		pd.read_c	sv('E://Ginu_St	tudyMaterial	s//Sem2//	Dissertatio	on//PRP.csv')					
[2]: [3]: t[3]:	data	date_of_sale	address	postal_code	county	price mar	ket_price VAT_	exclusive p	roperty_description	property_size_description	province	month_year	year
	1	01/01/2010	5 Braemor Drive, Churchtown, Co.Dublin 134 Ashewood Walk, Summerhill Lane, Portlaoise	NaN NaN		343000 185000	No No	No Yes	Second-Hand Dwelling house /Apartment New Dwelling house /Apartment	NaN greater than or equal to 38 sq metres and less	Leinster Leinster	2010-01	
	3	04/01/2010	1 Meadow Avenue, Dundrum, Dublin 14 1 The Haven, Mornington	NaN NaN		438500	No	No No	Second-Hand Dwelling house /Apartment Second-Hand Dwelling house /Apartment	NaN	Leinster	2010-04	
		04/01/2010	11 Melville Heights, Kilkenny LACKEN,	NaN 	Kilkenny 		No	No 	Second-Hand Dwelling house /Apartment Second-Hand	NaN	Leinster 	2010-04	
	516581 516582	28/01/2022	MULTYFARNHAM, MULLINGAR LARCH HILL, COLMAN, FETHARD SHERRYS	NaN NaN	Westmeath		No	No	Dwelling house /Apartment Second-Hand Dwelling house /Apartment	NaN NaN	Leinster Munster	2022-01	
	516583 516584	28/01/2022	WOOD, BELLEWSTOWN, CO MEATH ST JUDES, STONEYFORD, KILKENNY	NaN NaN	Meath Kilkenny	450000 242000	No	No	Second-Hand Dwelling house /Apartment Second-Hand Dwelling house /Apartment	NaN	Leinster Leinster	2022-01	
	516585 516586 ro	28/01/2022 ows × 13 colu	SYLVAN, DUBLIN ROAD, BRAY umns	NaN	Wicklow	620000	No	No	Second-Hand Dwelling house /Apartment	NaN	Leinster	2022-01	2022
[4]: t[4]:	countie array([es 'Dublin',	'county'].uniqu 'Laois', 'Meath lare', 'Sligo',	h', 'Kilkenn									
[5]:		'Roscommon 'Offaly', 'Monaghan'	y = [data['price	'Mayo', 'Don terford', 'L dtype=object	egal', 'L outh', 'k)	ongford', (erry', 'We	'Galway', stmeath',	ounty in c	ounties]				
t[5]:	median_ array([:	_per_count 308370., 1 140000., 1	y = np.asarray(y .39995., 233480 .10000., 104225 .01743., 84000	., 160000., ., 125000.,	150000., 275000.,	89000., 1	50000.,						
[6]:	q=[] for pri		42500., 138000 ian_per_county:		87000.])								
[7]: [8]:			d.DataFrame(q) unty'] = counti	ies									
[9]: [10]:	county_	_price.ren	ame(columns={0: ocoder import 0	:'price'},in		e)							
[11]:	key =	'40d783cbf er = OpenC	75143b48b8528d1 ageGeocode(key) # create empty	1804a3ccd'		key from:	https://ope	encagedata	.com				
	list_lo	ong = []	n county_price.		# iterat	e over rows	s in datafran	те					
	que #10 #qu res	ery = str(oc = row[' uery = str sults = ge	<pre>temp_add'] (loc) ocoder.geocode(</pre>										
	lor lis	ng = resul st_lat.app st_long.ap	<pre>s[0]['geometry' ts[0]['geometry' end(lat) pend(long) umns from lists</pre>	/']['lng']									
	county_	_price[<mark>'la</mark>	t'] = list_lat n'] = list_long										
[12]: [12]:	0 3083	orice co		lon -6.260273									
	 1 13999 2 23344 3 16000 4 15000 	80.0 M 00.0 Kilk	Meath 53.649784 tenny 52.651022	-7.398034 -6.588529 -7.248495 -8.630124									
	5 140006 200007 140008 11000	00.0	Cork 51.897077 Clare 52.857258	-6.825145 -8.465467 -8.937436 -8.730543									
	9 1042210 1250011 2750012 8900	00.0 Tippe	erary 52.684821 cklow 52.958147	-3.345742 -7.898128 -6.381971 84.616516									
	13 1500014 1100015 1017416 8400	00.0 M 43.0 Dor 00.0 Long	Mayo 53.908706 negal 54.920754 gford 53.731985	-6.606516 -9.298305 -7.952385 -7.695351									
	17 1800018 1300019 2500020 14200	00.0 C 00.0 Ki 00.0 Wate	Offaly 53.136172 Idare 53.154364 erford 52.244963	-9.049060 -7.810341 -6.818418 -7.101788									
	21 1700022 1425023 1380024 11800	00.0 H 00.0 Westm 00.0 Mona	Kerry 52.145334 neath 53.557790 ghan 54.161066	-6.532050 -9.517401 -7.347856 -6.946365									
13]:		method		-8.052478									
	# 0 geo # 3	create a no ojson = {' loop throu r _, row i	n(county_price, ew python dict type':'FeatureC gh each row in n county_price. a feature temp	to contain Collection', the datafra iterrows():	our geojs ' <mark>feature</mark> me and co	son data, us es':[]}	sing geojson						
		feature	<pre>= {'type':'Feat 'properties' 'geometry':{ n the coordinat</pre>	ture', ':{}, ['type':'Poi 'coordinate	nt',								
							t]]						
		feature[# for ea for prop feat	<pre>'geometry']['co ch column, get in properties: ure['properties</pre>	the value a : s'][prop] =	nd add it	as a new :	feature prope		ncido our diet				
14]:		feature[# for ea for prop feat # add th geojson[turn geojs	ch column, get in properties: ure['properties is feature (aka 'features'].app	the value a s'][prop] = a, converted pend(feature	nd add it row[prop] datafram	as a new :	feature prope		nside our dict				
	cols = geojsor #geojsor fig = p	<pre>feature[# for ea for prop feat # add th geojson[turn geojs ['price', n = df_to_ on = count on = df_to ox.choropl date_geos(date_layou</pre>	ch column, get in properties: ure['properties is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election geojson(county) eth(county_price	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() price) ce, geojson= 'county", fe "mercator" ations", vis	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope		nside our dict				
[14]:	cols = geojsor #geojsor fig = p	<pre>feature[# for ea for prop feat # add th geojson[turn geojs ['price', n = df_to_ on = count on = df_to ox.choropl date_geos(date_layou</pre>	ch column, get in properties: ure['properties is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election geojson(county_ eth(county_price_ locations=" projection=) fitbounds="loca	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() price) ce, geojson= 'county", fe "mercator" ations", vis	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict		→ ::::		rice ii
	cols = geojsor #geojsor fig = p	<pre>feature[# for ea for prop feat # add th geojson[turn geojs ['price', n = df_to_ on = count on = df_to ox.choropl date_geos(date_layou</pre>	ch column, get in properties: ure['properties is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election geojson(county_ eth(county_price_ locations=" projection=) fitbounds="loca	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() price) ce, geojson= 'county", fe "mercator" ations", vis	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict		→ ::::		300 250
	cols = geojsor #geojsor fig = p	<pre>feature[# for ea for prop feat # add th geojson[turn geojs ['price', n = df_to_ on = count on = df_to ox.choropl date_geos(date_layou</pre>	ch column, get in properties: ure['properties is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election geojson(county_ eth(county_price_ locations=" projection=) fitbounds="loca	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() price) ce, geojson= 'county", fe "mercator" ations", vis	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict		♣ ::::	→ + □ pr	250
	cols = geojsor #geojsor fig = p	<pre>feature[# for ea for prop feat # add th geojson[turn geojs ['price', n = df_to_ on = count on = df_to ox.choropl date_geos(date_layou</pre>	ch column, get in properties: ure['properties is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election geojson(county_ eth(county_price_ locations=" projection=) fitbounds="loca	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() price) ce, geojson= 'county", fe "mercator" ations", vis	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict		+ ::::		
	cols = geojsor #geojsor fig = p	<pre>feature[# for ea for prop feat # add th geojson[turn geojs ['price', n = df_to_ on = count on = df_to ox.choropl date_geos(date_layou</pre>	ch column, get in properties: ure['properties is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election geojson(county_ eth(county_price_ locations=" projection=) fitbounds="loca	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() price) ce, geojson= 'county", fe "mercator" ations", vis	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict				250
15]:	cols = geojsor #geojso fig = p fig.upo fig.sho	feature[# for ear for prop feat # add the geojson[turn geojse ['price', n = df_to_ ox.chorople date_geos(date_layou ow()	ch column, get in properties: ure['properties' is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election _geojson(county_ eth(county_price	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() oprice) ce, geojson= 'county", fe ="mercator" ations", vis 0, "t":0, "l":	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict				250 200 150
15]:	cols = geojsor #geojso #geojso fig = p fig.upo fig.upo fig.sho	feature[# for ear for prop feat # add th geojson[turn geojs ['price', n = df_to_ on = count on = df_to_ ox.choropl date_geos(date_layou ow() arice co	ch column, get in properties: ure['properties' is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election _geojson(county_ eth(county_price	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() oprice) ce, geojson= 'county", fe ="mercator" ations", vis 0, "t":0, "l":	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict				250 200 150
[15]:	cols = geojsor #geojsor #geojsor fig = p fig.upo fig.upo fig.sho	feature[# for ear for prop feat # add the geojson[turn geojse ['price', n = df_to_ ox.chorople date_geos(date_layou ow()) # add the geojson[turn geojse fon = count on = df_to_ ox.chorople date_geos(date_layou ow()) # add the geojson[# add the geoj	ch column, get in properties: ure['properties' is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election_geojson(county) eth(county_price_	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() v_price) ce, geojson= 'county", fe ="mercator" ations", vis 0, "t":0, "l":	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict				250 200 150
[15]:	Cols = geojsor #geojsor #geojsor fig = p fig.upo fig.upo fig.sho 0 3083 1 13999 2 23348 3 16000	feature[# for ear for prop feat # add the geojson[turn geojse ['price', n = df_to_ ox.chorople date_geos(date_layou ow()) # add the geojson[turn geojse # add the geoson geograph # add the geoson geoso	ch column, get in properties: ure['properties' is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.electic geojson(county) eth(county_price_	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols con_geojson() ce, geojson= 'county", fe "mercator" ations", vis conductor, "l": ations", vis conductor, "l": conductor, "lon'] co	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict				250 200 150
15]:	Cols = geojsor #geojsor #geojsor fig = fi fig.upo fig.upo fig.sho 0 3083 1 1399 2 23348 3 16000 4 15000 5 14000 6 20000 7 14000 9 1042 10 12500 11 27500	feature[# for ear for prop feat # add the geojson[turn geojs ['price', n = df_to ox.chorople date_geos(date_layou ow()) date_geos(date_layou ow()) date_geos(date_layou ow()) Constant C	ch column, get in properties: ure['properties' ure['properties' ure['properties' is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.election_geojson(county) eth(county_price_locations="projection=") fitbounds="locations="locati	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols ce, geojson= 'county", fe "mercator" ations", vis b, "t":0, "l": ations", vis cols co	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict				250 200 150
15]:	Cols = geojsor #geojsor #geojsor fig = p fig.upo fig.upo fig.sho fig.sho 0 3083 1 13999 2 23344 3 16000 4 15000 5 14000 6 20000 7 14000 8 11000 9 10425 10 12500 11 27500 11 27500 11 10174	feature[# for ear for prop feat # add the geojson[turn geojse ['price', n = df_to ox.chorople date_geos(date_layou ow()) date_geos(date_layou ow()) date_geos(date_layou ow()) Can con con con con con con con con con co	ch column, get in properties: ure['properties' ure['properties' is feature (aka 'features'].app on 'county', 'lat geojson(county y_price.electic _geojson(county) eth(county_price _locations=' projection=) fitbounds="locat t(margin={"r":6" Alaois 52.998458 Aleath 53.649784 Aleany 52.651022 Alerick 52.661252 Aleany 52.661252 Aleany 52.690789 Cork 51.897077 Clare 52.857258 Sligo 54.192986 Aleany 48.672173 Alerary 52.684821 Aleany 52.958147 Aleany 53.908706 A	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols ce, geojson= 'county", fe "mercator" ations", vis b, "t":0, "l": ations", vis c', "t":0, "l": ations", vis cols cols cols cols cols cols cols col	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict				250 200
15]:	Cols = geojsor #geojsor #geojsor fig = fi fig upor fig.upor fig.sho of a 3083 1 1399 2 23348 3 16000 4 15000 5 14000 6 20000 7 14000 9 1042 10 12500 11 27500 12 8900 13 15000 14 11000 15 1017 16 8400 17 18000 18 13000 19 25000	# for ear for prop feat # add the geojson[turn geojse ['price', n = df_to ox.chorople date_geos(date_layou ow())	ch column, get in properties: ure['properties: ure['properties: ure['feature (aka'features'].app on	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols on_geojson() /price) ce, geojson= 'county", fe "mercator" ations", vis b, "t":0, "l": ations", vis colored condition of the colored ations of the colored ations of the colored colo	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict				250 200
15]:	Cols = geojsor #geojsor #geojsor fig = fi fig = fi fig.upo fig.upo fig.upo fig.sho 0 3083 1 13999 2 23344 3 16000 4 15000 5 14000 6 20000 7 14000 8 11000 9 10425 10 12500 11 27500 12 8900 14 11000 15 10174 16 8400 17 18000 18 13000 19 25000 20 14200 21 17000 22 14250 23 13800	# for each for proposed feature [# for each for proposed feature [# add the geo json [turn	Ch column, get in properties: ure['properties: ure['prope	the value a s'][prop] = a, converted pend(feature c', 'lon'] price, cols c, geojson= 'county", fe "mercator" ations", vis c, "t":0, "l": ations", vis c, "t*:0, "l": c.8.465467 c.8.937436 c.8.730543 c.3.345742 c.7.898128 c.6.381971 c.6.666516 c.9.298305 c.7.952385 c.7.695351 c.9.049060 c.7.810341 c.6.818418 c.7.101788 c.6.532050 c.9.517401 c.7.347856	nd add it row[prop] datafram) geojson, atureidke	color="pricey="propert:	feature prope the list of a		nside our dict				250 200
16]: 16]:	Cols = geojsor #geojsor #geojsor fig = fi fig.upo fig.upo fig.sho 2 fig.upo fig.sho 3 1 13999 2 23344 3 16000 4 15000 5 14000 6 20000 7 14000 9 10425 10 12500 11 27500 12 8900 13 15000 14 11000 15 10174 16 8400 17 18000 18 13000 19 25000 20 14200 21 17000 22 14250 23 13800 24 11800 25 8700	# for each for prop feat # add the geojson [turn geojs for each for prop feat # add the geojson [turn geojs for each f	Ch column, get in properties: ure['properties: ure['properties: ure['properties: ure['properties: ure['properties: ure['s feature (aka' features'].app on	the value as s'][prop] = a, converted bend(feature bend(feature bend(feature bend(feature))]	nd add it row[prop] datafram) geojson, atureidke ible=Fals 0,"b":0})	color="prices" color="prices"	ce", ies.county",	features i	nside our dict				250 200 150
15]: 16]:	Cols = geojsor #geojsor #geojsor fig = fi fig.upo fig.upo fig.sho % 1 3999 2 23348 3 16000 4 15000 5 14000 6 20000 7 14000 8 11000 9 10425 10 12500 11 27500 12 8900 13 15000 14 11000 15 1017 16 8400 17 18000 18 13000 19 25000 20 14200 21 17000 22 1425 23 13800 24 11800 25 8700 # set a variable	feature[# for each for proportication geojson[turn geojson[turn geojson[con = df_to] fon = d	Ch column, get in properties: ure['properties: ure['properties: ure['properties: ure['properties: ure['seature (aka' features'].app on	the value a s'][prop] = a, converted bend(feature c', 'lon'] price, cols cn_geojson() price) ce, geojson= 'county", fe "mercator" ations", vis c', "t":0, "l": ations", vis c', "t":0, "l": ations', vis c', "t":0, "l": c', "systam c',	nd add it row[prop] datafram) geojson, atureidke ible=Fals 0,"b":0})	color="prices" color="prices"	ce", ies.county",	features i	nside our dict				250 200 150
15]: 16]:	Cols = geojsor #geojsor #geojsor fig = fi fig upor fig.upor fig.upor fig.sho 0 0 3083 1 13999 2 23348 3 16000 4 15000 5 14000 6 20000 7 14000 9 1042 10 12500 11 27500 12 8900 13 15000 14 11000 15 1017 16 8400 17 18000 18 13000 19 25000 20 14200 21 17000 22 1425 23 13800 24 11800 25 8700 # set a variable	# for each for proper feat # add the geojson [turn geojs for proper feat # add the geojson [turn geojs for proper feat # add the geojson [turn geojs for price for price feat for price	ch column, get in properties: ure ['properties' ure ['properties' is feature (aka 'features'].app on 'county', 'lat geojson(county_ y_price.electic_geojson(county_ eth(county_price_locations='projection=') fitbounds="locations='projection=') fitbounds="locations='locations='locations='projection=') fitbounds="locations='projection=') fitbounds="locations='locat	the value a is '][prop] = a, converted bend(feature c', 'lon'] price, cols c', geojson() price) ce, geojson() price) ce, geojson() ce, g	nd add it row[prop] datafram) geojson, atureidke ible=Fals 0,"b":0}) olumn we	color="pricy="propert:	the list of the li	he map	nside our dict				250 200 150
15]: 16]:	Cols = geojsor #geojsor #geojs	rice co ro.0 D rest. and the control con = count. con =	ch column, get in projectics: ure['properties' ure['properties' is feature (aka 'features'].app on 'county', 'lat geojson(county' yprice.electic_geojson(county' eth(county_price_locations='projection=') projection=') fitbounds="locatimargin={"r":6"	the value a is '][prop] = a, converted bend(feature a, 'lon'] price, cols con_geojson() price) ce, geojson= 'county", fe "mercator" ations", vis b, "t":0, "l": ations', vis congent of the second of the sec	nd add it row[prop] datafram) geojson, atureidke ible=Fals 0,"b":0}) linewidt 'fontsize 2019',xy	color="price" want to vis color="price" color="price" h=0.8, ax=6	ce", ies.county", fontweight'), xycoords:	he map r='0.8') : '3'}) ='figure f		ontalalignment='left'			250
15]: 16]:	#geojsor #geojsor #geojsor fig = fi fig.upo fig.upo fig.sho fig.sho fig = fi fig.upo fig.sho a soas 1 13999 2 23344 3 16000 4 15000 5 14000 6 20000 7 14000 8 11000 9 10422 10 12500 11 27500 12 8900 13 15000 14 11000 15 10174 16 8400 17 18000 18 13000 19 25000 20 14200 21 17000 22 14250 23 13800 24 11800 25 8700 # set a variable	rice co 70.0 D 80.0 Kilk 00.0 Ca 00.0	ch column, get in properties: ure['properties: ure['properties: is feature (aka' features']. app on 'county', 'lat geojson(county yerice.electic geojson(county eth(county_price locations="projections") fitbounds="locations="projections") fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" for search 52.651022 and search 52.651022 and search 52.857258 for search 53.731985 for search for	the value as a signification of the converted of the ature of the converted of the converte	nd add it row[prop] datafram) geojson, atureidke ible=Fals 0,"b":0}) linewidt 'fontsize 2019',xy	color="price" want to vis color="price" color="price" h=0.8, ax=6	ce", ies.county", fontweight'), xycoords:	he map r='0.8') : '3'}) ='figure f					250
15]: 16]:	Cols = geojsor #geojsor #geojs	rice co 70.0 D 80.0 Kilk 100.0 Ca 100.0 Ca	ch column, get in properties: ure['properties: ure['properties: is feature (aka' features']. app on 'county', 'lat geojson(county yerice.electic geojson(county eth(county_price locations="projections") fitbounds="locations="projections") fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" fitbounds="locations="projections" for search 52.651022 and search 52.651022 and search 52.857258 for search 53.731985 for search for	the value a s'][prop] = a, converted bend(feature a, 'lon'] price, cols c, geojson() price) ce, geojson= 'county", fe "mercator" ations", vis b, 't':0, 'l': ations', vis county', fe sesses and the sesses county', fe	olumn we olumn we olumn we ible=Fals 0,"b":0})	color="price" y="propert: h=0.8, ax=a h=0.8, ax=a h=0.1, .08 lormalize(vi	feature properties of the list	he map r='0.8') : '3'}) ='figure f ax=vmax))					250
16]: 16]:	Cols = geojsor #geojsor #geojs	feature[# for ear for prop feat # add th geojson[turn geojs ['price', n = df_to_ on = count, on = df_to_ on = d	ch column, get in properties ure ['properties ure ['projection ure	the value a s's's[prop] = a, converted dend(feature dend(feature c', 'lon's] price, cols ce, geojson= 'county", fe "mercator" ations", vis p, "t":0, "l": ations", vis p, "t":0, "l":1": atio	olumn we olumn we datafram datafram olinewidt 'fontsize 2019',xy orm=plt.N ating.py tting.py tting.py tting.py	color="print per row) to it the row) to it the row of	feature properties ax, edgecolor ax, edgecolor fontweight'), xycoords: min=vmin, vma nt call last .8, ax=ax, each	features i he map r='0.8') : '3'}) ='figure f ax=vmax)) dgecolor='	e.action', horize				250
15]: 16]:	Cols = geojsor #geojsor #geojsor fig = fi fig.upo fig.upo fig.sho fig.	feature[# for each for prop feat # add th geojson[turn geojs ['price', n = df_to_ ox.choroplicate_geos(date_geos(date_layou ox.choroplicate_geos(date_layou ox.choroplicate	ch column, get in properties: ure ['properties: ure ['properties: ure ['properties'] is feature (aka 'features'].app on 'county', 'lat geojson(county yerice.electic geojson(county projections) is fitbounds="locations="projections" projections of timargin={"r": 6	the value a s'][prop] = a, converted bend(feature a, conjourner a, conjourner c', 'lon'] price, cols con_geojson() /price) ce, geojson= 'county", fe ="mercator" ations", vis b, "t":0, "l": ations", vis c, "t":0, "l": ations", vis c, "t":0, "l": ations", vis con, "t":0, "l": ations", "	olumn we datafram datafram datafram linewidt fontsize 2019', xy orm=plt.N raceback 2906.py "ting.py ", "geo") me(data, _kinds: _ts	colors price price	feature property the list of it is a sualise on the sualise of the	features i he map r='0.8') : '3'}) ='figure f ax=vmax)) dgecolor=' s, **kwarg	0.8')	ontalalignment='left'	, vertic	alalignmen	250 200 100 t='top
15]: 16]:	## Create ## Cre	feature[# for ear for prop feat # add th geojson[turn geojs ['price', n' = df_to ox.choropl date_geos(date_layou ox.choropl don.o don.o	ch column, get in properties in properties in properties in feature (aka 'features'] appron 'county', 'lat geojson(county yurice.electic geojson(county yurice) geojson(county yurice	the value a s'][prop] = a, converted bend(feature a, conjourne) ce, geojson= 'county", fe "mercator" ations", vis b, "t":0, "l": ations", vis conjunt and a serial and	olumn we linewidt olumn we linewidt olumn we linewidt fontsize 2019', xy orm=plt.N ting.py """ """ """ """ """ """ """	want to vis the row) to it color="pricy" cy" fee) the o.8, ax=a cy" fee) formalize(vis finewidth=0 incall *args, **k in plot_da if ication_k cytonic incall incall *args, **k in plot_da incall cytonic i	feature properties of the list	features i he map r='0.8') : '3'}) ='figure f ax=vmax)) dgecolor=' s, **kwarg column, cm_kwds, asp	0.8')	cax, categorical, leg	, vertic	alalignmen	250 200 100
15]: 16]:	Cols = geojsor #geojsor #geojsor fig = fig. upor fig. upor fig. show fig. upor fig. show fig. upor fig. show fig. upor fig. show a show po 3083 1 13999 2 23344 3 16000 4 15000 7 14000 6 20000 7 14000 9 10422 10 12500 11 27500 12 8900 13 15000 14 11000 15 10177 16 8400 17 18000 20 14200 21 17000 22 14250 23 13800 24 11800 25 8700 # set a show # creat # wan # creat # add a ax.set_ # add a ax.set_ # add a ax.set_ # creat # add a ax.set_ # add a	feature[# for each for prop feat # add th geojson[turn geojs ['price', n = df_to_ ox.choropl. date_geos(date_layou ox.choropl. date_geos(date_geos(date_layou ox.choropl. d	ch column, get in properties in properties is reature (aka 'features'].app on 'county', 'lat geojson(county y_price.electic locations=' projection=') hinty lat cublin 53.349764 Laois 52.998458 Meath 53.649784 Laois 52.998458 Meath 53.649784 Laois 52.651022 Lerick 52.661252 Lerick 52.661252 Lerick 52.661252 Lerick 52.681252 Lerick 52.681252 Lerick 52.681253 L	the value a s'][prop] = a, converted bend(feature a', 'lon'] price, cols con_geojson() /price) ce, geojson= 'county", fe "mercator" ations", vis b, "t":0, "l": ations", vis b, "t":0, "l": ations", vis b, "t":0, "l": ations", vis con_geojson= con_geojson= con_geojson() /price) ce, geojson= ce, g	dadd it row[prop] datafram) geojson, atureidke ible=Fals 0,"b":0} raceback 2019',xy orm=plt.N raceback 2906.py in file geoms atureidke ible=rals 0,"b":0," raceback 2019',xy orm=plt.N	want to vis the row) to it color="prictive" propertion incall_ incall_ *args, **k in plot_da incall_ *args, **k in plot_da incall_ incal	sualise on the list of the lis	features i features i features i r='0.8') 'figure f ax=vmax)) dgecolor=' s, **kwarg column, cm kwds, asp	0.8') ap, color, ax,	cax, categorical, leg	, vertic	alalignmen	250 200 100
15]: 16]:	Cols = geojsor ## eojsor ## eojsor fig = pr fig.upo fig.upo fig.upo fig.sho ## cread merged 1 127500 12 8900 14 1000 15 14000 16 2000 17 14000 18 11000 19 1042: 10 12500 11 27500 12 8900 13 15000 14 11000 15 10174 16 8400 17 18000 18 13000 19 25000 20 14200 21 17000 22 14250 23 13800 24 11800 25 8700 ## cread fig, ax ## cread merged ## remov ax axis ## add ax set_ ## cread fig, ax ## cread fig,	feature[# for each for prop feat # add th geojson[turn geojs: ['price', n' = df_to_ ox.chorople date_geos(date_layou ox.chorople date_geos	ch column, get in properties ure ['properties ure ['properties ure ['properties is feature (aka 'features'].app on 'county', 'lat geojson(county_ geojson(c	the value a s'][[prop] = a, converted bend(feature c', 'lon'] price, cols con_geojson() //price) ce, geojson= county', fe "mercator" ations", vis con, "t"', "l": ations", "t" con, "t" c	linewidt datafram datafram linewidt datafram linewidt datafram linewidt datafram linewidt datafram column we linewidt datafram geojson, atureidke ible=Fals linewidt datafram geojson, atureidke ible=Fals linewidt datafram geojson, atureidke ible=Fals linewidt datafram geojson, atureidke column we linewidt datafram geojson, atureidke linewidt datafram geojson, atureidke linewidt linew	want to vis the row) to a color="price" color="p	feature proportions at the list of a second of a secon	features i he map r='0.8') : '3'}) ='figure f ax=vmax)) dgecolor=' s, **kwarg column, cm _kwds, asp Geom") name)	0.8') ap, color, ax,	cax, categorical, leg	, vertic	alalignmen	250 200 100
15]: 16]:	Cols = geojsor #geojsor #geojsor fig = fi fig.upo fig.upo fig.upo fig.sho a 1399 2 2334 3 16000 4 15000 5 14000 6 2000 7 14000 8 11000 9 1042 10 12500 11 27500 12 8900 14 11000 15 1017 16 8400 17 18000 19 1042 21 17000 22 1425 23 13800 24 11800 25 8700 # creat fig, as # c	feature[# for each for prop feat # add th geojson[turn geojs: ['price', n' = df_to_ ox.chorople date_geos(date_layou ow() do.0	ch column, get in properties ure ['properties ure ['properties ure ['properties is feature (ake 'features'].app on	the value as is [prop] = a, converted dend(feature dend(feature dend(feature dend(feature)) [price, cols dend(feature)]	row[prop] datafram) geojson, atureidke ible=Fals 0,"b"',xy orm=plt.N tting.py ", "geo", me(data, kinds: ts tting.py ", "geo") me(dataframe index, kinds: ts tting.py solution dataframe met yet (eselfgeo solution)	want to vis color="price" color="price" cy="properts incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes *args, **k in plot_da ifica	feature proportions at the list of state of stat	features i he map r='0.8') : '3'}) ='figure f ax=vmax)) dgecolor=' s, **kwarg column, cm _kwds, asp Geom") name)	0.8') ap, color, ax,	cax, categorical, leg	, vertic	alalignmen	250 200 100
15]: 16]:	Cols = geojsor #geojsor #geojsor fig = fi fig.upo fig.upo fig.upo fig.sho as a second po 0 3083 1 13999 2 23344 3 16000 4 15000 5 14000 6 2000 7 14000 1 12500 11 27500 12 8900 14 11000 15 1017 16 8400 17 18000 18 13000 19 25000 14 11000 15 1017 16 8400 17 18000 18 13000 19 25000 14 11000 21 17000 22 14250 23 13800 24 11800 25 8700 # set a second # creat fig, as # creat	feature[# for each for prop feat # add th geojson[turn geojs: ['price', n' = df_to_ ox.chorople date_geos(date_layou ow() do.0	ch column, get in properties: ure['properties: ure['properties: ure['properties: ure['properties: ure['properties: ure]' features'] apponent on the county of the county o	the value as is [prop] = a, converted dend(feature dend(feature dend(feature dend(feature)) [price, cols dend(feature)]	row[prop] datafram) geojson, atureidke ible=Fals 0,"b"',xy orm=plt.N tting.py ", "geo", me(data, kinds: ts tting.py ", "geo") me(dataframe index, kinds: ts tting.py solution dataframe met yet (eselfgeo solution)	want to vis color="price" color="price" cy="properts incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes *args, **k in plot_da ifica	feature proportions at the list of state of stat	features i he map r='0.8') : '3'}) ='figure f ax=vmax)) dgecolor=' s, **kwarg column, cm _kwds, asp Geom") name)	0.8') ap, color, ax,	cax, categorical, leg	, vertic	alalignmen	250 200 100
[15]:	Cols = geojsor #geojsor #geojsor #geojsor fig = propersor fig = propersor fig.upor fig.upor fig.upor fig.shood A	feature[# for each for prop feat # add th geojson[turn geojs: ['price', n' = df_to_ ox.chorople date_geos(date_layou ow() do.0	ch column, get in properties: ure['properties: ure['properties: ure['properties: ure['properties: ure['properties: ure]' features'] apponent on the county of the county o	the value as is [prop] = a, converted dend(feature dend(feature dend(feature dend(feature)) [price, cols dend(feature)]	row[prop] datafram) geojson, atureidke ible=Fals 0,"b"',xy orm=plt.N tting.py ", "geo", me(data, kinds: ts tting.py ", "geo") me(dataframe index, kinds: ts tting.py solution dataframe met yet (eselfgeo solution)	want to vis color="price" color="price" cy="properts incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes *args, **k in plot_da ifica	feature proportions at the list of state of stat	features i he map r='0.8') : '3'}) ='figure f ax=vmax)) dgecolor=' s, **kwarg column, cm _kwds, asp Geom") name)	0.8') ap, color, ax,	cax, categorical, leg	, vertic	alalignmen	250 200 100
15]: 17]:	Cols = geojsor #geojsor fig = fi fig.upo fig.upo fig.sho fig.sho	feature[#for each for proper feat # # for proper feat # # add th geojson[turn geojs: c'price', n = df_to_ concept = df_to_	ch column, get in properties in properties is feature (ake 'features']. appron county', 'lat geojson(county county', 'lat geojson(county county_price.electic geojson(county county_price.electic locations='projections projections projections if thounds="locations" fit bounds="locations" county_price locations projections if the county_price locations locati	the value a is [prop] = a, converted bend(feature a, converted bend(feature c, 'lon'] price, cols congeojson() //price) ce, geojson= 'county', fe "mercator" ations", vis b, "t":0, "l": ations", vis b, "t":0, "l":1 ations", vis b, "tolar the ations", vis b, "tolar the ations", vis b, "tolar the	linewidt 'fontsize atafram datafram linewidt 'fontsize 2019', xy orm=plt.N caceback 2906.py "geo") me(data, ts tting.py ", "geo")	want to vis color="price" color="price" cy="properts incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes *args, **k in plot_da ifica	feature proportions at the list of state of stat	features i he map r='0.8') : '3'}) ='figure f ax=vmax)) dgecolor=' s, **kwarg column, cm _kwds, asp Geom") name)	0.8') ap, color, ax,	cax, categorical, leg	, vertic	alalignmen	250 200 100
15]: 17]:	Cols = geojsor #geojsor #geojsor fig = fig fig .upo fig	feature[# for each for prop feat # add th geojson[turn geojs: ['price', n' = df_to_ ox.chorople date_geos(date_layou ow() do.0	ch column, get in properties: ure['properties: ure['properties: ure['properties: ure['properties: ure['properties: ure]' features'] apponent on the county of the county o	the value as is [prop] = a, converted dend(feature dend(feature dend(feature dend(feature)) [price, cols dend(feature)]	row[prop] datafram) geojson, atureidke ible=Fals 0,"b"',xy orm=plt.N tting.py ", "geo", me(data, kinds: ts tting.py ", "geo") me(dataframe index, kinds: ts tting.py solution dataframe met yet (eselfgeo solution)	want to vis color="price" color="price" cy="properts incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes incall *args, **k in plot_da ification_k classes *args, **k in plot_da ifica	feature proportions at the list of state of stat	features i he map r='0.8') : '3'}) ='figure f ax=vmax)) dgecolor=' s, **kwarg column, cm _kwds, asp Geom") name)	0.8') ap, color, ax,	cax, categorical, leg	, vertic	alalignmen	250 200 100

In [1]:

import pandas as pd