

Rを利用した経済統計データの取得(2)

メタデータ	言語: ja 出版者: 静岡大学人文社会科学部 公開日: 2021-08-16 キーワード (Ja): キーワード (En): 作成者: 遠山, 弘徳 メールアドレス: 所属:
URL	https://doi.org/10.14945/00028293

資料

Rを利用した経済統計データの取得 (2)

遠山弘徳

- I はじめに
- II RStudioのProject管理とさまざまなデータ形式の読み込み方法
- III Rパッケージを利用したマクロデータの取得
 - III-1 Penn World Table — pwt9
 - III-2 World Bank — WDI (以上, 第25巻4号)
 - III-3 Eurostat — Rパッケージ eurostat
 - III-3-1 Rパッケージ eurostatの利用法
 - III-3-2 応用例 — 地理空間情報の利用 (以上, 本号)

III-3 Eurostat — Rパッケージ eurostat

欧州連合の統計局であるEurostatは、オープンデータサービスを通じて、欧州の人口統計、経済、健康、インフラ、交通などに関する数千ものデータセットを提供しています。Rパッケージ eurostatはそうした欧州統計局のオープンデータ取得のためのパッケージです。ヨーロッパ経済を実証分析の対象とする場合、とても有益なパッケージです。

III-3-1 Rパッケージ eurostatの利用法

(1) パッケージ eurostatのインストール

最初に、次のようにコンソール画面に入力しエンターキーを押し、eurostatをインストールします。

```
install.packages("eurostat")      # eurostatのインストール
```

さらに、library()でパッケージ eurostatを呼び出し、利用できるようにしておきます。スクリプト画面に

```
library(eurostat) # eurostatの呼び出し
```

と入力し、実行 [Run] します。これでeurostatが使えるようになります。

(2) データを探す—get_eurostat_toc(), search_eurostat()

関数get_eurostat_toc()は、eurostatのデータセットの目次をダウンロードする関数です。それではこの関数を使って目次をダウンロードしてみましょう。ここではダウンロードした目次をオブジェクト—euという名前にしています—に格納します。

```
eu <- get_eurostat_toc() # eurostatの目次取得し、オブジェクトeuに格納
```

スクリプト画面にView(eu)と入力し、実行 [Run] し、euの内容を表示させると、図3-1のような、10,004行×8列のデータフレームが表示されます。1列目のtitleはデータセット名、2列目codeはデータセットのコード番号です。このcode列の値は、選択されたデータセットをダウンロードするために使用されます。次のtype列はデータセットかフォルダかどうかを示しています。last update of data列はデータの最新の更新日、last table structure changeはテーブル構造の最新の変更時点、そして次の2つの列はデータの開始時期と終了時期を示しています。

	title	code	type	last update of data	last table structure change	data start	data end	values
1	Database by themes	data	folder	NA	NA	NA	NA	NA
2	General and regional statistics	general	folder	NA	NA	NA	NA	NA
3	European and national indicators for short-term analy...	euroind	folder	NA	NA	NA	NA	NA
4	Business and consumer surveys (source: DG ECFIN)	ei_bcs	folder	NA	NA	NA	NA	NA
5	Consumer surveys (source: DG ECFIN)	ei_bcs_cs	folder	NA	NA	NA	NA	NA
6	Consumers - monthly data	ei_bscs_m	dataset	28.01.2021	28.01.2021	1980M01	2021M01	NA
7	Consumers - quarterly data	ei_bscs_q	dataset	28.01.2021	28.01.2021	1990Q1	2021Q1	NA
8	Business surveys - NACE Rev. 2 activity (source: DG E...	ei_bcs_bs	folder	NA	NA	NA	NA	NA
9	Industry - monthly data	ei_bsin_m_r2	dataset	28.01.2021	28.01.2021	1980M01	2021M01	NA
10	Industry - quarterly data	ei_bsin_q_r2	dataset	28.01.2021	28.01.2021	1980Q1	2021Q1	NA
11	Construction - monthly data	ei_bsbu_m_r2	dataset	28.01.2021	28.01.2021	1980M01	2021M01	NA
12	Construction - quarterly data	ei_bsbu_q_r2	dataset	28.01.2021	28.01.2021	1981Q1	2021Q1	NA

図3-1 eurostatデータ一覧

注. 最初の17行のみ表示。

特定の検索ワードを指定し、データを探すためにはsearch_eurostat()関数を使用します。たとえば、search_eurostat()関数の引数に“unemployment”と入力し、失業に関連したデータセットを探してみましょう。ここではその結果をパイプ(%>%)でView()関数につなげてViewウィン

ドウで表示させています (図3-2)。

```
search_eurostat("unemployment") %>% # "unemployment" の検索
  View() # View() で検索結果を表示
```

	title	code	type	last update of data	last table structure change	data start	data end	values
1	Long-term unemployment rate (12 months and more)...	tgs00053	table	01.09.2020	01.09.2020	2008	2019	NA
2	Total unemployment rate	tps00203	table	07.10.2020	07.10.2020	2008	2019	NA
3	Long-term unemployment rate by sex	sdg_08_40	table	11.01.2021	11.01.2021	2000	2019	NA
4	Harmonised unemployment by sex	teilm010	table	28.01.2021	28.01.2021	2020M01	2020M12	NA
5	Harmonised unemployment by sex - age group 15-24	teilm011	table	28.01.2021	28.01.2021	2020M01	2020M12	NA
6	Harmonised unemployment by sex - age group 25-74	teilm012	table	28.01.2021	28.01.2021	2020M01	2020M12	NA
7	Harmonised unemployment rate by sex	teilm020	table	28.01.2021	28.01.2021	2020M01	2020M12	NA
8	Harmonised unemployment rate by sex - age group 1...	teilm021	table	28.01.2021	28.01.2021	2020M01	2020M12	NA
9	Harmonised unemployment rate by sex - age group 2...	teilm022	table	28.01.2021	28.01.2021	2020M01	2020M12	NA

図3-2 search_eurostat (“キーワード”) 検索結果の出力

search_eurostat()関数の基本的な書き方は以下のとおりです。

```
search_eurostat(pattern= “検索語”, type = “dataset”, fixed = TRUE)
```

引数にpattern, type, fixedの3つをとります。

- pattern= “ ”, “ ” の部分に探したいデータに関連したキーワードを入力します。これにより, type=引数で指定した内容に応じて, データセットやテーブルが返されます。
- type= “ ” には, dataset, folder, table, もしくは以上のすべてを指定するallのいずれかを入力します。これにより対応したEurostatのテーブルが出力されます。デフォルトではdatasetです。
- fixedにはTRUE, FALSEの論理値をとります。TRUEの場合, patternはマッチされる文字列となります。より複雑なregexマッチングが必要とされる場合, FALSEをとります。

(3) eurostatでデータをダウンロードする—get_eurostat()

データtableをダウンロードするには, get_eurostat()を使います。もっとも基本的な書式は次のようになります。

```
get_eurostat(id, time_format = “ ”, filters = “ ”, type = “ ”)
```

この関数はおもな引数として次のものをとります。

- － id= “ ” : データセットのコード名を指定します。
- － time_format= “ ” : eurostat フォーマットの時間列をどのタイプに変換するかを指定します。“date” (既定値) と入力した場合、期間の最初のデータによって日付フォーマット Date へと変換されます。“num” を指定すると、数値へと変換されます。テーブルが年次データの場合、デフォルトの日付フォーマット Date を使用するよりも、数値 num の時間変数を使用した方が便利です。
- － filters= “ ” : “none” を指定した場合、データセット全体が取得されます。一部分を取得したい場合、ここにリストを入れます (以下の例を参照してください)。
- － type= “ ” : 変数のタイプを指定します。具体的には “code” (デフォルト) もしくは “label” となります。これは変数の値を知りたいときに便利です (以下の例を参照してください)。

それでは具体的な例を使ってデータをダウンロードしてみましょう。ここではEUの「住宅価格」データを探すことにします。最初に、上で説明した search_eurostat() 関数を使って取得するデータの id を確認します。その上で、get_eurostat() 関数を利用し、イギリス、フランス、スペインの「住宅価格」データを取得してみましょう。

```
house <- search_eurostat("House",type="table")
```

View(house) (図3-3) をみると、13行めに House price index - annual data (住宅価格指数, 年次データ) が表示されています。これをダウンロードします。

	title	code	type	last update of data	last table structure change	data start	data end	values
1	Households that have internet access at home by NUT...	tgs00047	table	26.01.2021	26.01.2021	2009	2020	NA
2	Households that have broadband access by NUTS 2 re...	tgs00048	table	26.01.2021	26.01.2021	2009	2020	NA
3	Household saving rate	tec01131	table	28.01.2021	28.01.2021	2008	2019	NA
4	Household investment rate	tec00098	table	28.01.2021	28.01.2021	2008	2019	NA
5	House price index (2015 = 100) - quarterly data	teicp270	table	14.01.2021	14.01.2021	2017Q4	2020Q3	NA
6	Households with broadband access	tin00073	table	26.01.2021	26.01.2021	2009	2020	NA
7	Households that have internet access at home by NUT...	tgs00047	table	26.01.2021	26.01.2021	2009	2020	NA
8	Households that have broadband access by NUTS 2 re...	tgs00048	table	26.01.2021	26.01.2021	2009	2020	NA
9	House price index, deflated - annual data	tipsho10	table	14.01.2021	14.01.2021	2000	2019	NA
10	Household debt, consolidated including Non-profit in...	tipspd22	table	14.01.2021	14.01.2021	1995	2019	NA
11	House price index - annual data	tipsho20	table	14.01.2021	14.01.2021	2000	2019	NA
12	House price index, deflated - quarterly data	tipsho30	table	14.01.2021	14.01.2021	2005Q1	2020Q3	NA
13	House price index - quarterly data	tipsho40	table	14.01.2021	14.01.2021	2005Q1	2020Q3	NA
14	Household final consumption expenditures by durabil...	teina022	table	29.01.2021	29.01.2021	2018Q1	2020Q4	NA

図3-3 キーワード “House” の検索結果

ここではダウンロードしたデータをhouse_priceと名前をつけたオブジェクトに容れます。引数time_format=“num”で数値を指定しています。

```
house_price<-get_eurostat(id="tipsho20",time_format = "num",type = "code")
```

これにより、4つの変数、1,403の観察値がダウンロードされます。str(house_price)でデータ構造をみると、図3-4の結果が出力されます。

```
> str(house_price)
tibble [1,403 × 4] (S3: tbl_df/tbl/data.frame)
 $ unit : chr [1:1403] "INX_A_AVG" "INX_A_AVG" "INX_A_AVG" "INX_A_AVG" ...
 $ geo  : chr [1:1403] "AT" "BE" "CZ" "DE" ...
 $ time : num [1:1403] 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
 $ values: num [1:1403] 60.4 49.6 50.2 84.3 57.2 ...
```

図3-4 house_priceのデータ構造

最初の変数unitは単位、2番目の変数geoは地理情報、3番目の変数timeは年、最後の変数valuesが住宅価格指数を示しています。しかし、実はこれだけではそれぞれの変数にとる値が何を意味しているかは分かりません、少なくとも分かりづらい表示になっています。とりわけ、unit変数がコード名で表記されているため、“INX_A_AVG”等がわかりません。ダウンロードするさいの引数type = “label”としておいた方が良いでしょう。デフォルト（既定値）では図3-4のようにコード名で表示されます（引数type = “code”）。変数の値の意味を知るには変数のラベルを表示させた方が、コード名よりも便利です。

```
house_price<-get_eurostat(id="tipsho20",time_format = "num",type = "label")
```

type = “label”でダウンロードした上で、次に、以下のように入力し、変数unitがどのような値を持つか確認してみてください。

```
house_price %>% distinct(unit)
```

distinct()関数はデータフレームから、()内に指定した変数のうちユニークな、異なった値（行）だけを選択します。つまり、unit変数は1,403行から形成されますが、すべて異なった値をとるわけではありません。ユニークな値は一以下のdistinct()関数の結果（図3-5）をみると—3種類

だけです。

```
> house_price %>%
+   distinct(unit)
# A tibble: 3 x 1
  unit
  <chr>
1 Annual average index
2 Annual average rate of change
3 Percentage change (t/t-3)
```

図3-5 distinct()関数の出力結果

(4) 住宅価格上昇率のグラフを描く

いくつかのヨーロッパ経済をピックアップし、住宅価格（年平均変化率）の推移と1時点における住宅価格変化率をみてみましょう。このために次のような作業を行います。

1. 複数の国を指定し、住宅データをダウンロードする。
2. unit変数にもとづき年平均変化率を示す値（行）を抽出する。
3. ggplot2を利用し、ダウンロードしたデータの時系列グラフを作成する。
4. 地理情報を利用し、ヨーロッパ地図上に2018年の住宅価格（年平均変化率）を表示させる。

1. 複数の国を指定し、住宅データをダウンロードする。

最初に、国コードを容れたベクトルを作成しておきます—isoという名前にします。直接get_eurostat()関数で指定しても良いのですがスクリプトが見づらくなりますので、事前に、この作業を行っておきます。

```
iso <- c("FR", "DE", "ES", "IT", "SE")
```

2. unit変数にもとづき年平均変化率を示す値（行）およびサンプル国を抽出する。

```
house_price_eu <- house_price %>%
  filter(unit == "RCH_A_AVG", geo %in% iso)
```

変数unitのとり値のうち年平均変化率を表現するラベルは“RCH_A_AVG”ですのでfilter()関数の中の抽出条件の1つにunit == “RCH_A_AVG”を指定しています。また、国を示す変数はgeo

ですが、このうちベクトルisoに容れた複数の国を選択することになりますのでgeo %in% isoを利用しています。この表記が意味しているのは、「geo変数がisoベクトルの中のいずれかに等しい国コードをとる行を選択しなさい」ということを意味しています。これによりgeo変数のうちisoベクトルの中の国のいずれかに等しい行だけが選択されます。

これで2つの行の抽出条件にマッチする行だけが選択され、house_price_euというオブジェクトに格納されます。View(house_price_eu)やstr(house_price_eu)を利用し、house_price_euの内容を確認してみてください。

3. ggplot2を利用し、ダウンロードしたデータの時系列グラフを作成する。

```
ggplot(house_price_eu,aes(time,values))+ # データフレームを指定. aes()でx軸とy軸を指定.
  geom_point(aes(shape = geo))+ # geom_point()でデータを点で表示するよう指定.
  # 点の形状で国別を示すためにaes()の中でshpae
  # = geoと指定
  geom_line(aes(lty=geo))+ # さらにgeom_line()で線を重ねる. ここでも線の
  # 種類で国別に表示するため, aes()の中でlty=geo
  # と指定.
  labs(x="year",y="Housing price", # labs()を利用しx, y軸のラベル, サブタイトルを
  # 表示
  subtitle = "Housing price (Annual average rate of change %)" +
  theme_bw() # テーマを白黒に設定.
```

このスクリプトを実行 [Run] すると、以下の図3-6が出力されます。

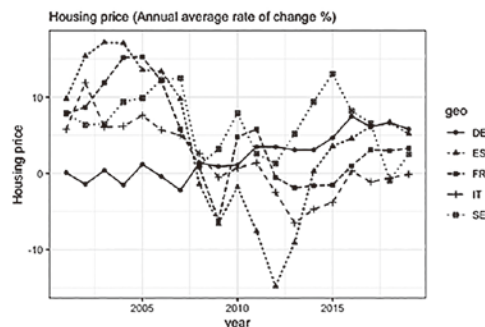


図3-6 ヨーロッパ主要経済の住宅価格の推移, 2001~2019

注. 住宅価格は年平均成長率(%). 国コードはドイツ(DE), スペイン(ES), フランス(FR), イタリア(IT), スウェーデン(SE).

4. 地理情報を利用し、ヨーロッパ地図上に2015年の住宅価格（年平均変化率）を表示させる。

図3-6において各国の住宅価格上昇率の推移を比較しましたが、今度は2015年の1時点を取り各国を比較してみましょう。このためにeurostatの提供する地理空間情報を利用し、2015年の住宅価格をヨーロッパの地図上に描きます。

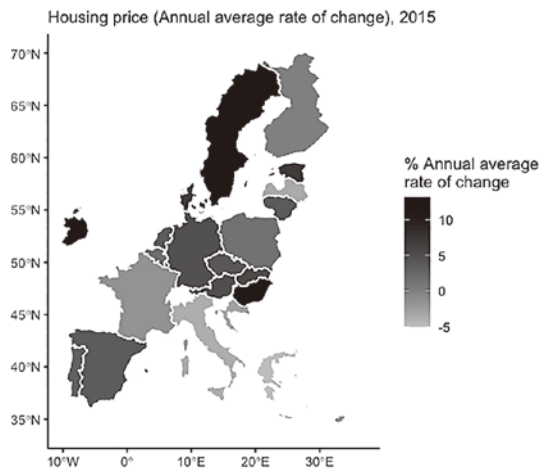


図3-7 ヨーロッパ経済の住宅価格上昇率，2015年。

出力される結果は図3-7のようになります。この図を描くための詳しい説明はIII-3-2に譲り、ここではスクリプトだけ示しておきます。手順は4つです。

- ① ヨーロッパの地図データを取得し、そのデータをshp_0という名前をつけたオブジェクトに容れます。

```
shp_0 <- get_eurostat_geospatial(resolution = 10, nuts_level = 0, year = 2016)
```

- ② 次に、上で取得した住宅価格データからヨーロッパ諸国と2015年のデータを抽出します。そしてその結果をhousing2015というオブジェクトに容れます。最初に、isoの内容は事前に上記のisoベクトルを、次のように書き換え、サンプル国を拡充しておきます。

```
iso <- c("AT", "BE", "CY", "CZ", "DE", "DK", "EL", "ES", "FI", "FR", "HR", "IE", "IT", "LT",  
"LU", "LV", "MT", "NL", "PT", "SE")
```

```
housing2015 <- house_price %>%
  filter (unit == "RCH_A_AVG", geo %in% iso, time == 2015)
```

- ③ そして次に、オブジェクト shp_0 と housing2015 を結合します。マージしたデータは地理データと住宅価格データを持つこととなります。マージしたデータは map_housing というオブジェクト名を与えています。2つのオブジェクトの結合には inner_join() 関数を使っています。

```
map_housing <- inner_join(shp_0, housing2015, by = "geo")
```

- ④ 最後に、ggplot2—geom_sf()—を利用し、ヨーロッパ地図上に、住宅価格上昇率を示します。

```
ggplot(map_housing) +
  geom_sf(aes(fill=values), color = "white", size = 0.5) +
  xlim(c(-10, 37)) +
  ylim(c(34, 70)) +
  scale_fill_continuous_tableau(palette = "Classic Gray") +
  labs (subtitle = "Housing price (Annual average rate of change), 2015",
        fill = "% Annual average\nrate of change") +
  theme_classic()
```

全体のスクリプトをまとめると、次のようになります。

ヨーロッパの地理空間データをダウンロード

```
shp_0 <- get_eurostat_geospatial(resolution = 10, nuts_level = 0, year = 2016)
```

サンプル国を指定.

```
iso <- c("AT", "BE", "CY", "CZ", "DE", "DK", "EL", "ES", "FI", "FR", "HR", "IE",
        "IT", "LT", "LU", "LV", "MT", "NL", "PT", "SE")
```

サンプル国の住宅価格（年平均変化率）データを取得.

```
housing2015 <- house_price %>%  
  filter (unit == "RCH_A_AVG", geo %in% iso,time == 2015)
```

地理空間データと住宅価格データを結合.

```
map_housing <- inner_join(shp_0,housing2015, by = "geo")
```

ヨーロッパの地図上に住宅価格を表示.

```
ggplot(map_housing) +  
  geom_sf(aes(fill=values),color = "white",size = 0.5)+  
  xlim(c(-10,37))+  
  ylim(c(34,70))+  
  scale_fill_continuous_tableau(palette = "Classic Gray")+  
  labs(subtitle = "Housing price (Annual average rate of change), 2015",  
  fill = "% Annual average\nrate of change") +  
  theme_classic()
```

このようにRパッケージeurostatを利用することで、わずか十数行のスクリプトを書くだけで図3-7のグラフを描くことができます。

Ⅲ-3-2 応用例 — 地理空間情報の利用

北欧諸国はミステリー小説の多くの傑作を生み出しています。そうした背景には社会経済的要因はもちろんのこと犯罪を身近なものを受け止める社会不安があるのかもしれませんが。そこで欧州連合統計局Eurostatが提供する犯罪や暴力に対する不安度を示す指標によって、そうした社会的不安を代理させ観察することにします。そして同指標を視覚的に示すためにEurostatの提供する地理空間情報と結合し、ヨーロッパ地図上に表示します。主な手順は次のようになります。

- (1) 地理空間情報（シェイプ・ファイル）をダウンロードし、ヨーロッパ地図を描く。
- (2) 犯罪・暴力・破壊行為に対する不安度を示す指標をダウンロードする。
- (3) 1と2のファイルを結合し、不安度指標を表示した北欧地図を描く。

(1) 地理空間情報のダウンロードとggplot2による描画—sfパッケージ

地理空間データの分析用のRパッケージとしてsfパッケージがあります。sfパッケージは、simple featuresというGIS (geographic information system) データ規格をRで扱うためのパッケージです。Eurostatの提供する地理空間情報データを扱うために、最初に、このパッケージをインストールしておきます。コンソール画面に次のように入力し、エンターキーを押してください。

```
install.packages("sf")
```

インストールを終えたら、スクリプト画面に次のように入力し、実行 [Run] します。

```
library(sf)
```

このパッケージを利用することによって地理空間データのさまざまな処理が可能となります。すでに幾度も登場しているggplot2も、バージョン3.0.0以後sfクラスのデータを扱えるようになりました。sfクラスのオブジェクトのプロットには、geom_sf()という関数を使います。

Eurostatの地理空間情報は、国レベルにとどまらず、それ以上に詳細な行政単位レベルの情報も提供しています。1970年代初頭、Eurostatは、欧州連合 (EU) の地域統計を作成するために、EUの領土を分割するための統一された、一貫したシステムとしてNUTS (Nomenclature of territorial units) 分類¹を設定しました。イタリアを例にとれば、NUTSレベル0はイタリアという国そのものの、NUTSレベル1はNord-Ovest (北西部)、Sud (南部) といった地域、さらにNUTSレベル2はPiemonte (ピエモンテ)、Liguria (リグーリア)、Lombardia (ロンバルディア) といった州、もっとも細かいレベルのNUTSレベル3はTorino (トリノ)、Genova (ジェノバ)、Milano (ミラノ) 等の都市空間を示します。これにより、たとえば、個人世帯の可処分所得の地理的分布を調べたいとき、NUTSレベル2すなわち各国の州レベルまで調べること (グラフ表示) も可能です。

それでは最初に、NUTSレベル1と2のヨーロッパ地図を描いてみましょう。地図を描くためにはシェイプファイルshapefileという図形情報と属性情報を入れたファイルをダウンロードする必要があります。RパッケージeurostatはGISGOからヨーロッパの地理情報を取得する関数get_eurostat_geospatial()を用意しています。この関数の基本的な用法は次のようになります。

¹ UTSについては<https://ec.europa.eu/eurostat/web/nuts/history>を参照してください。

```
get_eurostat_geospatial(  
  output_class = "sf",  
  resolution =  
  nuts_level = ,  
  year = 2016 )
```

以下はこの関数がとる引数の概要です。

- `output_class = "" : ""` に `sf`, `df`, もしくは `spdf` のいずれかの文字列を指定することで返されるオブジェクトのクラスを選択します。 `sf` は `simple features`, `df` は `data_frame`, `spdf` は `SpatialPolygonDataFrame` を表現します。 デフォルト (既定値) は `"sf"` です。
- `resolution = :` 地理空間データの解像度を指定します。 利用可能な解像度は `60` (1 : 6,000万), `20` (1 : 2,000万), `10` (1 : 1,000万), `03` (1 : 3,000万), `01` (1 : 1,000万) のいずれかです。
- `nuts_level = :` NUTS分類のレベルをしていします。 `0`, `1`, `2`, `3`, `all` のいずれかをとります。
- `year = :` NUTSのリリース年を指定します。 具体的には, `2003`, `2006`, `2010`, `2013`, `2016` のいずれかを入力します。

それでは地理空間情報をダウンロードしてみましょう。ここではNUTSレベルの異なる2つのオブジェクトを作成します。

```
shp0 <- get_eurostat_geospatial(output_class = "sf", resolution = 10, nuts_level = 0, year = 2016)  
# NUTS分類のレベル0で地図情報を取得し、オブジェクトshp0に容れる
```

```
shp2<- get_eurostat_geospatial(output_class = "sf", resolution = 10, nuts_level = 2, year = 2016)  
# NUTS分類のレベル2で地図情報を取得し、オブジェクトshp2に容れる
```

オブジェクト—`shp0`, `shp2`—は、引数`output_class="sf"`で指定していますので`sf`クラスです。`geom_sf()`は`sf`クラスのオブジェクトを扱いますから、この指定が必要となります (ただし、デフォルトでも`sf`クラスですから、指定しなくとも結果は`sf`クラスのオブジェクトとなります)。

NUTSレベル0は国単位を意味します。NUTSレベル2は、上述のように、イタリアで言えばロンバルディアといった州レベルの行政単位まで描かれます。

ヨーロッパの地図情報が取得できましたので、ggplot2を利用しヨーロッパ地図を描いてみましょう。ggplot2にはsfクラスのオブジェクトを描くためにgeom_sf()が開発されています。グラフの作成はこれまで紹介したggplot2の利用方法に同じです。基本はggplot(オブジェクト名)+geom_sf()です。

```
ggplot(shp0) +           # オブジェクト名を入力
geom_sf().              # sfクラスのオブジェクトを指定
```

これだけでヨーロッパ地図がプロット画面に出力されます。しかし、かなり広いヨーロッパ地域が表示されてしまいます。そこでx軸、y軸の範囲を限定し、わたしたちが通常目にするヨーロッパを描くとしましょう。そのためにはxlim()を利用し、x軸の範囲を決定します。xlim(c(西に-10度, 東に37度))のように東西の経度を入力します。y軸の範囲も同様に、ylim(c(),)を利用し、緯度を入力します。あるいはggplot2のcoord_map()という関数を使っても良いでしょう。引数xlim=()で経度、ylim=()で緯度の範囲を指定できます。たとえば、coord_map(xlim = c(-30, 45), ylim = c(30, 75))と入力することによってヨーロッパ地域を指定することができます。

```
nuts0 <- ggplot(shp0) +
geom_sf()+
xlim(c(-10,37))+
ylim(c(34,65))
```

わずか数行の入力で図3-8のようなヨーロッパ地図を描くことができます。あわせてNUTレベル2の地図も描いてみましょう。上述のグラフ作成との違いはggplot()にsfオブジェクト名shp2を指定するだけです。

```
nuts2 <- ggplot(shp2) +
geom_sf()+
xlim(c(-10,37))+
ylim(c(34,65))
```

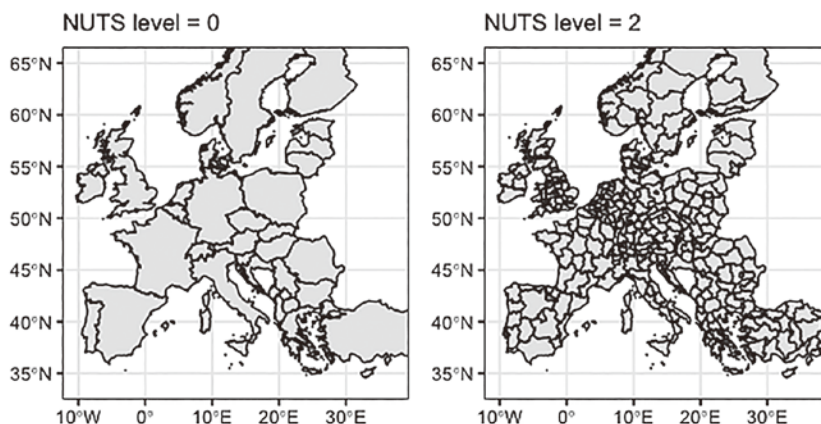


図3-8 eurostat地理空間情報を利用したヨーロッパ地図の作成

注. グラフのレイアウトにあたってはRパッケージpachwork利用し, nuts0+nuts2で横に並べています.

次に、北欧地域だけを描くには、x軸範囲とy軸の範囲を以下のように限定すれば、北欧地域だけの地図を描くことができます。

```
n_euro <- ggplot(shp0)+
  geom_sf()+
  xlim(c(0,32))+      # 北欧地域の範囲を経度で指定
  ylim(c(54,70))+    # 北欧地域の範囲を緯度で指定.
```

(2) 犯罪・暴力・破壊行為に対する不安度データのダウンロード

次に、犯罪や暴力に対する不安度データを地図上に表示するために、犯罪に関連するデータを探します。これはすでに紹介したsearch_eurostat()関数を利用します。“crime (犯罪)”をキーワードにデータを探してみましょう。

```
search_eurostat("crime",type="table") %>%
  View()
```

上のスクリプトを実行 [Run] すると、search_eurostat()の結果をView()関数に渡しているため、crimeに関連した検索結果が表示されます。関連するデータテーブルは1つだけでsdg_16_20というファイルです。このデータは地域における犯罪・暴力・破壊行為を報告した人の比率を示しています。そうした点では人々の犯罪や暴力に対する不安度を代理する指標として良いでしょ

う。

次に、データ・テーブル `sdg_16_20` をダウンロードし、結果を `crime` というオブジェクトに容れます。

```
crime<-get_eurostat(id="sdg_16_20",time_format = "num")
```

これによりヨーロッパ諸国の犯罪・暴力・破壊行為の報告者比率データを取得できます。ここから第1に、北欧4カ国とバルト3国の7カ国だけを抽出します。このために変数の `geo` の値が `"DK"`, `"SE"`, `"NO"`, `"FI"`, `"LV"`, `"EE"`, `"LT"` のどれかをとる行を抽出します。

```
filter(geo %in% c("DK","SE","NO","FI","LV","EE","LT"))
```

ここで `"geo %in% c()"` は「`geo`変数が `c` (国コード) の中のいずれかに等しい国コードをとる行を選択しなさい」を意味しています。また、このデータは、`incgrp` という変数を持っていますが、これは `income group` (所得グループ) を示し、所得階層別情報を提供しています。しかし、ここでは所得階層別情報は使いませんので `incgrp` 変数のうち値が `TOTAL` の行だけを抽出します。そこで上の抽出条件にくわえて、次の抽出条件も入力します。

```
filter(incgrp == "TOTAL")
```

2つの抽出条件をあわせると、次のようになります。

```
filter(geo %in% c("DK","SE","NO","FI","LV","EE","LT"), incgrp == "TOTAL")
```

さらに、時間も2017年に指定し — `time == 2017` —、最終的に次のように入力します。

```
ne_crime <- crime %>%
  filter(time == 2017, geo %in% c("DK","SE","NO","FI","LV","EE","LT"), incgrp == "TOTAL")
```

`filter()` の基本コードは `filter(データフレーム名, 抽出条件)` ですが、ここではパイプ (`%>%`) を利用し、データフレーム `crime` を関数 `filter()` に渡していますのでデータフレーム名が省略されています。

以上で、北欧地域の犯罪や暴力に対する不安度を示す指標が取得できました。これを地図上に表示させるために、地図情報の入ったshp0と結合します。

(3) シェイプファイルshp0と犯罪データファイルne_crimeの結合

ここではヨーロッパの地図情報の入った“shp0”ファイルと、北欧地域の犯罪不安度情報の入った“ne_crime”ファイルを結合します。データフレームの結合にあたってはinner_join()関数を使います。ジョイン_join()は2つのデータテーブルの変数を結合します。inner_join()はby=“キー変数”にしたがって、キーの値が等しい観察値が結合されます。したがってキー変数にマッチしない観察値は除外されます。

inner_join()のもっとも簡単な書き方は次のようになります。2つのデータフレームa, bがある場合、

```
inner_join(データフレーム a, データフレーム b, by = “キー変数”)
```

それではinner_join()を利用し、shp0とne_crimeを、両データフレームに共通のgeo変数を基準に結合します。その上でshp_neというオブジェクトに容れます。

```
shp_ne <- inner_join(shp_0, ne_crime, by = “geo”)
```

これで不安度指標と地理空間情報の入った、74行×17列のデータフレームが作成されました。残りの作業はggplot2を使って、図3-9のような不安度指標を色の濃淡で示した北欧地図を描くだけです。

shp_neはsfクラスのオブジェクトです。このため描画にあたってはgeom_sf()関数に渡すだけです。この場合、何よって色分けするかをaes()で指定します。この例では、犯罪・暴力・破壊行為の報告者比率を示すvalues変数の値の大きさを色分けします。このためにfill = valuesとします。

具体的には、まず最初に、ggplot()にデータフレーム名を指定します。

```
ggplot(shp_ne)
```

次に、geom_sf()関数を使ってデータを地図に描きますが、変数vlauesを地図上の色の濃淡で表現するために、aes(fill =)で指定します。

```
geom_sf(aes(fill=values),color = "white",size = 0.5)
```

value変数は犯罪・暴力・破壊行為の報告者比率です。さらに、color = "" でボーダー（国境）をwhite（白）で描き、その線の太さをsize = で指定しています。これで基本的な作業は終わりです。実行すると、犯罪・暴力・破壊行為の報告者比率がデフォルト（既定値）の色の濃淡によって描かれた北欧地図が出力されます。

この例ではさらに、scale_fill_continuous_tableau()を使って犯罪・暴力・破壊行為の報告者比率を描く色の濃淡をグレイに変更します。この関数によって連続したカラースケールを指定することができます。引数palette = "" 部分に“Blue”, “Red” のように入力します。ここでは白黒印刷にあわせ、グレイの色を採用しています。

```
scale_fill_continuous_tableau(palette = "Classic Gray")+
```

次に、labs()を使ってタイトルをつけます。

```
labs(title = "The share of the population who reported \nthat they face the problem of crime,
violence or vandalism, 2017", fill = "% the respondent \nfeels crime") +
```

英文タイトルの中のthatの前に“\n”が挿入されていることに気づいたと思います。これはその部分で文書の改行を命令する記号です。また、fill = "" によって色の濃淡の表示するバーのタイトルを指定しています。

最後に、背景のテーマを設定します。

```
theme_classic()
```

以上のスクリプトをまとめると、つぎのようになります。

```
ggplot(shp_ne) +
  geom_sf(aes(fill=values),color = "white",size = 0.5)+
  scale_fill_continuous_tableau(palette = "Classic Gray")+
  labs(title = "The share of the population who reported \nthat they face the problem of crime,
violence or vandalism, 2017", fill = "% the respondent \nfeels crime") +
  theme_classic()
```

これを実行 [Run] すると、図3-9のようなグラフが描かれます。

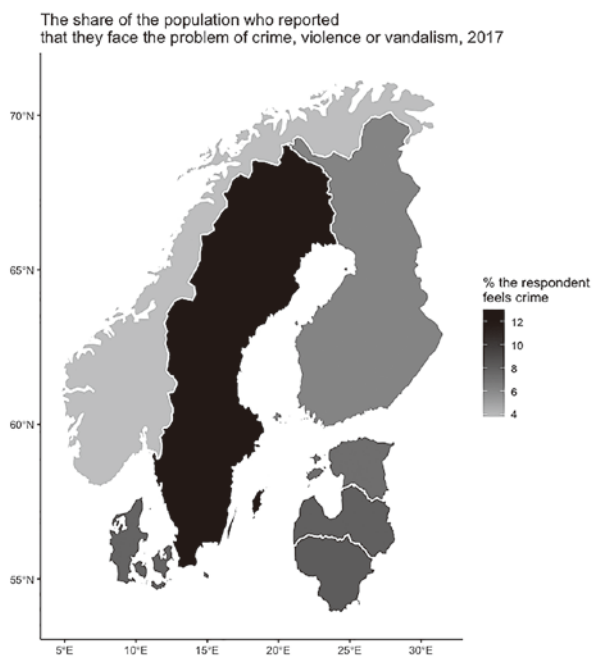


図3-9 犯罪や暴力、破壊行為の問題に直面していると回答した人の割合 (%)

注. デンマーク、フィンランド、ノルウェーおよびスウェーデン北欧4カ国、およびエストニア、ラトビアおよびリトアニアのバルト3国、2017年データ。回答は総人口に占める割合 (%)。

見られるように、サンプル経済の中では編集者ミカエル²の活躍するスウェーデンがもっとも高く、これにリガの犬³が蠢くラトビア等のバルト3国が続きます。バルト海を挟む諸国において犯罪や暴力、破壊行為の問題に直面している人々が多そうです。こうした社会的不安が犯罪小

² スティグ・ラーソン著『ミレニアム』(早川書房)シリーズの主人公。

³ ヴァルランダーシリーズ、ヘニング・マンケル著『リガの犬たち』(創元社)。

説の創造に寄与しているのかもしれませんが。

(4) 地理空間情報— NUTSレベル3の利用

さらに、スウェーデンに注目し、こうした犯罪の背景—犯罪や暴力、破壊行為—を地域レベルでみてみましょう。これまで地理空間をNUTSレベル0としてきました。つまり、国単位で情報を扱っていました。次に、北欧社会の中でも、犯罪や暴力、破壊行為の問題にもっとも強く直面しているスウェーデンに注目し、NUTSレベルを3までブレイクダウンし、より詳細な犯罪情報を地図上に描いてみます。

上で利用したデータはNUTSレベル2の情報を提供していませんが、犯罪に関連したNUTSレベル2、3のデータとしては「犯罪記録件数」(Crimes recorded by the police by NUTS 3 regions)があります。このデータを`get_eurostat()`を利用し、ダウンロードします。これを`crime`という名前をつけたオブジェクトに容れます。本データのidは`crim_gen_reg`です。

```
crime <- get_eurostat(id="crim_gen_reg")
```

最初に、スウェーデンをピックアップします。このためgeo変数に`filter()`をかけますが、抽出条件を`geo == "SE"`とすると、スウェーデンの国内の行政単位のデータ—たとえば、SE1、SE2という記号で表示—は抽出条件に合致しないため、削除されてしまいます。そこで最初に、`str.detect()`関数を使って“SE”という文字列を持つgeo変数の行を探します。そしてそれと一致したgeo変数の行を抽出します。

```
filter(str.detect(geo,"SE"))
```

`str.detect(geo, "SE")`はgeo変数が“SE”という文字列を含む場合、真TRUEを返します。これに`filter()`を適用することによって、`str.detect()`関数がTRUE返す行を抽出します。

データは2009年と2010年のみですが、ここでは2010年を採用します。

```
filter(Date == "2010-01-01")
```

以上、必要な行の抽出が終わりです。これまでのスクリプトをまとめると、以下のようになります。抽出結果を`crime_swe`という名前をつけたオブジェクトに容れてあります。

```
crime_swe <- crime %>%  
  filter(str_detect(geo, "SE"), time == "2009-01-01")
```

次に、NUTSレベル3のデータをダウンロードし、shp3という名前のオブジェクトに容れます。

```
shp3 <- get_eurostat_geospatial(resolution = 10, nuts_level = 3, year = 2016)  
# NUTS分類のレベル3で地図情報を取得し、オブジェクトshp3に容れる
```

それでは、crime_sweとNUTSレベル3のシェイプファイルをinner_join()関数を使って結合します。そして結合したファイルをshp3_crime_sweという名前のオブジェクトに容れます。

```
shp3_crime_swe <- inner_join(shp3,crime_swe, by = c("NUTS_ID"="geo"))
```

by = “キー変数” の部分がこれまでと異なります。2つのファイルを結合するさいに、異なる変数を利用する場合、上述のようにby = c(“NUTS_ID”=“geo”)とします。NUTS_IDはshp_2の中の変数、geoはcrime_sweの中の変数です。

これで地図を作成するデータが出来上がりました。あとは図3-9を作成した場合と同じようにggplot2を利用します。スクリプトは次のようになります。

```
ggplot(shp3_crime_swe) +  
  geom_sf(aes(fill=values),color = "white",size = 0.5)+  
  xlim(c(7,25))+  
  scale_fill_continuous_tableau(palette = "Classic Gray")+  
  labs(title = "Crimes recorded by the police,2010",  
fill = "% the number of \n crimes recorded by the police")+  
  theme_classic()
```

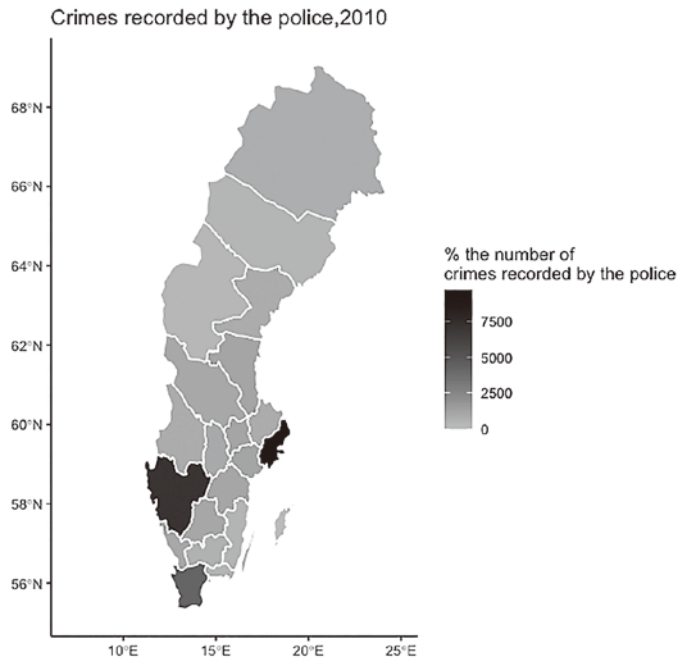


図3-10 スウェーデンにおける犯罪記録件数, 2010

ここでは単純に犯罪件数を表示させているだけですから、スウェーデンの中でも大都市圏—Stockholms県、Västra Götalands県やSkåne県等—で犯罪件数が多いことが理解されます。また図3-10から理解されるように、犯罪件数には地理的な関連があるかもしれません。犯罪が多発する県の隣接する県でも多そうです。これは空間的なデータが相関していることを示唆しており、空間情報を分析するさいに注意を要する点です。

【参考文献】

Chris Brunsdon, Lex Comber著 (湯谷啓明・工藤和奏・市川太祐訳)『Rによる地理空間データ解析入門』共立出版, 2018年.

スティーグ・ラーソン著『ミレニアム』早川書房, 2008~2019年.

ヘニング・マンケル著『リガの犬たち』創元社, 2003年.