SISTEMAS INFORMÁTICOS DE AYUDA A LA DECISÓN

INTEGRANTES:

- Maximiliano GRECO
- Victor ISARRE
- Alejandro MARTINEZ
- Brando SALDAÑA
- Pablo YANES
- Miaomiao YANG

PROBLEMA

Dada una empresa, cinco clientes y unos costes de transporte constantes. ¿En dónde debería situar su almacén esta empresa?

De la empresa conocemos el historial de las demandas de los clientes desde 1995 hasta 2015 y su situación geográfica (Latitud y Longitud). Además son conocidos los costes por km.

$$\min_{x_0,y_0} \sum_{i=1}^{5} w_i \cdot d_i \cdot c$$

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

DONDE:

- c: Coste unitario (€ / ud).
- w_i: Unidades demandadas de bienes (media de los últimos 3 años).
- d_i: Distancia al almacén.
- x_i: Coordenada eje x (Longitud, decimal).
- y_i: Coordenada eje y (Latitud, decimal).

con i ={1,...,5} (Clientes)

Variables de decisión: x_0, y_0

SIMULACIÓN

DEMANDA (CLIENTES)

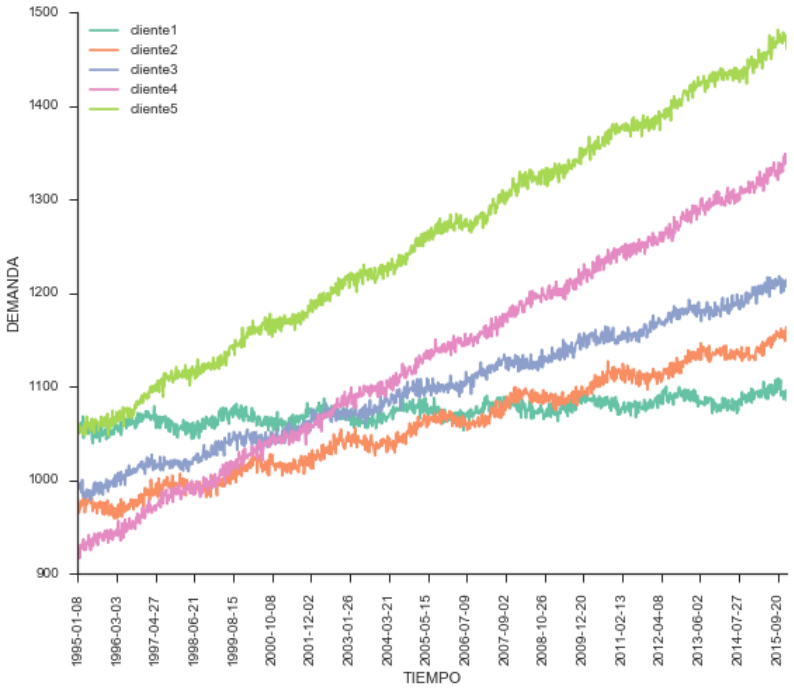
SERIE TEMPORAL TIPO:

$$y_t = \delta_0 + \delta_1 \cdot t + u_t$$

Donde:

- y_t : Ventas para cada período t.
- δ_0 : Ventas indendientes o autónomas. (Stock de Seguridad pe.)
- δ_1 : Pendiente la recta.
- u_t : Ruido ~ N(0, 5)

En este caso para darle más realismo, $\delta_0 = \delta_a + \delta_b \cdot sin(x)$, de esta forma introducimos un posible componente estacional, que depende de cuán grande sea δ_b.


diente1
diente2
diente3
diente4
diente5
DEMANDA
1500
1400
1300
1200
1100
1000
900
1995-01-08
1996-03-03
1997-04-27
1998-06-21
1999-08-15
2000-10-08
2001-12-02
2003-01-26
2004-03-21
2005-05-15
2006-07-09
2007-09-02
2008-10-26
2009-12-20
2011-02-13
2012-04-08
2013-06-02
2014-07-27
2015-09-20
TIEMPO

In [11]:
```
ventas.head()
```

Out[11]:

	cliente1	cliente2	cliente3	cliente4	cliente5
1995-01-08	1057.125631	981.139096	993.012834	921.485244	1054.929112
1995-01-15	1058.038797	965.762513	991.368924	925.506056	1061.149308
1995-01-22	1050.353065	965.344514	991.796473	926.670731	1051.103377
1995-01-29	1053.505761	968.235378	993.462416	916.929766	1051.009594
1995-02-05	1052.877501	970.913001	997.258392	930.976354	1056.968813

In [12]:
```
ventas.tail()
```

Out[12]:

	cliente1	cliente2	cliente3	cliente4	cliente5
2015-11-29	1089.545723	1156.598485	1210.117991	1337.511838	1470.752479
2015-12-06	1095.365517	1148.965002	1213.455964	1349.141883	1476.122176
2015-12-13	1095.118853	1155.781834	1207.221388	1341.626549	1469.859553
2015-12-20	1096.165708	1163.764572	1212.559956	1338.901084	1468.294863
2015-12-27	1084.971914	1150.177410	1214.876189	1347.971994	1460.618567

UBICACIÓN DE LOS CLIENTES

In [14]:
```
data_clientes
```

Out[14]:

	prov	lat	lng
cliente1	zaragoza	41.648823	-0.889085
cliente2	oviedo	43.361915	-5.849389
cliente3	huesca	42.131845	-0.407806
cliente4	madrid	40.416775	-3.703790
cliente5	bilbao	43.263013	-2.934985

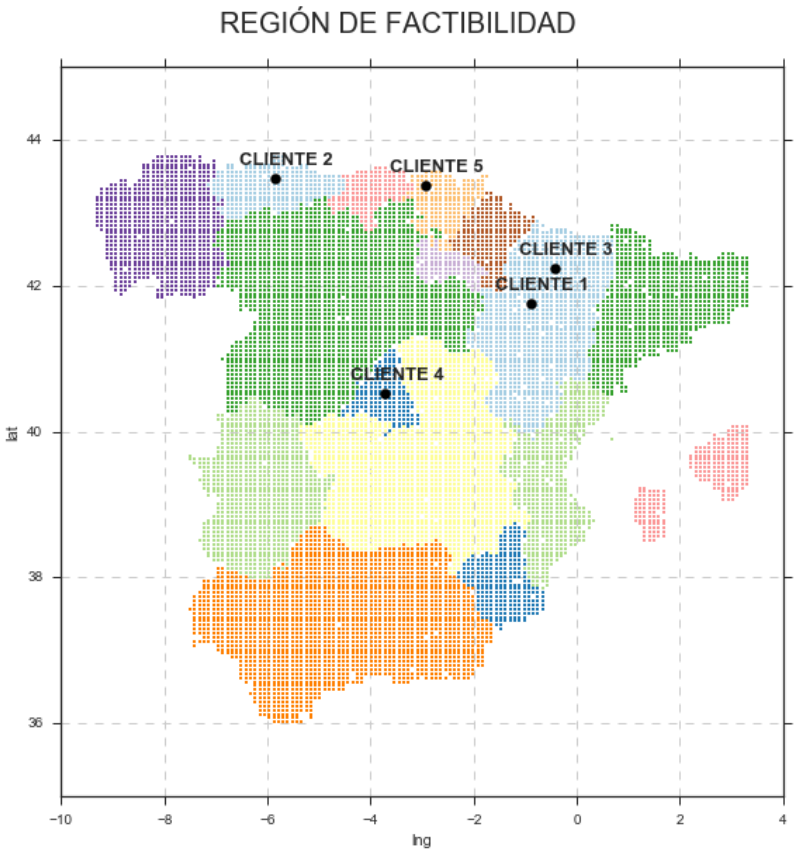
NOTA: *En los mapas físicos, las coordenadas están expresadas en grados minutos y segundos, por ejemplo Madrid sería 40°25′08″N 3°41′31″O*

Para convertir a decimal:

$$40°25′08″\ N = 40 + \frac{25}{60} + \frac{08}{3600} = 40.41888888888889$$

$$3°41′31″\ O = -(3 + \frac{41}{60} + \frac{31}{3600}) = -3.6919444444444447$$

*La longitud se multiplica por **negativo**, porque está a la **izquierda (oeste)** del punto 0,0.*


REGIÓN DE FACTIBILIDAD
CLIENTE 2
CLIENTE 5
CLIENTE 3
CLIENTE 1
CLIENTE 4
lat
lng
44
42
40
38
36
−10
−8
−6
−4
−2
0
2
4

COSTES y PESOS

COSTE (c)

Datos: Observatorio de Costes del Transporte de Mercancías por Carretera (Enero 2015), para VEHÍCULO ARTICULADO DE CARGA GENERAL 1 (http://www.fomento.gob.es/NR/rdonlyres/75019EB9-D1D4-48DD-B58C-91FBD81E8E3B/129718/ObservatorioCostesMercanciasenero2015.pdf)

In [17]:
```
c = (1.211 + 1.029) / 2
c
```

Out[17]:
```
1.12
```

PESO (w_i)

Suponemos que esta empresa ya maximiza beneficios dado los precios, es decir, la oferta igual a la demanda, siendo ésta última la previsión estimada como la media de los últimos 3 años.

$w_i = \frac{1}{T} \cdot \sum_{t=2012}^{2015}(ventas_{it})$

T = Nº de observaciones entre 2012 y 2015

In [18]:
```
ventas.head()
```

Out[18]:

	cliente1	cliente2	cliente3	cliente4	cliente5
1995-01-08	1057.125631	981.139096	993.012834	921.485244	1054.929112
1995-01-15	1058.038797	965.762513	991.368924	925.506056	1061.149308
1995-01-22	1050.353065	965.344514	991.796473	926.670731	1051.103377
1995-01-29	1053.505761	968.235378	993.462416	916.929766	1051.009594
1995-02-05	1052.877501	970.913001	997.258392	930.976354	1056.968813

In [19]:
```
ventas.loc['2012':].head()
```

Out[19]:

	cliente1	cliente2	cliente3	cliente4	cliente5
2012-01-01	1077.325321	1111.050666	1165.249280	1251.370094	1386.671676
2012-01-08	1078.676390	1110.072113	1173.512154	1264.479155	1383.848146
2012-01-15	1078.811238	1119.408951	1176.267080	1251.089708	1387.171679
2012-01-22	1082.287853	1115.102705	1161.736463	1256.147852	1375.243082
2012-01-29	1085.285927	1113.971479	1169.211906	1254.006147	1388.195087

In [20]:
```
wi = ventas.loc['2012':].mean()
```

In [22]:
```
yxi['wi'] = wi
yxi
```

Out[22]:

	lat	lng	wi
cliente1	41.648823	-0.889085	1088.188049
cliente2	43.361915	-5.849389	1133.466342
cliente3	42.131845	-0.407806	1188.070200
cliente4	40.416775	-3.703790	1297.616717
cliente5	43.263013	-2.934985	1429.070507

OPTIMIZACIÓN

In [23]:
```
def func_obj1(yx0):
    'Función Obejtivo, xy0: Tupla, pares de coordenadas'
    di = lambda yxj: vincenty(yx0, yxj).km
    dist = yxi.apply(di, axis=1)
    return sum(dist * wi) * c
```

In [25]:
```
res1 = brute(func_obj1, rrange)
```

In [30]:
```
r1(res1)
```

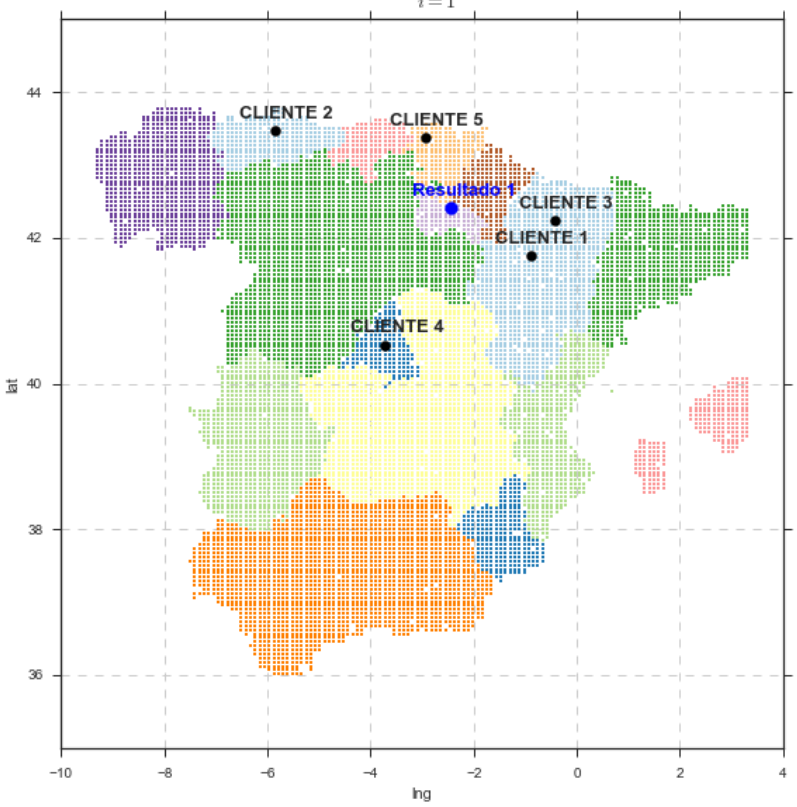
Out[30]:

RESULTADO:

- **LAT:** 42.3015617419
- **LONG:** -2.43768347258
- **CCAA:** RI: La Rioja
- **FUNCIÓN OBJETIVO:** 1312334.97165

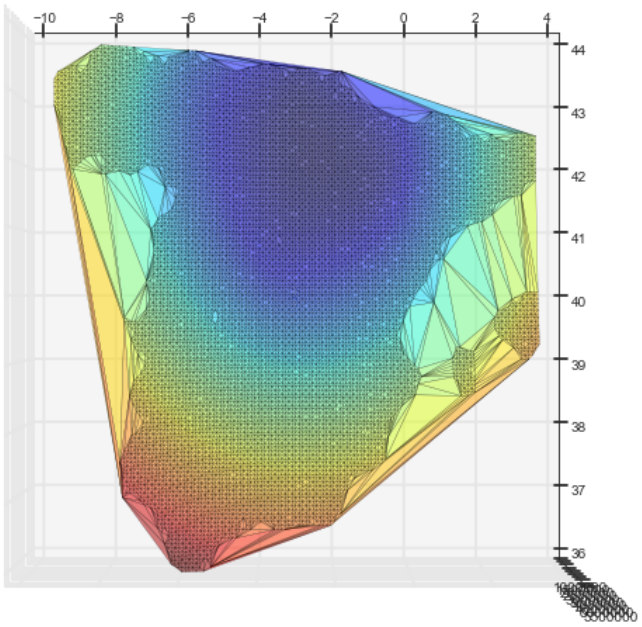
RESULTADO DE LA PRIMERA OPTIMIZACIÓN

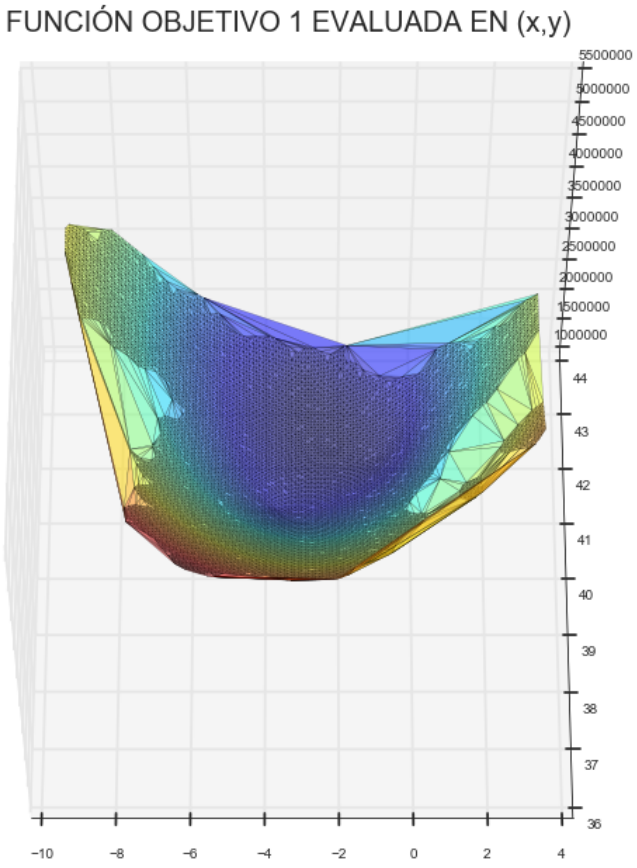
$$\min_{x_0,\, y_0} \sum_{i=1}^{5} w_i \cdot d_i \cdot c$$



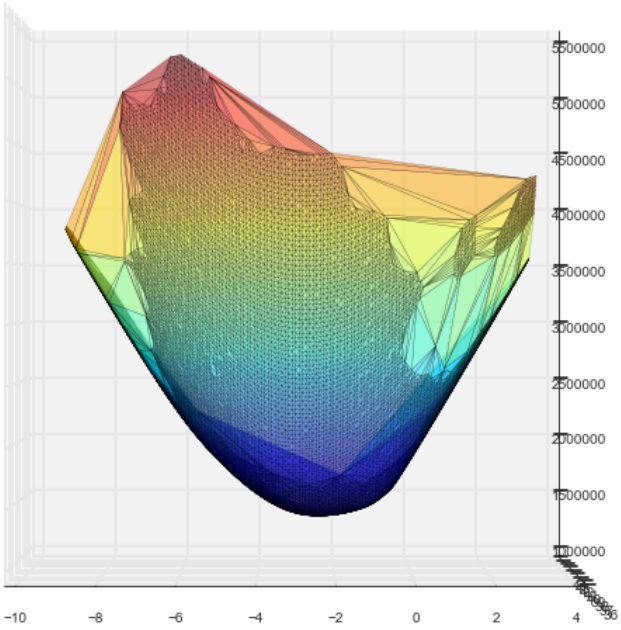
ANÁLISIS GRÁFICO

FUNCIÓN OBJETIVO 1 EVALUADA EN (x,y)




FUNCIÓN OBJETIVO 1 EVALUADA EN (x,y)
5500000
5000000
4500000
4000000
3500000
3000000
2500000
2000000
1500000
1000000
44
43
42
41
40
39
38
37
36
−10
−8
−6
−4
−2
0
2
4

FUNCIÓN OBJETIVO 1 EVALUADA EN (x,y)



INTRODUCCIÓN DE UNA RESTRICCIÓN ADICIONAL

Al problema anterior, le agregamos una variable adicional de tipo fiscal. Cada comunidad autónoma pasará a cobrar un impuesto por unidad vendida, el tipo impositivo dependerá de la comunidad autónoma, de forma que los costes para la empresa cambien y por tanto la elección óptima.

PROBLEMA INICIAL + RESTRICCIÓN

$$\min_{x_0, y_0} \sum_{i=1}^{5} w_i \cdot d_i \cdot c(1+t)$$

s.a. $(x_0, y_0) \in$ España

DONDE:

- c: Coste unitario (€/ud).
- w_i: Unidades demandadas de bienes (media de los últimos 3 años).
- d_i: Distancia al almacén (Km).
- x_i: Coordenada eje x (Longitud, decimal).
- y_i: Coordenada eje y (Latitud, decimal).
- t**: Impuesto por unidad vendida.**

Variables de decisión (Coordenadas): x_0, y_0

FUNCIÓN OBJETIVO CON IMPUESTOS

In [42]:
```
def func_obj2(yx0):
    'Función Obejtivo, xy0: Tupla, pares de coordenadas'
    di = lambda yxj: vincenty(yx0, yxj).km
    dist = yxi.apply(di, axis=1)
    res = sum(dist * wi) * c * (1 + impuesto_ca(yx0))
    return res
```

In [47]:
```
r2(res2)
```

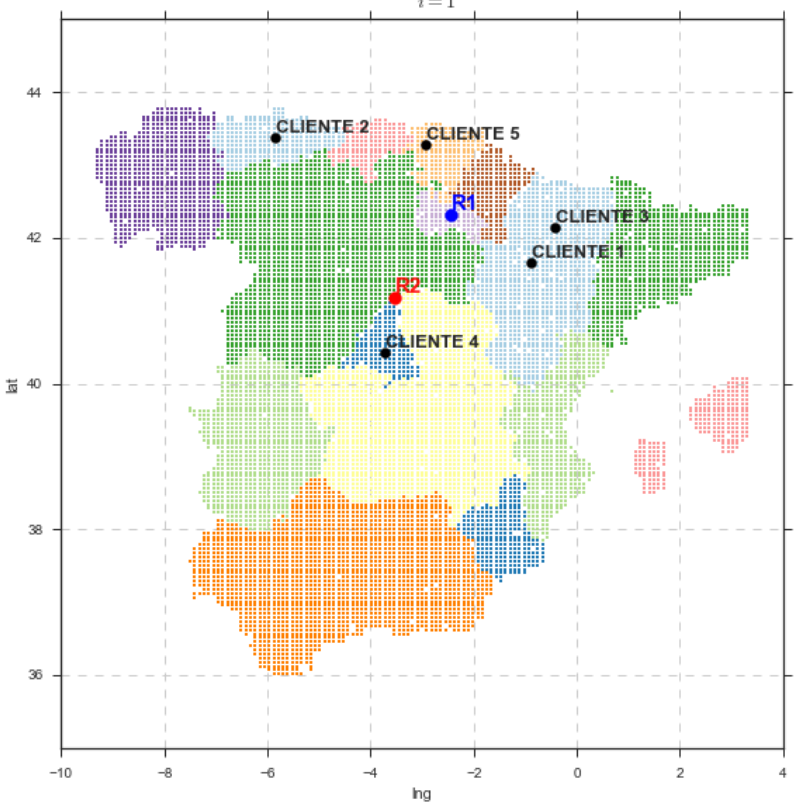
Out[47]:

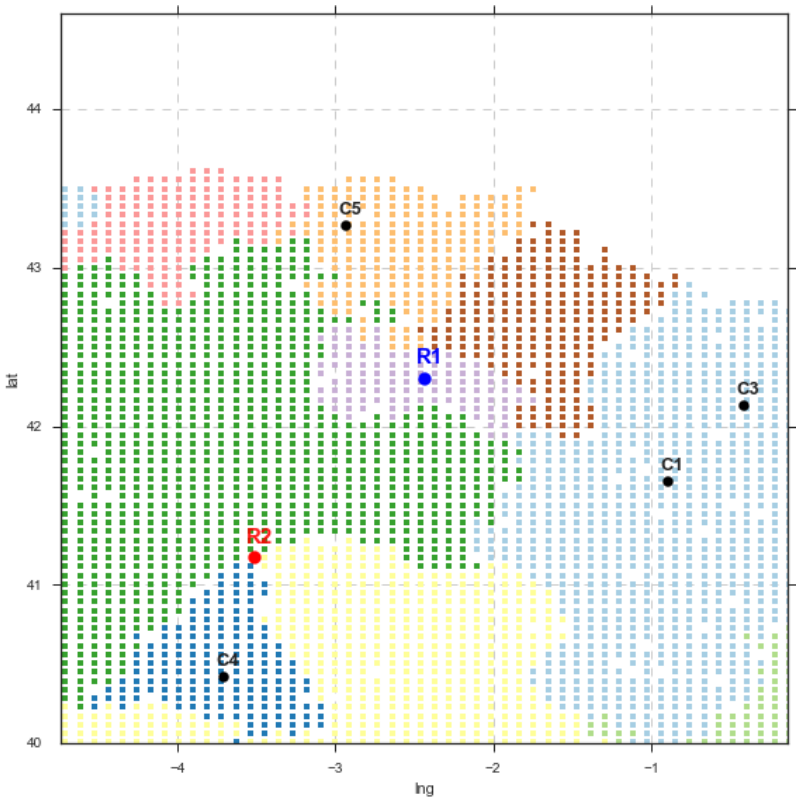
RESULTADO:

- **LAT:** 41.1707155712
- **LONG:** -3.51401089789
- **CCAA:** MD: Madrid, Comunidad de
- **FUNCIÓN OBJETIVO:** 1562079.96239

RESULTADO DE LA SEGUNDA OPTIMIZACIÓN

$$\min_{x_0,\, y_0} \sum_{i=1}^{5} w_i \cdot d_i \cdot c \cdot (1+t)$$




C5
R1
C3
C1
R2
C4
lat
lng
44
43
42
41
40
−4
−3
−2
−1

In [65]:
```
pd_res = pd.DataFrame.from_dict(resultados).T
pd_res.columns = ['lat', 'lng', 'fun_obj', 'ccaa']
#pd_res.to_csv('resultados.csv')
Markdown('# RESÚMEN\n' + pd_res.to_html())
```

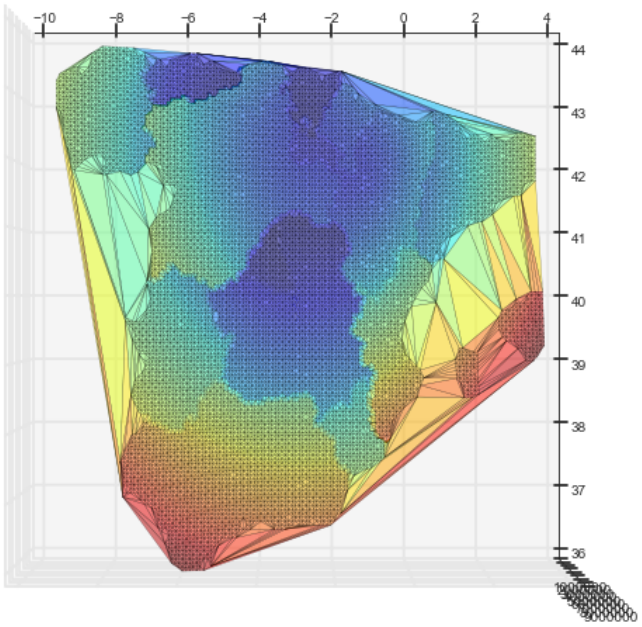
Out[65]:

RESÚMEN

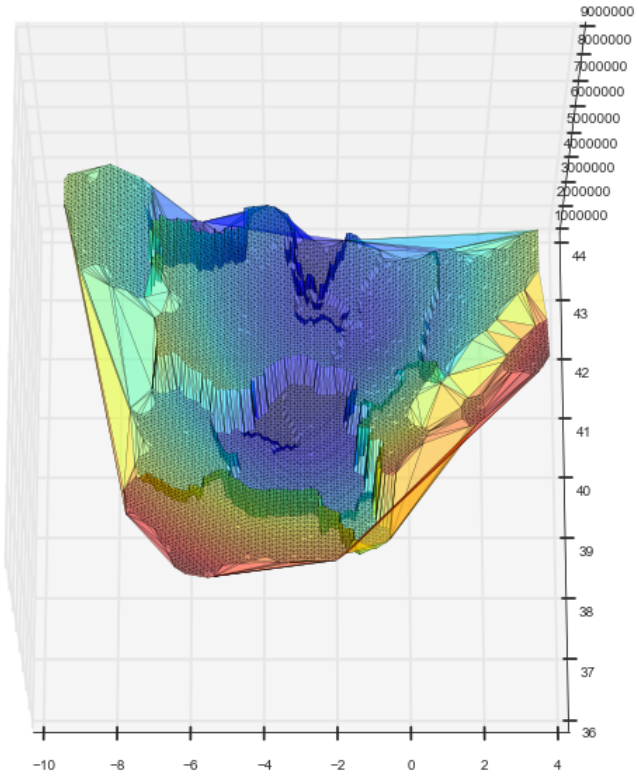
	lat	lng	fun_obj	ccaa
res1	42.3016	-2.43768	4.68691e+06	RI
res2	41.1707	-3.51401	5.57886e+06	MD

ANÁLISIS GRÁFICO

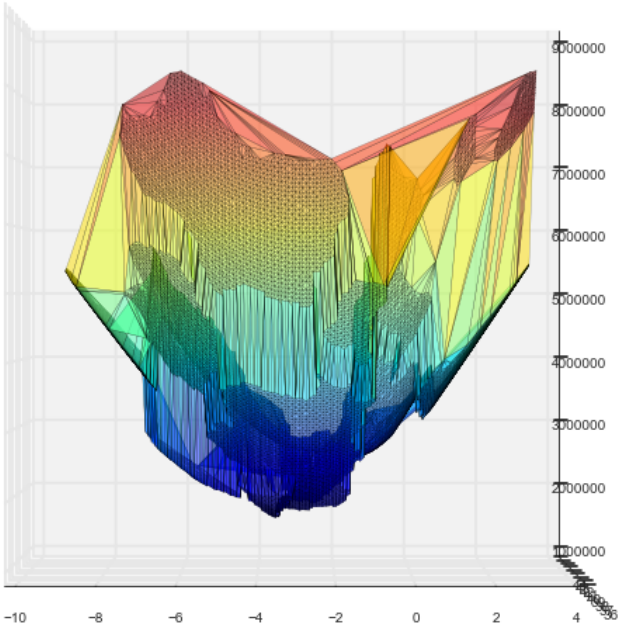
FUNCIÓN OBJETIVO 2 EVALUADA EN (x,y)



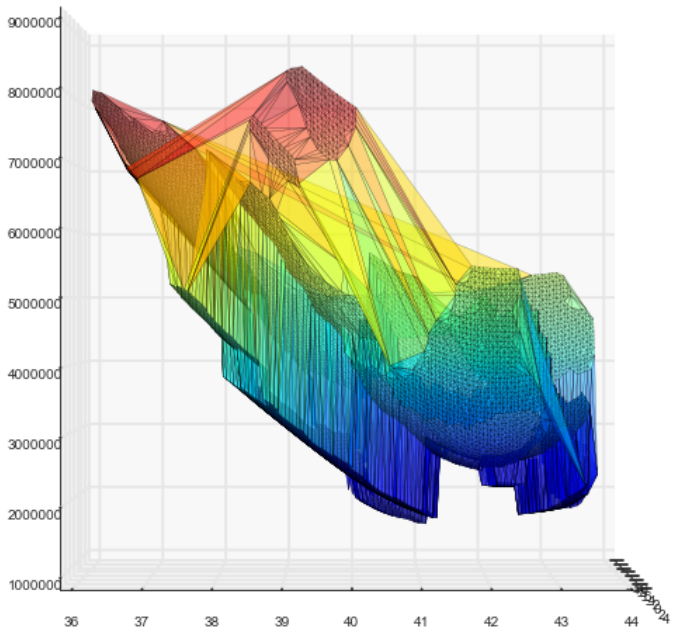
FUNCIÓN OBJETIVO 2 EVALUADA EN (x,y)



FUNCIÓN OBJETIVO 2 EVALUADA EN (x,y)



FUNCIÓN OBJETIVO 2 EVALUADA EN (x,y)



In [66]:
```
ca_xys.head()
```

Out[66]:

	codigo	nombre	name.1	ti
name				
AN	ES-AN	Andalucía	AN	0.582215
AR	ES-AR	Aragón	AR	0.805577
AS	ES-AS	Asturias, Principado de	AS	0.034853
CN	ES-CN	Canarias	CN	0.642867
CB	ES-CB	Cantabria	CB	0.990424

MULTICRITERIO

MINIMIZAR EL PAGO IMPOSITIVO

Objetivo 1:

$$\min_{x_0,y_0} Z_1 = \sum_{i=1}^{5} w_i \cdot t(x,y)$$

MINIMIZAR LOS COSTES DE TRASNSPORTE

Objetivo 2:

$$\min_{x_0,y_0} Z_2 = \sum_{i=1}^{5} w_i \cdot d_i(x,y) \cdot c$$

Programación por metas MINIMAX o Tchebychev

$\min\{\max D\}$

sa:

- $Z_1 + n_1 - p_1 \leq 0.2$
- $Z_2 + n_2 - p_2 \leq 200000$
- $p_1/0.2 \leq D$
- $p_2/2000000 \leq D$
- $x, y \in$ España

$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$

DONDE:

- c: Coste unitario (€/ud).
- w_i: Unidades demandadas de bienes (media de los últimos 3 años).
- d_i: Distancia al almacén (Km).
- x_i: Coordenada eje x (Longitud, decimal).
- y_i: Coordenada eje y (Latitud, decimal).
- $\boldsymbol{t_j}$**: Impuesto por unidad vendida.** j={1-16}

Variables de decisión (Coordenadas): x_0, y_0

In [67]:
```
p1 = lambda yx: (impuesto_ca(yx) - 0.2) / 0.2
p2 = lambda yx: (func_obj1(yx) - 2000000) / 2000000

def fobj3(yx):
    return max(p1(yx),p2(yx))
```

In [68]:
```
res3 = fmin(fobj3, res1)
```

```
Optimization terminated successfully.
         Current function value: 1.348887
         Iterations: 72
         Function evaluations: 139
```

In [75]:
```
res3
```

Out[75]:
```
array([ 42.47386784,  -2.53406359])
```

In [76]:
```
p1(_), p2(_)
```

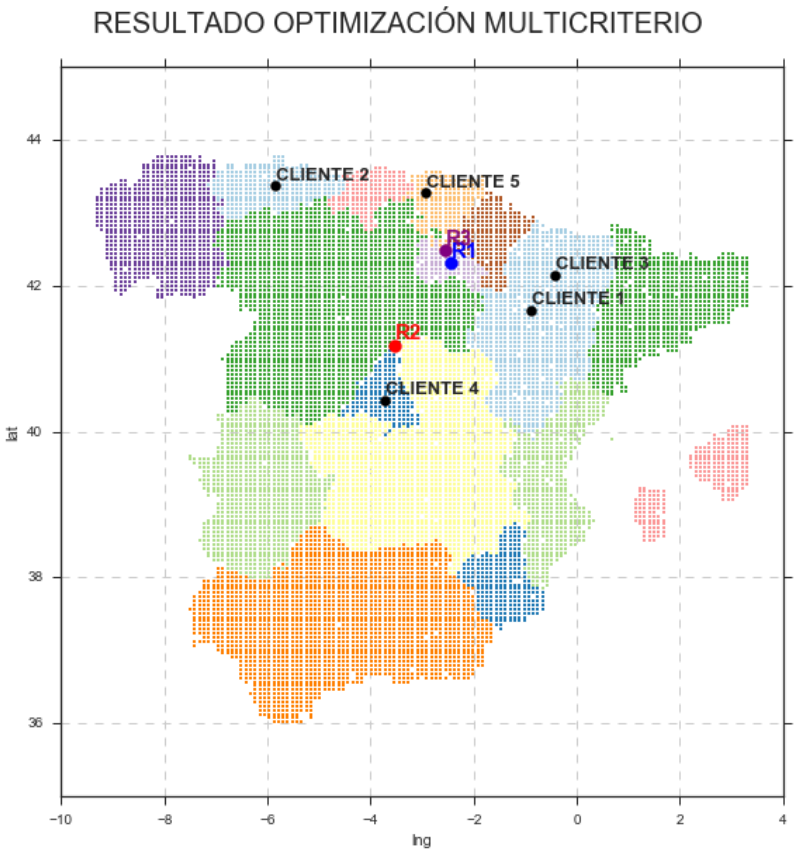
Out[76]:
```
(0.63608263254145392, 1.3488872033786961)
```

In [77]:
```
res_3(res3)
```

Out[77]:

RESULTADO

- **LAT:** 42.4738678398
- **LONG:** -2.53406358903
- **CCAA:** PV
- **ti:** 0.327216526508
- **FUNC. OBJ.:** 4699782.34212


RESULTADO OPTIMIZACIÓN MULTICRITERIO
CLIENTE 2
CLIENTE 5
R3
R1
CLIENTE 3
CLIENTE 1
R2
CLIENTE 4
lat
lng
44
42
40
38
36
−10
−8
−6
−4
−2
0
2
4


Google maps

In [80]:
```
IFrame("https://www.google.com/maps/d/u/0/embed?mid=zTtXepJtMlLc.k6VXMPFTI7DI", width=
940, height=580)
```

Out[80]:

SIAD ☆

In [82]:
```
data_clientes
```

Out[82]:

	prov	lat	lng	dist1	dist2
cliente1	zaragoza	41.648823	-0.889085	147.410125	225.796998
cliente2	oviedo	43.361915	-5.849389	302.771967	310.400117
cliente3	huesca	42.131845	-0.407806	168.657816	279.884499
cliente4	madrid	40.416775	-3.703790	234.601556	85.243187
cliente5	bilbao	43.263013	-2.934985	114.295030	237.271861

LIMITACIONES Y AMPLIACIONES

Errores de simplificación

- Se supone territorio es plano, no se tiene en cuenta la orografía. Se podría incluir la altitud en el análisis, para hacer el problema más real.

Distancia y Orografía

- La distancia entre cada punto es la recta que las une, introducir la distancia entre provincias en carretera sería más realista para hallar el lugar óptimo, pero muy costoso.
- Se ha usado el algoritmo de Vicenty para calcular la distancia entre dos puntos.[1][2] (Español)

---

1. https://en.wikipedia.org/wiki/Vincenty%27s_formulae (https://en.wikipedia.org/wiki/Vincenty%27s_formulae)↩
2. https://es.wikipedia.org/wiki/F%C3%B3rmulas_de_Vincenty (https://es.wikipedia.org/wiki/F%C3%B3rmulas_de_Vincenty)↩

Costes no reales.

- Los tipos impositivos han sido generados aleatoriamente a partir de una distribución uniforme 0,1. Bastaría sustituirlos por los reales para tener un resultado que se ajuste aún más a la realidad.

FIN

¡GRACIAS!