_date) %in% c(2022)),], color = "re
d", size=0.01) +
  geom_sf(data = countries[which(countries$CNTR_ID != "SE"),], color = "gray30", fill = "gray") +
  coord_sf(xlim = c(14.12598, 24.36674), ylim = c(63.30829, 69.15997), crs = st_crs(4326), expand = FAL
SE)+
  labs(
    x = "Longitude",
    y = "Latitude",
    title = "Average housing prices in Övre Norrland (2022)",
    subtitle = "Prices in SEK per square metre",
    caption = "Price data obtained from the hemnet.se portal.
    Data retrieved in July 2023.
    Raster generated through inverse distance weighted interpolation",
    fill = "Predicted property\nprice (SEK/sqm)"
  ) +
  theme_bw() +
  theme(
    panel.background = element_rect(fill = "#ADCFF1", color = NA),
    plot.caption = element_text(face="italic", hjust = 1, size = rel(.75), color = "#939184"),
    axis.title.x = element_text(margin = unit(c(.3, 0, 0, 0), "cm"), color = "grey20"),
    axis.title.y = element_text(margin = unit(c(0, .3, 0, 0), "cm"), color = "grey20"),
    panel.grid.major = element_blank()
    # legend.position = ""
  )
```



The map shows high or low property prices around the populated centres and very little variability on prices in areas outside the cities, towns and villages. In fact, the Västerbotten and Norrbotten Counties are very sparsely and evenly populated at the same time. This makes difficult to conduct this type of analysis at such a coarse scale.

Another problem could be the nature of the fitted model itself. NN classifiers are simple and powerful machine learning methods, but they are very sensitive to the parameters chosen, like the number of neighbours and the distance metrics used. At the same time, NN are *non-generalizing* algorithms, which implies that they simply "remember" the training data, making extrapolation tasks very challenging.

Alternative statistical methods, particularly Gaussian methods such as kriging, are more commonly found in the literature dealing with the spatial dynamics of housing markets (see e.g. Ewald, Sterner, and Sterner 2022). In the following chunk of code, we show how to apply a direct kriging method on the transactions occurred in the municipality of Skellefteå, in the Vasterbotten region.

Instead of filtering our previous dataset to focus on this municipality, we build a new dataset from scratch. By re-launching the scrap loop on a smaller area, we are able to cover a longer time series[2]. To produce this second scraped database, we followed exactly the same steps performed above (i.e. scraping, cleaning and geocoding. For the sake of space efficiency, we don't show the code again.

Once the data for Skelleftiå have been scraped, we interpolate the points for individual years using the direct kriging method implemented in the `automap` library (Hiemstra et al. 2008). Since the focus of this document is on the data collection part presented on the previous sections, we present a simplified interpolation workflow, focusing on median property values (measured in SEK per sqm) in each single address.

```
SE2482_bbox <- st_bbox(lau_SE2482)

# Create grid
grd_template_skell <- expand.grid(
  X = seq(from = SE2482_bbox["xmin"], to = SE2482_bbox["xmax"], by = 0.005),
  Y = seq(from = SE2482_bbox["ymin"], to = SE2482_bbox["ymax"], by = 0.005)
)

# Produce raster template
alt_grd_template_raster_skell <- grd_template_skell %>%
  raster::rasterFromXYZ(crs = 4326)

# Transform as sf
grd_template_sf_skell <- st_as_sf(grd_template_skell, coords = c("X","Y"), remove = FALSE)

# Focus on residential properties
geocoded_sf_skell <- geocoded_sf_skell[which(geocoded_sf_skell$type %in% c("Villa", "Lägenhet", "Radhu
s")) ,] %>%
  group_by(place_id) %>%
  mutate(sek_sqm_calc_m = median(sek_sqm_calc)) %>%
  ungroup()

  # Interpolate: Ordinary kriging (data for 2022)
  geocoded_sf_skell_22 <- geocoded_sf_skell[which(lubridate::year(geocoded_sf_skell$sold_date) %in% c(2
022)),]
  kriging_result_skell_22 <- autoKrige(sek_sqm_calc_m~1, geocoded_sf_skell_22, grd_template_sf_skell)
```

```
## Simple feature collection with 2750 features and 11 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 20.38282 ymin: 64.36278 xmax: 21.36977 ymax: 65.02733
## Geodetic CRS:  WGS 84
## # A tibble: 2,750 × 12
##    place_id  address          type  size_sqm room_n sold_date  price_sek sek_sqm
##    <chr>     <chr>            <chr> <chr>    <chr>  <date>         <dbl>   <dbl>
##  1 113295194 Norrbölegatan, … Läge… 55       3      2022-05-18   2000000   22715
##  2 110692651 Norrbölegatan, … Läge… 55       3      2022-08-26   1800000   25000
##  3 110692651 Norrbölegatan, … Läge… 55       3      2022-05-18   2000000   22715
##  4 110692651 Norrbölegatan, … Läge… 50       2      2022-09-07   1200000   30492
##  5 113295194 Norrbölegatan, … Läge… 50       2      2022-09-07   1200000   30492
##  6 110692651 Norrbölegatan, … Läge… 72,5     3      2022-05-18   2400000   30172
##  7 110692651 Norrbölegatan, … Läge… 70       3      2022-05-10   2035000   30238
##  8 110692651 Norrbölegatan, … Läge… 50       2      2022-02-24   1600000   22213
##  9 110692651 Norrbölegatan, … Villa 50       2      2022-02-24   1600000   21311
## 10 113295194 Norrbölegatan, … Läge… 50       2      2022-02-24   1600000   22213
## # ℹ 2,740 more rows
## # ℹ 4 more variables: sek_sqm_calc <dbl>, pricechng_per <chr>,
## #   geometry <POINT [°]>, sek_sqm_calc_m <dbl>
## [using ordinary kriging]
```

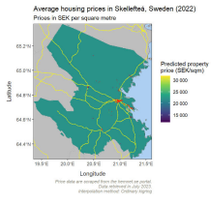
```
  kriging_ouputs_sf_skell_22 <- kriging_result_skell_22$krige_output
  kriging_ouputs_sf_skell_22 <-  st_set_crs(kriging_ouputs_sf_skell_22,4326)
  kriging_ouputs_raster_skell_22 <- raster::rasterize(kriging_ouputs_sf_skell_22,alt_grd_template_raste
r_skell)
  kriging_ouputs_raster_masked_SE2482_22 <- mask(kriging_ouputs_raster_skell_22, lau_SE2482)
  kriging_ouputs_df_masked_SE2482_22 <- raster::as.data.frame(kriging_ouputs_raster_masked_SE2482_22, x
y=T) %>% na.omit()

  # Interpolate: Ordinary kriging (data for 2023)
  geocoded_sf_skell_23 <- geocoded_sf_skell[which(lubridate::year(geocoded_sf_skell$sold_date) %in% c(2
023)),]
  kriging_result_skell_23 <- autoKrige(sek_sqm_calc_m~1, geocoded_sf_skell_23, grd_template_sf_skell)
```

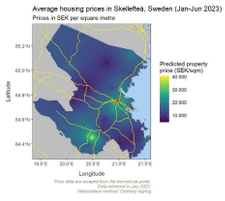
```
## Simple feature collection with 515 features and 11 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 20.37667 ymin: 64.39914 xmax: 21.25238 ymax: 64.94785
## Geodetic CRS:  WGS 84
## # A tibble: 515 × 12
##    place_id  address          type  size_sqm room_n sold_date  price_sek sek_sqm
##    <chr>     <chr>            <chr> <chr>    <chr>  <date>         <dbl>   <dbl>
##  1 112995759 Sunnanågatan, S… Villa 77+31    3      2023-07-06   2110000    3364
##  2 110278127 Sunnanågatan, S… Villa 77+31    3      2023-07-06   2110000   25581
##  3 110278127 Sunnanågatan, S… Villa 77+31    3      2023-07-06   2110000    3364
##  4 110278127 Sunnanågatan, S… Villa 77+31    3      2023-07-06   2110000   25581
##  5 110278127 Sunnanågatan, S… Villa 77+31    3      2023-07-06   2110000    3364
##  6 136571619 Tallvägen, Skel… Läge… 76+76    4      2023-07-06   1430000   19535
##  7 134015169 Tallvägen, Skel… Villa 76+76    4      2023-07-06   1430000   18902
##  8 134015169 Tallvägen, Skel… Läge… 76+76    4      2023-07-06   1430000   19535
##  9 134015169 Tallvägen, Skel… Villa 70+70    3      2023-01-24   1350000   17143
## 10 134015169 Tallvägen, Skel… Villa 76+76    4      2023-07-06   1430000   18902
## # ℹ 505 more rows
## # ℹ 4 more variables: sek_sqm_calc <dbl>, pricechng_per <chr>,
## #   geometry <POINT [°]>, sek_sqm_calc_m <dbl>
## [using ordinary kriging]
```

```
  kriging_ouputs_sf_skell_23 <- kriging_result_skell_23$krige_output
  kriging_ouputs_sf_skell_23 <-  st_set_crs(kriging_ouputs_sf_skell_23,4326)
  kriging_ouputs_raster_skell_23 <- raster::rasterize(kriging_ouputs_sf_skell_23,alt_grd_template_raste
r_skell)
  kriging_ouputs_raster_masked_SE2482_23 <- mask(kriging_ouputs_raster_skell_23, lau_SE2482)
  kriging_ouputs_df_masked_SE2482_23 <- raster::as.data.frame(kriging_ouputs_raster_masked_SE2482_23, x
y=T) %>% na.omit()
```

Now we can generate a series of maps showing the predicted housing prices, using the transaction information for properties sold during 2022 and the first half of 2023. Following, we plot the information for each of these periods.

```
# Plot the map for 2022
ggplot() +
  geom_sf(data = countries, color = "gray30", fill = "gray") +
  geom_raster(data=kriging_ouputs_df_masked_SE2482_22, aes(x = x, y = y, fill = var1.pred)) +
  scale_fill_viridis_c(labels = scales::label_number()) +
  geom_sf(data=highways$osm_lines, color = "yellow", size=0.01) +
  geom_sf(data = lau_SE, fill = "transparent", size = 0.1, color="darkgray") +
  geom_sf(data = countries, color = "gray30", fill = "transparent") +
  geom_sf(data = geocoded_sf_skell_22, color = "red", size=0.01) +
  coord_sf(xlim = c(SE2482_bbox[1], SE2482_bbox[3]), ylim = c(SE2482_bbox[2], SE2482_bbox[4]), crs = st
_crs(4326), expand = FALSE)+
  labs(
    x = "Longitude",
    y = "Latitude",
    title = "Average housing prices in Skellefteå, Sweden (2022)",
    subtitle = "Prices in SEK per square metre",
    caption = "Price data are scraped from the hemnet.se portal.
    Data retrieved in July 2023.
    Interpolation method: Ordinary kigring",
    fill = "Predicted property\nprice (SEK/sqm)"
  ) +
  theme_bw() +
  theme(
    panel.background = element_rect(fill = "#ADCFF1", color = NA),
    plot.caption = element_text(face="italic", hjust = 1, size = rel(.75), color = "#939184"),
    axis.title.x = element_text(margin = unit(c(.3, 0, 0, 0), "cm"), color = "grey20"),
    axis.title.y = element_text(margin = unit(c(0, .3, 0, 0), "cm"), color = "grey20"),
    panel.grid.major = element_blank()
    # legend.position = ""
  )

# Plot the map for 2023
ggplot() +
  geom_sf(data = countries, color = "gray30", fill = "gray") +
  geom_raster(data=kriging_ouputs_df_masked_SE2482_23, aes(x = x, y = y, fill = var1.pred)) +
  scale_fill_viridis_c(labels = scales::label_number()) +
  geom_sf(data=highways$osm_lines, color = "yellow", size=0.01) +
  geom_sf(data = lau_SE, fill = "transparent", size = 0.1, color="darkgray") +
  geom_sf(data = countries, color = "gray30", fill = "transparent") +
  geom_sf(data = geocoded_sf_skell_23, color = "red", size=0.01) +
  coord_sf(xlim = c(SE2482_bbox[1], SE2482_bbox[3]), ylim = c(SE2482_bbox[2], SE2482_bbox[4]), crs = st
_crs(4326), expand = FALSE)+
  labs(
    x = "Longitude",
    y = "Latitude",
    title = "Average housing prices in Skellefteå, Sweden (Jan-Jun 2023)",
    subtitle = "Prices in SEK per square metre",
    caption = "Price data are scraped from the hemnet.se portal.
    Data retrieved in July 2023.
    Interpolation method: Ordinary kigring",
    fill = "Predicted property\nprice (SEK/sqm)"
  ) +
  theme_bw() +
  theme(
    panel.background = element_rect(fill = "#ADCFF1", color = NA),
    plot.caption = element_text(face="italic", hjust = 1, size = rel(.75), color = "#939184"),
    axis.title.x = element_text(margin = unit(c(.3, 0, 0, 0), "cm"), color = "grey20"),
    axis.title.y = element_text(margin = unit(c(0, .3, 0, 0), "cm"), color = "grey20"),
    panel.grid.major = element_blank()
    # legend.position = ""
  )
```





6 Analysis

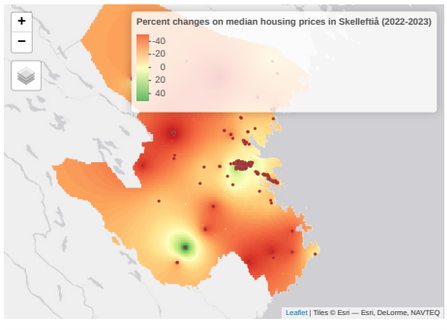
The predicted price information allows to perform some calculations on the dynamics of housing markets in Skellefteå. One relevant question we might like to answer is how much the construction of the new Nordvolt gigafactory might have affected the housing prices in the municipality. Allegedly, this effect must be noticeable considering the magnitude of this development. When fully operational in 2025, the new plant shall employ some 3000 workers, equivalent to 9 percent of Skellefteå's total work force. This is known as the "Nordvolt Effect" and has made the municipality develop a new strategic plan with a population target of 90,000 inhabitants by 2030, from 73,000 today.

In the following chunk, we produce a simple indicators on price changes between two consecutive years. We focus on the evolution of prices between 2022 and 2023. We visualise the indicator on an interactive map, using the `tmap` library (Tennekes 2018).

```
price_var_22_23 <- kriging_ouputs_df_masked_SE2482_22 %>%
  rename_with(.fn = ~ paste0(.x, ".2022"),var1.pred:var1.stdev) %>%
  inner_join(kriging_ouputs_df_masked_SE2482_23, by = c("x","y")) %>%
  rename_with(.fn = ~ paste0(.x, ".2023"),var1.pred:var1.stdev) %>%
  mutate(sek_sqm_23_22 = var1.pred.2023-var1.pred.2022) %>%
  mutate(sek_sqm_23_22_pc = 100*((var1.pred.2023/var1.pred.2022)-1))

# Prepare raster
price_var_22_23_raster <- raster::rasterFromXYZ(price_var_22_23[,c("x","y","sek_sqm_23_22_pc")],crs = 4
326)
```

According to the data scraped from the Hemnet portal, during the first quarter of 2023 the mean property value in Skellefteå fell by -21 percent in relation to the same period of the previous year. This fall is more than twice as large as the average price contraction in Sweden, which was -9 percent. Now we can produce an interactive map illustrating the evolution of the prices during the last quarter.

```
# Interactive maps
library(tmap)

m22_23 <- tm_shape(geocoded_sf_skell_23) +
  tm_dots(col = "red", size=0.01,  popup.vars = c("type", "size_sqm","room_n", "sold_date","price_se
k","sek_sqm","sek_sqm_calc"), legend.show = FALSE) +
  tm_shape(price_var_22_23_raster) +
  tm_raster("sek_sqm_23_22_pc", style="cont", title="Percent changes on median housing prices in Skelle
ftiå (2022-2023)")
tmap_mode("view")
m22_23
```




```
ttm() # back to initial mode: "plot"
```

As shown on the map, during the first half of 2023, the median property decline on the predicted property prices was -37 for the Skelleftiå municipality as a whole. However, the drop on property prices seems to affect more intensively the rural areas to the Western part of the municipality, while the pixels located in the urban centre over the Baltic coast experienced substantial price increases. This is taking place amid generalised property price drops in Sweden as a whole, due to the spike on interest rates. Hence, it could be hypothesised that the evolution of housing prices in the city of Skelleftiå could be related to the establishment of the Nordvolt factory and the attraction of population from other areas.

7 Conclussions

This document provides an illustration on how freely-available online data can be used to analyse housing markets in rural areas. We scraped housing registry data from the Hemnet portal, the largest digital housing marketplace in Sweden. Subsequently, we geolocated the points using the information associated to each property, and interpolated those points to generate areal maps of mean housing prices in the Övra Norrland region. Finally, we produced a very simple analysis focusing on the recent evolution of housing markets in the municipality of Skelleftiå.

Since this work was mostly developed for illustration, the data on property registration were not validated, while the interpolation models were fitted and selected through simplified procedures. Still, the scraped data and the modelling approaches tested show potential to overcome some of the limitations of the official statistics. The approach can be partially automated to deliver harmonised information on housing prices with high frequency and at a very fine-grained scale.

The transferability of this method is however limited by the availability and accessibility of similar data in other settings. In particular, the availability of observational data in specific areas seems to be a major limitation. The accuracy of the modelling results depend on the abundance, precision and reliability of the sampled data. Fewer property registrations in remote and isolated areas challenges the deployment and interpretation of interpolation models. Further work needs to be done to address these challenges.

8 References

Bivand, Roger S., Edzer J. Pebesma, and Virgilio Gómez-Rubio. 2008. *Applied Spatial Data Analysis with R*. Springer. https://doi.org/10.1007/978-0-387-78171-6.

Ewald, Jens, Thomas Sterner, and Erik Sterner. 2022. "Understanding the Resistance to Carbon Taxes: Drivers and Barriers Among the General Public and Fuel-Tax Protesters." *Resource and Energy Economics* 70 (November): 101331. https://doi.org/10.1016/j.reseneeco.2022.101331.

Gräler, Benedikt, Edzer Pebesma, and Gerard Heuvelink. 2016. "Spatio-Temporal Interpolation Using Gstat." *The R Journal* 8 (1): 204–18. https://journal.r-project.org/archive/2016/RJ-2016-014/index.html.

"Hemnet." 2023. In *Wikipedia*. https://sv.wikipedia.org/w/index.php?title=Hemnet&oldid=52992115.

Hernangómez, Diego. 2023. *giscoR: Download Map Data from GISCO API - Eurostat* (version 0.3.5). https://doi.org/10.5281/zenodo.4317946.

Hiemstra, P. H., E. J. Pebesma, C. J. W. Twenh"ofel, and G. B. M. Heuvelink. 2008. "Real-Time Automatic Interpolation of Ambient Gamma Dose Rates from the Dutch Radioactivity Monitoring Network." *Computers & Geosciences*.

Hijmans, Robert J. 2023. *Raster: Geographic Data Analysis and Modeling*. https://CRAN.R-project.org/package=raster.

Ihaddaden, Mohamed El Fodil. 2021. *Ralger: Easy Web Scraping*. https://CRAN.R-project.org/package=ralger.

Pebesma, Edzer. 2018. "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal* 10 (1): 439. https://doi.org/10.32614/RJ-2018-009.

Pebesma, Edzer J. 2004. "Multivariable Geostatistics in S: The Gstat Package." *Computers & Geosciences* 30 (7): 683–91. https://doi.org/10.1016/j.cageo.2004.03.012.

Pebesma, Edzer J., and Roger S. Bivand. 2005. "The R Journal: Classes and Methods for Spatial Data in R." *R News* 5 (2): 9–13. https://journal.r-project.org/articles/RN-2005-014/.

Pebesma, Edzer, and Roger Bivand. 2023. *Spatial Data Science: With Applications in R*. 1st ed. Boca Raton: Chapman and Hall/CRC. https://doi.org/10.1201/9780429459016.

Tennekes, Martijn. 2018. "tmap: Thematic Maps in R." *Journal of Statistical Software* 84 (6): 1–39. https://doi.org/10.18637/jss.v084.i06.

———. 2021. *Tmaptools: Thematic Map Tools*. https://CRAN.R-project.org/package=tmaptools.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan. 2023. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

9 Notes

1. When it comes to replicability of the scraping code, it is important to consider that it normally stops working upon minor changes on the web structure. This is quite frequent and it is of course beyond our control.↩

2. Recall that the hemnet portal only renders 50 pages per query, regardless of the scale.↩