PORTUGAL: Shapefiles gerais

Some tests using Portugal Maps and information about the country to plot then using the Geo Spacial reference

Manuel Robalinho - Set 2018

http://forest-gis.com/2012/01/portugal-shapefiles-gerais-do-pais.html/ (http://forest-gis.com/2012/01/portugal-shapefiles-gerais-do-pais.html/)

https://www.europeandataportal.eu/data/en/dataset?q=portugal&country=pt (https://www.europeandataportal.eu/data/en/dataset?q=portugal&country=pt)

http://www.mapcruzin.com/free-portugal-arcgis-maps-shapefiles.htm (http://www.mapcruzin.com/free-portugal-arcgis-maps-shapefiles.htm)

In [1]:

```
%matplotlib inline

import matplotlib.pyplot as plt
import geopandas as gpd
```

In [2]:

```
path = 'ML/GeoPandas/Portugal/'
```

In [3]:

```
# Tables Portugal / Distritos / Concelhos
uf_pt1 = gpd.read_file(path + 'PRT_adm_shp/PRT_adm0.shp')
uf_pt2 = gpd.read_file(path + 'PRT_adm_shp/PRT_adm1.shp')
uf_pt3 = gpd.read_file(path + 'PRT_adm_shp/PRT_adm2.shp')
```

In [4]:

```
# Portugal
uf_pt1.head()
```

Out[4]:

	ID_0	ISO	NAME_ENGLI	NAME_ISO	NAME_FAO	NAME_LOCAL	NAME_OBSC
0	182	PRT	Portugal	PORTUGAL	Portugal	Portugal	None

1 rows × 68 columns

In [5]:

```
# Portugal - Distritos
uf_pt2.head()
uf_pt2[uf_pt2.TYPE_1 == 'Regiões autônoma'].head()
```

Out[5]:

	ID_0	ISO	NAME_0	ID_1	NAME_1	HASC_1	CCN_1	CCA_1	TYPE_1	ENGTYP
2	182	PRT	Portugal	3	Azores	PT.AC	0	None	Regiões autônoma	'Autonom Region'
12	182	PRT	Portugal	13	Madeira	None	0	None	Regiões autônoma	'Autonom Region'

In [6]:

```
# Delete information about Autonomos Regions (Açores e Madeira)
uf_pt2x = uf_pt2[uf_pt2.TYPE_1 != 'Regiões autônoma']
```

In [7]:

```
# Portugal - Concelhos
uf_pt3.head()
```

Out[7]:

	ID_0	ISO	NAME_0	ID_1	NAME_1	ID_2	NAME_2	HASC_2	CCN_2	CCA_2	
0	182	PRT	Portugal	1	Évora	1	Évora	PT.EV.EV	0	0705	C
1	182	PRT	Portugal	1	Évora	2	Alandroal	PT.EV.AL	0	0701	C
2	182	PRT	Portugal	1	Évora	3	Arraiolos	PT.EV.AR	0	0702	C
3	182	PRT	Portugal	1	Évora	4	Borba	PT.EV.BO	0	0703	C
4	182	PRT	Portugal	1	Évora	5	Estremoz	PT.EV.ES	0	0704	C

In [8]:

```
# Delete information about Autonomos Regions (Açores e Madeira)
uf_pt3x = uf_pt3.copy()
uf_pt3x = uf_pt3x[uf_pt3x.NAME_1 != 'Madeira']
uf_pt3x = uf_pt3x[uf_pt3x.NAME_1 != 'Açores']
uf_pt3x = uf_pt3x[uf_pt3x.NAME_1 != 'Azores']

# Managing INTERSECTS in Geo Spacial points betwen DISTRITS and CONCELHOS - Excluded Ac
ores e Madeira

#concelhos_with_distritos = gpd.sjoin(uf_pt2x, uf_pt3, how='inner', op='intersects')
#concelhos_with_distritos = gpd.sjoin(uf_pt2x, uf_pt3, how='left', op='within')
uf_pt3x.head()
```

Out[8]:

	ID_0	ISO	NAME_0	ID_1	NAME_1	ID_2	NAME_2	HASC_2	CCN_2	CCA_2	TY
0	182	PRT	Portugal	1	Évora	1	Évora	PT.EV.EV	0	0705	Cor
1	182	PRT	Portugal	1	Évora	2	Alandroal	PT.EV.AL	0	0701	Cor
2	182	PRT	Portugal	1	Évora	3	Arraiolos	PT.EV.AR	0	0702	Cor
3	182	PRT	Portugal	1	Évora	4	Borba	PT.EV.BO	0	0703	Cor
4	182	PRT	Portugal	1	Évora	5	Estremoz	PT.EV.ES	0	0704	Cor

In [9]:

```
concelhos_with_distritos =  uf_pt3x.copy()
# Filter only one district Acores
#concelhos_with_distritos[concelhos_with_distritos.HASC_1 == 'PT.AC'].head()

concelhos_with_distritos[concelhos_with_distritos.NAME_1  == 'Porto'].head()
```

Out[9]:

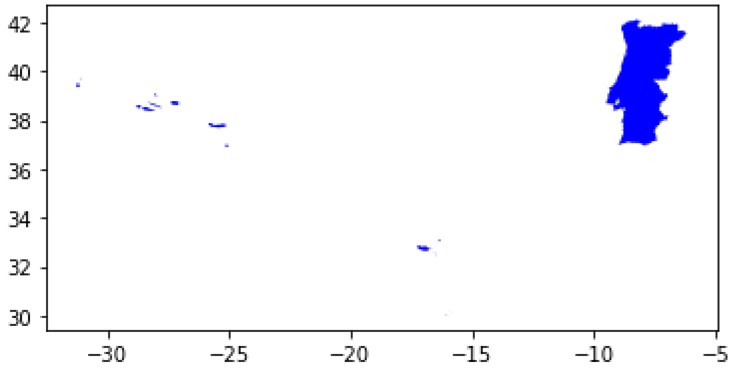
	ID_0	ISO	NAME_0	ID_1	NAME_1	ID_2	NAME_2	HASC_2	CCN_2	CCA_2
208	182	PRT	Portugal	15	Porto	207	Amarante	PT.PO.AM	0	1301
209	182	PRT	Portugal	15	Porto	208	Baião	PT.PO.BA	0	1302
210	182	PRT	Portugal	15	Porto	209	Felgueiras	PT.PO.FE	0	1303
211	182	PRT	Portugal	15	Porto	210	Gondomar	PT.PO.GO	0	1304
212	182	PRT	Portugal	15	Porto	211	Lousada	PT.PO.LO	0	1305

In [10]:

```
# Portugal
uf_pt1.plot(color='blue')
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25cf6ffd4a8>
```

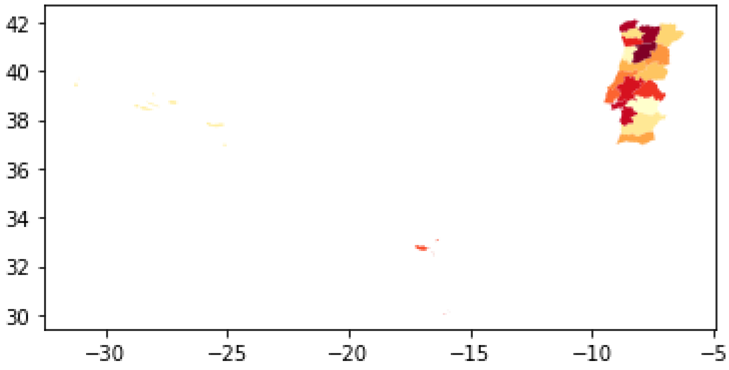


In [11]:

```
# Distritos
uf_pt2.plot(cmap='YlOrRd')
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25cf8e15ac8>
```

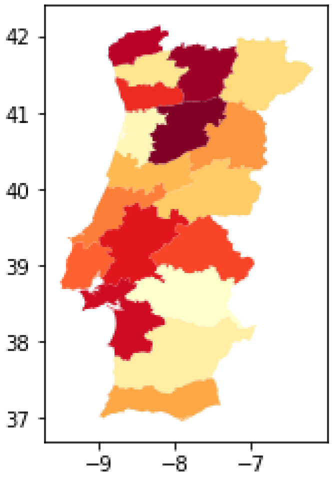


In [12]:

```
# Distritos - Exclude Açores and Madeira
uf_pt2x.plot(cmap='YlOrRd')
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25cf7a7bf28>
```



In [13]:

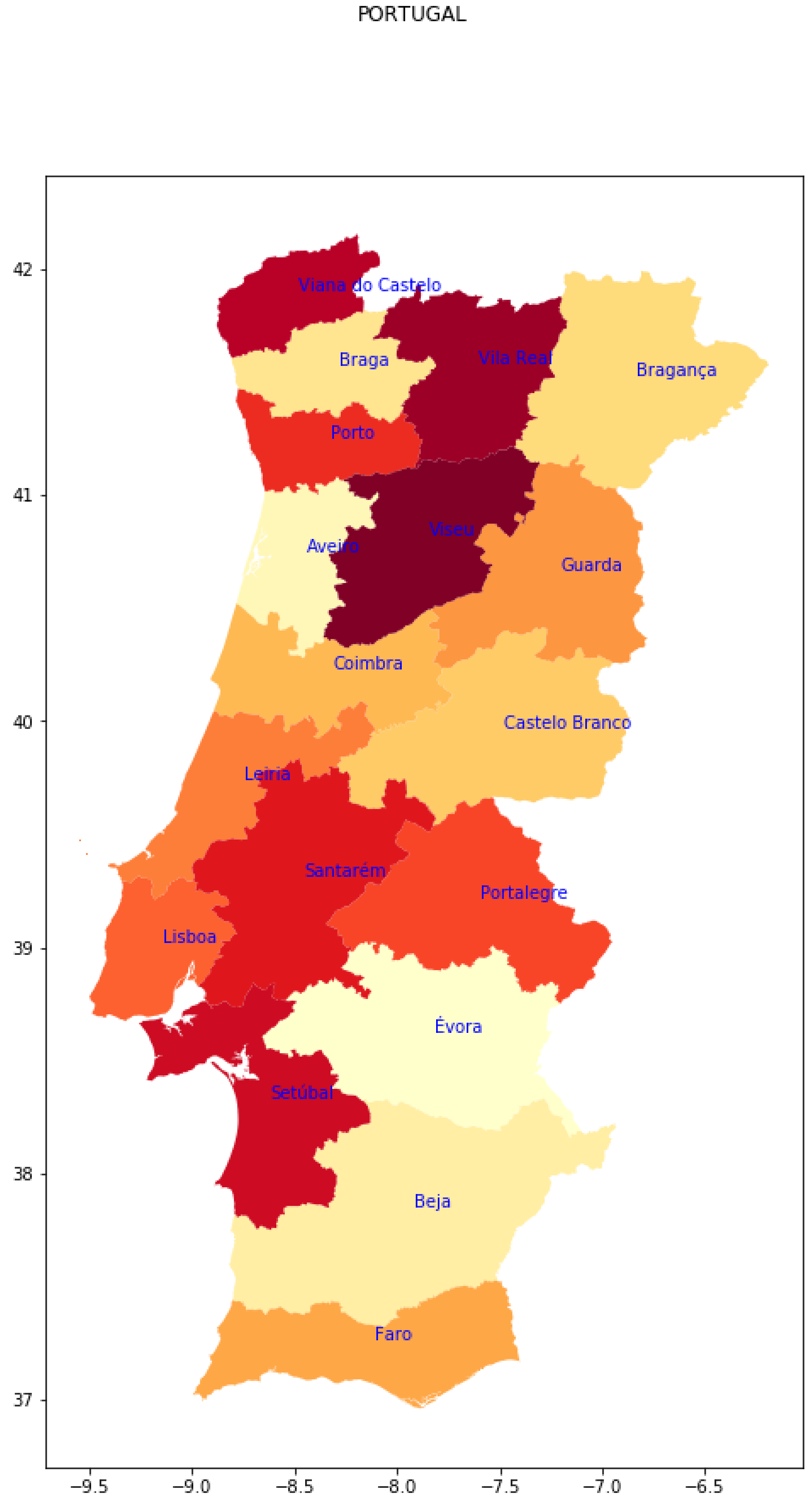
```
# Transform POLYGON to POINT
# copy poly to new GeoDataFrame
points = uf_pt2x.copy()
# change the geometry
points.geometry = points['geometry'].centroid
# same crs
#points.crs = poly.crs
points.head()
```

Out[13]:

	ID_0	ISO	NAME_0	ID_1	NAME_1	HASC_1	CCN_1	CCA_1	TYPE_1	ENGTYPE_
0	182	PRT	Portugal	1	Évora	PT.EV	0	None	Distrito	District
1	182	PRT	Portugal	2	Aveiro	PT.AV	0	01	Distrito	District
3	182	PRT	Portugal	4	Beja	PT.BE	0	02	Distrito	District
4	182	PRT	Portugal	5	Braga	PT.BR	0	03	Distrito	District
5	182	PRT	Portugal	6	Bragança	PT.BA	0	04	Distrito	District

In [14]:

```
# PLOT Distrits points
f, ax = plt.subplots(1, figsize=(8,14))
ax.set_axis_on()
f.suptitle('PORTUGAL')
# Plot the states area
ax = uf_pt2x.plot(ax=ax, facecolor='blue', alpha=1, linewidth=0, cmap='YlOrRd')

# Plot the labels
for x, y, label in zip(points.geometry.x, points.geometry.y, points.NAME_1):
    ax.annotate(label, xy=(x, y), xytext=(3, 3), alpha=3, textcoords="offset points",co
lor='blue')
```


PORTUGAL
Viana do Castelo
Braga
Vila Real
Bragança
Porto
Aveiro
Viseu
Guarda
Coimbra
Castelo Branco
Leiria
Santarém
Portalegre
Lisboa
Évora
Setúbal
Beja
Faro
42
41
40
39
38
37
−9.5
−9.0
−8.5
−8.0
−7.5
−7.0
−6.5

In [15]:

```
uf_pt3x[uf_pt3x.NAME_1  == 'Porto'].head(20)
```

Out[15]:

	ID_0	ISO	NAME_0	ID_1	NAME_1	ID_2	NAME_2	HASC_2	CCN_2	CCA_2
208	182	PRT	Portugal	15	Porto	207	Amarante	PT.PO.AM	0	1301
209	182	PRT	Portugal	15	Porto	208	Baião	PT.PO.BA	0	1302
210	182	PRT	Portugal	15	Porto	209	Felgueiras	PT.PO.FE	0	1303
211	182	PRT	Portugal	15	Porto	210	Gondomar	PT.PO.GO	0	1304
212	182	PRT	Portugal	15	Porto	211	Lousada	PT.PO.LO	0	1305
213	182	PRT	Portugal	15	Porto	212	Maia	PT.PO.MI	0	1306
214	182	PRT	Portugal	15	Porto	213	Marco de Canaveses	PT.PO.MC	0	1307
215	182	PRT	Portugal	15	Porto	214	Matosinhos	PT.PO.MT	0	1308
216	182	PRT	Portugal	15	Porto	215	Póvoa de Varzim	PT.PO.PV	0	1313
217	182	PRT	Portugal	15	Porto	216	Paços de Ferreira	PT.PO.PF	0	1309
218	182	PRT	Portugal	15	Porto	217	Paredes	PT.PO.PA	0	1310
219	182	PRT	Portugal	15	Porto	218	Penafiel	PT.PO.PE	0	1311
220	182	PRT	Portugal	15	Porto	219	Porto	PT.PO.PO	0	1312
221	182	PRT	Portugal	15	Porto	220	Santo Tirso	PT.PO.ST	0	1314

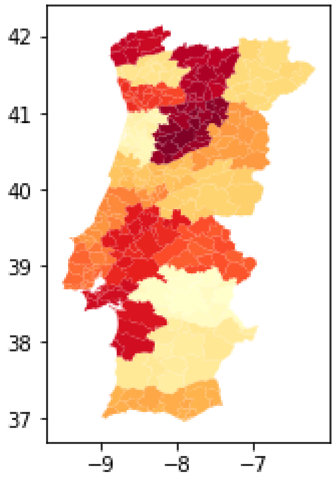
	ID_0	ISO	NAME_0	ID_1	NAME_1	ID_2	NAME_2	HASC_2	CCN_2	CCA_2
222	182	PRT	Portugal	15	Porto	221	Trofa	PT.PO.TR	0	1318
223	182	PRT	Portugal	15	Porto	222	Valongo	PT.PO.VA	0	1315
224	182	PRT	Portugal	15	Porto	223	Vila do Conde	PT.PO.VC	0	1316
225	182	PRT	Portugal	15	Porto	224	Vila Nova de Gaia	PT.PO.VG	0	1317

In [16]:

```
# Concelhos
uf_pt3x.plot(cmap='YlOrRd')
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25cf821d908>
```



In [17]:

```
# Transform POLYGON to POINT
# copy poly to new GeoDataFrame
points2 = concelhos_with_distritos.copy()
# change the geometry
points2.geometry = points2['geometry'].centroid
# same crs
#points.crs = poly.crs
points2.head()
```

Out[17]:

	ID_0	ISO	NAME_0	ID_1	NAME_1	ID_2	NAME_2	HASC_2	CCN_2	CCA_2	TY
0	182	PRT	Portugal	1	Évora	1	Évora	PT.EV.EV	0	0705	Cor
1	182	PRT	Portugal	1	Évora	2	Alandroal	PT.EV.AL	0	0701	Cor
2	182	PRT	Portugal	1	Évora	3	Arraiolos	PT.EV.AR	0	0702	Cor
3	182	PRT	Portugal	1	Évora	4	Borba	PT.EV.BO	0	0703	Cor
4	182	PRT	Portugal	1	Évora	5	Estremoz	PT.EV.ES	0	0704	Cor

In [18]:

```
# PLOT Concelhos points
f, ax = plt.subplots(1, figsize=(15,24))
ax.set_axis_on()
f.suptitle('PORTUGAL')
# Plot the states area
ax = concelhos_with_distritos.plot(ax=ax, facecolor='blue', alpha=1, linewidth=0, cmap=
'YlOrRd')

# Plot the labels
for x, y, label in zip(points2.geometry.x, points2.geometry.y, points2.NAME_2):
    ax.annotate(label, xy=(x, y), xytext=(3, 3), alpha=3, textcoords="offset points",co
lor='blue')
```

PORTUGAL

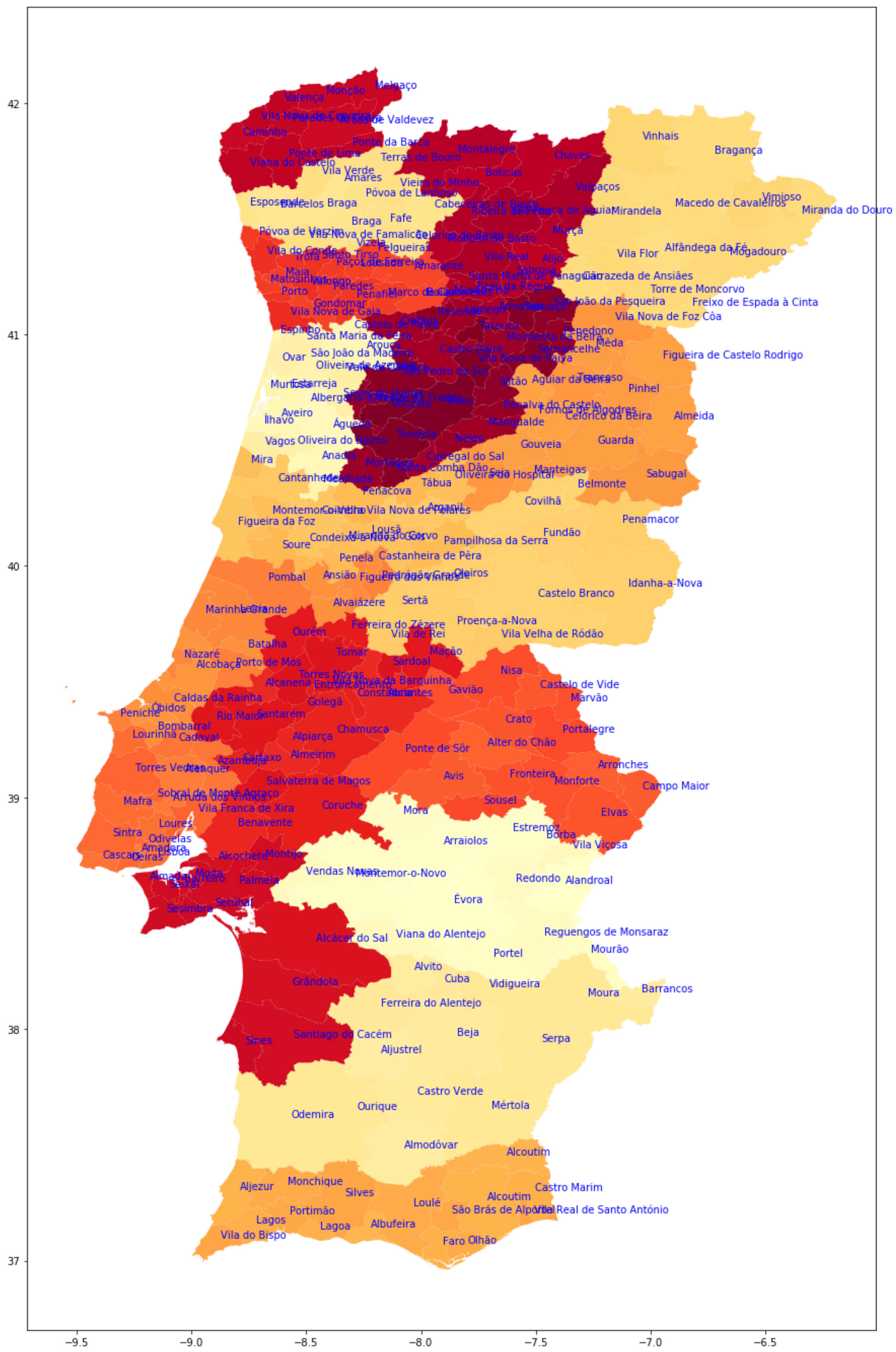

42
41
40
39
38
37
−9.5
−9.0
−8.5
−8.0
−7.5
−7.0
−6.5
Melgaço
Monção
Valença
Vila Nova de Cerveira
Paredes de Coura
Arcos de Valdevez
Caminha
Ponte da Barca
Ponte de Lima
Terras de Bouro
Montalegre
Chaves
Viana do Castelo
Vila Verde
Amares
Boticas
Vieira do Minho
Póvoa de Lanhoso
Valpaços
Vinhais
Bragança
Esposende
Barcelos
Braga
Cabeceiras de Basto
Ribeira de Pena
Vila Pouca de Aguiar
Mirandela
Macedo de Cavaleiros
Vimioso
Miranda do Douro
Fafe
Póvoa de Varzim
Vila Nova de Famalicão
Celorico de Basto
Mondim de Basto
Murça
Vizela
Felgueiras
Vila do Conde
Santo Tirso
Trofa
Paços de Ferreira
Lousada
Amarante
Vila Real
Alijó
Vila Flor
Alfândega da Fé
Mogadouro
Maia
Matosinhos
Valongo
Paredes
Penafiel
Marco de Canaveses
Sabrosa
Santa Marta de Penaguião
Carrazeda de Ansiães
Peso da Régua
Torre de Moncorvo
Porto
Gondomar
Vila Nova de Gaia
Resende
Lamego
Armamar
São João da Pesqueira
Freixo de Espada à Cinta
Vila Nova de Foz Côa
Espinho
Castelo de Paiva
Cinfães
Tarouca
Penedono
Santa Maria da Feira
Moimenta da Beira
Mêda
Arouca
Castro Daire
Sernancelhe
Ovar
São João da Madeira
Oliveira de Azeméis
Vale de Cambra
Vila Nova de Paiva
Figueira de Castelo Rodrigo
São Pedro do Sul
Murtosa
Estarreja
Sátão
Aguiar da Beira
Trancoso
Pinhel
Albergaria-a-Velha
Sever do Vouga
Oliveira de Frades
Vouzela
Viseu
Penalva do Castelo
Fornos de Algodres
Celorico da Beira
Almeida
Aveiro
Ílhavo
Águeda
Mangualde
Tondela
Nelas
Vagos
Oliveira do Bairro
Gouveia
Guarda
Anadia
Mira
Carregal do Sal
Mortágua
Santa Comba Dão
Oliveira do Hospital
Seia
Manteigas
Sabugal
Cantanhede
Mealhada
Belmonte
Tábua
Penacova
Covilhã
Montemor-o-Velho
Coimbra
Vila Nova de Poiares
Arganil
Figueira da Foz
Penamacor
Lousã
Góis
Soure
Condeixa-a-Nova
Miranda do Corvo
Pampilhosa da Serra
Fundão
Penela
Castanheira de Pêra
Pombal
Ansião
Figueiró dos Vinhos
Pedrógão Grande
Oleiros
Idanha-a-Nova
Castelo Branco
Alvaiázere
Sertã
Marinha Grande
Leiria
Ferreira do Zêzere
Proença-a-Nova
Ourém
Vila de Rei
Vila Velha de Ródão
Batalha
Nazaré
Tomar
Mação
Alcobaça
Porto de Mós
Sardoal
Torres Novas
Nisa
Alcanena
Vila Nova da Barquinha
Castelo de Vide
Entroncamento
Constância
Abrantes
Gavião
Marvão
Caldas da Rainha
Golegã
Óbidos
Peniche
Rio Maior
Santarém
Crato
Chamusca
Bombarral
Cadaval
Alpiarça
Portalegre
Lourinhã
Alter do Chão
Ponte de Sor
Cartaxo
Almeirim
Azambuja
Arronches
Torres Vedras
Alenquer
Fronteira
Monforte
Avis
Campo Maior
Salvaterra de Magos
Sobral de Monte Agraço
Arruda dos Vinhos
Mafra
Sousel
Coruche
Vila Franca de Xira
Mora
Elvas
Benavente
Loures
Estremoz
Sintra
Odivelas
Borba
Arraiolos
Vila Viçosa
Amadora
Cascais
Oeiras
Lisboa
Alcochete
Montijo
Vendas Novas
Montemor-o-Novo
Almada
Moita
Barreiro
Seixal
Palmela
Redondo
Alandroal
Sesimbra
Setúbal
Évora
Alcácer do Sal
Viana do Alentejo
Reguengos de Monsaraz
Portel
Mourão
Alvito
Cuba
Vidigueira
Barrancos
Moura
Grândola
Ferreira do Alentejo
Beja
Serpa
Sines
Santiago do Cacém
Aljustrel
Castro Verde
Mértola
Odemira
Ourique
Almodôvar
Alcoutim
Aljezur
Monchique
Silves
Castro Marim
Portimão
Loulé
São Brás de Alportel
Vila Real de Santo António
Lagos
Lagoa
Albufeira
Vila do Bispo
Faro
Olhão

Tabela com população de Portugal por Distrito / Concelho /Freguesia - Censos 2010

https://www.portugal2020.pt/Portal2020/canal-2020 (https://www.portugal2020.pt/Portal2020/canal-2020)

Arquivo: PopulacaoResidentePSexoGruposEtarios2010.xls

In [19]:

```
import pandas as pd
import numpy as np
```

In [20]:

```
path = 'ML/GeoPandas/Portugal/'
path
```

Out[20]:

```
'ML/GeoPandas/Portugal/'
```

In [21]:

```
# Read table about Portugal population

df0 = pd.read_excel(path +'PopulacaoResidentePSexoGruposEtarios2010.xls')
```

In [22]:

```
df0.head()
```

Out[22]:

	NUTS I, II e III e Municípios (NUTS 2002)	Género	Grupos etários - TOTAL	Grupos etários - 0 - 4	Grupos etários - 5 - 9	Grupos etários - 10 - 14	Grupos etários - 15-19	Grupos etários - 20-24	Grupos etários - 25-29
0	Portugal	HM	10 572 157	483 751	528 150	563 999	566 908	581 782	659 248
1	Portugal	H	5 054 330	246 975	270 450	288 429	289 249	293 007	326 559
2	Portugal	M	5 517 827	236 776	257 700	275 570	277 659	288 775	332 689
3	Continente	HM	10 057 380	457 458	498 402	531 617	533 132	547 700	622 885
4	Continente	H	4 806 367	233 515	255 260	271 840	272 047	275 632	308 312

5 rows × 21 columns

In [23]:

```
# Read table about Portuguese population

df1 = pd.read_excel(path +'Censos2011_Pop_Freguesias.xls')
```

In [24]:

```
df1.head(10)
```

Out[24]:

	Fonte: CENSOS 2011 - Instituto Nacional de Estatistica	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnam
0	CAOP 2013 - Carta Administrativa Oficial Portu...	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Distrito (DT)	Designação DT	Concelho (CC)	Designação CC	Freguesia (FR)	Designação FR	PopRes_ (nº)
3	'01'	Aveiro	'0101'	Águeda	'010103'	Aguada de Cima	4013
4	'01'	Aveiro	'0101'	Águeda	'010109'	Fermentelos	3258
5	'01'	Aveiro	'0101'	Águeda	'010112'	Macinhata do Vouga	3406
6	'01'	Aveiro	'0101'	Águeda	'010119'	Valongo do Vouga	4877
7	'01'	Aveiro	'0101'	Águeda	'010121'	União das freguesias de Águeda e Borralha	13576
8	'01'	Aveiro	'0101'	Águeda	'010122'	União das freguesias de Barrô e Aguada de Baixo	3209
9	'01'	Aveiro	'0101'	Águeda	'010123'	União das freguesias de Belazaima do Chão, Cas...	1611

In [25]:

```
# Rename columns
names = df1.columns.tolist()
names[names.index('Unnamed: 1')] = 'Distrito'
names[names.index('Unnamed: 3')] = 'Concelho'
names[names.index('Unnamed: 5')] = 'Freguesia'
names[names.index('Unnamed: 6')] = 'Num_Residentes'
df1.columns = names
```

In [26]:

```
df1.head()
```

Out[26]:

	Fonte: CENSOS 2011 - Instituto Nacional de Estatistica	**Distrito**	**Unnamed: 2**	**Concelho**	**Unnamed: 4**	**Freguesia**	**Num_**
0	CAOP 2013 - Carta Administrativa Oficial Portu...	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Distrito (DT)	Designação DT	Concelho (CC)	Designação CC	Freguesia (FR)	Designação FR	PopR
3	'01'	Aveiro	'0101'	Águeda	'010103'	Aguada de Cima	4013
4	'01'	Aveiro	'0101'	Águeda	'010109'	Fermentelos	3258

In [27]:

```
# See lines and columns
df1.shape
```

Out[27]:

```
(2885, 9)
```

In [28]:

```
# New table with the relevant informations
df1_x = df1[['Distrito','Concelho','Freguesia','Num_Residentes']]
```

In [29]:

```
# Delete firts 3 lines, not relevants
df1_x1 = df1_x.copy()
df1_x1.drop([0,1,2], axis='rows', inplace=True)
df1_x1.head()
```

Out[29]:

	Distrito	Concelho	Freguesia	Num_Residentes
3	Aveiro	Águeda	Aguada de Cima	4013
4	Aveiro	Águeda	Fermentelos	3258
5	Aveiro	Águeda	Macinhata do Vouga	3406
6	Aveiro	Águeda	Valongo do Vouga	4877
7	Aveiro	Águeda	União das freguesias de Águeda e Borralha	13576

In [30]:

```
df1_x1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2882 entries, 3 to 2884
Data columns (total 4 columns):
Distrito          2882 non-null object
Concelho          2882 non-null object
Freguesia         2882 non-null object
Num_Residentes    2882 non-null object
dtypes: object(4)
memory usage: 112.6+ KB
```

In [31]:

```
# Transform type to Num_Residente from object to int
df1_x1['Num_Residentes'] = df1_x1['Num_Residentes'].astype(float)
```

In [32]:

```
df1_x1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2882 entries, 3 to 2884
Data columns (total 4 columns):
Distrito          2882 non-null object
Concelho          2882 non-null object
Freguesia         2882 non-null object
Num_Residentes    2882 non-null float64
dtypes: float64(1), object(3)
memory usage: 112.6+ KB
```

In [33]:

```
# Sum table by Distrito
df1_x2 = df1_x1.groupby('Distrito')['Num_Residentes'].sum()
# Transform table pandas
s_table = pd.DataFrame(df1_x2)
# Rename columns
s_table.columns = ['Num_Residentes']
s_table = s_table.sort_values(by=['Num_Residentes'],ascending=False)
```

In [34]:

```
s_table
```

Out[34]:

	Num_Residentes
Distrito	
Lisboa	2250533.0
Porto	1817175.0
Setúbal	851258.0
Braga	848185.0
Aveiro	714197.0
Leiria	470922.0
Santarém	453646.0
Faro	451006.0
Coimbra	430104.0
Viseu	377653.0
Viana do Castelo	244836.0
Vila Real	206661.0
Castelo Branco	196264.0
Évora	166726.0
Guarda	160939.0
Beja	152758.0
Bragança	136252.0
Portalegre	118506.0

In [35]:

```
# Total population
Tot_population = s_table['Num_Residentes'].sum()
print ('Numero Residentes em Portugal(Censos 2011):', Tot_population)
```

```
Numero Residentes em Portugal(Censos 2011): 10047621.0
```

In [36]:

```
# Create % from populatio by district
if (s_table['Num_Residentes'].empty) != True:
    s_table['Perc_Population'] = round(100*(s_table['Num_Residentes'] / Tot_population
),2)
else:
    s_table['Perc_Population'] = 0

s_table
```

Out[36]:

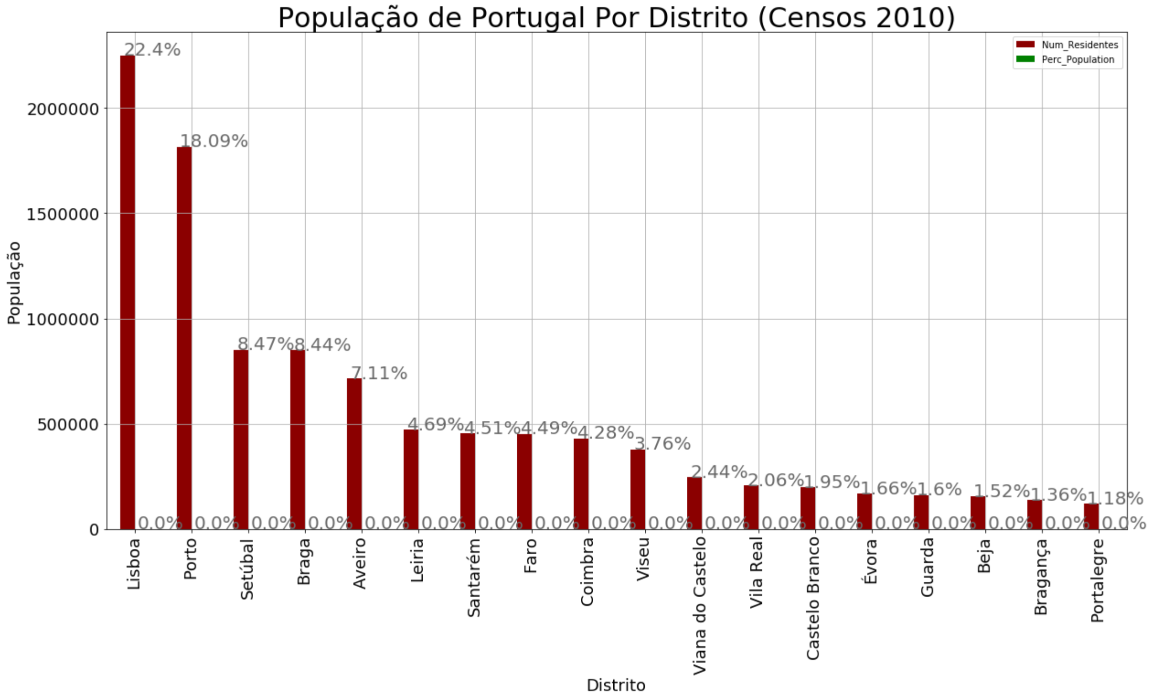
	Num_Residentes	Perc_Population
Distrito		
Lisboa	2250533.0	22.40
Porto	1817175.0	18.09
Setúbal	851258.0	8.47
Braga	848185.0	8.44
Aveiro	714197.0	7.11
Leiria	470922.0	4.69
Santarém	453646.0	4.51
Faro	451006.0	4.49
Coimbra	430104.0	4.28
Viseu	377653.0	3.76
Viana do Castelo	244836.0	2.44
Vila Real	206661.0	2.06
Castelo Branco	196264.0	1.95
Évora	166726.0	1.66
Guarda	160939.0	1.60
Beja	152758.0	1.52
Bragança	136252.0	1.36
Portalegre	118506.0	1.18

In [37]:

```
df = s_table.copy()
#del df['Perc_Population']
```

In [38]:

```
# Plot a bar graphic
ax = df.plot( kind='bar', figsize=(20,10), grid=True, color=('darkred','Green'),fontsiz
e=18)
plt.title('População de Portugal Por Distrito (Censos 2010)',fontsize=30)
plt.xlabel('Distrito',fontsize=18)
plt.ylabel('População',fontsize=18)

# ----
# Create % to the graph
# create a list to collect the plt.patches data
totals = []

# find the values and append to list
for i in ax.patches:
    totals.append(i.get_height())

# set individual bar lables using above list
total = sum(totals)

# set individual bar lables using above list (porcentagem )
for i in ax.patches:
    # get_x pulls left or right; get_height pushes up or down
    ax.text(i.get_x()+.05, i.get_height()+.02, \
            str(round((i.get_height()/total)*100, 2))+'%', fontsize=20,
                color='dimgrey')
# set individual bar lables using above list (Valor)
#for i in ax.patches:
    # get_x pulls left or right; get_height pushes up or down
#     ax.text(i.get_x()+.07, i.get_height()+20, \
#             str(round((i.get_height()), 10)), fontsize=12, color='dimgrey',
#                 rotation=45)
plt.savefig('Populacao por distrito 1')
plt.show()
```



In [39]:

```
# Merge tables Districts with Geo Spacial
# Distritos
uf_pt2x.head()
```

Out[39]:

	ID_0	ISO	NAME_0	ID_1	NAME_1	HASC_1	CCN_1	CCA_1	TYPE_1	ENGTYPE_
0	182	PRT	Portugal	1	Évora	PT.EV	0	None	Distrito	District
1	182	PRT	Portugal	2	Aveiro	PT.AV	0	01	Distrito	District
3	182	PRT	Portugal	4	Beja	PT.BE	0	02	Distrito	District
4	182	PRT	Portugal	5	Braga	PT.BR	0	03	Distrito	District
5	182	PRT	Portugal	6	Bragança	PT.BA	0	04	Distrito	District

In [40]:

```
# Rename columns
names = uf_pt2x.columns.tolist()
names[names.index('NAME_1')] = 'Distrito'
uf_pt2x.columns = names
```

In [41]:

```
uf_pt2x.head()
```

Out[41]:

	ID_0	ISO	NAME_0	ID_1	Distrito	HASC_1	CCN_1	CCA_1	TYPE_1	ENGTYPE_
0	182	PRT	Portugal	1	Évora	PT.EV	0	None	Distrito	District
1	182	PRT	Portugal	2	Aveiro	PT.AV	0	01	Distrito	District
3	182	PRT	Portugal	4	Beja	PT.BE	0	02	Distrito	District
4	182	PRT	Portugal	5	Braga	PT.BR	0	03	Distrito	District
5	182	PRT	Portugal	6	Bragança	PT.BA	0	04	Distrito	District

In [42]:

```
# Merge two tables by Distrito
result = pd.merge(uf_pt2x, df, on='Distrito')
```

In [43]:

```
result.head()
```

Out[43]:

	ID_0	ISO	NAME_0	ID_1	Distrito	HASC_1	CCN_1	CCA_1	TYPE_1	ENGTYPE_
0	182	PRT	Portugal	1	Évora	PT.EV	0	None	Distrito	District
1	182	PRT	Portugal	2	Aveiro	PT.AV	0	01	Distrito	District
2	182	PRT	Portugal	4	Beja	PT.BE	0	02	Distrito	District
3	182	PRT	Portugal	5	Braga	PT.BR	0	03	Distrito	District
4	182	PRT	Portugal	6	Bragança	PT.BA	0	04	Distrito	District

In [44]:

```
# Sotr Table descending by number population
result = result.sort_values(by=['Num_Residentes'],ascending=False)
```

In [45]:

```
# result[result.Distrito  == 'Porto'].head(20)
result.head()
```

Out[45]:

	ID_0	ISO	NAME_0	ID_1	Distrito	HASC_1	CCN_1	CCA_1	TYPE_1	ENGTYPE_
10	182	PRT	Portugal	12	Lisboa	PT.LI	0	None	Distrito	District
12	182	PRT	Portugal	15	Porto	PT.PO	0	None	Distrito	District
14	182	PRT	Portugal	17	Setúbal	PT.SE	0	None	Distrito	District
3	182	PRT	Portugal	5	Braga	PT.BR	0	03	Distrito	District
1	182	PRT	Portugal	2	Aveiro	PT.AV	0	01	Distrito	District

In [46]:

```
# Make new index resulting new sort values
result.reset_index()
```

Out[46]:

	index	ID_0	ISO	NAME_0	ID_1	Distrito	HASC_1	CCN_1	CCA_1	TYPE_1	EI
0	10	182	PRT	Portugal	12	Lisboa	PT.LI	0	None	Distrito	Di
1	12	182	PRT	Portugal	15	Porto	PT.PO	0	None	Distrito	Di
2	14	182	PRT	Portugal	17	Setúbal	PT.SE	0	None	Distrito	Di
3	3	182	PRT	Portugal	5	Braga	PT.BR	0	03	Distrito	Di
4	1	182	PRT	Portugal	2	Aveiro	PT.AV	0	01	Distrito	Di
5	9	182	PRT	Portugal	11	Leiria	PT.LE	0	None	Distrito	Di
6	13	182	PRT	Portugal	16	Santarém	PT.SA	0	None	Distrito	Di
7	7	182	PRT	Portugal	9	Faro	PT.FA	0	08	Distrito	Di
8	6	182	PRT	Portugal	8	Coimbra	PT.CO	0	06	Distrito	Di
9	17	182	PRT	Portugal	20	Viseu	PT.VI	0	None	Distrito	Di
10	15	182	PRT	Portugal	18	Viana do Castelo	PT.VC	0	None	Distrito	Di
11	16	182	PRT	Portugal	19	Vila Real	PT.VR	0	None	Distrito	Di
12	5	182	PRT	Portugal	7	Castelo Branco	PT.CB	0	05	Distrito	Di
13	0	182	PRT	Portugal	1	Évora	PT.EV	0	None	Distrito	Di

	index	ID_0	ISO	NAME_0	ID_1	Distrito	HASC_1	CCN_1	CCA_1	TYPE_1	EI
14	8	182	PRT	Portugal	10	Guarda	PT.GU	0	09	Distrito	Di
15	2	182	PRT	Portugal	4	Beja	PT.BE	0	02	Distrito	Di
16	4	182	PRT	Portugal	6	Bragança	PT.BA	0	04	Distrito	Di
17	11	182	PRT	Portugal	14	Portalegre	PT.PA	0	None	Distrito	Di

In [47]:

```
#https://matplotlib.org/examples/color/named_colors.html

# List of colors
color_list = ['red', 'tomato', 'darkorange', 'magenta','deepskyblue','plum',
              'cyan', 'lightsalmon','palegreen','wheat','silver','seashell',
              'linen','aliceblue','azure','mintcream', 'oldlace',
              'floralwhite','honeydew','cornsilk','lightyellow','whitesmoke',
              'lavender','lightcyan','white','ghostwhite','snow',
              'oldlace','oldlace','oldlace','oldlace']

#color_list
```

In [48]:

```
# Transform POLYGON to POINT
# copy poly to new GeoDataFrame
points = result.copy()
# change the geometry
points.geometry = points['geometry'].centroid
# same crs
#points.crs = poly.crs
points.head()
```

Out[48]:

	ID_0	ISO	NAME_0	ID_1	Distrito	HASC_1	CCN_1	CCA_1	TYPE_1	ENGTYPE_
10	182	PRT	Portugal	12	Lisboa	PT.LI	0	None	Distrito	District
12	182	PRT	Portugal	15	Porto	PT.PO	0	None	Distrito	District
14	182	PRT	Portugal	17	Setúbal	PT.SE	0	None	Distrito	District
3	182	PRT	Portugal	5	Braga	PT.BR	0	03	Distrito	District
1	182	PRT	Portugal	2	Aveiro	PT.AV	0	01	Distrito	District

In [49]:

```
x1_table = result.copy().reset_index()
x1_table.head()
```

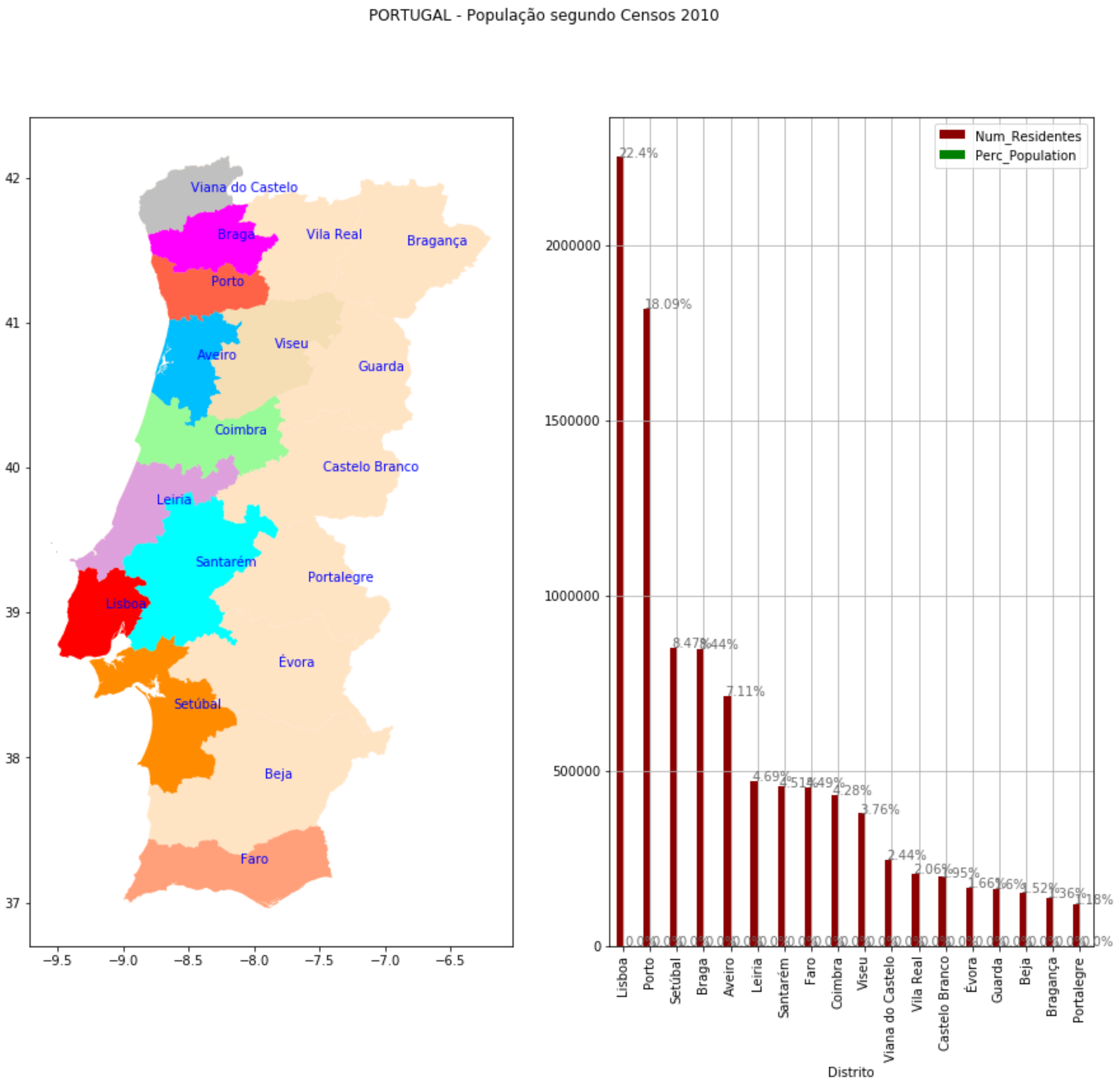
Out[49]:

	index	ID_0	ISO	NAME_0	ID_1	Distrito	HASC_1	CCN_1	CCA_1	TYPE_1	ENGT
0	10	182	PRT	Portugal	12	Lisboa	PT.LI	0	None	Distrito	Distric
1	12	182	PRT	Portugal	15	Porto	PT.PO	0	None	Distrito	Distric
2	14	182	PRT	Portugal	17	Setúbal	PT.SE	0	None	Distrito	Distric
3	3	182	PRT	Portugal	5	Braga	PT.BR	0	03	Distrito	Distric
4	1	182	PRT	Portugal	2	Aveiro	PT.AV	0	01	Distrito	Distric

In [50]:

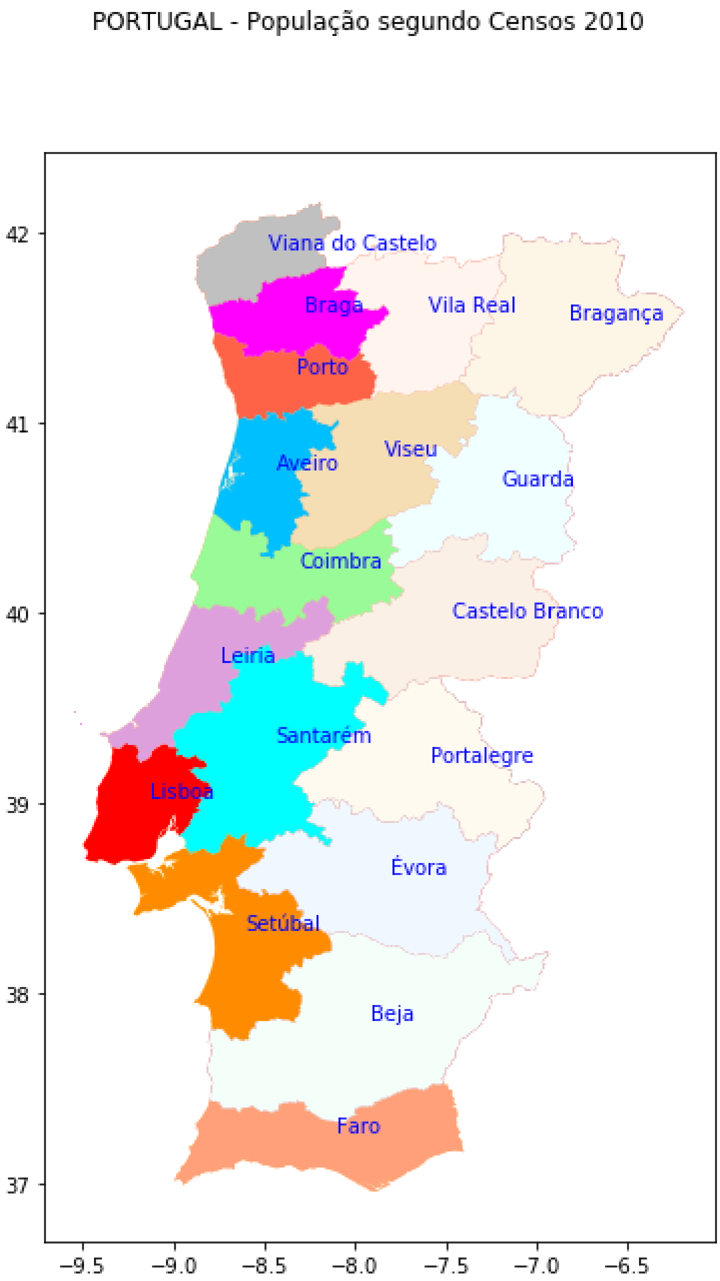
```
# Plot two Graphs

# Defining two area to plot
f, (ax1, ax2) = plt.subplots(ncols=2, sharex=False, sharey=False, figsize=(15,12))
ax1.set_axis_on()
f.suptitle('PORTUGAL - População segundo Censos 2010')

#-- First Graph ----- Colored by greatest Number
# PLOT State points
# Plot the states area
ax1 = x1_table.plot(ax=ax1, facecolor='blue', alpha=1, linewidth=3, color = 'bisque') #
 cmap='YlOrRd')    cmap=plt.cm.rainbow
# 1 registo
ax1 = x1_table[x1_table.index == 0].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[0])
# 2 registo
ax1 = x1_table[x1_table.index == 1].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[1])
# 3 registo
ax1 = x1_table[x1_table.index == 2].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[2])
# 4 registo
ax1 = x1_table[x1_table.index == 3].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[3])
# 5 registo
ax1 = x1_table[x1_table.index == 4].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[4])
# 6 registo
ax1 = x1_table[x1_table.index == 5].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[5])
# 7 registo
ax1 = x1_table[x1_table.index == 6].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[6])
# 8 registo
ax1 = x1_table[x1_table.index == 7].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[7])
# 9 registo
ax1 = x1_table[x1_table.index == 8].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[8])
# 10 registo
ax1 = x1_table[x1_table.index == 9].plot(ax=ax1,  alpha=1, linewidth=1, color=color_lis
t[9])
# 10 registo
ax1 = x1_table[x1_table.index == 10].plot(ax=ax1,  alpha=1, linewidth=1, color=color_li
st[10])

# Plot the labels
for x, y, label in zip(points.geometry.x, points.geometry.y, points.Distrito):
    ax1.annotate(label, xy=(x, y), xytext=(3, 3), alpha=3, textcoords="offset points",c
olor='blue')

# ---  Second Graf -----
df = s_table
ax2.set_axis_on()
df.plot(ax=ax2, kind='bar', grid=True, color=('darkred','Green'),fontsize=10)

# create a list to collect the plt.patches data
totals = []
```

```
# find the values and append to list
for i in ax2.patches:
    totals.append(i.get_height())

# set individual bar lables using above list
total = sum(totals)

# set individual bar lables using above list (porcentagem )
for i in ax2.patches:
    # get_x pulls left or right; get_height pushes up or down
    ax2.text(i.get_x()+.05, i.get_height()+.02, \
            str(round((i.get_height()/total)*100, 2))+'%', fontsize=10,
                color='dimgrey')
```


PORTUGAL - População segundo Censos 2010
Viana do Castelo
Braga
Vila Real
Bragança
Porto
Viseu
Aveiro
Guarda
Coimbra
Castelo Branco
Leiria
Santarém
Portalegre
Lisboa
Évora
Setúbal
Beja
Faro
42
41
40
39
38
37
−9.5
−9.0
−8.5
−8.0
−7.5
−7.0
−6.5
Num_Residentes
Perc_Population
2000000
1500000
1000000
500000
0
22.4%
18.09%
8.47%
8.44%
7.11%
4.69%
4.51%
4.49%
4.28%
3.76%
2.44%
2.06%
1.95%
1.66%
1.6%
1.52%
1.36%
1.18%
0.0%
Lisboa
Porto
Setúbal
Braga
Aveiro
Leiria
Santarém
Faro
Coimbra
Viseu
Viana do Castelo
Vila Real
Castelo Branco
Évora
Guarda
Beja
Bragança
Portalegre
Distrito

In [51]:

```
# PLOT State points
f, ax = plt.subplots(1, figsize=(6,10))
ax.set_axis_on()
f.suptitle('PORTUGAL - População segundo Censos 2010')

# Plot the states area
nrof_labels = len(x1_table)
# Plot all the map
ax = x1_table.plot(ax=ax, facecolor='blue', alpha=1, linewidth=0, cmap='YlOrRd', legend
 = True)   # cmap=plt.cm.rainbow

# Aplicar cor na regiao de acordo com o maior numero
for i,c in enumerate(x1_table.Distrito):
    # Plot a UF map
    ax = x1_table[x1_table.index == i].plot(ax=ax,  facecolor='blue', alpha=1, linewidt
h=0, color=color_list[i])

# Plot the labels
for x, y, label in zip(points.geometry.x, points.geometry.y, points.Distrito):
    ax.annotate(label, xy=(x, y), xytext=(3, 3), alpha=3, textcoords="offset points",co
lor='blue')
```



PORTUGAL - Roads ,Trails and Rail Roads

In [154]:

```
# Tables Portugal - Roads and Trails
rds_pt1 = gpd.read_file(path + 'PRT_rds/PRT_roads.shp')
# Rail Road
rds_pt2 = gpd.read_file(path + 'PRT_rrd/PRT_rails.shp')
# Water
rds_pt3 = gpd.read_file(path + 'PRT_WATER/waterways.shp')
```

In [123]:

```
rds_pt1.head()
```

Out[123]:

	MED_DESCRI	RTT_DESCRI	F_CODE_DES	ISO	ISOCOUNTRY	geome
0	Without Median	Secondary Route	Road	PRT	PORTUGAL	(LINESTRING (-8.2289265506142 42.13301116472...
1	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.589778012939ŧ 42.048389438887..
2	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.589778012939ŧ 42.048389438887..
3	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.633639242857ŧ 42.028831529923..
4	Without Median	Primary Route	Road	PRT	PORTUGAL	LINESTRING (-8.633639242857ŧ 42.028831529923..

In [124]:

```
# Rail Road
rds_pt2.head()
```

Out[124]:

	FID_rail_d	F_CODE_DES	EXS_DESCRI	FCO_DESCRI	FID_countr	ISO	ISOCOUN
0	96319	Railroad	Operational	Single	179	PRT	PORTUGA
1	96485	Railroad	Operational	Single	179	PRT	PORTUGA
2	96497	Railroad	Operational	Single	179	PRT	PORTUGA
3	97844	Railroad	Operational	Single	179	PRT	PORTUGA
4	98035	Railroad	Operational	Single	179	PRT	PORTUGA

In [125]:

```
# Only roads
rds_roads = rds_pt1[rds_pt1.F_CODE_DES == 'Road']
rds_roads.head()
```

Out[125]:

	MED_DESCRI	RTT_DESCRI	F_CODE_DES	ISO	ISOCOUNTRY	geome
0	Without Median	Secondary Route	Road	PRT	PORTUGAL	(LINESTRING (-8.2289265506142 42.13301116472...
1	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.5897780129395 42.048389438887..
2	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.5897780129395 42.048389438887..
3	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.6336392428575 42.028831529923..
4	Without Median	Primary Route	Road	PRT	PORTUGAL	LINESTRING (-8.6336392428575 42.028831529923..

In [126]:

```
# Only trails
rds_trail = rds_pt1[rds_pt1.F_CODE_DES == 'Trail']
rds_trail.head()
```

Out[126]:

	MED_DESCRI	RTT_DESCRI	F_CODE_DES	ISO	ISOCOUNTRY	geor
281	None	None	Trail	PRT	PORTUGAL	(LINESTRING (-31.0930270608 39.71252823725
296	None	None	Trail	PRT	PORTUGAL	(LINESTRING (-31.2140779696 39.52375031260
304	None	None	Trail	PRT	PORTUGAL	LINESTRING (-31.1960564416 39.44124992276
308	None	None	Trail	PRT	PORTUGAL	LINESTRING (-31.2396659235 39.42680748778
309	None	None	Trail	PRT	PORTUGAL	LINESTRING (-31.2574443975 39.45894618172

In [169]:

```
rds_pt3.head()
```

Out[169]:

	osm_id	name	type	width	geometry
0	4718597	None	river	0	LINESTRING (-8.490691699999999 38.9709642, -8....
1	4856704	Rio Varosa	stream	0	LINESTRING (-7.820399 40.964353, -7.819519 40....
2	10577666	None	stream	0	LINESTRING (-6.8283313 41.297276, -6.8284171 4...
3	12641715	None	river	0	LINESTRING (-8.8354898 39.0048754, -8.83484600...
4	12641805	None	river	0	LINESTRING (-8.8395753 39.0185739, -8.83923200...

In [174]:

```
# Rename columns
names = rds_pt3.columns.tolist()
names[names.index('type')] = 'tipo'
rds_pt3.columns = names
```

In [175]:

```
#List unique values in the df['name'] column
rds_pt3.tipo.unique()
```

Out[175]:

```
array(['river', 'stream', 'weir', 'dock', 'dam', 'canal'], dtype=object)
```

In [176]:

```
# Rivers
rds_pt3[rds_pt3.tipo == 'river'].head()
```

Out[176]:

	osm_id	name	tipo	width	geometry
0	4718597	None	river	0	LINESTRING (-8.490691699999999 38.9709642, -8....
3	12641715	None	river	0	LINESTRING (-8.8354898 39.0048754, -8.83484600...
4	12641805	None	river	0	LINESTRING (-8.8395753 39.0185739, -8.83923200...
5	23102773	Rio Ceira	river	0	LINESTRING (-8.394681500000001 40.1725883, -8....
6	23109746	Rio Ceira	river	0	LINESTRING (-8.325596000000001 40.174491, -8.3...

In [180]:

```
rivers = rds_pt3[rds_pt3.tipo == 'river']
rivers.head()
```

Out[180]:

	osm_id	name	tipo	width	geometry
0	4718597	None	river	0	LINESTRING (-8.490691699999999 38.9709642, -8....
3	12641715	None	river	0	LINESTRING (-8.8354898 39.0048754, -8.83484600...
4	12641805	None	river	0	LINESTRING (-8.8395753 39.0185739, -8.83923200...
5	23102773	Rio Ceira	river	0	LINESTRING (-8.394681500000001 40.1725883, -8....
6	23109746	Rio Ceira	river	0	LINESTRING (-8.325596000000001 40.174491, -8.3...

In [127]:

```
# plotting using geopandas
country_boundary = x1_table
roads_only =  rds_roads

road_colors = ['black', 'grey', 'grey', 'black', 'grey', 'grey' ]
line_widths = [1, .5, .5, 1, .5,.5]

# plot the data
fig, ax = plt.subplots(figsize  = (12, 8))
country_boundary.plot(alpha = 1, color="white",
                       edgecolor = "black",
                       ax = ax)
roads_only.plot(ax=ax,
                   color = road_colors,
                   linewidth = line_widths,
                   legend = True)
ax.set_axis_off()
plt.axis('equal');
```

In [128]:

```
#List unique values in the df['name'] column
rds_pt1.F_CODE_DES.unique()
```

Out[128]:

```
array(['Road', 'Trail'], dtype=object)
```

In [129]:

```
#List unique values in the df['name'] column
rds_pt1.ISOCOUNTRY.unique()
```

Out[129]:

```
array(['PORTUGAL', None], dtype=object)
```

In [130]:

```
rds_pt1[rds_pt1.ISOCOUNTRY != 'PORTUGAL'].head()
```

Out[130]:

	MED_DESCRI	RTT_DESCRI	F_CODE_DES	ISO	ISOCOUNTRY	ge
713	None	None	Trail	PRT	None	(LINESTRING (-31.0820826 39.707493197
714	None	None	Trail	PRT	None	(LINESTRING (-31.2193467 39.520973255
715	Without Median	Secondary Route	Road	PRT	None	LINESTRING (-9.35409346 39.348124061
716	Without Median	Secondary Route	Road	PRT	None	LINESTRING (-28.0135142 39.026527331
717	Without Median	Primary Route	Road	PRT	None	LINESTRING (-8.96759734 38.95538621

In [131]:

```
#List unique values in the df['name'] column
rds_pt1.RTT_DESCRI.unique()
```

Out[131]:

```
array(['Secondary Route', 'Primary Route', 'Unknown', None], dtype=object)
```

In [132]:

```
rds_pt2 = rds_pt1[rds_pt1.ISOCOUNTRY == 'PORTUGAL']
```

In [133]:

```
rds_pt2.head()
```

Out[133]:

	MED_DESCRI	RTT_DESCRI	F_CODE_DES	ISO	ISOCOUNTRY	geome
0	Without Median	Secondary Route	Road	PRT	PORTUGAL	(LINESTRING (-8.2289265506142 42.13301116472...
1	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.5897780129395 42.048389438887..
2	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.5897780129395 42.048389438887..
3	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.6336392428575 42.028831529923..
4	Without Median	Primary Route	Road	PRT	PORTUGAL	LINESTRING (-8.6336392428575 42.028831529923..

In [134]:

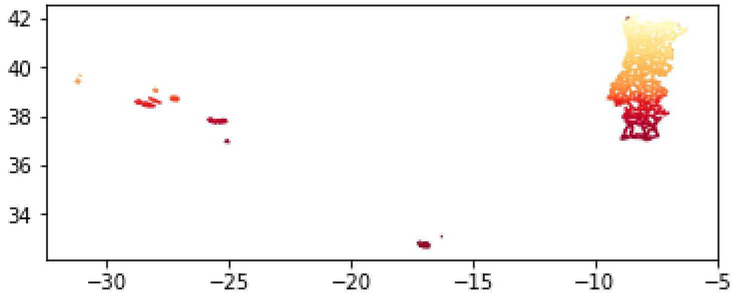
```
#https://matplotlib.org/examples/color/named_colors.html

# List of colors
color_roads = ['blue', 'red', 'darkorange', 'magenta','deepskyblue','plum',
              'cyan', 'lightsalmon','palegreen','wheat','silver','seashell',
              'linen','aliceblue','azure','mintcream', 'oldlace',
              'floralwhite','honeydew','cornsilk','lightyellow','whitesmoke',
              'lavender','lightcyan','white','ghostwhite','snow',
              'oldlace','oldlace','oldlace','oldlace']
```

In [135]:

```
rds_pt2.plot(cmap='YlOrRd')
```

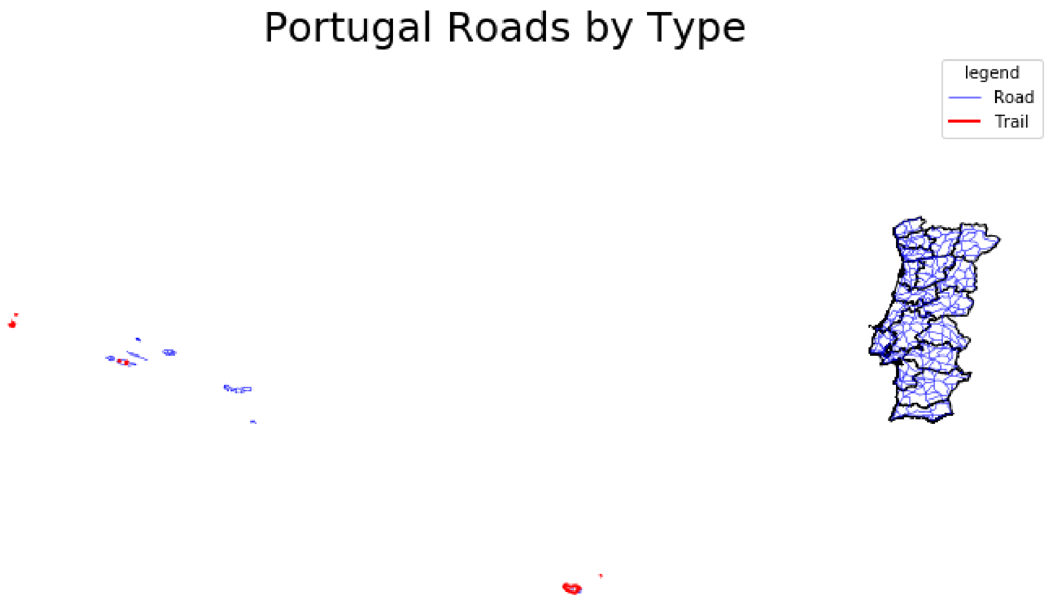
Out[135]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25cf8fbdcc0>
```



In [136]:

```
# PLOT The Roads
country_boundary = x1_table
roads_only =  rds_pt2

# make it a bit nicer using a dictionary to assign colors and line widths
road_attrs = { 'Road': ['blue',.5],
               'Trail': ['red',2]}

# plot the data
fig, ax = plt.subplots(figsize  = (12, 8))

for ctype, data in roads_only.groupby('F_CODE_DES'):
    data.plot(color=road_attrs[ctype][0],
              label = ctype,
              ax = ax,
              linewidth=road_attrs[ctype][1])

country_boundary.plot(alpha = 1, color="white", edgecolor = "black", ax = ax)
ax.legend(title="legend")
ax.set_title("Portugal Roads by Type", fontsize=25)
ax.set_axis_off()
plt.axis('equal');
```



In [137]:

```
# Reading mapas from Geopandas
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
```

In [138]:

```
#filtering Europe
world[world.continent == 'Europe'].head()
```

Out[138]:

	pop_est	continent	name	iso_a3	gdp_md_est	geometry
2	3639453.0	Europe	Albania	ALB	21810.0	POLYGON ((20.59024743010491 41.85540416113361,...
9	8210281.0	Europe	Austria	AUT	329500.0	POLYGON ((16.97966678230404 48.12349701597631,...
12	10414336.0	Europe	Belgium	BEL	389300.0	POLYGON ((3.314971144228537 51.34578095153609,...
16	7204687.0	Europe	Bulgaria	BGR	93750.0	POLYGON ((22.65714969248299 44.23492300066128,...
18	4613414.0	Europe	Bosnia and Herz.	BIH	29700.0	POLYGON ((19.00548628101012 44.86023366960916,...

In [139]:

```
# Obtain Portugal - Without Acores e Madeira
portugal = world[world.iso_a3 == 'PRT'].head()
```

In [140]:

```
portugal
```

Out[140]:

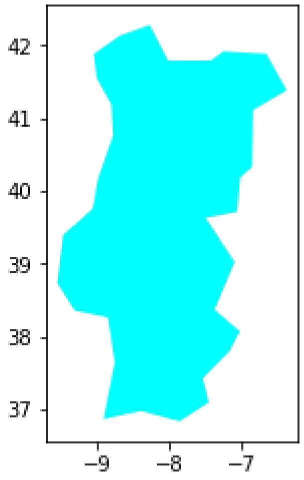
	pop_est	continent	name	iso_a3	gdp_md_est	geometry
130	10707924.0	Europe	Portugal	PRT	208627.0	POLYGON ((-9.034817674180246 41.88057058365968...

In [141]:

```
portugal.plot(color='cyan')
```

Out[141]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25cf9c6cef0>
```



In [142]:

```
# Separate Primary roads
rds_primary = rds_pt1[rds_pt1.RTT_DESCRI == 'Primary Route']
```

In [143]:

```
# Separate Secondary roads
rds_secondary = rds_pt1[rds_pt1.RTT_DESCRI == 'Secondary Route']
```

In [144]:

```
rds_secondary.head()
```

Out[144]:

	MED_DESCRI	RTT_DESCRI	F_CODE_DES	ISO	ISOCOUNTRY	geome
0	Without Median	Secondary Route	Road	PRT	PORTUGAL	(LINESTRING (-8.2289265506142 42.13301116472...
1	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.5897780129395 42.048389438887..
2	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.5897780129395 42.048389438887..
3	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-8.6336392428575 42.028831529923..
6	Without Median	Secondary Route	Road	PRT	PORTUGAL	LINESTRING (-6.8603887041730 41.841915066273..

In [181]:

```
rivers.head()
```

Out[181]:

	osm_id	name	tipo	width	geometry
0	4718597	None	river	0	LINESTRING (-8.490691699999999 38.9709642, -8....
3	12641715	None	river	0	LINESTRING (-8.8354898 39.0048754, -8.83484600...
4	12641805	None	river	0	LINESTRING (-8.8395753 39.0185739, -8.83923200...
5	23102773	Rio Ceira	river	0	LINESTRING (-8.394681500000001 40.1725883, -8....
6	23109746	Rio Ceira	river	0	LINESTRING (-8.325596000000001 40.174491, -8.3...

In [224]:

```
# Main road
line_p_roads = gpd.GeoDataFrame(rds_primary.geometry)
# Secondary road
line_s_roads = gpd.GeoDataFrame(rds_secondary.geometry)
# rail road
lines_rail_roads = gpd.GeoDataFrame(rds_pt2.geometry)
# Rivers
lines_rivers = gpd.GeoDataFrame(rivers)
# All rivers
lines_rivers_all = gpd.GeoDataFrame(rds_pt3)

poly_gdf = gpd.GeoDataFrame(portugal.geometry)
```

In [225]:

```
lines_rivers.head()
```

Out[225]:

	osm_id	name	tipo	width	geometry
index_left					
0	4718597	None	river	0	LINESTRING (-8.490691699999999 38.9709642, -8....
3	12641715	None	river	0	LINESTRING (-8.8354898 39.0048754, -8.83484600...
4	12641805	None	river	0	LINESTRING (-8.8395753 39.0185739, -8.83923200...
5	23102773	Rio Ceira	river	0	LINESTRING (-8.394681500000001 40.1725883, -8....
6	23109746	Rio Ceira	river	0	LINESTRING (-8.325596000000001 40.174491, -8.3...

In [226]:

```
# Making interset into road from continental part of Portugal to obtain only continenta
l lines
continental_p_roads = gpd.sjoin(line_p_roads, poly_gdf, op='intersects')
```

In [227]:

```
# Making interset into secondary roads road from continental part of Portugal to obtain
 only continental lines
continental_s_roads = gpd.sjoin(lines_rail_roads, poly_gdf, op='intersects')
```

In [228]:

```
# Making interset into train road from continental part of Portugal to obtain only cont
inental lines
continental_lines_rail_roads = gpd.sjoin(line_s_roads, poly_gdf, op='intersects')
```

In [229]:

```
# Making interset into rivers from continental part of Portugal to obtain only continental lines
continental_lines_rivers = gpd.sjoin(lines_rivers, poly_gdf, op='intersects')
```

In [230]:

```
continental_all_rivers =  gpd.sjoin(lines_rivers_all, poly_gdf, op='intersects')
```

In [231]:

```
# Main road lines - Portugal continental
continental_p_roads.head()
```

Out[231]:

	geometry	index_right
4	LINESTRING (-8.633639242857511 42.028831529923...	130
5	LINESTRING (-8.651361394824374 41.982833844009...	130
20	LINESTRING (-7.44277765008411 41.7486114364478...	130
23	LINESTRING (-7.44277765008411 41.7486114364478...	130
27	LINESTRING (-7.463972110044481 41.737335129468...	130

In [232]:

```
# Secondary road lines - Portugal continental
continental_s_roads.head()
```

Out[232]:

	geometry	index_right
0	LINESTRING (-8.480833136999998 42.076667815473...	130
1	LINESTRING (-8.632008566999998 42.031692430473...	130
2	LINESTRING (-8.632008566999998 42.031692430473...	130
3	LINESTRING (-6.761750183999998 41.801334316473...	130
4	LINESTRING (-8.583110864999998 41.771640715473...	130

In [233]:

```
# Train lines - Portugal continental
continental_lines_rail_roads.head()
```

Out[233]:

	geometry	index_right
0	(LINESTRING (-8.228926550614217 42.13301116472...	130
1	LINESTRING (-8.589778012939519 42.048389438887...	130
2	LINESTRING (-8.589778012939519 42.048389438887...	130
3	LINESTRING (-8.633639242857511 42.028831529923...	130
6	LINESTRING (-6.860388704173024 41.841915066273...	130

In [234]:

```
# Rivers lines - Portugal continental
continental_lines_rivers.head()
```

Out[234]:

	osm_id	name	tipo	width	geometry	index_right
0	4718597	None	river	0	LINESTRING (-8.490691699999999 38.9709642, -8....	130
3	12641715	None	river	0	LINESTRING (-8.8354898 39.0048754, -8.83484600...	130
4	12641805	None	river	0	LINESTRING (-8.8395753 39.0185739, -8.83923200...	130
5	23102773	Rio Ceira	river	0	LINESTRING (-8.394681500000001 40.1725883, -8....	130
6	23109746	Rio Ceira	river	0	LINESTRING (-8.325596000000001 40.174491, -8.3...	130

In [235]:

```
# Transform POLYGON to POINT
# copy poly to new GeoDataFrame
points_rivers = lines_rivers.copy()
# change the geometry
points_rivers.geometry = points_rivers['geometry'].centroid
# same crs
#points.crs = poly.crs
points_rivers.head()
```

Out[235]:

	osm_id	name	tipo	width	geometry
index_left					
0	4718597	None	river	0	POINT (-8.53636000877035 38.95057174670018)
3	12641715	None	river	0	POINT (-8.815323953550291 38.99793898428005)
4	12641805	None	river	0	POINT (-8.833120099449404 39.02158483033733)
5	23102773	Rio Ceira	river	0	POINT (-8.389164796075022 40.17138655096804)
6	23109746	Rio Ceira	river	0	POINT (-8.310443633606551 40.17122159181996)

In [236]:

```
# Transform POLYGON to POINT
# copy poly to new GeoDataFrame
points_rivers_all = lines_rivers_all.copy()
# change the geometry
points_rivers_all.geometry = points_rivers_all['geometry'].centroid
# same crs
#points.crs = poly.crs
points_rivers_all.head()
```

Out[236]:

	osm_id	name	tipo	width	geometry
index_left					
0	4718597	None	river	0	POINT (-8.53636000877035 38.95057174670018)
1	4856704	Rio Varosa	stream	0	POINT (-7.789857125426186 40.95773154953566)
2	10577666	None	stream	0	POINT (-6.829238689841858 41.29666055101189)
3	12641715	None	river	0	POINT (-8.815323953550291 38.99793898428005)
4	12641805	None	river	0	POINT (-8.833120099449404 39.02158483033733)

```
In [237]:
```

```
points.head()
```

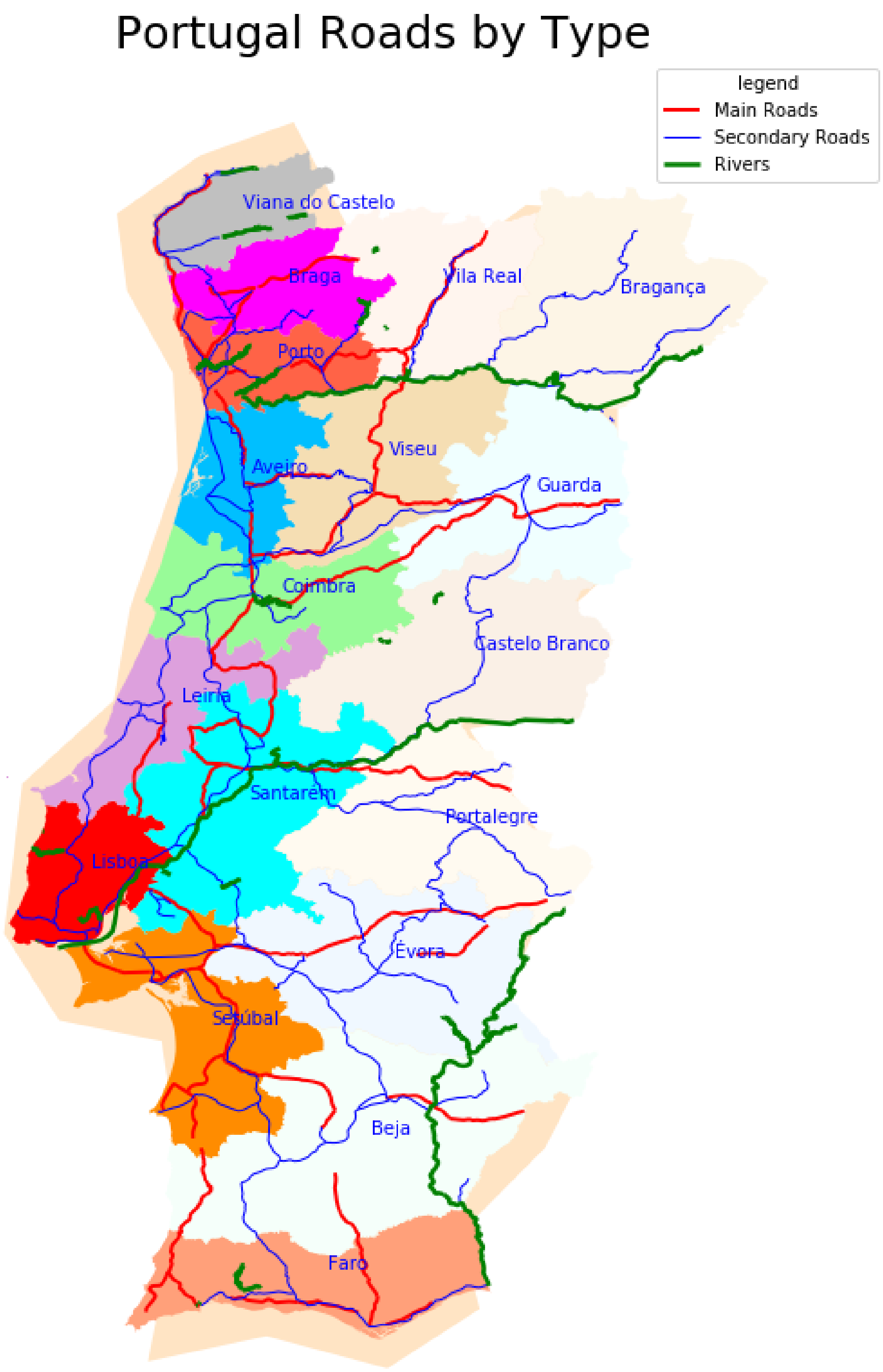
```
Out[237]:
```

	ID_0	ISO	NAME_0	ID_1	Distrito	HASC_1	CCN_1	CCA_1	TYPE_1	ENGTYPE_
10	182	PRT	Portugal	12	Lisboa	PT.LI	0	None	Distrito	District
12	182	PRT	Portugal	15	Porto	PT.PO	0	None	Distrito	District
14	182	PRT	Portugal	17	Setúbal	PT.SE	0	None	Distrito	District
3	182	PRT	Portugal	5	Braga	PT.BR	0	03	Distrito	District
1	182	PRT	Portugal	2	Aveiro	PT.AV	0	01	Distrito	District

In [254]:

```
# PLOT The Roads
country_boundary = portugal
lines_p  =  continental_p_roads
lines_s  =  continental_s_roads
lines_r  =  continental_lines_rivers

# make it a bit nicer using a dictionary to assign colors and line widths
road_color = [['red',2], ['blue',1], ['green',2.5]]

# PLOT State points
f, ax = plt.subplots(1, figsize=(10,14))
ax.set_axis_on()
f.suptitle('PORTUGAL - Roads')

# Plot all the map
country_boundary.plot(ax=ax, edgecolor = "black",
                      facecolor='blue', alpha=1,
                      linewidth=0, color='bisque', legend = True)

# Plot all the main roads
lines_p.plot(ax=ax, color=road_color[0][0],
                       label = 'Main Roads',
                       linewidth=road_color[0][1])



# Plot all the secondary roads
lines_s.plot(ax=ax, color=road_color[1][0],
                       label = 'Secondary Roads',
                       linewidth=road_color[1][1])



# Plot all the rivers
lines_r.plot(ax=ax, color=road_color[2][0],
                       label = 'Rivers',
                       linewidth=road_color[2][1])

# Put colors by district d'accord the greatest population
for i,c in enumerate(x1_table.Distrito):
    # Plot a UF map
    ax = x1_table[x1_table.index == i].plot(ax=ax,  facecolor='blue', alpha=1, linewidt
h=0, color=color_list[i])

# Plot the Districts labels
for x, y, label in zip(points.geometry.x, points.geometry.y, points.Distrito):
    ax.annotate(label, xy=(x, y), xytext=(3, 3), alpha=3, textcoords="offset points",co
lor='blue')

ax.legend(title="legend")
ax.set_title("Portugal Roads by Type", fontsize=25)
ax.set_axis_off()
plt.axis('equal');
```

PORTUGAL - Roads



In [216]:

```
rivers.head()
```

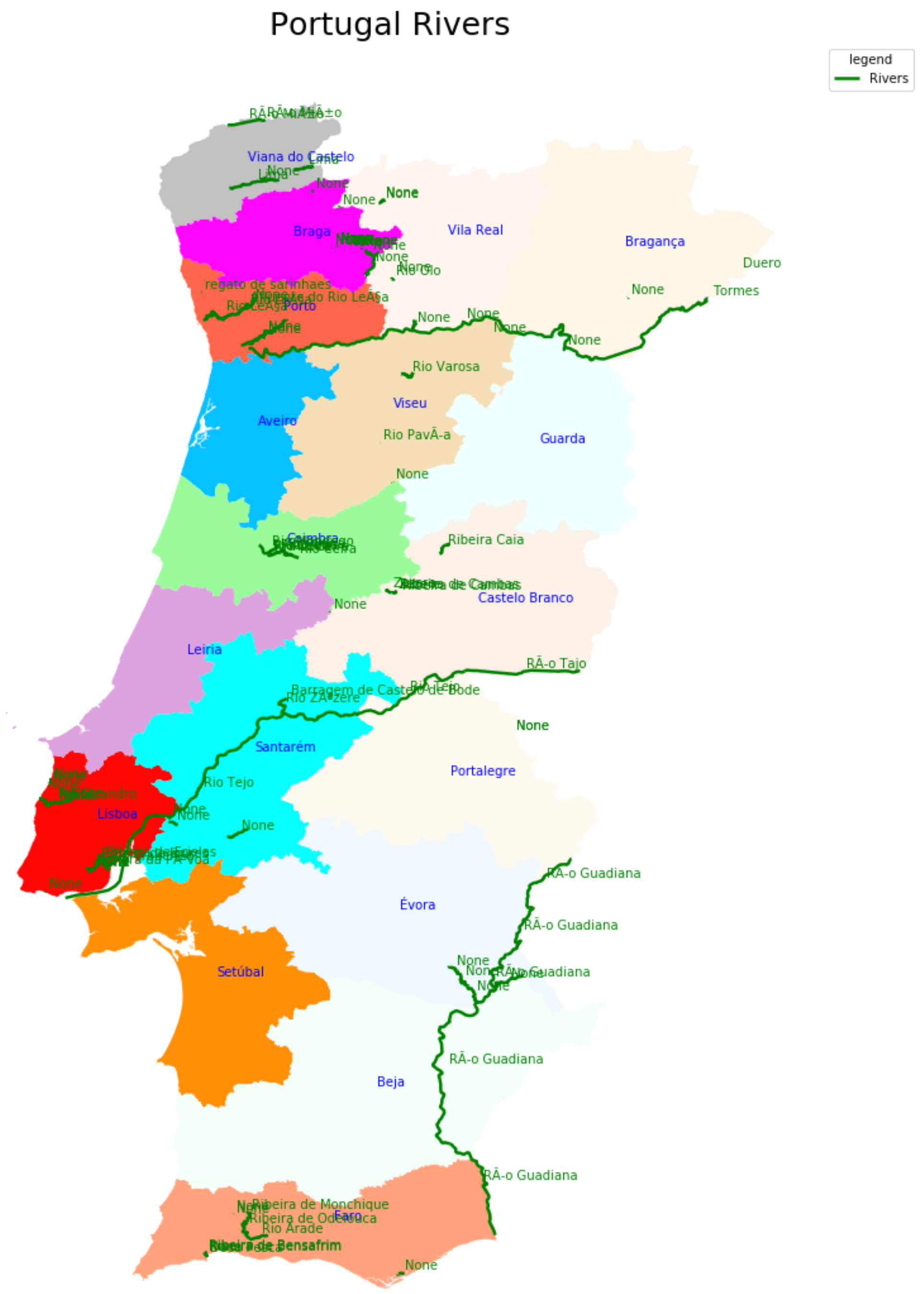
Out[216]:

	osm_id	name	tipo	width	geometry
index_left					
0	4718597	None	river	0	LINESTRING (-8.490691699999999 38.9709642, -8....
3	12641715	None	river	0	LINESTRING (-8.8354898 39.0048754, -8.83484600...
4	12641805	None	river	0	LINESTRING (-8.8395753 39.0185739, -8.83923200...
5	23102773	Rio Ceira	river	0	LINESTRING (-8.394681500000001 40.1725883, -8....
6	23109746	Rio Ceira	river	0	LINESTRING (-8.325596000000001 40.174491, -8.3...

In [256]:

```
# PLOT all The Rivers
country_boundary = portugal
lines_r  =  continental_all_rivers

# make it a bit nicer using a dictionary to assign colors and line widths
road_color = [['red',2], ['blue',1], ['green',2.5]]

# PLOT State points
f, ax = plt.subplots(1, figsize=(15,19))
ax.set_axis_on()
f.suptitle('PORTUGAL - Roads')

# Plot all the map
#country_boundary.plot(ax=ax, edgecolor = "black",
#                      facecolor='blue', alpha=1,
#                      linewidth=0, color='bisque', legend = True)

# Plot all the rivers
lines_r.plot(ax=ax, color=road_color[2][0],
                       label = 'Rivers',
                       linewidth=road_color[2][1])

# Put colors by district d'accord the greatest population
for i,c in enumerate(x1_table.Distrito):
    # Plot a UF map
    ax = x1_table[x1_table.index == i].plot(ax=ax, alpha=8, linewidth=1, color=color_li
st[i])

# Plot the Districts labels
for x, y, label in zip(points.geometry.x, points.geometry.y, points.Distrito):
    ax.annotate(label, xy=(x, y), xytext=(3, 3), alpha=3, textcoords="offset points",co
lor='blue')

# Plot the River labels
for x, y, label in zip(points_rivers_all.geometry.x, points_rivers_all.geometry.y, poin
ts_rivers_all.name):
    ax.annotate(label, xy=(x, y), xytext=(3, 3), alpha=3, textcoords="offset points",co
lor='Green')

ax.legend(title="legend")
ax.set_title("Portugal Rivers", fontsize=25)
ax.set_axis_off()
plt.axis('equal');
```

PORTUGAL - Roads



In [ ]:

```
```