Mapping Nutritional Inequality: A Spatial Analysis of Food Purchases Across MSOA Regions in Greater London

Aksa Johnson

2025-07-03

Loading packages

```
pacman::p_load(
  tidyverse,   # includes ggplot2, dplyr, readr, etc.
  sf,
  tmap,
  here,
  janitor,
  stringr
)
```

```
tesco <- read_csv("Sep_msoa_grocery.csv")
```

```
## Rows: 977 Columns: 202
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): area_id
## dbl (201): weight, weight_perc2.5, weight_perc25, weight_perc50, weight_perc...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
tesco
```

```
## # A tibble: 977 x 202
##    area_id   weight weight_perc2.5 weight_perc25 weight_perc50 weight_perc75
##    <chr>      <dbl>          <dbl>         <dbl>         <dbl>         <dbl>
##  1 E02000001   315.           37             150           250           400
##  2 E02000002   415.           32.5           160           300           500
##  3 E02000003   421.           32.5           154           320           500
##  4 E02000004   349.           30             125           265           450
##  5 E02000005   429.           30             157           300           500
##  6 E02000007   388.           32.5           150           300           500
##  7 E02000008   399.           32.5           151           300           500
##  8 E02000009   399.           30             150           300           500
##  9 E02000010   409.           32.5           150           300           494
## 10 E02000011   385.           25             150           300           470
```

```
## # i 967 more rows
## # i 196 more variables: weight_perc97.5 <dbl>, weight_std <dbl>,
## #   weight_ci95 <dbl>, volume <dbl>, volume_perc2.5 <dbl>, volume_perc25 <dbl>,
## #   volume_perc50 <dbl>, volume_perc75 <dbl>, volume_perc97.5 <dbl>,
## #   volume_std <dbl>, volume_ci95 <dbl>, fat <dbl>, fat_perc2.5 <dbl>,
## #   fat_perc25 <dbl>, fat_perc50 <dbl>, fat_perc75 <dbl>, fat_perc97.5 <dbl>,
## #   fat_std <dbl>, fat_ci95 <dbl>, saturate <dbl>, saturate_perc2.5 <dbl>, ...
```

```
#view column names and structure
str(tesco)
```

```
## spc_tbl_ [977 x 202] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ area_id                : chr [1:977] "E02000001" "E02000002" "E02000003" "E02000004" ...
##  $ weight                 : num [1:977] 315 415 421 349 429 ...
##  $ weight_perc2.5         : num [1:977] 37 32.5 32.5 30 30 32.5 32.5 30 32.5 25 ...
##  $ weight_perc25          : num [1:977] 150 160 154 125 157 150 151 150 150 150 ...
##  $ weight_perc50          : num [1:977] 250 300 320 265 300 300 300 300 300 300 ...
##  $ weight_perc75          : num [1:977] 400 500 500 450 500 500 500 500 494 470 ...
##  $ weight_perc97.5        : num [1:977] 1000 1380 1500 1000 1500 1000 1500 1000 1000 1000 ...
##  $ weight_std             : num [1:977] 305 610 643 415 679 ...
##  $ weight_ci95            : num [1:977] 3.23 21.68 17.55 15.21 20.88 ...
##  $ volume                 : num [1:977] 100 111 121 102 125 ...
##  $ volume_perc2.5         : num [1:977] 10.5 18.2 15 16 11 ...
##  $ volume_perc25          : num [1:977] 50 45.5 35.5 50 40 ...
##  $ volume_perc50          : num [1:977] 75 80 75 75 75 75 75 100 75 70 ...
##  $ volume_perc75          : num [1:977] 114 150 160 114 200 ...
##  $ volume_perc97.5        : num [1:977] 264 341 400 341 341 ...
##  $ volume_std             : num [1:977] 87.1 100.5 139.9 88.1 116.9 ...
##  $ volume_ci95            : num [1:977] 1.94 8.38 8.68 7.38 8.15 ...
##  $ fat                    : num [1:977] 8.71 9.12 8.75 8.78 8.81 ...
##  $ fat_perc2.5            : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ fat_perc25             : num [1:977] 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 ...
##  $ fat_perc50             : num [1:977] 1.8 2.5 2.1 2.8 2 2.6 2.6 2.7 2.2 2.6 ...
##  $ fat_perc75             : num [1:977] 13.2 14.1 13.5 14.5 13.6 ...
##  $ fat_perc97.5           : num [1:977] 46.8 42.6 45 34.9 44 ...
##  $ fat_std                : num [1:977] 13.9 13.9 13.5 12 13.7 ...
##  $ fat_ci95               : num [1:977] 0.119 0.394 0.298 0.349 0.337 ...
##  $ saturate               : num [1:977] 3.53 3.55 3.32 3.34 3.49 ...
##  $ saturate_perc2.5       : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ saturate_perc25        : num [1:977] 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ saturate_perc50        : num [1:977] 0.5 0.8 0.7 1 0.6 0.9 0.9 0.9 0.8 0.9 ...
##  $ saturate_perc75        : num [1:977] 3.7 4.1 3.5 4.1 4.1 4.1 4.5 4.1 4.1 4.5 ...
##  $ saturate_perc97.5      : num [1:977] 21.3 19.6 19 18.5 19.5 ...
##  $ saturate_std           : num [1:977] 6.48 6.38 6.08 5.3 6.19 ...
##  $ saturate_ci95          : num [1:977] 0.0554 0.1801 0.1348 0.1538 0.1516 ...
##  $ salt                   : num [1:977] 0.55 0.522 0.497 0.6 0.534 ...
##  $ salt_perc2.5           : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ salt_perc25            : num [1:977] 0.01 0.05 0.05 0.05 0.01 0.05 0.01 0.03 0.02 0.01 ...
##  $ salt_perc50            : num [1:977] 0.11 0.22 0.18 0.26 0.13 0.3 0.2 0.24 0.2 0.2 ...
##  $ salt_perc75            : num [1:977] 0.7 0.8 0.782 0.8 0.7 ...
##  $ salt_perc97.5          : num [1:977] 2.6 2.2 2 2.13 2.13 ...
##  $ salt_std               : num [1:977] 4.65 1.25 1.05 1.39 1.81 ...
##  $ salt_ci95              : num [1:977] 0.0398 0.0352 0.0233 0.0405 0.0444 ...
##  $ sugar                  : num [1:977] 9.7 11.2 11.4 11 11.9 ...
```

```
##  $ sugar_perc2.5            : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ sugar_perc25             : num [1:977] 1 1.4 1.5 1.3 1.4 1.2 1.4 1.1 1.3 1.2 ...
##  $ sugar_perc50             : num [1:977] 3.5 4.4 4 3.7 4.6 4.1 4.2 3.7 3.8 4 ...
##  $ sugar_perc75             : num [1:977] 10 11.8 11.8 11.8 12.7 ...
##  $ sugar_perc97.5           : num [1:977] 57 58.3 59.9 58.5 59.9 ...
##  $ sugar_std                : num [1:977] 15.7 16.8 17.4 17 17.7 ...
##  $ sugar_ci95               : num [1:977] 0.134 0.473 0.386 0.495 0.435 ...
##  $ protein                  : num [1:977] 5.36 5.11 5.34 5.71 4.76 ...
##  $ protein_perc2.5          : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ protein_perc25           : num [1:977] 0.6 0.6 0.5 0.6 0.4 0.4 0.4 0.4 0.4 0.3 ...
##  $ protein_perc50           : num [1:977] 2.8 3.2 3.5 3.6 2.4 3.3 3 3.2 2.9 2.2 ...
##  $ protein_perc75           : num [1:977] 7.8 7.9 8.5 8.8 7.4 ...
##  $ protein_perc97.5         : num [1:977] 24 22.5 21.9 23.3 21.9 ...
##  $ protein_std              : num [1:977] 6.73 5.94 5.95 6.32 5.88 ...
##  $ protein_ci95             : num [1:977] 0.0576 0.1677 0.1318 0.1835 0.144 ...
##  $ carb                     : num [1:977] 16 21.2 23.7 20.7 19.5 ...
##  $ carb_perc2.5             : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ carb_perc25              : num [1:977] 0.5 1.5 1.8 1.5 0.3 0.4 0.2 0.2 1.2 0.1 ...
##  $ carb_perc50              : num [1:977] 5.6 10.9 11.8 11.3 8.9 10.9 9.6 9 10 7 ...
##  $ carb_perc75              : num [1:977] 20.9 38.4 45.6 30.9 30.3 ...
##  $ carb_perc97.5            : num [1:977] 71.3 75.1 80.4 73.5 73.5 ...
##  $ carb_std                 : num [1:977] 21.6 24.1 26.5 23.7 23.7 ...
##  $ carb_ci95                : num [1:977] 0.184 0.679 0.586 0.687 0.582 ...
##  $ fibre                    : num [1:977] 1.64 1.68 1.74 1.69 1.55 ...
##  $ fibre_perc2.5            : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ fibre_perc25             : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ fibre_perc50             : num [1:977] 1 1.1 1 1.1 1 1 1 1 1 0.9 ...
##  $ fibre_perc75             : num [1:977] 2.2 2.3 2.4 2.3 2.2 2.2 2.2 2.2 2.2 2.2 ...
##  $ fibre_perc97.5           : num [1:977] 8 7.07 8.5 8 6.9 ...
##  $ fibre_std                : num [1:977] 2.56 2.67 2.59 2.36 2.33 ...
##  $ fibre_ci95               : num [1:977] 0.0219 0.0753 0.0575 0.0685 0.057 ...
##  $ alcohol                  : num [1:977] 0.299 0.131 0.226 0.126 0.129 ...
##  $ alcohol_perc2.5          : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ alcohol_perc25           : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ alcohol_perc50           : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ alcohol_perc75           : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ alcohol_perc97.5         : num [1:977] 7.04 0 0 0 0 0 0 0 0 0 ...
##  $ alcohol_std              : num [1:977] 1.76 1.43 2.13 1.23 1.37 ...
##  $ alcohol_ci95             : num [1:977] 0.0151 0.0403 0.0472 0.0357 0.0336 ...
##  $ energy_fat               : num [1:977] 78.4 82.1 78.8 79 79.3 ...
##  $ energy_fat_perc2.5       : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ energy_fat_perc25        : num [1:977] 1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8 ...
##  $ energy_fat_perc50        : num [1:977] 16.2 22.5 18.9 25.2 18 23.4 23.4 24.3 19.8 23.4 ...
##  $ energy_fat_perc75        : num [1:977] 119 127 122 130 123 ...
##  $ energy_fat_perc97.5      : num [1:977] 421 383 405 314 396 ...
##  $ energy_fat_std           : num [1:977] 125 126 121 108 124 ...
##  $ energy_fat_ci95          : num [1:977] 1.07 3.54 2.69 3.14 3.03 ...
##  $ energy_saturate          : num [1:977] 31.8 31.9 29.9 30 31.4 ...
##  $ energy_saturate_perc2.5  : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##  $ energy_saturate_perc25   : num [1:977] 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 ...
##  $ energy_saturate_perc50   : num [1:977] 4.5 7.2 6.3 9 5.4 8.1 8.1 8.1 7.2 8.1 ...
##  $ energy_saturate_perc75   : num [1:977] 33.3 36.9 31.5 36.9 36.9 36.9 40.5 36.9 36.9 40.5 ...
##  $ energy_saturate_perc97.5 : num [1:977] 192 176 171 166 175 ...
##  $ energy_saturate_std      : num [1:977] 58.3 57.5 54.8 47.7 55.7 ...
```

```
##  $ energy_saturate_ci95     : num [1:977] 0.499 1.621 1.213 1.385 1.364 ...
##  $ energy_sugar             : num [1:977] 38.8 44.8 45.5 44.1 47.7 ...
##  $ energy_sugar_perc2.5     : num [1:977] 0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
##  - attr(*, "spec")=
##   .. cols(
##   ..   area_id = col_character(),
##   ..   weight = col_double(),
##   ..   weight_perc2.5 = col_double(),
##   ..   weight_perc25 = col_double(),
##   ..   weight_perc50 = col_double(),
##   ..   weight_perc75 = col_double(),
##   ..   weight_perc97.5 = col_double(),
##   ..   weight_std = col_double(),
##   ..   weight_ci95 = col_double(),
##   ..   volume = col_double(),
##   ..   volume_perc2.5 = col_double(),
##   ..   volume_perc25 = col_double(),
##   ..   volume_perc50 = col_double(),
##   ..   volume_perc75 = col_double(),
##   ..   volume_perc97.5 = col_double(),
##   ..   volume_std = col_double(),
##   ..   volume_ci95 = col_double(),
##   ..   fat = col_double(),
##   ..   fat_perc2.5 = col_double(),
##   ..   fat_perc25 = col_double(),
##   ..   fat_perc50 = col_double(),
##   ..   fat_perc75 = col_double(),
##   ..   fat_perc97.5 = col_double(),
##   ..   fat_std = col_double(),
##   ..   fat_ci95 = col_double(),
##   ..   saturate = col_double(),
##   ..   saturate_perc2.5 = col_double(),
##   ..   saturate_perc25 = col_double(),
##   ..   saturate_perc50 = col_double(),
##   ..   saturate_perc75 = col_double(),
##   ..   saturate_perc97.5 = col_double(),
##   ..   saturate_std = col_double(),
##   ..   saturate_ci95 = col_double(),
##   ..   salt = col_double(),
##   ..   salt_perc2.5 = col_double(),
##   ..   salt_perc25 = col_double(),
##   ..   salt_perc50 = col_double(),
##   ..   salt_perc75 = col_double(),
##   ..   salt_perc97.5 = col_double(),
##   ..   salt_std = col_double(),
##   ..   salt_ci95 = col_double(),
##   ..   sugar = col_double(),
##   ..   sugar_perc2.5 = col_double(),
##   ..   sugar_perc25 = col_double(),
##   ..   sugar_perc50 = col_double(),
##   ..   sugar_perc75 = col_double(),
##   ..   sugar_perc97.5 = col_double(),
##   ..   sugar_std = col_double(),
```

```
##   ..   sugar_ci95 = col_double(),
##   ..   protein = col_double(),
##   ..   protein_perc2.5 = col_double(),
##   ..   protein_perc25 = col_double(),
##   ..   protein_perc50 = col_double(),
##   ..   protein_perc75 = col_double(),
##   ..   protein_perc97.5 = col_double(),
##   ..   protein_std = col_double(),
##   ..   protein_ci95 = col_double(),
##   ..   carb = col_double(),
##   ..   carb_perc2.5 = col_double(),
##   ..   carb_perc25 = col_double(),
##   ..   carb_perc50 = col_double(),
##   ..   carb_perc75 = col_double(),
##   ..   carb_perc97.5 = col_double(),
##   ..   carb_std = col_double(),
##   ..   carb_ci95 = col_double(),
##   ..   fibre = col_double(),
##   ..   fibre_perc2.5 = col_double(),
##   ..   fibre_perc25 = col_double(),
##   ..   fibre_perc50 = col_double(),
##   ..   fibre_perc75 = col_double(),
##   ..   fibre_perc97.5 = col_double(),
##   ..   fibre_std = col_double(),
##   ..   fibre_ci95 = col_double(),
##   ..   alcohol = col_double(),
##   ..   alcohol_perc2.5 = col_double(),
##   ..   alcohol_perc25 = col_double(),
##   ..   alcohol_perc50 = col_double(),
##   ..   alcohol_perc75 = col_double(),
##   ..   alcohol_perc97.5 = col_double(),
##   ..   alcohol_std = col_double(),
##   ..   alcohol_ci95 = col_double(),
##   ..   energy_fat = col_double(),
##   ..   energy_fat_perc2.5 = col_double(),
##   ..   energy_fat_perc25 = col_double(),
##   ..   energy_fat_perc50 = col_double(),
##   ..   energy_fat_perc75 = col_double(),
##   ..   energy_fat_perc97.5 = col_double(),
##   ..   energy_fat_std = col_double(),
##   ..   energy_fat_ci95 = col_double(),
##   ..   energy_saturate = col_double(),
##   ..   energy_saturate_perc2.5 = col_double(),
##   ..   energy_saturate_perc25 = col_double(),
##   ..   energy_saturate_perc50 = col_double(),
##   ..   energy_saturate_perc75 = col_double(),
##   ..   energy_saturate_perc97.5 = col_double(),
##   ..   energy_saturate_std = col_double(),
##   ..   energy_saturate_ci95 = col_double(),
##   ..   energy_sugar = col_double(),
##   ..   energy_sugar_perc2.5 = col_double(),
##   ..   energy_sugar_perc25 = col_double(),
##   ..   energy_sugar_perc50 = col_double(),
##   ..   energy_sugar_perc75 = col_double(),
```

```
## .. energy_sugar_perc97.5 = col_double(),
## .. energy_sugar_std = col_double(),
## .. energy_sugar_ci95 = col_double(),
## .. energy_protein = col_double(),
## .. energy_protein_perc2.5 = col_double(),
## .. energy_protein_perc25 = col_double(),
## .. energy_protein_perc50 = col_double(),
## .. energy_protein_perc75 = col_double(),
## .. energy_protein_perc97.5 = col_double(),
## .. energy_protein_std = col_double(),
## .. energy_protein_ci95 = col_double(),
## .. energy_carb = col_double(),
## .. energy_carb_perc2.5 = col_double(),
## .. energy_carb_perc25 = col_double(),
## .. energy_carb_perc50 = col_double(),
## .. energy_carb_perc75 = col_double(),
## .. energy_carb_perc97.5 = col_double(),
## .. energy_carb_std = col_double(),
## .. energy_carb_ci95 = col_double(),
## .. energy_fibre = col_double(),
## .. energy_fibre_perc2.5 = col_double(),
## .. energy_fibre_perc25 = col_double(),
## .. energy_fibre_perc50 = col_double(),
## .. energy_fibre_perc75 = col_double(),
## .. energy_fibre_perc97.5 = col_double(),
## .. energy_fibre_std = col_double(),
## .. energy_fibre_ci95 = col_double(),
## .. energy_alcohol = col_double(),
## .. energy_alcohol_perc2.5 = col_double(),
## .. energy_alcohol_perc25 = col_double(),
## .. energy_alcohol_perc50 = col_double(),
## .. energy_alcohol_perc75 = col_double(),
## .. energy_alcohol_perc97.5 = col_double(),
## .. energy_alcohol_std = col_double(),
## .. energy_alcohol_ci95 = col_double(),
## .. energy_tot = col_double(),
## .. energy_tot_perc2.5 = col_double(),
## .. energy_tot_perc25 = col_double(),
## .. energy_tot_perc50 = col_double(),
## .. energy_tot_perc75 = col_double(),
## .. energy_tot_perc97.5 = col_double(),
## .. energy_tot_std = col_double(),
## .. energy_tot_ci95 = col_double(),
## .. f_energy_fat = col_double(),
## .. f_energy_saturate = col_double(),
## .. f_energy_sugar = col_double(),
## .. f_energy_protein = col_double(),
## .. f_energy_carb = col_double(),
## .. f_energy_fibre = col_double(),
## .. f_energy_alcohol = col_double(),
## .. energy_density = col_double(),
## .. h_nutrients_weight = col_double(),
## .. h_nutrients_weight_norm = col_double(),
## .. h_nutrients_calories = col_double(),
```

```
##   ..   h_nutrients_calories_norm = col_double(),
##   ..   f_beer = col_double(),
##   ..   f_dairy = col_double(),
##   ..   f_eggs = col_double(),
##   ..   f_fats_oils = col_double(),
##   ..   f_fish = col_double(),
##   ..   f_fruit_veg = col_double(),
##   ..   f_grains = col_double(),
##   ..   f_meat_red = col_double(),
##   ..   f_poultry = col_double(),
##   ..   f_readymade = col_double(),
##   ..   f_sauces = col_double(),
##   ..   f_soft_drinks = col_double(),
##   ..   f_spirits = col_double(),
##   ..   f_sweets = col_double(),
##   ..   f_tea_coffee = col_double(),
##   ..   f_water = col_double(),
##   ..   f_wine = col_double(),
##   ..   f_dairy_weight = col_double(),
##   ..   f_eggs_weight = col_double(),
##   ..   f_fats_oils_weight = col_double(),
##   ..   f_fish_weight = col_double(),
##   ..   f_fruit_veg_weight = col_double(),
##   ..   f_grains_weight = col_double(),
##   ..   f_meat_red_weight = col_double(),
##   ..   f_poultry_weight = col_double(),
##   ..   f_readymade_weight = col_double(),
##   ..   f_sauces_weight = col_double(),
##   ..   f_sweets_weight = col_double(),
##   ..   h_items = col_double(),
##   ..   h_items_norm = col_double(),
##   ..   h_items_weight = col_double(),
##   ..   h_items_weight_norm = col_double(),
##   ..   representativeness_norm = col_double(),
##   ..   transaction_days = col_double(),
##   ..   num_transactions = col_double(),
##   ..   man_day = col_double(),
##   ..   population = col_double(),
##   ..   male = col_double(),
##   ..   female = col_double(),
##   ..   age_0_17 = col_double(),
##   ..   age_18_64 = col_double(),
##   ..   `age_65+` = col_double(),
##   ..   avg_age = col_double(),
##   ..   area_sq_km = col_double(),
##   ..   people_per_sq_km = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(tesco)
```

```
## # A tibble: 6 x 202
##   area_id   weight weight_perc2.5 weight_perc25 weight_perc50 weight_perc75
##   <chr>      <dbl>          <dbl>         <dbl>         <dbl>         <dbl>
```

```
## 1 E02000001   315.           37             150           250           400
## 2 E02000002   415.           32.5           160           300           500
## 3 E02000003   421.           32.5           154           320           500
## 4 E02000004   349.           30             125           265           450
## 5 E02000005   429.           30             157           300           500
## 6 E02000007   388.           32.5           150           300           500
## # i 196 more variables: weight_perc97.5 <dbl>, weight_std <dbl>,
## #   weight_ci95 <dbl>, volume <dbl>, volume_perc2.5 <dbl>, volume_perc25 <dbl>,
## #   volume_perc50 <dbl>, volume_perc75 <dbl>, volume_perc97.5 <dbl>,
## #   volume_std <dbl>, volume_ci95 <dbl>, fat <dbl>, fat_perc2.5 <dbl>,
## #   fat_perc25 <dbl>, fat_perc50 <dbl>, fat_perc75 <dbl>, fat_perc97.5 <dbl>,
## #   fat_std <dbl>, fat_ci95 <dbl>, saturate <dbl>, saturate_perc2.5 <dbl>,
## #   saturate_perc25 <dbl>, saturate_perc50 <dbl>, saturate_perc75 <dbl>, ...
```

```r
names(tesco)
```

```
##   [1] "area_id"                    "weight"
##   [3] "weight_perc2.5"             "weight_perc25"
##   [5] "weight_perc50"              "weight_perc75"
##   [7] "weight_perc97.5"            "weight_std"
##   [9] "weight_ci95"                "volume"
##  [11] "volume_perc2.5"             "volume_perc25"
##  [13] "volume_perc50"              "volume_perc75"
##  [15] "volume_perc97.5"            "volume_std"
##  [17] "volume_ci95"                "fat"
##  [19] "fat_perc2.5"                "fat_perc25"
##  [21] "fat_perc50"                 "fat_perc75"
##  [23] "fat_perc97.5"               "fat_std"
##  [25] "fat_ci95"                   "saturate"
##  [27] "saturate_perc2.5"           "saturate_perc25"
##  [29] "saturate_perc50"            "saturate_perc75"
##  [31] "saturate_perc97.5"          "saturate_std"
##  [33] "saturate_ci95"              "salt"
##  [35] "salt_perc2.5"               "salt_perc25"
##  [37] "salt_perc50"                "salt_perc75"
##  [39] "salt_perc97.5"              "salt_std"
##  [41] "salt_ci95"                  "sugar"
##  [43] "sugar_perc2.5"              "sugar_perc25"
##  [45] "sugar_perc50"               "sugar_perc75"
##  [47] "sugar_perc97.5"             "sugar_std"
##  [49] "sugar_ci95"                 "protein"
##  [51] "protein_perc2.5"            "protein_perc25"
##  [53] "protein_perc50"             "protein_perc75"
##  [55] "protein_perc97.5"           "protein_std"
##  [57] "protein_ci95"               "carb"
##  [59] "carb_perc2.5"               "carb_perc25"
##  [61] "carb_perc50"                "carb_perc75"
##  [63] "carb_perc97.5"              "carb_std"
##  [65] "carb_ci95"                  "fibre"
##  [67] "fibre_perc2.5"              "fibre_perc25"
##  [69] "fibre_perc50"               "fibre_perc75"
##  [71] "fibre_perc97.5"             "fibre_std"
##  [73] "fibre_ci95"                 "alcohol"
##  [75] "alcohol_perc2.5"            "alcohol_perc25"
```

```
##  [77] "alcohol_perc50"            "alcohol_perc75"
##  [79] "alcohol_perc97.5"          "alcohol_std"
##  [81] "alcohol_ci95"              "energy_fat"
##  [83] "energy_fat_perc2.5"        "energy_fat_perc25"
##  [85] "energy_fat_perc50"         "energy_fat_perc75"
##  [87] "energy_fat_perc97.5"       "energy_fat_std"
##  [89] "energy_fat_ci95"           "energy_saturate"
##  [91] "energy_saturate_perc2.5"   "energy_saturate_perc25"
##  [93] "energy_saturate_perc50"    "energy_saturate_perc75"
##  [95] "energy_saturate_perc97.5"  "energy_saturate_std"
##  [97] "energy_saturate_ci95"      "energy_sugar"
##  [99] "energy_sugar_perc2.5"      "energy_sugar_perc25"
## [101] "energy_sugar_perc50"       "energy_sugar_perc75"
## [103] "energy_sugar_perc97.5"     "energy_sugar_std"
## [105] "energy_sugar_ci95"         "energy_protein"
## [107] "energy_protein_perc2.5"    "energy_protein_perc25"
## [109] "energy_protein_perc50"     "energy_protein_perc75"
## [111] "energy_protein_perc97.5"   "energy_protein_std"
## [113] "energy_protein_ci95"       "energy_carb"
## [115] "energy_carb_perc2.5"       "energy_carb_perc25"
## [117] "energy_carb_perc50"        "energy_carb_perc75"
## [119] "energy_carb_perc97.5"      "energy_carb_std"
## [121] "energy_carb_ci95"          "energy_fibre"
## [123] "energy_fibre_perc2.5"      "energy_fibre_perc25"
## [125] "energy_fibre_perc50"       "energy_fibre_perc75"
## [127] "energy_fibre_perc97.5"     "energy_fibre_std"
## [129] "energy_fibre_ci95"         "energy_alcohol"
## [131] "energy_alcohol_perc2.5"    "energy_alcohol_perc25"
## [133] "energy_alcohol_perc50"     "energy_alcohol_perc75"
## [135] "energy_alcohol_perc97.5"   "energy_alcohol_std"
## [137] "energy_alcohol_ci95"       "energy_tot"
## [139] "energy_tot_perc2.5"        "energy_tot_perc25"
## [141] "energy_tot_perc50"         "energy_tot_perc75"
## [143] "energy_tot_perc97.5"       "energy_tot_std"
## [145] "energy_tot_ci95"           "f_energy_fat"
## [147] "f_energy_saturate"         "f_energy_sugar"
## [149] "f_energy_protein"          "f_energy_carb"
## [151] "f_energy_fibre"            "f_energy_alcohol"
## [153] "energy_density"            "h_nutrients_weight"
## [155] "h_nutrients_weight_norm"   "h_nutrients_calories"
## [157] "h_nutrients_calories_norm" "f_beer"
## [159] "f_dairy"                   "f_eggs"
## [161] "f_fats_oils"               "f_fish"
## [163] "f_fruit_veg"               "f_grains"
## [165] "f_meat_red"                "f_poultry"
## [167] "f_readymade"               "f_sauces"
## [169] "f_soft_drinks"             "f_spirits"
## [171] "f_sweets"                  "f_tea_coffee"
## [173] "f_water"                   "f_wine"
## [175] "f_dairy_weight"            "f_eggs_weight"
## [177] "f_fats_oils_weight"        "f_fish_weight"
## [179] "f_fruit_veg_weight"        "f_grains_weight"
## [181] "f_meat_red_weight"         "f_poultry_weight"
## [183] "f_readymade_weight"        "f_sauces_weight"
```

```
## [185] "f_sweets_weight"          "h_items"
## [187] "h_items_norm"             "h_items_weight"
## [189] "h_items_weight_norm"      "representativeness_norm"
## [191] "transaction_days"         "num_transactions"
## [193] "man_day"                  "population"
## [195] "male"                     "female"
## [197] "age_0_17"                 "age_18_64"
## [199] "age_65+"                  "avg_age"
## [201] "area_sq_km"               "people_per_sq_km"
```

```
#checking for missing values
any(colSums(is.na(tesco)) > 0)
```

```
## [1] FALSE
```

To compare the nutritional quality of food purchases across MSOAs, I created a factor Nutritional quality Score(NQS) by combining multiple nutritional indicators into a single index. The variables included total energy, sugar, saturated fat, fibre and nutrient diversity. These variables are selected in evidence-based dietary guidelines provided by World Health Organisation(WHO) and some articles about dietary requirements. These organisations identify high intake of sugar, saturated fat, and energy(calories) as risk factor for obesity and non-communicable diseases, including diabetes, heart diseases, stroke and cancer.On the other hand, fibre and nutrient diversity are linked to overall diet quality and protective benefits. These variables were standardized using z scores to ensure that they are on same scale which wont be affect the final score.

```
#standardize each variable

tesco <- tesco%>%
  mutate(
    z_calories = scale(energy_tot),
    z_sugar = scale(sugar),
    z_sat_fat = scale(saturate),
    z_fibre = scale(fibre),
    z_nutrient_diversity = scale(h_nutrients_weight_norm)
    )
```

Computing Nutritional Quality Score The NQS formula is adapted from similar composite methods used in public health. Nutrients linked to healthy diets, such as fibre and nutrient diversity, contributed positively to the quality score, whereas components that are often recommeneded for limited intake, such as, sugars, saturated fat, and total energy, contributed negatively. (The Nutri-Score: A Science-Based Front-of-Pack Nutrition Label 2021)

For computing Nutrient Quality Score, fibre and nutrient diversity were positively weighted in the score to reflect their contribution to healthy diets, while enrgy, sugar, fat were neagtively weighted due to their contribution with poor dietary outcomes. The resulting NQS provides a composite measure of diet quality for each, allowing for meaningful spatial comparison and visualisation.

```
 tesco <-  tesco%>%
   mutate(
     NQS = z_fibre +z_nutrient_diversity - z_sat_fat - z_sugar - z_calories
   )
head(tesco$NQS)
```

```
##           [,1]
```

```
## [1,]  3.240370
## [2,] -3.606438
## [3,] -3.753462
## [4,] -1.597517
## [5,] -3.988615
## [6,] -6.346609
```

Compute proportions of age groups

```
#
tesco <- tesco %>%
  mutate(
    prop_children = age_0_17 / population,
    prop_adults = age_18_64 / population,
    prop_elderly = `age_65+` / population
  )
```

Calculate age-specific NQS values

```
tesco <- tesco %>%
  mutate(
    NQS_children = NQS * prop_children,
    NQS_adults = NQS * prop_adults,
    NQS_elderly = NQS * prop_elderly
  )
head(tesco %>% select(area_id, NQS, NQS_children, NQS_adults, NQS_elderly))
```

```
## # A tibble: 6 x 5
##   area_id   NQS[,1] NQS_children[,1] NQS_adults[,1] NQS_elderly[,1]
##   <chr>       <dbl>            <dbl>          <dbl>           <dbl>
## 1 E02000001    3.24            0.493           2.11           0.640
## 2 E02000002   -3.61           -1.05           -2.03          -0.527
## 3 E02000003   -3.75           -0.967          -2.33          -0.458
## 4 E02000004   -1.60           -0.355          -0.988         -0.254
## 5 E02000005   -3.99           -1.16           -2.40          -0.436
## 6 E02000007   -6.35           -1.91           -3.73          -0.707
```

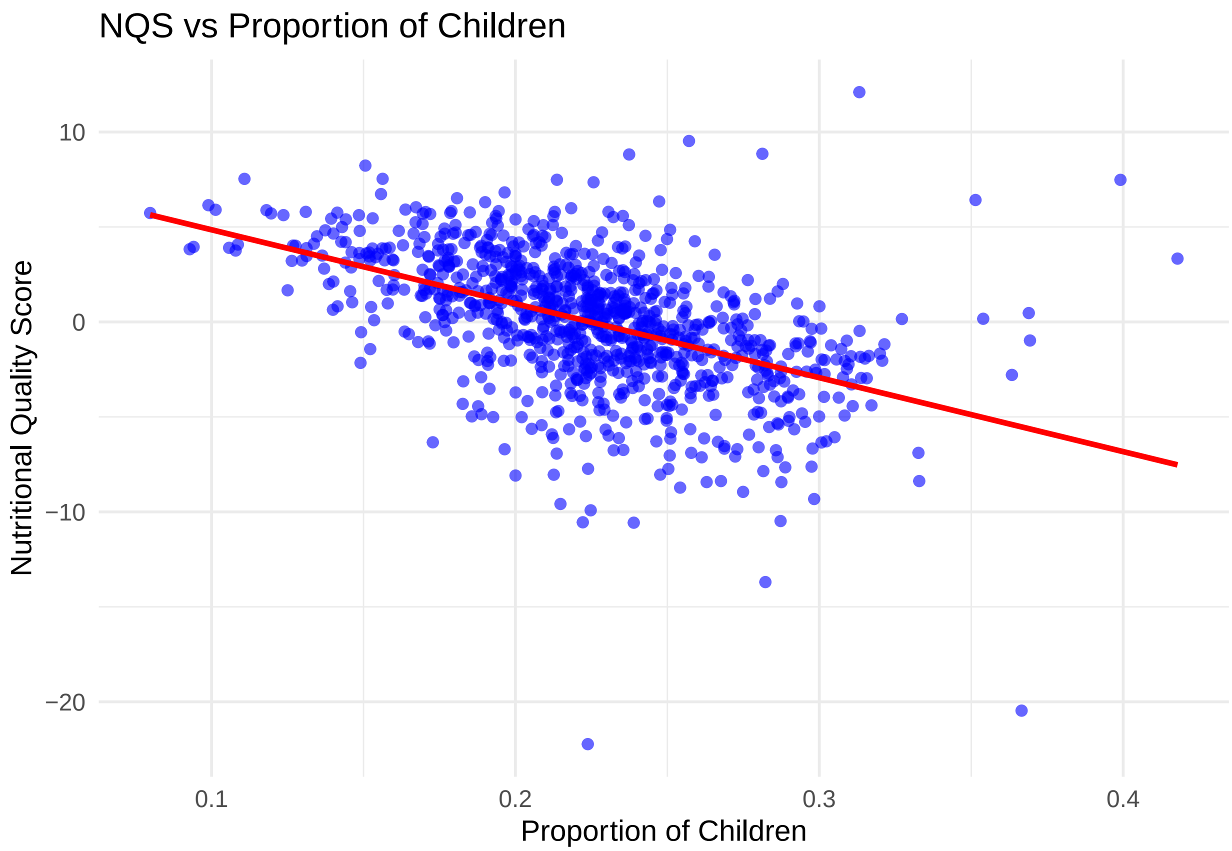
```
#Standardize it
tesco <- tesco %>%
  mutate(
    z_NQS_children = scale(NQS_children),
    z_NQS_adults = scale(NQS_adults),
    z_NQS_elderly = scale(NQS_elderly)
    )
head(tesco %>% select(area_id, NQS, z_NQS_children, z_NQS_adults, z_NQS_elderly))
```

```
## # A tibble: 6 x 5
##   area_id   NQS[,1] z_NQS_children[,1] z_NQS_adults[,1] z_NQS_elderly[,1]
##   <chr>       <dbl>              <dbl>            <dbl>             <dbl>
## 1 E02000001    3.24              0.691            0.900              1.46
## 2 E02000002   -3.61             -1.18            -0.940             -1.17
## 3 E02000003   -3.75             -1.08            -1.07              -1.01
```

```
## 4 E02000004   -1.60               -0.340          -0.477            -0.552
## 5 E02000005   -3.99               -1.31           -1.10             -0.963
## 6 E02000007   -6.35               -2.23           -1.70             -1.57
```

```r
# Children vs NQS
ggplot(tesco, aes(x = prop_children, y = NQS)) +
  geom_point(alpha = 0.6, color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "NQS vs Proportion of Children", x = "Proportion of Children", y = "Nutritional Quality S
  theme_minimal()
```

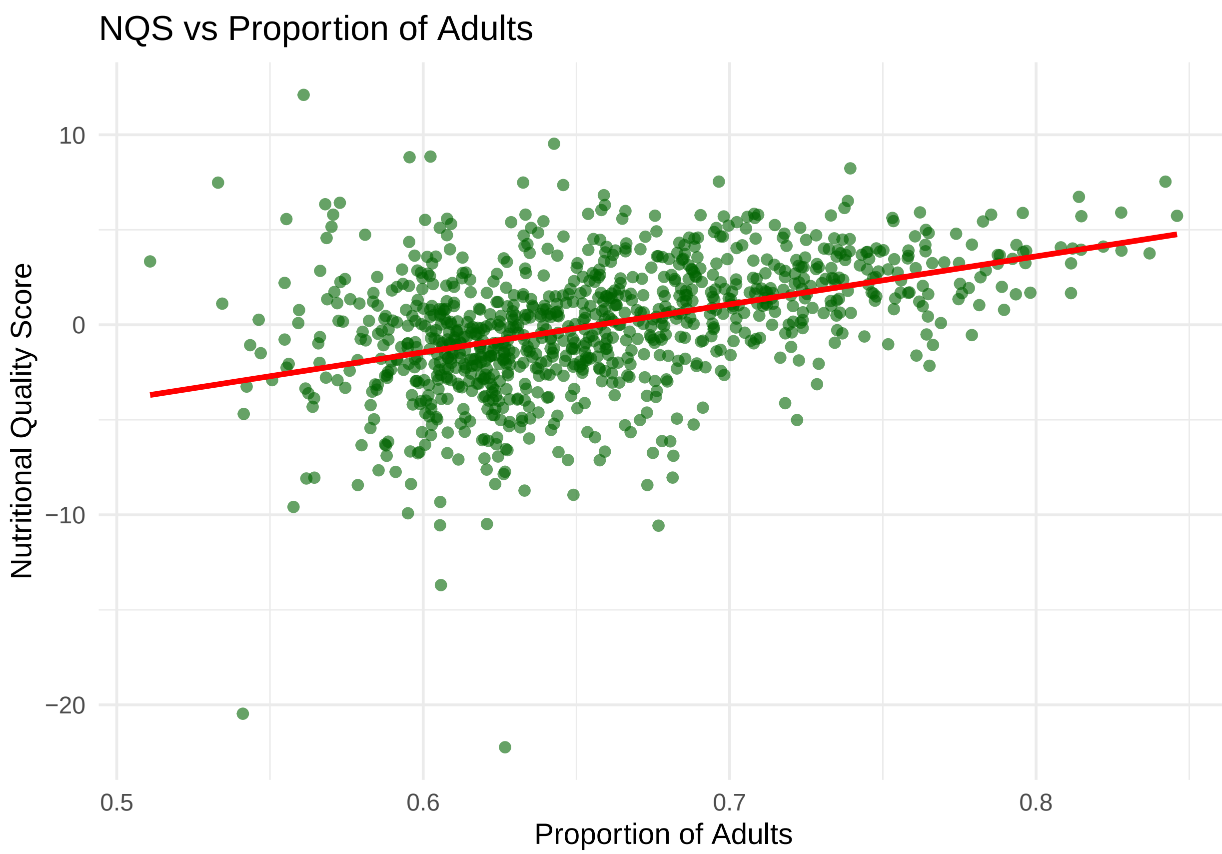
```
## `geom_smooth()` using formula = 'y ~ x'
```



This plot shows a downward trend.There is a negative relationship between the proportion of children and the NQS. That means, the more children in an area, the lower the nutritional quality of food.Therefore, it is essential to examine the main NQS and weight it by age group to assess the nutritional vulnerability of children across London.

```r
# Adults vs NQS
ggplot(tesco, aes(x = prop_adults, y = NQS)) +
  geom_point(alpha = 0.6, color = "darkgreen") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "NQS vs Proportion of Adults", x = "Proportion of Adults", y = "Nutritional Quality Scor
  theme_minimal()
```

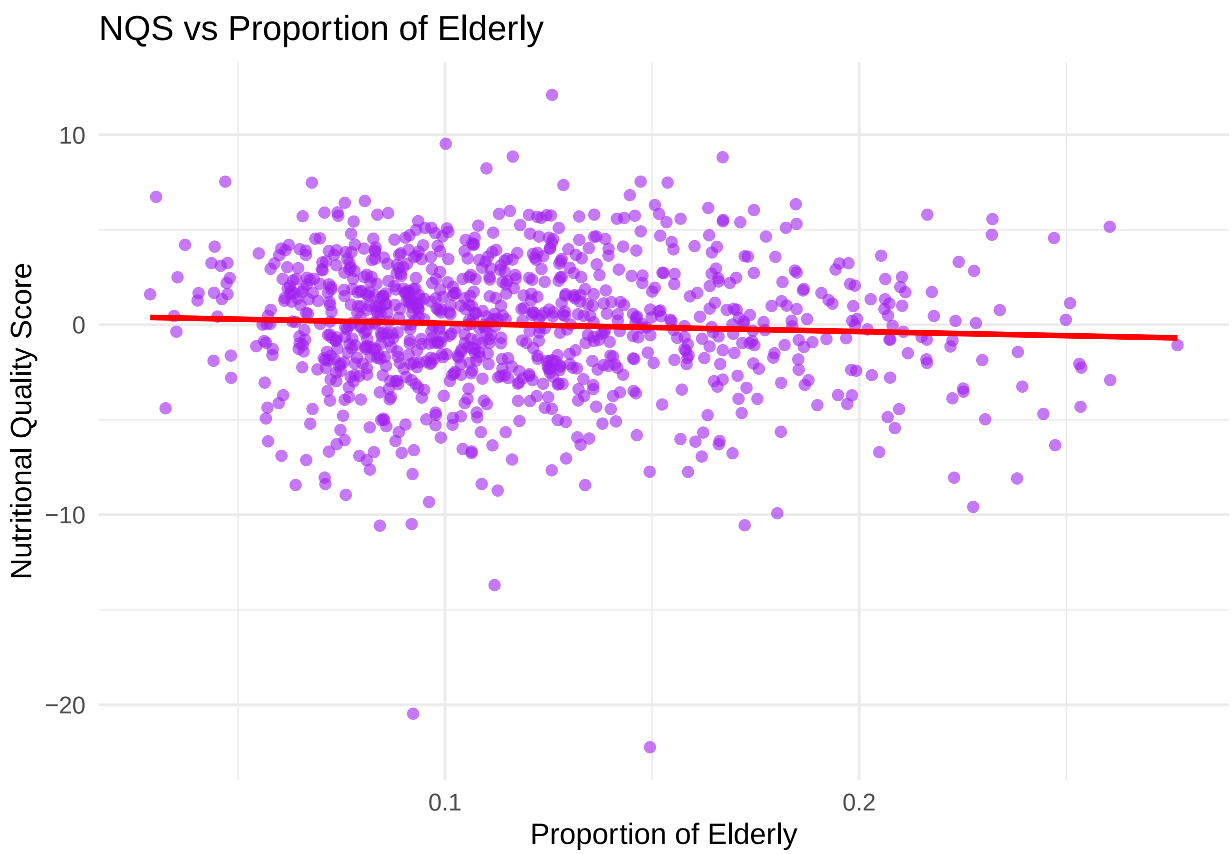
```
## `geom_smooth()` using formula = 'y ~ x'
```



This scatterplot shows a positive relatuonship between adults and NQS.That means the areas with more adults (18–64 years) tend to have better nutritional quality in their food purchases.This trend contrasts with the negative association observed in areas with more children, underlining potential age-related disparities in dietary choices or food access.

```r
# Elderly vs NQS
ggplot(tesco, aes(x = prop_elderly, y = NQS)) +
  geom_point(alpha = 0.6, color = "purple") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "NQS vs Proportion of Elderly", x = "Proportion of Elderly", y = "Nutritional Quality Sc
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

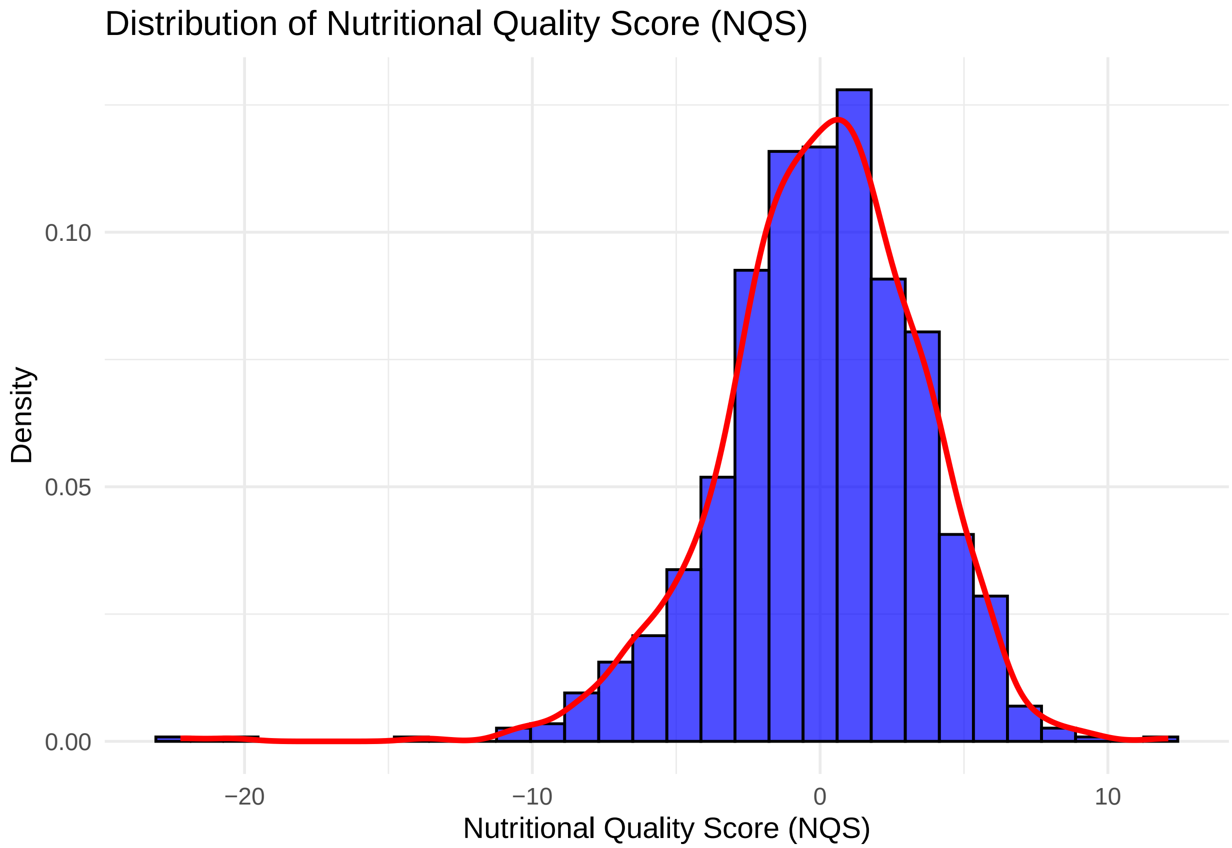


The red regression line is almost flat.This shows no clear relationship between the proportion of elderly and the Nutritional Quality Score (NQS).

```
ggplot(tesco, aes(x = NQS)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "blue", color = "black", alpha = 0.7) +
  geom_density(color = "red", size = 1) +
  labs(
    title = "Distribution of Nutritional Quality Score (NQS)",
    x = "Nutritional Quality Score (NQS)",
    y = "Density"
  ) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



The distribution of the Nutritional Quality Score (NQS) across MSOAs in Greater London follows an approximately normal pattern, with the majority of areas falling near the average. However, there are several regions with significantly low NQS values, highlighting pockets of poor nutritional purchasing behavior. These regions warrant further attention, especially when a high proportion of children reside in them. ###Creating Maps

```
packageVersion("tmap")
```

```
## [1] '4.2'
```

```
year_msoa_grocery <- tesco%>%
  rename(msoa_code = area_id)
msoas <-st_read(here("statistical-gis-boundaries-london/ESRI/MSOA_2011_London_gen_MHW.shp")) %>%rename(
```

```
## Reading layer `MSOA_2011_London_gen_MHW' from data source
##   `C:\Users\aksaj\OneDrive\Documents\PROJECT\statistical-gis-boundaries-london\ESRI\MSOA_2011_London_
##   using driver `ESRI Shapefile'
## Simple feature collection with 983 features and 12 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 503574.2 ymin: 155850.8 xmax: 561956.7 ymax: 200933.6
## Projected CRS: OSGB36 / British National Grid
```

```
tesco_and_msoas <- inner_join(msoas,year_msoa_grocery)
```

```
## Joining with `by = join_by(msoa_code)`
```

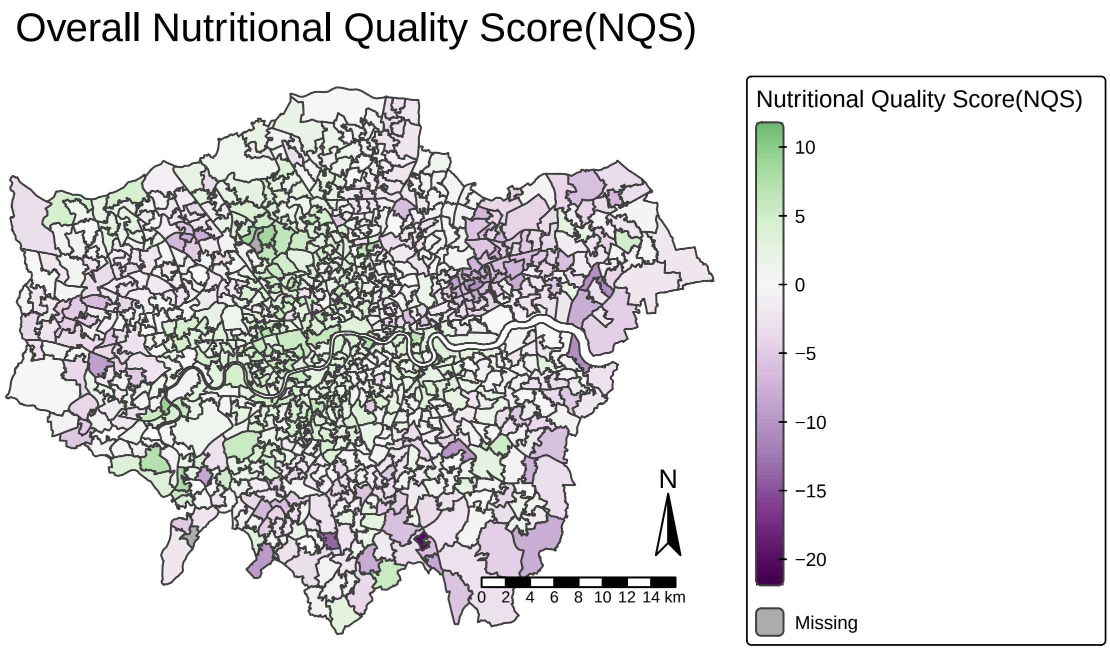
##Overall NQS Map

```r
tmap_mode("plot")
```

```
## i tmap modes "plot" - "view"
## i toggle with `tmap::ttm()`
```

```r
tm_shape(tesco_and_msoas) +
  tm_polygons(
    fill = "NQS",
    fill.legend = tm_legend(title = "Nutritional Quality Score(NQS)", reverse = TRUE),
    fill.scale = tm_scale_continuous(limits = c(-22,12),values = "brewer.PRGn")
  ) +
  tm_title(" Overall Nutritional Quality Score(NQS)") +
  tm_compass() +
  tm_scalebar() +
  tm_layout(frame = FALSE, bg.color = "white")
```

```
## Values have been found that are lower than the lowest limit. These 'outliers' have been set to NA. U
## Values have been found that are higher than the upper limit. These 'outliers' have been set to NA. U
## Variable(s) "fill" contains positive and negative values, so midpoint is set to 0. Set midpoint = NA
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "brewer.PRGn" is
## named "prgn" (in long format "brewer.prgn")[plot mode] fit legend/component: Some legend items or ma
## fit well, and are therefore rescaled.
## i Set the tmap option `component.autoscale = FALSE` to disable rescaling.
```



This map presents the overall diet quality of food purchases across Greater London.

High NQS (dark green): Predominantly in central, west, and southwest London, indicating healthier purchasing behaviour - higher fibre, greater nutrient diversity, and lower sugar, saturated fat, and calories.

Low NQS (purple): Concentrated in eastern and southeastern London, suggesting more processed and less diverse food purchases. White to pale shades: Areas close to the London average approximately NQS= 0.

Grey (Missing): Areas without NQS data due to no matching purchase records or values beyond the set scale limits. The spatial pattern reveals a nutritional divide, with healthier purchasing trends more common in wealthier, better-connected areas.

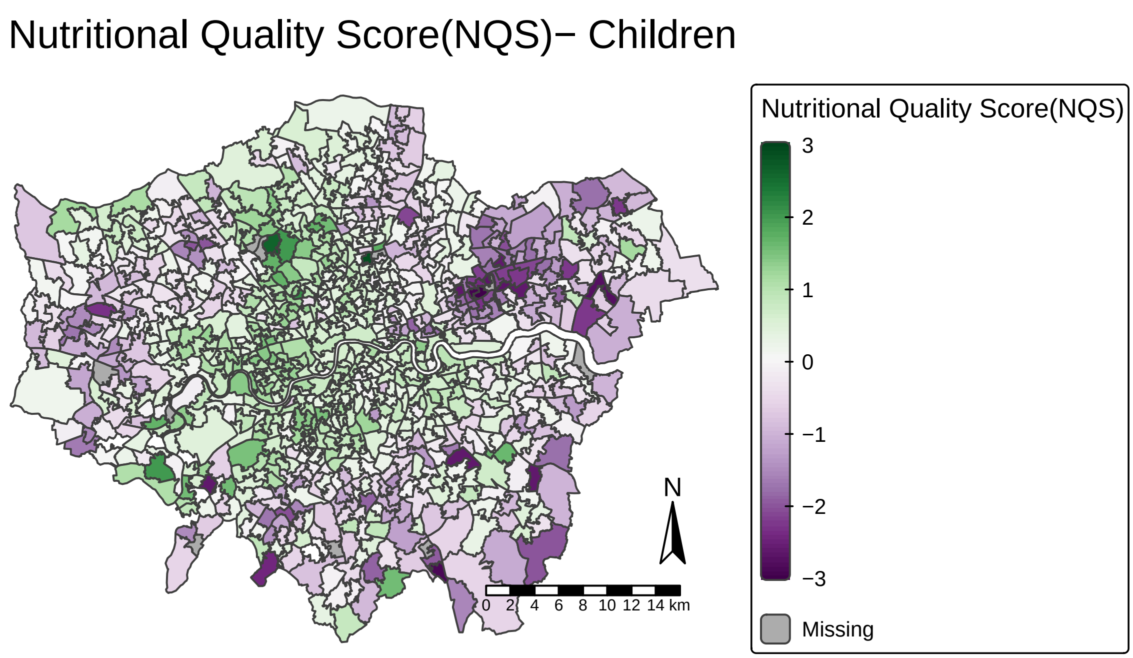
```r
tm_shape(tesco_and_msoas) +
  tm_polygons(
    fill = "z_NQS_children",
    fill.legend = tm_legend(title = "Nutritional Quality Score(NQS)", reverse = TRUE),
    fill.scale = tm_scale_continuous(limits = c(-3,3),values = "brewer.PRGn")
    ) +
  tm_title("Nutritional Quality Score(NQS)- Children") +
  tm_compass() +
  tm_scalebar() +
  tm_layout(frame = FALSE, bg.color = "white")
```

```
## Values have been found that are lower than the lowest limit. These 'outliers' have been set to NA. U
```

```
## Values have been found that are higher than the upper limit. These 'outliers' have been set to NA. U
```

```
## Variable(s) "fill" contains positive and negative values, so midpoint is set to 0. Set midpoint = NA
```

```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "brewer.PRGn" is
## named "prgn" (in long format "brewer.prgn")
## [plot mode] fit legend/component: Some legend items or map components do not
## fit well, and are therefore rescaled.
## i Set the tmap option `component.autoscale = FALSE` to disable rescaling.
```



This map isolates the nutritional quality of purchases weighted by the proportion of children (aged 0-17) in each MSOA.

High NQS (dark green): Found mostly in west and southwest London, potentially reflecting higher parental health awareness, better access to fresh produce, and healthier retail environments for families.

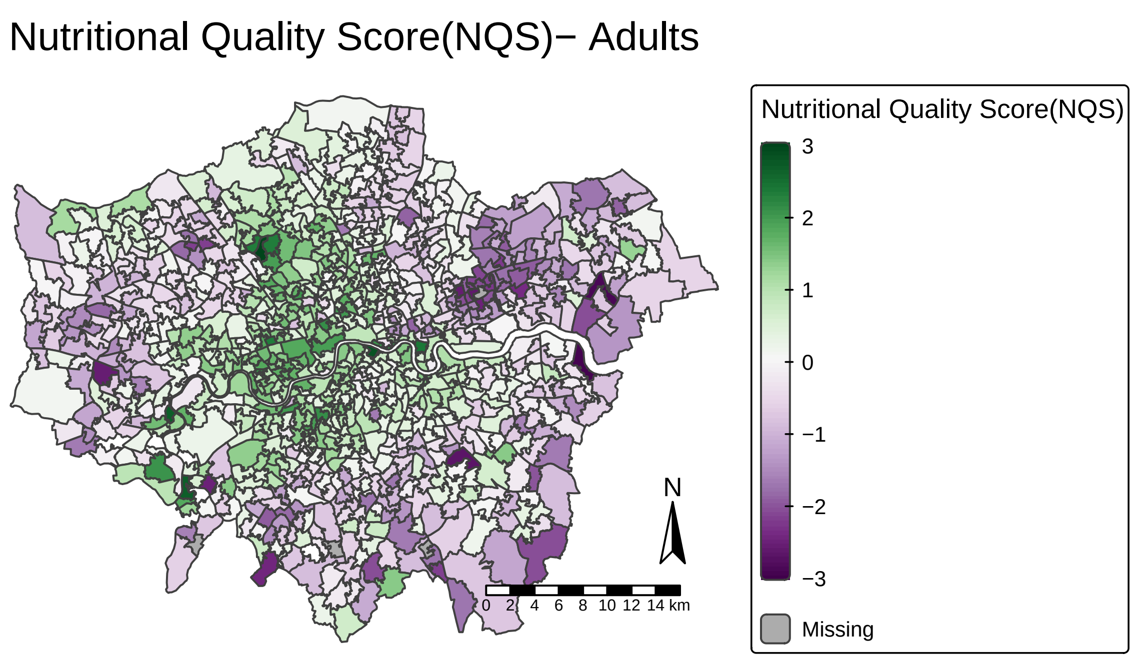
Low NQS (Purple): Concentrated in east and southeast London, possibly linked to affordability constraints, higher exposure to fast food outlets, or limited access to varied, nutritious foods. White to pale shades mark areas near the London average, while grey - Missing zones highlight MSOAs with insufficient data for calculation.

These patterns suggest children in some regions are disproportionately exposed to lower-quality diets, a public health concern given the long-term impacts on growth and wellbeing.

```r
tm_shape(tesco_and_msoas) +
  tm_polygons(
    fill = "z_NQS_adults",
    fill.legend = tm_legend(title = "Nutritional Quality Score(NQS)", reverse = TRUE),
    fill.scale = tm_scale_continuous(limits = c(-3,3),values = "brewer.PRGn")
  ) +
  tm_title("Nutritional Quality Score(NQS)- Adults") +
```

```r
  tm_compass() +
  tm_scalebar() +
  tm_layout(frame = FALSE, bg.color = "white")
```

```
## Values have been found that are lower than the lowest limit. These 'outliers' have been set to NA. U

## Variable(s) "fill" contains positive and negative values, so midpoint is set to 0. Set midpoint = NA

## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "brewer.PRGn" is
## named "prgn" (in long format "brewer.prgn")
## [plot mode] fit legend/component: Some legend items or map compoments do not
## fit well, and are therefore rescaled.
## i Set the tmap option `component.autoscale = FALSE` to disable rescaling.
```



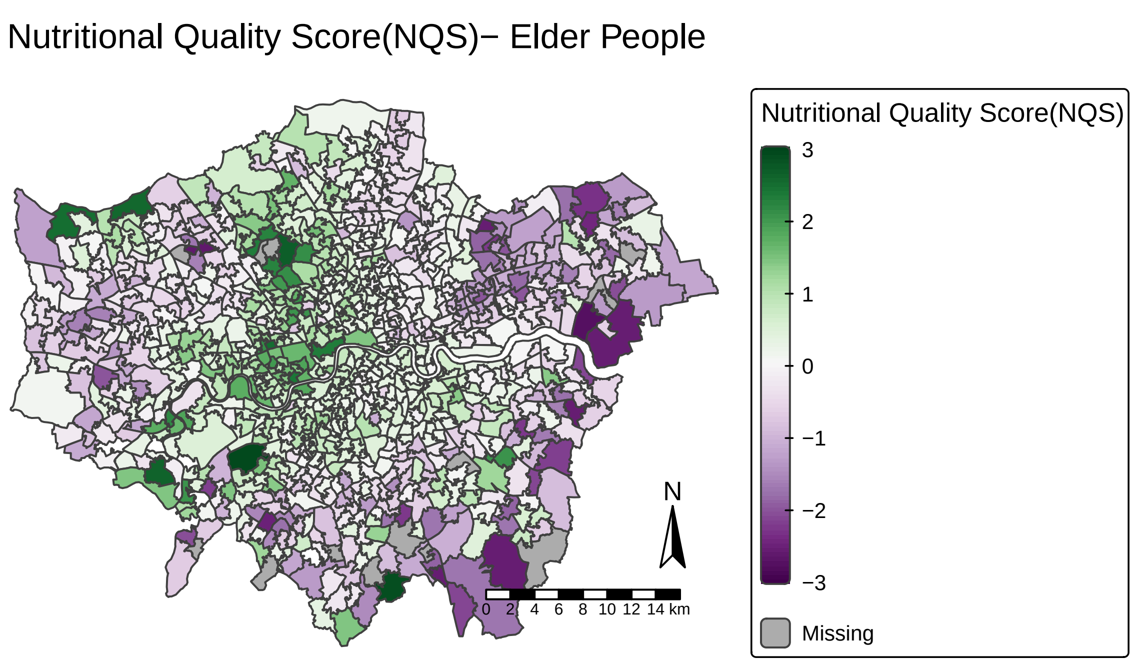
This map focuses on purchases weighted by the proportion of adults (18-64 years).

High NQS (dark green): Especially visible in central and northwestern MSOAs, where working-age adults appear to make healthier dietary choices. This could be influenced by higher disposable incomes, workplace wellness culture, or better supermarket provision.

Low NQS (Purple): Scattered pockets across outer boroughs, where adults may face affordability barriers or opt for convenience foods due to lifestyle demands. White to pale shades mark areas near the London average, while grey - Missing zones highlight MSOAs with insufficient data for calculation. Compared to children's NQS, adult scores tend to be higher overall, suggesting adults generally purchase healthier food than other age groups.

```r
tm_shape(tesco_and_msoas) +
  tm_polygons(
    fill = "z_NQS_elderly",
    fill.legend = tm_legend(title = "Nutritional Quality Score(NQS)", reverse = TRUE),
    fill.scale = tm_scale_continuous(limits = c(-3,3),values = "brewer.PRGn")
  ) +
  tm_title("Nutritional Quality Score(NQS)- Elder People") +
  tm_compass() +
  tm_scalebar() +
  tm_layout(frame = FALSE, bg.color = "white")
```

```
## Values have been found that are lower than the lowest limit. These 'outliers' have been set to NA. U

## Values have been found that are higher than the upper limit. These 'outliers' have been set to NA. U

## Variable(s) "fill" contains positive and negative values, so midpoint is set to 0. Set midpoint = NA

## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "brewer.PRGn" is
## named "prgn" (in long format "brewer.prgn")
## [plot mode] fit legend/component: Some legend items or map components do not
## fit well, and are therefore rescaled.
## i Set the tmap option `component.autoscale = FALSE` to disable rescaling.
```



This map weights the NQS by the proportion of elderly residents (65+ years).

High NQS (dark green): Found in certain central and northern areas, suggesting older residents there maintain healthier purchasing habits, possibly due to established dietary routines and access to quality food outlets.

Low NQS (Purple): More common in peripheral and some southern MSOAs, where older residents may face physical access challenges, fixed incomes, or reliance on less healthy nearby shops.

White to pale shades mark areas near the London average, while grey - Missing zones highlight MSOAs with insufficient data for calculation.

We can observe that several central and northern MSOAs demonstrate relatively better nutritional outcomes for the elderly, while some peripheral or southern areas show lower NQS values. This variation may reflect differences in access to nutritious food, socioeconomic status, or health awareness among older residents. Identifying such patterns is essential to target dietary interventions where older populations may be at higher nutritional risk.

After visualising the spatial distribution of the Nutritional Quality Score (NQS) using a choropleth map, the next step is to statistically assess whether the observed spatial patterns are random or exhibit clustering.

After mapping the Nutritional Quality Score (NQS), I tested whether the observed pattern is non-random—i.e., whether MSOAs with similar values cluster together geographically.

#Moran's I

```r
# We already have: tesco_and_msoas (sf with MSOA polygons + your NQS columns)
library(spdep)
```

```
## Warning: package 'spdep' was built under R version 4.4.3
```

```
## Loading required package: spData
```

```
## Warning: package 'spData' was built under R version 4.4.3
```

```
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```r
library(sf)
stopifnot(inherits(tesco_and_msoas, "sf"))

#repairing any invalid polygons present
if(any(!st_is_valid(tesco_and_msoas$geometry))){
  tesco_and_msoas<- st_make_valid(tesco_and_msoas)
}
```

I defined neighbours using queen contiguity (areas sharing a boundary or a corner are neighbours) and built row-standardised (W) spatial weights so each MSOA's neighbour weights sum to 1. This matches common practice for areal data and ensures each area's influence is comparable. I validated geometries and handled potential "island" units (no neighbours) via safe options in the code.

```r
nb<- poly2nb(tesco_and_msoas, queen = TRUE, snap = 1e-6)
nb
```

```
## Neighbour list object:
## Number of regions: 977
## Number of nonzero links: 5564
## Percentage nonzero weights: 0.5829053
## Average number of links: 5.694985
```

```
#define our neighbour matrix- how are our neighbours weighted?

wt_matrix <- nb2listw(nb, style = "W")
wt_matrix
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 977
## Number of nonzero links: 5564
## Percentage nonzero weights: 0.5829053
## Average number of links: 5.694985
##
## Weights style: W
## Weights constants summary:
##     n     nn  S0       S1      S2
## W 977 954529 977 357.2154 3992.14
```

```
morans_i_NQS <- moran.test(tesco_and_msoas$NQS, listw = wt_matrix)

morans_i_NQS_children <- moran.test(tesco_and_msoas$NQS_children, listw = wt_matrix)
morans_i_NQS_adults <- moran.test(tesco_and_msoas$NQS_adults, listw = wt_matrix)
morans_i_NQS_elderly <- moran.test(tesco_and_msoas$NQS_elderly, listw = wt_matrix)

collect <- function(obj, name) tibble(
  variable     = name,
  morans_I     = unname(obj$estimate[["Moran I statistic"]]),
  expectation  = unname(obj$estimate[["Expectation"]]),
  variance     = unname(obj$estimate[["Variance"]]),
  z_score      = unname(obj$statistic)
)

moran_table <- bind_rows(
  collect(morans_i_NQS,          "NQS"),
  collect(morans_i_NQS_children, "NQS_children"),
  collect(morans_i_NQS_adults,   "NQS_adults"),
  collect(morans_i_NQS_elderly,  "NQS_elderly")
) %>%
  mutate(across(where(is.numeric), ~ round(., 4)))

moran_table
```

```
## # A tibble: 4 x 5
##   variable     morans_I expectation variance z_score
##   <chr>           <dbl>       <dbl>    <dbl>   <dbl>
## 1 NQS             0.552      -0.001   0.0004    28.7
## 2 NQS_children    0.516      -0.001   0.0004    26.9
## 3 NQS_adults      0.568      -0.001   0.0004    29.5
## 4 NQS_elderly     0.475      -0.001   0.0004    24.8
```

I computed Global Moran's I for each variable:

Moran's I > 0 and statistically significant - positive spatial autocorrelation (clustering of similar values).

Moran's I 0 - spatial randomness.

Moran's I < 0 and significant - dispersion (checkerboard-like, rare in socio-spatial settings).

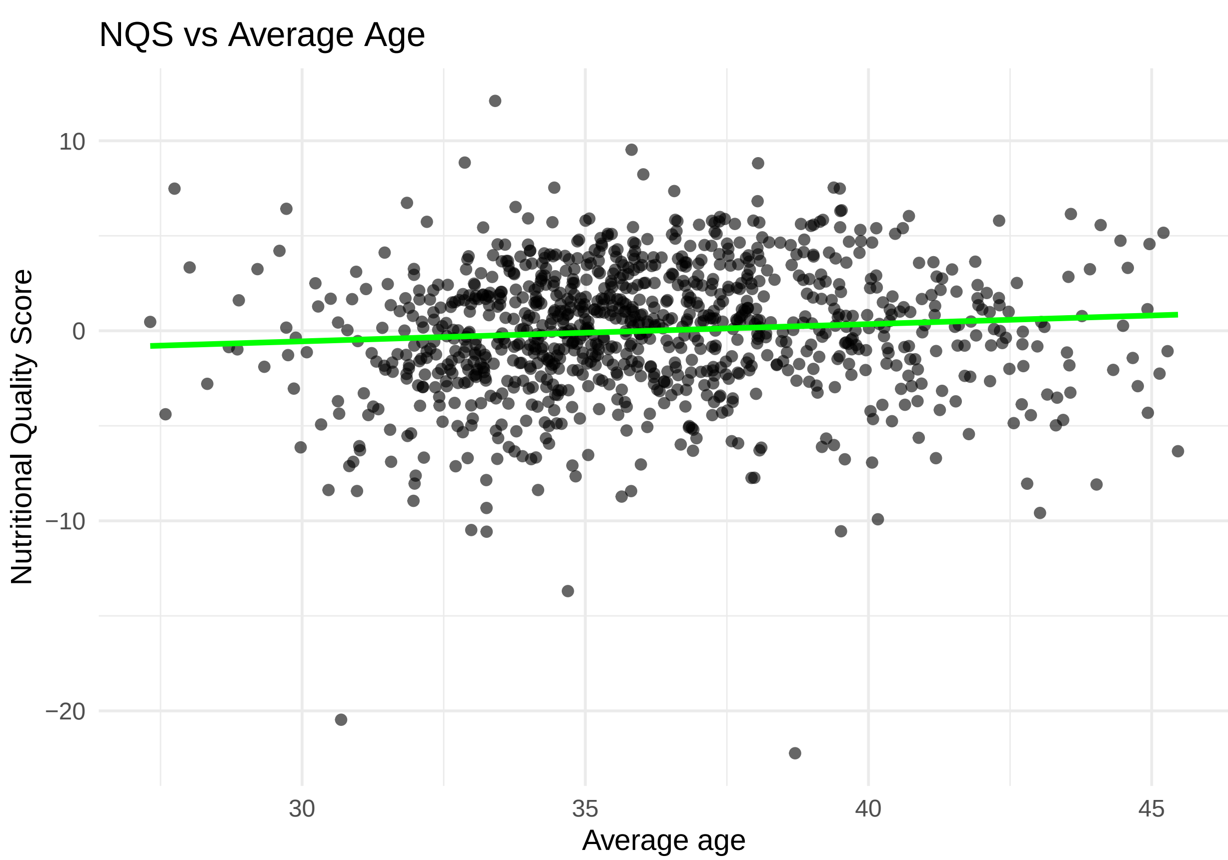
All are well above 0, which indicates positive spatial autocorrelation: nearby MSOAs tend to have similar values (high with high, low with low) rather than being randomly scattered.

Given magnitudes around 0.48–0.57, these will almost certainly be statistically significant

The Tesco dataset alone provides detailed information on food purchases and allows the construction of a Nutritional Quality Score (NQS). However, it does not capture the broader social, economic, and health context of each MSOA. To address this, I integrated an additional MSOA-level dataset containing demographic, health, and deprivation indicators. This enabled the analysis to move beyond mapping diet quality towards explaining the underlying drivers of nutritional inequality, such as income deprivation, obesity prevalence, and household structure. The inclusion of this dataset therefore strengthens the study by linking dietary patterns to wider social determinants of health.

```
ggplot(tesco_and_msoas, aes(x = avg_age, y = NQS)) +
  geom_point(alpha = 0.6, color = "black") +
  geom_smooth(method = "lm", se = FALSE, color = "green") +
  labs(title = "NQS vs Average Age", x = "Average age", y = "Nutritional Quality Score") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
# reading new dataset
df <- read_csv("new_data.csv", show_col_types = FALSE) %>%

  remove_empty(c("rows","cols"))
```

```
## New names:
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...11`
## * `` -> `...12`
## * `` -> `...14`
## * `` -> `...15`
## * `` -> `...16`
## * `` -> `...17`
```

```r
col_names <- c("geog","msoa_code","msoa_name","la_code","la_name","reception_total_measured","reception_

colnames(df) <- col_names

df <- df[-c(1,2,3), ]

to_numeric_percent <- function(x) {
  x <- as.character(x)
  x <- str_replace_all(x, "%", "")       # remove % sign
  x <- str_replace_all(x, "[^0-9.\\-]", "") # remove other symbols
  suppressWarnings(as.numeric(x))
}

pct_cols <- c("reception_excess_percent", "reception_excess_lower_95cl", "reception_excess_upper_95cl",

              "year6_excess_percent",     "year6_excess_lower_95cl",     "year6_excess_upper_95cl")
df[pct_cols] <- lapply(df[pct_cols], to_numeric_percent)
count_cols <- c(
  "reception_total_measured","reception_excess_count",
  "year6_total_measured","year6_excess_count"
)
df[count_cols] <- lapply(df[count_cols], function(x) as.numeric(as.character(x)))
```

```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

```r
df
```

```
## # A tibble: 983 x 15
##    geog  msoa_code msoa_name              la_code la_name reception_total_meas~1
##    <chr> <chr>     <chr>                  <chr>   <chr>                    <dbl>
```

```
##  1 MSOA  E02000001 City of London 001     E09000~ City o~                    81
##  2 MSOA  E02000002 Barking and Dagenham ~ E09000~ Barkin~                   359
##  3 MSOA  E02000003 Barking and Dagenham ~ E09000~ Barkin~                   401
##  4 MSOA  E02000004 Barking and Dagenham ~ E09000~ Barkin~                   208
##  5 MSOA  E02000005 Barking and Dagenham ~ E09000~ Barkin~                   475
##  6 MSOA  E02000007 Barking and Dagenham ~ E09000~ Barkin~                   502
##  7 MSOA  E02000008 Barking and Dagenham ~ E09000~ Barkin~                   657
##  8 MSOA  E02000009 Barking and Dagenham ~ E09000~ Barkin~                   502
##  9 MSOA  E02000010 Barking and Dagenham ~ E09000~ Barkin~                   512
## 10 MSOA  E02000011 Barking and Dagenham ~ E09000~ Barkin~                   295
## # i 973 more rows
## # i abbreviated name: 1: reception_total_measured
## # i 9 more variables: reception_excess_count <dbl>,
## #   reception_excess_percent <dbl>, reception_excess_lower_95cl <dbl>,
## #   reception_excess_upper_95cl <dbl>, year6_total_measured <dbl>,
## #   year6_excess_count <dbl>, year6_excess_percent <dbl>,
## #   year6_excess_lower_95cl <dbl>, year6_excess_upper_95cl <dbl>
```

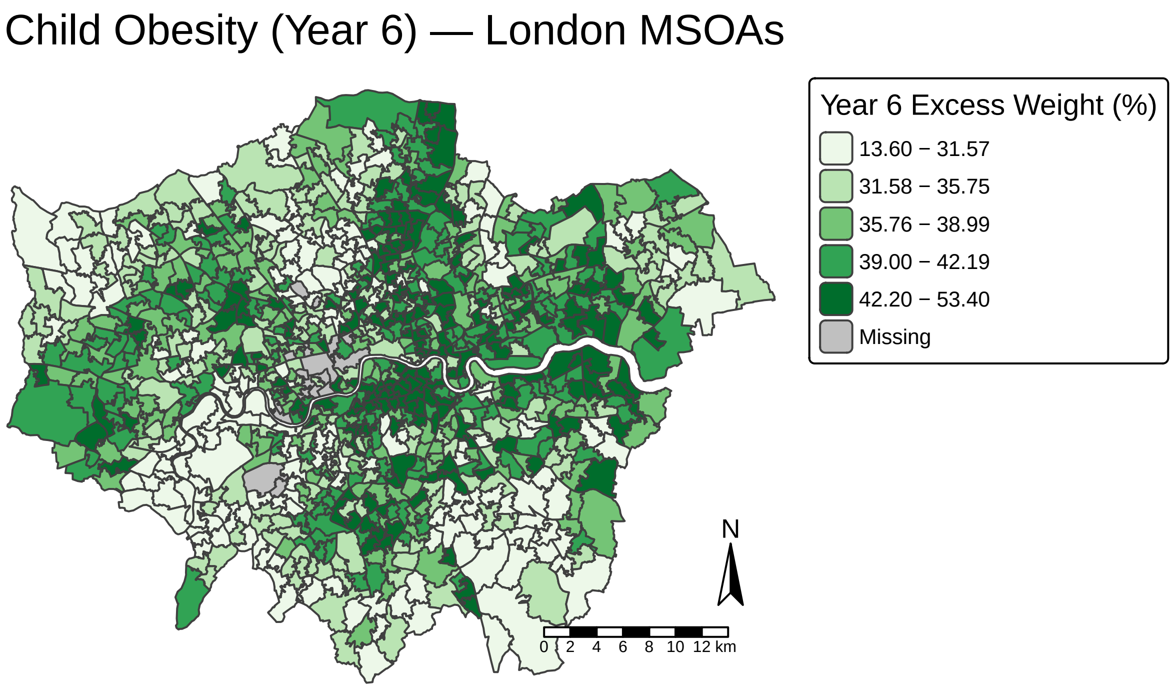
```r
tesco <- tesco %>% mutate(area_id = as.character(area_id))
tesco_joined <- tesco %>%
  left_join(df, by = c("area_id" = "msoa_code"))
```

###Visualising the obesity in children

```r
child <- df %>%
  transmute(
    msoa_code = as.character(msoa_code),
    reception_excess_percent = as.numeric(reception_excess_percent),
    year6_excess_percent     = as.numeric(year6_excess_percent)
  )

# 2) Join children data to polygons
obesity_msoa <- msoas %>%
  left_join(child, by = "msoa_code")
#3) map

tm_shape(obesity_msoa) +
  tm_polygons(
    fill = "year6_excess_percent",
    palette = "brewer.greens",
    style = "quantile", n = 5,
    title = "Year 6 Excess Weight (%)"
  ) +
  tm_title("Child Obesity (Year 6) - London MSOAs") +
  tm_compass() +
  tm_scale_bar() +
  tm_layout(frame = FALSE, bg.color = "white")
```

```
##

## -- tmap v3 code detected -------------------------------------------------------

## [v3->v4] `tm_polygons()`: instead of `style = "quantile"`, use fill.scale =
```

```
## `tm_scale_intervals()`.
## i Migrate the argument(s) 'style', 'n', 'palette' (rename to 'values') to
##   'tm_scale_intervals(<HERE>)'
## [v3->v4] `tm_polygons()`: migrate the argument(s) related to the legend of the
## visual variable `fill` namely 'title' to 'fill.legend = tm_legend(<HERE>)'
## ! `tm_scale_bar()` is deprecated. Please use `tm_scalebar()` instead.
```



The map above illustrates the spatial distribution of excess weight among Year 6 children (ages 10–11) across Middle Layer Super Output Areas (MSOAs) in Greater London. The data represents the percentage of children classified as overweight or obese based on the National Child Measurement Programme (NCMP).

Each MSOA is colour-coded to show the level of excess weight prevalence:

Light green areas (13.6–31.5%) indicate lower rates of child obesity.

Medium green shades (31.6–39%) represent moderate prevalence levels.

Dark green zones (39–53%) highlight areas with high rates of obesity among Year 6 children.

Grey areas correspond to missing or unavailable data.

```r
# CLEANING MSOA INCOME DATASET

# 1. Read the dataset
income_data <- read_csv("msoa-data.csv", show_col_types = FALSE)
```

```
## New names:
## * `House Prices;Sales;2011` -> `House Prices;Sales;2011...129`
## * `House Prices;Sales;2011` -> `House Prices;Sales;2011...130`
```

```r
# 2. Remove empty rows and columns
income_data <- income_data %>% remove_empty(c("rows", "cols"))

# 3. Clean column names (lowercase + underscores)
income_data <- income_data %>% clean_names()
```

```
## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8
```

```
## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8
```

```
## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 114 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 115 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 116 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 117 is invalid UTF-8

## Warning in grepl(x = string, pattern = current_unicode, fixed = TRUE): input
## string 118 is invalid UTF-8
```

```r
# 4. View column names to identify key fields
names(income_data)
```

```
##  [1] "middle_super_output_area"
##  [2] "msoa_name"
##  [3] "age_structure_2011_census_all_ages"
##  [4] "age_structure_2011_census_0_15"
##  [5] "age_structure_2011_census_16_29"
##  [6] "age_structure_2011_census_30_44"
##  [7] "age_structure_2011_census_45_64"
##  [8] "age_structure_2011_census_65"
##  [9] "age_structure_2011_census_working_age"
## [10] "mid_year_estimate_totals_all_ages_2002"
## [11] "mid_year_estimate_totals_all_ages_2003"
## [12] "mid_year_estimate_totals_all_ages_2004"
## [13] "mid_year_estimate_totals_all_ages_2005"
## [14] "mid_year_estimate_totals_all_ages_2006"
## [15] "mid_year_estimate_totals_all_ages_2007"
## [16] "mid_year_estimate_totals_all_ages_2008"
## [17] "mid_year_estimate_totals_all_ages_2009"
## [18] "mid_year_estimate_totals_all_ages_2010"
## [19] "mid_year_estimate_totals_all_ages_2011"
## [20] "mid_year_estimate_totals_all_ages_2012"
## [21] "mid_year_estimates_2012_by_age_percent_0_to_14"
## [22] "mid_year_estimates_2012_by_age_percent_15_64"
## [23] "mid_year_estimates_2012_by_age_percent_65"
## [24] "mid_year_estimates_2012_by_age_0_4"
## [25] "mid_year_estimates_2012_by_age_5_9"
## [26] "mid_year_estimates_2012_by_age_10_14"
## [27] "mid_year_estimates_2012_by_age_15_19"
## [28] "mid_year_estimates_2012_by_age_20_24"
## [29] "mid_year_estimates_2012_by_age_25_29"
## [30] "mid_year_estimates_2012_by_age_30_34"
## [31] "mid_year_estimates_2012_by_age_35_39"
## [32] "mid_year_estimates_2012_by_age_40_44"
## [33] "mid_year_estimates_2012_by_age_45_49"
## [34] "mid_year_estimates_2012_by_age_50_54"
## [35] "mid_year_estimates_2012_by_age_55_59"
## [36] "mid_year_estimates_2012_by_age_60_64"
## [37] "mid_year_estimates_2012_by_age_65_69"
## [38] "mid_year_estimates_2012_by_age_70_74"
## [39] "mid_year_estimates_2012_by_age_75_79"
## [40] "mid_year_estimates_2012_by_age_80_84"
## [41] "mid_year_estimates_2012_by_age_85_89"
## [42] "mid_year_estimates_2012_by_age_90"
## [43] "households_2011_all_households"
## [44] "household_composition_2011_numbers_couple_household_with_dependent_children"
## [45] "household_composition_2011_numbers_couple_household_without_dependent_children"
## [46] "household_composition_2011_numbers_lone_parent_household"
## [47] "household_composition_2011_numbers_one_person_household"
## [48] "household_composition_2011_numbers_other_household_types"
## [49] "household_composition_2011_percentages_couple_household_with_dependent_children"
## [50] "household_composition_2011_percentages_couple_household_without_dependent_children"
```

```
## [51] "household_composition_2011_percentages_lone_parent_household"
## [52] "household_composition_2011_percentages_one_person_household"
## [53] "household_composition_2011_percentages_other_household_types"
## [54] "ethnic_group_2011_census_white"
## [55] "ethnic_group_2011_census_mixed_multiple_ethnic_groups"
## [56] "ethnic_group_2011_census_asian_asian_british"
## [57] "ethnic_group_2011_census_black_african_caribbean_black_british"
## [58] "ethnic_group_2011_census_other_ethnic_group"
## [59] "ethnic_group_2011_census_bame"
## [60] "ethnic_group_2011_census_white_percent"
## [61] "ethnic_group_2011_census_mixed_multiple_ethnic_groups_percent"
## [62] "ethnic_group_2011_census_asian_asian_british_percent"
## [63] "ethnic_group_2011_census_black_african_caribbean_black_british_percent"
## [64] "ethnic_group_2011_census_other_ethnic_group_percent"
## [65] "ethnic_group_2011_census_bame_percent"
## [66] "country_of_birth_2011_united_kingdom"
## [67] "country_of_birth_2011_not_united_kingdom"
## [68] "country_of_birth_2011_united_kingdom_percent"
## [69] "country_of_birth_2011_not_united_kingdom_percent"
## [70] "household_language_2011_at_least_one_person_aged_16_and_over_in_household_has_english_as_a_mai
## [71] "household_language_2011_no_people_in_household_have_english_as_a_main_language"
## [72] "household_language_2011_percent_of_people_aged_16_and_over_in_household_have_english_as_a_mai
## [73] "household_language_2011_percent_of_households_where_no_people_in_household_have_english_as_a_m
## [74] "religion_2011_christian"
## [75] "religion_2011_buddhist"
## [76] "religion_2011_hindu"
## [77] "religion_2011_jewish"
## [78] "religion_2011_muslim"
## [79] "religion_2011_sikh"
## [80] "religion_2011_other_religion"
## [81] "religion_2011_no_religion"
## [82] "religion_2011_religion_not_stated"
## [83] "religion_2011_christian_percent"
## [84] "religion_2011_buddhist_percent"
## [85] "religion_2011_hindu_percent"
## [86] "religion_2011_jewish_percent"
## [87] "religion_2011_muslim_percent"
## [88] "religion_2011_sikh_percent"
## [89] "religion_2011_other_religion_percent"
## [90] "religion_2011_no_religion_percent"
## [91] "religion_2011_religion_not_stated_percent"
## [92] "tenure_2011_owned_owned_outright"
## [93] "tenure_2011_owned_owned_with_a_mortgage_or_loan"
## [94] "tenure_2011_social_rented"
## [95] "tenure_2011_private_rented"
## [96] "tenure_2011_owned_owned_outright_percent"
## [97] "tenure_2011_owned_owned_with_a_mortgage_or_loan_percent"
## [98] "tenure_2011_social_rented_percent"
## [99] "tenure_2011_private_rented_percent"
## [100] "dwelling_type_2011_household_spaces_with_at_least_one_usual_resident"
## [101] "dwelling_type_2011_household_spaces_with_no_usual_residents"
## [102] "dwelling_type_2011_whole_house_or_bungalow_detached"
## [103] "dwelling_type_2011_whole_house_or_bungalow_semi_detached"
## [104] "dwelling_type_2011_whole_house_or_bungalow_terraced_including_end_terrace"
```

```
## [105] "dwelling_type_2011_flat_maisonette_or_apartment"
## [106] "dwelling_type_2011_household_spaces_with_at_least_one_usual_resident_percent"
## [107] "dwelling_type_2011_household_spaces_with_no_usual_residents_percent"
## [108] "dwelling_type_2011_whole_house_or_bungalow_detached_percent"
## [109] "dwelling_type_2011_whole_house_or_bungalow_semi_detached_percent"
## [110] "dwelling_type_2011_whole_house_or_bungalow_terraced_including_end_terrace_percent"
## [111] "dwelling_type_2011_flat_maisonette_or_apartment_percent"
## [112] "land_area_hectares"
## [113] "population_density_persons_per_hectare_2012"
## [114] "house_prices_median_house_price_2005"
## [115] "house_prices_median_house_price_2006"
## [116] "house_prices_median_house_price_2007"
## [117] "house_prices_median_house_price_2008"
## [118] "house_prices_median_house_price_2009"
## [119] "house_prices_median_house_price_2010"
## [120] "house_prices_median_house_price_2011"
## [121] "house_prices_median_house_price_2012"
## [122] "house_prices_median_house_price_2013_p"
## [123] "house_prices_sales_2005"
## [124] "house_prices_sales_2006"
## [125] "house_prices_sales_2007"
## [126] "house_prices_sales_2008"
## [127] "house_prices_sales_2009"
## [128] "house_prices_sales_2010"
## [129] "house_prices_sales_2011_129"
## [130] "house_prices_sales_2011_130"
## [131] "house_prices_sales_2013_p"
## [132] "qualifications_2011_census_no_qualifications"
## [133] "qualifications_2011_census_highest_level_of_qualification_level_1_qualifications"
## [134] "qualifications_2011_census_highest_level_of_qualification_level_2_qualifications"
## [135] "qualifications_2011_census_highest_level_of_qualification_apprenticeship"
## [136] "qualifications_2011_census_highest_level_of_qualification_level_3_qualifications"
## [137] "qualifications_2011_census_highest_level_of_qualification_level_4_qualifications_and_above"
## [138] "qualifications_2011_census_highest_level_of_qualification_other_qualifications"
## [139] "qualifications_2011_census_schoolchildren_and_full_time_students_age_18_and_over"
## [140] "economic_activity_2011_census_economically_active_total"
## [141] "economic_activity_2011_census_economically_active_unemployed"
## [142] "economic_activity_2011_census_economically_inactive_total"
## [143] "economic_activity_2011_census_economically_active_percent"
## [144] "economic_activity_2011_census_unemployment_rate"
## [145] "economic_activity_2011_census_economically_inactive_percent"
## [146] "adults_in_employment_2011_census_no_adults_in_employment_in_household_with_dependent_children
## [147] "adults_in_employment_2011_census_percent_of_households_with_no_adults_in_employment_with_depen
## [148] "household_income_estimates_2011_12_total_mean_annual_household_income"
## [149] "household_income_estimates_2011_12_total_median_annual_household_income"
## [150] "income_deprivation_2010_percent_living_in_income_deprived_households_reliant_on_means_tested_b
## [151] "income_deprivation_2010_percent_of_people_aged_over_60_who_live_in_pension_credit_households"
## [152] "lone_parents_2011_census_all_lone_parent_housholds_with_dependent_children"
## [153] "lone_parents_2011_census_lone_parents_not_in_employment"
## [154] "lone_parents_2011_census_lone_parent_not_in_employment_percent"
## [155] "central_heating_2011_census_households_with_central_heating_percent"
## [156] "health_2011_census_day_to_day_activities_limited_a_lot"
## [157] "health_2011_census_day_to_day_activities_limited_a_little"
## [158] "health_2011_census_day_to_day_activities_not_limited"
```

```
## [159] "health_2011_census_day_to_day_activities_limited_a_lot_percent"
## [160] "health_2011_census_day_to_day_activities_limited_a_little_percent"
## [161] "health_2011_census_day_to_day_activities_not_limited_percent"
## [162] "health_2011_census_very_good_health"
## [163] "health_2011_census_good_health"
## [164] "health_2011_census_fair_health"
## [165] "health_2011_census_bad_health"
## [166] "health_2011_census_very_bad_health"
## [167] "health_2011_census_very_good_health_percent"
## [168] "health_2011_census_good_health_percent"
## [169] "health_2011_census_fair_health_percent"
## [170] "health_2011_census_bad_health_percent"
## [171] "health_2011_census_very_bad_health_percent"
## [172] "low_birth_weight_births_2007_2011_low_birth_weight_births_percent"
## [173] "low_birth_weight_births_2007_2011_lcl_lower_confidence_limit"
## [174] "low_birth_weight_births_2007_2011_ucl_upper_confidence_limit"
## [175] "obesity_percent_of_measured_children_in_year_6_who_were_classified_as_obese_2009_10_2011_12"
## [176] "obesity_percentage_of_the_population_aged_16_with_a_bmi_of_30_modelled_estimate_2006_2008"
## [177] "incidence_of_cancer_all"
## [178] "incidence_of_cancer_breast_cancer"
## [179] "incidence_of_cancer_colorectal_cancer"
## [180] "incidence_of_cancer_lung_cancer"
## [181] "incidence_of_cancer_prostate_cancer"
## [182] "life_expectancy_males"
## [183] "life_expectancy_females"
## [184] "car_or_van_availability_2011_census_no_cars_or_vans_in_household"
## [185] "car_or_van_availability_2011_census_1_car_or_van_in_household"
## [186] "car_or_van_availability_2011_census_2_cars_or_vans_in_household"
## [187] "car_or_van_availability_2011_census_3_cars_or_vans_in_household"
## [188] "car_or_van_availability_2011_census_4_or_more_cars_or_vans_in_household"
## [189] "car_or_van_availability_2011_census_sum_of_all_cars_or_vans_in_the_area"
## [190] "car_or_van_availability_2011_census_no_cars_or_vans_in_household_percent"
## [191] "car_or_van_availability_2011_census_1_car_or_van_in_household_percent"
## [192] "car_or_van_availability_2011_census_2_cars_or_vans_in_household_percent"
## [193] "car_or_van_availability_2011_census_3_cars_or_vans_in_household_percent"
## [194] "car_or_van_availability_2011_census_4_or_more_cars_or_vans_in_household_percent"
## [195] "car_or_van_availability_2011_census_cars_per_household"
## [196] "road_casualties_2010_fatal"
## [197] "road_casualties_2010_serious"
## [198] "road_casualties_2010_slight"
## [199] "road_casualties_2010_2010_total"
## [200] "road_casualties_2011_fatal"
## [201] "road_casualties_2011_serious"
## [202] "road_casualties_2011_slight"
## [203] "road_casualties_2011_2011_total"
## [204] "road_casualties_2012_fatal"
## [205] "road_casualties_2012_serious"
## [206] "road_casualties_2012_slight"
## [207] "road_casualties_2012_2012_total"
```

```r
income_data <- income_data%>% rename(household_mean_income = household_income_estimates_2011_12_total_me
# 5. Keep only relevant columns (update if your names differ)
income_data <- income_data %>%
  select(msoa_name, household_mean_income)
```

```r
# 7. Final cleaned dataset
glimpse(income_data)
```

```
## Rows: 984
## Columns: 2
## $ msoa_name             <chr> "City of London 001", "Barking and Dagenham 001"~
## $ household_mean_income <dbl> 59728, 31788, 43357, 46701, 34294, 29976, 31413,~
```

```r
summary(income_data$household_mean_income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   22367   34137   41536   45804   53875  141363
```

```r
head(income_data)
```

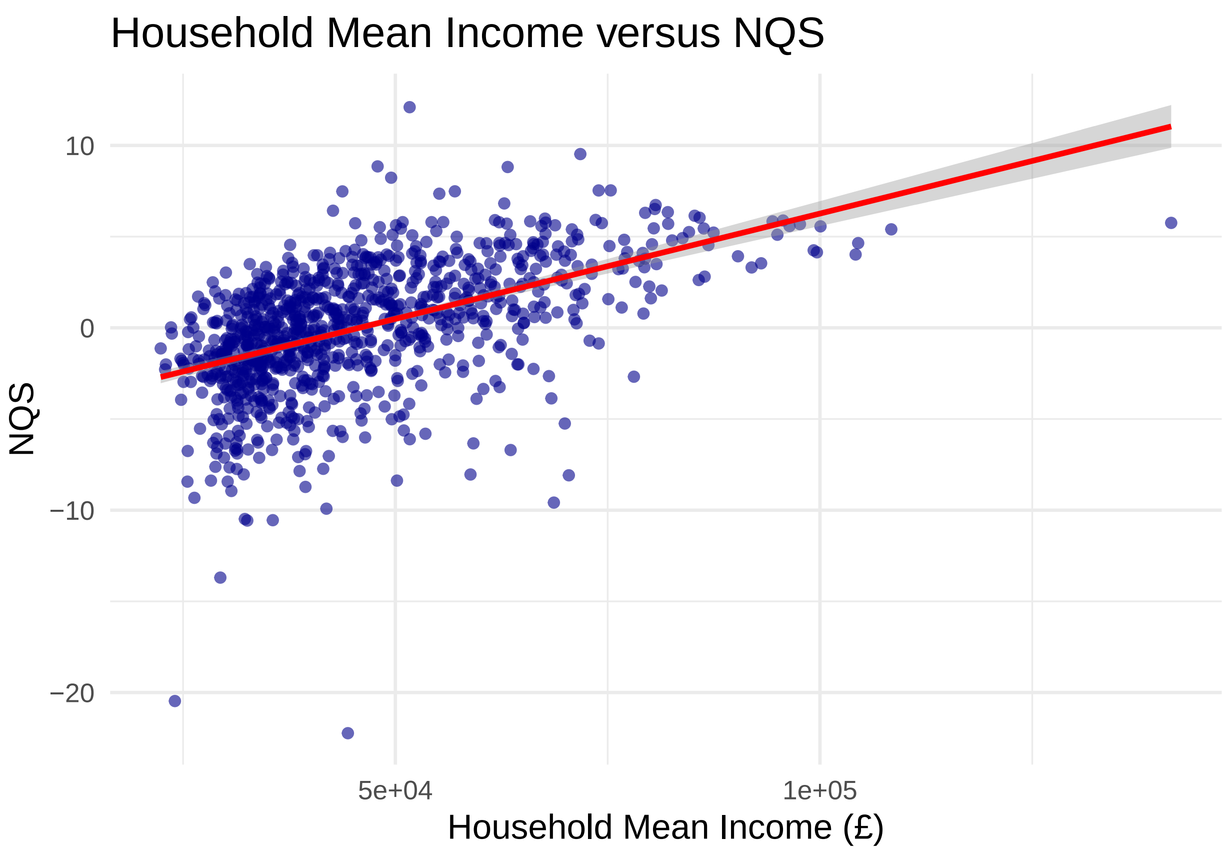
```
## # A tibble: 6 x 2
##   msoa_name                household_mean_income
##   <chr>                                    <dbl>
## 1 City of London 001                       59728
## 2 Barking and Dagenham 001                 31788
## 3 Barking and Dagenham 002                 43357
## 4 Barking and Dagenham 003                 46701
## 5 Barking and Dagenham 004                 34294
## 6 Barking and Dagenham 006                 29976
```

```r
# Joining the datset with tesco joined
final_joined <- tesco_joined %>%
  left_join(income_data, by = "msoa_name")
final_joined
```

```
## # A tibble: 977 x 232
##    area_id   weight weight_perc2.5 weight_perc25 weight_perc50 weight_perc75
##    <chr>      <dbl>          <dbl>         <dbl>         <dbl>         <dbl>
##  1 E02000001   315.           37             150           250           400
##  2 E02000002   415.           32.5           160           300           500
##  3 E02000003   421.           32.5           154           320           500
##  4 E02000004   349.           30             125           265           450
##  5 E02000005   429.           30             157           300           500
##  6 E02000007   388.           32.5           150           300           500
##  7 E02000008   399.           32.5           151           300           500
##  8 E02000009   399.           30             150           300           500
##  9 E02000010   409.           32.5           150           300           494
## 10 E02000011   385.           25             150           300           470
## # i 967 more rows
## # i 226 more variables: weight_perc97.5 <dbl>, weight_std <dbl>,
## #   weight_ci95 <dbl>, volume <dbl>, volume_perc2.5 <dbl>, volume_perc25 <dbl>,
## #   volume_perc50 <dbl>, volume_perc75 <dbl>, volume_perc97.5 <dbl>,
## #   volume_std <dbl>, volume_ci95 <dbl>, fat <dbl>, fat_perc2.5 <dbl>,
## #   fat_perc25 <dbl>, fat_perc50 <dbl>, fat_perc75 <dbl>, fat_perc97.5 <dbl>,
## #   fat_std <dbl>, fat_ci95 <dbl>, saturate <dbl>, saturate_perc2.5 <dbl>, ...
```

```r
ggplot(final_joined, aes(x = household_mean_income, y = NQS)) +
  geom_point(alpha = 0.6, color = "darkblue") +
  geom_smooth(method = "lm", se = TRUE, color = "red", linewidth = 1) +
  labs(
    title = "Household Mean Income versus NQS",
    x = "Household Mean Income (£)",
    y = "NQS"
  ) +
  theme_minimal(base_size = 13)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



The scatterplot shows a clear positive linear trend between household mean income and Nutritional Quality Score (NQS) across London MSOAs. This indicates that higher-income regions tend to purchase healthier, more nutrient-rich foods, whereas lower-income areas are associated with lower NQS values and poorer dietary quality.

A multiple linear regression model was used to assess how the Nutritional Quality Score (NQS) varies with demographic and socioeconomic factors across London's MSOAs. The predictors included the proportion of residents aged 0–17 years (as an indicator of child population density) and mean household income. This approach allowed quantifying how age structure and income levels relate to dietary quality, while controlling for their simultaneous effects.

```r
library(broom)
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.4.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.4.3

##
## Attaching package: 'zoo'

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

```r
library(sandwich)
```

```
## Warning: package 'sandwich' was built under R version 4.4.3
```

```r
# Ensure your data is ready
model_data <- final_joined %>%
  mutate(
    prop_children = ifelse(population > 0, age_0_17 / population, NA_real_),
    NQS = as.numeric(NQS),
    income = as.numeric(household_mean_income)
  ) %>%
  select(NQS, prop_children, income) %>%
  na.omit()

# Fit the multiple linear regression model
model <- lm(NQS ~ prop_children + income, data = model_data)

# Summary with robust SEs (HC3)
summary(model)
```

```
##
## Call:
## lm(formula = NQS ~ prop_children + income, data = model_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.1428  -1.3229   0.3448   1.7360  13.7722
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.849e+00  7.776e-01   2.379   0.0176 *
## prop_children -2.412e+01  2.448e+00  -9.855   <2e-16 ***
## income         7.799e-05  7.002e-06  11.137   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.831 on 974 degrees of freedom
## Multiple R-squared:  0.3315, Adjusted R-squared:  0.3302
## F-statistic: 241.5 on 2 and 974 DF,  p-value: < 2.2e-16
```

A multiple linear regression showed that MSOAs with a larger share of children (0–17) have lower Nutritional Quality Scores, while areas with higher household mean income have higher NQS. Quantitatively, a 10-percentage-point increase in the child population is associated with a 2.4-point decrease in NQS, whereas a £10,000 increase in mean household income is associated with a 0.78-point increase in NQS. Both predictors are statistically significant ($p < 0.001$). The model explains ~33% of the spatial variation in NQS.

```r
# Clean table for report
tidy(model, conf.int = TRUE, vcov = vcovHC(model, type = "HC3"))
```

```
## # A tibble: 3 x 7
##   term           estimate  std.error statistic  p.value    conf.low  conf.high
##   <chr>             <dbl>      <dbl>     <dbl>    <dbl>       <dbl>      <dbl>
## 1 (Intercept)     1.85      0.778         2.38 1.76e- 2   0.324        3.38e+0
## 2 prop_children -24.1       2.45         -9.86 6.60e-22 -28.9         -1.93e+1
## 3 income          0.0000780 0.00000700   11.1  3.34e-27   0.0000642    9.17e-5
```

This modelling strengthens the findings of earlier mapping and spatial analysis. The negative effect of child population share aligns with the higher obesity rates among age 10- 11 years of children, as visualised in the obesity map. Areas with more children and lower average incomes often overlap with zones of low NQS and high excess-weight prevalence, highlighting nutritional vulnerability among young populations.

Income, in contrast, exerts a protective effect, where better-off households are more able to access or choose higher-quality food purchases. Together, these results demonstrate that nutritional inequality is not random but socially structured related to income disparities and demographic composition.

```r
# Prepare modelling data
model_data <- tesco_joined %>%
  select(area_id, NQS, year6_excess_percent) %>%
  mutate(
    NQS = as.numeric(NQS),
    year6_excess_percent = as.numeric(year6_excess_percent)
  ) %>%
  drop_na()
```

To examine whether nutritional quality is linked to childhood obesity, I fitted a simple linear regression model with Year 6 excess-weight (%) as the outcome and the Nutritional Quality Score (NQS) as the predictor.

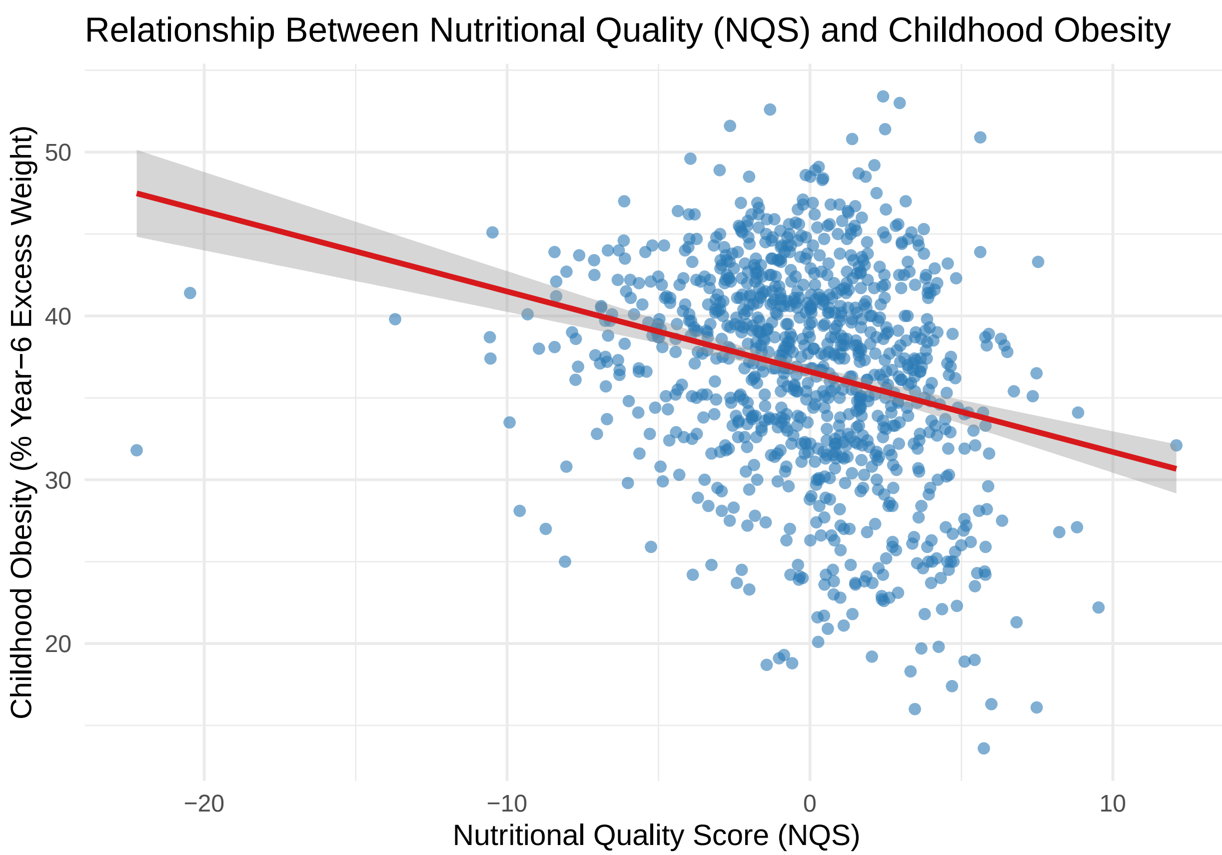
```r
# Fit a linear regression model
nqs_obesity_model <- lm(year6_excess_percent ~ NQS, data = model_data)
summary(nqs_obesity_model)
```

```
##
## Call:
## lm(formula = year6_excess_percent ~ NQS, data = model_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.1806  -3.9415   0.7547   4.3508  17.9897
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.59307    0.20605 177.594  < 2e-16 ***
## NQS         -0.48994    0.06036  -8.118 1.45e-15 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.378 on 957 degrees of freedom
## Multiple R-squared:  0.06442,    Adjusted R-squared:  0.06344
## F-statistic:  65.9 on 1 and 957 DF,  p-value: 1.454e-15
```

```r
# Visualise the relationship
ggplot(model_data, aes(x = NQS, y = year6_excess_percent)) +
  geom_point(alpha = 0.6, color = "#2C7BB6") +
  geom_smooth(method = "lm", se = TRUE, color = "#D7191C") +
  labs(
    title = "Relationship Between Nutritional Quality (NQS) and Childhood Obesity",
    x = "Nutritional Quality Score (NQS)",
    y = "Childhood Obesity (% Year-6 Excess Weight)"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



This scatterplot shows that neighbourhoods with lower NQS-meaning people are buying less healthy foods-also tend to have higher obesity rates among Year 6 children. As NQS increases, obesity rates generally fall. In other words, places where families purchase better-quality foods are also the places where fewer children are struggling with excess weight. This suggests a meaningful connection between the overall diet quality in an area and childhood health outcomes.