

# Enabling backwards compatibility in the Index of Multiple Deprivation with an illustrative analysis of mortality rates

Matthew Tibbles

06/02/2022

## Summary

## Context

The Index of Multiple Deprivation (IMD) is the standard measure of deprivation and poverty in England. It is widely used by government and researchers to allocate resources and inform recommendations on health and social policy. Since 2004 the IMD has been based on the weighted measurement of seven “domains” of deprivation, each of which is comprised of multiple indicators based on official government data for around 32,000 neighborhoods - Lower Level Super Output Areas (LSOA). The weight attributed to each domain has remained consistent over time. However, the composition and measurement of the underlying indicators of each domain has undergone multiple changes - often in successive versions of the IMD. On the one hand, this adaptive design philosophy makes the IMD highly robust to policy changes and shifts in research and data standards, ensuring that each version of the IMD captures the determinants of deprivation as defined by the contexts of that time. On the other hand, differences between each IMD preclude reliable statistical analysis of the effects of deprivation across time. Therefore, in a distinct lack of understanding of the effects of relative change in deprivation levels. Put another way, while the IMD is a rich source for analyzing the effects of *being* more or less deprived, its widespread usage has somewhat stymied our understanding of the effects of *becoming* more or less deprived, as well as of possible inter-dependencies between these effects.

## Objectives

This research is guided by three main objectives:

1. To devise a method for creating a time-adjusted IMD, enabling analysis of the effects of deprivation across time
2. To apply of this method to the analysis of mortality rates, leveraging the explanatory potential of Random Effects Within Between panel models # Enabling backwards compatibility

## Method

## Crtiteria

In the 2015 and 2019 technical reports, the architects of the IMD state that previous versions “should not be used as a time series” for the following reasons:

1. The addition of new indicators
2. Changes to the measurement of existing indicators
3. Changes to LSOA boundaries

I thus present a method to enable backwards compatibility via the removal/minimization of the impacts of the above.

## Regression scaling

The method I employ is adapted from the work of Abel and colleagues, who have constructed a UK-wide IMD which enables between-country comparisons despite variation in the weighting and measurement of the domains which form each country's IMD. At its core, their method is designed to exploit the fact that the weighting and measurement of the income and employment domains is "approximately comparable across countries". They first fit a simple linear regression for each country, with IMD score as the outcome variable and income and employment domain scores as covariates. They then scale the income and employment scores - and the regression residuals, which "represent the unique contribution of the domains, other than income and employment, to the overall IMD" - of each country by the regression equation of a given reference country.

Taking Northern Ireland,  $n$ , as the reference, adjusted IMD for country  $c$  can be expressed as:

$$IMD_{c-ADJUSTEDi} = \beta_{n0} + \beta_{n1}I_{ci} + \beta_{n2}E_{ci} + \frac{\epsilon_{ci}\sigma_n}{\sigma_c}$$

where  $\beta_0$ ,  $\beta_1$  and  $\beta_2$  are the regression coefficients,  $I_i$  is income score,  $E_i$  is employment score,  $\epsilon_i$  is the residual value and  $\sigma$  is the residual S.D.

The resulting country-adjusted IMD is thus predominately based on domains which are comparable between countries and which together account for approximately 95% of within-country variation in IMD score. The 5% of variation which is not explained by these domains is captured by the residuals from each country regression, which are standardized and scaled by the standard deviation of the residuals from the reference country regression. In this way, the contribution of the other five domains is reduced from 50% to roughly 5%.

## Reducing the influence of time-variant domains

The potential transferability of this method to the issue of backwards compatibility is clear, given that variation in the measurement of domain scores is the primary barrier to the use of IMD in both time series and inter-country analysis. The particular exigencies of backwards compatibility are, however, different to that of between-country comparability. And so an adapted version of the above method is required.

In the first instance, the domains which are comparable across countries vary considerably across time. Specifically, the measurement of nearly every indicator used for the income and employment domains was changed in the 2019 IMD to account for the transformation of the benefit system under UC. While these two domains alone have a combined weight of 50% within the overall IMD, it is significant that the weighting and measurement of the other five domains - the remaining 50% - is largely the same across the 2010, 2015 and 2019 indices.

Table 1: By-year comparison of the weighting and underlying indicators of five domains of IMD

	IMD 2010	IMD 2015	IMD 2019
<b>Education</b>	13.5% <sup>a</sup>	13.5%	13.5%
Key Stage 2 attainment	✓	✓	✓
Key Stage 4 attainment	✓	✓	✓
Secondary school absence	✓	✓	✓
Staying on in education post 16	✓	✓	✓
Entry to higher education	✓	✓	✓
Adults with no or low qualifications	✓	✓	✓
English language proficiency	✓	✓	✓
<b>Health and Disability</b> <sup>†</sup>	13.5%	13.5%	13.5%
Years of potential life lost	✓	✓	✓
Comparative illness and disability ratio <sup>b</sup>	✓	✓	✓
Acute morbidity	✓	✓	✓
Mood and anxiety disorders <sup>c</sup>	✓	✓	✓
<b>Crime</b> <sup>d†</sup>	9.3%	9.3%	9.3%
Rate per 1,000 at-risk population:			
• violence	✓	✓	✓
• burglary	✓	✓	✓
• theft	✓	✓	✓
• criminal damage	✓	✓	✓
<b>Barriers to Housing and Services</b>	9.3%	9.3%	9.3%
Road distance to a:			
• post office	✓	✓	✓
• primary school	✓	✓	✓
• general store	✓	✓	✓
• GP surgery	✓	✓	✓
Household overcrowding	✓	✓	✓
Homelessness	✓	✓	✓
Housing affordability	✓	✓	✓
<b>Living Environment</b>	9.3%	9.3%	9.3%
Houses without central heating	✓	✓	✓
Housing in poor condition	✓	✓	✓
Air quality	✓	✓	✓
Road traffic	✓	✓	✓

<sup>a</sup> % represents the weight attributed to each domain within the IMD.

<sup>b</sup> The introduction of Universal Credit (UC) and Personal Independence Payments (PIP) post IMD 2015 does not appear to have meaningfully altered the conditionality of health-related benefit claims for the 2019 IMD given that 1) only a very small number of those eligible for health-related benefits were receiving UC for the data period (March 2016) on which the indicator is based and 2) eligibility conditions for PIP and the Disability Living Allowance benefit it replaced are consistent on core dimensions i.e. non-means tested, non-taxable and payable to people who are in or out of work.

<sup>c</sup> The benefit-based component of this indicator was dropped for IMD 2019 due to concerns around data quality. No changes were made to the other three components.

<sup>d</sup> IMD 2019 uses two years of crime data, while IMD 2015 and IMD 2010 use only one year.

<sup>†</sup> Factor analysis was used to generate the weights assigned to each indicator within this domain.

FALSE

In fact, even after accounting for a few minor differences (see Table 1 footnotes), the weighting and measurement of these five domains across time is more consistent than that of the income and employment domains across countries. *Prima facie*, time-adjusted IMD can thus be expressed as:

$$IMD_{t-ADJUSTEDi} = \beta_{y0} + \beta_{y1}E_{ti} + \beta_{y2}H_{ti} + \beta_{y3}C_{ti} + \beta_{y4}B_{ti} + \beta_{y5}L_{ti} + \frac{\epsilon_{ti}\sigma_y}{\sigma_t}$$

where  $y$  is the reference year,  $E_i$  is education score,  $H_i$  is health score,  $C_i$  is crime score,  $B_i$  is barriers score and  $L_i$  is living environment score.

Yet domain measurement is not only the source of temporal variation. Changes to LSOA boundaries were made in 2011; this means the 2015 and 2019 indices are not geographically compatible with the 2010 IMD. In total, 1,398 of 34,917 pre-2011 England LSOAs were affected by these changes. 163 of these were subject to “complex correction”, for which it is not possible to map to a 2011 equivalent. A further 922 were split into two separate LSOAs, while 313 were merged into a single LSOA. It is possible to map both split and merged LSOAs to 2011 equivalents. However, this results in a “hybrid” LSOA geography, which is not readily

compatible with other UK area-based datasets. Therefore, with a view to creating a time-adjusted IMD with out-of-the-box functionality, I drop the 922 split LSOAs, and and their post-2011 equivalents. But keep the 313 merged LSOAs by aggregating 2010 IMD scores for affected pair-wise areas. Overall, 1085 LSOAs were omitted, which equates to 3.1% of England LSOAs.

# Implementation

## Data preparation and geographical aggregation

The 2010 IMD dataset with domain scores:

```
head(IMD_2010.df, 10)
```

##	LSOA_code_old	IMD	health	educ	barr	crime	livenv
## 1	E01000001	6.16	-2.11	0.21	32.60	-1.64	26.28
## 2	E01000002	5.59	-2.78	0.26	30.26	-1.93	25.73
## 3	E01000003	13.29	-0.97	7.16	40.32	-1.21	36.48
## 4	E01000004	11.17	-1.04	1.76	37.92	-1.32	46.72
## 5	E01000005	21.36	0.07	20.24	34.66	-1.02	40.94
## 6	E01000006	17.08	-0.38	13.84	32.81	0.42	19.39
## 7	E01000007	37.24	0.63	29.22	34.52	1.23	49.82
## 8	E01000008	45.22	0.55	29.85	34.25	0.95	51.89
## 9	E01000009	29.41	0.34	20.75	30.60	1.06	33.99
## 10	E01000010	38.45	0.41	22.72	30.68	1.50	43.65

Trim LSOA strings and merge with LSOA boundary changes lookup table:

```
IMD_2010.df$LSOA_code_old <- stringr::str_trim(IMD_2010.df$LSOA_code_old)
LSOA_bound.df$LSOA_code_old <- stringr::str_trim(LSOA_bound.df$LSOA_code_old)
LSOA_bound.df$LSOA_code <- stringr::str_trim(LSOA_bound.df$LSOA_code)

IMD_2010_bound.df <- dplyr::left_join(IMD_2010.df, LSOA_bound.df,
                                     by = "LSOA_code_old")
```

Aggregate merged LSOAs:

```
attach(IMD_2010_bound.df)
IMD_2010_2011B.df <- IMD_2010_bound.df %>%
  dplyr::group_by(LSOA_code) %>%
  dplyr::mutate(across(c(IMD, health, educ, barr, crime, livenv), ~
    case_when(change == "M" ~ mean(.),
              TRUE ~ .))) %>%
  dplyr::distinct(LSOA_code, .keep_all = TRUE) %>%
  dplyr::ungroup()
detach(IMD_2010_bound.df)
```

## OLS regression

I first fit OLS regressions models of IMD score on IMD domains by year.

2010:

```
lm.2010 <- lm(IMD ~ educ + health + crime + barr + livenv, data = IMD_2010_2011B.df)
paste("$R^{2}$ = ", round(summary(lm.2010)$r.squared,3))
```

[1] " $R^2 = 0.921$ "

2015:

```
lm.2015 <- lm(IMD ~ educ + health + crime + barr + livenv, data = IMD_2015.df)
paste("$R^{2}$ = ", round(summary(lm.2015)$r.squared,3))
```

[1] " $R^2 = 0.928$ "

2019:

```
lm.2019 <- lm(IMD ~ educ + health + crime + barr + livenv, data = IMD_2019.df)
paste("$R^{2}$ = ", round(summary(lm.2019)$r.squared,3))
```

[1] " $R^2 = 0.919$ "

Two things to note here:

1. 2015 has the highest  $R^2$ , and so will be used as the reference (scaling) year.
2.  $R^2$  values fall short of the 95% benchmark set by Abel and colleagues.

## Inspecting residuals

None of the models above are able to explain more than 92.28% of variance in IMD. This amount of unexplained variance is potentially problematic. Recall that the percentage of variation which is not explained by the model is represented by the residual, which is equivalent to the influence of the time-varying income and employment domains. I thus plot residuals by domain score so as to assess model fit and whether variable transformation is required.

Residual plot function with LOESS smoother:

```

# Define function
residplot.fun = function(xF, yF) {

  # Get loess smmothed predictions
  f <- paste("residual ~", xF)
  domcount <- domcount + 1
  loess <- loess(f, data = IMD_2015.df, span = 0.25)
  loess_fit <- as.data.frame(predict(loess))
  names(loess_fit) <- c("fit")
  IMD_2015.df$fit <- loess_fit$fit
  IMD_2015.df <- IMD_2015.df %>%
    dplyr::arrange(xF)

  # Residual plot
  resdom.gg <-
    ggplot(IMD_2015.df,
            aes(x = .data[[xF]], y = .data[[yF]]) ) +
    geom_hline(yintercept=0, size= 0.25,
               color="black", linetype = "dashed") +
    geom_point(shape = 1, size = .2, color = "#868686",
               alpha = .4, position = position_jitter(w = 0.2)) +
    geom_line(IMD_2015.df, mapping =
              aes(x = .data[[xF]], y = fit), col = "#EFC000",size= 1) +
    labs(x = domnames.vec[domcount], y = "Residual") +
    theme_bw() +
    theme(
      plot.margin = unit(c(.1, .1, .1, .1), "cm"))

  # Axis options depending on plot position
  if (xF == "educ" || xF == "barr") {
    resdom.gg <- resdom.gg + theme(axis.title.y=element_text(vjust = -0.05))
    return(resdom.gg)
  }

  else {
    resdom.gg <- resdom.gg + theme(axis.title.y=element_blank(),
                                   axis.text.y = element_blank(),
                                   axis.ticks.y = element_blank())
    return(resdom.gg)
  }
}

# Apply function to each IMD domain
domcount <- 0
domresplots <- lapply(doms, FUN = residplot.fun, y= "residual")

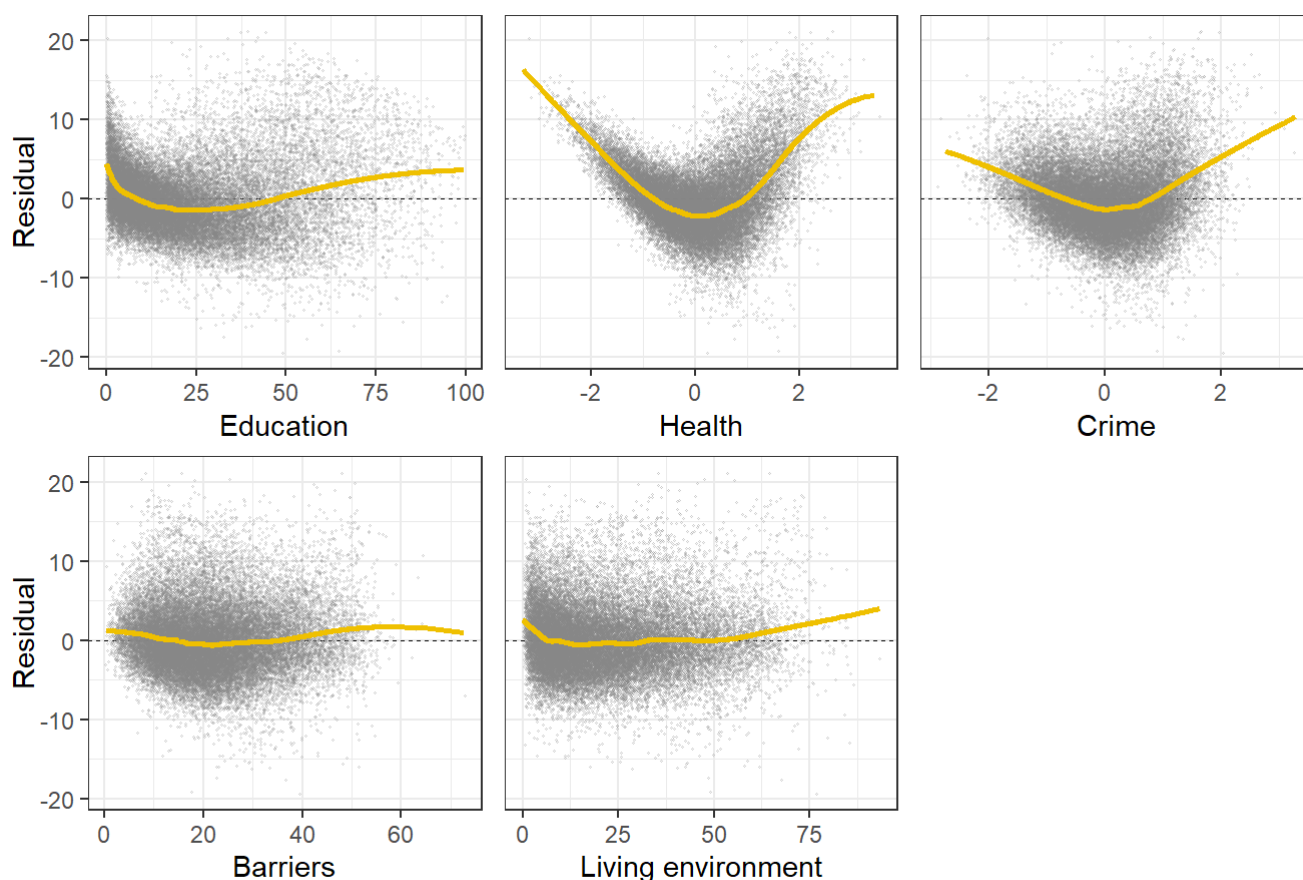
```

Final combined plot:

```
# Combine residual plots
resdomC.gg <- cowplot::plot_grid(domresplots[[1]], domresplots[[2]],
                                domresplots[[3]], domresplots[[4]],
                                domresplots[[5]],nrow=2,
                                rel_widths = c(1.115, .94,
                                                .94, 1.15, .94)) +
  theme(plot.margin = margin(2,2, 8.5,1,"mm"))

# Add plot note
resdomC.gg <- resdomC.gg +
  draw_label(P1_line_1, x = .485, y = -0.02, size = 7.5) +
  draw_label(P1_line_2, x = .1625, y = -0.05, size = 7.5)

resdomC.gg
```



Plot of residuals by domain from OLS regression of 2015 IMD score on 2015 IMD domain scores. Solid yellow lines represent locally smoothed fit from LOESS regression using a bandwidth of 0.25.

### Residual plot by domain score

From the above plot we can see that the residuals are generally not well behaved. The health domain exhibits a near-perfect parabolic curve, while a less severe parabola can be seen in the crime domain. Even the education and living environment domains exhibit some semblance of non-uniformity, with systemic under-prediction occurring at the far end of both left and right tails.

## MFP

I employ the Multivariable Fractional Polynomial procedure to determine the best-fitting functional form for each domain score. A generalization of polynomials, fractional polynomials provide a more flexible class of transformation functions. Although it is true that the inclusion of a simple square polynomial term would lead to a significant reduction in nonlinearity, I prefer the added power of MFP given the paramount importance of model fit in minimizing the influence of time-varying domains.

Fit MFP model with 4 degrees of freedom (FP2-max):

```
mfp2015 <- mfp(IMD ~ fp(educ) + fp(health) + fp(crime) + fp(barr) +
               fp(livenv), family = gaussian, data = IMD_2015.df, verbose = FALSE)
```

Best fitting forms:

```
mfp2015$powers
```

```
##      power1 power2
## educ      1      1
## health    1      2
## crime     1      2
## barr      3      3
## livenv    2      2
```

Coefficients

```
mfp2015$coefficients
```

```
##      Intercept      educ.1      educ.2      health.1      health.2      crime.1
##  7.0713011    2.5423613    0.4072984 -77.9982958  226.6046843  -2.6348268
##      crime.2      barr.1      barr.2      livenv.1      livenv.2
##  1.1142323    0.3385441  -0.1510216    0.4942839  -0.1831674
```

## OLS versus MFP

Matrix to store coefficients:

```
domreg.mat <- matrix(nrow = 16, ncol = 6,
                     dimnames = list(c(rep(c("Linear", "FP1", "FP2"),5) ,"$R^{2}$"),
                                     c(rep(c("(OLS)", "(MFP)"),3))))
```

Get coefficients:



```

# Set increments
mcount <- 0
for (m in c("", "mfp")) {
  mcount <- mcount + 1
  if (mcount == 1) {
    ycount <- 1
  }
  else {
    ycount <- 2
  }
  for (y in c(2010, 2015, 2019)) {
    print(y)
    pcount <- 1

    # OLS Loop
    if (mcount == 1) {
      for (i in seq(from = 1, to = 13, by = 3)) {
        print(i)
        pcount <- pcount + 1
        domreg.mat[i, ycount] <- eval(parse(text=paste0("lm.", y, ".bcoef[[1]][[pcount]]")))
      }
      domreg.mat[16, ycount] <- eval(parse(text=paste0("summary(lm.", y, ")$r.squared")))
      ycount <- ycount + 2
    }

    # MFP Loop
    else {
      for (i in c(2, 3, 5, 6, 8, 9, 11, 12, 14, 15)) {

        pcount <- pcount + 1

        domreg.mat[i, ycount] <- eval(parse(text=paste0("lm.mfp.", y, ".bcoef[[1]][[pcount]]"
)))
      }

      domreg.mat[16, ycount] <- eval(parse(text=paste0("summary(lm.mfp.", y, ")$r.squared")))
      ycount <- ycount + 2
    }
  }
}

# Round values and replace recode NAs as blank
domreg.mat <- round(domreg.mat, 3)
domreg.mat[is.na(domreg.mat)] <- ""

```

Create table:

```

# Table structure/style
domreg.tab <- kbl(domreg.mat, align = c(rep("c", 5))) %>%
  add_header_above(c(" " = 1, "2010" = 2, "2015" = 2, "2019" = 2),
    line = T, line_sep = 7.5
    #caption = "Table 2: Table of coefficients for OLS versus MFP regression models by year"
  ) %>%
  kable_classic(full_width = F,html_font = "Cambria", font_size = 11)%>%

# Group rows by domains
pack_rows("Education", 1, 3, bold= T, italic = T) %>%
pack_rows("Health", 4, 6, bold = T, italic = T) %>%
pack_rows("Crime", 7, 9, bold = T, italic = T) %>%
pack_rows("Barriers", 10, 12, bold = T, italic = T) %>%
pack_rows("Environment",13, 15, bold = T, italic = T) %>%

# Rule lines to separate domains
row_spec(1, extra_css = "border-top: .7px solid; border-color: grey") %>%
row_spec(4, extra_css = "border-top: .7px solid; border-color: grey") %>%
row_spec(7, extra_css = "border-top: .7px solid; border-color: grey") %>%
row_spec(10, extra_css = "border-top: .7px solid; border-color: grey") %>%
row_spec(13, extra_css = "border-top: .7px solid; border-color: grey") %>%
row_spec(16, extra_css = "border-top: .7px solid; border-color: grey") %>%

# Table note
footnote(general = "Standardized beta coefficients.",
  general_title = "Note.",
  footnote_as_chunk = T, title_format = c("italic"))

domreg.tab

```

	2010		2015		2019	
	(OLS)	(MFP)	(OLS)	(MFP)	(OLS)	(MFP)
<b>Education</b>						
Linear	0.366		0.388		0.392	
FP1		3.22		2.549		2.765
FP2		0.009		0.399		0.293
<b>Health</b>						
Linear	7.565		6.955		7.053	
FP1		-96.067		-73.206		-85.631
FP2		266.073		226.344		247.067
<b>Crime</b>						
Linear	2.671		3.358		3.354	
FP1		-18.915		-42.444		-30.45
FP2		67.334		111.976		91.35
<b>Barriers</b>						
Linear	0.236		0.261		0.244	
FP1		0.576		0.643		0.73
FP2		-0.023		-0.029		-0.05
<b>Environment</b>						
Linear	0.157		0.122		0.135	
FP1		-2.431		-1.52		-2.261
FP2		2.476		1.752		2.192
<i>R</i> <sup>2</sup>	0.921	0.956	0.928	0.962	0.919	0.954

Note. Standardized beta coefficients.

Table 2 compares coefficients and model performance for OLS versus MFP regressions. Across all years, the MFP model provides a substantially better fit than OLS. For IMD 2015, the  $R^2$  value jumps from 0.928 to 0.962, which amounts to a 50% reduction in the influence of the time-varying income and employment domains from 7.72% to 3.8%. This meets and exceeds the 5% benchmark set by Abel and colleagues. As expected, it is the transformation of the health and crime domains which is the major stimulus of this performance increase. Beta coefficients for these domains are drastically different for MFP as compared to OLS models; the 1<sup>st</sup>-degree FP turns negative (capturing the downwards slope on the left-hand side of the parabola), while the 2<sup>nd</sup>-degree FP is positive (capturing the the right-hand side of the curve). Coefficients are highly consistent across years. This is significant because it implies that model residuals are correlated, and that the regression scaling will, *prima facie*, produce reliable results.

## The adjustment

I create time-adjusted IMD score by scaling the 2010 and 2019 domain scores and MFP model residuals by the coefficient values residual SD for the 2019 MFP model.

2010 adjustment:

```
IMD_2010_2011B.df$IMD_tadj <-
  #| 2015 Constant
  (coef_2015.mat[1] +
  #| Plus 2010 domain values scaled by 2015 MFP coefficients
  (coef_2015.mat[2]*IMD_2010_2011B.df$educ_FP1) + (coef_2015.mat[3]*IMD_2010_2011B.df$educ_F
P2) +
  (coef_2015.mat[4]*IMD_2010_2011B.df$health_FP1) + (coef_2015.mat[5]*IMD_2010_2011B.df$healt
h_FP2) +
  (coef_2015.mat[6]*IMD_2010_2011B.df$crime_FP1) + (coef_2015.mat[7]*IMD_2010_2011B.df$crime_
FP2) +
  (coef_2015.mat[8]*IMD_2010_2011B.df$barr_FP1) + (coef_2015.mat[9]*IMD_2010_2011B.df$barr_FP
2) +
  (coef_2015.mat[10]*IMD_2010_2011B.df$livenv_FP1) + (coef_2015.mat[11]*IMD_2010_2011B.df$liv
env_FP2) +
  #| Plus 2010 standardized residual scaled by by 2015 residual SD
  ((IMD_2010_2011B.df$MFPres*MFPresSD2015)/MFPresSD2010))
```

2019 adjustment:

```
IMD_2019.df$IMD_tadj <-
  #| 2015 Constant
  (coef_2015.mat[1] +
  #| Plus 2019 values scaled by 2015 MFP coefficients
  (coef_2015.mat[2]*IMD_2019.df$educ_FP1) + (coef_2015.mat[3]*IMD_2019.df$educ_FP2) +
  (coef_2015.mat[4]*IMD_2019.df$health_FP1) + (coef_2015.mat[5]*IMD_2019.df$health_FP2) +
  (coef_2015.mat[6]*IMD_2019.df$crime_FP1) + (coef_2015.mat[7]*IMD_2019.df$crime_FP2) +
  (coef_2015.mat[8]*IMD_2019.df$barr_FP1) + (coef_2015.mat[9]*IMD_2019.df$barr_FP2)+
  (coef_2015.mat[10]*IMD_2019.df$livenv_FP1) + (coef_2015.mat[11]*IMD_2019.df$livenv_FP2) +
  #| Plus 2019 standardized residual scaled by 2015 residual SD
  ((IMD_2019.df$MFPres*MFPresSD2015)/MFPresSD2019))
```

I then expand the 2010 and 2015 dataset to include observations for each year between 2010 and 2019:

```
# Expand function (mimics expand function in Stata)
expand <- function(df, ...) {
  as.data.frame(lapply(df, rep, ...))
}
# Expand up to (and including) 2018
IMD_2010_14.df <- expand(IMD_2010_2011B.df, times = 5)
IMD_2015.df$IMD_tadj <- IMD_2015.df$IMD
IMD_2015_18.df <- expand(IMD_2015.df, times = 4)
```

Combine the resulting datasets:

```
# Append
IMD_TS.df <- dplyr::bind_rows(IMD_2010_14.df, IMD_2015_18.df) %>%
  dplyr::group_by(LSOA_code) %>%
  dplyr::mutate(obno = row_number()) %>%
  dplyr::ungroup()
# Map year values
IMD_TS.df$year = plyr::mapvalues(IMD_TS.df$obno, from = c(1:9),
                                to = c(2010:2018))
IMD_TS.df$year <- as.numeric(as.character(IMD_TS.df$year))
```

And generate estimated IMD and time-adjusted IMD scores for missing years via linear interpolation:

```
# Set IMD score as missing for added years
IMD_TS.df <- IMD_TS.df %>%
  dplyr::mutate(IMD = case_when(year == 2010 | year == 2015 ~ as.numeric(IMD),
                                TRUE ~ as.numeric(NA))) %>%
  dplyr::mutate(IMD_tadj = case_when(year == 2010 | year == 2015 ~ as.numeric(IMD_tadj),
                                    TRUE ~ as.numeric(NA)))
# Add 2019 values
IMD_TS.df <- dplyr::bind_rows(IMD_TS.df, IMD_2019.df)
# Linear interpolation
IMD_TS.df <- IMD_TS.df %>%
  dplyr::group_by(LSOA_code) %>%
  dplyr::mutate(IMD_ipol = na.approx(IMD, na.rm=FALSE)) %>%
  dplyr::mutate(IMD_tadj_ipol = na.approx(IMD_tadj, na.rm=FALSE)) %>%
  dplyr::ungroup() %>%
  # Decile rank of interpolated scores (by year)
  dplyr::group_by(year) %>%
  mutate(IMD_dec_ipol = ntile(IMD_ipol, 10)) %>%
  mutate(IMD_tadj_dec_ipol = ntile(IMD_tadj_ipol, 10)) %>%
  arrange(LSOA_code, year) %>%
  dplyr::ungroup()
head(select(IMD_TS.df, LSOA_code, year, IMD_ipol, IMD_tadj_ipol), 10)
```

```
## # A tibble: 10 x 4
##   LSOA_code year IMD_ipol IMD_tadj_ipol
##   <chr>     <dbl>   <dbl>     <dbl>
## 1 E01000001 2010     6.16     6.13
## 2 E01000001 2011     6.14     6.12
## 3 E01000001 2012     6.12     6.10
## 4 E01000001 2013     6.10     6.09
## 5 E01000001 2014     6.08     6.08
## 6 E01000001 2015     6.06     6.06
## 7 E01000001 2016     6.10     6.08
## 8 E01000001 2017     6.14     6.10
## 9 E01000001 2018     6.17     6.12
## 10 E01000001 2019     6.21     6.14
```

## Descriptive statistics (*t*-adjusted IMD versus IMD)

### Mean difference plot

Having created a time-adjusted IMD score, I now turn to descriptive statistics to assess the extent to which this measure differs from the original IMD.

I first use Turkey-style mean difference plots to see the average difference between scores, as well as to compare distributions/variability.

I generate required variables and store distribution quantities:

```
# Mean and mean difference vars
IMD_2010_2011B.df$IMD_dif <- IMD_2010_2011B.df$IMD_tadj - IMD_2010_2011B.df$IMD
IMD_2010_2011B.df$IMD_mean <- (IMD_2010_2011B.df$IMD + IMD_2010_2011B.df$IMD_tadj)/2
# Mean + SD values
meanline <- round(mean(IMD_2010_2011B.df$IMD_dif), digits=2)
difSD <- sd(IMD_2010_2011B.df$IMD_dif)
lbciline <- round(meanline - (1.96*difSD), digits=2)
ubciline <- round(meanline + (1.96*difSD), digits=2)
```

Plot mean differences for 2010 and 2019 scores (I include 2010 only for brevity):

```

meandif2010.gg <- ggplot(IMD_2010_2011B.df, aes(x=IMD_mean, y=IMD_dif)) +
  geom_point(color="#868686FF", size=.6, shape=1, alpha=0.4,
            position = position_jitter(w = 0.5, h = 0.75)) +
  geom_hline(yintercept=meanline, size=0.8, color="#efc000") +
  xlim(0, 100) +
  geom_hline(yintercept = c(lbciline, ubciline),
            linetype="solid", color="#3b3b3b",
            size=0.6) +
  annotate("text", x = 96.5, y = 0.41,
          label = paste0("Mean = ", meanline),
          size=2.75, fontface="bold") +
  annotate("text", x = 93.5, y = -2.075,
          label = paste0("-1.96 S.D. = ", lbciline), size=2.75, fontface="bold") +
  annotate("text", x = 94.25, y = 2.23,
          label = paste0("+1.96 S.D. = ", ubciline),
          size=2.75, fontface="bold") +
  coord_cartesian(xlim = c(0, 100), ylim = c(-5, 5)) +
  theme_bw()+
  theme(axis.title.y=element_blank(),
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        panel.background = element_rect(colour = "black",
                                          size=.5, fill=NA)) +
  theme(plot.margin = unit(c(.25, .25, 0, .575), "cm"))

```

And then combine 2019 and 2019 plots:

```

# Combine plots
meandifC.gg <- cowplot::plot_grid(meandif2010.gg, meandif2019.gg, ncol=1,
                                rel_heights = c(.875, 1), labels = c(2010, 2019),
                                label_x = c(.065, 0.065), label_y = c(.92, .97),
                                label_size = 16, label_fontfamily = "serif")

# Add spanning y-axis title
meandifC.gg <- cowplot::ggdraw(meandifC.gg) +
  annotate(geom = "richtext", x = 0.02, y=0.525,
          label = "<i>t</i>-adjusted IMD score minus IMD score",
          angle = 90, size = 3.88,
          fill = NA, label.color = NA) +

# Add markdown x-axis title
annotate(geom = "richtext", x = .695, y=.0203,
          label = "<i>t</i>-adjusted IMD score",
          size = 3.88,
          fill = NA, label.color = NA) +

# Add plot note
theme(plot.margin = margin(2,2, 14,1, "mm")) +
draw_label(P2_line_1, x = .5025, y = -0.0225, size = 7.5) +
draw_label(P2_line_2, x = .472, y = -0.0465, size = 7.5) +
draw_label(P2_line_3, x = .466, y = -0.0725, size = 7.5)
meandifC.gg

```

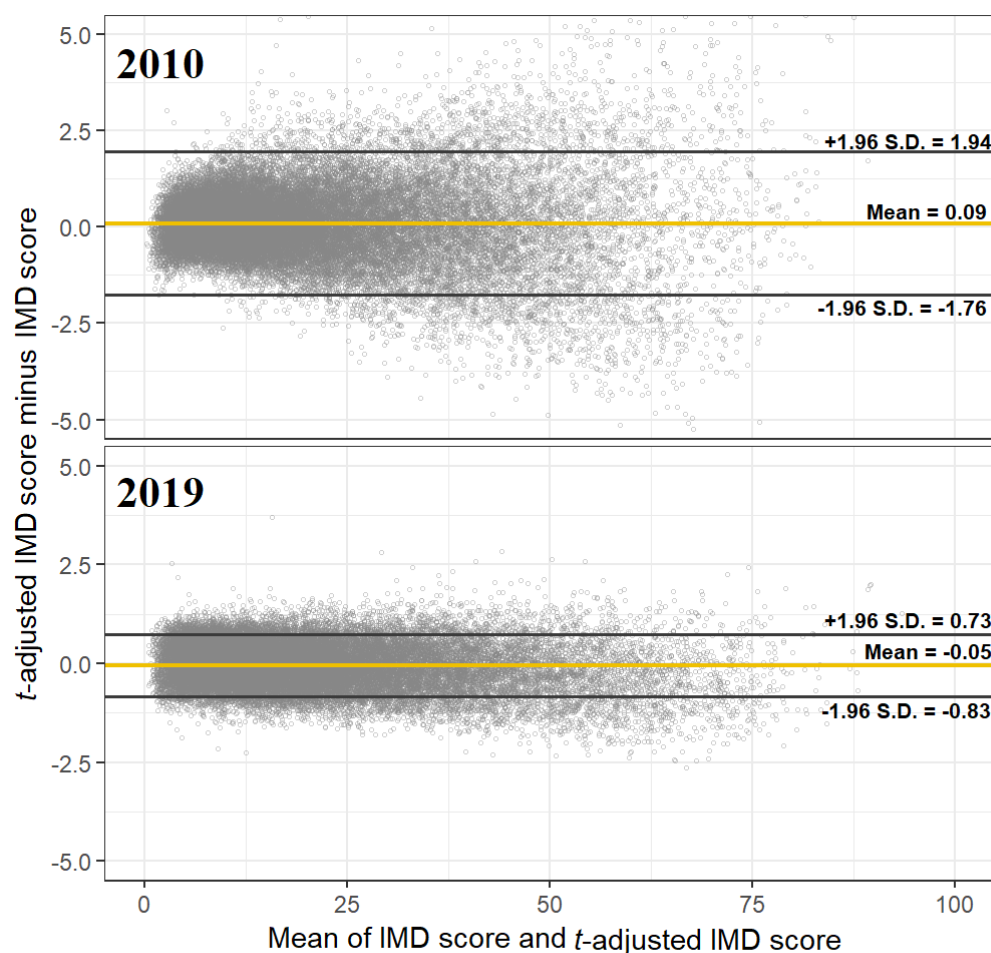


Figure 2 shows that the

Note. Mean difference plot of 2010/2019 IMD score versus time-adjusted 2010/2019 IMD score. X-axis represents mean score. Y-axis represent difference between scores. A solid yellow line represents the mean difference across all LSOAs, while solid black lines represent upper and lower distribution limits at the 95<sup>th</sup> percentile.

mean difference between IMD and *t*-adjusted IMD score is similar across years; strong evidence of a significant correlation between models. This difference is, admittedly, not especially large. Yet there is no reason to expect it to be so. Variation in time-varying domains, while significant enough to warrant adjustment, is not, by any standard, drastic. A low average difference in IMD scores is, in this sense, a further indicator of the validity of the adjusted index. Nevertheless, it is notable that variability is much higher for 2010 scores as compared to 2019. Of course, the implication of this is that the 2010 IMD is substantively different to the 2015 IMD in a way that the 2019 IMD is not. However, it is difficult to reconcile this notion with the fact that differences in domain measurement are actually more pronounced in the 2019 IMD. The most likely explanation is that there is a greater quantitative and/or qualitative difference in the factor weighting of one more domains for 2010 IMD as compared to 2019 IMD. Unfortunately, it is not possible adjust for these differences, since the raw sub-domain scores are not available to the public. Yet given that everything else suggests the adjusted index is broadly robust to temporal differences, additional adjustment on these terms would not likely alter the final index in any meaningful way.

## Change IMD rank versus *t*-adjusted rank

I now create simple bar charts to obtain a better idea of the *substantive* difference between IMD and *t*-adjusted IMD. For both 2010 and 2019, I plot the change in decile rank for IMD versus *t*-adjusted IMD relative to 2015 IMD.

I first generate the required variables/quantities:

```

### 5 year lag/lead
IMD_all.df <- IMD_all.df %>%
  group_by(LSOA_code) %>%
  dplyr::mutate(IMD_dec_LAG = lag(IMD_dec, n = 1)) %>%
  dplyr::mutate(IMD_tadj_dec_LAG = lag(IMD_tadj_dec, n = 1)) %>%
  dplyr::mutate(IMD_dec_LEAD = lead(IMD_dec, n = 1)) %>%
  dplyr::mutate(IMD_tadj_dec_LEAD = lead(IMD_tadj_dec, n = 1)) %>%
  dplyr::ungroup()

### Drop redundant 2010/2019 obs
IMD_dif.df <- IMD_all.df[!(IMD_all.df$year != "2015"),]
attach(IMD_dif.df)

### Difference in decile rank (2019 minus 2015/2010)
IMD_dif.df$IMD_dec_LAG_D <- IMD_dec - IMD_dec_LAG
IMD_dif.df$IMD_tadj_dec_LAG_D <- IMD_tadj_dec - IMD_tadj_dec_LAG
IMD_dif.df$IMD_dec_LEAD_D <- IMD_dec - IMD_dec_LEAD
IMD_dif.df$IMD_tadj_dec_LEAD_D <- IMD_tadj_dec - IMD_tadj_dec_LEAD
attach(IMD_dif.df)

### Difference in difference (t-adj IMD minus IMD)
IMD_dif.df$IMD_dec_LAG_DinD <- IMD_tadj_dec_LAG_D - IMD_dec_LAG_D
IMD_dif.df$IMD_dec_LEAD_DinD <- IMD_tadj_dec_LEAD_D - IMD_dec_LEAD_D
detach(IMD_dif.df)

### Tag obs. by difference
IMD_dif.df <- IMD_dif.df %>%
  dplyr::mutate(change_LAG = case_when(IMD_dec_LAG_DinD < 0 ~ 2,
                                       IMD_dec_LAG_DinD > 0 ~ 1,
                                       IMD_dec_LAG_DinD == 0 ~ 3)) %>%
  dplyr::mutate(change_LEAD = case_when(IMD_dec_LEAD_DinD < 0 ~ 2,
                                       IMD_dec_LEAD_DinD > 0 ~ 1,
                                       IMD_dec_LEAD_DinD == 0 ~ 3))

### Sum of tagged obs. by decile rank/tag
IMD_dif.df <- IMD_dif.df %>%
  dplyr::group_by(IMD_dec, change_LAG) %>%
  dplyr::add_count(change_LAG, name = "change_LAG_cnt") %>%
  dplyr::ungroup() %>%
  dplyr::group_by(IMD_dec, change_LEAD) %>%
  dplyr::add_count(change_LEAD, name = "change_LEAD_cnt") %>%
  dplyr::ungroup()

### Total no. of obs by decile rank
IMD_dif.df <- IMD_dif.df %>%
  dplyr::group_by(IMD_dec) %>%
  dplyr::mutate(dec_cnt = n()) %>%
  dplyr::ungroup()
detach(IMD_dif.df)

### Difference in difference as proportion of decile
IMD_dif.df$DinD_LAG_perc <- (IMD_dif.df$change_LAG_cnt / IMD_dif.df$dec_cnt)*100
IMD_dif.df$DinD_LEAD_perc <- (IMD_dif.df$change_LEAD_cnt / IMD_dif.df$dec_cnt)*100
IMD_dif.df <- subset(IMD_dif.df, !is.na(year))

```

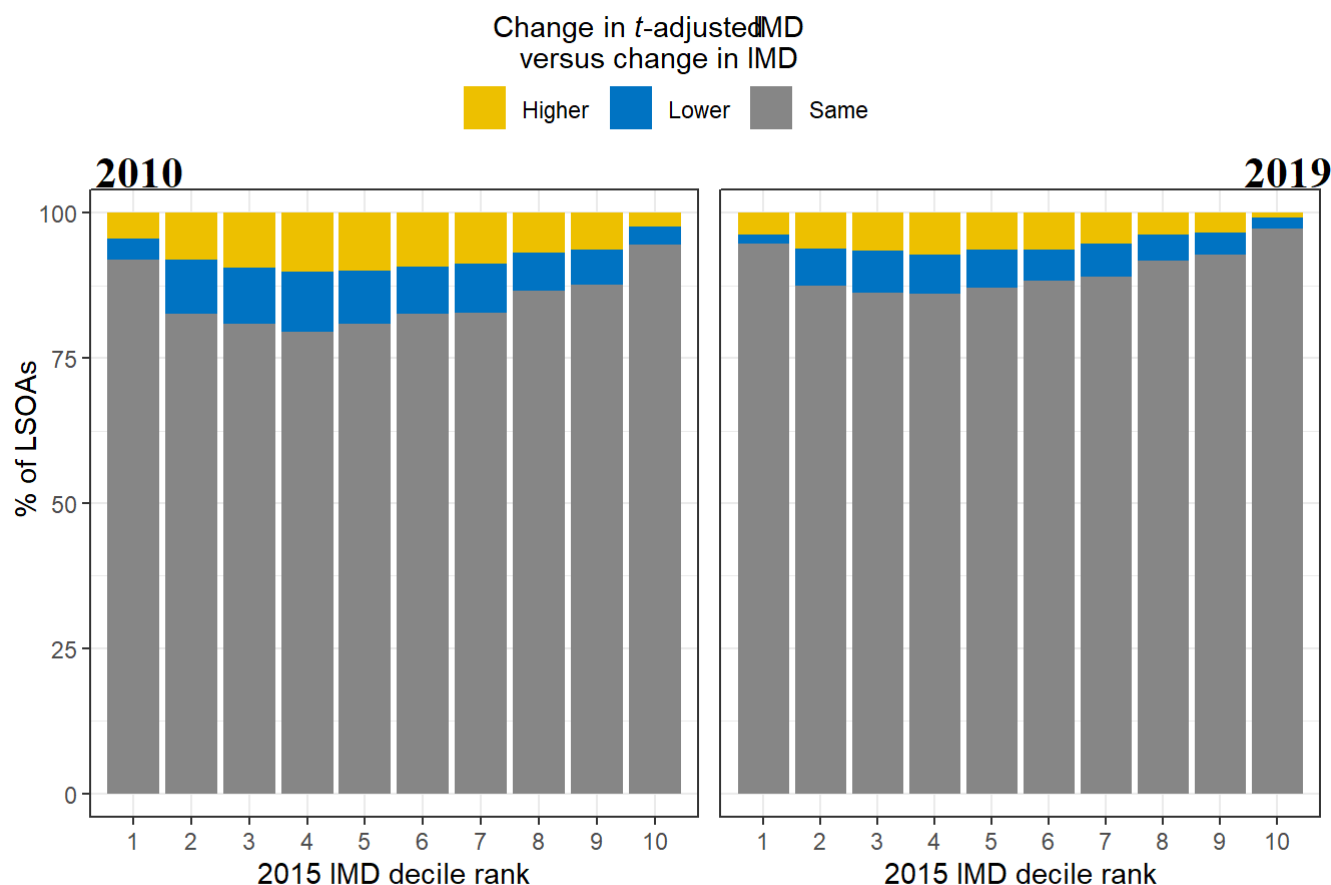
Create bar chart for 2010 and 2019 scores (I include 2010 only for brevity):



```
# Plot
dind2010.gg <- ggplot(data=IMD_dif.df, mapping = aes(
  x = factor(IMD_dec), y=DinD_LAG_perc, fill=factor(change_LAG), group=factor(change_LAG))) +
  geom_bar( stat="unique") +
  scale_fill_manual(labels = c("Higher", "Lower", "Same"),
    values = c("1" = "#EDC000FF",
               "2" = "#0073c2FF",
               "3" = "#868686FF")) +
  scale_y_continuous(expand = c(.04,.04)) +
  scale_x_discrete(expand = c(.075,.075)) +
  xlab("2015 IMD decile rank") +
  guides(fill = guide_legend(title =
    "Change in <i>t</i>-adjusted <br>",
    title.position = "top")) +
  coord_cartesian(xlim = c(1, 10), ylim = c(0, 100), clip = "off")+
  theme_bw() +
  theme(legend.position="top", legend.title.align=0.05,
    legend.title=element_markdown(),
    plot.tag.position = c(.5, .93),
    plot.tag=element_text(size=11),
    axis.title.y = element_blank(),
    plot.margin = unit(c(.575, .1, .1, .575), "cm"))
```

And then combine 2010 and 2019 plots:

```
# Combine plots
dindC.gg <- cowplot::plot_grid(
  dind2010.gg + theme(legend.position = "none"),
  dind2019.gg + theme(legend.position = "none"),
  ncol=2,rel_widths = c(1, .89), labels = c(2010, 2019),
  label_x = c(0.08, 0.78), label_y = c(1.005, 1.005),
  label_size = 16, label_fontfamily = "serif")
# Add Legend
legend <- get_legend(
  dind2010.gg + theme(legend.box.margin=margin(11.5,0,4.25,0),
    legend.position = "top",
    legend.title.align = 0.25))
# Add Labels and axis titles
dindC.gg <- plot_grid(legend,dindC.gg, nrow =2, rel_heights = c(.175, .8)) +
  annotate(geom="text", label = "IMD",
    x = .584, y = .952, size = 3.88) +
  annotate(geom = "text", label = "versus change in IMD",
    x = 0.4945, y=.9175,
    size = 3.88) +
  annotate(geom = "text", x = 0.0225, y=0.5,
    label = "% of LSOAs",
    angle = 90, size = 3.88) +
  theme(panel.border = element_blank(),
    plot.margin = unit(c(.075, .075, 6, .075), "mm")) +
  draw_label(P3_line_1, x = .4655, y = -0.028, size = 7.5)
dindC.gg
```



Note. Bar chart of the change in IMD rank versus time-adjusted IMD rank between a) 2015 and 2010 and b) 2015 and 2019.

Figure 3 shows that there is no difference between IMD rank and time-adjusted IMD ranks for a large majority of LSOAs. In line with Figure 2, there is substantially more movement in 2010 scores compared to 2019. What is of particular interest here, however, is that the adjusted index has nearly zero substantive impact on the least and most deprived LSOAs. And that, conversely, the effect of differences in IMD domain measurement over time is strongest for areas which fall within the middle of the deprivation scale. Together, these findings appear to distinguish three broad strata of relative deprivation levels in England, with two distant poles separating a large and somewhat homogeneous majority. In this model, differences between deprivation levels are relatively small for the majority of areas, and so even minor changes to the measurement of IMD can result in a substantial change in IMD rank. In contrast, differences between deprivation levels at each pole vis-a-vis the majority are comparatively larger, and so this type of small adjustment is highly unlikely to lead to a meaningful difference in interpretation. This type of explanation is, of course, speculative, But it does broadly align with the social class structure described by Great British Class Survey and other studies of social structure in England.

## An illustrative analysis of mortality rates

I now present a short and strictly illustrative analysis of mortality rates using time-adjusted IMD ranks.

### RE and REWB models

### REWB with interactions