Brexit voting in “left-behind” communities: Was NHS spending a deciding factor?

Emmanouil Xagoraris, 2290710

University of Warwick, Centre for the Interdisciplinary Methodologies

Table of Contents

[Abstract 2](#_Toc123564495)

[Introduction and Background 3](#_Toc123564496)

[Data and Methodologies 4](#_Toc123564497)

[Local analysis 4](#_Toc123564498)

[Nationwide analysis 8](#_Toc123564499)

[Conclusion 10](#_Toc123564500)

[Limitations and Reflections 11](#_Toc123564501)

[Appendix A 12](#_Toc123564502)

[Appendix B 13](#_Toc123564503)

[Appendix C 14](#_Toc123564504)

[Appendix D: Code 15](#_Toc123564505)

[Works Cited 34](#_Toc123564506)

Abstract

NHS is the backbone of English society as it is the only public healthcare provider in the country. That is the reason why social inequalities are reflected in it. On the other hand, Brexit voting is considered an echo of social disparities. Thus, this paper tries to identify the causal relationship between NHS spending at a GP level and the EU Referendum voting propensity. The results were surprising as there was an association between the two factors that could be used by the public and health policymakers.

Keywords: NHS, North-South Divide, London, Brexit Voting, Regression, Classification, Clustering, Visualization

# Introduction and Background

Almost 200 years have passed since the Unification of England, Scotland, and Northern Ireland and the creation of the United Kingdom but people feel more divided than ever. This division is attributed to socio-economic factors. More specifically, the conservative think tank “Policy Exchange” published a report in 2008 claiming that North England is economically failing while the southern part of the country is flourishing (Spiers, 2018, p. 59).

Kontopantelis et al. (2018) claimed that societal disparities are mirrored within the National Health System (NHS) as it is the only healthcare provider in the county. In their research, they found out that regions that are poorer like the North tend to have underfunded General Practices (GPs) which leads ultimately to a sicker population.

Those disparities started to sharpen, as a referendum related to UK’s relationship with the European Union was announced, splitting the population into “Remainers” and “Leavers”. Various researchers tried to find the factors that affected people’s voting decisions. Beecham et al. (2018) found that Brexit voting was affected by local variables such as income, level of education, and political orientation. The considered “left-behind” communities thought that Brexit was the way to boost their local economies, create meaningful relationships with the other communities within the United Kingdom, and raise their opposition to the government and the EU that eventually “forgot” them. While in the flourishing part of the country where commerce and banking were conducted “Bremain” was the only reasonable option. That is the reason why only London and Scotland voted in favor of Remain, while the rest of the counties voted for Leave.

Therefore, this data-informed essay will take the studies mentioned above as springboard to find a new association between two variables which is NHS spending and Brexit voting propensity in different geographical regions of England. The results of this study may be able to answer questions related to inequalities, health policy, public policy, and social norms.

# Data and Methodologies

The NHS dataset is derived from the National Health System Database, and it contains the spending per General Practice, the overall experience of GP surgery, and the overall satisfaction of people with long-term illnesses for the year 2015-2016. This year was chosen deliberately as people tend to be “victims of availability bias” meaning that their most recent memory is indicative of their whole experience (Park et al., 2018).

Moreover, referendum data from five out of six Metropolitan Counties will be used (Greater Manchester, Merseyside, Tyne and Wear, South Yorkshire, West Yorkshire) as they are in the Northern part of the country, while Greater London that will be studied, as well, is in the South; to test the existence of North-South divide. West Midlands are excluded, as some researchers were qualifying them as North and some others as South (Smith, 2018).

A preliminary analysis, using data visualization, will be conducted in the area of London to see if there is variability within the city as Kochan (2011) claimed that London is the most unequal city in the UK. Moreover, a nationwide analysis will be conducted to confirm the hypothesis that there is a statistically significant between different geographies in NHS funding, and Brexit voting using ANOVA tests. Finally, inferential statistics and more specifically regression, will be used to identify the exact effect of NHS spending on Brexit voting. Data processing, cleaning, aggregation, analysis, and visualization were conducted within the Python programming language. Full data analysis scripts and explanations can be accessed at this GitHub™ repository: <https://github.com/mxagoraris/Data_Science_Project_Brexit_and_NHS>

# Local analysis

The region of Greater London is full of dichotomies as some of its boroughs are considered the most prosperous and at the same time some others the poorest ones in the nation (Kochan, 2011). That is the reason why it is ideal for a preliminary analysis, as if the hypothesis is confirmed for this polarizing region, it could be confirmed also for the whole country as it represents all social classes. To conduct this analysis data cleaning, and merging was required as the fields between the two databases were not compatible. The final form of the database can be found in [Appendix A](#_Appendix_A).

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Figure 1: Long term illness satisfaction Scatterplot

Chart, scatter chart

Description automatically generated

Figure 2: Surgery satisfaction Scatterplot

Figure 3: Spending per patient Scatterplot

The scatterplots above are visualized using Python’s data visualization library called *Seaborn*. Just from visualizing our data, it can be understood that there is no relationship between long-term ill-patient satisfaction, Surgery Satisfaction, and the people’s tendency to vote in favour of remaining, therefore, those variables will be dropped for the rest of the study. However, there seems to be a strong linear relationship between spending per patient and vote propensity if the outlier is removed.

Chart, scatter chart

Description automatically generatedKochan (2011) claimed that there are disparities between the inner and outer London boroughs, therefore, it will be interesting to cluster the data of Figure 3 to see if that hypothesis is confirmed. There are two clustering methods that can be used; the first one is K-means and the second one is Hierarchical Clustering. In this case, K-means seems to be a better choice as the number of centroids is known, and it is either 2 if the outlier is removed or 3 if it is not. Apart from that K-means method is faster and less costly in terms of memory usage (Xu and Wunsch, 2009).

Figure 4: London spending per patient clusters

Map

Description automatically generated

Figure 5: London map clustering

The maps of Figure 5,6,7 were created using the *Geospatial Data Abstraction Library* (GDAL), *GeoPandas[[1]](#endnote-2)*, geospatial data[[2]](#endnote-3) downloaded by data.gov.uk and Python’s *matplotlib library* (see [Appendix D](#_Appendix_D:_Code)). Indeed, clustering confirms Kochan’s hypothesis (2011) that London is a very diverse city with a lot of inequalities. In cluster two which is the outer ring of London people have low Spending per Patient and low Remain-voting, as in cluster one Spending per Patient is increased while Remain-voting is increased. Finally, the borough of Kensington is clustered alone as it is the most affluent and the remain percentage was high (Spending per Patient: 381.16, Remain Percentage: 68.89). This behaviour can be also identified by the following colourmaps and table:

A picture containing chart

Description automatically generatedA picture containing map

Description automatically generated

Figure 6: Spending per patient colourmap

Figure 7: Remain Percentage colourmap

Table 4: Max and Min remain percentages

|  |  |  |
| --- | --- | --- |
| Borough | Pct  Remain | Spending per Patient |
| Lambeth | 78.62 | £219.16 |
| Hackney | 78.36 | £284.49 |
| Haringey | 75.57 | £218.99 |
| Sutton | 46.28 | £166.93 |
| Bexley | 37.05 | £151.44 |
| Havering | 30.34 | £157.21 |

The closer someone gets to the center, the higher the probability to find a “Remainer” voter who has better healthcare than his/her counterpart “Remainer” that lives in the outer ring of London. As this preliminary analysis shows there is a relationship between Brexit voting and Spending per Patient in the city of London, however, this hypothesis should be substantiated and quantified for the whole Nation.

# Nationwide analysis

In this part of the analysis, counties will be classified as North and South to test if there is a significant difference in Brexit voting and NHS spending due to geographical location. This step is necessary as if, indeed, NHS spending is significantly lower in the North where people were against remaining; then the initial hypothesis of “left-behind” communities as “leavers” is confirmed.

This test will be conducted using ANOVA and follow-up tests[[3]](#endnote-4). The significance level for this test will be α = 5% which is used in all social sciences. If the p-value of the test is greater than 5% then the null hypothesis cannot be rejected.

In the first test there seems to be a statistically significant difference between North and South in Brexit voting, as the null hypothesis is rejected, F(1, 61) = 37.6115, p < .001. Accordingly, the null hypothesis is rejected for the second test, F(1, 61) = 12.7411, p < .001, therefore, there is a statistically significant difference in NHS spending between North and South. Details can be found in [Appendix B](#_Appendix_B).

One-way ANOVA does not offer the full information needed to confirm the initial hypothesis as the direction of the difference is needed. Consequently, a follow-up test will be conducted. The p-values of those tests have been adjusted using Bonferroni-Holm correction to control multiple testing (Lesack and Naugler, 2011). As two active treatment population means will be compared, an analytic comparison or contrast hypothesis test should be used (Seltman, 2013). The null hypothesis of the test is the following:

Both tests are statistically significant as expected, as the ANOVA test was significant. The first test confirms that the average Remain percentage in North was 6.2719% less than in the South. The second test comes to confirm that NHS spending in the North is significantly reduced than in the South and the average difference is Δ = £3.654. Details can be found in [Appendix B](#_Appendix_B).

The combination of those two tests come to confirm the initial hypothesis that “left-behind” communities voted in favour of leaving the European Union on average, while NHS spending per Patient affected their voting decision. Therefore, NHS spending per Patient is an indicator of inequalities within the health system.

Finally, the following linear model is estimated to predict the percentage of Remain voting as a function of Spending per Patient. Initially OLS was used, however GLS with AR(1) was deemed as a more appropriate method, due to the fact that the serial correlation hypothesis of OLS was violated[[4]](#endnote-5).

Both are statistically significant as for t(61) = 3.699, p < .001 and for t(61) = 4.1115, p < .001 . Details can be found in [Appendix C](#_Appendix_C).

Therefore, the final form of the model is the following:

Hence, for every increase/decrease of £1 on the NHS Spending the result in favour of remain increases/decreases by 0.1752% on average. This indicator is quite important as it can be used in public policy from NHS and government officials. Finally, 27.9% of the variability of the Remain Percentage is explained by the fluctuation of Spending per Patient. This is quite reasonable as Beecham et al. (2018) indicated that voting decision-making is quite a complex procedure that cannot be explained by a handful of variables.

# Conclusion

Consistent with the existing bibliography, it is found that NHS spending per patient at a GP level can causally explain Brexit voting results, as NHS mirrors all forms of inequality within the English society. State healthcare should be equal to all as if disparities exist, they could be institutionalized and amplified.

Thus, a change in NHS funding procedure is essential; this topic offers a great opportunity for future research endeavors as researchers could investigate how AI and Machine Learning algorithms could be used to allocate funding efficiently based on real-life needs and why the obsolete Carr-Hill formula[[5]](#endnote-6) (Kontopantelis et al., 2018) should be quitted in favour of such technologies.

Finally, the state of the National Health System seems not to affect only a community’s well-being but politics as well. Hence, in the era of populism, NHS could be used to serve certain political agendas. That is the reason why public health policy should be above politics as it affects individuals’ quality of life and societal prosperity.

# Limitations and Reflections

One of the limitations identified while conducting the study was the incompatibility of NHS and Referendum databases as there is no General Practice in every county. That is the reason why Metropolitan Counties were chosen instead as few manipulations were needed. Therefore, someone could claim that the scope of the study is limited, however, that claim would be unsubstantiated as the regions chosen account for more than 45% of the English population. Furthermore, the linear model estimated might be deceiving; as there were outliers in the data that inflated or deflated the constant, but their removal was rejected as they represent the inequalities within the society.

If this essay was conducted under different circumstances, we could try to either find compatible databases or an automated way, using python to match the two databases[[6]](#endnote-7). In that case, a broader study could be conducted using data from the whole of England. As for the linear model, instead of using Spending per Patient variable, it could be interesting to classify the data using K-Nearest Neighbour Algorithm[[7]](#endnote-8) and then estimate a dummy logistic regression model. In that way outliers could be classified together and offer complementary information on their weight to the model.

# Appendix A

Table 1: NHS Data 2015-2016

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| General Practice | Metropolis | NHS funding | Surgery Satisfaction % | Long-term ill Satisfaction % |
| Bury | Greater Manchester | £34,829,332 | 86.49 | 66.69 |
| Bolton | Greater Manchester | £21,251,084 | 86.43 | 65.59 |
| … | … | … | … | … |
| Kirkless | West Yorkshire | £52,891,084 | 83.02 | 66.72 |
| Wakefield | West Yorkshire | £53,436,477 | 86.09 | 64.27 |

Table 2: Referendum Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| County | Metropolis | Pct\_Remain | Pct\_Leave | Electorate |
| Bury | Greater Manchester | 41.71 | 58.29 | 197,109 |
| Bolton | Greater Manchester | 45.88 | 54.12 | 141,600 |
| … | … | … | … | … |
| Kirkless | West Yorkshire | 45.33 | 54.67 | 307,081 |
| Wakefield | West Yorkshire | 33.64 | 66.36 | 246,096 |

Table 3: London Analysis Aggregated Data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Borough | Electorate | Pct  Remain | Pct  Leave | NHS Payment | Spending/ Patient | Surgery Satisf. | Long term ill |
| Barnet | 223,467 | 62.23 | 37.77 | £43,137,567 | £193.03 | 80.82 | 55.72 |
| Bexley | 170,779 | 37.05 | 54.12 | £25,864,082 | £151.44 | 78.10 | 60.51 |
| … | … | … | … | … | … | … | … |
| Wandsworth | 219,521 | 75.03 | 24.97 | £45,888,817 | £209.04 | 86.41 | 60.86 |
| Westminster | 120,524 | 68.97 | 31.03 | £24,905,012 | £206.64 | 78.46 | 61.66 |

# Appendix B

Table 5: ANOVA Pct\_Remain Geography Results

|  |  |  |  |
| --- | --- | --- | --- |
| Class. | Degrees of Freedom | F-Test | Prob. |
| Geography | 1, 61 | 37.6115 | <.01 |

Table 6: ANOVA Spending per Patient and Geography Results

|  |  |  |  |
| --- | --- | --- | --- |
| Class. | Degrees of Freedom | F-Test | Prob. |
| Geography | 1, 61 | 12.7411 | .0007 |

Table 7: North-South Contrast on Remain Percentage

|  |  |  |  |
| --- | --- | --- | --- |
| Contrast | Difference | Degrees of Freedom | Prob. |
| North-South | -6.2719 | 53.4248 | <.001 |

Table 8: North-South Contrast on NHS spending per Patient

|  |  |  |  |
| --- | --- | --- | --- |
| Contrast | Difference | Degrees of Freedom | Prob. |
| North-South | -3.6540 | 52.6538 | <.001 |

# Appendix C

Table 9: GLS output coefficients

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Regressors | Coeff. | Std err | t-value | Prob. |
| Const. | 25.5509 | 6.907 | 3.699 | .000 |
| Spending | .142 | .034 | 4.115 | .000 |

Table 10: GLS output metrics

|  |  |
| --- | --- |
|  | Results |
| Df Residuals | 61 |
| Df Model | 1 |
| R-squared | .217 |
| Adj. R-squared | .204 |
| F-Statistic | 16.94 |
| AIC | 484.1 |
| Durbin-Watson | 1.871 |
| Jarque-Bera | 2.460 |

# Appendix D: Code

Importing some of the libraries

import pandas as pd  
import seaborn as sns  
import numpy as np  
import matplotlib.pyplot as plt  
from sklearn.linear\_model import LinearRegression  
from sklearn import metrics  
from sklearn.datasets import load\_iris  
from sklearn.preprocessing import MinMaxScaler  
from sklearn.cluster import KMeans  
from sklearn.cluster import AgglomerativeClustering

Doing some data cleaning and manipulation

brexit\_results = pd.read\_csv("EU-referendum-result-data.csv")  
brexit\_results=brexit\_results.filter(["Region\_Code","Region","Area","Electorate","Pct\_Remain","Pct\_Leave"])  
london=brexit\_results[brexit\_results["Region"]=="London"] #Choosing london for the preliminary research  
london.index = london["Area"]  
london=london.drop(["Region\_Code","Region","Area"],axis=1) #Droping the not needed areas  
#combining thr results of Hackney and City of London to be consistent with nhs database  
london.loc["Hackney","Pct\_Remain"]=london.loc["City of London","Pct\_Remain"]\*(london.loc["City of London","Electorate"]/(london.loc["Hackney","Electorate"]+london.loc["City of London","Electorate"]))+london.loc["Hackney","Pct\_Remain"]\*(london.loc["Hackney","Electorate"]/(london.loc["Hackney","Electorate"]+london.loc["City of London","Electorate"]))  
london.loc["Hackney","Pct\_Leave"]=london.loc["City of London","Pct\_Leave"]\*(london.loc["City of London","Electorate"]/(london.loc["Hackney","Electorate"]+london.loc["City of London","Electorate"]))+london.loc["Hackney","Pct\_Leave"]\*(london.loc["Hackney","Electorate"]/(london.loc["Hackney","Electorate"]+london.loc["City of London","Electorate"]))  
london.loc["Hackney","Electorate"]=london.loc["Westminster","Electorate"]+london.loc["City of London","Electorate"]  
london.drop("City of London",inplace=True)

nhs\_london\_payments = pd.read\_excel("nhspaymentsgp-14-15-ann1.xlsx")

nhs\_london\_payments.index = nhs\_london\_payments["CCG Description"]

#changing names to make them consistent  
london.rename(index={"Hackney":"City and Hackney","Kensington and Chelsea":"West London","Kingston upon Thames":"Kingston","Richmond upon Thames":"Richmond"},inplace=True)  
nhs\_london\_payments.rename(index={nhs\_london\_payments.index[-1]:"Westminster"},inplace=True)  
  
  
#merge data  
df\_merged = pd.merge(london,nhs\_london\_payments.iloc[:,2],left\_index=True, right\_index=True)  
df\_merged  
df\_merged =pd.merge(df\_merged,nhs\_london\_payments.iloc[:,-3],left\_index=True, right\_index=True)  
df\_merged  
df\_merged.rename(columns={df\_merged.columns[-1]:"NHS Payments",df\_merged.columns[-2]:"Registered Patients"},inplace=True)

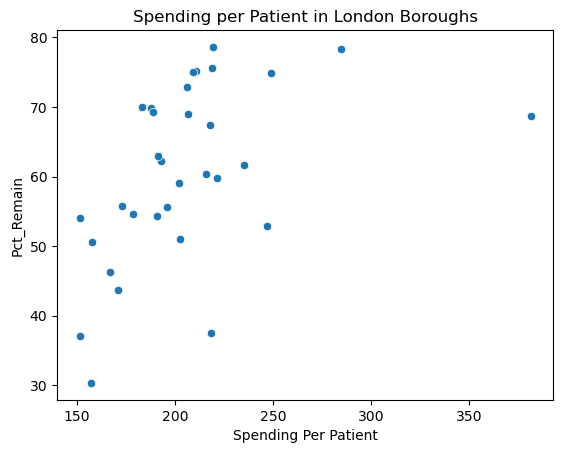
df\_merged = pd.merge(df\_merged,nhs\_london\_payments.iloc[:,-1],left\_index=True, right\_index=True)  
df\_merged = pd.merge(df\_merged,nhs\_london\_payments.iloc[:,-2],left\_index=True, right\_index=True)

df\_merged  
df\_merged["Spending Per Patient"]=df\_merged["NHS Payments"]/df\_merged["Electorate"]

Scatterplots creation using seaborn

sns.scatterplot(data=df\_merged,x="Spending Per Patient",y="Pct\_Remain").set(title="Spending per Patient in London Boroughs")  
#plt.savefig('Spending per Patient.png')

[Text(0.5, 1.0, 'Spending per Patient in London Boroughs')]



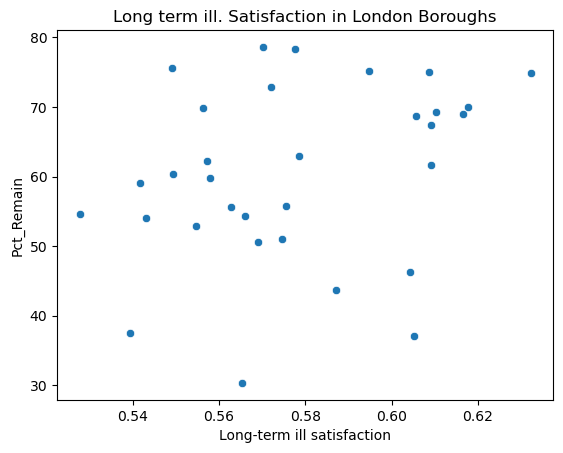
sns.scatterplot(data=df\_merged,x="Surgery Satisfaction",y="Pct\_Remain").set(title='Surgery Satisfaction in London Boroughs')  
#plt.savefig('Surgery Satisfaction.png')

[Text(0.5, 1.0, 'Surgery Satisfaction in London Boroughs')]



sns.scatterplot(data=df\_merged,x="Long-term ill satisfaction",y="Pct\_Remain").set(title="Long term ill. Satisfaction in London Boroughs")  
#plt.savefig('Long term Ill Satisfaction.png')

[Text(0.5, 1.0, 'Long term ill. Satisfaction in London Boroughs')]

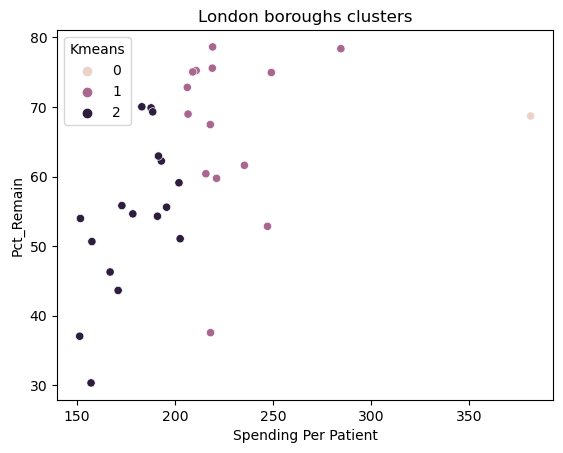


df\_merged\_1=df\_merged

**Clustering with Kmeans**

X = df\_merged\_1.loc[:,["Pct\_Remain","Spending Per Patient"]]  
k\_means=KMeans(n\_clusters = 3, init = 'random')   
  
k\_means.fit(X)   
labels = k\_means.labels\_  
  
df\_merged\_1["Kmeans"]=labels

sns.scatterplot(data = df\_merged\_1, x="Spending Per Patient",y="Pct\_Remain", hue = "Kmeans").set(title='London boroughs clusters')  
  
plt.savefig('London Boroughs Clusters.png')



*Map Reading*

import geopandas as gpd   
  
from osgeo import gdal  
gdal.SetConfigOption('SHAPE\_RESTORE\_SHX','YES')  
fp = "statistical-gis-boundaries-london/ESRI/London\_Borough\_Excluding\_MHW.shp"  
map\_df=gpd.read\_file(fp)

Data Cleaning

list(set(df\_merged.index)-set(map\_df["NAME"])) #checking if there is a difference between the names

['Richmond', 'Kingston', 'City and Hackney', 'West London']

list(set(map\_df["NAME"])-set(df\_merged.index)) #checking if there is a difference between the names

['Richmond upon Thames',  
 'Kingston upon Thames',  
 'Kensington and Chelsea',  
 'City of London',  
 'Hackney']

df\_merged.rename(index={"Richmond":"Richmond upon Thames","Kingston":"Kingston upon Thames","West London":"Kensington and Chelsea"},inplace=True)  
#renaming to have compatible indices

list(set(map\_df["NAME"])-set(df\_merged.index))

['City of London', 'Hackney']

df\_merged.loc["City of London",:]=df\_merged.loc["City and Hackney",:]  
df\_merged.rename(index={"City and Hackney":"Hackney"},inplace=True)

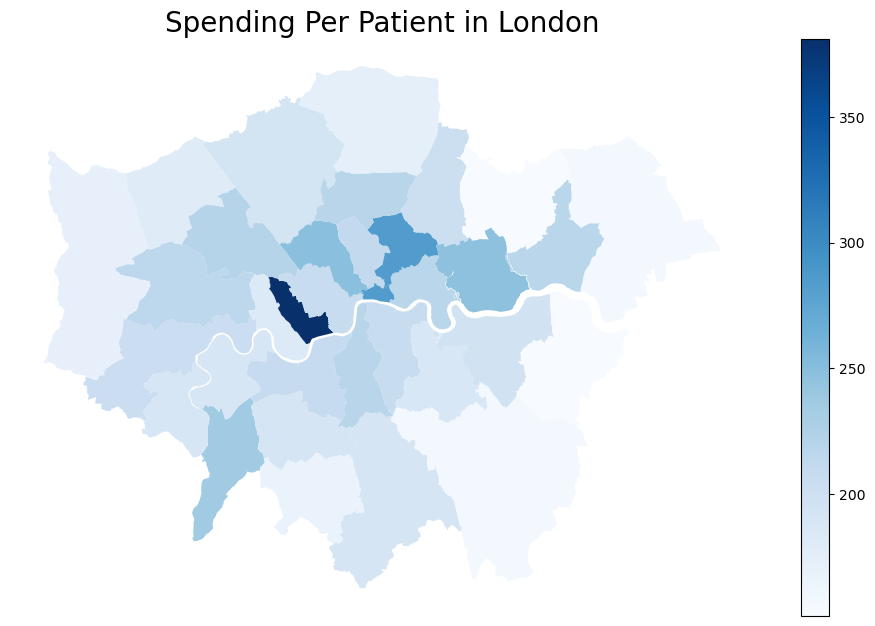
df\_merged.sort\_index() #sorting based on the index  
df\_merged["Kmeans"]=df\_merged["Kmeans"].apply(int) #making sure that clusters are integers

***Merging gdp with pd dataframe***

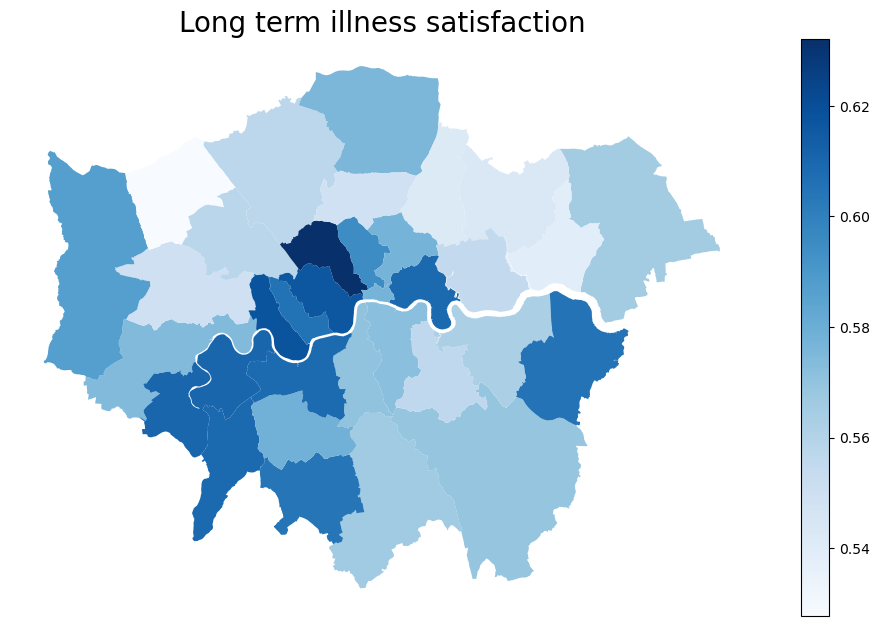
merged=map\_df.set\_index('NAME').join(df\_merged.set\_index(df\_merged.index)) #creating a new gdp dataframe

**Map Creation**

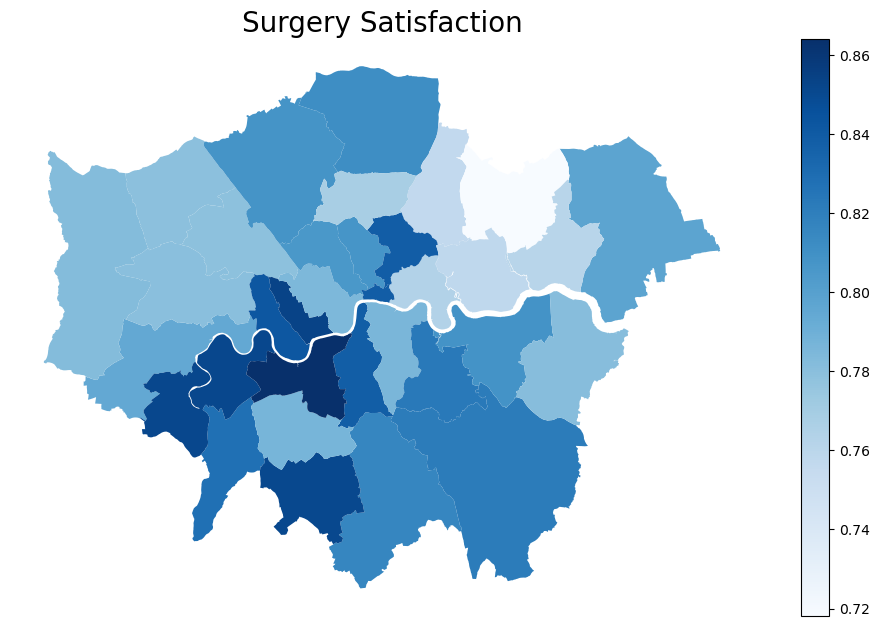
hap='Spending Per Patient'  
vmin,vmax=merged[hap].min(),merged[hap].max()  
fig,ax=plt.subplots(1,figsize=(12,7.5))  
merged.plot(column=hap,cmap='Blues',ax=ax)  
ax.axis('off')  
plt.title('Spending Per Patient in London',{'fontsize': '20',  
 'fontweight' :'100'})  
sm = plt.cm.ScalarMappable(cmap='Blues', norm=plt.Normalize(vmin=vmin, vmax=vmax))  
sm.\_A = []  
cbar = fig.colorbar(sm)  
#plt.savefig('Spending map London.png')  
  
#map creation



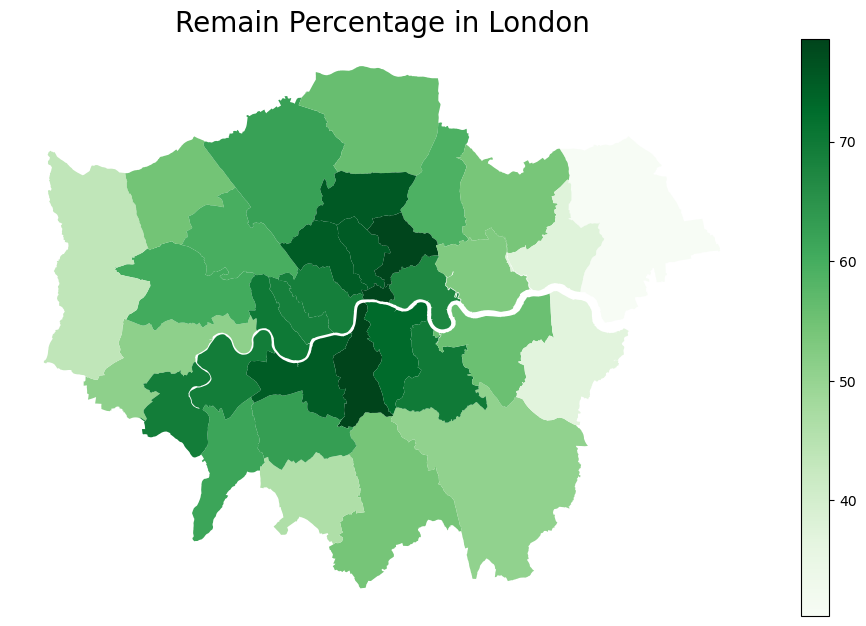
hap='Long-term ill satisfaction'  
vmin,vmax=merged[hap].min(),merged[hap].max()  
fig,ax=plt.subplots(1,figsize=(12,7.5))  
merged.plot(column=hap,cmap='Blues',ax=ax)  
ax.axis('off')  
plt.title('Long term illness satisfaction',{'fontsize': '20',  
 'fontweight' :'100'})  
sm = plt.cm.ScalarMappable(cmap='Blues', norm=plt.Normalize(vmin=vmin, vmax=vmax))  
sm.\_A = []  
cbar = fig.colorbar(sm)



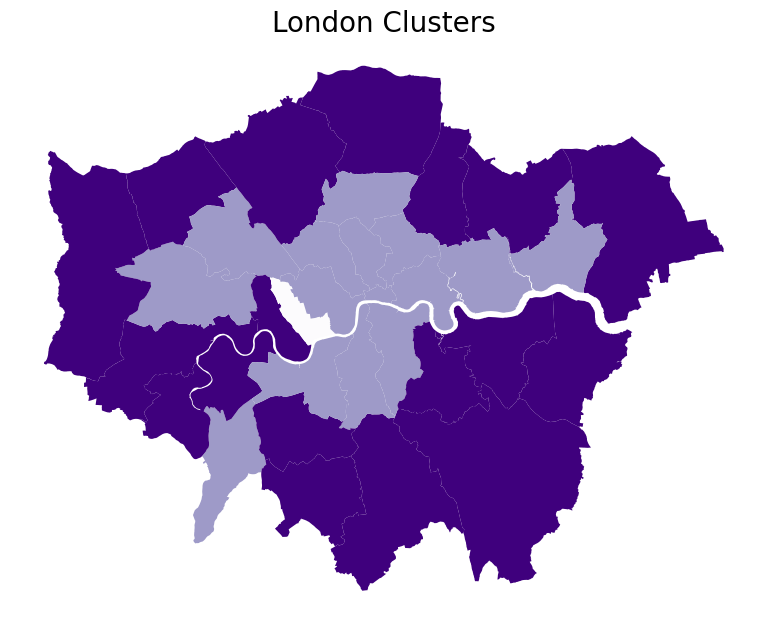
hap='Surgery Satisfaction'  
vmin,vmax=merged[hap].min(),merged[hap].max()  
fig,ax=plt.subplots(1,figsize=(12,7.5))  
merged.plot(column=hap,cmap='Blues',ax=ax)  
ax.axis('off')  
plt.title('Surgery Satisfaction',{'fontsize': '20',  
 'fontweight' :'100'})  
sm = plt.cm.ScalarMappable(cmap='Blues', norm=plt.Normalize(vmin=vmin, vmax=vmax))  
sm.\_A = []  
cbar = fig.colorbar(sm)



hap='Pct\_Remain'  
vmin,vmax=merged[hap].min(),merged[hap].max()  
fig,ax=plt.subplots(1,figsize=(12,7.5))  
merged.plot(column=hap,cmap='Greens',ax=ax)  
ax.axis('off')  
plt.title('Remain Percentage in London',{'fontsize': '20',  
 'fontweight' :'100'})  
sm = plt.cm.ScalarMappable(cmap='Greens', norm=plt.Normalize(vmin=vmin, vmax=vmax))  
sm.\_A = []  
cbar = fig.colorbar(sm)  
plt.savefig('Remain map London.png')



hap='Kmeans'  
fig,ax=plt.subplots(1,figsize=(12,7.5))  
merged.plot(column=hap,cmap='Purples',ax=ax)  
ax.axis('off')  
plt.title('London Clusters',{'fontsize': '20',  
 'fontweight' :'100'})  
sm.\_A = []  
#plt.savefig('Clusters map.png')



Reading new data

nhs\_spending\_metro = pd.read\_excel("nhspaymentsgp-14-15-ann1.xlsx",sheet\_name=1,index\_col=0) #reading the new excel file  
all\_uk = brexit\_results[brexit\_results.Region!="London"] #exluding London  
all\_uk = all\_uk.loc[:,["Area","Electorate","Pct\_Leave","Pct\_Remain"]]  
all\_uk

Area Electorate Pct\_Leave Pct\_Remain  
0 Peterborough 120892 60.89 39.11  
1 Luton 127612 56.55 43.45  
2 Southend-on-Sea 128856 58.08 41.92  
3 Thurrock 109897 72.28 27.72  
4 Bedford 119530 51.78 48.22  
.. ... ... ... ...  
377 Bradford 342817 54.23 45.77  
378 Calderdale 149195 55.68 44.32  
379 Kirklees 307081 54.67 45.33  
380 Leeds 543033 49.69 50.31  
381 Wakefield 246096 66.36 33.64  
  
[349 rows x 4 columns]

nhs\_spending\_metro

Metropolitan County Total NHS Payments \  
Area   
Bolton Greater Manchester 3.482933e+07   
Bury Greater Manchester 2.125108e+07   
Manchester Greater Manchester 6.316014e+07   
Oldham Greater Manchester 2.806523e+07   
Rochdale Greater Manchester 2.467366e+07   
Salford Greater Manchester 2.880450e+07   
Stockport Greater Manchester 3.059116e+07   
Tameside Greater Manchester 2.752122e+07   
Trafford Greater Manchester 2.704179e+07   
Wigan Greater Manchester 3.790654e+07   
Halton Merseyside 1.507064e+07   
Knowsley Merseyside 2.285274e+07   
Sefton Merseyside 1.839741e+07   
Wirral Merseyside 3.985323e+07   
St. Helens Merseyside 2.587738e+07   
Liverpool Merseyside 7.006474e+07   
Gateshead Tyne and Wear 2.536375e+07   
Newcastle upon Tyne Tyne and Wear 3.463477e+07   
South Tyneside Tyne and Wear 1.976675e+07   
Sunderland Tyne and Wear 3.610431e+07   
North Tyneside Tyne and Wear 2.601453e+07   
Barnsley South Yorkshire 3.205748e+07   
Doncaster South Yorkshire 3.959763e+07   
Rotherham South Yorkshire 3.382336e+07   
Sheffield South Yorkshire 6.858681e+07   
Bradford West Yorkshire 6.052863e+07   
Calderdale West Yorkshire 2.668486e+07   
Leeds West Yorkshire 1.319730e+08   
Kirklees West Yorkshire 5.289108e+07   
Wakefield West Yorkshire 5.343648e+07   
  
 Surgery Satisfaction Long-term ill satisfaction   
Area   
Bolton 0.864921 0.666929   
Bury 0.864399 0.655950   
Manchester 0.821198 0.621552   
Oldham 0.830067 0.648928   
Rochdale 0.834897 0.649811   
Salford 0.862242 0.678768   
Stockport 0.875478 0.647970   
Tameside 0.811689 0.623909   
Trafford 0.867968 0.651478   
Wigan 0.874374 0.662293   
Halton 0.866262 0.654903   
Knowsley 0.870023 0.671953   
Sefton 0.804790 0.637548   
Wirral 0.898507 0.677501   
St. Helens 0.842995 0.656131   
Liverpool 0.874703 0.672319   
Gateshead 0.872378 0.687039   
Newcastle upon Tyne 0.871732 0.661459   
South Tyneside 0.891334 0.701460   
Sunderland 0.868007 0.637755   
North Tyneside 0.885928 0.657579   
Barnsley 0.833587 0.664256   
Doncaster 0.830084 0.671637   
Rotherham 0.851525 0.668100   
Sheffield 0.838628 0.637051   
Bradford 0.804472 0.643127   
Calderdale 0.875827 0.678653   
Leeds 0.856263 0.647404   
Kirklees 0.830282 0.667203   
Wakefield 0.860949 0.642704

**Data Cleaning**

list(set(nhs\_spending\_metro.index)-set(all\_uk.Area)) #check to see if there are differences in the names,   
#we were quite lucky not to find something

[]

all\_uk.index = all\_uk.Area

df\_merged\_metro = pd.merge(nhs\_spending\_metro,all\_uk,left\_index=True, right\_index=True)  
df\_merged\_metro["Spending Per Patient"]=df\_merged\_metro["Total NHS Payments"]/df\_merged\_metro["Electorate"]  
#creating the spending per patient dividing total spending per gp by the number of eligable voters

df\_merged["Metropolitan County"]="Greater London"

df\_merged = df\_merged.loc[:,["Metropolitan County","Electorate","Pct\_Remain","Pct\_Leave","Spending Per Patient","Surgery Satisfaction","Long-term ill satisfaction"]]

df\_merged\_metro = df\_merged\_metro.loc[:,["Metropolitan County","Electorate","Pct\_Remain","Pct\_Leave","Spending Per Patient","Surgery Satisfaction","Long-term ill satisfaction"]]

df\_all\_uk = pd.concat([df\_merged,df\_merged\_metro])  
df\_all\_uk  
  
#choosing specific fields in two dataframes and concat them

Metropolitan County Electorate Pct\_Remain Pct\_Leave \  
Barking and Dagenham Greater London 115812.0 37.56 62.44   
Barnet Greater London 223467.0 62.23 37.77   
Bexley Greater London 170779.0 37.05 62.95   
Brent Greater London 186793.0 59.74 40.26   
Bromley Greater London 231473.0 50.65 49.35   
... ... ... ... ...   
Bradford West Yorkshire 342817.0 45.77 54.23   
Calderdale West Yorkshire 149195.0 44.32 55.68   
Leeds West Yorkshire 543033.0 50.31 49.69   
Kirklees West Yorkshire 307081.0 45.33 54.67   
Wakefield West Yorkshire 246096.0 33.64 66.36   
  
 Spending Per Patient Surgery Satisfaction \  
Barking and Dagenham 218.098466 0.760976   
Barnet 193.037751 0.808196   
Bexley 151.447675 0.781041   
Brent 221.163597 0.778854   
Bromley 157.660619 0.821555   
... ... ...   
Bradford 176.562510 0.804472   
Calderdale 178.858923 0.875827   
Leeds 243.029431 0.856263   
Kirklees 172.238217 0.830282   
Wakefield 217.136714 0.860949   
  
 Long-term ill satisfaction   
Barking and Dagenham 0.539321   
Barnet 0.557214   
Bexley 0.605147   
Brent 0.557819   
Bromley 0.569056   
... ...   
Bradford 0.643127   
Calderdale 0.678653   
Leeds 0.647404   
Kirklees 0.667203   
Wakefield 0.642704   
  
[63 rows x 7 columns]

**Inferential Statistics**

import statsmodels.api as sm  
x = list(df\_all\_uk["Spending Per Patient"])  
y= list(df\_all\_uk["Pct\_Remain"])  
x = sm.add\_constant(x)  
model = sm.OLS(y, x).fit()  
print(model.summary())  
  
#OLS regression

OLS Regression Results   
==============================================================================  
Dep. Variable: y R-squared: 0.279  
Model: OLS Adj. R-squared: 0.267  
Method: Least Squares F-statistic: 23.59  
Date: Fri, 30 Dec 2022 Prob (F-statistic): 8.65e-06  
Time: 13:28:59 Log-Likelihood: -242.92  
No. Observations: 63 AIC: 489.8  
Df Residuals: 61 BIC: 494.1  
Df Model: 1   
Covariance Type: nonrobust   
==============================================================================  
 coef std err t P>|t| [0.025 0.975]  
------------------------------------------------------------------------------  
const 19.4266 7.079 2.744 0.008 5.272 33.581  
x1 0.1752 0.036 4.857 0.000 0.103 0.247  
==============================================================================  
Omnibus: 3.815 Durbin-Watson: 1.371  
Prob(Omnibus): 0.148 Jarque-Bera (JB): 1.875  
Skew: -0.006 Prob(JB): 0.392  
Kurtosis: 2.155 Cond. No. 949.  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

import scipy.stats as stats  
  
#normality of residuals test  
  
k,p=stats.normaltest(model.resid)  
p #null hypotheis is not rejected, residuals are normal

0.14843589177375383

import statsmodels.stats.api as sms  
  
#Heteroskedasticity test  
  
test = sms.het\_breuschpagan(model.resid, model.model.exog)  
test[1]  
  
#null hypothesis There is no heteroskedasticity:It cannot be rejected

0.13083566677279443

import statsmodels.stats.diagnostic as dg  
  
#Serial correlation test  
  
print(dg.acorr\_breusch\_godfrey(model, nlags=1))  
  
#Null hypothesis: There is no serial correlation within 1 lag: It is rejected.   
#Therefore, GLS is used

(4.755138517608326, 0.02921094165667399, 4.898428870721105, 0.030695479889219415)

ols\_resid = model.resid  
ols\_resid  
res\_fit = sm.OLS(ols\_resid[1:], ols\_resid[:-1]).fit() #test to see if residuals can be estimated with the residuals  
res\_fit.pvalues[0] #pval < .05 null hypothesis is rejected. It is indeed statistically significant  
  
from scipy.linalg import toeplitz  
order = toeplitz(range(len(ols\_resid)))  
order  
rho = res\_fit.params[0]  
rho  
sigma = rho\*\*order #we create the cov matrix  
gls\_model = sm.GLS(y,x,sigma=sigma)  
gls\_results = gls\_model.fit()  
print(gls\_results.summary()) #the results

GLS Regression Results   
==============================================================================  
Dep. Variable: y R-squared: 0.217  
Model: GLS Adj. R-squared: 0.204  
Method: Least Squares F-statistic: 16.94  
Date: Fri, 30 Dec 2022 Prob (F-statistic): 0.000118  
Time: 13:38:04 Log-Likelihood: -240.06  
No. Observations: 63 AIC: 484.1  
Df Residuals: 61 BIC: 488.4  
Df Model: 1   
Covariance Type: nonrobust   
==============================================================================  
 coef std err t P>|t| [0.025 0.975]  
------------------------------------------------------------------------------  
const 25.5509 6.907 3.699 0.000 11.740 39.362  
x1 0.1420 0.034 4.115 0.000 0.073 0.211  
==============================================================================  
Omnibus: 6.266 Durbin-Watson: 1.871  
Prob(Omnibus): 0.044 Jarque-Bera (JB): 2.460  
Skew: 0.069 Prob(JB): 0.292  
Kurtosis: 2.042 Cond. No. 721.  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

*Anovas*

df\_all\_uk["Geography"]="North" #classification 1

df\_all\_uk["Geography"][df\_all\_uk["Metropolitan County"]=="Greater London"]="South" #classification 2

/var/folders/mx/p2w4hq8s647\_rdmg660fckgh0000gn/T/ipykernel\_7161/2776133965.py:1: SettingWithCopyWarning:   
A value is trying to be set on a copy of a slice from a DataFrame  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
 df\_all\_uk["Geography"][df\_all\_uk["Metropolitan County"]=="Greater London"]="South"

#Anova test 1  
stats.f\_oneway(df\_all\_uk["Pct\_Remain"][df\_all\_uk["Geography"]=="North"],  
 df\_all\_uk["Pct\_Remain"][df\_all\_uk["Geography"]=="South"])

F\_onewayResult(statistic=37.61150836500316, pvalue=6.98246512082644e-08)

#Anova Test 2  
stats.f\_oneway(df\_all\_uk["Spending Per Patient"][df\_all\_uk["Geography"]=="North"],  
 df\_all\_uk["Spending Per Patient"][df\_all\_uk["Geography"]=="South"])

F\_onewayResult(statistic=12.741106197263345, pvalue=0.0007042770905840006)

#Anova test 1 using different method  
import statsmodels.api as sm  
from statsmodels.formula.api import ols  
  
formula = 'Pct\_Remain ~ Geography'  
  
model = ols(formula, data=df\_all\_uk).fit()  
  
aov\_table = sm.stats.anova\_lm(model, typ=2)  
aov\_table

sum\_sq df F PR(>F)  
Geography 4360.181434 1.0 37.611508 6.982465e-08  
Residual 7071.534194 61.0 NaN NaN

#Comparison testing 1  
  
import pingouin as pg  
  
pg.pairwise\_ttests(dv='Pct\_Remain',   
 between='Geography',   
 padjust='holm',   
 data=df\_all\_uk)

/Users/emmanouilxagoraris/opt/anaconda3/lib/python3.9/site-packages/pingouin/pairwise.py:27: UserWarning: pairwise\_ttests is deprecated, use pairwise\_tests instead.  
 warnings.warn("pairwise\_ttests is deprecated, use pairwise\_tests instead.", UserWarning)

Contrast A B Paired Parametric T dof \  
0 Geography North South False True -6.271932 53.424822   
  
 alternative p-unc BF10 hedges   
0 two-sided 6.474901e-08 2.524e+05 -1.527981

#Comparison testing 2  
  
pg.pairwise\_ttests(dv='Spending Per Patient',   
 between='Geography',   
 padjust='holm',   
 data=df\_all\_uk)

/Users/emmanouilxagoraris/opt/anaconda3/lib/python3.9/site-packages/pingouin/pairwise.py:27: UserWarning: pairwise\_ttests is deprecated, use pairwise\_tests instead.  
 warnings.warn("pairwise\_ttests is deprecated, use pairwise\_tests instead.", UserWarning)

Contrast A B Paired Parametric T dof \  
0 Geography North South False True -3.654082 52.653869   
  
 alternative p-unc BF10 hedges   
0 two-sided 0.000596 52.102 -0.889326

# Works Cited

Beecham, R., Slingsby, A., Brunsdon, C., 2018. Locally-varying explanations behind the United Kingdom’s vote to leave the European Union. J. Spat. Inf. Sci. 117–136. https://doi.org/10.5311/JOSIS.2018.16.377

Brandeis, 2020. Geospatial Data. Brandeis.

Doncaster LMC, 2011. The Carr Hill Formula.

IBM, 2020. K-Nearest Neighbors Algorithm.

Kochan, B., 2011. Poverty and inequality in London: anticipating the effects of tax and benefit reforms. London School of Economics, London.

Kontopantelis, E., Mamas, M.A., van Marwijk, H., Ryan, A.M., Bower, P., Guthrie, B., Doran, T., 2018. Chronic morbidity, deprivation and primary medical care spending in England in 2015-16: a cross-sectional spatial analysis. BMC Med. 16, 19. https://doi.org/10.1186/s12916-017-0996-0

Lesack, K., Naugler, C., 2011. An open-source software program for performing Bonferroni and related corrections for multiple comparisons. J. Pathol. Inform. 2, 52. https://doi.org/10.4103/2153-3539.91130

Pagano, R.R., 2013. Understanding statistics in the behavioral sciences, 10th ed. ed. Wadsworth/Cengage Learning, Australia ; Belmont, CA.

Park, K., Cha, M., Rhim, E., 2018. Positivity Bias in Customer Satisfaction Ratings, in: Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW ’18. pp. 631–638. https://doi.org/10.1145/3184558.3186579

Seltman, H., 2013. Experimental Design and Analysis. Carnegie Mellon University.

Smith, M., 2018. What regions make up the North and South of England? YourGov.co.uk.

Spiers, S., 2018. How to build houses and save the countryside, Policy Press shorts insights. Policy Press, Bristol Chicago, IL.

Wooldridge, J.M., 2013. Introductory econometrics: a modern approach, 5th ed. ed. South-Western Cengage Learning, Mason, OH.

Xu, R., Wunsch, D.C., 2009. Clustering, IEEE Press series on computational intelligence. Wiley ; IEEE Press, Hoboken, N.J. : Piscataway, NJ.

1. GeoPandas is a form of data that allows easier reading and manipulation of GDAL data [↑](#endnote-ref-2)
2. “Geospatial data contains coordinate information (e.g., latitude and longitude), which allows features to be drawn on a map. Geospatial data can be used to create maps and analyze data in GIS (Geographic Information System).” (Brandeis, 2020) [↑](#endnote-ref-3)
3. ANOVA stands for Analysis of Variation, and it tests if there is a significant difference between the mean of two or more populations (Pagano, 2013). The null hypothesis of ANOVA test is that there is no significant difference between the mean of the populations, and the alternate that there is. [↑](#endnote-ref-4)
4. A Breusch-Godfrey serial autocorrelation test was conducted (Wooldridge, 2013). The null hypothesis of the test is that no serial correlation is detected within the residuals. As Prob. = .029 < .05 within one lag the null hypothesis is rejected. Therefore, serial correlation is detected within 1 lag. [↑](#endnote-ref-5)
5. “The Carr-Hill formula is the formula that is applied to calculate the Global Sum payments for essential and some additional services. It replaced the Jarman index. This allows payments to be made based upon the cost of providing primary care services for a given population and their respective needs.” (Doncaster LMC, 2011) [↑](#endnote-ref-6)
6. We were nearly there as in the NHS database some areas were aggregate e.g., NHS entry: “Coventry and Rugby” and Referendum entry: “Coventry”, “Rugby”. We tried to do with startswith() and endswith() and using re.split() but even after that a lot of manual work was required. [↑](#endnote-ref-7)
7. “The k-nearest neighbors’ algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.” (IBM, 2020) [↑](#endnote-ref-8)