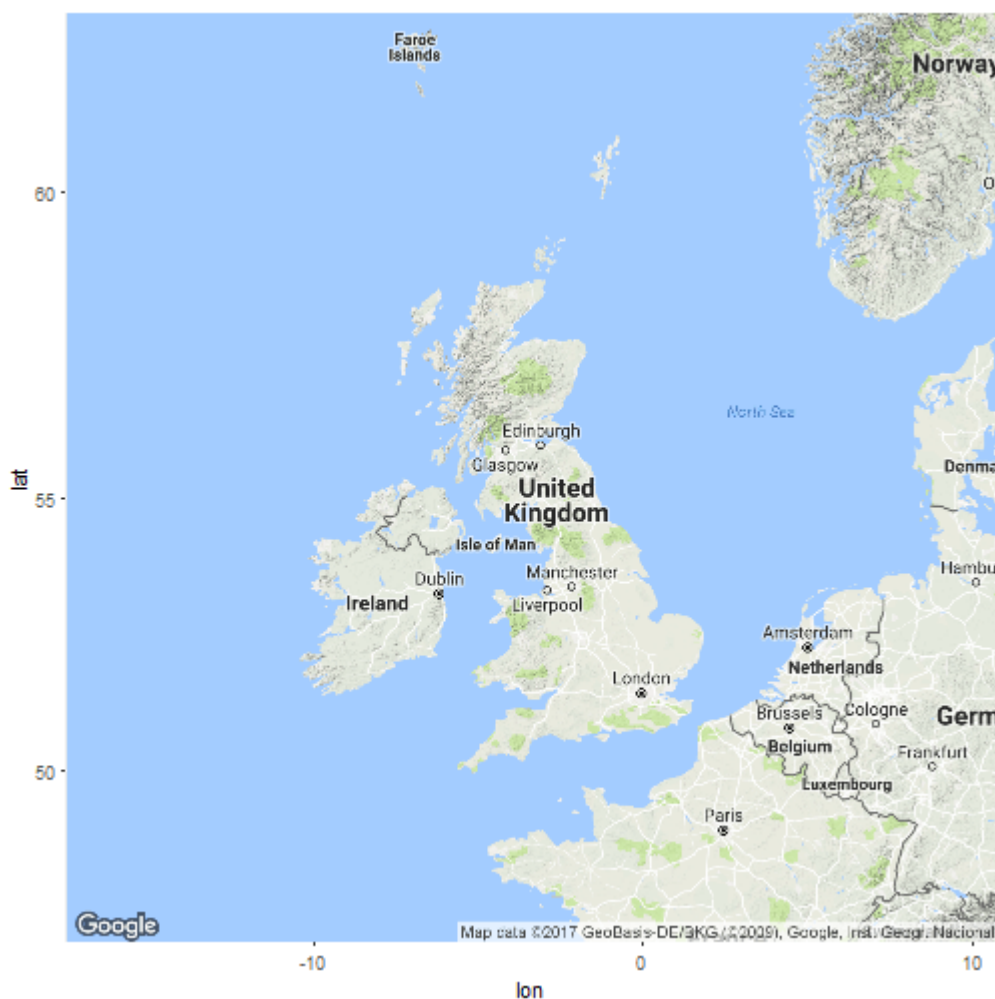


Spatial analysis with ggmap Solutions (part-1)

Below are the solutions to [these](#) exercises on Spatial Analysis with R.

```
#####  
#           #  
#   Exercise 1   #  
#           #  
#####  
library(ggmap)  
uk <- get_map(location="United Kingdom", zoom=5,  
maptype='terrain', source='google', color='color')  
ggmap(uk)
```

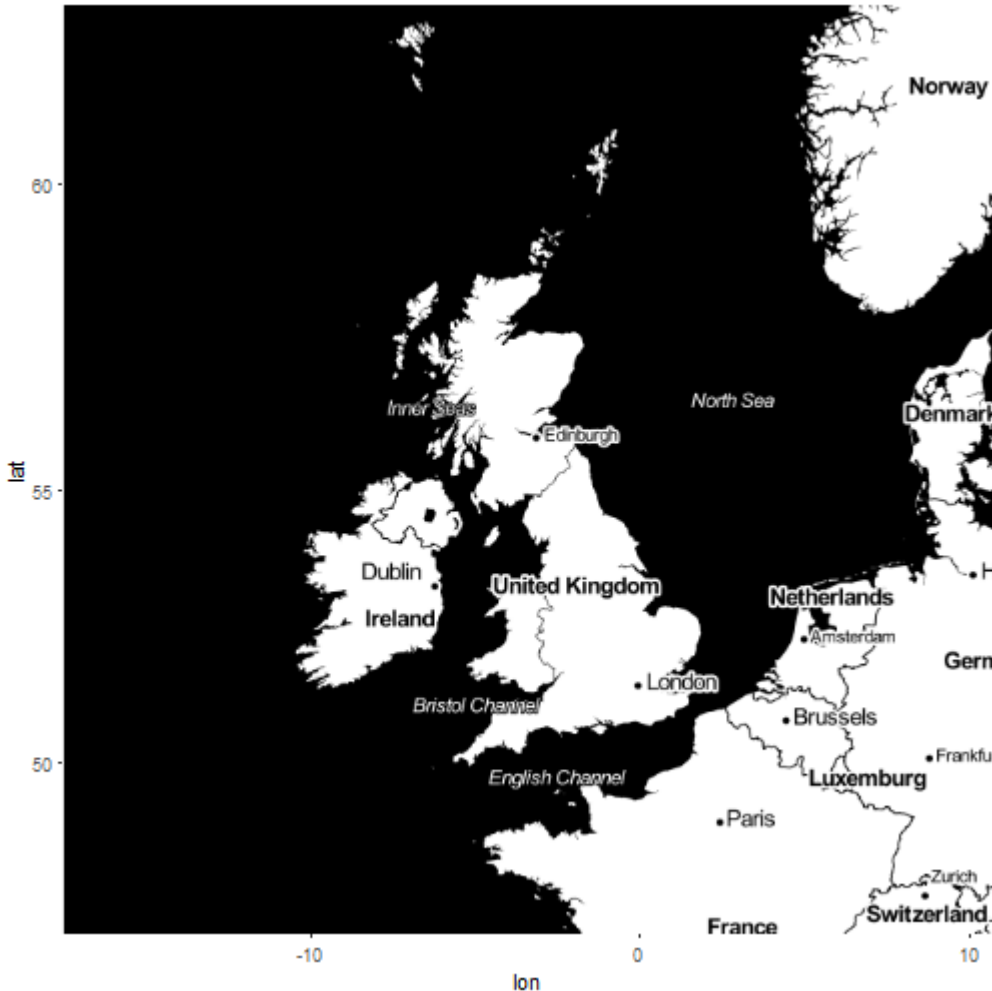


```
#####
```

```

#           #
#   Exercise 2   #
#           #
#####
uk  <-  get_map(location="United Kingdom", zoom=5,
mptype='toner', source='stamen', color='color')
ggmap(uk)

```



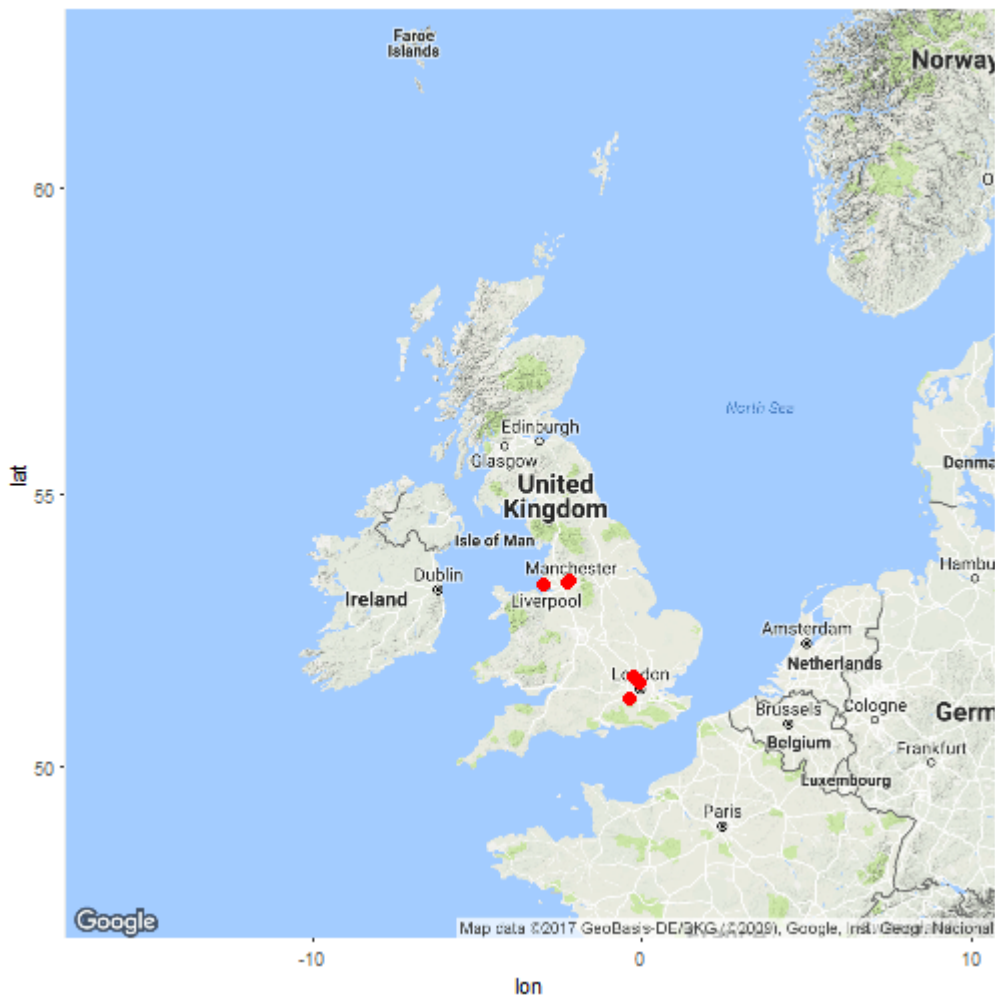
```

#####
#           #
#   Exercise 3   #
#           #
#####

clubs<-c("Arsenal FC", "Manchester City FC", "Manchester
United FC",
         "Liverpool FC", "Chelsea FC", "Tottenham Hotspur FC")
coord<-geocode(clubs)
clubs<-cbind(clubs,coord)

```

```
uk <- get_map(location="United Kingdom", zoom=5,
maptype='terrain', source='google', color='color')
ggmap(uk) +
  geom_point(
    aes(x = lon, y = lat),
    data = clubs, colour = "red", size = 3
  )
```



```
#####
```

```
# #
# Exercise 4 #
# #
```

```
#####
```

```
revgeocode(c(-0.119543, 51.50332), output="address")
```

```
## [1] "30 The Queen's Walk, Lambeth, London SE1 8XX, UK"
```

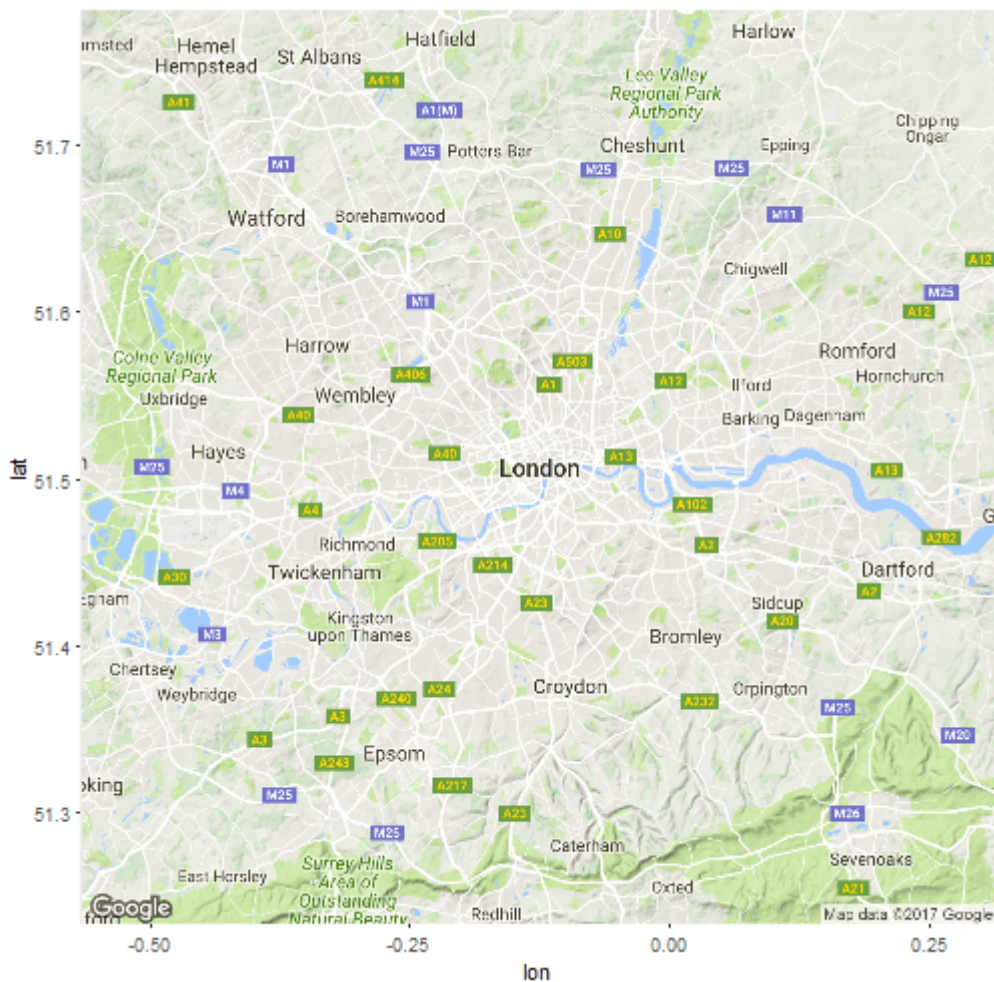
```
#####
```

```
# #
```

```

# Exercise 5 #
# #
#####
london <- get_map(location="london", zoom=10,
maptype='terrain', source='google', color='color')
ggmap(london)

```



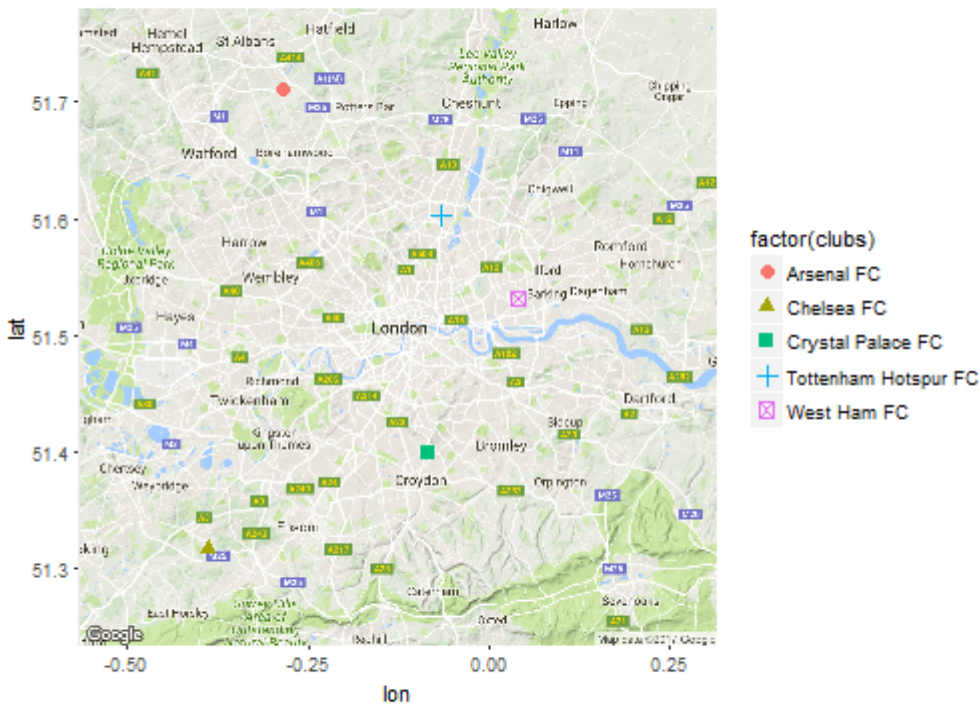
```

#####
# #
# Exercise 6 #
# #
#####
clubs<-c("Arsenal FC", "Tottenham Hotspur FC", "Chelsea FC",
"West Ham FC", "Crystal Palace FC")
coord<-geocode(clubs)
clubs<-cbind(clubs,coord)

london <- get_map(location="london", zoom=10,
maptype='terrain', source='google', color='color')
ggmap(london) +

```

```
geom_point(
  aes(x = lon, y = lat, shape = factor(clubs),
  colour=factor(clubs)),
  data = clubs, size = 3
)
```



```
#####
```

```
# #
# Exercise 7 #
# #
```

```
#####
```

```
distance<-mapdist("Emirates Stadium, London", "Wembley,
London", mode="driving", output="simple")
distance$km
```

```
## [1] 19.526
```

```
distance$minutes
```

```
## [1] 41.41667
```

```
#####
```

```
# #
```

```
# Exercise 8 #
```

```
# #
```

```
#####
```

```
coords<-geocode(c("Emirates Stadium, London", "Wembley,  
London"))
```

```
calc_zoom(lon, lat, coords)
```

```
## [1] 16
```

```
#####
```

```
# #
```

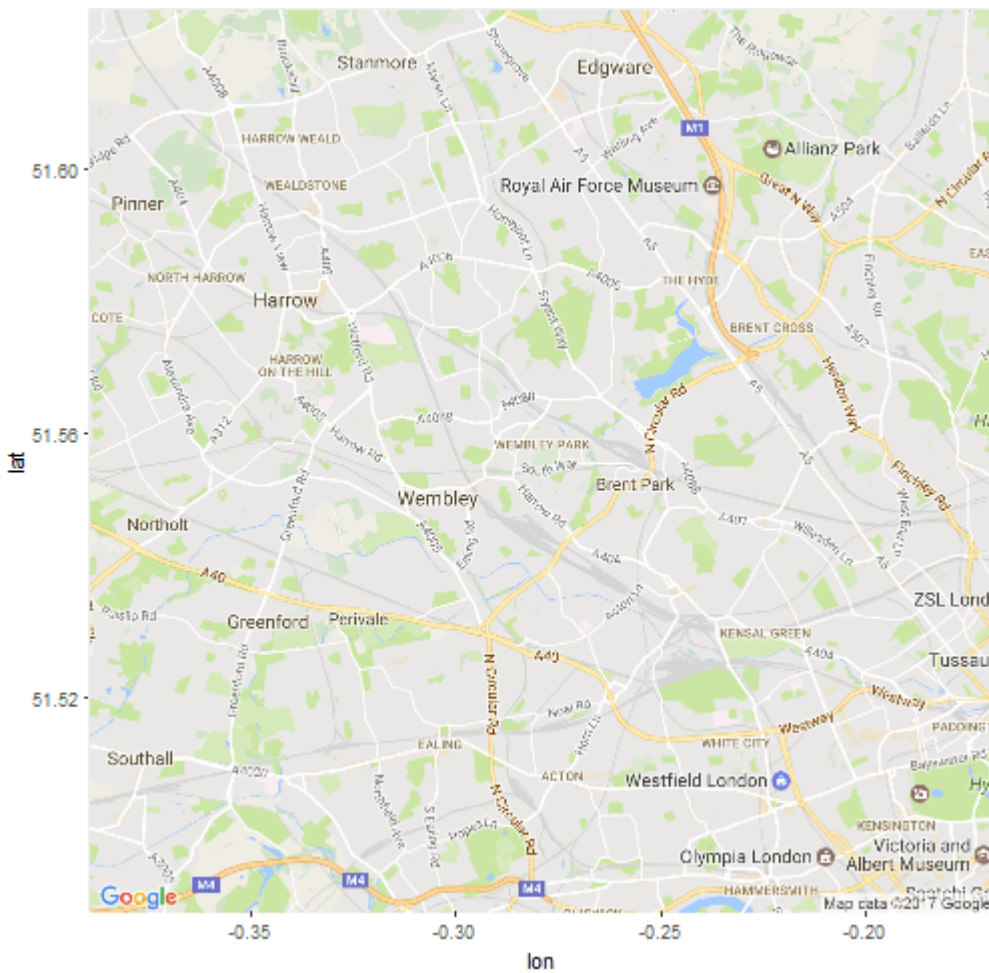
```
# Exercise 9 #
```

```
# #
```

```
#####
```

```
map<-get_map("Wembley", zoom = 12, maptype='roadmap',  
source='google', color='color')
```

```
ggmap(map)
```

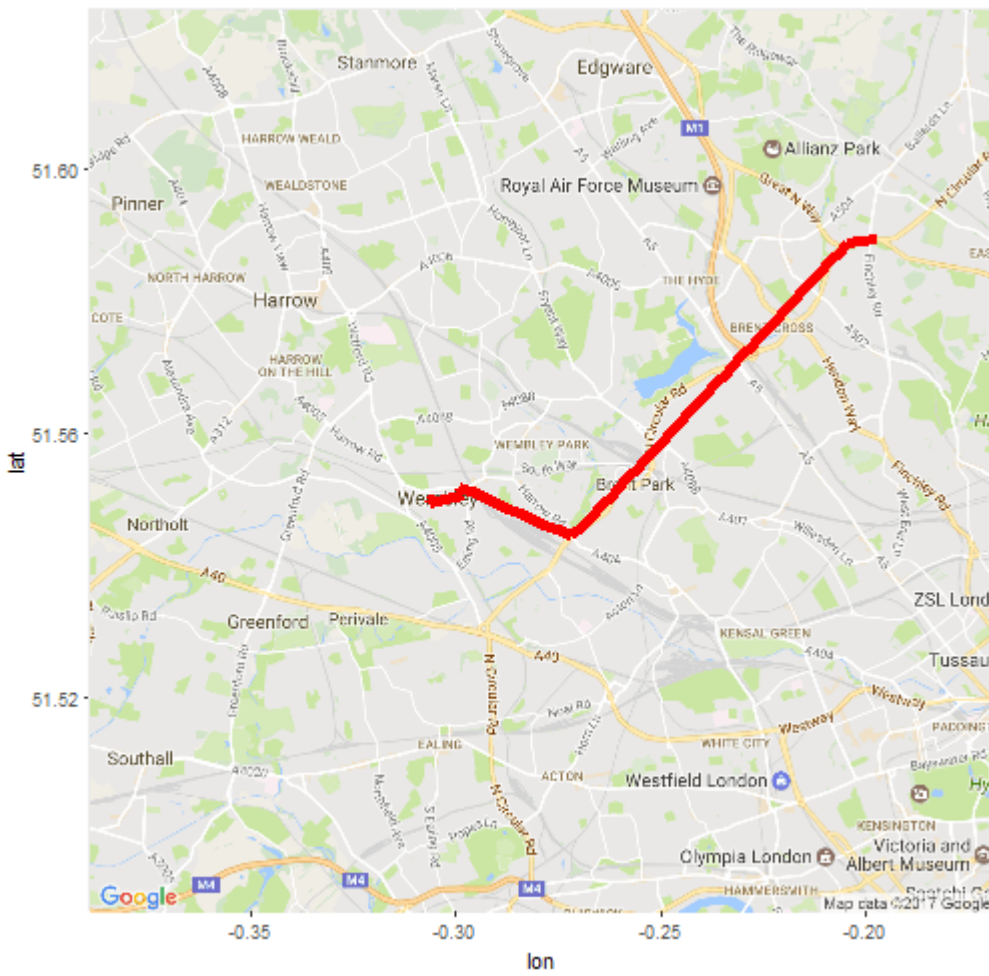
```
#####
```

```
# #
# Exercise 10 #
# #
```

```
#####
```

```
Emirates2Wembley<- route("Emirates Stadium, London", "Wembley,
London")
map<-get_map("Wembley", zoom = 12, maptype='roadmap',
source='google', color='color')
ggmap(map)+
  geom_segment(
    aes(x = startLon, y = startLat, xend = endLon, yend =
endLat),
    colour ="red", size = 2, data = Emirates2Wembley
  )
```

```
## Warning: Removed 9 rows containing missing values
(geom_segment).
```



Spatial analysis with ggmap Exercises (part-1)



R has many powerful libraries to handle spatial data, and the things that R can do with maps can only grow. This exercise tries to demonstrate a few basic functionalities of the ggmap package in R while dealing with raster images.

The ggmap package can be used to access maps from the Google Maps API and other APIs as raster layers and perform various raster operations on it. Moreover, many other features such as points, polygons, lines etc. can be added to the basemap using layered grammar of graphics on the lines of the ggplot2 package. In addition, the package provides Geocoding facilities using the popular Google API.

Answers to the exercises are available [here](#). If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Please install and load the package ggmap before starting the exercises.

Exercise 1

Get a basemap for United Kingdom from Google Maps API and plot the map. Keep the zoom level in such a way that entire UK is visible in the map.

Exercise 2

Repeat Exercise 1 to get a Stamen map for United Kingdom. Keep the maptype as toner.

Exercise 3

Consider the following football clubs currently playing in the English Premier League, Arsenal FC, Manchester City FC, Manchester United FC, Liverpool FC, Chelsea FC and Tottenham Hotspur FC. Please locate these clubs on the basemap obtained in Exercise 1 as red points.

Exercise 4

Consider the geolocation (-0.119543, 51.50332). Find out an address on the map that corresponds to this location..

Exercise 5

Get a basemap for London from Google Maps API and plot it.

Choose an appropriate Zoom Level.

Exercise 6

Consider the following London based Football clubs: Arsenal FC, Tottenham Hotspur FC, Chelsea FC, West Ham FC and Crystal Palace FC. Like in Exercise 4, plot these clubs on the London map generated in Exercise 5. Use different colors and shape for each club.

Exercise 7

Calculate the distance(while driving) from Emirates Stadium, London to Wembley, London in terms of kilometers and time taken (minutes).

Exercise 8

Calculate the maximum zoom level which can be used to fetch a basemap such that both Emirates Stadium and Wembley are included in the map.

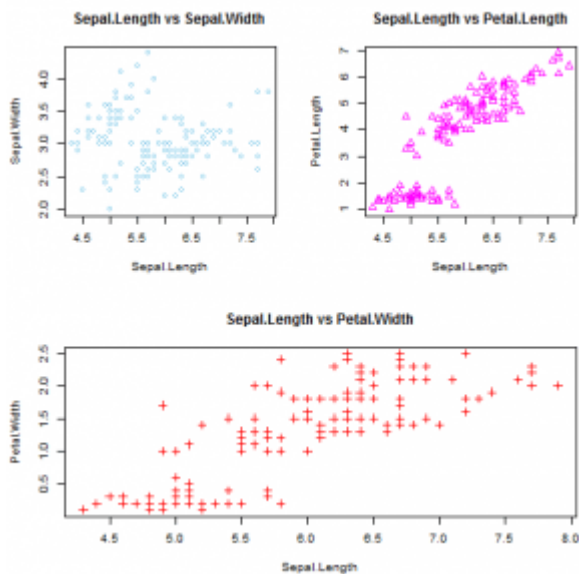
Exercise 9

Get a basemap around Wembley at a zoom level 12 from Google Maps API. Keep the maptype as roadmap.

Exercise 10

Draw the driving route from Emirates Stadium, London to Wembley, London on the basemap obtained in Example 9. Keep the color of the route as red.

Multipanel Graphics in R **(part 1)**



In many situations, we require that several plots are placed in the same figure as subplots. R has various ways of doing it. Base Graphics has three different ways to draw subplots, i.e. `mfrow`, `layout` and `split.screen`, with increasing degree of complexity, and, at the same time, with increased control over the plot elements.

This example introduces the `mfrow`, `mfcoll` and `layout` functions in Base Graphics. We use the familiar iris dataset for the illustrations.

Answers to the exercises are available [here](#). If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1

Consider the iris dataset, draw the following scatterplots, a) Sepal.Length vs Sepal.Width, b) Sepal.Length vs Petal.Length, and c) Sepal.Length vs Petal.Width. Annotate each scatterplot with a title. Use separate colors and plotting characters for each plot.

Exercise 2

Plot the three scatterplots in the same figure as subplots arranged in one row. Use `mfrow`.

Exercise 3

Plot the three scatterplots in the same figure as subplots arranged in one column. Use `mfrow`.

Exercise 4

Repeat the same scatterplots. Partition in such a way that the first row contains plots a and b, and the second row contains

plot c. Use mfrow.

Exercise 5

Repeat Exercise 2 with mfcpl.

Exercise 6

Repeat Exercise 3 with mfcpl.

Exercise 7

Repeat Exercise 4 with mfcpl.

Exercise 8

Repeat Exercise 2 with layout.

Exercise 9

Repeat Exercise 3 with layout.

Exercise 10

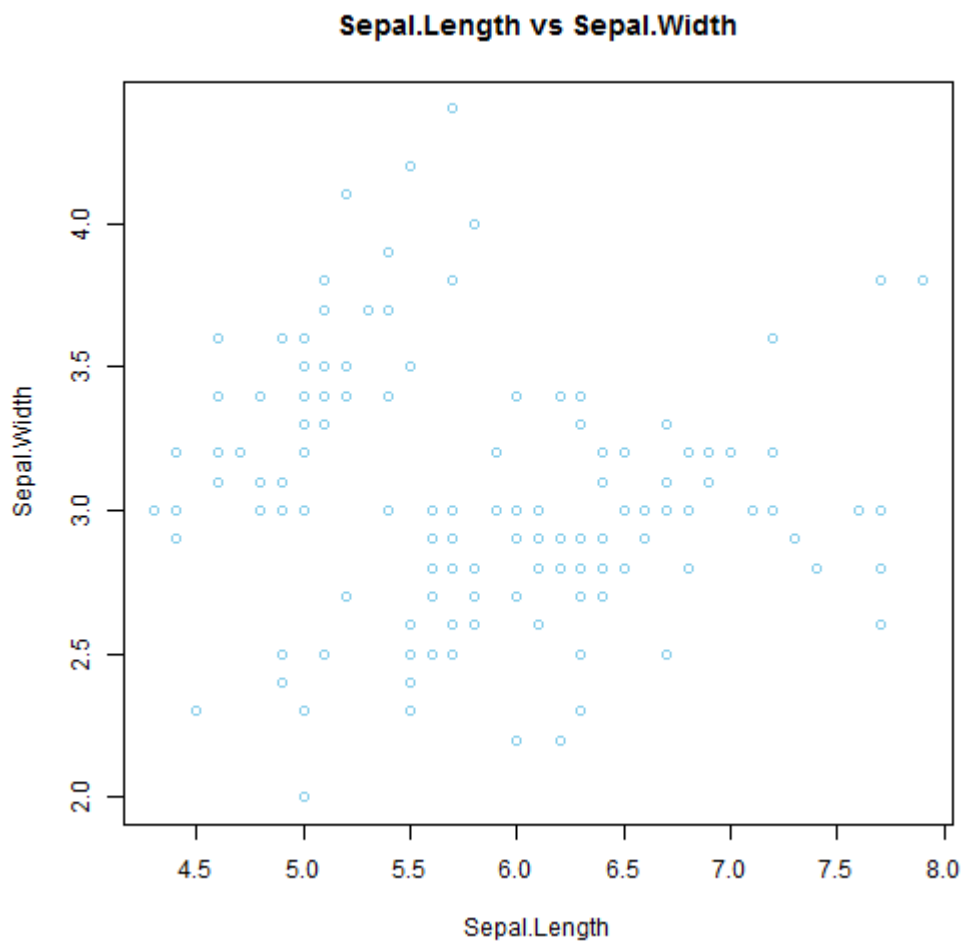
Repeat Exercise 4 with layout. In this case, let scatterplot c occupy the second row completely.

Multipanel Graphics in R (part 1) Solutions

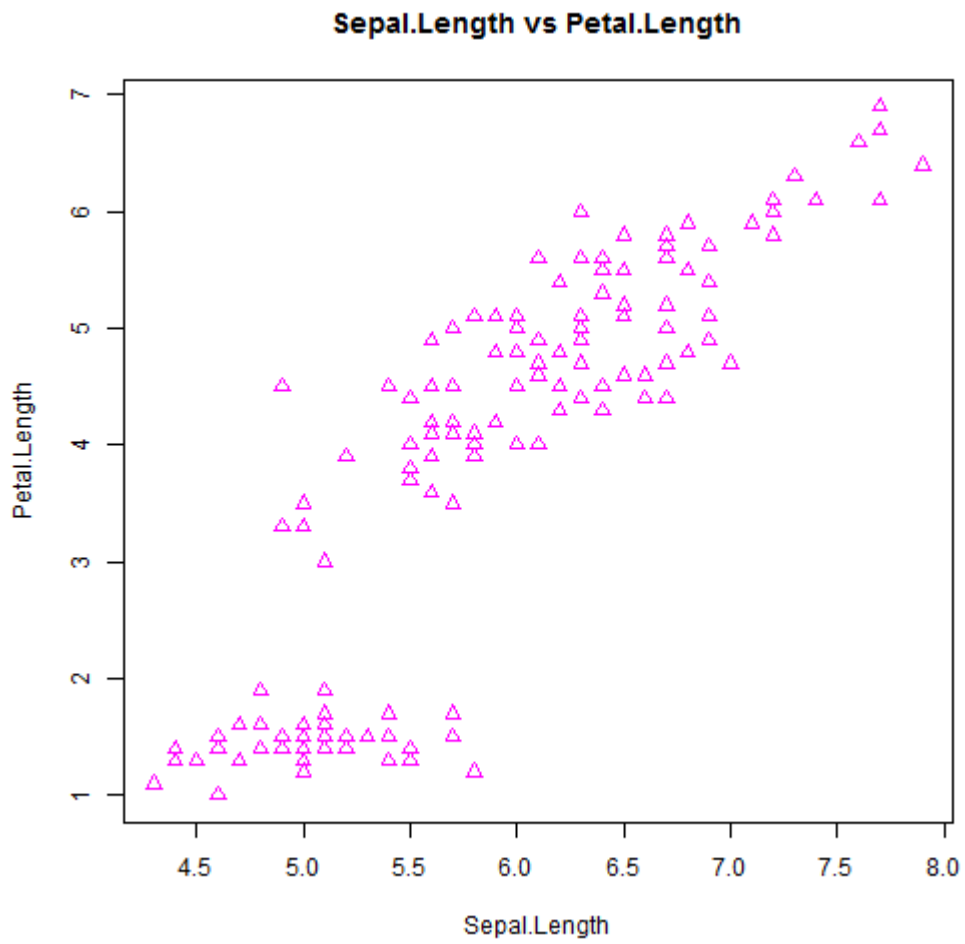
Below are the solutions to [these](#) exercises on Multipanel Graphics in R.

```
library(rgdal)
#####
#           #
#   Exercise 1   #
#           #
#####
attach(iris)
```

```
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)
title("Sepal.Length vs Sepal.Width")
```

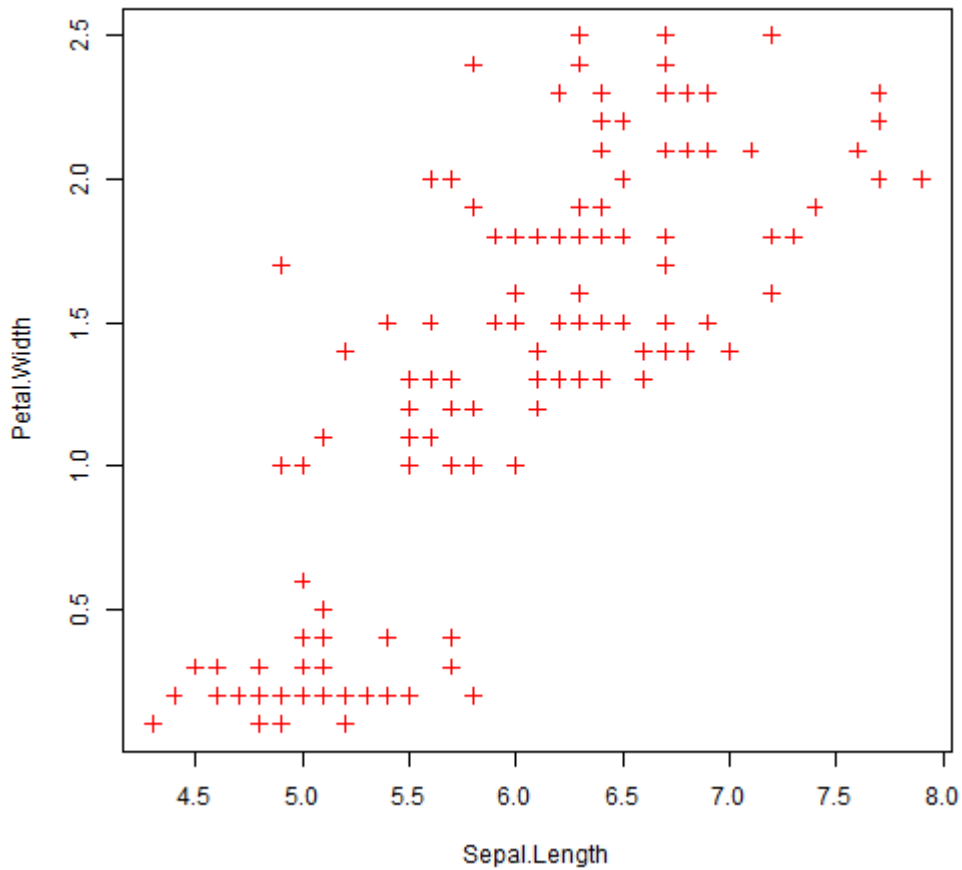


```
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)
title("Sepal.Length vs Petal.Length")
```



```
plot(Sepal.Length, Petal.Width, col='red', pch=3)  
title("Sepal.Length vs Petal.Width")
```


Sepal.Length vs Petal.Width



```
#####
```

```
# #  
# Exercise 2 #  
# #
```

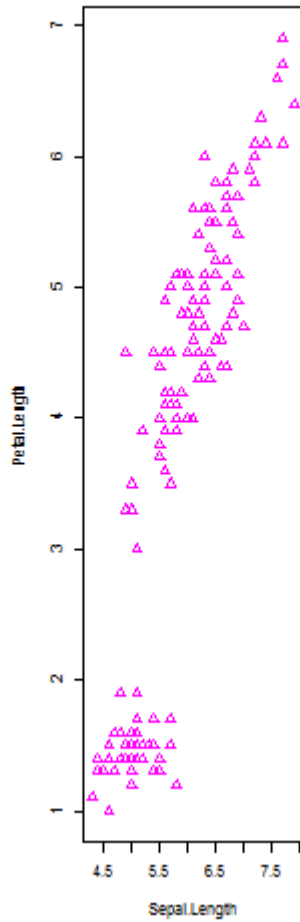
```
#####
```

```
par(mfrow=c(1,3))  
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)  
title("Sepal.Length vs Sepal.Width")  
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)  
title("Sepal.Length vs Petal.Length")  
plot(Sepal.Length, Petal.Width, col='red', pch=3)  
title("Sepal.Length vs Petal.Width")
```

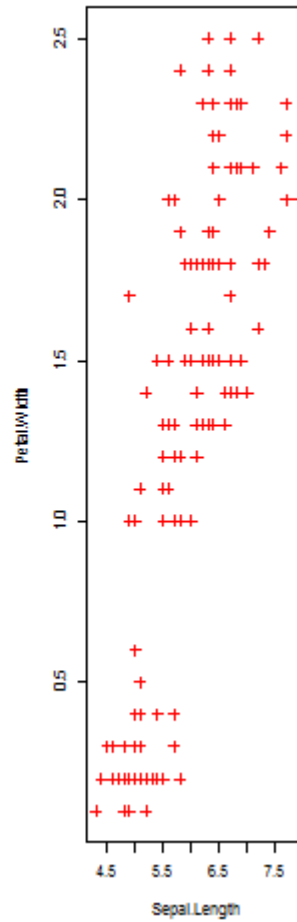
Sepal.Length vs Sepal.Width



Sepal.Length vs Petal.Length



Sepal.Length vs Petal.Width

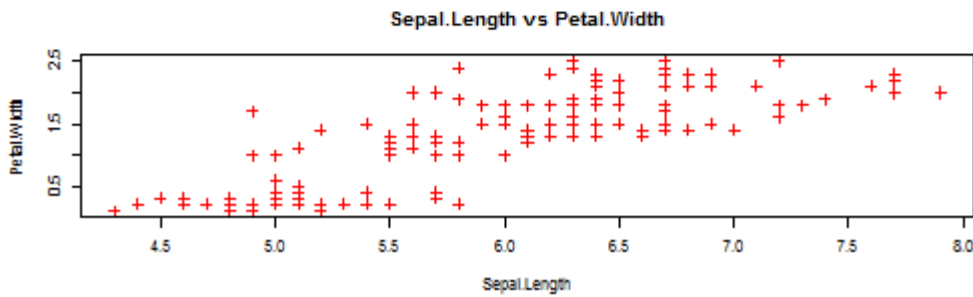
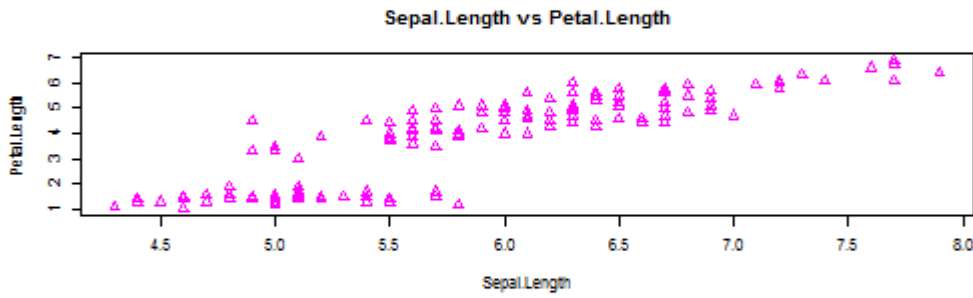
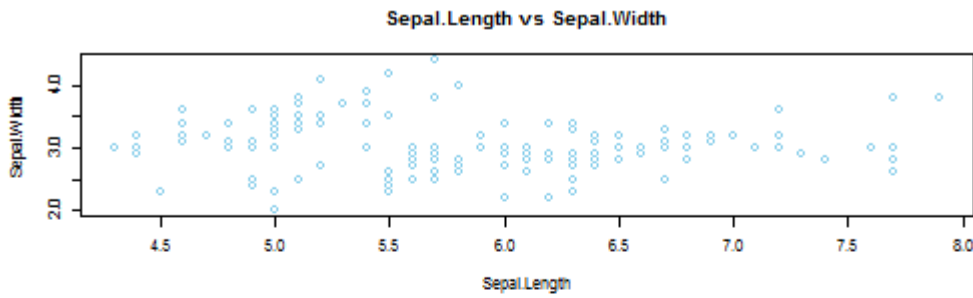


#####

```
#                               #
#   Exercise 3                   #
#                               #
```

#####

```
par(mfrow=c(3,1))
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)
title("Sepal.Length vs Sepal.Width")
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)
title("Sepal.Length vs Petal.Length")
plot(Sepal.Length, Petal.Width, col='red', pch=3)
title("Sepal.Length vs Petal.Width")
```



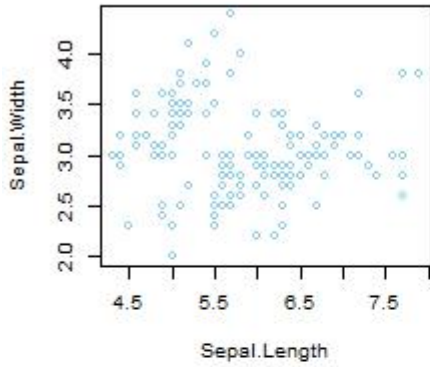
#####

```
#                               #
#   Exercise 4                   #
#                               #
```

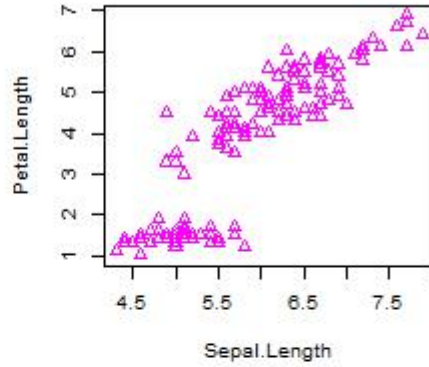
#####

```
par(mfrow=c(2,2))
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)
title("Sepal.Length vs Sepal.Width")
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)
title("Sepal.Length vs Petal.Length")
plot(Sepal.Length, Petal.Width, col='red', pch=3)
title("Sepal.Length vs Petal.Width")
```

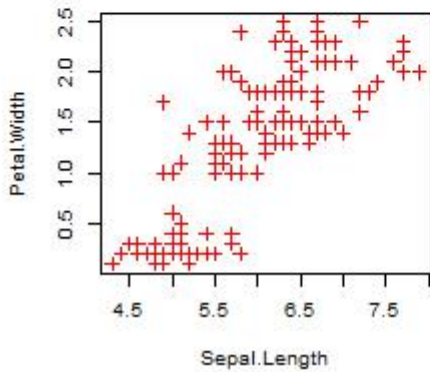
Sepal.Length vs Sepal.Width



Sepal.Length vs Petal.Length



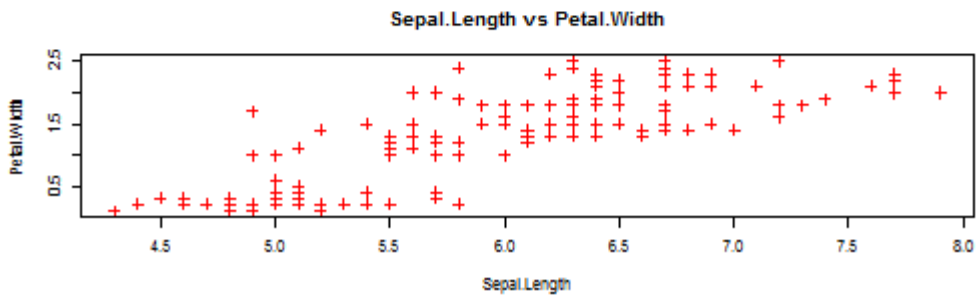
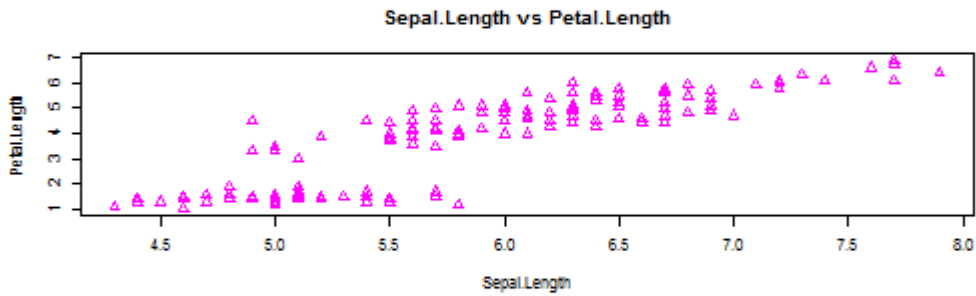
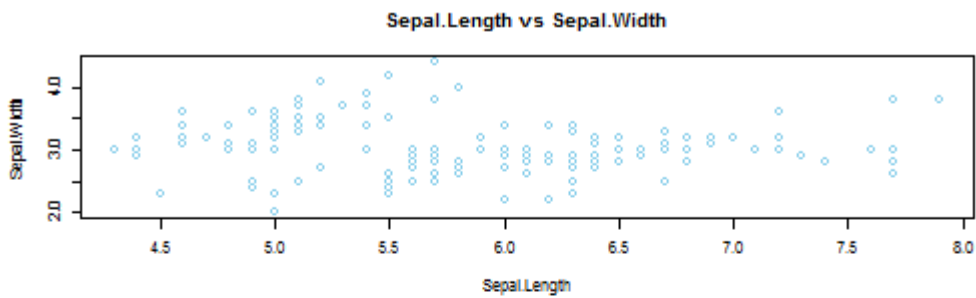
Sepal.Length vs Petal.Width



```
#####  
#                               #  
#   Exercise 5                   #  
#                               #
```

```
#####
```

```
par(mfcol=c(3,1))  
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)  
title("Sepal.Length vs Sepal.Width")  
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)  
title("Sepal.Length vs Petal.Length")  
plot(Sepal.Length, Petal.Width, col='red', pch=3)  
title("Sepal.Length vs Petal.Width")
```



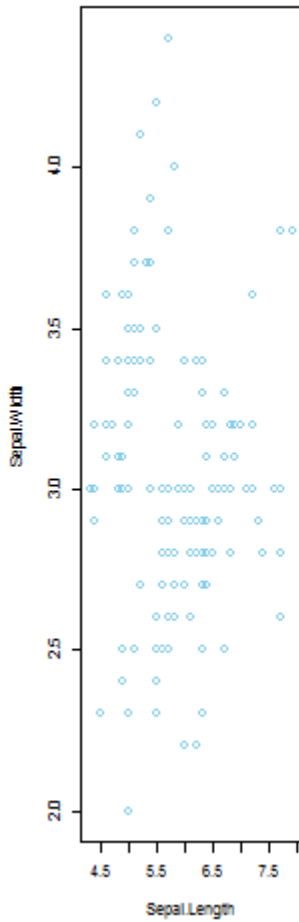
#####

```
#                               #
#   Exercise 6                   #
#                               #
```

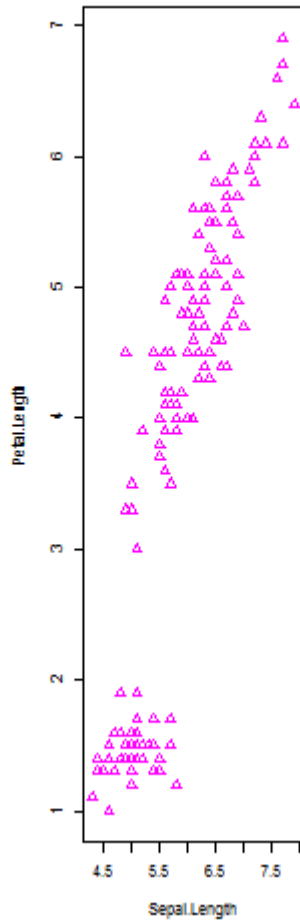
#####

```
par(mfcol=c(1,3))
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)
title("Sepal.Length vs Sepal.Width")
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)
title("Sepal.Length vs Petal.Length")
plot(Sepal.Length, Petal.Width, col='red', pch=3)
title("Sepal.Length vs Petal.Width")
```

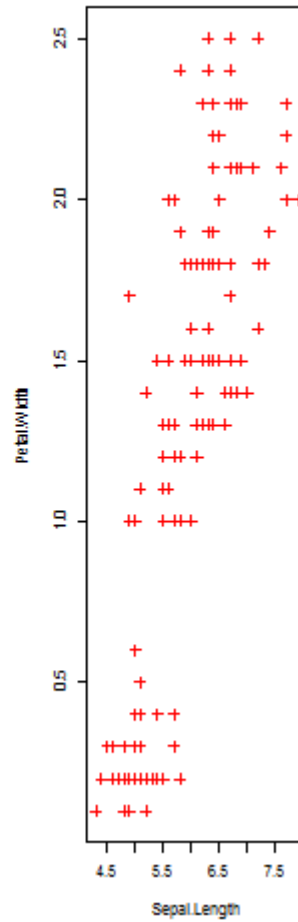
Sepal.Length vs Sepal.Width



Sepal.Length vs Petal.Length



Sepal.Length vs Petal.Width



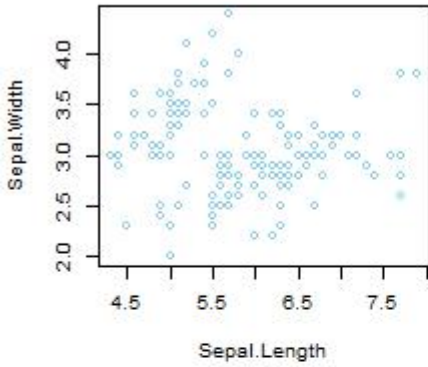
#####

```
# #
# Exercise 7 #
# #
```

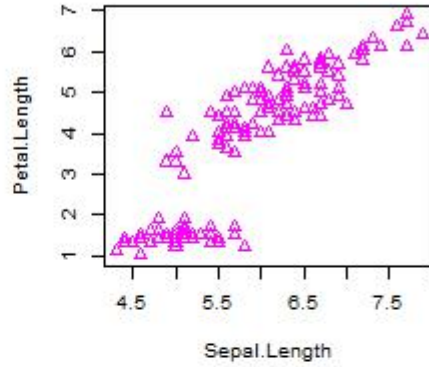
#####

```
par(mfcol=c(2,2))
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)
title("Sepal.Length vs Sepal.Width")
plot(Sepal.Length, Petal.Width, col='red', pch=3)
title("Sepal.Length vs Petal.Width")
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)
title("Sepal.Length vs Petal.Length")
```

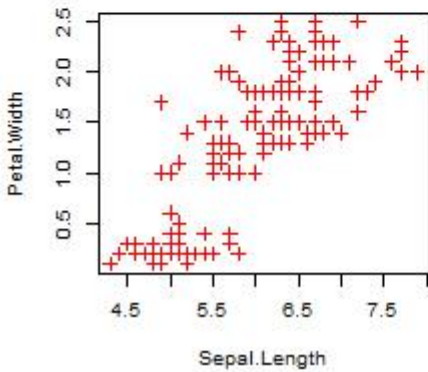

Sepal.Length vs Sepal.Width



Sepal.Length vs Petal.Length



Sepal.Length vs Petal.Width



#####

```
#                               #
#   Exercise 8                   #
#                               #
```

#####

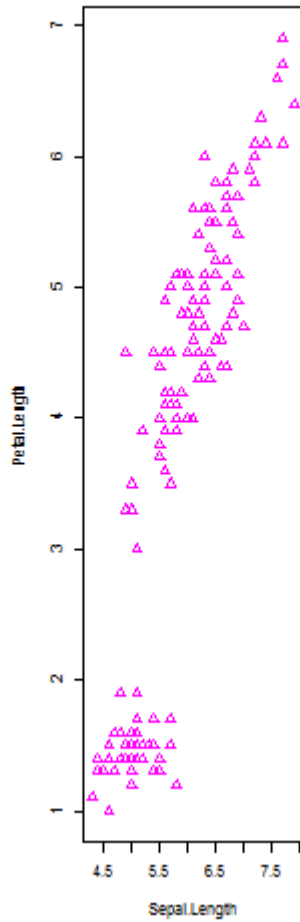
```
layout(matrix(c(1,2,3), 1, 3, byrow = TRUE))
```

```
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)
title("Sepal.Length vs Sepal.Width")
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)
title("Sepal.Length vs Petal.Length")
plot(Sepal.Length, Petal.Width, col='red', pch=3)
title("Sepal.Length vs Petal.Width")
```

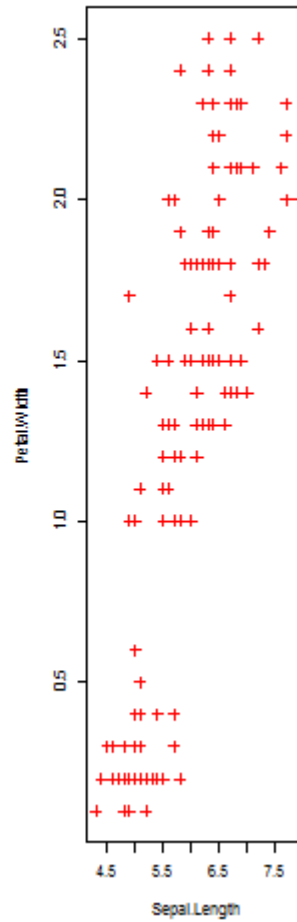
Sepal.Length vs Sepal.Width



Sepal.Length vs Petal.Length



Sepal.Length vs Petal.Width

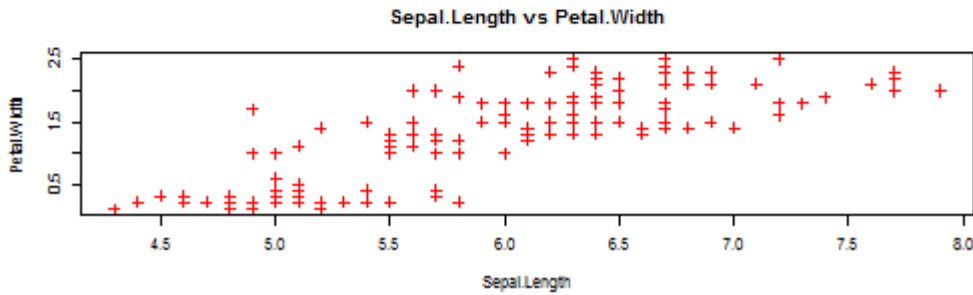
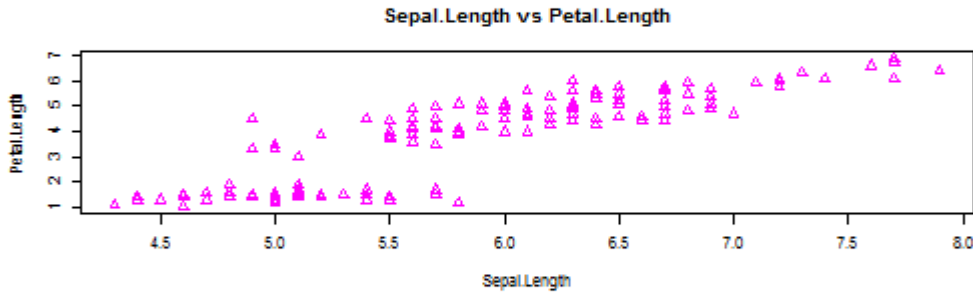
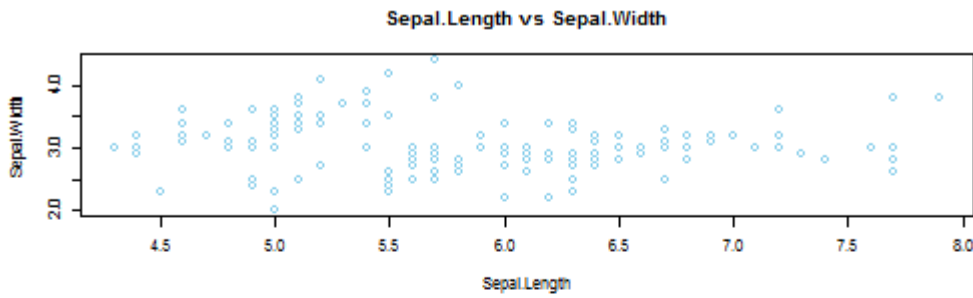


#####

```
# #
# Exercise 9 #
# #
```

#####

```
layout(matrix(c(1,2,3), 3, 1, byrow = TRUE))
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)
title("Sepal.Length vs Sepal.Width")
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)
title("Sepal.Length vs Petal.Length")
plot(Sepal.Length, Petal.Width, col='red', pch=3)
title("Sepal.Length vs Petal.Width")
```

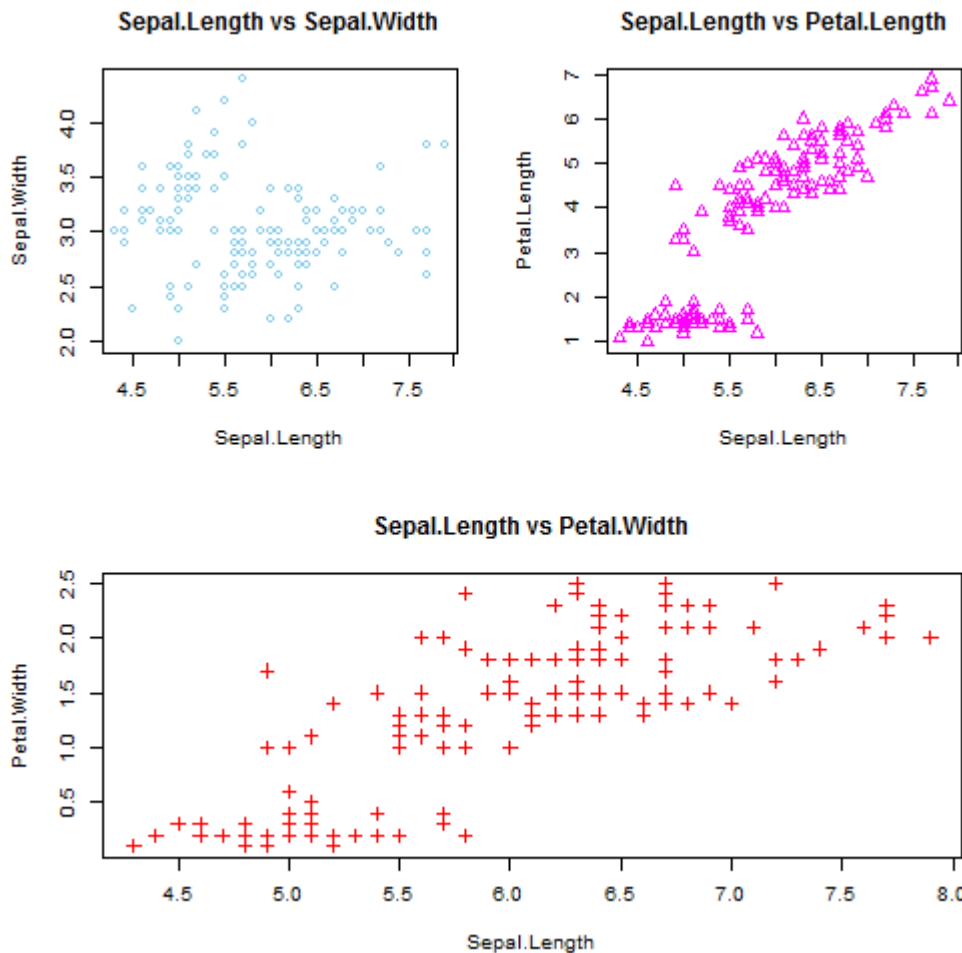


```
#####
```

```
# #
# Exercise 10 #
# #
```

```
#####
```

```
layout(matrix(c(1,2,3,3), 2, 2, byrow = TRUE))
plot(Sepal.Length, Sepal.Width, col='skyblue', pch=1)
title("Sepal.Length vs Sepal.Width")
plot(Sepal.Length, Petal.Length, col='magenta', pch=2)
title("Sepal.Length vs Petal.Length")
plot(Sepal.Length, Petal.Width, col='red', pch=3)
title("Sepal.Length vs Petal.Width")
```



```
detach(iris)
```

Working with Shapefiles in R Exercises



R has many powerful libraries to handle spatial data, and the things that R can do with maps can only grow. This exercise tries to demonstrate a few basic functionalities of R while dealing with shapefiles.

A shapefile is a simple, nontopological format for storing the

geometric location and attribute information of geographic features. Geographic features in a shapefile can be represented by points, lines, or polygons (ESRI). The geographic features are associated with an attribute table which is very similar to an R dataframe.

The `rgdal` package in R provides bindings to the popular Geospatial Data Abstraction Library (GDAL) for reading, writing and converting between spatial formats. We are using a very popular dataset of London sports participation shapefile ([download here](#)). The attributes `Pop_2001` and `Partic_Per` represents the population of London Boroughs in 2001 and the percentage of the population participating in sporting activities.

Answers to the exercises are available [here](#). If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Please install and load the package `rgdal` before starting the exercises.

Exercise 1

Read the London Sports map from the shapefile `london_sports.shp` .

Exercise 2

Change the coordinate system of the map to WGS 84.

Exercise 3

Find the names of the zones where sports participation rates is more than 25%.

Exercise 4

Plot the london map in Sky Blue, along with a title.

Exercise 5

Plot the zones in London with Sports Participation Rates less

than 15% in red. Retain the earlier blue color for other zones.

Exercise 6

Plot the zones in London with Sports Participation Rates more than 25% in green. Retain the earlier color for other zones.

Exercise 7

Place a black circle marker at the centre of each zone. Retain previous maps.

Exercise 8

Put labels for each zone. Place the labels to the right of the black marker.

Exercise 9

Add another categorical attribute `sports_part` which has values "low", "medium" and "high" for sports participation rates less than equal to 15%, between 15 to 25% and greater than 25% respectively.

Exercise 10

Save the new map object with modified attribute table as a new shapefile "london_sport2.shp".

[Working with Shapefiles in R Solutions](#)

Below are the solutions to [these](#) exercises on Working with Shapefiles in R.

```
if (!require(rgdal)){install.packages(rgdal, dep=T)}
```



```
## Warning: package 'rgdal' was built under R version 3.3.2
```

```
## Warning: package 'sp' was built under R version 3.3.2
```

```
library(rgdal)
```

```
#####
```

```
# #
```

```
# Exercise 1 #
```

```
# #
```

```
#####
```

```
london <- readOGR(dsn = ".", layer = "london_sport")
```

```
## OGR data source with driver: ESRI Shapefile
```

```
## Source: ".", layer: "london_sport"
```

```
## with 33 features
```

```
## It has 4 fields
```

```
## Integer64 fields read as strings: Pop_2001
```

```
#####
```

```
# #
```

```
# Exercise 2 #
```

```
# #
```

```
#####
```

```
london<-spTransform(london, CRS("+proj=longlat +datum=WGS84"))
```

```
#####
```

```
# #
```

```
# Exercise 3 #
```

```
# #
```

```
#####
```

```
london[which(london@data$Partic_Per>25),]@data$name
```

```
## [1] Richmond upon Thames Wandsworth
```

```
Hammersmith and Fulham
```

```
## [4] Kensington and Chelsea Camden
```

```
## 33 Levels: Barking and Dagenham Barnet Bexley Brent Bromley
```

```
... Westminster
```

```
#####
```

```
# #  
# Exercise 4 #  
# #  
#####  
plot(london, col="skyblue")  
title("Map of London")
```

Map of London



```
#####  
# #  
# Exercise 5 #  
# #  
#####  
plot(london[which(london@data$Partic_Per<15),], col="red",  
add=T)
```

Map of London



```
#####
```

```
# #
```

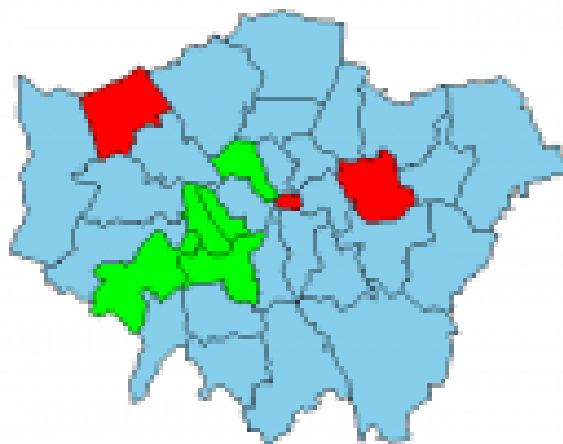
```
# Exercise 6 #
```

```
# #
```

```
#####
```

```
plot(london[which(london@data$Partic_Per>25),], col="green",  
add=T)
```

Map of London



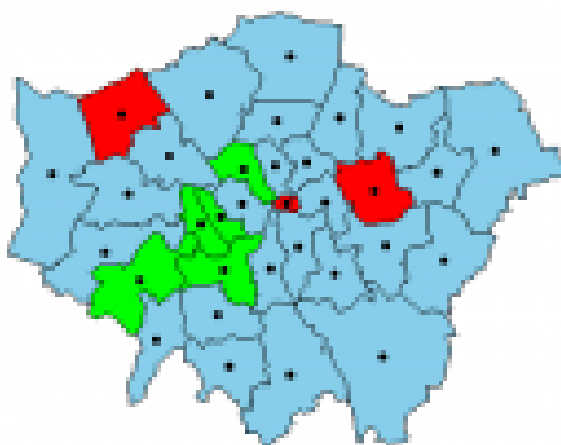
```
#####
```

```
# #  
# Exercise 7 #  
# #
```

```
#####
```

```
midpoints<-coordinates(london)  
points(midpoints, col="black", pch=20)
```

Map of London



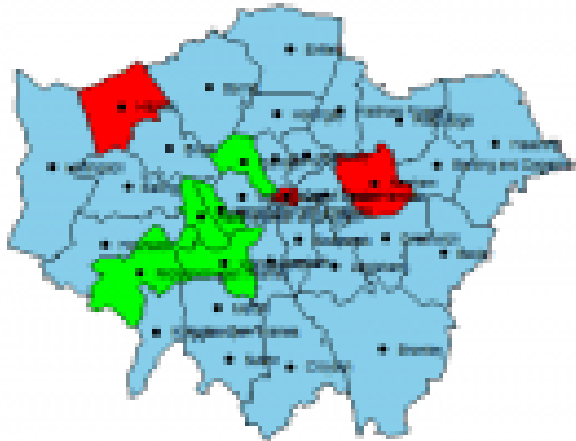
```
#####
```

```
# #  
# Exercise 8 #  
# #
```

```
#####
```

```
text(coordinates(london),as.character(london@data$name),  
pos=4, cex=0.5)
```

Map of London



```
#####
```

```
# #  
# Exercise 9 #  
# #
```

```
#####
```

```
london@data$sports_part[which(london@data$Partic_Per<=15)]<-  
"low"  
london@data$sports_part[which(london@data$Partic_Per>15 &  
london@data$Partic_Per<=25)]<- "medium"  
london@data$sports_part[which(london@data$Partic_Per>25)]<-  
"high"
```

```
#####
```

```
# #  
# Exercise 10 #  
# #
```

```
#####
```

```
writeOGR(london, ".", "london_sport2", driver="ESRI  
Shapefile")
```

```
## Warning in writeOGR(london, ".", "london_sport2", driver =  
"ESRI  
## Shapefile"): Field names abbreviated for ESRI Shapefile
```

driver