Untitled

workshop

2025-06-13

```
 pacman::p_load(tidyverse, dplyr, readr, here, sf, tmap)
```

```
hello <- read_csv("year_osward_grocery.csv")  %>%
  rename(ward_code = area_id)
```

```
## Rows: 638 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.
```

```
hello
```

```
## # A tibble: 638 x 202
##    ward_code weight weight_perc2.5 weight_perc25 weight_perc50 weight_perc75
##    <chr>      <dbl>          <dbl>         <dbl>         <dbl>         <dbl>
##  1 E05000026   450.           32.5          166.           300           500
##  2 E05000027   413.           32.5          150            300           500
##  3 E05000028   407.           32.5          160            300           500
##  4 E05000029   384.           30            150            250           454
##  5 E05000030   357.           30            140            250           450
##  6 E05000031   431.           37            176            320           500
##  7 E05000032   442.           35            175            325           500
##  8 E05000033   428.           32.5          170            320           500
##  9 E05000034   392.           32.5          150            300           500
## 10 E05000035   429.           37.5          173.           305           500
## # i 628 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>, ...
```

```
wards<- st_read(here("London-wards-2018/London-wards-2018_ESRI/London_Ward_CityMerged.shp"))  %>%
 rename(ward_code = GSS_CODE, ward_name = NAME)
```

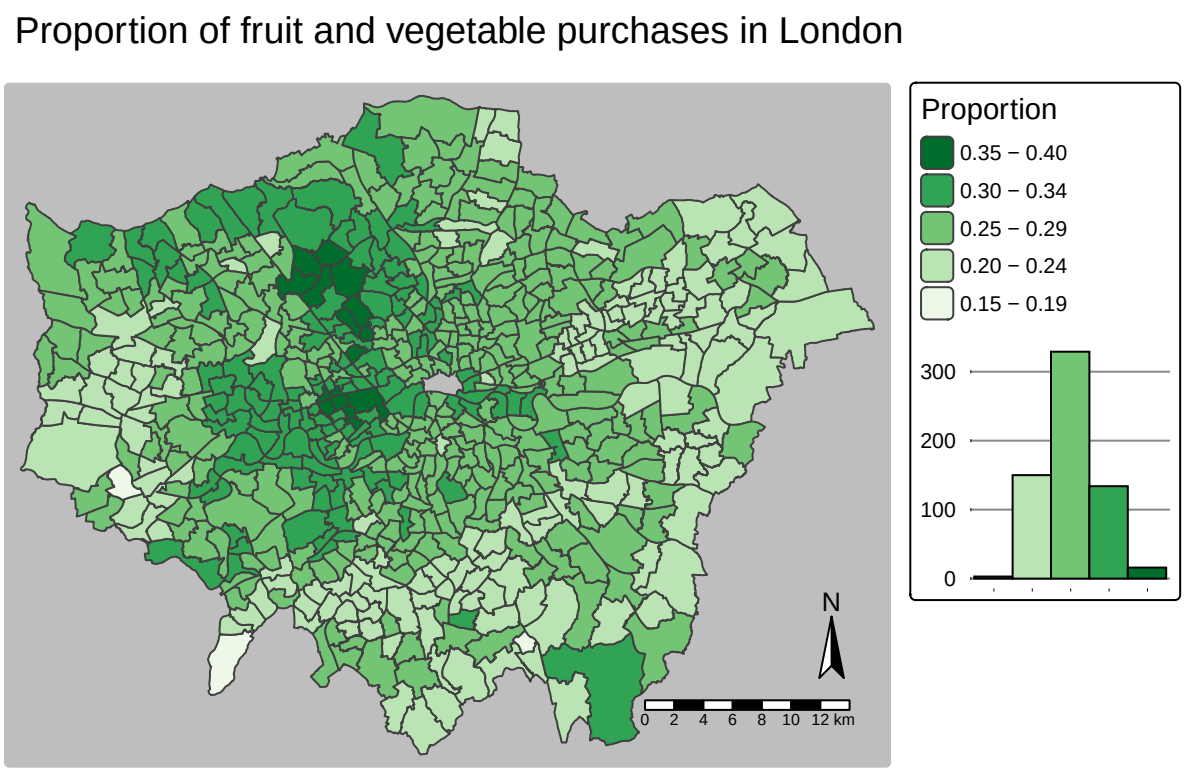
```
## Reading layer ‘London_Ward_CityMerged’ from data source
##   ‘C:\Users\kamar\OneDrive\Desktop\London-wards-2018\London-wards-2018_ESRI\London_Ward_CityMerged.sh
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 633 features and 6 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 503568.2 ymin: 155850.8 xmax: 561957.5 ymax: 200933.9
## Projected CRS: OSGB36 / British National Grid
```

```
tesco_and_wards <- inner_join(wards, hello)
```

```
## Joining with ‘by = join_by(ward_code)‘
```

```
tm_shape(tesco_and_wards) + tm_polygons(fill = "f_fruit_veg",
                                        fill.chart = tm_chart_histogram(),
                                        fill.legend = tm_legend(title = "Proportion", reverse = TRUE),
                                        fill.scale = tm_scale(values = "brewer.greens")
)+ tm_title("Proportion of fruit and vegetable purchases in London") + tm_compass() + tm_scalebar() + t
```

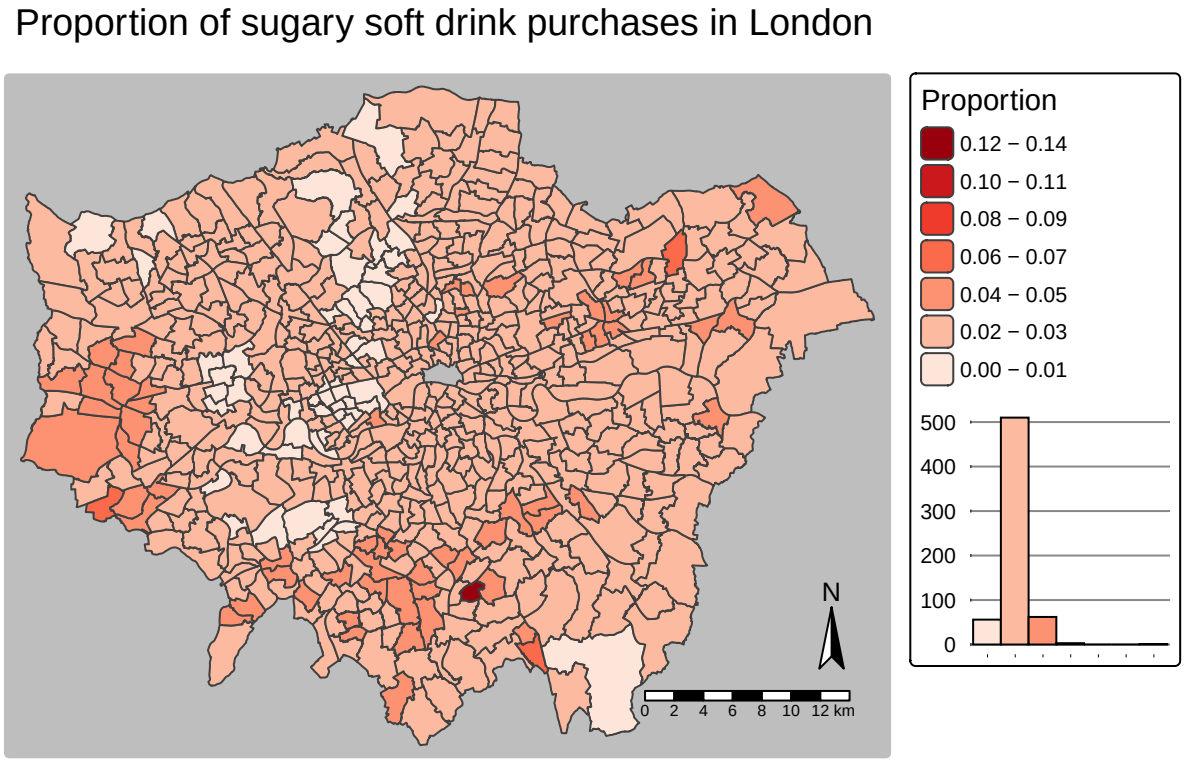
```
## [plot mode] fit legend/component: Some legend items or map compoments do not
## fit well, and are therefore rescaled.
## i Set the tmap option ‘component.autoscale = FALSE‘ to disable rescaling.
```



Map 2: Sugary Soft Drink Purchases (Unhealthy)

```
tm_shape(tesco_and_wards) +
  tm_polygons(
    "f_soft_drinks",
    fill.chart = tm_chart_histogram(),
    fill.legend = tm_legend(title = "Proportion", reverse = TRUE),
    fill.scale = tm_scale(values = "brewer.reds")
  ) +
  tm_title("Proportion of sugary soft drink purchases in London") +
  tm_compass() +
  tm_scalebar() +
  tm_layout(frame = FALSE, bg.color = "grey")
```

```
## [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.
```

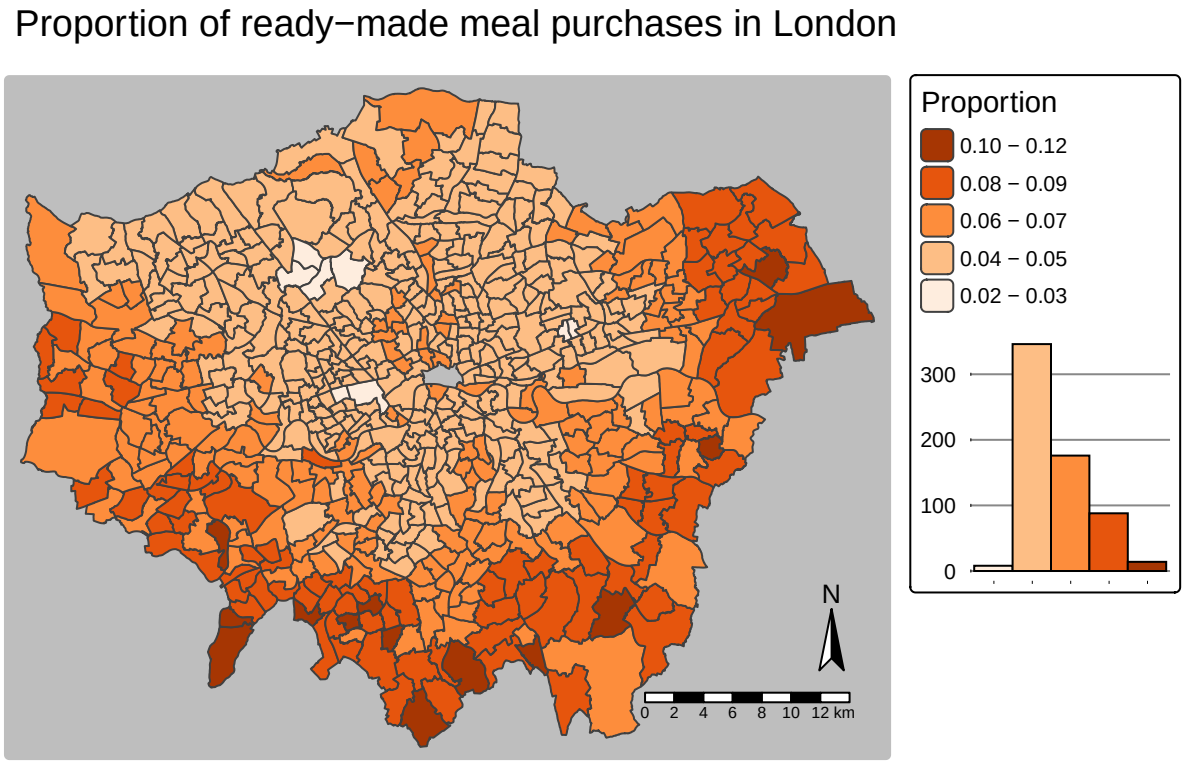


Map 3: Ready-Made Meal Purchases (Unhealthy)

```
tm_shape(tesco_and_wards) +
  tm_polygons(
```

```
    "f_readymade",
    fill.chart = tm_chart_histogram(),
    fill.legend = tm_legend(title = "Proportion", reverse = TRUE),
    fill.scale = tm_scale(values = "brewer.oranges")
  ) +
  tm_title("Proportion of ready-made meal purchases in London") +
  tm_compass() +
  tm_scalebar() +
  tm_layout(frame = FALSE, bg.color = "grey")
```

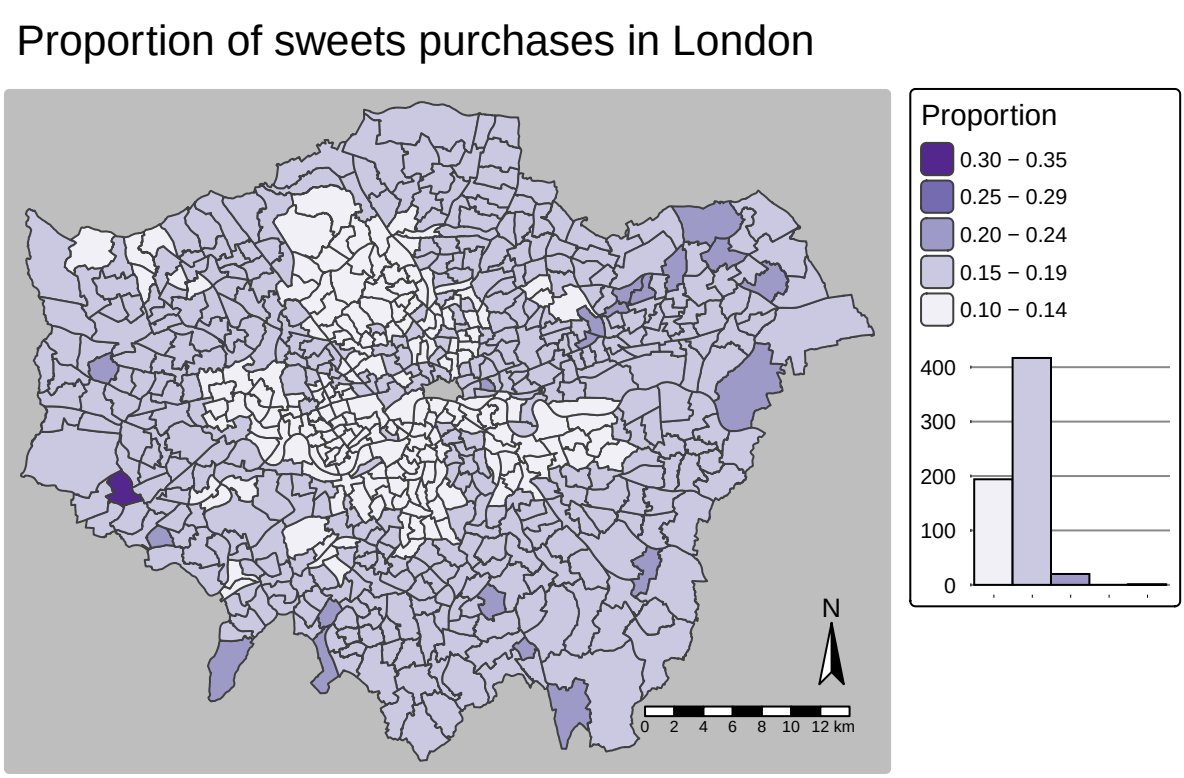
```
## [plot mode] fit legend/component: Some legend items or map compoments do not
## fit well, and are therefore rescaled.
## i Set the tmap option 'component.autoscale = FALSE' to disable rescaling.
```



Map 4: Sweets Purchases (Unhealthy)

```
tm_shape(tesco_and_wards) +
  tm_polygons(
    "f_sweets",
    fill.chart = tm_chart_histogram(),
    fill.legend = tm_legend(title = "Proportion", reverse = TRUE),
    fill.scale = tm_scale(values = "brewer.purples")
  ) +
```

```
  tm_title("Proportion of sweets purchases in London") +
  tm_compass() +
  tm_scalebar() +
  tm_layout(frame = FALSE, bg.color = "grey")
```




```
library(tidyverse)

grocery_data <- read_csv("year_osward_grocery.csv")
```

```
## Rows: 638 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.
```

```
eatwell_food_groups <- list(
  Fruit_and_Vegetables    = "f_fruit_veg",
  Starchy_Carbohydrates   = "f_grains",
  Protein_Foods           = c("f_poultry", "f_meat_red", "f_fish", "f_eggs"),
  Dairy_and_Alternatives  = "f_dairy",
  Oils_and_Spreads        = "f_fats_oils",
```

```
  Foods_High_in_Fat_Sugar = c("f_sweets", "f_readymade", "f_sauces", "f_soft_drinks")
)

eatwell_recommendations <- c(
  Fruit_and_Vegetables = 0.33,
  Starchy_Carbohydrates = 0.33,
  Protein_Foods = 0.12,
  Dairy_and_Alternatives = 0.15,
  Oils_and_Spreads = 0.07,
  Foods_High_in_Fat_Sugar = 0.05
)

grocery_data <- grocery_data %>%
  mutate(
    Fruit_and_Vegetables    = rowSums(across(all_of(eatwell_food_groups$Fruit_and_Vegetables)), na.rm =
    Starchy_Carbohydrates   = rowSums(across(all_of(eatwell_food_groups$Starchy_Carbohydrates)), na.rm =
    Protein_Foods           = rowSums(across(all_of(eatwell_food_groups$Protein_Foods)), na.rm = TRUE),
    Dairy_and_Alternatives  = rowSums(across(all_of(eatwell_food_groups$Dairy_and_Alternatives)), na.rm
    Oils_and_Spreads        = rowSums(across(all_of(eatwell_food_groups$Oils_and_Spreads)), na.rm = TRU
    Foods_High_in_Fat_Sugar = rowSums(across(all_of(eatwell_food_groups$Foods_High_in_Fat_Sugar)), na.r
  ) %>%
  mutate(
    total_basket = Fruit_and_Vegetables + Starchy_Carbohydrates + Protein_Foods +
                   Dairy_and_Alternatives + Oils_and_Spreads + Foods_High_in_Fat_Sugar
  ) %>%
  mutate(across(
    c(Fruit_and_Vegetables, Starchy_Carbohydrates, Protein_Foods,
      Dairy_and_Alternatives, Oils_and_Spreads, Foods_High_in_Fat_Sugar),
    ~ .x / total_basket
  )) %>%
  mutate(
    euclidean_distance = sqrt(
      (Fruit_and_Vegetables - eatwell_recommendations["Fruit_and_Vegetables"])^2 +
      (Starchy_Carbohydrates - eatwell_recommendations["Starchy_Carbohydrates"])^2 +
      (Protein_Foods - eatwell_recommendations["Protein_Foods"])^2 +
      (Dairy_and_Alternatives - eatwell_recommendations["Dairy_and_Alternatives"])^2 +
      (Oils_and_Spreads - eatwell_recommendations["Oils_and_Spreads"])^2 +
      (Foods_High_in_Fat_Sugar - eatwell_recommendations["Foods_High_in_Fat_Sugar"])^2
    )
  )

print(grocery_data %>%
  select(Fruit_and_Vegetables, Starchy_Carbohydrates, Protein_Foods,
         Dairy_and_Alternatives, Oils_and_Spreads, Foods_High_in_Fat_Sugar, euclidean_distance))
```

```
## # A tibble: 638 x 7
##    Fruit_and_Vegetables Starchy_Carbohydrates Protein_Foods
##                   <dbl>                 <dbl>         <dbl>
##  1                0.275                 0.162        0.0994
##  2                0.278                 0.163        0.110
##  3                0.257                 0.173        0.110
##  4                0.235                 0.167        0.0845
##  5                0.258                 0.169        0.103
```

```
##  6                0.271                 0.155        0.127
##  7                0.287                 0.160        0.121
##  8                0.265                 0.170        0.115
##  9                0.252                 0.181        0.107
## 10                0.271                 0.160        0.112
## # i 628 more rows
## # i 4 more variables: Dairy_and_Alternatives <dbl>, Oils_and_Spreads <dbl>,
## #   Foods_High_in_Fat_Sugar <dbl>, euclidean_distance <dbl>
```

```r
library(sf)
library(tmap)
library(dplyr)
library(readr)
library(here)
library(RColorBrewer)

# --- Load and prepare datasets ---
grocery_data <- read_csv("year_osward_grocery.csv") %>%
  rename(ward_code = area_id)
```

```
## Rows: 638 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.
```

```r
wards <- st_read(here("London-wards-2018/London-wards-2018_ESRI/London_Ward_CityMerged.shp")) %>%
  rename(ward_code = GSS_CODE, ward_name = NAME)
```

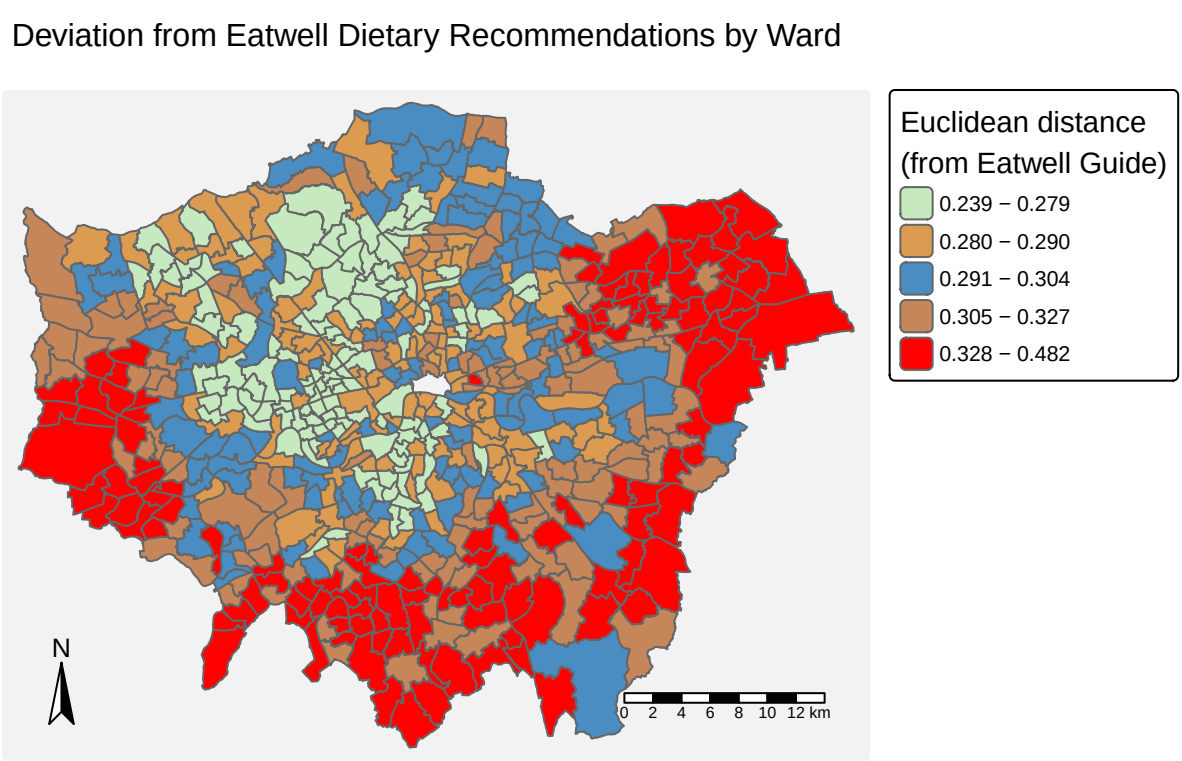
```
## Reading layer `London_Ward_CityMerged' from data source
##   `C:\Users\kamar\OneDrive\Desktop\London-wards-2018\London-wards-2018_ESRI\London_Ward_CityMerged.sh
##   using driver `ESRI Shapefile'
## Simple feature collection with 633 features and 6 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 503568.2 ymin: 155850.8 xmax: 561957.5 ymax: 200933.9
## Projected CRS: OSGB36 / British National Grid
```

```r
ward_distance_map <- inner_join(wards, grocery_data, by = "ward_code")

# --- Euclidean distance (your logic) ---
ward_distance_map <- ward_distance_map %>%
  mutate(
    Fruit_and_Vegetables = f_fruit_veg,
    Starchy_Carbohydrates = f_grains,
    Protein_Foods = f_poultry + f_meat_red + f_fish + f_eggs,
    Dairy_and_Alternatives = f_dairy,
    Oils_and_Spreads = f_fats_oils,
    Foods_High_in_Fat_Sugar = f_sweets + f_readymade + f_sauces + f_soft_drinks,
```

```
    total_basket = Fruit_and_Vegetables + Starchy_Carbohydrates +
                   Protein_Foods + Dairy_and_Alternatives +
                   Oils_and_Spreads + Foods_High_in_Fat_Sugar
  ) %>%
  mutate(across(
    c(Fruit_and_Vegetables, Starchy_Carbohydrates, Protein_Foods,
      Dairy_and_Alternatives, Oils_and_Spreads, Foods_High_in_Fat_Sugar),
    ~ .x / total_basket
  )) %>%
  mutate(
    euclidean_distance = sqrt(
      (Fruit_and_Vegetables - 0.33)^2 +
      (Starchy_Carbohydrates - 0.33)^2 +
      (Protein_Foods - 0.12)^2 +
      (Dairy_and_Alternatives - 0.15)^2 +
      (Oils_and_Spreads - 0.07)^2 +
      (Foods_High_in_Fat_Sugar - 0.05)^2
    )
  )

# --- Breaks ---
breaks <- quantile(ward_distance_map$euclidean_distance,
                   probs = seq(0, 1, length.out = 6),
                   na.rm = TRUE) %>% unique()

if(length(breaks) < 2) {
  breaks <- pretty(ward_distance_map$euclidean_distance, n = 6)
}

n_intervals <- length(breaks) - 1  # number of colour classes

# --- CUSTOM COLOURS (your request) ---
pal <- c(
  brewer.pal(9, "Greens")[3],
  brewer.pal(9, "YlOrBr")[5],
  brewer.pal(9, "PuBu")[5],
  "#2171B5",        # blue
  "#FD8D3C",        # orange
  "red"             # red
)

# tmap requires palette length == number of intervals
pal <- colorRampPalette(pal)(n_intervals)

# --- Map ---
tm_shape(ward_distance_map) +
  tm_polygons(
    fill = "euclidean_distance",
    col = "grey40",
    fill.scale = tm_scale_intervals(
      breaks = breaks,
      values = pal
    ),
```

```
    fill.legend = tm_legend(title = "Euclidean distance\n(from Eatwell Guide)")
  ) +
  tm_title("Deviation from Eatwell Dietary Recommendations by Ward") +
  tm_compass(position = c("left", "bottom")) +
  tm_scalebar(position = c("right", "bottom")) +
  tm_layout(
    legend.outside = TRUE,
    frame = FALSE,
    bg.color = "grey95"
  )
```

```
## [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.
```




```
library(dplyr)

ward_ranked <- ward_distance_map %>%
  st_drop_geometry() %>%
  select(ward_name, euclidean_distance) %>%
  arrange(euclidean_distance)

closest_wards <- head(ward_ranked, 10)
```

```r
furthest_wards <- tail(ward_ranked, 10)

cat("Top 10 wards MOST aligned with the Eatwell Guide:\n")
```

```
## Top 10 wards MOST aligned with the Eatwell Guide:
```

```r
print(closest_wards)
```

```
##           ward_name euclidean_distance
## 1         Newington          0.2389451
## 2  Camberwell Green          0.2399773
## 3          Colville          0.2547153
## 4     Ealing Common          0.2607325
## 5           Addison          0.2613217
## 6   Wembley Central          0.2626928
## 7    Clapham Common          0.2631993
## 8          Golborne          0.2637365
## 9        Queenstown          0.2645130
## 10     Notting Dale          0.2645479
```

```r
cat("\nTop 10 wards LEAST aligned with the Eatwell Guide:\n")
```

```
##
## Top 10 wards LEAST aligned with the Eatwell Guide:
```

```r
print(furthest_wards)
```

```
##                  ward_name euclidean_distance
## 623       Beddington North          0.3724628
## 624         Worcester Park          0.3729308
## 625 Tolworth and Hook Rise          0.3769392
## 626         Sutton Central          0.3786457
## 627    New Addington South          0.3919165
## 628         Chadwell Heath          0.3945639
## 629      Chessington South          0.4023940
## 630    New Addington North          0.4135884
## 631        Addiscombe East          0.4153577
## 632          Feltham North          0.4821914
```

```r
grocery_data <- read_csv("year_osward_grocery.csv") %>% rename(ward_code = area_id)
```

```
## Rows: 638 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.
```

```r
wards <- st_read(here("London-wards-2018/London-wards-2018_ESRI/London_Ward_CityMerged.shp")) %>%
  rename(ward_code = GSS_CODE, ward_name = NAME)
```

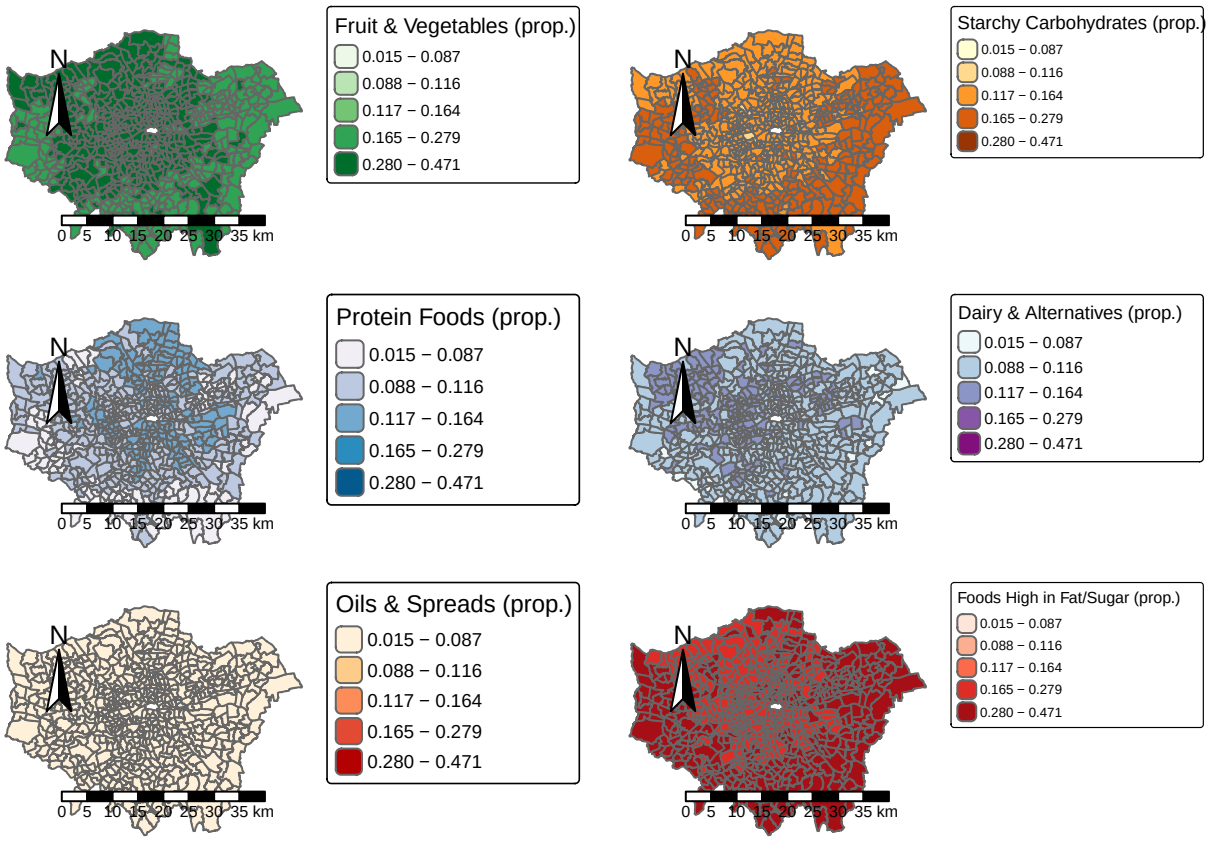
```
## Reading layer 'London_Ward_CityMerged' from data source
##   'C:\Users\kamar\OneDrive\Desktop\London-wards-2018\London-wards-2018_ESRI\London_Ward_CityMerged.sh
##   using driver 'ESRI Shapefile'
## Simple feature collection with 633 features and 6 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 503568.2 ymin: 155850.8 xmax: 561957.5 ymax: 200933.9
## Projected CRS: OSGB36 / British National Grid
```

```r
ward_sf <- inner_join(wards, grocery_data, by = "ward_code")

ward_sf <- ward_sf %>%
  mutate(
    Fruit_and_Vegetables = f_fruit_veg,
    Starchy_Carbohydrates = f_grains,
    Protein_Foods = f_poultry + f_meat_red + f_fish + f_eggs,
    Dairy_and_Alternatives = f_dairy,
    Oils_and_Spreads = f_fats_oils,
    Foods_High_in_Fat_Sugar = f_sweets + f_readymade + f_sauces + f_soft_drinks
  ) %>%
  mutate(total_basket = Fruit_and_Vegetables + Starchy_Carbohydrates + Protein_Foods +
           Dairy_and_Alternatives + Oils_and_Spreads + Foods_High_in_Fat_Sugar) %>%
  mutate(across(
    c(Fruit_and_Vegetables, Starchy_Carbohydrates, Protein_Foods,
      Dairy_and_Alternatives, Oils_and_Spreads, Foods_High_in_Fat_Sugar),
    ~ ifelse(is.na(total_basket) | total_basket == 0, NA_real_, .x / total_basket)
  ))

vars <- c(
  "Fruit_and_Vegetables",
  "Starchy_Carbohydrates",
  "Protein_Foods",
  "Dairy_and_Alternatives",
  "Oils_and_Spreads",
  "Foods_High_in_Fat_Sugar"
)

titles <- c(
  "Fruit & Vegetables (prop.)",
  "Starchy Carbohydrates (prop.)",
  "Protein Foods (prop.)",
  "Dairy & Alternatives (prop.)",
  "Oils & Spreads (prop.)",
  "Foods High in Fat/Sugar (prop.)"
)

palettes <- c("Greens", "YlOrBr", "PuBu", "BuPu", "OrRd", "Reds")

all_values <- unlist(lapply(vars, function(v) ward_sf[[v]]))
quantile_breaks <- quantile(all_values, probs = seq(0, 1, length.out = 6), na.rm = TRUE) %>% unique()
```

```r
if(length(quantile_breaks) < 2) quantile_breaks <- pretty(all_values, n = 6)

maps_list <- vector("list", length(vars))
for (i in seq_along(vars)) {
  v <- vars[i]
  maps_list[[i]] <- tm_shape(ward_sf) +
    tm_polygons(
      fill = v,
      col = "grey40",                        # outline color
      fill.scale = tm_scale_intervals(
        breaks = quantile_breaks,
        values = palettes[i]
      ),
      fill.legend = tm_legend(title = titles[i])
    ) +
    tm_layout(
      frame = FALSE,
      bg.color = "white",
      legend.outside = TRUE
    ) +
    tm_compass(position = c("left", "top")) +
    tm_scalebar(position = c("right", "bottom"))
}

comparison <- tmap_arrange(maps_list[[1]], maps_list[[2]], maps_list[[3]],
                           maps_list[[4]], maps_list[[5]], maps_list[[6]],
                           ncol = 2, nrow = 3)

comparison
```

```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## ‘cols4all::c4a_gui()‘ to explore them. The old palette name "Greens" is named
## "brewer.greens"
## Multiple palettes called "greens" found: "brewer.greens", "matplotlib.greens". The first one, "brewe
##
## [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.
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## ‘cols4all::c4a_gui()‘ to explore them. The old palette name "YlOrBr" is named
## "brewer.yl_or_br"
## Multiple palettes called "yl_or_br" found: "brewer.yl_or_br", "tol.yl_or_br", "matplotlib.yl_or_br".
##
## [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.
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## ‘cols4all::c4a_gui()‘ to explore them. The old palette name "PuBu" is named
## "brewer.pu_bu"
## Multiple palettes called "pu_bu" found: "brewer.pu_bu", "matplotlib.pu_bu". The first one, "brewer.pu
##
## [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.
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "BuPu" is named
## "brewer.bu_pu"
## Multiple palettes called "bu_pu" found: "brewer.bu_pu", "matplotlib.bu_pu". The first one, "brewer.bu
##
## [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.
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "OrRd" is named
## "brewer.or_rd"
## Multiple palettes called "or_rd" found: "brewer.or_rd", "matplotlib.or_rd". The first one, "brewer.or
##
## [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.
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "Reds" is named
## "brewer.reds"
## Multiple palettes called "reds" found: "brewer.reds", "matplotlib.reds". The first one, "brewer.reds"
##
## [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.
```



```
tmap_save(comparison, filename = "eatwell_groups_comparison.png", width = 2000, height = 3000, dpi = 300
```

```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "Greens" is named
## "brewer.greens"
## Multiple palettes called "greens" found: "brewer.greens", "matplotlib.greens". The first one, "brewer
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "YlOrBr" is named
## "brewer.yl_or_br"
## Multiple palettes called "yl_or_br" found: "brewer.yl_or_br", "tol.yl_or_br", "matplotlib.yl_or_br".
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "PuBu" is named
## "brewer.pu_bu"
## Multiple palettes called "pu_bu" found: "brewer.pu_bu", "matplotlib.pu_bu". The first one, "brewer.pu
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "BuPu" is named
## "brewer.bu_pu"
## Multiple palettes called "bu_pu" found: "brewer.bu_pu", "matplotlib.bu_pu". The first one, "brewer.bu
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "OrRd" is named
## "brewer.or_rd"
## Multiple palettes called "or_rd" found: "brewer.or_rd", "matplotlib.or_rd". The first one, "brewer.or
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "Reds" is named
## "brewer.reds"
## Multiple palettes called "reds" found: "brewer.reds", "matplotlib.reds". The first one, "brewer.reds"
##
## Map saved to C:\Users\kamar\OneDrive\Desktop\eatwell_groups_comparison.png
##
## Resolution: 2000 by 3000 pixels
##
## Size: 6.666667 by 10 inches (300 dpi)
```

```
for (i in seq_along(vars)) {
  fname <- paste0("map_", i, "_", vars[i], ".png")
  tmap_save(maps_list[[i]], filename = fname, width = 1200, height = 1000, dpi = 300)
}
```

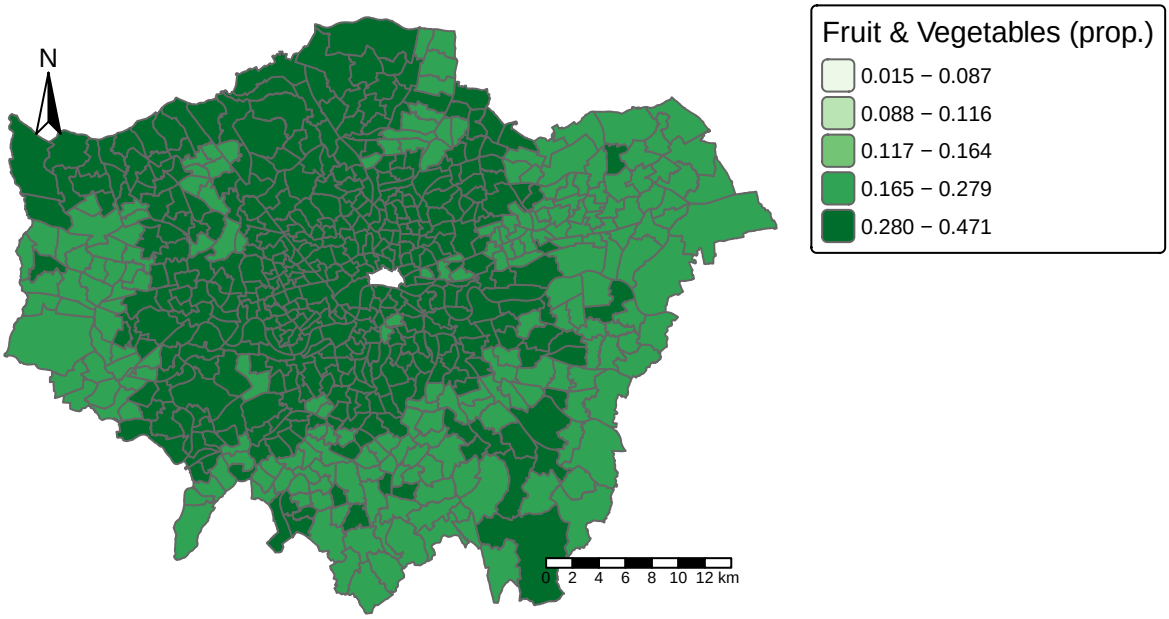
```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "Greens" is named
## "brewer.greens"
## Multiple palettes called "greens" found: "brewer.greens", "matplotlib.greens". The first one, "brewer
##
## Map saved to C:\Users\kamar\OneDrive\Desktop\map_1_Fruit_and_Vegetables.png
##
## Resolution: 1200 by 1000 pixels
##
## Size: 4 by 3.333333 inches (300 dpi)
```

```
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "YlOrBr" is named
## "brewer.yl_or_br"
## Multiple palettes called "yl_or_br" found: "brewer.yl_or_br", "tol.yl_or_br", "matplotlib.yl_or_br".
##
## Map saved to C:\Users\kamar\OneDrive\Desktop\map_2_Starchy_Carbohydrates.png
##
## Resolution: 1200 by 1000 pixels
##
## Size: 4 by 3.333333 inches (300 dpi)
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "PuBu" is named
## "brewer.pu_bu"
## Multiple palettes called "pu_bu" found: "brewer.pu_bu", "matplotlib.pu_bu". The first one, "brewer.pu
##
## Map saved to C:\Users\kamar\OneDrive\Desktop\map_3_Protein_Foods.png
##
## Resolution: 1200 by 1000 pixels
##
## Size: 4 by 3.333333 inches (300 dpi)
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "BuPu" is named
## "brewer.bu_pu"
## Multiple palettes called "bu_pu" found: "brewer.bu_pu", "matplotlib.bu_pu". The first one, "brewer.bu
##
## Map saved to C:\Users\kamar\OneDrive\Desktop\map_4_Dairy_and_Alternatives.png
##
## Resolution: 1200 by 1000 pixels
##
## Size: 4 by 3.333333 inches (300 dpi)
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "OrRd" is named
## "brewer.or_rd"
## Multiple palettes called "or_rd" found: "brewer.or_rd", "matplotlib.or_rd". The first one, "brewer.or
##
## Map saved to C:\Users\kamar\OneDrive\Desktop\map_5_Oils_and_Spreads.png
##
## Resolution: 1200 by 1000 pixels
##
## Size: 4 by 3.333333 inches (300 dpi)
##
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "Reds" is named
## "brewer.reds"
## Multiple palettes called "reds" found: "brewer.reds", "matplotlib.reds". The first one, "brewer.reds"
##
## Map saved to C:\Users\kamar\OneDrive\Desktop\map_6_Foods_High_in_Fat_Sugar.png
##
## Resolution: 1200 by 1000 pixels
##
```

```
## Size: 4 by 3.333333 inches (300 dpi)
```

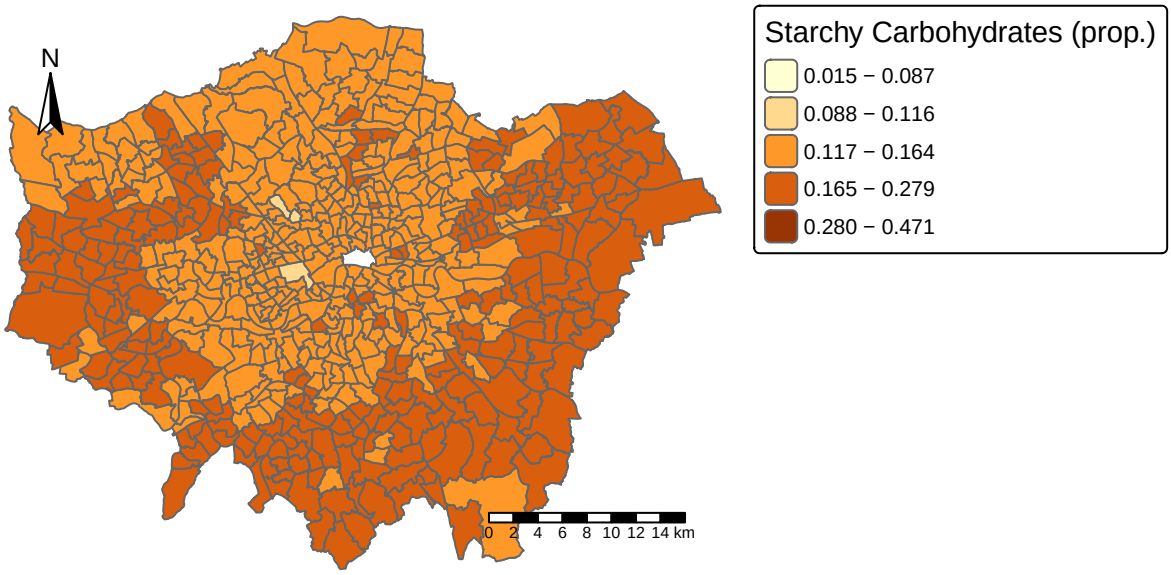
```
maps_list[[1]]
```

```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## ‘cols4all::c4a_gui()‘ to explore them. The old palette name "Greens" is named
## "brewer.greens"
## Multiple palettes called "greens" found: "brewer.greens", "matplotlib.greens". The first one, "brewe
```



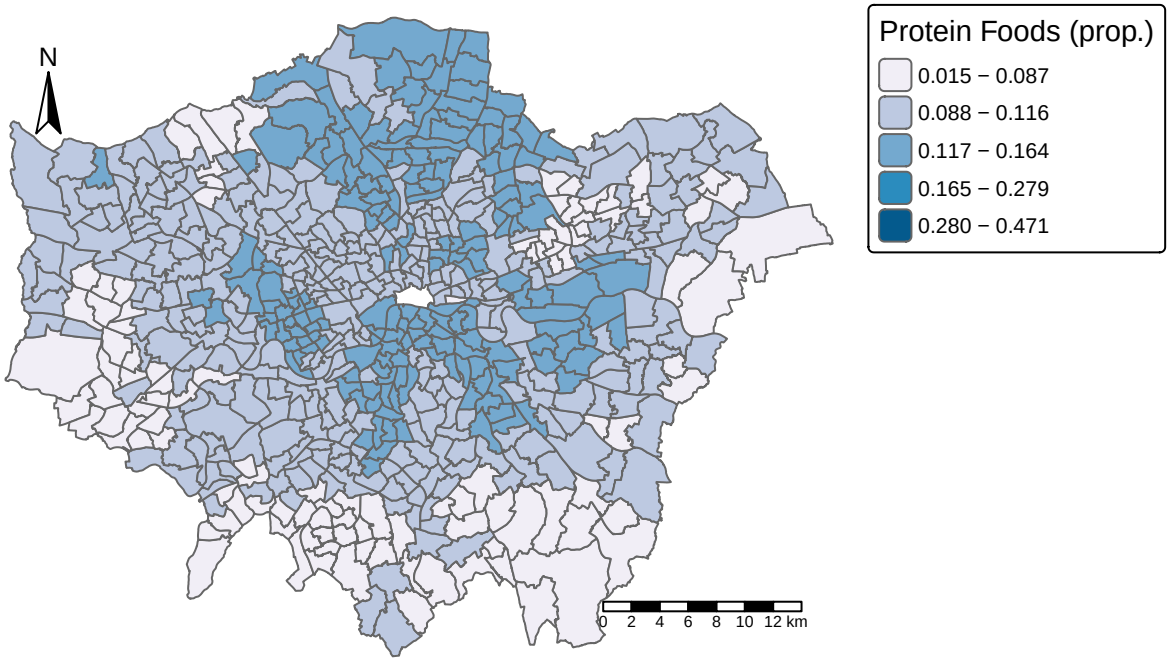

```
maps_list[[2]]
```

```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## ‘cols4all::c4a_gui()‘ to explore them. The old palette name "YlOrBr" is named
## "brewer.yl_or_br"
## Multiple palettes called "yl_or_br" found: "brewer.yl_or_br", "tol.yl_or_br", "matplotlib.yl_or_br".
```



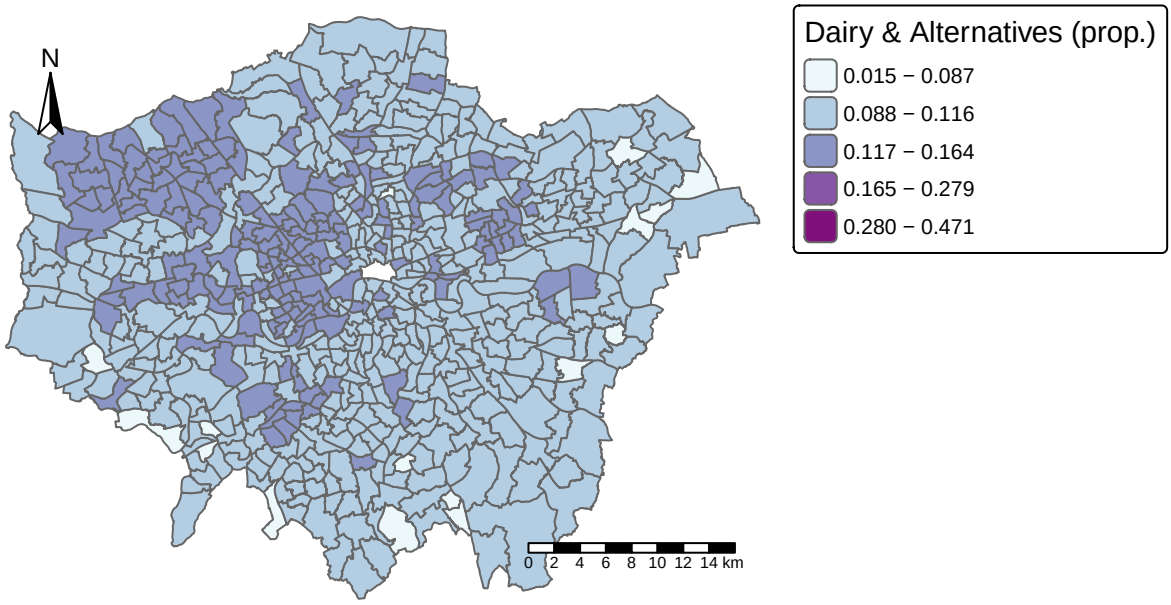

```
maps_list[[3]]
```

```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "PuBu" is named
## "brewer.pu_bu"
## Multiple palettes called "pu_bu" found: "brewer.pu_bu", "matplotlib.pu_bu". The first one, "brewer.pu
```



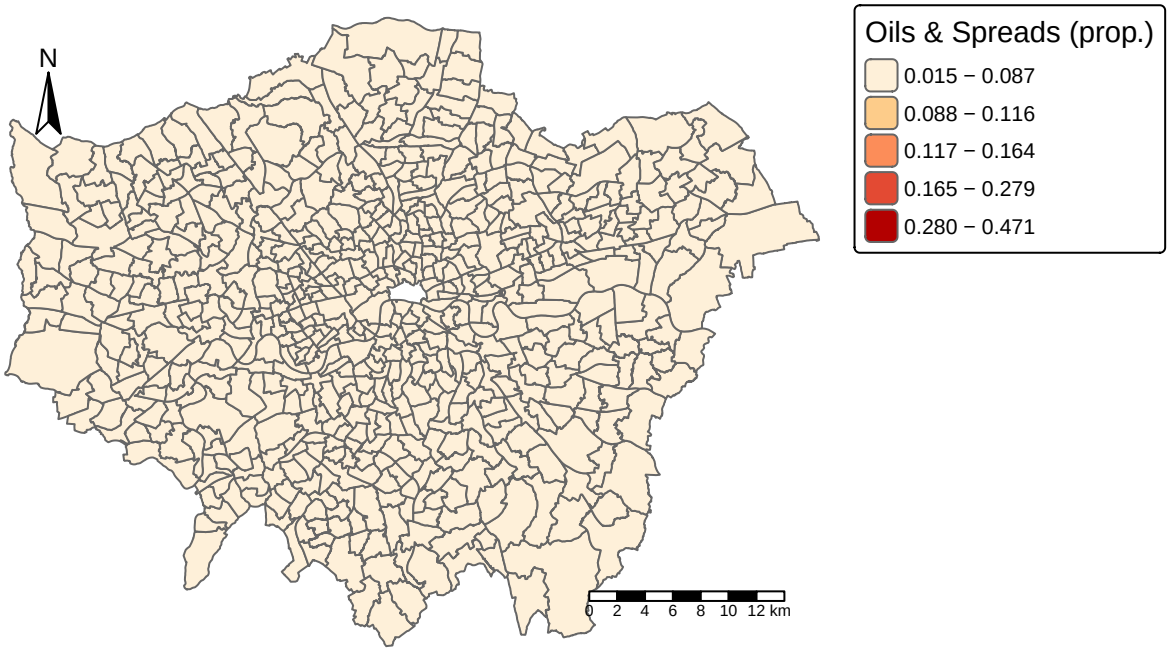

```
maps_list[[4]]
```

```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## ‘cols4all::c4a_gui()‘ to explore them. The old palette name "BuPu" is named
## "brewer.bu_pu"
## Multiple palettes called "bu_pu" found: "brewer.bu_pu", "matplotlib.bu_pu". The first one, "brewer.bu
```



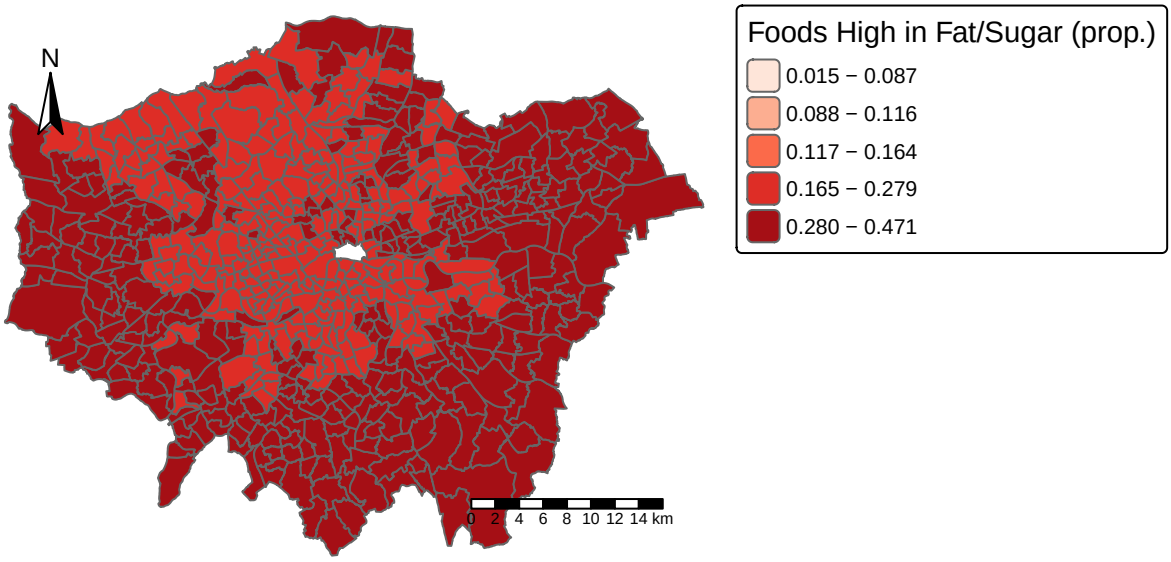

```
maps_list[[5]]
```

```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## ‘cols4all::c4a_gui()‘ to explore them. The old palette name "OrRd" is named
## "brewer.or_rd"
## Multiple palettes called "or_rd" found: "brewer.or_rd", "matplotlib.or_rd". The first one, "brewer.o
```




```
maps_list[[6]]
```

```
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## ‘cols4all::c4a_gui()‘ to explore them. The old palette name "Reds" is named
## "brewer.reds"
## Multiple palettes called "reds" found: "brewer.reds", "matplotlib.reds". The first one, "brewer.reds
```


N
0 2 4 6 8 10 12 14 km
Foods High in Fat/Sugar (prop.)
0.015 − 0.087
0.088 − 0.116
0.117 − 0.164
0.165 − 0.279
0.280 − 0.471