Accessibility analysis to the NHS vaccination centre

Won Do Lee

10 July 2021

Accessibility analysis to the mass COVID-19 vaccination centres across England, by car, bike, and on foot.

Set up the workspace

Key reference

- Mapping Inequalities in COVID-19 Vaccine Accessibility, developed by MSOA level using Python code.
- Yet, further analysis will be required to examine the association between accessibility and vaccination uptake, such as correlation test and regression model.

Research framework

1) Geocoding the NHS England vaccination sites.
2) Generating the OD matrices using the near function to find the shortest distance between centres and each centroid of LTLAs.
3) Updating the routes using map matching algorithm from the OD matrices by different transport modes.
4) Measuring the accessibility to the NHS mass vaccination centres - to retain travel duration and distance by transport modes.
5) Correlation test between the growth of vaccination rates and accessibility.
6) Additional analysis to examine the relationship between vaccination rates and accessibility with demographic attributes.

Analysis

1. Geocoding the COVID-19 vaccination sites across England

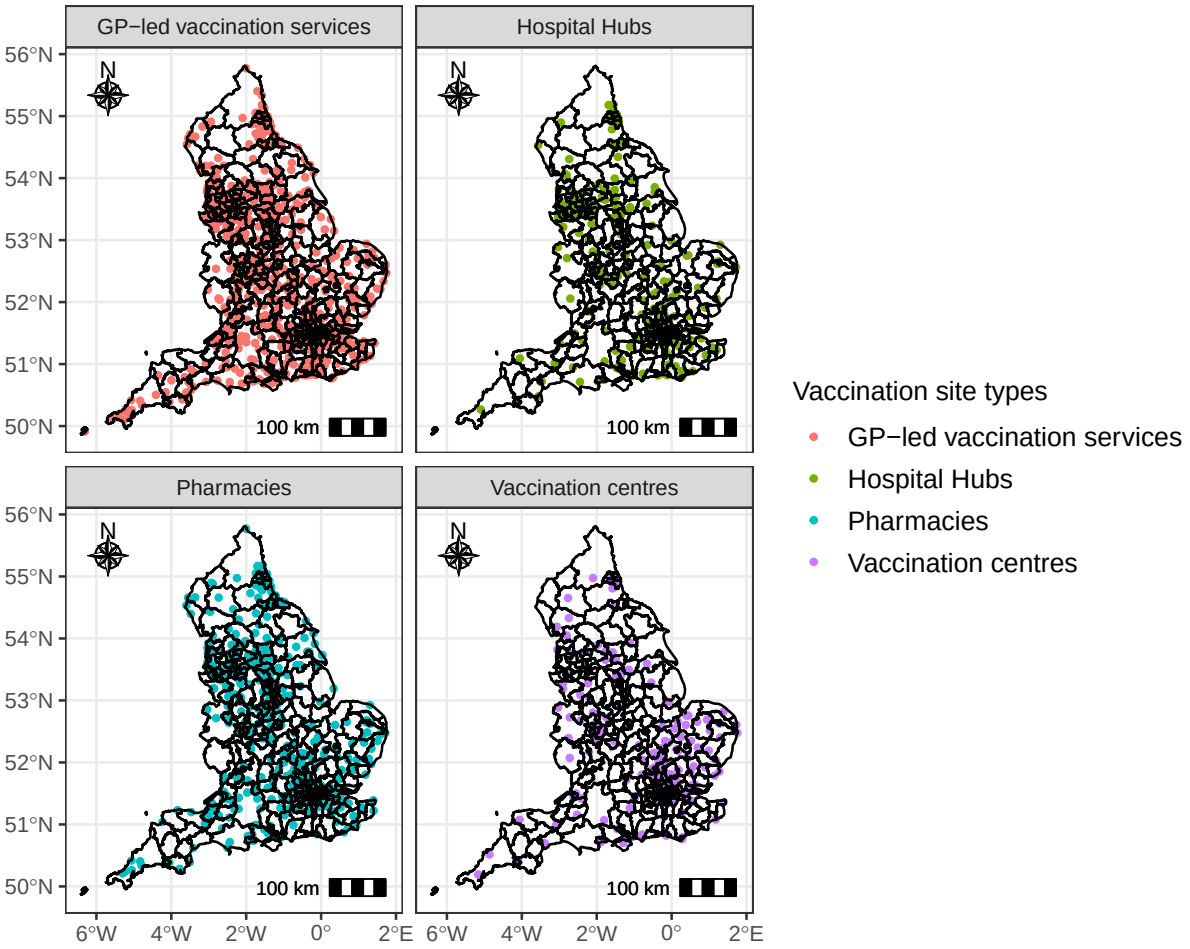
- Data retrieved from the NHS England webpage.
- It encompassing the different type of sites; **(mass) vaccination centres**, pharmacies, GP-led vaccination sites, and Hospital Hubs.
- This exercise extract the **(mass) vaccination centres**, which contains 198 centres across England.
- Geocoding the 198 centres based on the postcode, because of the Site Address require the manual process and update.
- In this exercise, **OpenCage Geocoding API** has been deployed to use *opencage* package in R software. The usage limit is 2,500 requests per day without the payment for free trail.
- The official UK government website for data and insights on coronavirus (i.e. Dashboard webpage) serves to search and download the vaccination metrics by area types, LTLA level is the finer level of granularity.

```r
# Geographic map of England vacciation
# centres
vc.sites <- read.csv("https://raw.githubusercontent.com/wondolee/accessibility/main/data/en.vc.sites.
vc.sites$oc_lat <- as.numeric(vc.sites$oc_lat)
vc.sites$oc_lng <- as.numeric(vc.sites$oc_lng)

require(sf)
require(maptools)
require(rgeos)
s.vc.sites <- st_as_sf(vc.sites, coords = c("oc_lng",
    "oc_lat"), crs = 4326)
# st_write(s.vc.sites,'en.vc.sites.geojson')
en.ltla <- st_read("https://raw.githubusercontent.com/wondolee/accessibility/main/data/simple.en.ltla
```

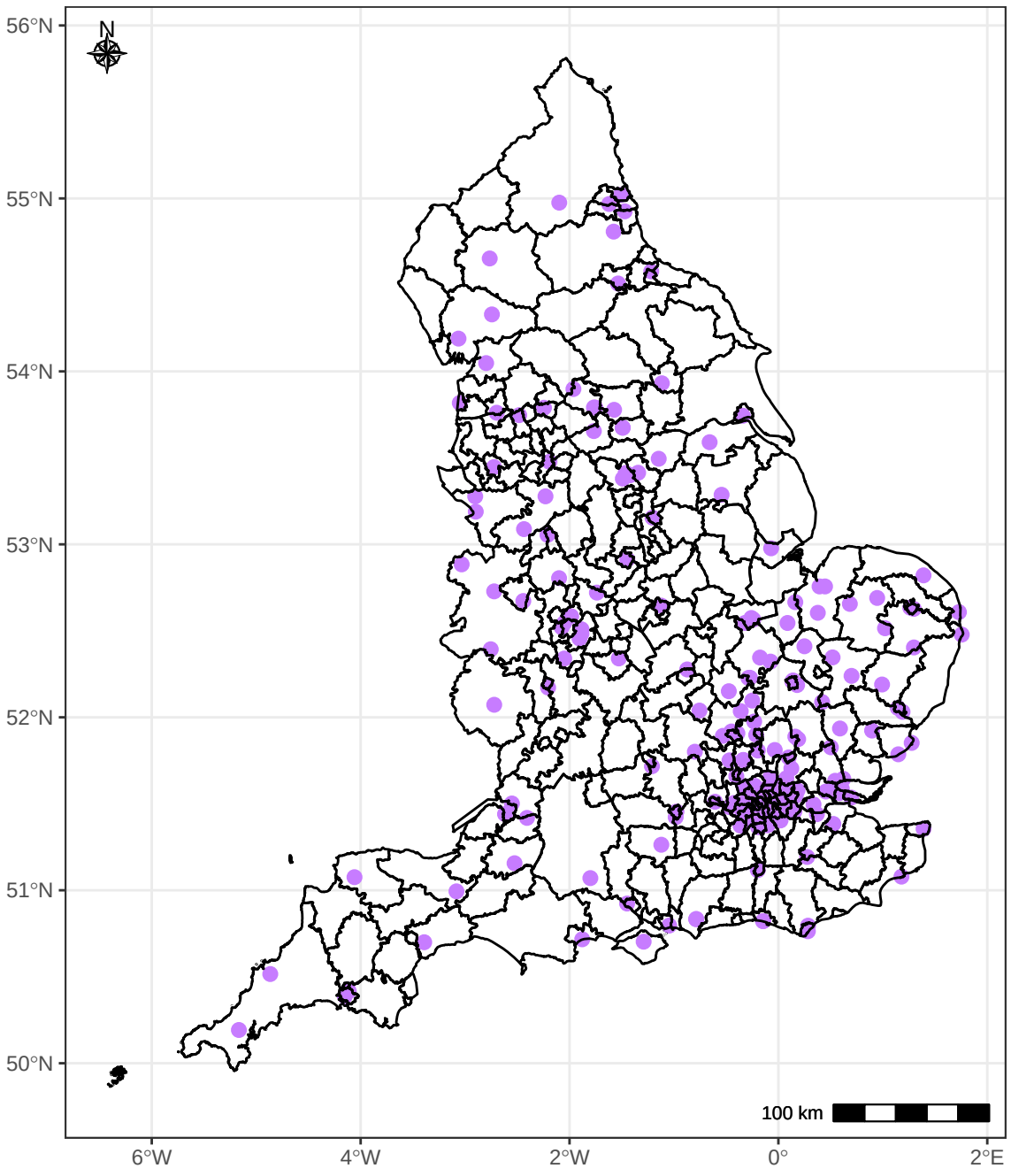
```
## Reading layer `simple' from data source
##   `https://raw.githubusercontent.com/wondolee/accessibility/main/data/simple.en.ltla.for.ep.geojso
##   using driver `GeoJSON'
## Simple feature collection with 315 features and 6 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -6.418924 ymin: 49.86476 xmax: 1.768566 ymax: 55.81166
## CRS:           4326
```

```r
s.vc.centres <- subset(s.vc.sites, Type ==
    "Vaccination centres")
# en.vac.centres<-leaflet(s.vc.centres)
# %>% addTiles() %>% addPolygons(data =
# en.ltla,weight = 1,fill = F,color =
# 'black',highlightOptions =
# highlightOptions(bringToFront =
# TRUE)) %>% addMarkers(popup =
# s.vc.centres$Site.Name,
# clusterOptions =
# markerClusterOptions())
# mapshot(en.vac.centres, url =
# 'en.vac.centres.html')

require(ggplot2)
require(RColorBrewer)
require(classInt)
require(maptools)
require(ggrepel)
require(scales)
require(rgdal)
require(ggspatial)
require(extrafont)
map.vac.sites <- ggplot() + geom_sf(data = s.vc.sites,
    aes(colour = Type), size = 1) + facet_wrap(~Type) +
    labs(colour = "Vaccination site types") +
    geom_sf(data = en.ltla, fill = NA, color = "black",
        size = 0.5, show.legend = FALSE) +
    theme_bw() + theme(legend.position = "right",
    plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
    legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
```

```
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.en.vac.site.pdf", map.vac.sites,
    width = 250, height = 250, units = "mm",
    dpi = 300, bg = "white")
map.vac.sites
```




```
map.vac.centres <- ggplot() + geom_sf(data = subset(s.vc.sites,
    Type == "Vaccination centres"), aes(colour = Type),
    color = "#C77CFF", size = 2.5) + labs(colour = "Vaccination site types") +
    geom_sf(data = en.ltla, fill = NA, color = "black",
        size = 0.5, show.legend = FALSE) +
    theme_bw() + theme(legend.position = "bottom",
    plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
    legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.en.vac.centres.pdf", map.vac.centres,
    width = 250, height = 250, units = "mm",
    dpi = 300, bg = "white")
map.vac.centres
```



2. Identifying the nearest vaccination centres from each centroid of LTLAs in England

```
s.vc.centres.grid <- st_transform(s.vc.centres,
    27700)
s.vc.centres.grid <- as(s.vc.centres.grid,
    "Spatial")
en.ltla <- st_transform(en.ltla, 27700)
en.ltla <- as(en.ltla, "Spatial")
near.centre <- as.data.frame(gDistance(s.vc.centres.grid,
    en.ltla, byid = TRUE))
min.d <- as.data.frame(apply(near.centre,
    1, function(x) order(x, decreasing = F)[2]))
colnames(min.d) <- "centre.id"
```

```
min.d$ltla.id <- rownames(min.d)

# Get the coordinate information from
# centres and each LTLAs
s.vc.centre.coordinate <- as.data.frame(s.vc.centres.grid[c(1,
    6)])
colnames(s.vc.centre.coordinate) <- c("centre.id",
    "type", "centre.x", "centre.y")
require(dplyr)
min.d <- left_join(min.d, s.vc.centre.coordinate,
    by = "centre.id")
ltla.grid <- as.data.frame(as(st_centroid(st_as_sf(en.ltla,
    27700)), "Spatial"))
colnames(ltla.grid)[7:8] <- c("ltla.x", "ltla.y")
ltla.grid$ltla.id <- rownames(ltla.grid)
ltla.grid <- ltla.grid[c("ltla.id", "ltla.x",
    "ltla.y", "LTLA19CD", "LTLA19NM")]
min.d <- left_join(min.d, ltla.grid, by = "ltla.id")
min.d <- min.d[c("ltla.x", "ltla.y", "centre.x",
    "centre.y", "ltla.id", "centre.id", "type",
    "LTLA19CD", "LTLA19NM")]
saveRDS(min.d, "shortest.distance.to.vac.centres.from.ltlas.rda")
```

3. Generating the shortest distance from each origin and destination; polylines to display the OD matrices

```
rows <- split(min.d, seq(nrow(min.d)))
lines <- lapply(rows, function(row) {
    lmat <- matrix(unlist(row[1:4]), ncol = 2,
        byrow = TRUE)
    st_linestring(lmat)
})
od.lines <- st_as_sf(st_sfc(lines, crs = 27700),
    crs = 27700)
od.lines$id <- rownames(od.lines)
min.d$id <- rownames(min.d)
od.lines <- left_join(od.lines, min.d, by = c("id"))
```

4. Map matching algorithm

```
require(stplanr)
require(osrm)
od.lines.lat <- st_transform(od.lines, 4326)
saveRDS(od.lines.lat, "polylines.shortest.distance.gcs.rda")
trans.network <- line2route(od.lines.lat,
    route_osrm)  #GCS (EPSG:4326) projection
saveRDS(trans.network, "map.match.trans.network.rda")

od.lines.lat.coord = od_coords(od.lines.lat)
from = od.lines.lat.coord[, 1:2]
to = od.lines.lat.coord[, 3:4]
car.trans.network = route(from, to, route_fun = route_osrm,
    osrm.profile = "car")
```

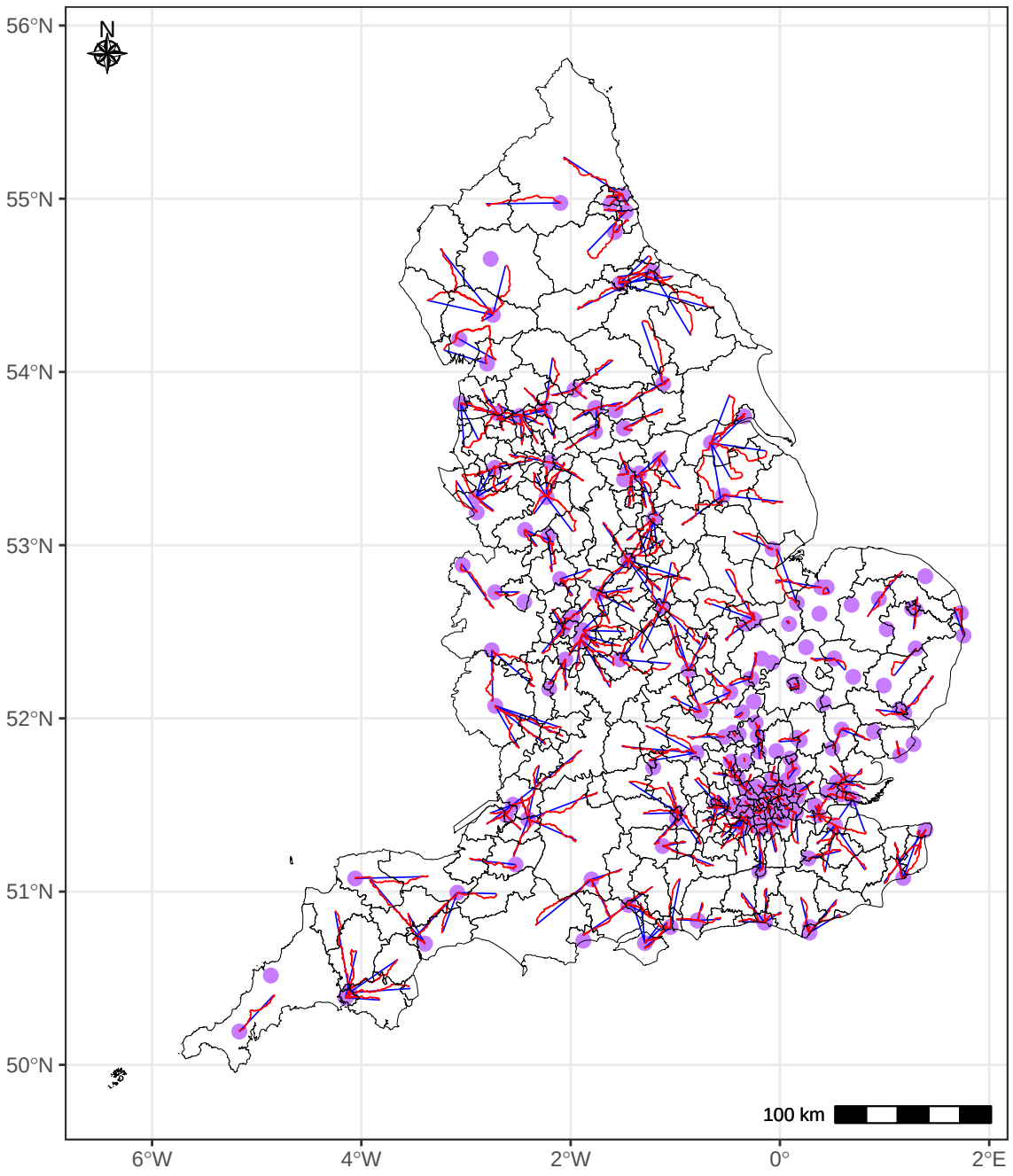
```
bike.trans.network = route(from, to, route_fun = route_osrm,
    osrm.profile = "bike")
foot.trans.network = route(from, to, route_fun = route_osrm,
    osrm.profile = "foot")
saveRDS(car.trans.network, "car.road.network.rda")
saveRDS(bike.trans.network, "bike.road.network.rda")
saveRDS(foot.trans.network, "foot.road.network.rda")
```

5. Accessibility analysis

```
require(stplanr)
en.ltla <- st_read("https://raw.githubusercontent.com/wondolee/accessibility/main/data/simple.en.ltla
```

```
## Reading layer `simple' from data source
##   `https://raw.githubusercontent.com/wondolee/accessibility/main/data/simple.en.ltla.for.ep.geojso
##   using driver `GeoJSON'
## Simple feature collection with 315 features and 6 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -6.418924 ymin: 49.86476 xmax: 1.768566 ymax: 55.81166
## CRS:           4326
```

```
trans.network <- readRDS("map.match.trans.network.rda")
map.route.vac.centres <- ggplot() + geom_sf(data = subset(s.vc.sites,
    Type == "Vaccination centres"), aes(colour = Type),
    color = "#C77CFF", size = 2.5) + labs(colour = "Vaccination site types") +
    geom_sf(data = od.lines, fill = NA, color = "blue",
        size = 0.3, show.legned = TRUE) +
    geom_sf(data = trans.network, fill = NA,
        color = "red", size = 0.3, show.legned = TRUE) +
    geom_sf(data = en.ltla, fill = NA, color = "black",
        size = 0.1, show.legend = FALSE) +
    theme_bw() + theme(legend.position = "bottom",
    plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
    legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.route.vac.centres.pdf", map.route.vac.centres,
    width = 250, height = 250, units = "mm",
    dpi = 300, bg = "white")
map.route.vac.centres
```




```
car.routes <- readRDS("car.road.network.rda")
acc.car <- as.data.frame(car.routes[c("route_number",
    "distance", "duration")])
acc.car <- acc.car[c(-4)]
colnames(acc.car) <- c("routes", "car.dist",
    "car.dur")
bike.routes <- readRDS("bike.road.network.rda")
acc.bike <- as.data.frame(bike.routes[c("route_number",
    "distance", "duration")])
acc.bike <- acc.bike[c(-4)]
colnames(acc.bike) <- c("routes", "bike.dist",
    "bike.dur")
foot.routes <- readRDS("foot.road.network.rda")
acc.foot <- as.data.frame(foot.routes[c("route_number",
    "distance", "duration")])
```

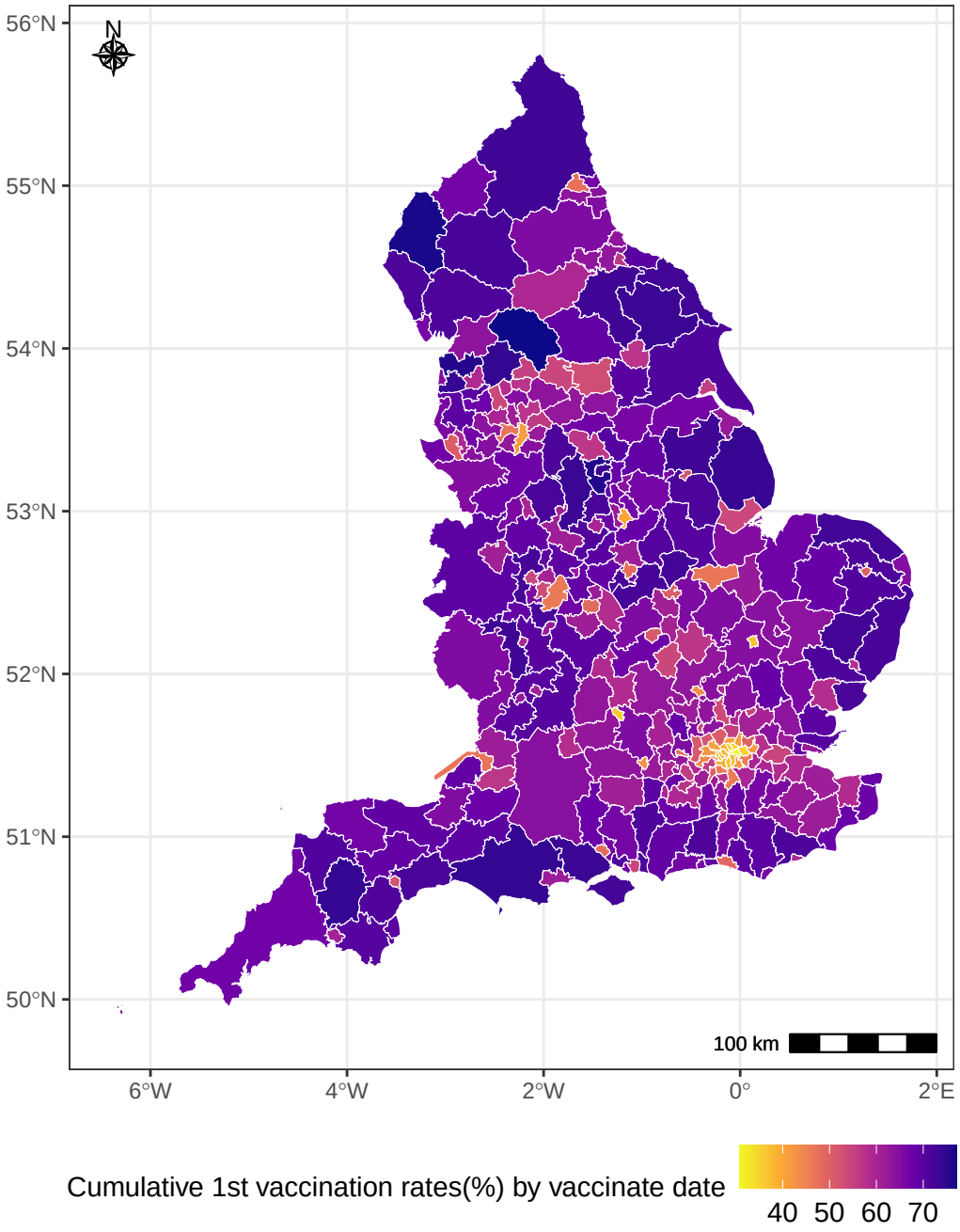
```
acc.foot <- acc.foot[c(-4)]
colnames(acc.foot) <- c("routes", "foot.dist",
    "foot.dur")

require(plyr)
min.d <- readRDS("shortest.distance.to.vac.centres.from.ltlas.rda")
min.d <- min.d[c("ltla.id", "LTLA19CD", "LTLA19NM")]
colnames(min.d) <- c("routes", "LTLA19CD",
    "LTLA19NM")
acc.all <- join_all(list(acc.car, acc.bike,
    acc.foot, min.d), by = "routes", type = "left")
acc.all <- acc.all[order(acc.all$LTLA19CD),
    ]
saveRDS(acc.all, "acc.all.modes.rda")
```

6. Geographic illustration of vaccination rates and accessibility to the vaccination centres.

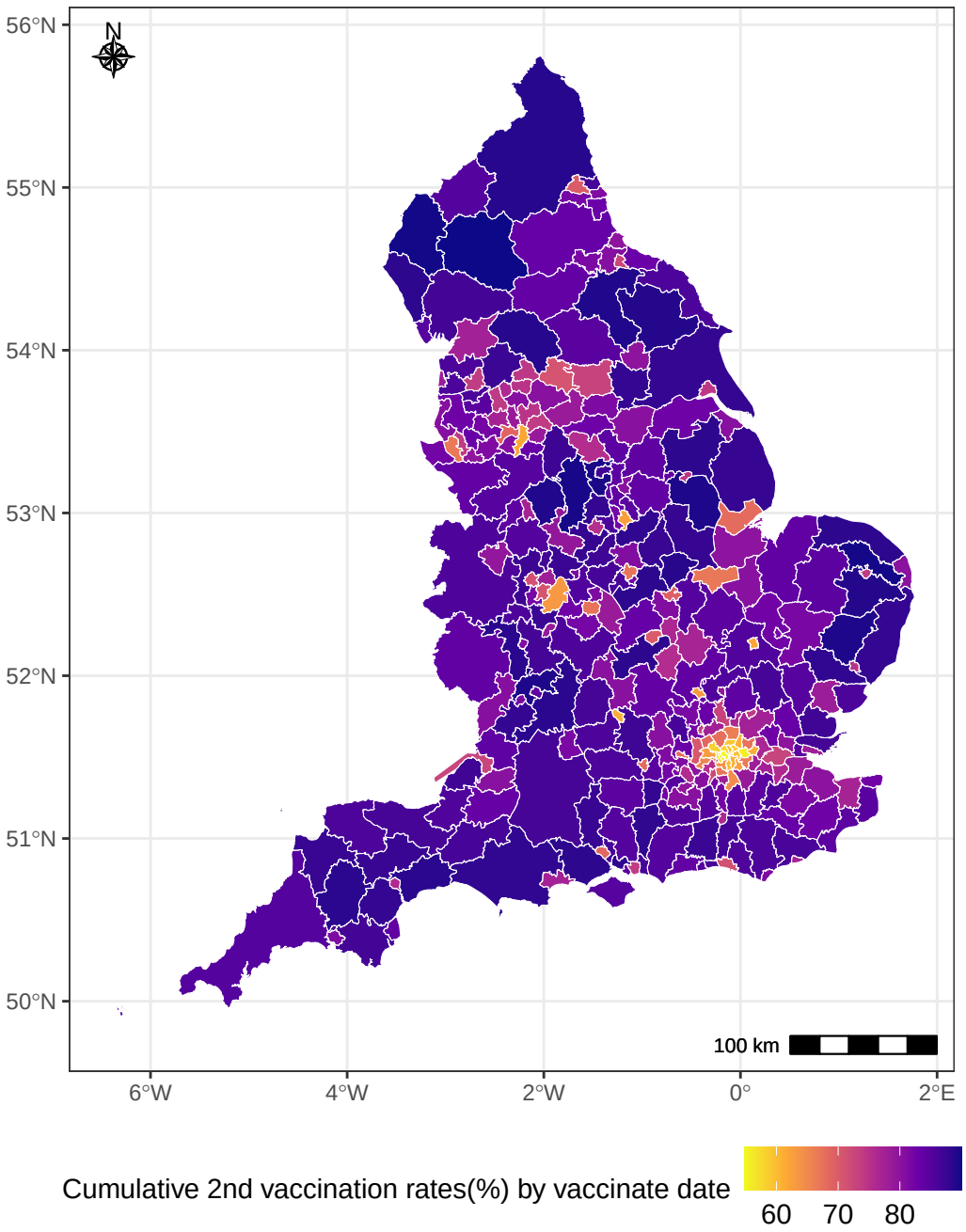
- Cumulative COVID-19 vaccination status has been updated on NHS Digital webpage, it covers the first/second dose of COVID-19 vaccination breakdown age bands or gender by LTLA level in England.
- However, their total number of LTLA level is 309 (315 in UK official dashboard), which does not overlay with accessibility analysis results.
- Thus, Vaccination rates by geographic area has been searched and downloaded by the UK official COVID-19 dashboard website, it serves to search and download the cumulative COVID-19 vaccination status and percentage by LTLA level (served as the lowest geographical level) in England.

```
# Downalod weekly COVID-19 vaccinations
# (published each Thursday at 2 pm)
# breakdown age bands by each LTLA.
# Webpage:
# https://www.england.nhs.uk/statistics/wp-content/uploads/sites/2/2021/07/COVID-19-weekly-announced-
vac.data <- read.csv("https://api.coronavirus.data.gov.uk/v2/data?areaType=ltla&metric=cumVaccination

en.vac.data <- subset(vac.data, grepl("E0",
    areaCode))
colnames(en.vac.data) <- c("LTLA19CD", "LTLA19NM",
    "areaType", "last.date", "first.does.per",
    "second.dose.per", "total.dose.per")
en.vac.data <- en.vac.data[order(en.vac.data$LTLA19CD),
    ]

require(lubridate)
en.vac.data$last.date <- ymd(en.vac.data$last.date)
en.vac.data <- subset(en.vac.data, last.date ==
    max(last.date))
saveRDS(en.vac.data, "en.vac.data.latest.10Jul21.rda")

ltla.vac.data <- left_join(en.ltla, en.vac.data,
    by = c("LTLA19CD", "LTLA19NM"))

require(viridis)
map.vac.1st.dose <- ggplot(ltla.vac.data) +
    geom_sf(aes(fill = first.does.per), color = NA) +
    geom_sf(colour = "white", size = 0.1,
        fill = NA) + scale_fill_viridis_c(option = "plasma",
```

```
    direction = -1) + theme_bw() + labs(fill = "Cumulative 1st vaccination rates(%) by vaccinate date
    theme(legend.position = "bottom", plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
        legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.route.1st.dose.pdf", map.vac.1st.dose,
    width = 250, height = 250, units = "mm",
    dpi = 300, bg = "white")
map.vac.1st.dose
```

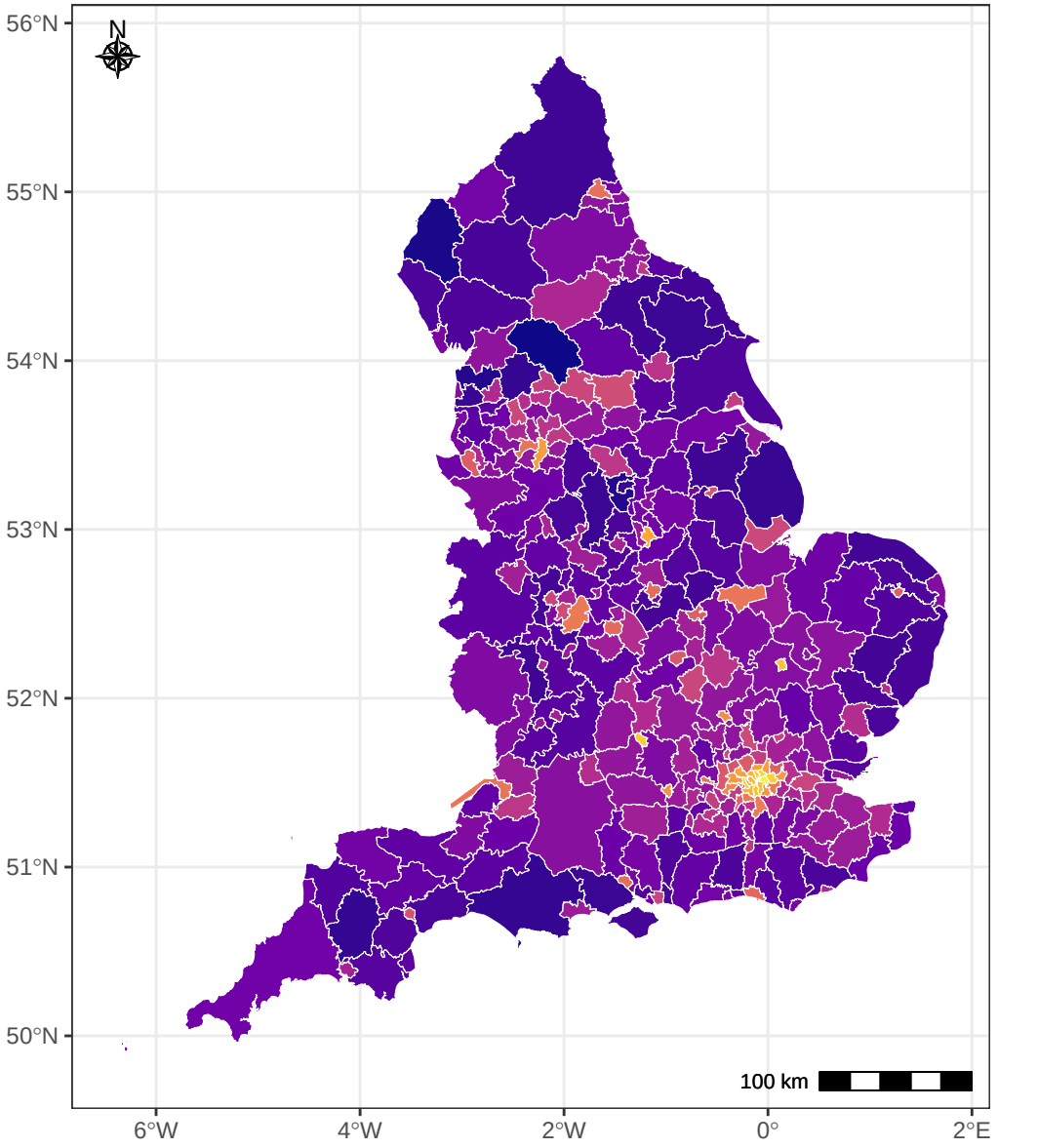


```
map.vac.2nd.dose <- ggplot(ltla.vac.data) +
    geom_sf(aes(fill = second.dose.per),
        color = NA) + geom_sf(colour = "white",
    size = 0.1, fill = NA) + scale_fill_viridis_c(option = "plasma",
    direction = -1) + theme_bw() + labs(fill = "Cumulative 2nd vaccination rates(%) by vaccinate date
    theme(legend.position = "bottom", plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
        legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.route.2nd.dose.pdf", map.vac.2nd.dose,
    width = 250, height = 250, units = "mm",
    dpi = 300, bg = "white")
map.vac.2nd.dose
```



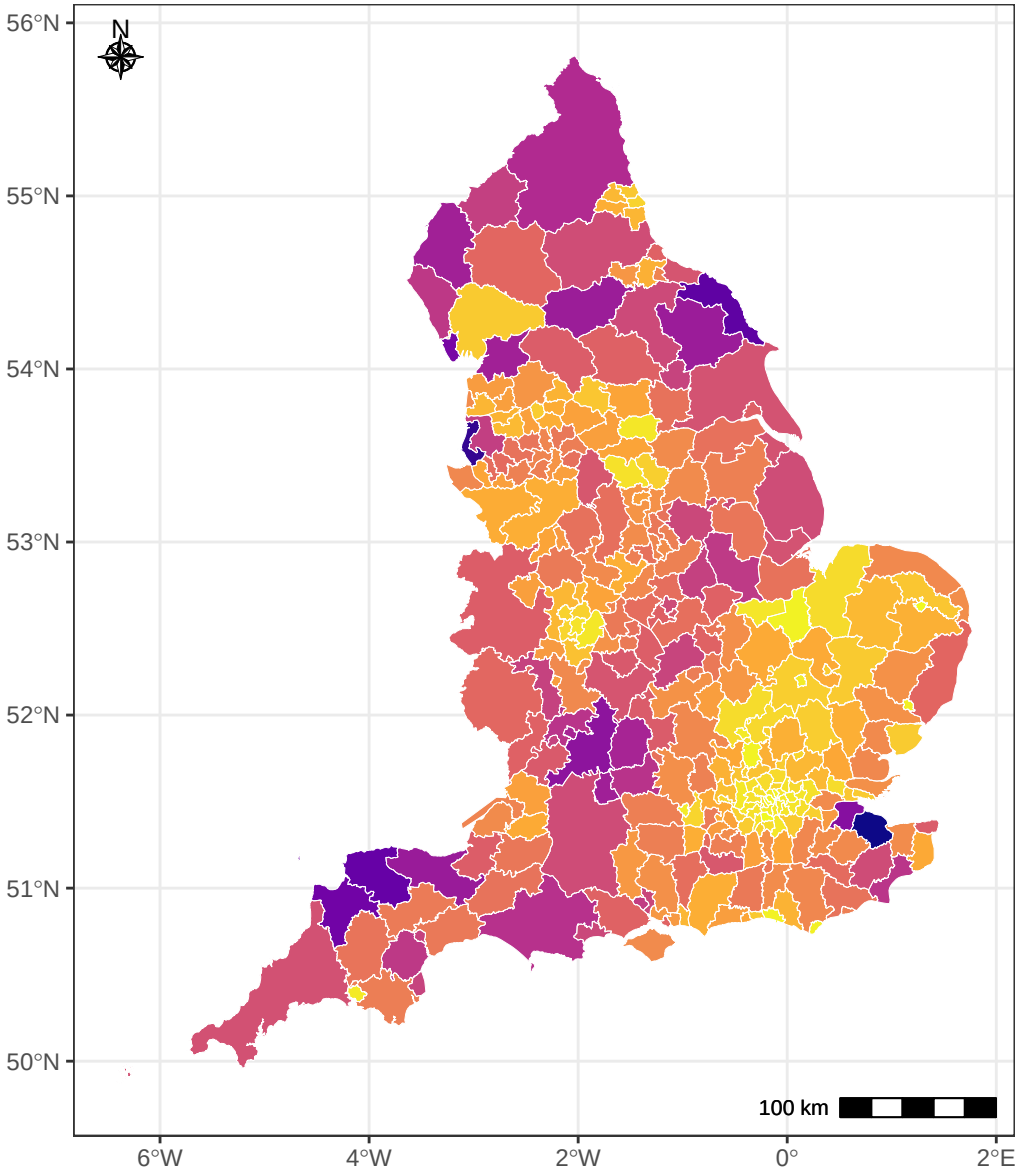

```
map.vac.total.dose <- ggplot(ltla.vac.data) +
    geom_sf(aes(fill = total.dose.per), color = NA) +
    geom_sf(colour = "white", size = 0.1,
        fill = NA) + scale_fill_viridis_c(option = "plasma",
    direction = -1) + theme_bw() + labs(fill = "Cumulative total dose vaccination rates(%) by vaccina
    theme(legend.position = "bottom", plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
        legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.route.total.dose.pdf", map.vac.total.dose,
    width = 250, height = 250, units = "mm",
```

```
    dpi = 300, bg = "white")
map.vac.total.dose
```

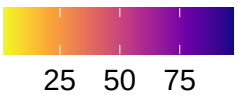



```
# Producing the final input data
ltla.vac.data <- left_join(ltla.vac.data,
    acc.all, by = c("LTLA19CD", "LTLA19NM"))
saveRDS(ltla.vac.data, "final.input.data.rda")

map.car.dist <- ggplot(ltla.vac.data) + geom_sf(aes(fill = car.dist/1000),
    color = NA) + geom_sf(colour = "white",
    size = 0.1, fill = NA) + scale_fill_viridis_c(option = "plasma",
    direction = -1) + theme_bw() + labs(fill = "Distance (km) to the nearest vaccination centres by c
    theme(legend.position = "bottom", plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
        legend.text = element_text(size = rel(1))) +
```

```
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.car.dist.pdf", map.car.dist,
    width = 250, height = 250, units = "mm",
    dpi = 300, bg = "white")
map.car.dist
```

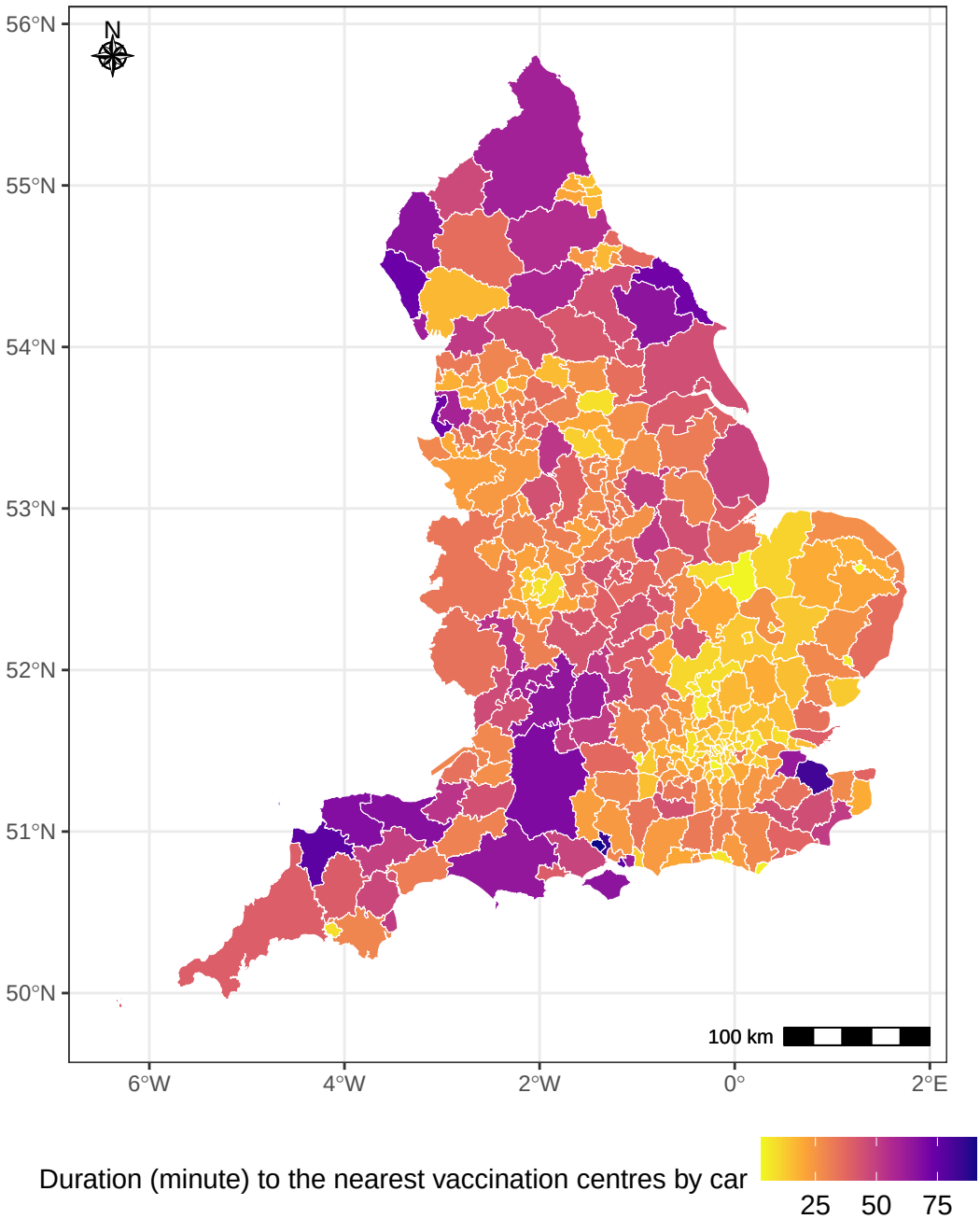


Distance (km) to the nearest vaccination centres by car




```
map.car.dur <- ggplot(ltla.vac.data) + geom_sf(aes(fill = car.dur/60),
    color = NA) + geom_sf(colour = "white",
    size = 0.1, fill = NA) + scale_fill_viridis_c(option = "plasma",
    direction = -1) + theme_bw() + labs(fill = "Duration (minute) to the nearest vaccination centres
    theme(legend.position = "bottom", plot.title = element_text(size = rel(1),
```

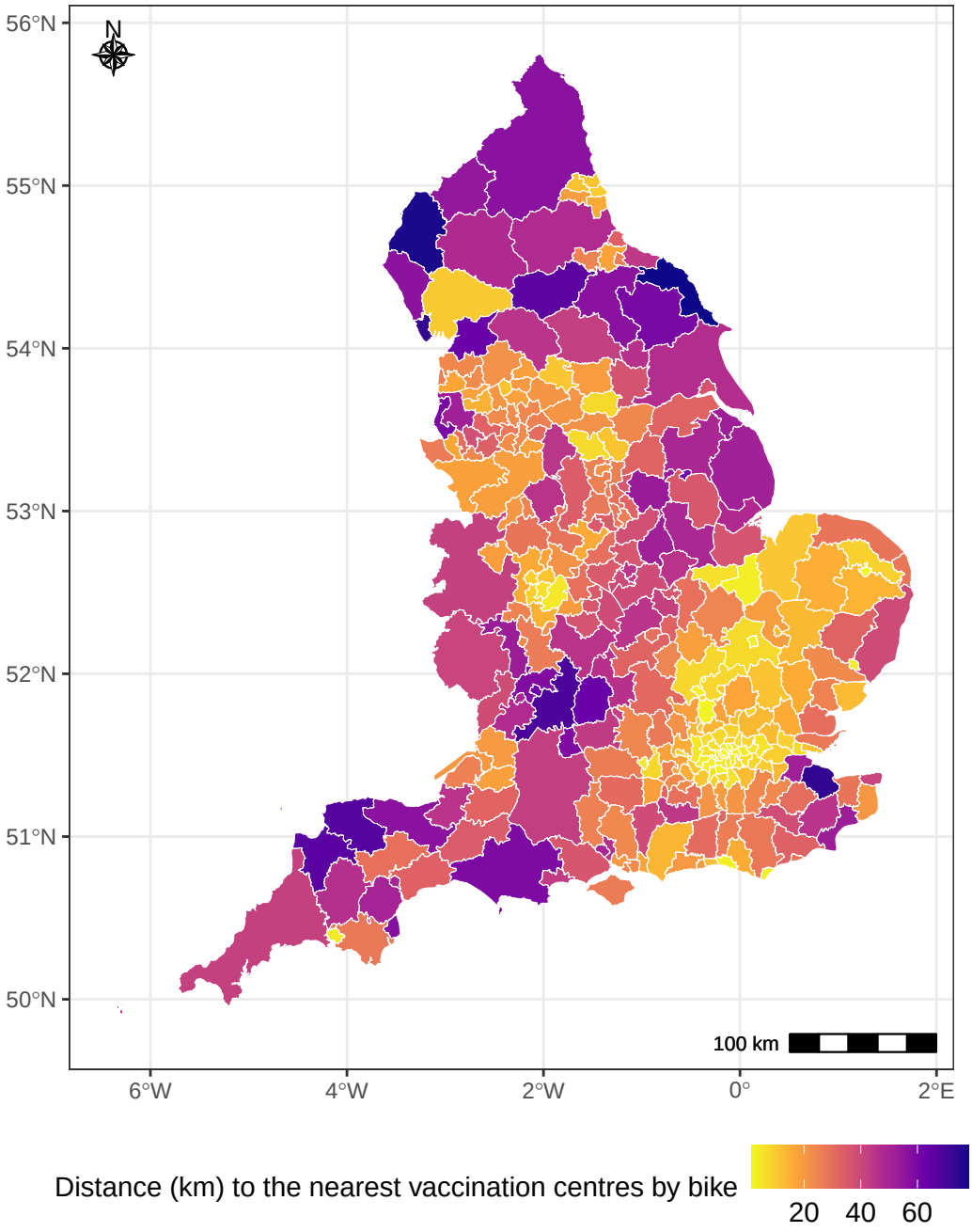
```
        face = "bold"), legend.title = element_text(size = rel(1)),
        legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.car.dur.pdf", map.car.dur, width = 250,
    height = 250, units = "mm", dpi = 300,
    bg = "white")
map.car.dur
```



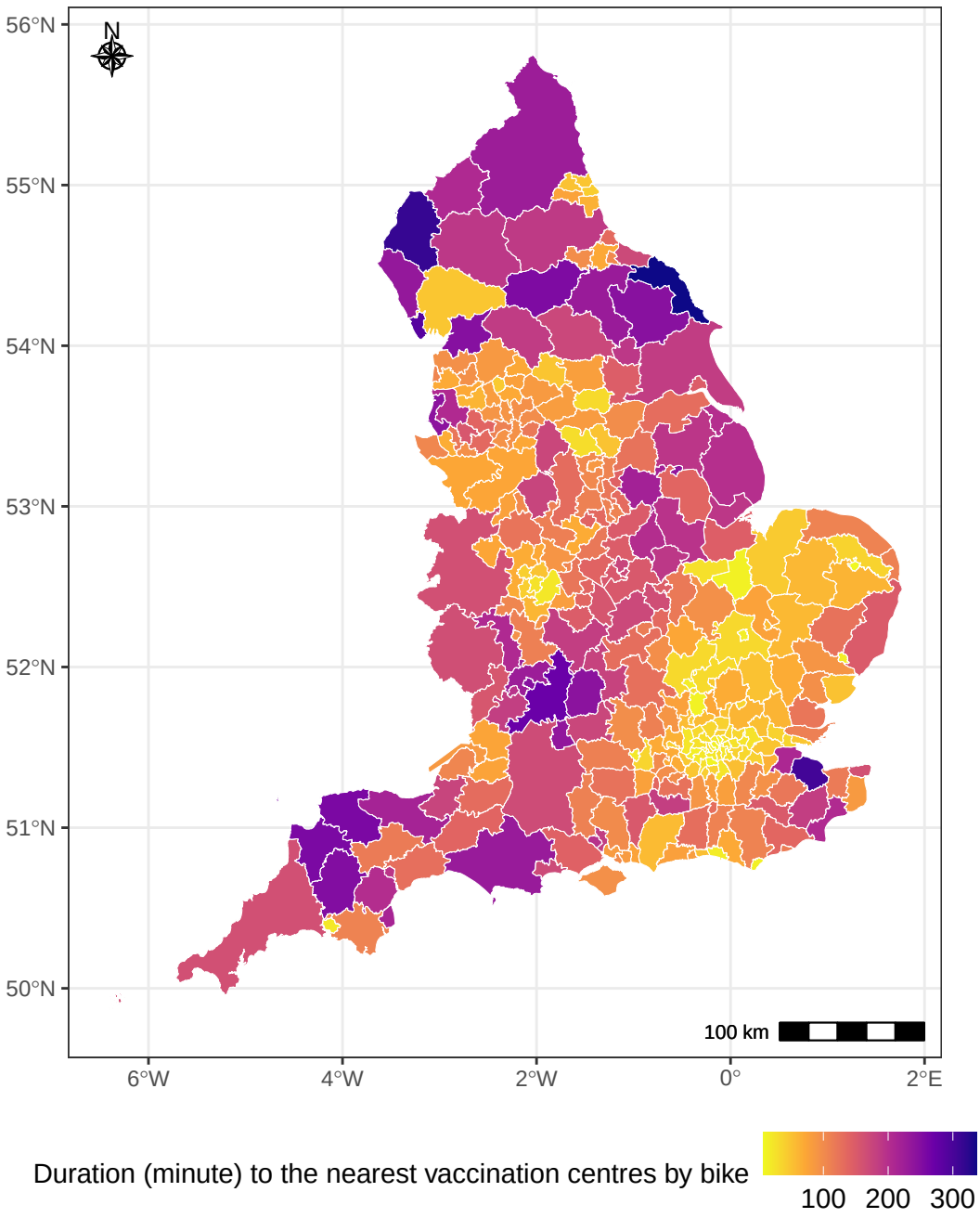
Duration (minute) to the nearest vaccination centres by car


```
map.bike.dist <- ggplot(ltla.vac.data) +
    geom_sf(aes(fill = bike.dist/1000), color = NA) +
    geom_sf(colour = "white", size = 0.1,
```

```
        fill = NA) + scale_fill_viridis_c(option = "plasma",
    direction = -1) + theme_bw() + labs(fill = "Distance (km) to the nearest vaccination centres by b
    theme(legend.position = "bottom", plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
        legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.bike.dist.pdf", map.bike.dist,
    width = 250, height = 250, units = "mm",
    dpi = 300, bg = "white")
map.bike.dist
```

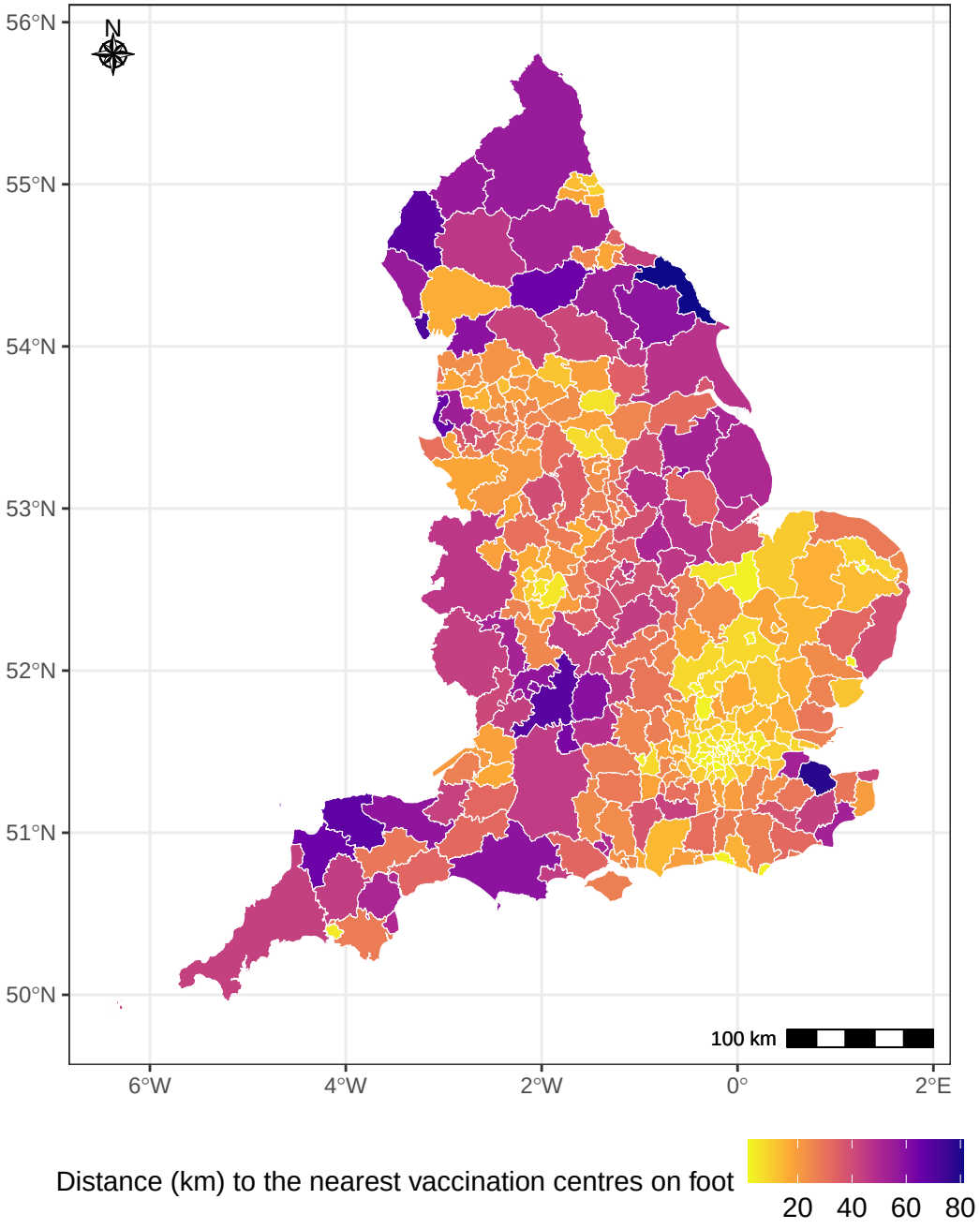


```
map.bike.dur <- ggplot(ltla.vac.data) + geom_sf(aes(fill = bike.dur/60),
    color = NA) + geom_sf(colour = "white",
    size = 0.1, fill = NA) + scale_fill_viridis_c(option = "plasma",
    direction = -1) + theme_bw() + labs(fill = "Duration (minute) to the nearest vaccination centres
    theme(legend.position = "bottom", plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
        legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.bike.dur.pdf", map.bike.dur,
    width = 250, height = 250, units = "mm",
    dpi = 300, bg = "white")
map.bike.dur
```



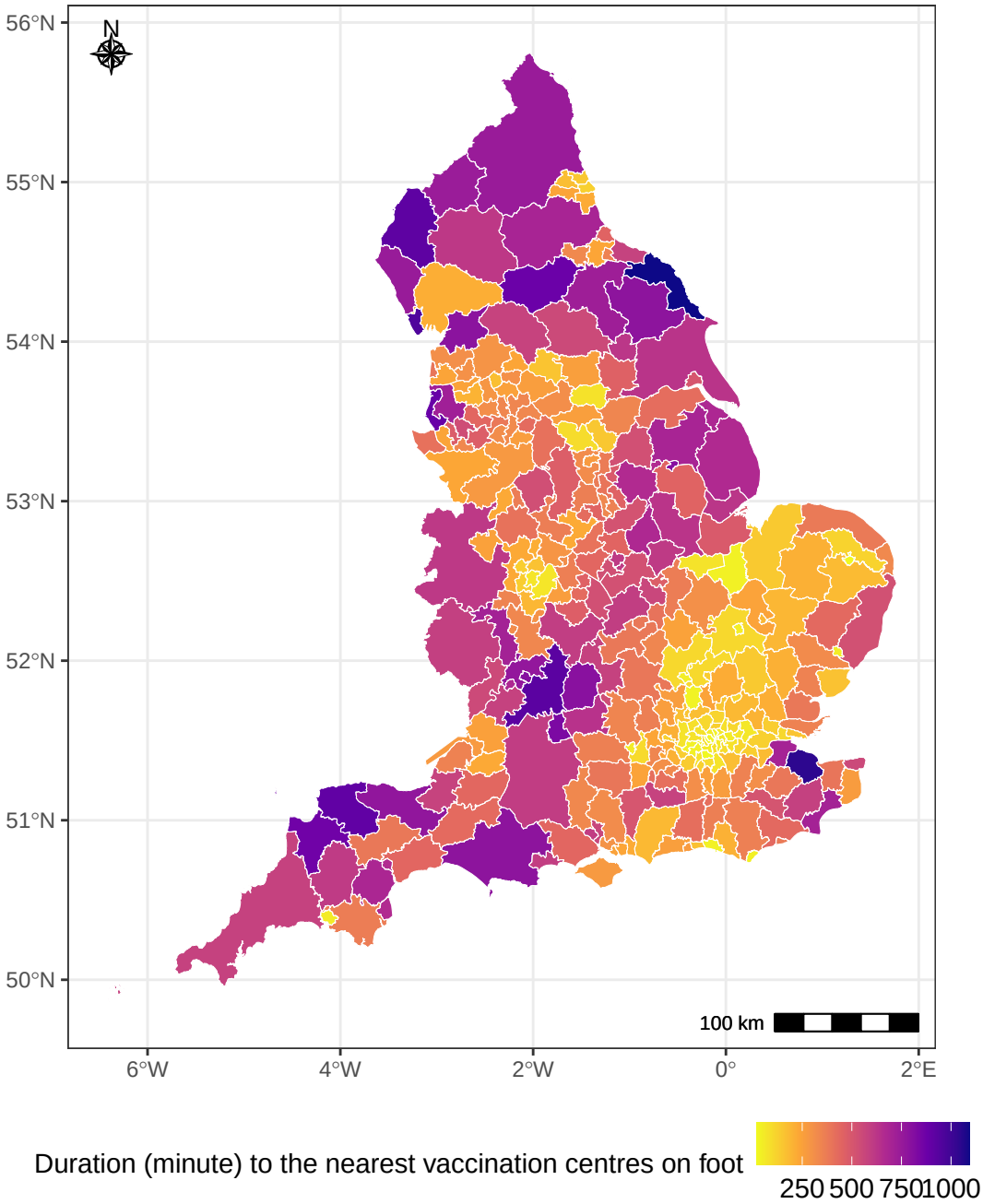

```
map.foot.dist <- ggplot(ltla.vac.data) +
    geom_sf(aes(fill = foot.dist/1000), color = NA) +
    geom_sf(colour = "white", size = 0.1,
        fill = NA) + scale_fill_viridis_c(option = "plasma",
    direction = -1) + theme_bw() + labs(fill = "Distance (km) to the nearest vaccination centres on f
    theme(legend.position = "bottom", plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
        legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
ggsave("map.foot.dist.pdf", map.foot.dist,
    width = 250, height = 250, units = "mm",
```

```
    dpi = 300, bg = "white")
map.foot.dist
```




```
map.foot.dur <- ggplot(ltla.vac.data) + geom_sf(aes(fill = foot.dur/60),
    color = NA) + geom_sf(colour = "white",
    size = 0.1, fill = NA) + scale_fill_viridis_c(option = "plasma",
    direction = -1) + theme_bw() + labs(fill = "Duration (minute) to the nearest vaccination centres
    theme(legend.position = "bottom", plot.title = element_text(size = rel(1),
        face = "bold"), legend.title = element_text(size = rel(1)),
        legend.text = element_text(size = rel(1))) +
    annotation_scale(location = "br", height = unit(0.25,
        "cm")) + annotation_scale(location = "br",
    height = unit(0.25, "cm")) + annotation_north_arrow(location = "tl",
    style = north_arrow_nautical, height = unit(0.75,
        "cm"), width = unit(0.75, "cm"))
```

```
ggsave("map.foot.dur.pdf", map.foot.dur,
    width = 250, height = 250, units = "mm",
    dpi = 300, bg = "white")
map.foot.dur
```

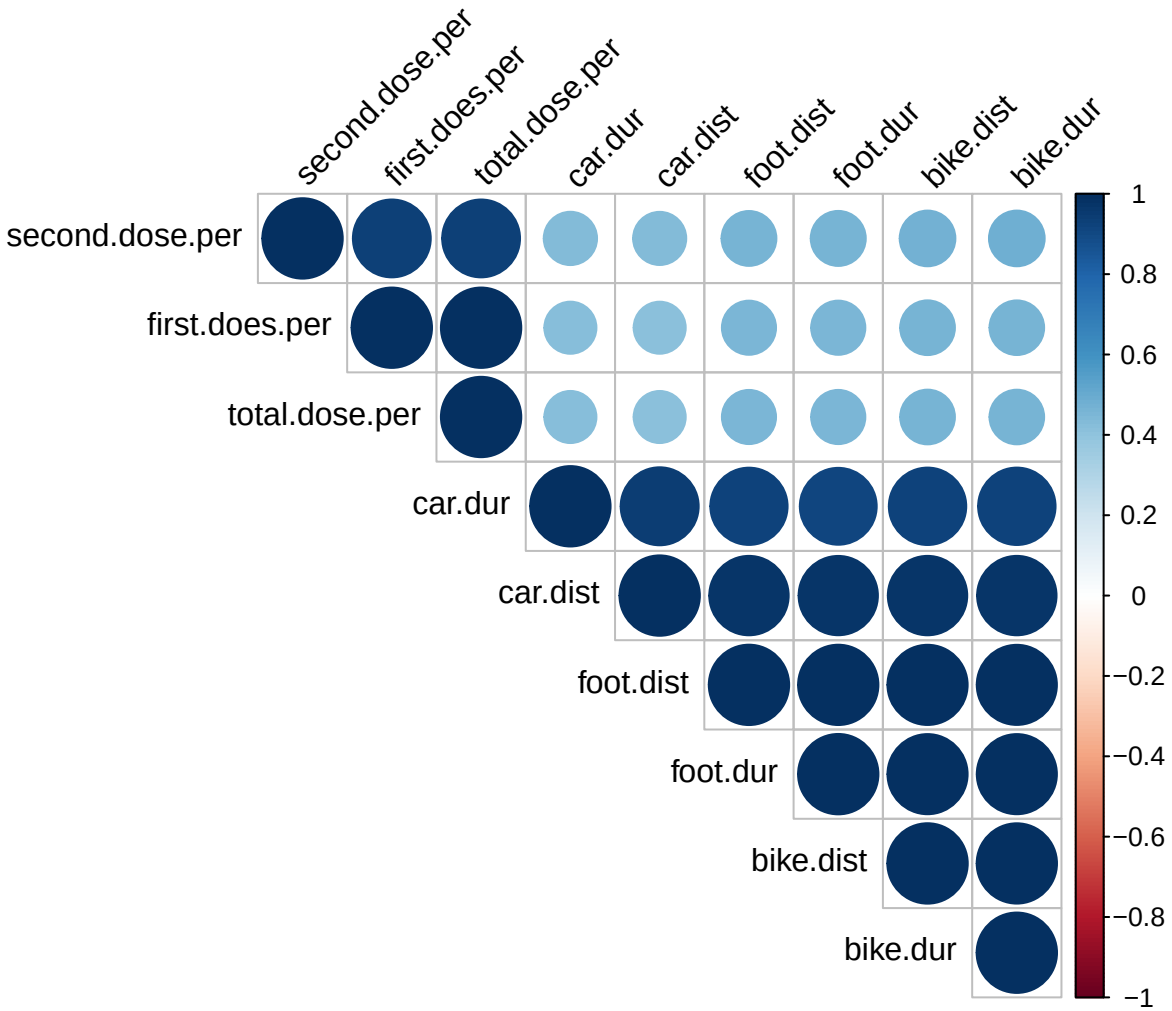



```
rm(list = ls())
```

6. Correlation test

```
input.data <- readRDS("final.input.data.rda")
input.matrix <- as.data.frame(input.data[,
    c(9:11, 13:18)])
input.matrix <- input.matrix[-c(10)]
```

```
require(BBmisc)
std.input.matrix <- normalize(input.matrix,
    method = "standardize", range = c(0,
        1), margin = 1L, on.constant = "quiet")

input.matrix <- as.matrix(input.matrix)
std.input.matrix <- as.matrix(std.input.matrix)

require(corrplot)
cor.matrix <- cor(input.matrix, method = c("spearman"))
std.cor.matrix <- cor(std.input.matrix, method = c("spearman"))

plot.cor <- corrplot(cor.matrix, type = "upper",
    order = "hclust", tl.col = "black", tl.srt = 45)
plot.std.cor <- corrplot(std.cor.matrix,
    type = "upper", order = "hclust", tl.col = "black",
    tl.srt = 45)
```




```
pdf("cor.matrix.pdf")
plot.cor
```

```
## $corr
##                 second.dose.per first.does.per total.dose.per   car.dur
## second.dose.per       1.0000000      0.9383020      0.9383020 0.4396916
## first.does.per        0.9383020      1.0000000      1.0000000 0.4201666
```

```
## total.dose.per        0.9383020      1.0000000      1.0000000 0.4201666
## car.dur               0.4396916      0.4201666      0.4201666 1.0000000
## car.dist              0.4371243      0.4195434      0.4195434 0.9415582
## foot.dist             0.4661050      0.4535021      0.4535021 0.9234119
## foot.dur              0.4683834      0.4543992      0.4543992 0.9146081
## bike.dist             0.4798857      0.4634469      0.4634469 0.9277916
## bike.dur              0.4800792      0.4618280      0.4618280 0.9278409
##                   car.dist foot.dist  foot.dur bike.dist  bike.dur
## second.dose.per 0.4371243 0.4661050 0.4683834 0.4798857 0.4800792
## first.does.per  0.4195434 0.4535021 0.4543992 0.4634469 0.4618280
## total.dose.per  0.4195434 0.4535021 0.4543992 0.4634469 0.4618280
## car.dur         0.9415582 0.9234119 0.9146081 0.9277916 0.9278409
## car.dist        1.0000000 0.9761575 0.9734592 0.9760895 0.9763130
## foot.dist       0.9761575 1.0000000 0.9985683 0.9950484 0.9935438
## foot.dur        0.9734592 0.9985683 1.0000000 0.9939872 0.9932866
## bike.dist       0.9760895 0.9950484 0.9939872 1.0000000 0.9981548
## bike.dur        0.9763130 0.9935438 0.9932866 0.9981548 1.0000000
##
## $corrPos
##              xName           yName x y      corr
## 1  second.dose.per second.dose.per 1 9 1.0000000
## 2   first.does.per second.dose.per 2 9 0.9383020
## 3   first.does.per  first.does.per 2 8 1.0000000
## 4   total.dose.per second.dose.per 3 9 0.9383020
## 5   total.dose.per  first.does.per 3 8 1.0000000
## 6   total.dose.per  total.dose.per 3 7 1.0000000
## 7          car.dur second.dose.per 4 9 0.4396916
## 8          car.dur  first.does.per 4 8 0.4201666
## 9          car.dur  total.dose.per 4 7 0.4201666
## 10         car.dur         car.dur 4 6 1.0000000
## 11        car.dist second.dose.per 5 9 0.4371243
## 12        car.dist  first.does.per 5 8 0.4195434
## 13        car.dist  total.dose.per 5 7 0.4195434
## 14        car.dist         car.dur 5 6 0.9415582
## 15        car.dist        car.dist 5 5 1.0000000
## 16       foot.dist second.dose.per 6 9 0.4661050
## 17       foot.dist  first.does.per 6 8 0.4535021
## 18       foot.dist  total.dose.per 6 7 0.4535021
## 19       foot.dist         car.dur 6 6 0.9234119
## 20       foot.dist        car.dist 6 5 0.9761575
## 21       foot.dist       foot.dist 6 4 1.0000000
## 22        foot.dur second.dose.per 7 9 0.4683834
## 23        foot.dur  first.does.per 7 8 0.4543992
## 24        foot.dur  total.dose.per 7 7 0.4543992
## 25        foot.dur         car.dur 7 6 0.9146081
## 26        foot.dur        car.dist 7 5 0.9734592
## 27        foot.dur       foot.dist 7 4 0.9985683
## 28        foot.dur        foot.dur 7 3 1.0000000
## 29       bike.dist second.dose.per 8 9 0.4798857
## 30       bike.dist  first.does.per 8 8 0.4634469
## 31       bike.dist  total.dose.per 8 7 0.4634469
## 32       bike.dist         car.dur 8 6 0.9277916
## 33       bike.dist        car.dist 8 5 0.9760895
## 34       bike.dist       foot.dist 8 4 0.9950484
## 35       bike.dist        foot.dur 8 3 0.9939872
## 36       bike.dist       bike.dist 8 2 1.0000000
```

```
## 37        bike.dur second.dose.per 9 9 0.4800792
## 38        bike.dur  first.does.per 9 8 0.4618280
## 39        bike.dur  total.dose.per 9 7 0.4618280
## 40        bike.dur         car.dur 9 6 0.9278409
## 41        bike.dur        car.dist 9 5 0.9763130
## 42        bike.dur       foot.dist 9 4 0.9935438
## 43        bike.dur        foot.dur 9 3 0.9932866
## 44        bike.dur       bike.dist 9 2 0.9981548
## 45        bike.dur        bike.dur 9 1 1.0000000
##
## $arg
## $arg$type
## [1] "upper"
```

```
dev.off()
```

```
## pdf
##   2
```

```
pdf("std.cor.matrix.pdf")
plot.std.cor
```

```
## $corr
##                 second.dose.per first.does.per total.dose.per   car.dur
## second.dose.per       1.0000000      0.9383020      0.9383020 0.4396916
## first.does.per        0.9383020      1.0000000      1.0000000 0.4201666
## total.dose.per        0.9383020      1.0000000      1.0000000 0.4201666
## car.dur               0.4396916      0.4201666      0.4201666 1.0000000
## car.dist              0.4371243      0.4195434      0.4195434 0.9415582
## foot.dist             0.4661050      0.4535021      0.4535021 0.9234119
## foot.dur              0.4683834      0.4543992      0.4543992 0.9146081
## bike.dist             0.4798857      0.4634469      0.4634469 0.9277916
## bike.dur              0.4800792      0.4618280      0.4618280 0.9278409
##                  car.dist foot.dist  foot.dur bike.dist  bike.dur
## second.dose.per 0.4371243 0.4661050 0.4683834 0.4798857 0.4800792
## first.does.per  0.4195434 0.4535021 0.4543992 0.4634469 0.4618280
## total.dose.per  0.4195434 0.4535021 0.4543992 0.4634469 0.4618280
## car.dur         0.9415582 0.9234119 0.9146081 0.9277916 0.9278409
## car.dist        1.0000000 0.9761575 0.9734592 0.9760895 0.9763130
## foot.dist       0.9761575 1.0000000 0.9985683 0.9950484 0.9935438
## foot.dur        0.9734592 0.9985683 1.0000000 0.9939872 0.9932866
## bike.dist       0.9760895 0.9950484 0.9939872 1.0000000 0.9981548
## bike.dur        0.9763130 0.9935438 0.9932866 0.9981548 1.0000000
##
## $corrPos
##              xName           yName x y      corr
## 1  second.dose.per second.dose.per 1 9 1.0000000
## 2   first.does.per second.dose.per 2 9 0.9383020
## 3   first.does.per  first.does.per 2 8 1.0000000
## 4   total.dose.per second.dose.per 3 9 0.9383020
## 5   total.dose.per  first.does.per 3 8 1.0000000
## 6   total.dose.per  total.dose.per 3 7 1.0000000
## 7          car.dur second.dose.per 4 9 0.4396916
## 8          car.dur  first.does.per 4 8 0.4201666
## 9          car.dur  total.dose.per 4 7 0.4201666
## 10         car.dur         car.dur 4 6 1.0000000
## 11        car.dist second.dose.per 5 9 0.4371243
## 12        car.dist  first.does.per 5 8 0.4195434
```

```
## 13        car.dist  total.dose.per 5 7 0.4195434
## 14        car.dist         car.dur 5 6 0.9415582
## 15        car.dist        car.dist 5 5 1.0000000
## 16       foot.dist second.dose.per 6 9 0.4661050
## 17       foot.dist  first.does.per 6 8 0.4535021
## 18       foot.dist  total.dose.per 6 7 0.4535021
## 19       foot.dist         car.dur 6 6 0.9234119
## 20       foot.dist        car.dist 6 5 0.9761575
## 21       foot.dist       foot.dist 6 4 1.0000000
## 22        foot.dur second.dose.per 7 9 0.4683834
## 23        foot.dur  first.does.per 7 8 0.4543992
## 24        foot.dur  total.dose.per 7 7 0.4543992
## 25        foot.dur         car.dur 7 6 0.9146081
## 26        foot.dur        car.dist 7 5 0.9734592
## 27        foot.dur       foot.dist 7 4 0.9985683
## 28        foot.dur        foot.dur 7 3 1.0000000
## 29       bike.dist second.dose.per 8 9 0.4798857
## 30       bike.dist  first.does.per 8 8 0.4634469
## 31       bike.dist  total.dose.per 8 7 0.4634469
## 32       bike.dist         car.dur 8 6 0.9277916
## 33       bike.dist        car.dist 8 5 0.9760895
## 34       bike.dist       foot.dist 8 4 0.9950484
## 35       bike.dist        foot.dur 8 3 0.9939872
## 36       bike.dist       bike.dist 8 2 1.0000000
## 37        bike.dur second.dose.per 9 9 0.4800792
## 38        bike.dur  first.does.per 9 8 0.4618280
## 39        bike.dur  total.dose.per 9 7 0.4618280
## 40        bike.dur         car.dur 9 6 0.9278409
## 41        bike.dur        car.dist 9 5 0.9763130
## 42        bike.dur       foot.dist 9 4 0.9935438
## 43        bike.dur        foot.dur 9 3 0.9932866
## 44        bike.dur       bike.dist 9 2 0.9981548
## 45        bike.dur        bike.dur 9 1 1.0000000
##
## $arg
## $arg$type
## [1] "upper"
```

```
dev.off()
```

```
## pdf
##   2
```

7. Regression model? To be continued...

8. Spatial regression model? To be continued...

Findings

- The analysis result illustrates i) the spatial variation in accessibility to vaccination centres in England, and also ii) explored the relative poor accessibility to vaccination centre with the areas in peripheral areas nearby major cities across England, particularly travel duration by bike and on foot.
- "The high drive times that appear with these remote areas may be a barrier for uptake, reflecting isolated communities who are unable or discouraged to make this journey."
- Separate the urban areas and peripheral and rural areas?

- Does it necessary to explore the association between accessibility to vaccination centres and age-specific vaccination rates by LTLA or MSOA level?