Analyzing the Impact of Socio-Economic Factors on Health Deprivation and Disability Deprivation 2019

Ruben Singh Sodhi

2024-12-08

#Introduction Health deprivation and disability are critical indicators of inequality in communities. This analysis leverages the Index of Multiple Deprivation (IMD) 2019 data to examine the relationship between socio-economic factors and health outcomes across England's Lower Layer Super Output Areas (LSOAs).

```
# Loading the dataset

library(foreign)

# Specify the path to your DBF file
dbf_file_path <- "F:\\documents copy\\DataScience\\English IMD 2019\\English IMD 2019\\IMD_2019.dbf"

# Read the DBF file
imd_data <- read.dbf(dbf_file_path)
```

```
## Field name: 'Pop60+' changed to: 'Pop60.'
```

```
# Print the first few rows of the data
head(df)
```

```
##
## 1 function (x, df1, df2, ncp, log = FALSE)
## 2 {
## 3     if (missing(ncp))
## 4         .Call(C_df, x, df1, df2, log)
## 5     else .Call(C_dnf, x, df1, df2, ncp, log)
## 6 }
```

```
# Checking for missing values
missing_values <- sapply(imd_data, function(x) sum(is.na(x)))

missing_values
```

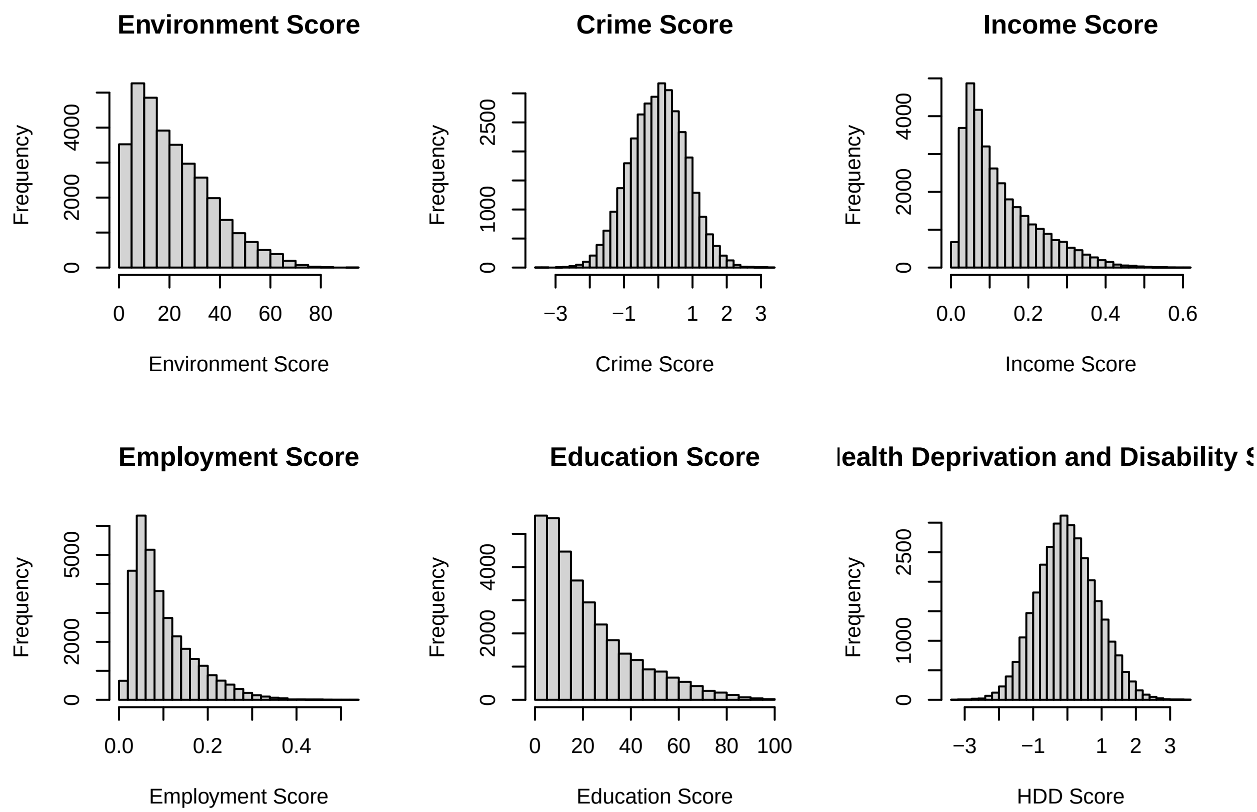
```
##   lsoa11cd   lsoa11nm  lsoa11nmw st_areasha st_lengths   IMD_Rank IMD_Decile
##          0          0          0          0          0          0          0
##   LSOA01NM      LADcd      LADnm   IMDScore   IMDRank0    IMDDec0   IncScore
##          0          0          0          0          0          0          0
##    IncRank     IncDec   EmpScore    EmpRank     EmpDec   EduScore    EduRank
##          0          0          0          0          0          0          0
```

```
##      EduDec    HDDScore     HDDRank      HDDDec    CriScore     CriRank      CriDec
##           0           0           0           0           0           0           0
##    BHSScore     BHSRank      BHSDec    EnvScore     EnvRank      EnvDec    IDCScore
##           0           0           0           0           0           0           0
##     IDCRank      IDCDec    IDOScore     IDORank      IDODec    CYPScore     CYPRank
##           0           0           0           0           0           0           0
##      CYPDec     ASScore      ASRank       ASDec     GBScore      GBRank       GBDec
##           0           0           0           0           0           0           0
##     WBScore      WBRank       WBDec    IndScore     IndRank      IndDec    OutScore
##           0           0           0           0           0           0           0
##     OutRank      OutDec      TotPop      DepChi    Pop16_59      Pop60.     WorkPop
##           0           0           0           0           0           0           0
```

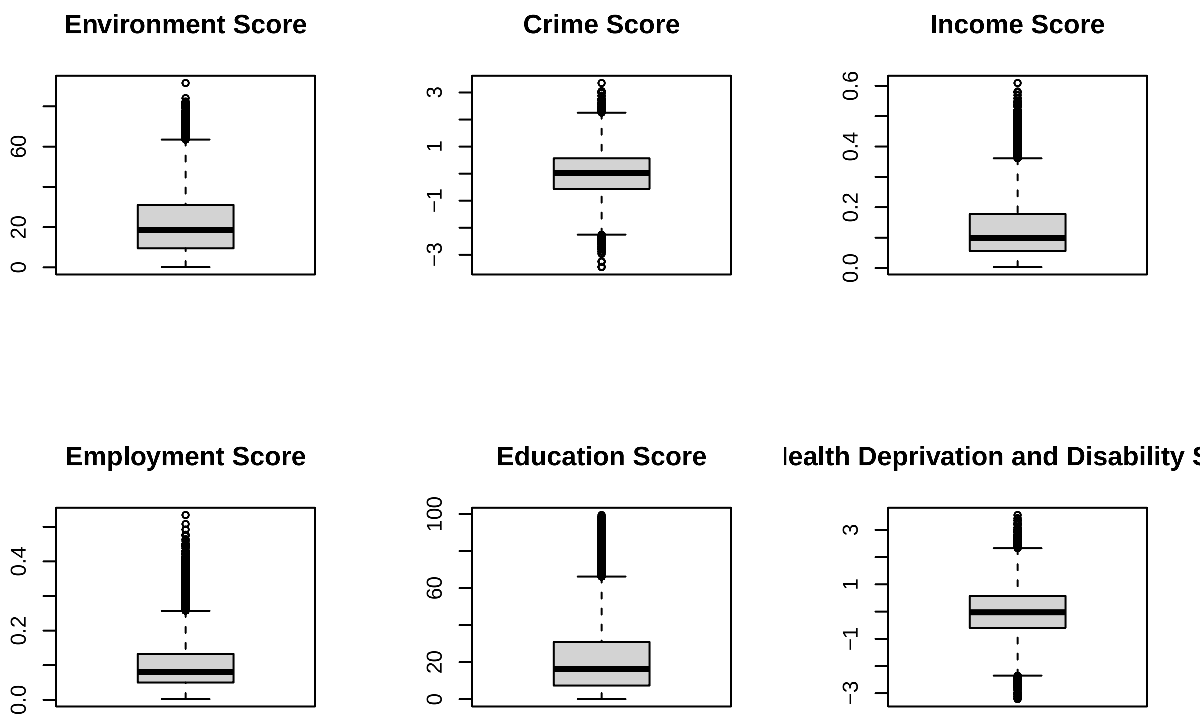
```
# Load required libraries
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

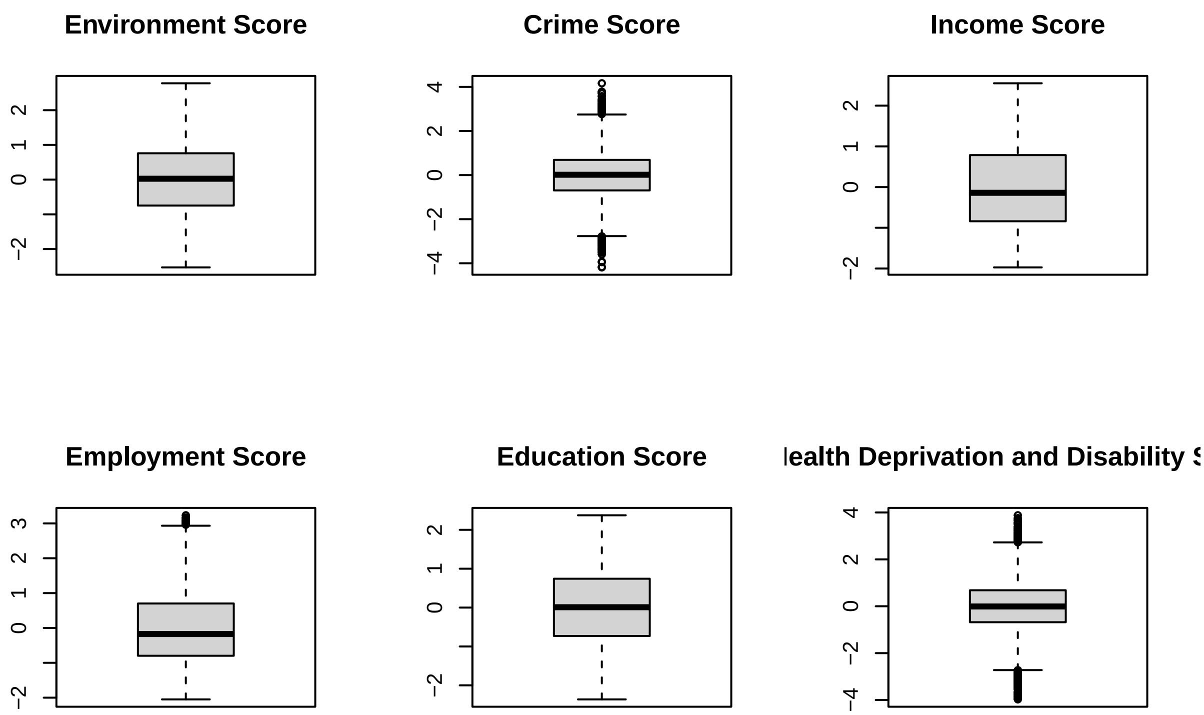
```
library(bestNormalize) # for Yeo-Johnson transformation
library(e1071)


# Selecting relevant score columns for analysis
relevant_data <- imd_data %>% select(EnvScore, CriScore, IncScore, EmpScore, EduScore, HDDScore)

par(mfrow = c(2,3)) # setting layout for multiple plots
hist(relevant_data$EnvScore, main = "Environment Score", xlab = "Environment Score", breaks = 30)
hist(relevant_data$CriScore, main = "Crime Score", xlab = "Crime Score", breaks = 30)
hist(relevant_data$IncScore, main = "Income Score", xlab = "Income Score", breaks = 30)
hist(relevant_data$EmpScore, main = "Employment Score", xlab = "Employment Score", breaks = 30)
hist(relevant_data$EduScore, main = "Education Score", xlab = "Education Score", breaks = 30)
hist(relevant_data$HDDScore, main = "Health Deprivation and Disability Score", xlab = "HDD Score", breal
```



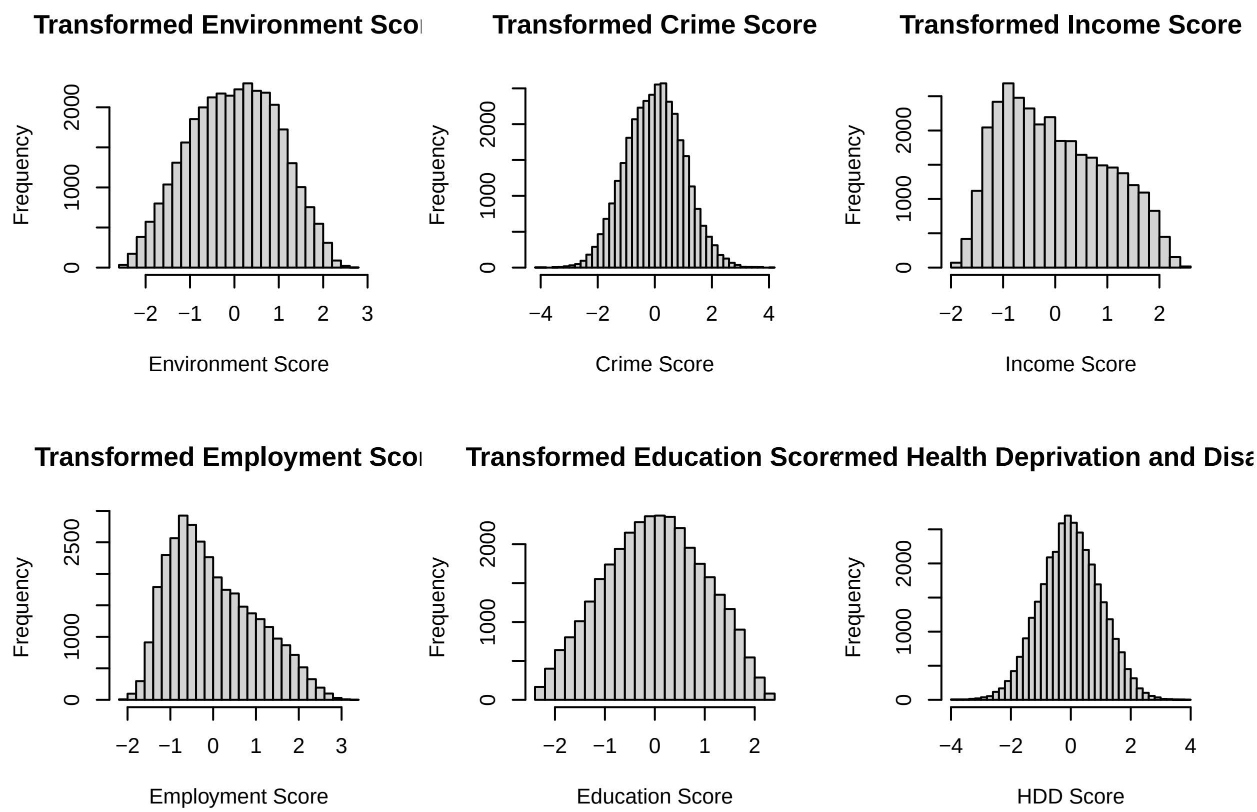

```
# Checking for outliers with boxplots
par(mfrow = c(2,3)) # setting layout for multiple plots
boxplot(relevant_data$EnvScore, main = "Environment Score")
boxplot(relevant_data$CriScore, main = "Crime Score")
boxplot(relevant_data$IncScore, main = "Income Score")
boxplot(relevant_data$EmpScore, main = "Employment Score")
boxplot(relevant_data$EduScore, main = "Education Score")
boxplot(relevant_data$HDDScore, main = "Health Deprivation and Disability Score")
```




```r
# Applying the Yeo-Johnson transformation
transformed_data <- relevant_data
transformed_data$CriScore <- yeojohnson(relevant_data$CriScore)$x.t
transformed_data$EnvScore <- yeojohnson(relevant_data$EnvScore)$x.t
transformed_data$IncScore <- yeojohnson(relevant_data$IncScore)$x.t
transformed_data$EmpScore <- yeojohnson(relevant_data$EmpScore)$x.t
transformed_data$EduScore <- yeojohnson(relevant_data$EduScore)$x.t
transformed_data$HDDScore <- yeojohnson(relevant_data$HDDScore)$x.t

# Checking for outliers with boxplots
par(mfrow = c(2,3)) # setting layout for multiple plots
boxplot(transformed_data$EnvScore, main = "Environment Score")
boxplot(transformed_data$CriScore, main = "Crime Score")
boxplot(transformed_data$IncScore, main = "Income Score")
boxplot(transformed_data$EmpScore, main = "Employment Score")
boxplot(transformed_data$EduScore, main = "Education Score")
boxplot(transformed_data$HDDScore, main = "Health Deprivation and Disability Score")
```




```r
# Visualizing the transformed distributions
par(mfrow = c(2,3)) # setting layout for multiple plots
hist(transformed_data$EnvScore, main = "Transformed Environment Score", xlab = "Environment Score", brea
hist(transformed_data$CriScore, main = "Transformed Crime Score", xlab = "Crime Score", breaks = 30)
hist(transformed_data$IncScore, main = "Transformed Income Score", xlab = "Income Score", breaks = 30)
hist(transformed_data$EmpScore, main = "Transformed Employment Score", xlab = "Employment Score", breaks
hist(transformed_data$EduScore, main = "Transformed Education Score", xlab = "Education Score", breaks =
hist(transformed_data$HDDScore, main = "Transformed Health Deprivation and Disability Score", xlab = "HI
```




```r
# Standardizing the transformed data
standardized_data <- scale(transformed_data)

# Viewing the first few rows of the standardized data
head(standardized_data)
```

```
##         EnvScore   CriScore   IncScore    EmpScore    EduScore      HDDScore
## [1,] 0.7989338 -2.4479749 -1.8738490 -1.81489303 -2.35542108 -1.984591311
## [2,] 0.3219819 -2.8467017 -1.2674645 -1.35366137 -2.33229052 -1.315967410
## [3,] 1.1900178 -1.2630107 -0.3347932 -0.06235003 -0.92459217 -0.096219412
## [4,] 0.6526825 -1.6083970  1.0740507  0.74535344  0.35591738 -0.118513244
## [5,] 0.5409139 -0.1849376  0.1055477 -0.59998515 -0.08291078 -0.399985290
## [6,] 0.4914391  1.0387284  1.0415984  0.30364300 -0.34144358 -0.008496015
```

```r
# Converting the standardized data back into a data frame
standardized_data_df <- as.data.frame(standardized_data)

# Calculating skewness for the original data
original_skewness <- sapply(relevant_data, skewness)

# Calculating skewness for the transformed data
transformed_skewness <- sapply(transformed_data, skewness)

# Displaying the skewness results
print("Original Data Skewness")
```

```
## [1] "Original Data Skewness"
```

```r
print(original_skewness)
```

```
##    EnvScore    CriScore    IncScore    EmpScore    EduScore    HDDScore
##  0.87257486 -0.01866299  1.19521021  1.34432277  1.22379385  0.11235197
```

```r
print("Transformed Data Skewness")
```

```
## [1] "Transformed Data Skewness"
```

```r
print(transformed_skewness)
```

```
##     EnvScore     CriScore     IncScore     EmpScore     EduScore     HDDScore
## -0.059663502  0.005393514  0.337828664  0.536872102 -0.045826460 -0.004834012
```

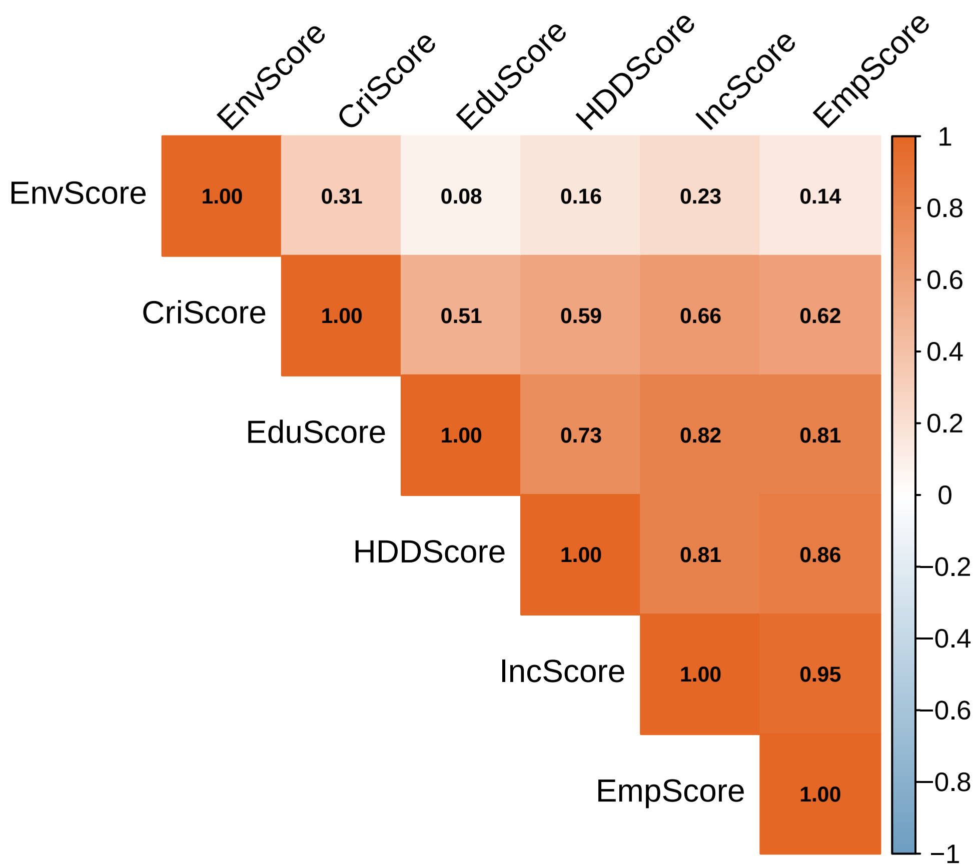
```r
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
# Assuming transformed_data is your Yeo-Johnson transformed dataframe

# Correlation Analysis using Pearson's method
correlation_matrix <- cor(transformed_data, method = "pearson")

# Displaying the correlation matrix
print(correlation_matrix)
```

```
##            EnvScore  CriScore  IncScore  EmpScore   EduScore  HDDScore
## EnvScore 1.00000000 0.3111339 0.2315555 0.1447280 0.08187578 0.1646777
## CriScore 0.31113386 1.0000000 0.6574347 0.6173720 0.51173112 0.5884259
## IncScore 0.23155553 0.6574347 1.0000000 0.9517945 0.81545997 0.8123385
## EmpScore 0.14472802 0.6173720 0.9517945 1.0000000 0.81145122 0.8584064
## EduScore 0.08187578 0.5117311 0.8154600 0.8114512 1.00000000 0.7344767
## HDDScore 0.16467766 0.5884259 0.8123385 0.8584064 0.73447666 1.0000000
```

```r
# Advanced Visualization of the Correlation Matrix
corrplot(correlation_matrix, method = "color",
         type = "upper", # Display only the upper half of the matrix
         order = "hclust", # Hierarchical clustering to group similar variables
         tl.col = "black", # Color for text labels
         tl.srt = 45, # Text label rotation
         addCoef.col = "black", # Color for correlation coefficients
         number.cex = .7, # Size of the correlation coefficients
         col = colorRampPalette(c("#6D9EC1", "white", "#E46726"))(200)) # Custom color palette
```




```
# Load necessary library
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# Linear regression models
model_env_hdd <- lm(HDDScore ~ EnvScore, data = standardized_data_df)
model_cri_hdd <- lm(HDDScore ~ CriScore, data = standardized_data_df)
model_inc_hdd <- lm(HDDScore ~ IncScore, data = standardized_data_df)
model_emp_hdd <- lm(HDDScore ~ EmpScore, data = standardized_data_df)
model_edu_hdd <- lm(HDDScore ~ EduScore, data = standardized_data_df)

# Summary of the models
summary(model_env_hdd)
```

```
##
## Call:
```

```
## lm(formula = HDDScore ~ EnvScore, data = standardized_data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2332 -0.6677 -0.0107  0.6549  3.6148
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.553e-16  5.443e-03    0.00        1
## EnvScore     1.647e-01  5.443e-03   30.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9864 on 32842 degrees of freedom
## Multiple R-squared:  0.02712,    Adjusted R-squared:  0.02709
## F-statistic: 915.5 on 1 and 32842 DF,  p-value: < 2.2e-16
```

```
summary(model_cri_hdd)
```

```
##
## Call:
## lm(formula = HDDScore ~ CriScore, data = standardized_data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5322 -0.5161  0.0325  0.5501  3.4861
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.959e-16  4.462e-03     0.0        1
## CriScore     5.884e-01  4.462e-03   131.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8086 on 32842 degrees of freedom
## Multiple R-squared:  0.3462, Adjusted R-squared:  0.3462
## F-statistic: 1.739e+04 on 1 and 32842 DF,  p-value: < 2.2e-16
```

```
summary(model_inc_hdd)
```

```
##
## Call:
## lm(formula = HDDScore ~ IncScore, data = standardized_data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7694 -0.3733  0.0090  0.3950  3.2520
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.311e-15  3.218e-03     0.0        1
## IncScore     8.123e-01  3.218e-03   252.4   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5832 on 32842 degrees of freedom
## Multiple R-squared:  0.6599, Adjusted R-squared:  0.6599
## F-statistic: 6.372e+04 on 1 and 32842 DF,  p-value: < 2.2e-16
```

```
summary(model_emp_hdd)
```

```
##
## Call:
## lm(formula = HDDScore ~ EmpScore, data = standardized_data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4389 -0.3184 -0.0016  0.3249  3.5036
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.245e-15  2.831e-03     0.0        1
## EmpScore     8.584e-01  2.831e-03   303.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.513 on 32842 degrees of freedom
## Multiple R-squared:  0.7369, Adjusted R-squared:  0.7369
## F-statistic: 9.197e+04 on 1 and 32842 DF,  p-value: < 2.2e-16
```

```
summary(model_edu_hdd)
```

```
##
## Call:
## lm(formula = HDDScore ~ EduScore, data = standardized_data_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.91405 -0.46652 -0.00347  0.46118  2.75259
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.296e-15  3.745e-03     0.0        1
## EduScore     7.345e-01  3.745e-03   196.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6786 on 32842 degrees of freedom
## Multiple R-squared:  0.5395, Adjusted R-squared:  0.5394
## F-statistic: 3.847e+04 on 1 and 32842 DF,  p-value: < 2.2e-16
```

```
# Additional: To get p-values and confidence intervals
coeftest(model_env_hdd)
```

```
##
```

```
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -9.5529e-16  5.4426e-03   0.000        1
## EnvScore     1.6468e-01  5.4427e-03  30.257   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(model_env_hdd)
```

```
##                   2.5 %     97.5 %
## (Intercept) -0.01066774 0.01066774
## EnvScore     0.15400975 0.17534556
```

```
coeftest(model_cri_hdd)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -8.9588e-16  4.4616e-03    0.00        1
## CriScore     5.8843e-01  4.4616e-03  131.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(model_cri_hdd)
```

```
##                    2.5 %      97.5 %
## (Intercept) -0.008744807 0.008744807
## CriScore     0.579680911 0.597170791
```

```
coeftest(model_inc_hdd)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -1.3109e-15  3.2180e-03    0.00        1
## IncScore     8.1234e-01  3.2180e-03  252.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(model_inc_hdd)
```

```
##                    2.5 %      97.5 %
## (Intercept) -0.006307393 0.006307393
## IncScore     0.806031031 0.818646010
```

```r
coeftest(model_emp_hdd)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -4.2445e-15  2.8305e-03    0.00        1
## EmpScore     8.5841e-01  2.8306e-03  303.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
confint(model_emp_hdd)
```

```
##                    2.5 %      97.5 %
## (Intercept) -0.005547979 0.005547979
## EmpScore     0.852858360 0.863954487
```

```r
coeftest(model_edu_hdd)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -1.2963e-15  3.7447e-03    0.00        1
## EduScore     7.3448e-01  3.7447e-03  196.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
confint(model_edu_hdd)
```

```
##                    2.5 %      97.5 %
## (Intercept) -0.007339699 0.007339699
## EduScore     0.727136851 0.741816473
```

```r
# Load necessary library
library(lmtest)


# Multivariate Linear Regression Analysis
# Model with IncScore, EmpScore, and EduScore as independent variables
model_multi <- lm(HDDScore ~ IncScore + EmpScore + EduScore + CriScore + EnvScore, data = standardized_d

# Summary of the multivariate model
summary(model_multi)
```

```
##
## Call:
## lm(formula = HDDScore ~ IncScore + EmpScore + EduScore + CriScore +
##     EnvScore, data = standardized_data_df)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.51240 -0.31612 -0.00378  0.31303  3.13388
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.329e-15  2.752e-03    0.00        1
## IncScore    -2.369e-01  1.005e-02  -23.57   <2e-16 ***
## EmpScore     8.985e-01  9.398e-03   95.60   <2e-16 ***
## EduScore     1.425e-01  4.907e-03   29.05   <2e-16 ***
## CriScore     1.022e-01  3.743e-03   27.31   <2e-16 ***
## EnvScore     4.602e-02  3.025e-03   15.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4988 on 32838 degrees of freedom
## Multiple R-squared:  0.7512, Adjusted R-squared:  0.7512
## F-statistic: 1.983e+04 on 5 and 32838 DF,  p-value: < 2.2e-16
```

```
# Additional: To get p-values and confidence intervals
coeftest(model_multi)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -4.3288e-15  2.7524e-03   0.000        1
## IncScore    -2.3690e-01  1.0050e-02 -23.572   <2e-16 ***
## EmpScore     8.9847e-01  9.3980e-03  95.602   <2e-16 ***
## EduScore     1.4251e-01  4.9065e-03  29.045   <2e-16 ***
## CriScore     1.0224e-01  3.7432e-03  27.313   <2e-16 ***
## EnvScore     4.6022e-02  3.0246e-03  15.216   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(model_multi)
```

```
##                   2.5 %       97.5 %
## (Intercept) -0.005394867  0.005394867
## IncScore    -0.256601399 -0.217204312
## EmpScore     0.880047883  0.916888702
## EduScore     0.132894786  0.152128750
## CriScore     0.094901166  0.109574907
## EnvScore     0.040093994  0.051950614
```

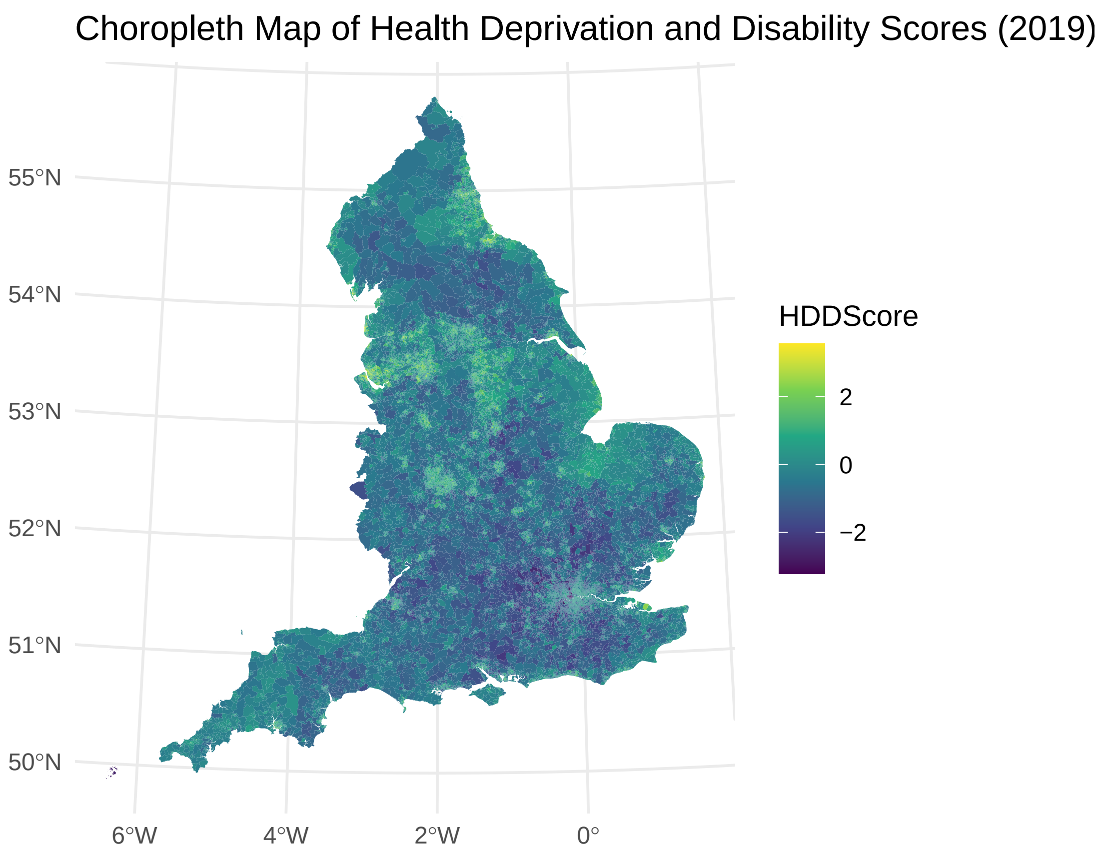
```
#Spatial Data Analysis

library(sf)
```

```
## Linking to GEOS 3.12.2, GDAL 3.9.3, PROJ 9.4.1; sf_use_s2() is TRUE
```

```r
library(ggplot2)
library(dplyr)


# Loading the shapefile
shapefile_path <- "F:\\documents copy\\DataScience\\English IMD 2019\\English IMD 2019\\IMD_2019.shp"
gdf <- st_read(shapefile_path)
```

```
## Reading layer `IMD_2019' from data source
##   `F:\documents copy\DataScience\English IMD 2019\English IMD 2019\IMD_2019.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 32844 features and 63 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 82679.8 ymin: 5343.899 xmax: 655604.7 ymax: 657534.1
## Projected CRS: OSGB36 / British National Grid
```

```r
# Merging the HDDScore data with the shapefile data
gdf_merged <- merge(gdf, imd_data, by = "lsoa11cd")


library(sf)
library(ggplot2)

# Assuming 'gdf_merged' is your merged GeoDataFrame with HDDScore data
# Make sure to replace 'HDDScore' with the actual column name for the HDDScore in your dataset

# Creating a choropleth map
ggplot(data = gdf_merged) +
  geom_sf(aes(fill = HDDScore.x), color = NA) +  # Replace HDDScore with the correct column name
  scale_fill_viridis_c(option = "D") +  # Color scale, can be adjusted as needed
  labs(title = "Choropleth Map of Health Deprivation and Disability Scores (2019)",
       fill = "HDDScore") +
  theme_minimal()
```



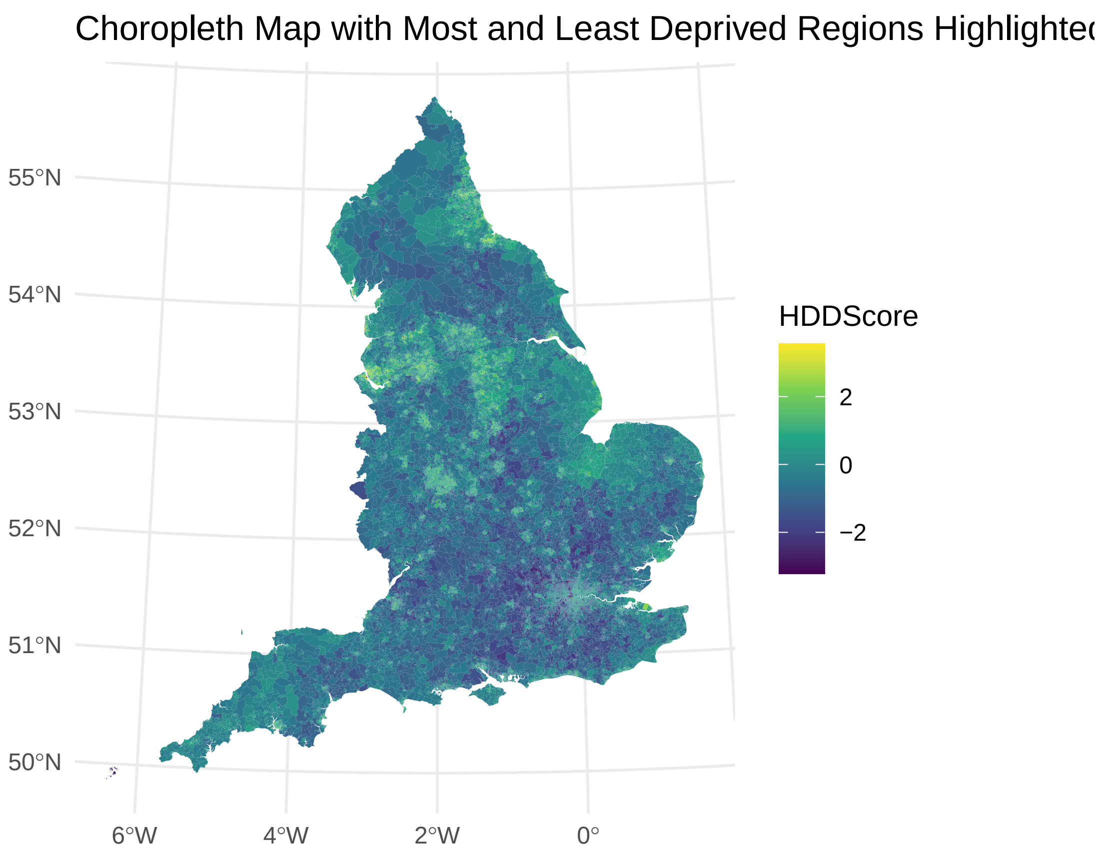

```
# Assuming 'gdf_merged' is your merged GeoDataFrame with HDDScore data in R

# Sorting the data by HDDRank
sorted_gdf <- imd_data %>% arrange(HDDRank)

# Identifying the most and least deprived regions
most_deprived_regions <- head(sorted_gdf, 5)
least_deprived_regions <- tail(sorted_gdf, 5)

# Extracting relevant information
most_deprived_info <- most_deprived_regions %>% select(lsoa11cd, lsoa11nm, HDDScore, HDDRank)
least_deprived_info <- least_deprived_regions %>% select(lsoa11cd, lsoa11nm, HDDScore, HDDRank)

# Displaying the information
print(most_deprived_info)
```

```
##    lsoa11cd                   lsoa11nm HDDScore HDDRank
## 1 E01012681             Blackpool 006A    3.547       1
## 2 E01007126                Wirral 016A    3.429       2
## 3 E01012673             Blackpool 010A    3.351       3
## 4 E01012655 Blackburn with Darwen 006E    3.345       4
## 5 E01012751             Blackpool 013D    3.255       5
```

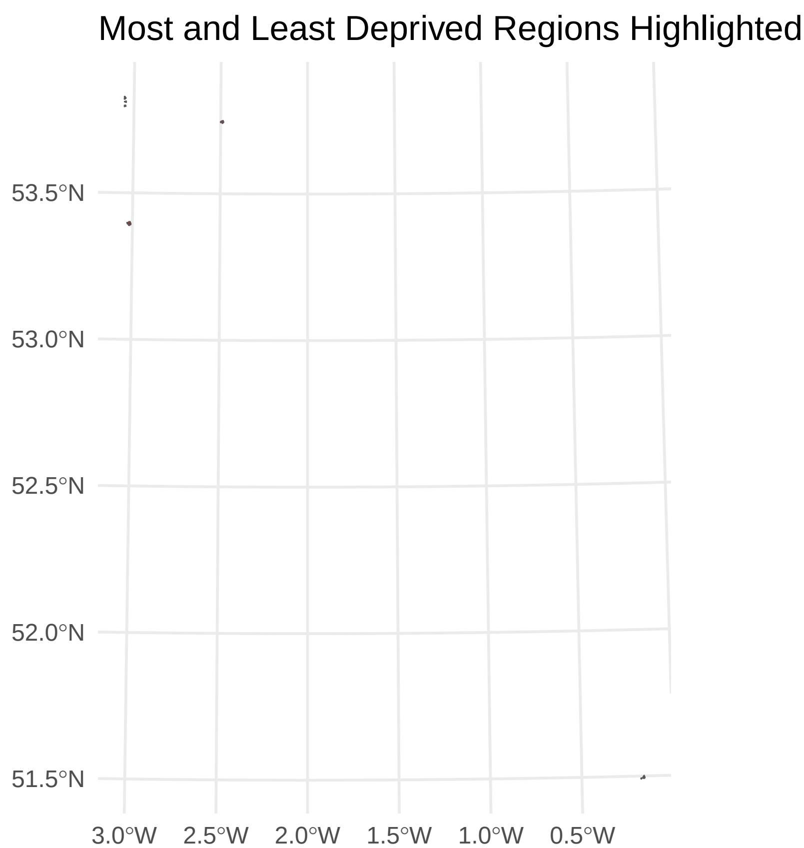
```
print(least_deprived_info)
```

```
##        lsoa11cd                    lsoa11nm HDDScore HDDRank
## 32840 E01002822 Kensington and Chelsea 016B   -3.105   32840
## 32841 E01002818 Kensington and Chelsea 012A   -3.123   32841
## 32842 E01002863 Kensington and Chelsea 012E   -3.134   32842
## 32843 E01002860 Kensington and Chelsea 012D   -3.188   32843
## 32844 E01002859 Kensington and Chelsea 012C   -3.215   32844
```

```r
# Sorting the data by HDDRank
sorted_gdf <- gdf_merged %>% arrange(HDDRank.x)

# Identifying the most and least deprived regions
most_deprived_regions <- head(sorted_gdf, 5)
least_deprived_regions <- tail(sorted_gdf, 5)

# Creating the base choropleth map
base_map <- ggplot(data = gdf_merged) +
  geom_sf(aes(fill = HDDScore.x), color = NA) +  # Replace HDDScore with the actual column name
  scale_fill_viridis_c(option = "D") +  # Color scale
  labs(title = "Choropleth Map with Most and Least Deprived Regions Highlighted",
       fill = "HDDScore") +
  theme_minimal()

# Adding layers for most and least deprived regions
final_map <- base_map +
  geom_sf(data = most_deprived_regions, fill = "red", color = NA) +
  geom_sf(data = least_deprived_regions, fill = "green", color = NA)

# Display the map
print(final_map)
```




```r
# Creating the map with only the most and least deprived regions
highlight_map <- ggplot() +
  geom_sf(data = most_deprived_regions, fill = "red") +
  geom_sf(data = least_deprived_regions, fill = "green") +
  labs(title = "Most and Least Deprived Regions Highlighted",
       fill = "Region Type") +
  theme_minimal()

# Display the map
print(highlight_map)
```




```
# Load necessary libraries
library(dplyr)
library(sf)
library(stringr)
library(ggplot2)


# 'lsoa11nm.x' (LSOA names) and 'HDDScore.x' (HDD score)

# Extract the prefix of the LSOA names (first part before the 4-digit code)
gdf_merged <- gdf_merged %>%
  mutate(LSOA_prefix = str_sub(lsoa11nm.x, 1, nchar(lsoa11nm.x) - 4))  # Get prefix by removing last 4

# Group the data by LSOA prefix and calculate the mean HDDScore for each group
grouped_data <- gdf_merged %>%
  group_by(LSOA_prefix) %>%
  summarise(mean_HDDScore = mean(HDDScore.x, na.rm = TRUE))  # Replace 'HDDScore.x' with your actual co

# Sort the groups by mean HDDScore
sorted_groups <- grouped_data %>% arrange(mean_HDDScore)

# Identify the top 20 and bottom 20 groups based on mean HDDScore
most_deprived_groups <- head(sorted_groups, 20)
least_deprived_groups <- tail(sorted_groups, 20)

# Print the tables for the top 5 and bottom 5 groups
```

```r
print("Top 20 Most Deprived Groups:")
```

```
## [1] "Top 20 Most Deprived Groups:"
```

```r
print(most_deprived_groups)
```

```
## Simple feature collection with 20 features and 2 fields
## Geometry type: GEOMETRY
## Dimension:     XY
## Bounding box:  xmin: 82679.8 ymin: 5343.899 xmax: 573611.2 ymax: 349291.8
## Projected CRS: OSGB36 / British National Grid
## # A tibble: 20 x 3
##    LSOA_prefix              mean_HDDScore                             geometry
##    <chr>                            <dbl>                       <GEOMETRY [m]>
##  1 "Isles of Scilly "               -2.62 MULTIPOLYGON (((91614.5 12996.4, 917~
##  2 "Wokingham "                     -1.53 POLYGON ((470925.9 162796.6, 470645.~
##  3 "Chiltern "                      -1.53 POLYGON ((493953.2 192009.5, 493859.~
##  4 "Hart "                          -1.52 POLYGON ((476473.5 146015.7, 476547.~
##  5 "Elmbridge "                     -1.46 POLYGON ((511809.2 158109, 511886.1 ~
##  6 "East Hertfordshire "            -1.40 POLYGON ((536286.3 210251.7, 536333.~
##  7 "Richmond upon Thames "          -1.38 MULTIPOLYGON (((513103.4 168953.4, 5~
##  8 "St Albans "                     -1.31 POLYGON ((515227.5 201318.8, 515171.~
##  9 "Uttlesford "                    -1.30 POLYGON ((554460.9 212294.8, 554409.~
## 10 "South Bucks "                   -1.29 POLYGON ((493217.2 180906.2, 493290 ~
## 11 "South Oxfordshire "             -1.22 POLYGON ((458861 182616.7, 458535.8 ~
## 12 "Mid Sussex "                    -1.22 POLYGON ((531747.4 115179.7, 531656.~
## 13 "Kensington and Chelsea "        -1.21 POLYGON ((524466.8 179009.1, 524355.~
## 14 "Epsom and Ewell "               -1.15 POLYGON ((519354.6 159452.8, 519287.~
## 15 "South Cambridgeshire "          -1.12 POLYGON ((529968.6 238384.2, 529614.~
## 16 "Vale of White Horse "           -1.10 POLYGON ((440755.3 182741.2, 440583.~
## 17 "Rushcliffe "                    -1.09 POLYGON ((459398.7 324846.3, 459301.~
## 18 "Three Rivers "                  -1.09 POLYGON ((503939.7 193345.5, 503899.~
## 19 "Barnet "                        -1.09 POLYGON ((523954.5 185541.5, 523906.~
## 20 "Brentwood "                     -1.05 POLYGON ((556495.2 191535.1, 556470.~
```

```r
print("Bottom 20 Least Deprived Groups:")
```

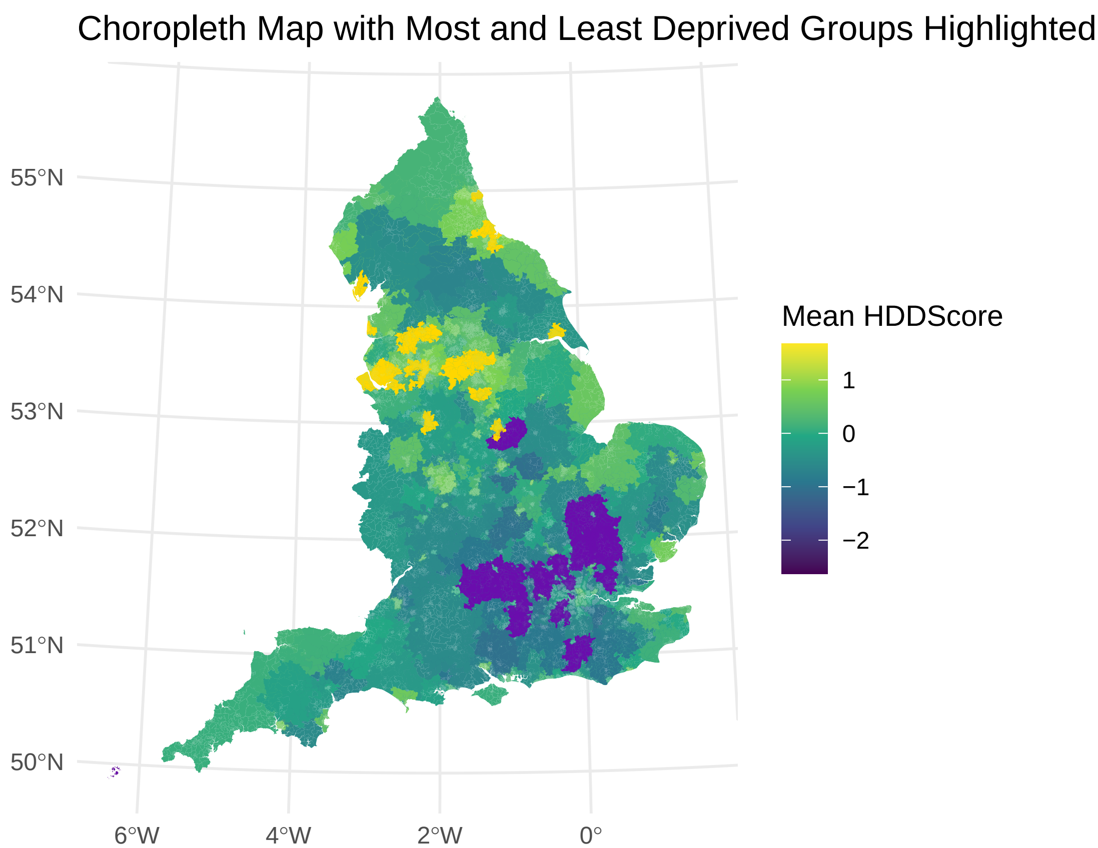
```
## [1] "Bottom 20 Least Deprived Groups:"
```

```r
print(least_deprived_groups)
```

```
## Simple feature collection with 20 features and 2 fields
## Geometry type: GEOMETRY
## Dimension:     XY
## Bounding box:  xmin: 316695.1 ymin: 332712.1 xmax: 516044.6 ymax: 568696.3
## Projected CRS: OSGB36 / British National Grid
## # A tibble: 20 x 3
##    LSOA_prefix             mean_HDDScore                              geometry
##    <chr>                           <dbl>                        <GEOMETRY [m]>
##  1 "Barnsley "                     0.843 POLYGON ((433553.6 398232.6, 433320.3~
##  2 "Kingston upon Hull "           0.849 MULTIPOLYGON (((505417.9 426319.8, 50~
```

```
## 3 "South Tyneside "                0.866 POLYGON ((433359.2 559582.8, 433292.6~
## 4 "Nottingham "                    0.874 POLYGON ((454190.5 333611.8, 453964.6~
## 5 "Hartlepool "                    0.874 MULTIPOLYGON (((449414.7 526577.4, 44~
## 6 "Chesterfield "                  0.900 POLYGON ((437289.9 369337.3, 437209.5~
## 7 "Wirral "                        0.961 MULTIPOLYGON (((325361.6 380194.7, 32~
## 8 "Halton "                        0.968 MULTIPOLYGON (((351310.6 379608.4, 35~
## 9 "Salford "                       0.991 POLYGON ((370228.7 391203, 370145.8 3~
## 10 "Stoke-on-Trent "               1.02  POLYGON ((386746.3 341049.1, 386635.5~
## 11 "Blackburn with Darwen "        1.06  POLYGON ((364487.2 424712.6, 364467.6~
## 12 "Hyndburn "                     1.07  POLYGON ((372966.5 423266.5, 372829 4~
## 13 "St. Helens "                   1.07  POLYGON ((347786.2 389660.5, 347792.8~
## 14 "Burnley "                      1.08  POLYGON ((380225 428107, 380132.9 428~
## 15 "Middlesbrough "                1.19  MULTIPOLYGON (((450106.1 513238.5, 44~
## 16 "Manchester "                   1.20  POLYGON ((382484 384151.8, 382529.5 3~
## 17 "Barrow-in-Furness "            1.26  MULTIPOLYGON (((317158.8 468686.6, 31~
## 18 "Liverpool "                    1.34  POLYGON ((335613.3 386985.1, 335562.9~
## 19 "Knowsley "                     1.40  POLYGON ((341896.1 389707.7, 341741.7~
## 20 "Blackpool "                    1.67  POLYGON ((330832.4 431761.3, 330640 4~
```

```r
# Join the grouped data back with the original spatial data to get the mean HDDScore in the spatial dat
gdf_merged_with_group <- st_join(gdf_merged, grouped_data, by = "LSOA_prefix")

# Create the base choropleth map, coloring by mean HDDScore
base_map <- ggplot(data = gdf_merged_with_group) +
  geom_sf(aes(fill = mean_HDDScore), color = NA) +  # Color regions by mean HDDScore
  scale_fill_viridis_c(option = "D") +  # Use a color scale
  labs(title = "Choropleth Map with Most and Least Deprived Groups Highlighted",
       fill = "Mean HDDScore") +
  theme_minimal()

# Highlight the top 5 and bottom 5 groups by their prefixes
final_map <- base_map +
  geom_sf(data = gdf_merged_with_group %>% filter(LSOA_prefix.y %in% most_deprived_groups$LSOA_prefix),
          fill = "#6A0DAD", color = NA) +
  geom_sf(data = gdf_merged_with_group %>% filter(LSOA_prefix.y %in% least_deprived_groups$LSOA_prefix)
          fill = "#FFD700", color = NA)

# Display the final map
print(final_map)
```




```r
# Optionally, create a map showing only the top 5 and bottom 5 groups
highlight_map <- ggplot() +
  geom_sf(data = gdf_merged_with_group %>% filter(LSOA_prefix.y %in% most_deprived_groups$LSOA_prefix),
          fill = "#6A0DAD", color = NA) +
  geom_sf(data = gdf_merged_with_group %>% filter(LSOA_prefix.y %in% least_deprived_groups$LSOA_prefix)
          fill = "#FFD700", color = NA) +
  labs(title = "Most and Least Deprived Groups Highlighted") +
  theme_minimal()

# Display the highlight map
print(highlight_map)
```

Most and Least Deprived Groups Highlighted