UPC & IDBR Business Premises Exploratory Analysis

Fushu Beauthier

15 August 2023

This is the step-by-step methodology, with R code, for the data cleaning and pre-processing for the Point-Topic UPC v. IDBR UK Business Counts exporatory analysis.

The downloaded IDBR .csv's had to be converted to .xlsx because files wouldn't separate columns correctly when read in R.

N.B. For each file, the directory/file path and file names in the code below need to be modified according to where you have stored each file and how you saved it when you downloaded it.

Steps:

1. Load the UPC estimated business premises
2. Load the post-code to MSOA lookup (Census 2011 boundaries)
3. Join MSOA lookup onto UPC table
4. Group UPC business premises by MSOA
5. Load the IDBR Business Premises
6. Load the IDBR Local Units
7. Join all IDBR and UPC in a new table
8. Analysis & summary statistics

Loading the necessary packages

```
library(ggpubr)
library(tidyverse)
library(qacEDA)
```

1. Load the UPC estimated business premises 2019, pulled from Snowflake table

```
# business sites obtained at post-code level straight from UPC table in Snowflake
upc <- read_csv("data/demos/UPC_busprems/UPC_bussites_2019.csv") #
head(upc)
```

```
## # A tibble: 6 x 2
##   POSTCODE BUS_SITES_TOTAL
##   <chr>              <dbl>
## 1 KA1 5DG                0
## 2 M43 7FW                0
## 3 NE23 6BA               0
## 4 G53 7YW                0
## 5 DH8 7AT                0
## 6 BS22 7TQ               0
```

2. Load the post-code to MSOA lookup (2011 MSOAs)

```r
# lookup obtained from ONS Open Geography Portal
lookup <- read_csv("data/Lookups/NSPCL_2011_UK_LU.csv")
head(lookup)
```

```
## # A tibble: 6 x 25
##   pcd7  pcd8  pcds  dointr doterm usert~1 oseas~2 osnrt~3 oa11cd oac11cd oac11nm
##   <chr> <chr> <chr>  <dbl>  <dbl>   <dbl>   <dbl> <chr>   <chr>  <chr>   <chr>
## 1 AB1 ~ AB1 ~ AB1 ~ 198001 199606       0  385386 0801193 S0009~ 1C3     Detach~
## 2 AB1 ~ AB1 ~ AB1 ~ 198001 199606       0  385177 0801314 S0009~ 1C3     Detach~
## 3 AB1 ~ AB1 ~ AB1 ~ 198001 199606       0  385053 0801092 S0009~ 6A1     Indian~
## 4 AB1 ~ AB1 ~ AB1 ~ 199402 199606       0  384600 0799300 S0009~ 1A2     Establ~
## 5 AB1 ~ AB1 ~ AB1 ~ 199012 199207       1  384460 0800660 S0009~ 6A4     Ageing~
## 6 AB1 ~ AB1 ~ AB1 ~ 199012 199207       1  383890 0800710 S0009~ 7C3     Outer ~
## # ... with 14 more variables: wz11cd <chr>, wzc11cd <chr>, wzc11nm <chr>,
## #   lsoa11cd <chr>, lsoa11nm <chr>, msoa11cd <chr>, msoa11nm <chr>,
## #   soac11cd <chr>, soac11nm <chr>, ladcd <chr>, ladnm <chr>, ladnmw <chr>,
## #   laccd <chr>, lacnm <chr>, and abbreviated variable names 1: usertype,
## #   2: oseast1m, 3: osnrth1m
```

```r
length(unique(lookup$msoa11cd)) # view number of MSOAs
```

```
## [1] 8484
```

```r
# select only post-code and msoa code columns
lookup_msoa <- dplyr::select(lookup,
                             pcds, msoa11cd)
```

3. Join MSOA lookup onto UPC table

```r
# use left-join
upc <- left_join(upc, lookup_msoa, by = c("POSTCODE" = "pcds"))
head(upc)
```

```
## # A tibble: 6 x 3
##   POSTCODE BUS_SITES_TOTAL msoa11cd
##   <chr>              <dbl> <chr>
## 1 KA1 5DG                0 S02001492
## 2 M43 7FW                0 E02001239
## 3 NE23 6BA               0 E02005718
## 4 G53 7YW                0 S02001844
## 5 DH8 7AT                0 E02004304
## 6 BS22 7TQ               0 E02003079
```

```r
# check NAs
colSums(is.na(upc))
```

```
##        POSTCODE BUS_SITES_TOTAL        msoa11cd
##               0             856              698
```

```r
# 856 business sites missing at post-code level, i.e. 856 post-codes where business sites not counted

# we choose to omit these NAs, otherwise when aggregating entire MSOAs will become NA
upc <- na.omit(upc)

# view unique number of MSOAs left
length(unique(upc$msoa11cd))
```

```
## [1] 8478
```

4. We now group UPC business premises by MSOA in a new table

```
# select only MSOAs and business sites columns, group by MSOA code
upc_MSOA <- upc %>%
  dplyr::select(2,3) %>%  # column index
  group_by(msoa11cd) %>%
  summarise(upc_bus_prems = sum(BUS_SITES_TOTAL))

head(upc_MSOA)
```

```
## # A tibble: 6 x 2
##   msoa11cd  upc_bus_prems
##   <chr>             <dbl>
## 1 E02000001       16893.
## 2 E02000002          91.8
## 3 E02000003         310.
## 4 E02000004          43.0
## 5 E02000005         120.
## 6 E02000007          97.2
```

```
# finally, round to whole numbers
upc_MSOA$upc_bus_prems <- round(upc_MSOA$upc_bus_prems, digits = 0)
```

5. Load the IDBR Business Premises for cleaning, last updated 2019 (as of 2023)

```
idbr_ent <- readxl::read_xlsx("data/demos/UPC_busprems/IDBR_enterprises_2019.xlsx")
# check data classes using: str(idbr_ent)

# convert count columns to numeric
idbr_ent <- mutate_at(idbr_ent, c(3:20), as.numeric)
# select only the first table in the .xlsx file
# (remember .xlsx file contains many tables one below the other)
idbr_ent <- idbr_ent[9:7209,]

# missing business sites Total column, so we create Totals per MSOA column
idbr_ent$idbr_sum_ents <- rowSums(idbr_ent[,3:20])

# rename MSOA code column and only select relevant columns
idbr_ent <- idbr_ent %>%
  rename(MSOA = 2) %>%
  dplyr::select(MSOA, idbr_sum_ents)

head(idbr_ent)
```

```
## # A tibble: 6 x 2
##   MSOA      idbr_sum_ents
##   <chr>             <dbl>
## 1 E02002559           540
## 2 E02002560           115
## 3 E02002561            70
## 4 E02002562           110
## 5 E02002563            65
## 6 E02002564           110
```

6. Load the IDBR Local Units for cleaning, last updated 2019 (as of 2023)

```
idbr_units <- readxl::read_xlsx("data/demos/UPC_busprems/IDBR_localunits_2019.xlsx")
# check data classes using: str(idbr_units)

# convert count columns to numeric
idbr_units <- mutate_at(idbr_units, c(2:19), as.numeric)
# select only the first table in the .xlsx file
# (remember .xlsx file contains many tables one below the other)
idbr_units <- idbr_units[9:7209,]

# missing business sites Total column, so we create Totals per MSOA column
idbr_units$idbr_sum_units <- rowSums(idbr_units[,2:19])

# rename MSOA column
idbr_units <- rename(idbr_units, MSOA = 1)

# split first column to obtain MSOA code only using sapply() function
idbr_units$MSOA <- sapply(strsplit(idbr_units$MSOA, " "), "[", 1)

# select relevant columns
idbr_units <- dplyr::select(idbr_units,
                            MSOA, idbr_sum_units)

head(idbr_units)
```

```
## # A tibble: 6 x 2
##   MSOA      idbr_sum_units
##   <chr>              <dbl>
## 1 E02002559            625
## 2 E02002560            135
## 3 E02002561             90
## 4 E02002562            150
## 5 E02002563             80
## 6 E02002564            135
```

7. We can create a new table for all the totals at MSOA level

```
# join IDBR tables together, then join UPC table
all_prems <- left_join(idbr_ent, idbr_units, by = "MSOA") %>%
  left_join(y = upc_MSOA, by = c("MSOA" = "msoa11cd"))

head(all_prems)
```

```
## # A tibble: 6 x 4
##   MSOA      idbr_sum_ents idbr_sum_units upc_bus_prems
##   <chr>             <dbl>          <dbl>         <dbl>
## 1 E02002559           540            625           235
## 2 E02002560           115            135            89
## 3 E02002561            70             90            19
## 4 E02002562           110            150           148
## 5 E02002563            65             80            40
## 6 E02002564           110            135            75
```

8. Analysis & summary statistics

```
# Create difference column of UPC vs IDBR enterprises/businesses (UPC - IDBR)
all_prems$UPC_idbr_ents <- (all_prems$upc_bus_prems) - (all_prems$idbr_sum_ents)
```

```r
# Create difference column of UPC vs IDBR local units (UPC - IDBR)
all_prems$UPC_idbr_units <- (all_prems$upc_bus_prems) - (all_prems$idbr_sum_units)

# create simple absolute differences columns from the differences
all_prems$UPC_idbr_entsA <- abs(all_prems$UPC_idbr_ents)
all_prems$UPC_idbr_unitsA <- abs(all_prems$UPC_idbr_units)


# View total sums of business premises and local units in the UK measured in UPC and each IDBR
all_prems %>%
  summarise(sum(idbr_sum_ents),
            sum(idbr_sum_units),
            sum(upc_bus_prems))
```

```
## # A tibble: 1 x 3
##   `sum(idbr_sum_ents)` `sum(idbr_sum_units)` `sum(upc_bus_prems)`
##                  <dbl>                 <dbl>                <dbl>
## 1              2508090               2900910              1819602
```

```r
# UPC estimates really low, even lower than Code-Point raw data

summary(all_prems)
```

```
##      MSOA           idbr_sum_ents     idbr_sum_units    upc_bus_prems
##  Length:7201        Min.   :   30.0   Min.   :   40.0   Min.   :    4.0
##  Class :character   1st Qu.:  190.0   1st Qu.:  215.0   1st Qu.:   96.0
##  Mode  :character   Median :  290.0   Median :  330.0   Median :  178.0
##                     Mean   :  348.3   Mean   :  402.8   Mean   :  252.7
##                     3rd Qu.:  420.0   3rd Qu.:  485.0   3rd Qu.:  309.0
##                     Max.   :22305.0   Max.   :25605.0   Max.   :16893.0
##  UPC_idbr_ents       UPC_idbr_units     UPC_idbr_entsA    UPC_idbr_unitsA
##  Min.   :-11262.00   Min.   :-12152.0   Min.   :    0.0   Min.   :    0.0
##  1st Qu.:  -189.00   1st Qu.:  -229.0   1st Qu.:   54.0   1st Qu.:   66.0
##  Median :   -87.00   Median :  -127.0   Median :  118.0   Median :  138.0
##  Mean   :   -95.61   Mean   :  -150.2   Mean   :  157.2   Mean   :  176.8
##  3rd Qu.:    -7.00   3rd Qu.:   -47.0   3rd Qu.:  213.0   3rd Qu.:  235.0
##  Max.   :  1544.00   Max.   :   962.0   Max.   :11262.0   Max.   :12152.0
```

```r
# save csv
#write_csv(all_prems, "data/demos/UPC_busprems/all_premises_clean.csv")
```