Geospatial Health Informatics

In [1]:
```
using Pkg
Pkg.activate(".")
```

```
  Activating project at `~/Knowledgebase/WORKSHOPS/warsaw_juliahealth_2024`
```

Dependency Set-Up

Loading packages needed

In [2]:
```
using CairoMakie
using Chain
using CSV
using DataFrames
using GeoDataFrames
using GeoInterfaceMakie
using GeoMakie
using StatsBase

import IPUMS:
  load_ipums_extract,
  load_ipums_nhgis,
  parse_ddi
```

```
[ Info: Precompiling GeoMakie [db073c08-6b98-4ee5-b6a4-5efafb3259c6]
[ Info: Skipping precompilation since __precompile__(false). Importing GeoMakie [db073c08-6b98-4ee5-b6a4-5efafb3259c6].
```

Loading Required Data

Load data dictionary (xml) and IPUMS International data file (dat):

In [3]:
```
ddi = parse_ddi("poland_data/ipumsi_00001.xml");
df = load_ipums_extract(ddi, "poland_data/ipumsi_00001.dat");
```

Basic Exploration of IPUMS Data

Examining Metadata

By `DataFrame`:

In [4]:
```
md_df = metadata(df)
for md in keys(md_df)
  println("$(md):\n----------------\n\n $(md_df[md])\n\n")
end
```

extract_notes:

 User-provided description: Most recent Poland data to explore different outcomes and demographics across geospatial boundaries

citation:

 . Steven Ruggles, Lara Cleveland, Rodrigo Lovaton, Sula Sarkar, Matthew Sobek, Derek Burk, Dan Ehrlich, Quinn Heimann, Jane Lee. Integrated Public Use Microdata Series, International: Version 7.5 [dataset]. Minneapolis, MN: IPUMS, 2024. https://doi.org/10.1 [dataset]. Minneapolis, MN: IPUMS, 2024. https://doi.org/10.18128/D020.V7.5

Researchers should also acknowledge the statistical agency that originally produced the data.

The licensing agreement for use of IPUMS International data requires that users supply IPUMS International with the title and full citation for any publications, research reports, or educational materials making use of the data or documentation.

Copies of such materials are also gratefully received at ipums@umn.edu.

Printed matter should be sent to:
IPUMS International
Minnesota Population Center
University of Minnesota
50 Willey Hall
225 19th Avenue South
Minneapolis, MN 55455

extract_date:

 2024-06-27

conditions:

 An adapted version of the dataset, harmonized for international comparability, is available from IPUMS International (https://international.ipums.org/international/) under the following conditions:

IPUMS International distributes integrated microdata of individuals and households only by agreement of collaborating national statistical offices and under the strictest of confidence. Before data may be distributed to an individual researcher, an electronic license agreement must be signed and approved. To gain access to the data, a researcher must agree to the following:

(1) Implement security measures to prevent unauthorized access to census microdata. Under IPUMS International agreements with collaborating agencies, redistribution of the data to third parties is prohibited.

(2) Use the microdata for the exclusive purposes of scholarly research and education. Researchers must explicitly agree to not use microdata acquired for any commercial or income-generating venture.

(3) Maintain the confidentiality of persons, households, and other entities. Any attempt to ascertain the identity of persons or households from the microdata is prohibited. Alleging that a person or household has been identified is also prohibited.

(4) Report all publications based on these data to IPUMS International, which will in turn pass the information on to the relevant national statistical agencies.

Once a project is approved, a password is issued and data may be acquired through the Internet. Penalties for violating the license include: revocation of the license, recall of all microdata acquired, filing of a motion of censure to the appropriate professional organizations, and civil prosecution under the relevant national or international statutes.

These safeguards mirror the principles from the Joint ECE/Eurostat Work Session on Statistical Data Confidentiality. Employees of the Minnesota Population Center who work with the census microdata to produce the harmonized database also sign agreements to respect the confidentiality of the data.

```
ipums_project:
----------------

 IPUMS International
```

By each column of the `DataFrame`:

In [5]:
```
for colname in names(df)
  println("$(colname):\n----------------\n\n $(colmetadata(df, colname, "description"))\n\n")
end
```

COUNTRY:

 COUNTRY gives the country from which the sample was drawn. The codes assigned to each country are those u
sed by the UN Statistics Division and the ISO (International Organization for Standardization).

YEAR:

 YEAR gives the year in which the census or survey was taken. For samples that span years, the midpoint or
first year of the interval is reported.

SAMPLE:

 SAMPLE identifies the IPUMS sample from which the case is drawn. Each sample receives a unique 9-digit cod
e. The code is structured as follows:

The first 3 digits are the ISO/UN codes used in COUNTRY

The next 4 digits are the year of the census/survey

The final 2 digits identify the sample within the year. For the last two digits, censuses or large census-
like surveys have a value "0" (e.g, 01) in the second-to-last digit, household surveys have a value of "2"
(e.g., 21), and employment surveys have a value of "4" (e.g., 41).

SERIAL:

 SERIAL is an identifying number unique to each household in a given sample. All person records are assigne
d the same serial number as the household record that they follow. (Person records also have their own uniq
ue identifiers -- see PERNUM.) The combination of SAMPLE and SERIAL provides a unique identifier for every
household in the IPUMS-International database; SAMPLE, SERIAL and PERNUM uniquely identify every person in
the database.

SERIAL can be used to identify dwellings in some samples. In these samples, the first 7 digits of SERIAL p
rovide the dwelling number common to all households that were sampled from the same structure. The last thr
ee digits give the sequence of the household within the dwelling. The following is a list of samples in whi
ch dwellings can be inferred:
Chile 1970, 1992, 2002Colombia 1993, 2005Costa Rica 1984, 2000Cuba 2002Dominican Republic 1981, 2002, 2010E
cuador 1990, 2001Germany 1971Hungary 1980, 1990, 2001Jamaica 1982, 1991, 2001Malaysia 1970, 1991, 2000Mexic
o 1995, 1990, 2000, 2005Nigeria 2006Panama 2000Peru 1993, 2007Portugal 1981, 1991, 2001Spain 1991Uruguay 20
11Venezuela 1990, 2001Vietnam 1989In all other samples, the last 3 digits are always zeroes.

SERIAL was constructed for IPUMS-International, and has no relation to the serial number in the original da
tasets.

The U.S. 1900 sample and 1880 10% sample have multi-household dwellings that can be identified using the la
st 3 digits of SERIAL.

HHWT:

 HHWT indicates the number of households in the population represented by the household in the sample.

For the samples that are truly weighted (see the comparability discussion), HHWT must be used to yield accu
rate household-level statistics.

NOTE: HHWT has 2 implied decimal places. That is, the last two digits of the eight-digit variable are decim
al digits, but there is no actual decimal in the data.

ENUTS1_2013:

 ENUTS1_2013 is the third regular amendment to the annexes to the ENUTS1 classification. ENUTS1_2013 was en
forced on 31 December 2013 and was applied from 1 January 2015. ENUTS1_2013 identifies the Nomenclature of
Territorial Units for Statistics (NUTS) within Europe in which the household was enumerated. NUTS1 is the f

irst level territorial units within countries. NUTS is a standard administrative division of the European U
nion, and was developed by the EU. The European Free Trade Association extends the NUTS system to several a
dditional countries outside of the EU (e.g. Turkey), and they are also incorporated into this variable.

The first 3-digits of ENUTS1_2013 variable provide the COUNTRY code followed by the 2-digit NUTS1 code. The
labels include the standard code for the NUTS1 (2013) system and the name of the NUTS1 region, separated by
a slash.

Smaller sub-national units are available for most countries in ENUTS2_2013 and ENUTS3_2013.
The full set of geography variables for the countries can be found in the IPUMS International Geography var
iables list. For cross-national geographic analysis on the first and second major administrative level refe
r to GEOLEV1, and GEOLEV2. More information on IPUMS-International geography can be found here.

ENUTS2_2013:

 ENUTS2_2013 is the third regular amendment to the annexes to the ENUTS2 classification. ENUTS2_2013 was en
forced on 31 December 2013 and was applied from 1 January 2015. ENUTS2_2013 identifies the Nomenclature of
Territorial Units for Statistics (NUTS) within Europe in which the household was enumerated. NUTS2 is the s
econd level territorial units within countries. NUTS is a standard administrative division of the European
Union, and was developed by the EU. The European Free Trade Association extends the NUTS system to several
additional countries outside of the EU (e.g. Turkey), and they are also incorporated into this variable.

The first 3-digits of ENUTS2_2013 variable provide the COUNTRY code followed by the 3-digit NUTS2 code. The
labels include the standard code for the NUTS1 (2013) system and the name of the NUTS2 region, separated by
a slash.

Smaller sub-national units are available for most countries in ENUTS3_2013. Larger sub-national units are a
vailable for most countries in ENUTS1_2013. The full set of geography variables for the countries can be fo
und in the IPUMS International Geography variables list. For cross-national geographic analysis on the fir
st and second major administrative level refer to GEOLEV1, and GEOLEV2. More information on IPUMS-Internati
onal geography can be found here.

GEO1_PL2011:

 GEO1_PL2011 identifies the household's voivodship within Poland in 2011. Voivodships are the first level a
dministrative units of the country. A GIS map (in shapefile format), corresponding to GEO1_PL2011 can be d
ownloaded from the GIS Boundary files page in the IPUMS International web site.

The full set of geography variables for Poland can be found in the IPUMS International Geography variable
s list. For cross-national geographic analysis on the first and second major administrative level of any c
ountry refer to GEOLEV1, and GEOLEV2. More information on IPUMS-International geography can be found here.

PERNUM:

 PERNUM numbers all persons within each household consecutively (starting with "1" for the first person rec
ord of each household). When combined with SAMPLE and SERIAL, PERNUM uniquely identifies each person in the
IPUMS-International database.

PERWT:

 PERWT indicates the number of persons in the actual population represented by the person in the sample.

For the samples that are truly weighted (see the comparability discussion), PERWT must be used to yield acc
urate statistics for the population.

NOTE: PERWT has 2 implied decimal places. That is, the last two digits of the eight-digit variable are dec
imal digits, but there is no actual decimal in the data.

AGE:

 AGE gives age in years as of the person's last birthday prior to or on the day of enumeration.

```
EMARST:
----------------

 EMARST describes for the European samples the person's current marital status according to law or custom.
Individuals who remarried should report the status relevant to their most recent marriage.  European census
instructions generally limit marital status to legal unions, but there are exceptions.

EMARST has been classified according to the recommendations given by the Conference of European Statisticia
ns for the 2010 Population and Housing Censuses.


EDUCPL:
----------------

 EDUCPL indicates the person's educational attainment in Poland in terms of the level of schooling complete
d.
```

Exploring Geospatial Data

Load Shapefile for `ENUTS2`:

In [6]:
```
geodf = load_ipums_nhgis("shapefiles/ENUTS2_2013.shp").geodataframe;
filter!(x -> x.CNTRY_NAME == "Poland", geodf);
dropmissing!(geodf, :geometry)
geodf[!, :ENUTS2] = parse.(Int64, geodf[!, :ENUTS2])
```

Out[6]:
```
16-element Vector{Int64}:
 616021
 616022
 616041
 616042
 616043
 616051
 616052
 616061
 616062
 616063
 616071
 616072
 616081
 616082
 616084
 616091
```

In [7]:
```
geodf
```

Out[7]: 16×6 DataFrame

Row	geometry	CNTRY_NAME	CNTRY_CODE	BPL_CODE	ENUTS2	ADMIN_NAME
	IGeometry	String	String	Float64	Int64	String?
1	Geometry: wkbPolygon	Poland	616	41050.0	616021	PL21 / Małopolskie
2	Geometry: wkbPolygon	Poland	616	41050.0	616022	PL22 / Śląskie
3	Geometry: wkbPolygon	Poland	616	41050.0	616041	PL41 / Wielkopolskie
4	Geometry: wkbMultiPolygon	Poland	616	41050.0	616042	PL42 / Zachodniopomorskie
5	Geometry: wkbPolygon	Poland	616	41050.0	616043	PL43 / Lubuskie
6	Geometry: wkbPolygon	Poland	616	41050.0	616051	PL51 / Dolnośląskie
7	Geometry: wkbPolygon	Poland	616	41050.0	616052	PL52 / Opolskie
8	Geometry: wkbPolygon	Poland	616	41050.0	616061	PL61 / Kujawsko-pomorskie
9	Geometry: wkbPolygon	Poland	616	41050.0	616062	PL62 / Warmińsko-mazurskie
10	Geometry: wkbMultiPolygon	Poland	616	41050.0	616063	PL63 / Pomorskie
11	Geometry: wkbPolygon	Poland	616	41050.0	616071	PL71 / Łódzkie
12	Geometry: wkbPolygon	Poland	616	41050.0	616072	PL72 / Świętokrzyskie
13	Geometry: wkbPolygon	Poland	616	41050.0	616081	PL81 / Lubelskie
14	Geometry: wkbPolygon	Poland	616	41050.0	616082	PL82 / Podkarpackie
15	Geometry: wkbPolygon	Poland	616	41050.0	616084	PL84 / Podlaskie
16	Geometry: wkbPolygon	Poland	616	41050.0	616091	PL91 / Warszawski stołeczny + PL92 / Mazowieckie regionalny

Visualizing Geospatial Data

In [8]:
```
fig = Figure(
  size = (1200, 1400),
  fontsize = 20
);

ax = CairoMakie.Axis(
  fig[1, 1],
);
```

In [9]:
```
poly!(ax, geodf.geometry, color = 1:16, colormap = :Reds, strokecolor = :black, strokewidth = 3);
```

In [10]:
```
Label(fig[:, :, Top()], "Voivodeships of Poland", fontsize = 50)
```

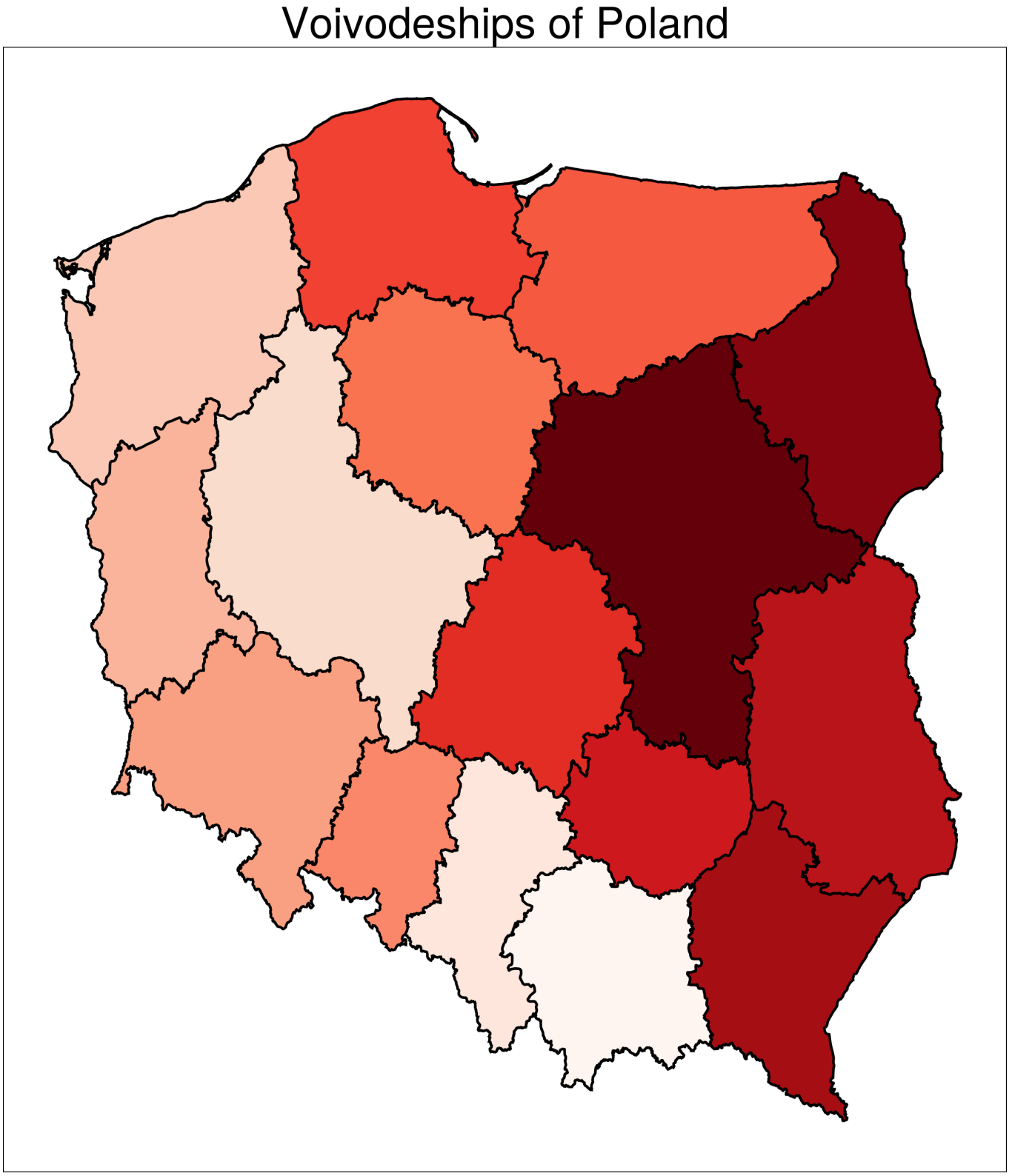
Out[10]: Label()

In [11]:
```
hidedecorations!(ax)
```

Out[11]: false

In [12]:
```
fig
```

Out[12]:



Rough Exploration of Educational Attainment Level

Find education attainment levels across vovoideships:

In [ ]:
```
edu_df = @chain df begin
  groupby(_[:, [:ENUTS2_2013, :EDUCPL]], [:ENUTS2_2013, :EDUCPL])
  combine(nrow => :Count)
end
```

Creating Grouping for Educational Levels

In [14]:
```
primary = [12, 20]
secondary = [40, 41, 42, 43, 50]
university = [70, 71, 72, 73]
```

Out[14]:
```
4-element Vector{Int64}:
 70
 71
 72
 73
```

In [15]:
```
colmetadata(edu_df, :EDUCPL, "description")
```

Out[15]:
```
"EDUCPL indicates the person's educational attainment in Poland in terms of the level of schooling complet
ed."
```

Assigning labels for level of educational attainment:

In [16]:
```
edu_df.EDUCPL = convert(Vector{Any}, edu_df.EDUCPL)
replace!(x -> in(x, primary) ? "PRIMARY" : x, edu_df.EDUCPL)
replace!(x -> in(x, secondary) ? "SECONDARY" : x, edu_df.EDUCPL)
replace!(x -> in(x, university) ? "UNIVERSITY" : x, edu_df.EDUCPL)
```

Additional Processing of Data

Getting raw counts of educational attainment across voivodeships:

In [ ]:
```
edu_counts = @chain edu_df begin
  filter!(row -> !isa(row.EDUCPL, Real), _)
  groupby([:ENUTS2_2013, :EDUCPL])
  combine(:Count => sum => :Count)
  groupby([:EDUCPL])
end
```

Plotting and Normalization of Data

Next, normalize this and then create several heatmaps for differing education levels

In [18]:
```
fig = Figure(
  size = (1200, 1400),
  fontsize = 20
);

axs = [Axis(fig[x, y]) for x in 1:2 for y in 1:2]
colors = [:Purples, :Greens, :Blues, :Reds]
poly!(axs[1], geodf.geometry, color = :white, strokecolor = :black, strokewidth = 3);

hidedecorations!(axs[1])
axs[1].title = "Voivodeships of Poland"
```

Out[18]:
```
"Voivodeships of Poland"
```

In [19]:
```
Label(fig[:, :, Top()], "Normalized Education Counts across Poland", fontsize = 50, padding = (0, 0, 30, 0
```

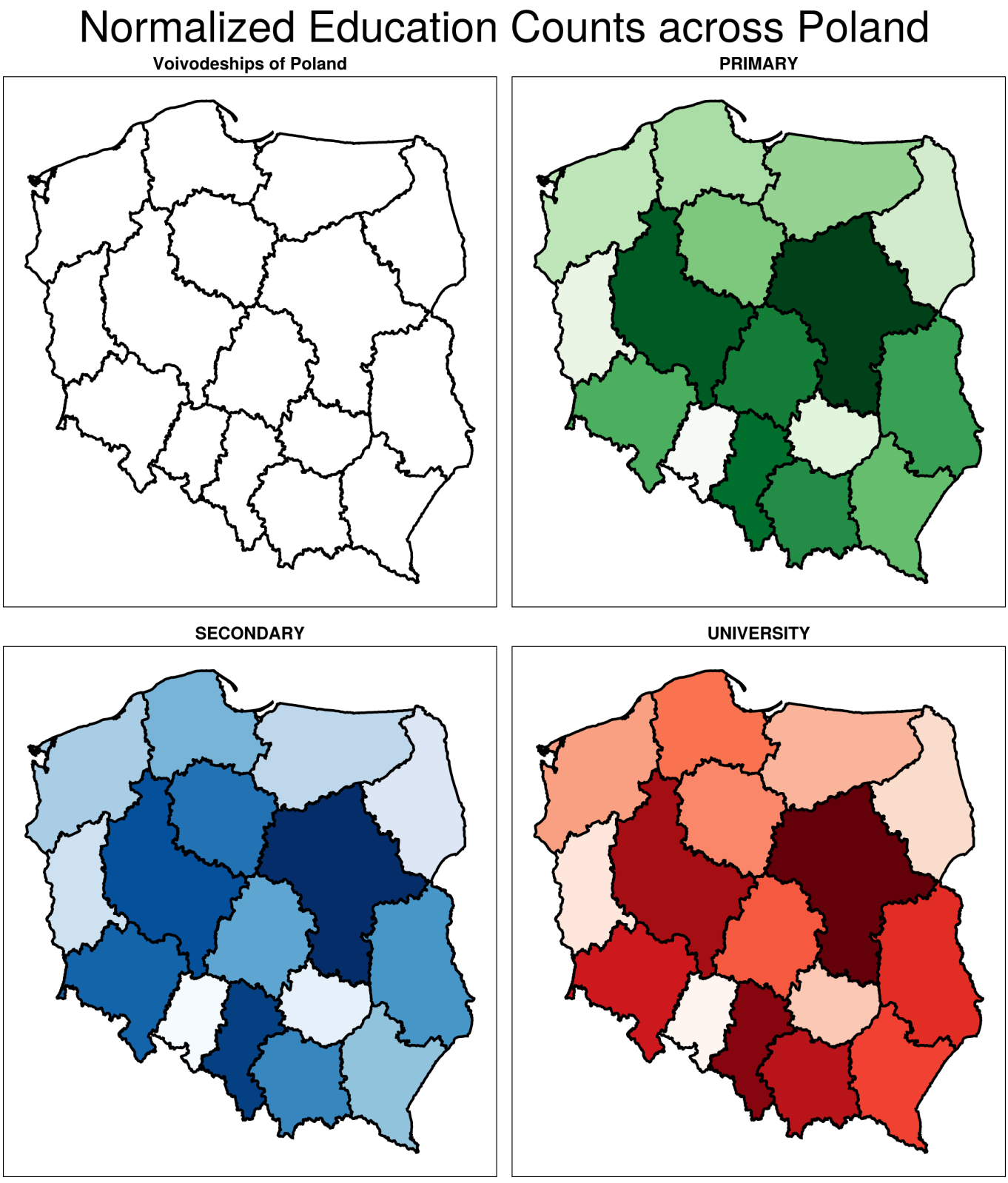
Out[19]:
```
Label()
```

In [20]:
```
for (idx, counts) in enumerate(edu_counts)
    geo_counts = outerjoin(counts, geodf; on = [:ENUTS2_2013 => :ENUTS2])
    norm_counts = (geo_counts.Count .- minimum(geo_counts.Count)) / (maximum(geo_counts.Count) .- minimum(g

    cmap = cgrad(colors[idx + 1], norm_counts)
    dropmissing!(geo_counts, :geometry)

    ax = axs[idx + 1]
    poly!(ax, geo_counts.geometry, color = cmap[norm_counts], strokecolor = :black, strokewidth = 3);

    ax.title = "$(counts.EDUCPL |> first)"
    hidedecorations!(ax)
end
```

It looks nice but not super helpful:

In [21]:
```
fig
```

Out[21]:



Final Cleaning

Let's make the figure more useful!

In [22]:
```
fig = Figure(
  size = (1200, 1400),
  fontsize = 20
);
```

```
axs = [Axis(fig[x, y]) for x in 1:2 for y in 1:2]
poly!(axs[1], geodf.geometry, color = :white, strokecolor = :black, strokewidth = 2);

hidedecorations!(axs[1])
axs[1].title = "Voivodeships of Poland"
```

Out[22]: "Voivodeships of Poland"

In [23]:
```
Label(fig[:, :, Top()], "Normalized Educational Attainment across Poland", fontsize = 50, padding = (0, 0,
```

Out[23]: Label()

In [24]:
```
for (idx, counts) in enumerate(edu_counts)
    geo_counts = outerjoin(counts, geodf; on = [:ENUTS2_2013 => :ENUTS2])
    norm_counts = (geo_counts.Count .- minimum(geo_counts.Count)) / (maximum(geo_counts.Count) .- minimum(g

    cmap = cgrad(:Wistia, norm_counts)
    dropmissing!(geo_counts, :geometry)

    ax = axs[idx + 1]
    poly!(ax, geo_counts.geometry, color = cmap[norm_counts], strokecolor = :black, strokewidth = 2);

    ax.title = "$(counts.EDUCPL |> first)"
    hidedecorations!(ax)
end
```

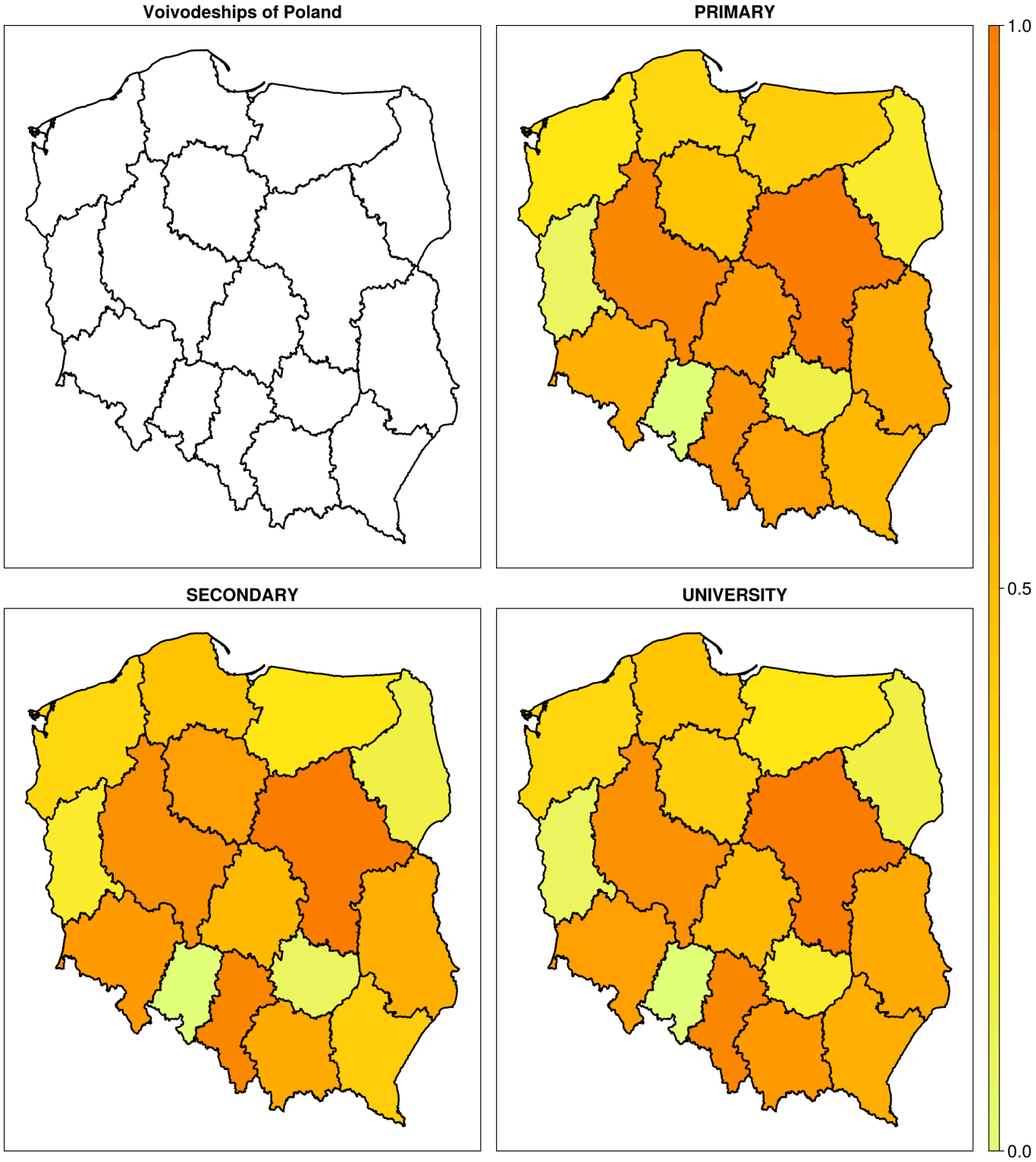
Add a colorbar!

In [25]:
```
Colorbar(fig[:, 3], limits = (0, 1), colormap = :Wistia)
```

Out[25]: Colorbar()

In [26]:
```
fig
```

Out[26]:


Normalized Educational Attainment across Poland
Voivodeships of Poland
PRIMARY
SECONDARY
UNIVERSITY
1.0
0.5
0.0