Syria Refugee Flows to Iraq

This project will map Syrian Refugee Flows to Iraq. We will use UNHCR data sources here and make a few assumptions to get this to plot. If you are able to get better data from UNHCR regarding refugee origins, you can make more accurate plots. For our case we don’t have a clear picture on place of origin for refugees coming from Syria and going to Iraq. UNHCR publishes detailed information online about these refugee flows and refugees in Iraq so part of the picture is accurate, yet we need to make a few assumptions to complete this project. These assumptions are based on UNHCR’s publications and maps. If you’d like to reference UNHCR’s maps looking at recent refugee flow data, check it out online here: http://data.unhcr.org/syrianrefugees/download.php?id=12667 (http://data.unhcr.org/syrianrefugees/download.php?id=12667) and http://data.unhcr.org/syrianrefugees/regional.php (http://data.unhcr.org/syrianrefugees/regional.php)

We are going to need some shapefiles for this project.

Download Syria (or any other country) shapefile data from http://www.gadm.org/country (http://www.gadm.org/country) you can either download it manually from the website, or use the getData function below to download it with R.

The level refers to the administrative level, so level 0 will give you the country outline and level 1 will give you the regional boundaries.

The countries are identified by their ISO Country Code, a list of these can be found here: http://www.nationsonline.org/oneworld/country_code_list.htm (http://www.nationsonline.org/oneworld/country_code_list.htm)

```
# if you chose to pull directly from the site, use the below syntax for your specific countr
y.
library(maptools)
```

```
## Warning: package 'maptools' was built under R version 3.3.3
```

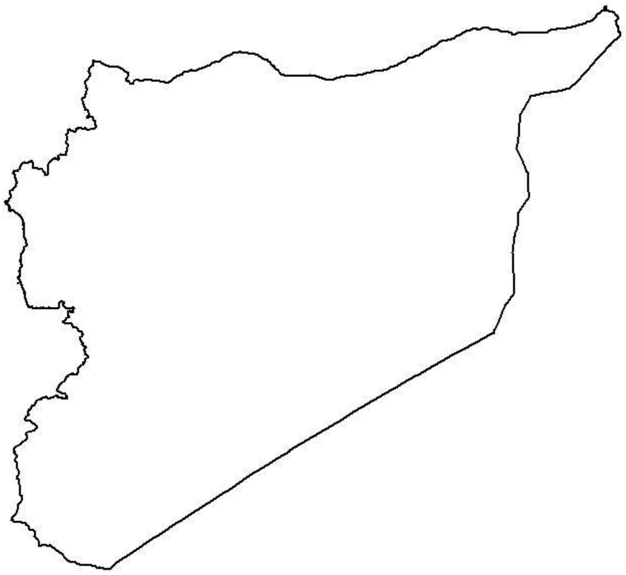
```
## Loading required package: sp
```

```
## Checking rgeos availability: FALSE
##      Note: when rgeos is not available, polygon geometry     computations in maptools depe
nd on gpclib,
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()
```

```
library(raster)
```

```
## Warning: package 'raster' was built under R version 3.3.3
```

```
library(ggplot2)

#let's pull the country outlines here.
sy_outline <- getData("GADM", country="SY", level=0)
iq_outline <- getData("GADM", country="IQ", level=0)

plot(sy_outline)
```



In this case, I want to read in my shapefile that I already downloaded. We will use the readRDS fuction as shown below. ```

```
library(sp)
library("maptools", lib.loc="~/R/win-library/3.3")
library("maps", lib.loc="~/R/win-library/3.3")
library("mapproj", lib.loc="~/R/win-library/3.3")
library("mapdata", lib.loc="~/R/win-library/3.3")
library("ggplot2", lib.loc="~/R/win-library/3.3")
library("RgoogleMaps", lib.loc="~/R/win-library/3.3")
```

```
## Warning: package 'RgoogleMaps' was built under R version 3.3.3
```

```
library("ggmap", lib.loc="~/R/win-library/3.3")
```

```
## Warning: package 'ggmap' was built under R version 3.3.3
```

```
library("ggthemes", lib.loc="~/R/win-library/3.3")
library(RgoogleMaps)
library(rworldmap)
```

```
## Warning: package 'rworldmap' was built under R version 3.3.3
```

```
## ### Welcome to rworldmap ###
```

```
## For a short introduction type :   vignette('rworldmap')
```

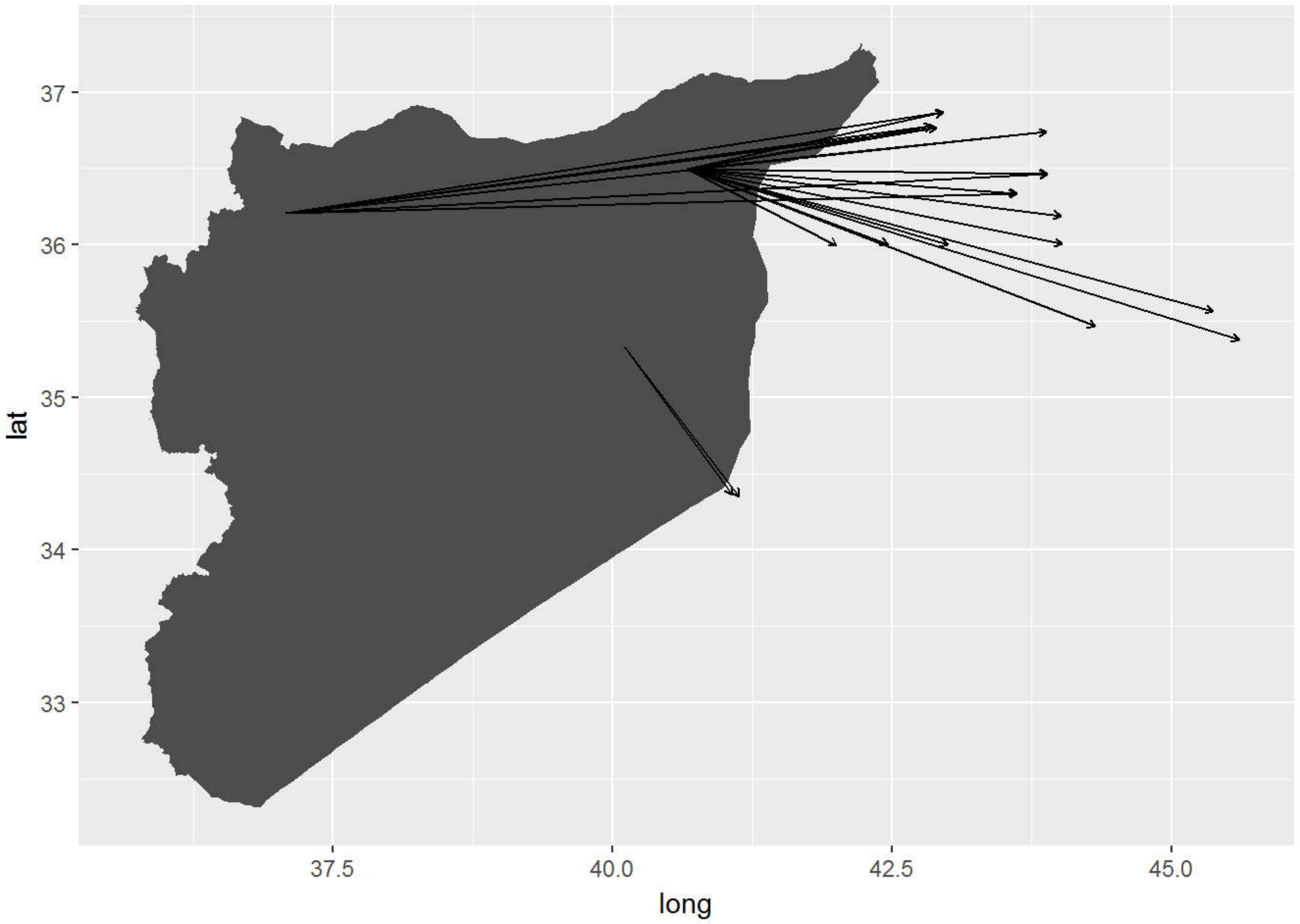
```
library(readr)


# load in the shapefiles
syria_border <- readRDS("C:/Users/john.mataya/Desktop/Datasets/Syria Data/spatial_data/SYR_ad
m2.rds")

# read in your datafile
iraq <- read_csv("C:/Users/john.mataya/Desktop/Datasets/Syria Data/UNHRC/Custom_Camp/Iraq_tra
vel_assumptions.csv",)
```

```
## Warning: Duplicated column names deduplicated: 'Lat.o' => 'Lat.o_1' [7]
```

```
## Parsed with column specification:
## cols(
##   Country = col_character(),
##   Camp = col_integer(),
##   Syria_Origin = col_character(),
##   Destination = col_character(),
##   Refugees = col_integer(),
##   Lat.o = col_double(),
##   Lat.o_1 = col_double(),
##   Lat.d = col_double(),
##   Lon.d = col_double()
## )
```

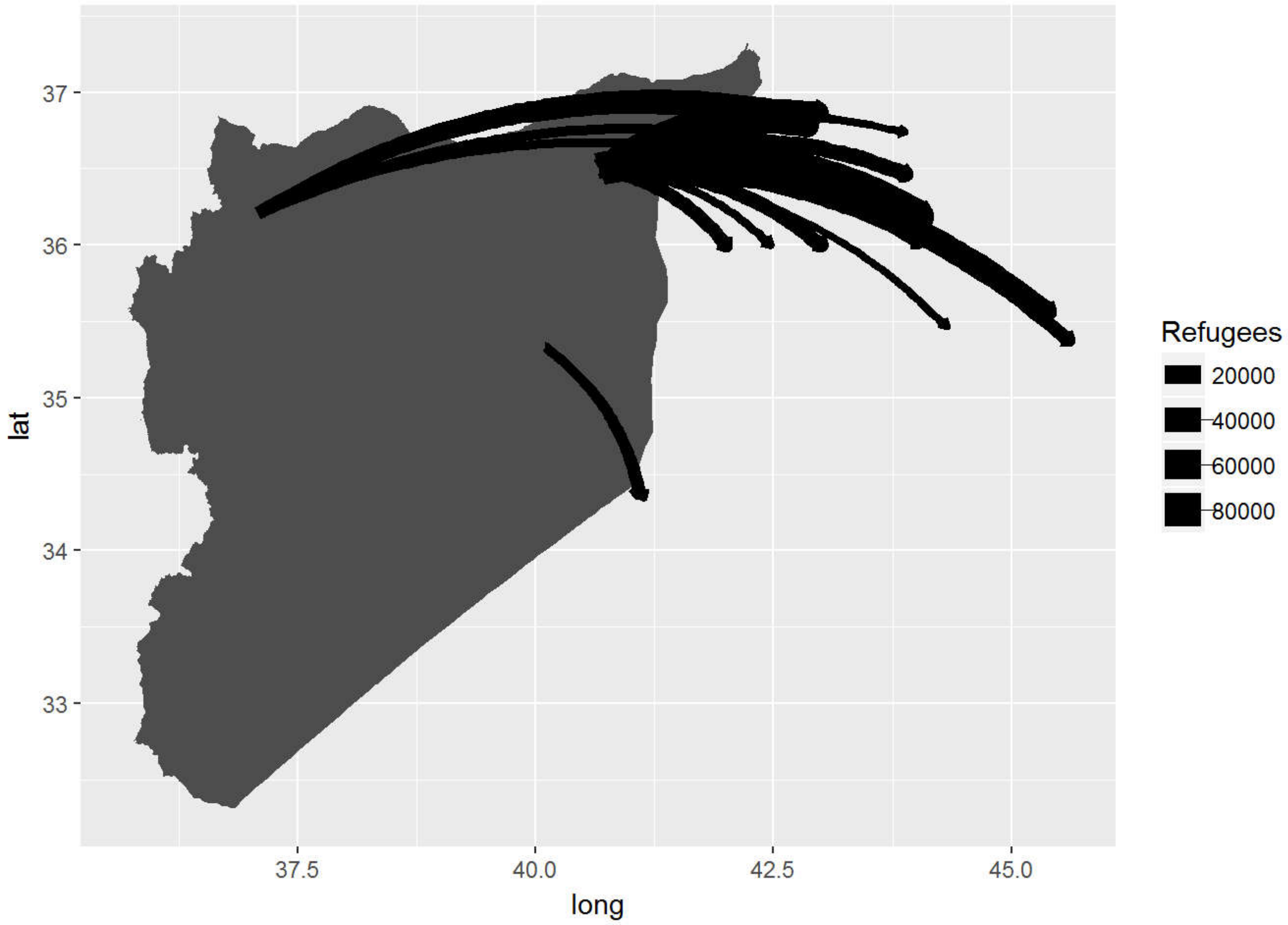
```
# let's plot the shapefile
# this plots straight arrows
ggplot() + geom_polygon(data = syria_border, aes(long, lat, group=group), fill = 'gray30') +
geom_segment(data = iraq, aes(x = Lat.o_1, y = Lat.o, xend = Lon.d, yend = Lat.d), arrow = ar
row(length = unit(0.01, 'npc')))
```

```
## Regions defined for each Polygons
```



```
# use the curvature function to plot curved arrows
ggplot() + geom_polygon(data = syria_border, aes(long, lat, group=group), fill = 'gray30') +
geom_curve(data = iraq, aes(x = Lat.o_1, y = Lat.o, xend = Lon.d, yend = Lat.d, size = Refuge
es), curvature = -0.2, arrow = arrow(length = unit(0.01, 'npc')))
```

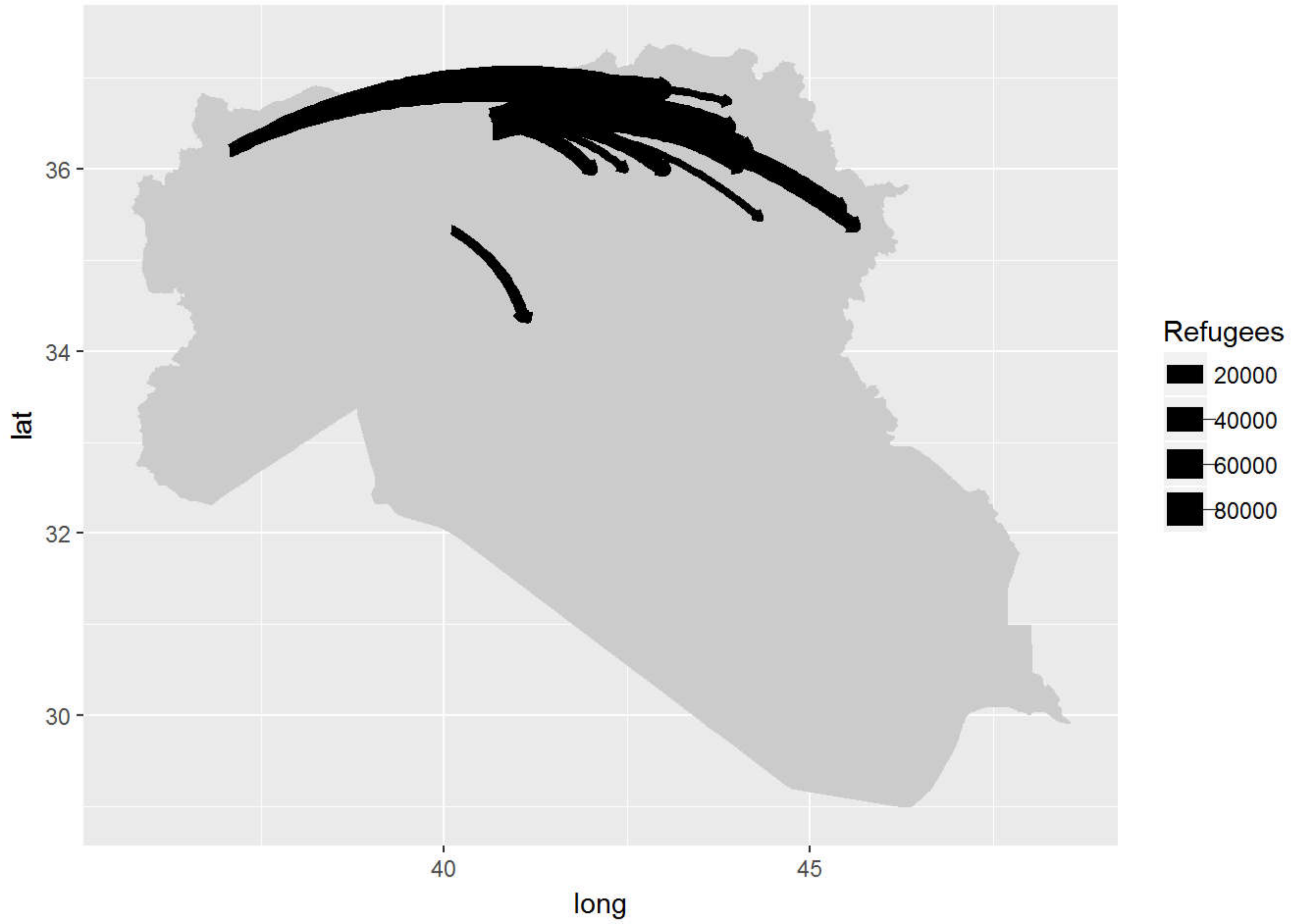
```
## Regions defined for each Polygons
```



```
# let's read in the second shapefile of iraq
iraq_border <- readRDS("C:/Users/john.mataya/Desktop/Datasets/Syria Data/spatial_data/IRQ_adm
2.rds")


# plot the two countries together and use the curved arrows.
ggplot() + geom_polygon(data = syria_border, aes(long, lat, group=group), fill = '#CCCCCC') +

  geom_polygon(data = iraq_border, aes(long, lat, group=group), fill = '#CCCCCC') +
  geom_curve(data = iraq, aes(x = Lat.o_1, y = Lat.o, xend = Lon.d, yend = Lat.d, size = Refu
gees), curvature = -0.2, arrow = arrow(length = unit(0.01, 'npc')))
```

```
## Regions defined for each Polygons
```

```
## Regions defined for each Polygons
```



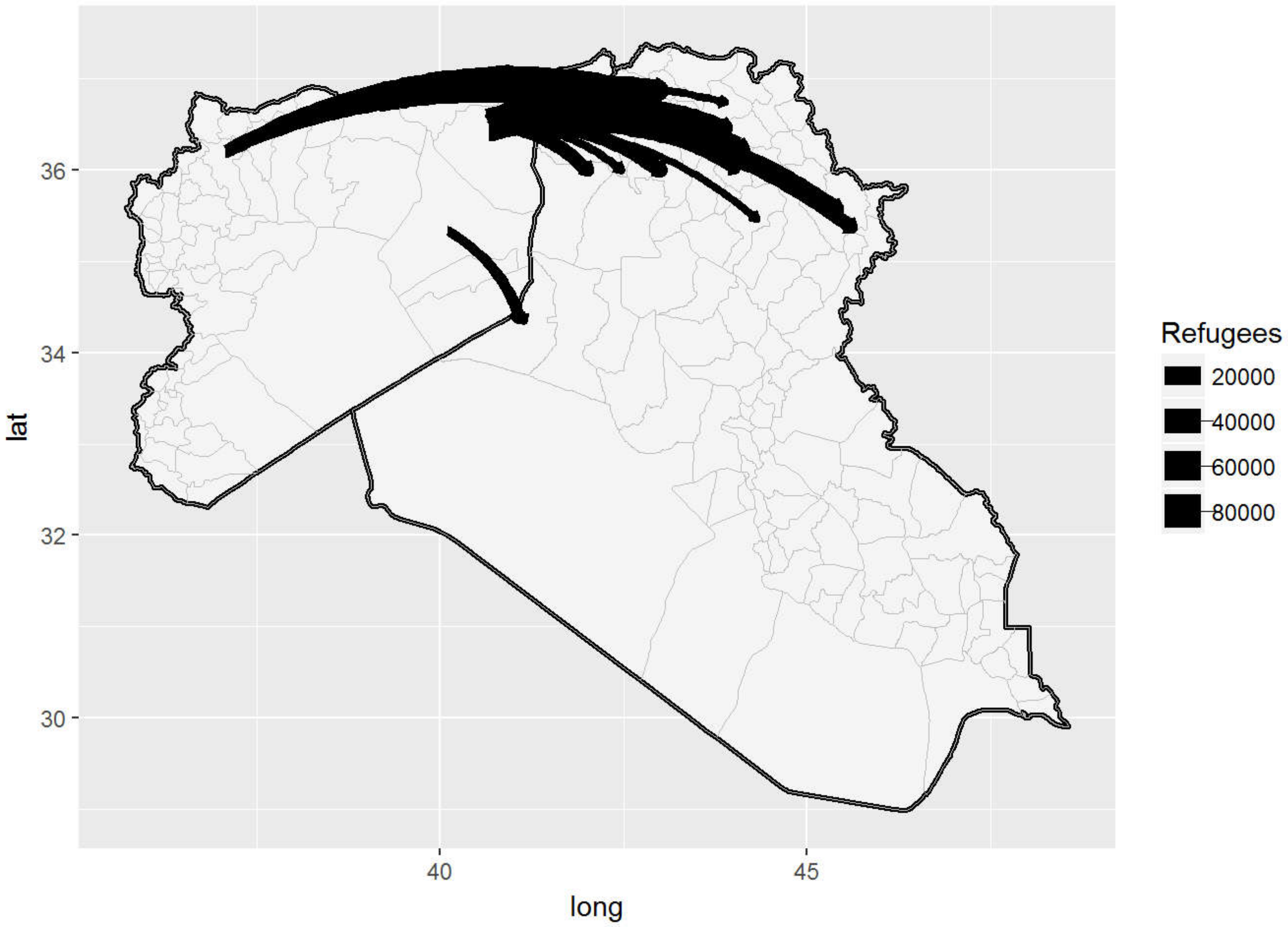
```
# we need to add the administrative borders.  use geom_path function for this.
p <- ggplot() + geom_polygon(data = syria_border, aes(long, lat, group=group), fill = 'whitesmoke') +
     geom_path(data=syria_border, aes(long,lat, group=group), color="grey", size=0.1) +
     geom_polygon(data = iraq_border, aes(long, lat, group=group), fill = 'whitesmoke') +
     geom_path(data=iraq_border, aes(long,lat, group=group), color="grey", size=0.1)
```

```
## Regions defined for each Polygons
## Regions defined for each Polygons
## Regions defined for each Polygons
## Regions defined for each Polygons
```

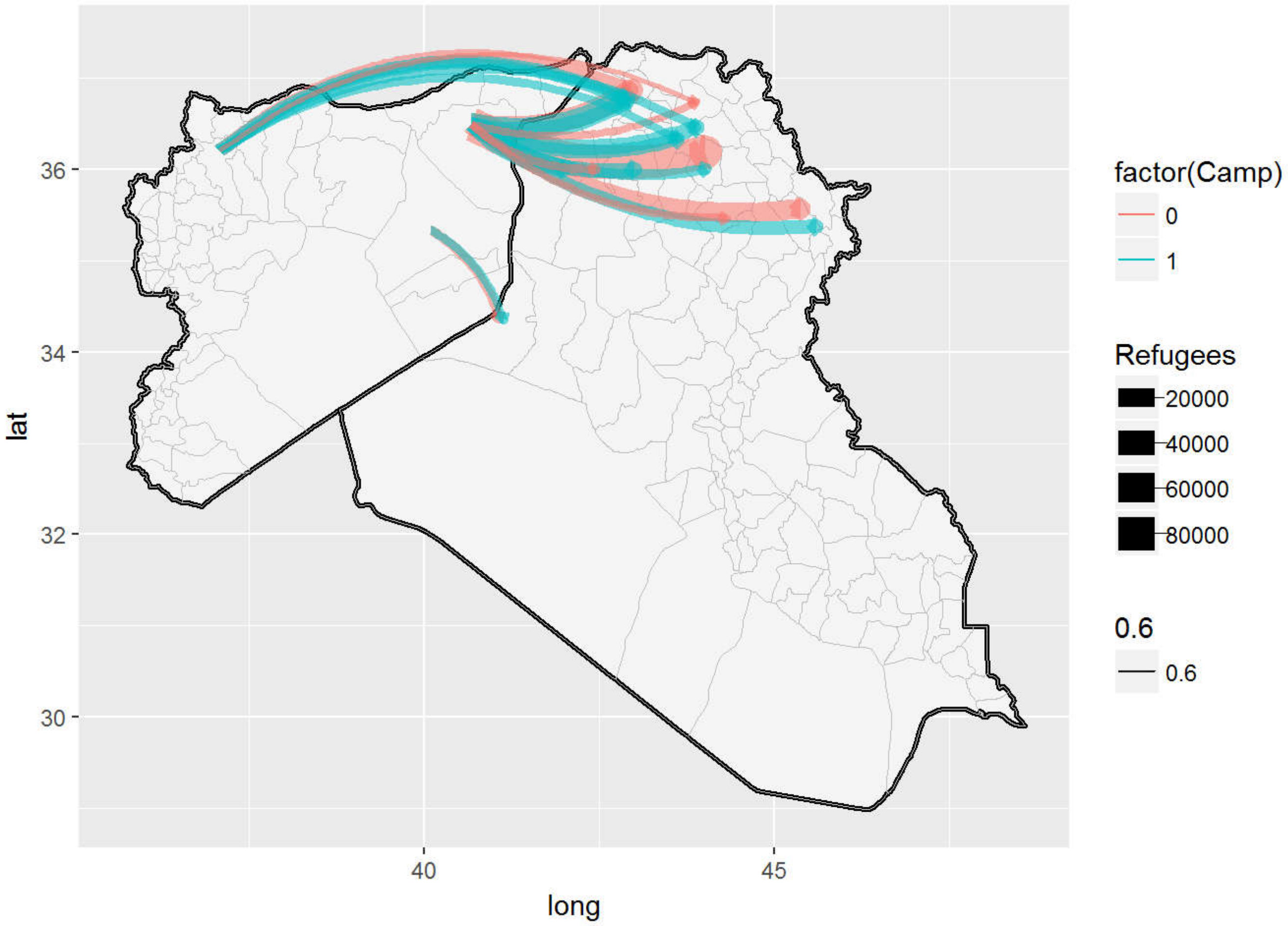
```
# Now add country borders to administrative borders.
border <- ggplot() + geom_polygon(data = syria_border, aes(long, lat, group=group), fill = 'whitesmoke') +
        geom_path(data=sy_outline, aes(long,lat, group=group), size=1) +
        geom_path(data=syria_border, aes(long,lat, group=group), color="grey", size=0.1) +
        geom_polygon(data = iraq_border, aes(long, lat, group=group), fill = 'whitesmoke') +
        geom_path(data=iq_outline, aes(long,lat, group=group), size=1) +
        geom_path(data=iraq_border, aes(long,lat, group=group), color="grey", size=0.1)
```

```
## Regions defined for each Polygons
## Regions defined for each Polygons
## Regions defined for each Polygons
## Regions defined for each Polygons
## Regions defined for each Polygons
## Regions defined for each Polygons
```

```
# now add the curved arrows
border + geom_curve(data = iraq, aes(x = Lat.o_1, y = Lat.o, xend = Lon.d, yend = Lat.d, size = Refugees), curvature = -0.2, arrow = arrow(length = unit(0.01, 'npc')))
```



```
# subset your data by origin
iraq_hasakah <- iraq[iraq$Syria_Origin == 'Al-Hasakah',]
iraq_allepo <- iraq[iraq$Syria_Origin == 'Aleppo',]
iraq_other <- iraq[iraq$Syria_Origin == 'Dayr Az Zawr',]

# change the direction of your arrows and add some color showing refugees going to camps and non-camps
pp <- border + geom_curve(data = iraq_hasakah, aes(x = Lat.o_1, y = Lat.o, xend = Lon.d, yend = Lat.d, size = Refugees, color = factor(Camp), alpha=0.6), curvature = 0.2, arrow = arrow(length = unit(0.01, 'npc'))) +
   geom_curve(data = iraq_allepo, aes(x = Lat.o_1, y = Lat.o, xend = Lon.d, yend = Lat.d, size = Refugees, color = factor(Camp), alpha=0.6), curvature = -0.3, arrow = arrow(length = unit(0.01, 'npc'))) +
   geom_curve(data = iraq_other, aes(x = Lat.o_1, y = Lat.o, xend = Lon.d, yend = Lat.d, size = Refugees, color = factor(Camp), alpha=0.6), curvature = -0.2, arrow = arrow(length = unit(0.01, 'npc')))

# look at the map
pp
```



So we've got a decent looking map at this point, but it is not ready for prime time. We need to add cities and clean this up a little bit.

```
# add cities
# I just created a csv file pulling the Lat and Lon from google maps.  You could also do this
 right in the R environment.

cities <- read_csv("C:/Users/john.mataya/Desktop/Datasets/Syria Data/UNHRC/Custom_Camp/citie
s.csv",
     col_types = cols(Lat = col_number(),
         Lon = col_number()))

# run cities to take a look at the data
cities
```

```
## # A tibble: 9 × 4
##   Country          City       Lat       Lon
##     <chr>         <chr>     <dbl>     <dbl>
## 1   Syria        Aleppo 36.19561 37.05520
## 2   Syria   Al-Hasakah 36.50447 40.70853
## 3   Syria Dayr Az Zawr 35.33512 40.10296
## 4   Syria          Homs 34.73029 36.64206
## 5   Syria      Damascus 33.50747 36.24788
## 6    Iraq         Mosul 36.35288 43.07061
## 7    Iraq         Erbil 36.19728 43.93862
## 8    Iraq        Kirkuk 35.46042 44.29670
## 9    Iraq       Baghdad 33.31167 44.28587
```

```
# now add cities to our maps and label the cities.
pp <- pp + geom_point(data = cities, aes(x = Lon, y = Lat), size = 2, alpha = 0.5) + geom_tex
t(data=cities, hjust=0.5, vjust=-0.5, aes(x=Lon, y=Lat, label=City), colour="black", size=3,
alpha = 0.6)
```

Let’s continue to clean this map up a bit.

Drop our legend - it’s hard to interpret

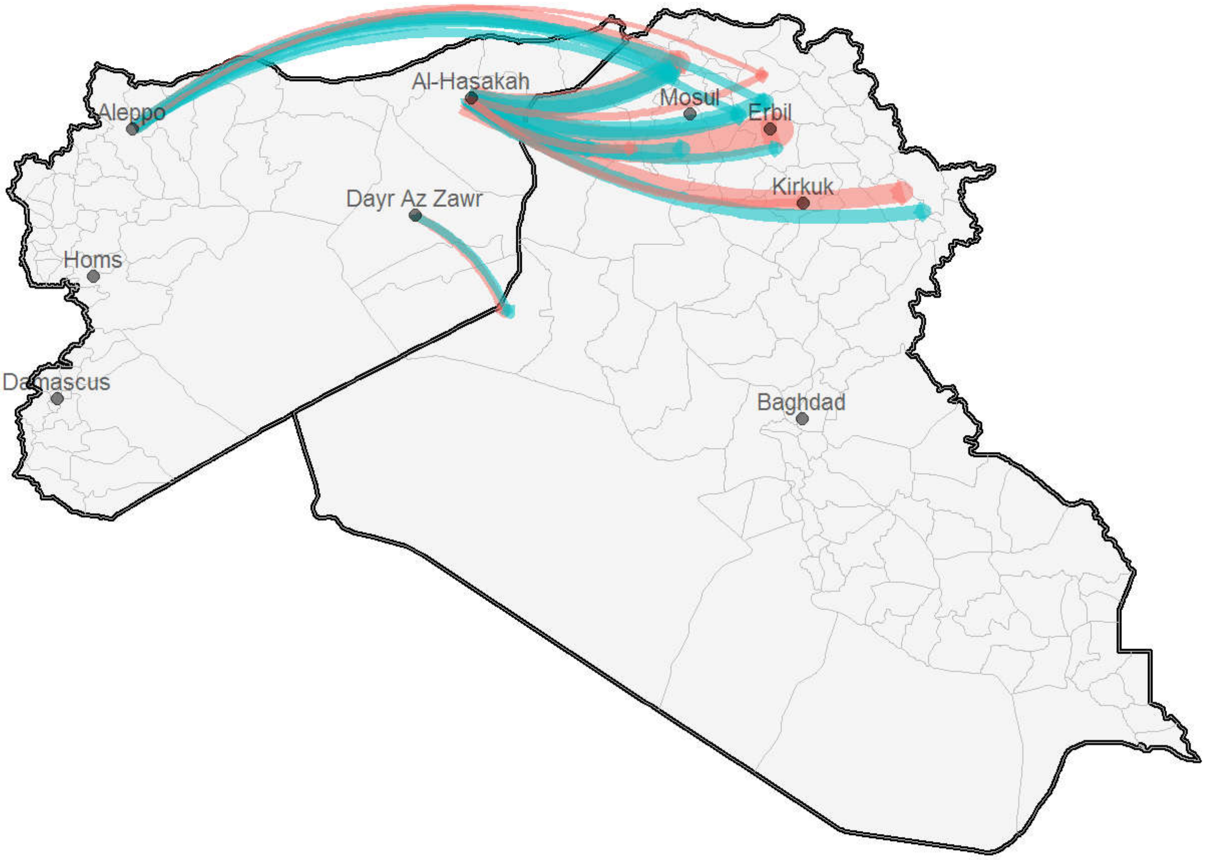
```
pp <- pp + theme(legend.position="none")
```

this serries of code will clean up the map

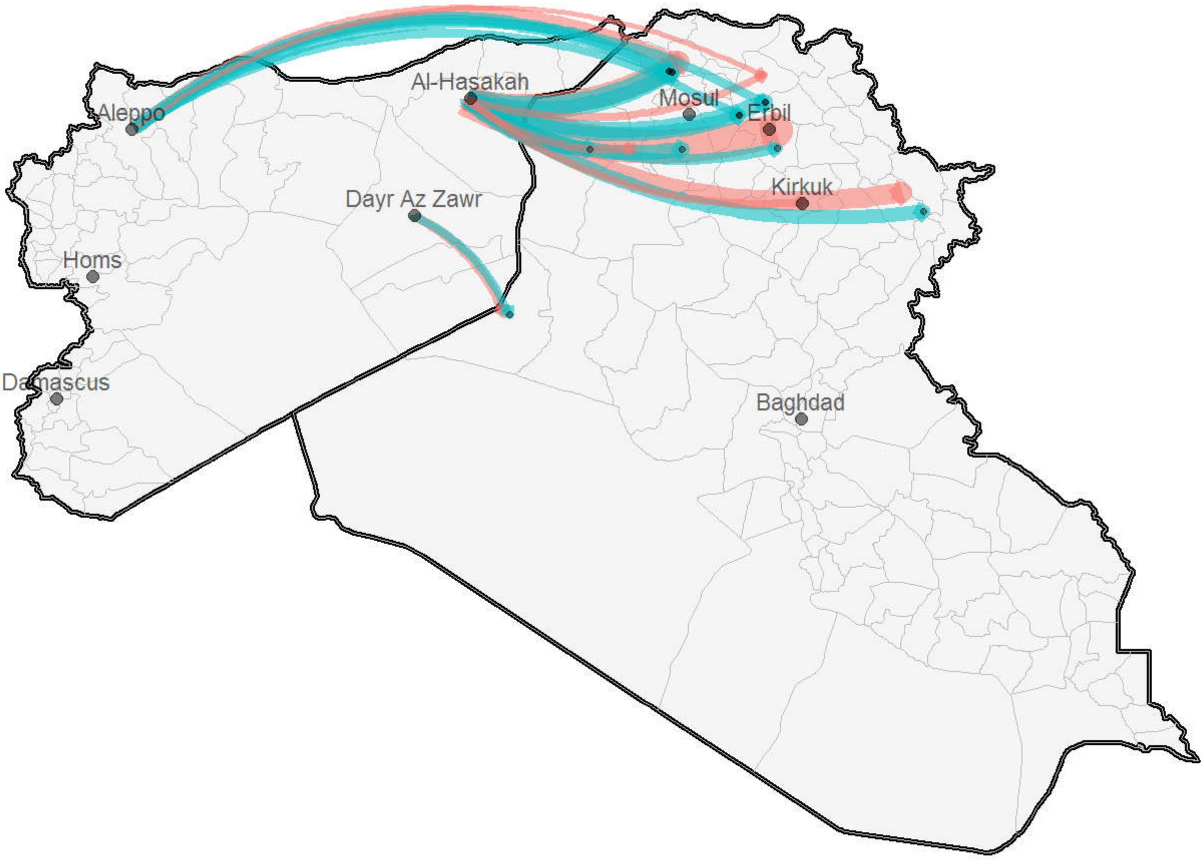
```
theme_opts<-list(theme(panel.grid.minor = element_blank(),
                       panel.grid.major = element_blank(),
                       panel.background = element_blank(),
                       plot.background = element_blank(),
                       axis.line = element_blank(),
                       axis.text.x = element_blank(),
                       axis.text.y = element_blank(),
                       axis.ticks = element_blank(),
                       axis.title.x = element_blank(),
                       axis.title.y = element_blank(),
                       plot.title = element_blank()))
```

Put things together

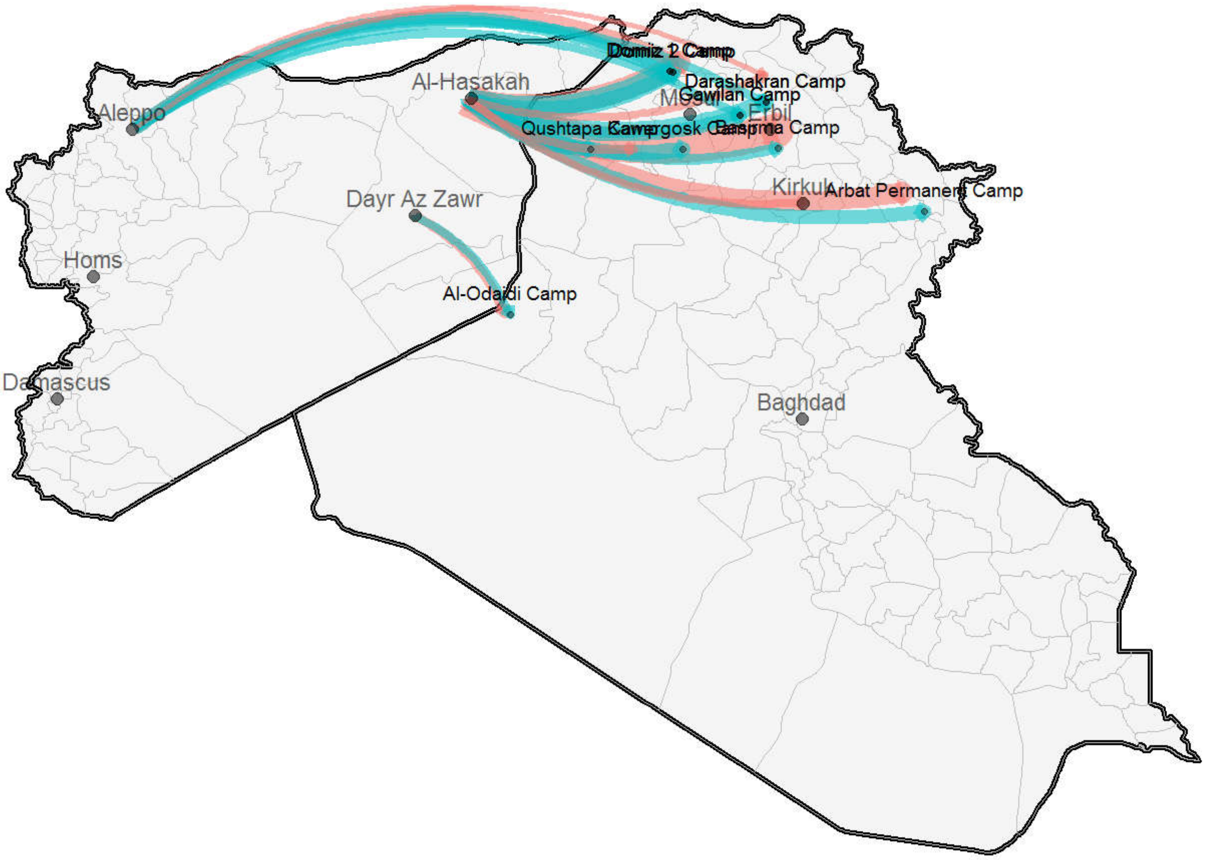
```
pp + theme_opts
```



```
# If you want to add the refugee camps use this bit of code.
iraq_camp <- iraq[iraq$Camp == 1,]

#plot just the camp points for a cleaner map.
pp + theme_opts + geom_point(data = iraq_camp, aes(x = Lon.d, y = Lat.d), size = 0.95, alpha = 0.4)
```



```
# if you need the camp labels and points use this piece of code - note this makes for a busy map.
# i recommend manually putting the labels on in a seperate software platform like Adobe InDesign or even powerpoint
pp + theme_opts + geom_point(data = iraq_camp, aes(x = Lon.d, y = Lat.d), size = 0.95, alpha = 0.4) + geom_text(data=iraq_camp, hjust=0.5, vjust=-1, aes(x=Lon.d, y=Lat.d, label=Destination), colour="black", size=2.5)
```


Domiz 1 Camp
Domiz 2 Camp
Al-Hasakah
Darashakran Camp
Mosul
Gawilan Camp
Erbil
Aleppo
Qushtapa Camp
Kawergosk Camp
Basirma Camp
Dayr Az Zawr
Kirkuk
Arbat Permanent Camp
Homs
Al-Odaidi Camp
Damascus
Baghdad