Analysing Helipad Data

Danny Wong

Sunday, April 05, 2015

We have some data from the KSS HEMS patients conveyed to KCH.

#Read the data into a dataframe, specify the NA strings to incorporate the missing values  
data <- read.csv("data/HEMSdata.csv", na.strings=c("NA","n/a", ""))  
  
#Let's look at the table  
head(data)

## Job.ID Date Day Time Job.Type Dispatch  
## 1 176 01/07/2013 Monday 05:28 RTC Immediate  
## 2 217 06/07/2013 Saturday 19:06 Accidental injury Crew Request  
## 3 224 08/07/2013 Monday 10:17 Accidental injury Interrogate  
## 4 225 08/07/2013 Monday 15:27 RTC Interrogate  
## 5 229 09/07/2013 Tuesday 11:57 RTC Interrogate  
## 6 234 10/07/2013 Wednesday 07:25 RTC Interrogate  
## Location County Grid Callsign  
## 1 tonbridge Kent TQ 612 435 DA33  
## 2 Royal Pier Road, Gravesend Kent TQ 649 743 DA33  
## 3 Wittersham Nr Tenterden, Kent Kent TQ 903 285 DA55  
## 4 Farnborough Surrey SU 881 541 H60  
## 5 M26 Sevenoaks Kent TQ 497 570 H60  
## 6 M20 Junc 9-8, London-bound carriageway Kent TQ 801 566 DA55  
## Vehicle Patients Injuries  
## 1 Volvo 1 Head, Thorax, Upper arm, Lower arm, Abdomen, Upper leg  
## 2 Volvo 1 Head, Thorax, Abdomen  
## 3 Volvo 1 Head, Thorax, Abdomen  
## 4 G-KSSA 1 Head  
## 5 G-KSSA 1 Head, Upper arm, Thorax  
## 6 Volvo 1 Lower arm, Lower leg  
## RSI Interventions  
## 1 No Splintage, Packaging, Access  
## 2 No Access, Packaging, Splintage  
## 3 No Packaging, Splintage, Ultrasound  
## 4 Yes Access, Airway, Packaging  
## 5 Yes Access, Packaging, Splintage, Thoracostomy  
## 6 No Access, Packaging, Ultrasound  
## Drugs  
## 1 Fentanyl, Ketamine, Paracetamol, Tranexamic Acid, Ondansetron, Midazolam  
## 2 Ondansetron, Ketamine, Midazolam  
## 3 Morphine, Paracetamol  
## 4 Paracetamol, Fentanyl, Ketamine, Midazolam, Tranexamic Acid, Rocuronium  
## 5 Fentanyl, Ketamine, Midazolam, Morphine, Rocuronium, Tranexamic Acid  
## 6 Ketamine, Midazolam, Ondansetron  
## Blood Code.Red Direct.to.CT Hospital Result  
## 1 No No No King's College Hospital Patient Conveyed  
## 2 No No No King's College Hospital Patient Conveyed  
## 3 No No No King's College Hospital Patient Treated  
## 4 No No No King's College Hospital Patient Conveyed  
## 5 No No No King's College Hospital Patient Conveyed  
## 6 No No No King's College Hospital Patient Conveyed  
## Job.Done D.D.flag MR.flag Postcode D.D.outcome Team Safeguarding  
## 1 Yes NA NA tn110qf NA <NA> 0  
## 2 Yes NA NA DA12 2AZ NA <NA> 0  
## 3 Yes NA NA TN30 7HL NA <NA> 0  
## 4 Yes NA NA gu1 6lj NA <NA> 0  
## 5 Yes NA NA TN13 2SA NA <NA> 0  
## 6 Yes NA NA ME144PW NA <NA> 0

#Let's see how many cases there are  
print(nrow(data))

## [1] 569

Notice that the coordinates are in the Ordnance Survey Grid format under the column "Grid". For us to work with this meaningfully, we need to convert this to latitude and longitude (WGS84) coordinates.

Fortunately there are others who have encountered the same problem before:

Sources:

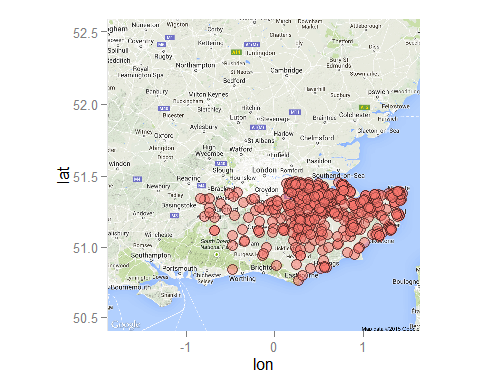
\* http://stackoverflow.com/questions/23017053/how-to-convert-uk-grid-reference-to-latitude-and-longitude-in-r  
\* https://stat.ethz.ch/pipermail/r-sig-geo/2010-November/010141.html  
\* http://www.hannahfry.co.uk/blog/2012/02/01/converting-british-national-grid-to-latitude-and-longitude-ii  
\* http://cran.r-project.org/web/packages/rnrfa/rnrfa.pdf

#Load the required packages  
require(rnrfa)  
require(dplyr)  
  
#Remove the rows with missing Grid references  
data <- filter(data, !is.na(Grid))  
   
#Parse the OS Grid References into Eastings and Northings, then pipe it into WSG84 Coordinates  
coordinates <- OSGParse(data$Grid) %>% OSG2LatLon()  
  
#Add these onto the data  
data <- mutate(data, lat = coordinates$Latitude, lon = coordinates$Longitude)  
  
#Write to .csv  
write.csv(data, file="data/cleaned.csv")

We can now start making some maps!

Let's first visualise the locations of each HEMS pickup as a quick visualisation

#Load the required libraries  
require(ggplot2)  
require(ggmap)  
require(dplyr)  
  
#Get map from Google  
ukmap <- get\_map(location = "London, UK", zoom = 8, scale = 4)  
ggmap(ukmap) + geom\_point(data = data, aes(x = data$lon, y = data$lat, fill = "red", alpha = 0.8), size = 4, shape = 21) + guides(fill=FALSE, alpha=FALSE, size=FALSE)



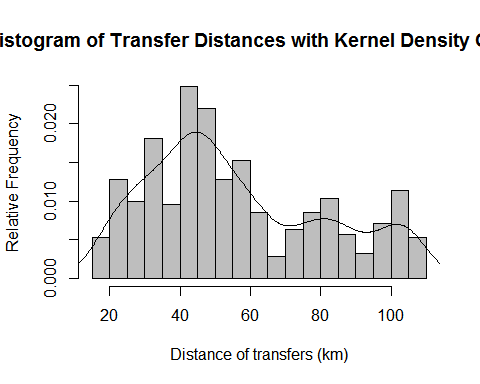
#We can separate the cases by the county where they come from, let's see the counties where they come from  
county <- count(data, County)  
county

## Source: local data frame [7 x 2]  
##   
## County n  
## 1 East Sussex 21  
## 2 Greater London 1  
## 3 Hampshire 1  
## 4 Kent 490  
## 5 Surrey 33  
## 6 West Sussex 12  
## 7 NA 5

#Let's write this into a .csv file for QGIS to use  
write.csv(county, "data/county.csv")  
  
#Let's measure the distances between all the HEMS pickup sites and KCH  
#We need the coordinates for KCH  
KCH <- geocode("King's College Hospital, Denmark Hill, London")  
  
#Let's load the required library to calculate distances  
require(geosphere)  
  
#We apply the VincentyEllipsoid method of calculating straight-line distance, and   
coordinates <- data.frame("lon" = coordinates$Longitude, "lat" = coordinates$Latitude)  
KCH <- data.frame("lon" = KCH$lon, "lat" = KCH$lat)   
coordinates$distance <- distVincentyEllipsoid(p1 = coordinates, p2 = KCH) #This calculates distances in meters  
  
#Lets add this distance to the master data frame, but convert it to kilometers first  
data <- mutate(data, distance = coordinates$distance/1000)  
  
summary(data$distance)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 16.02 37.90 49.11 56.03 75.51 107.80

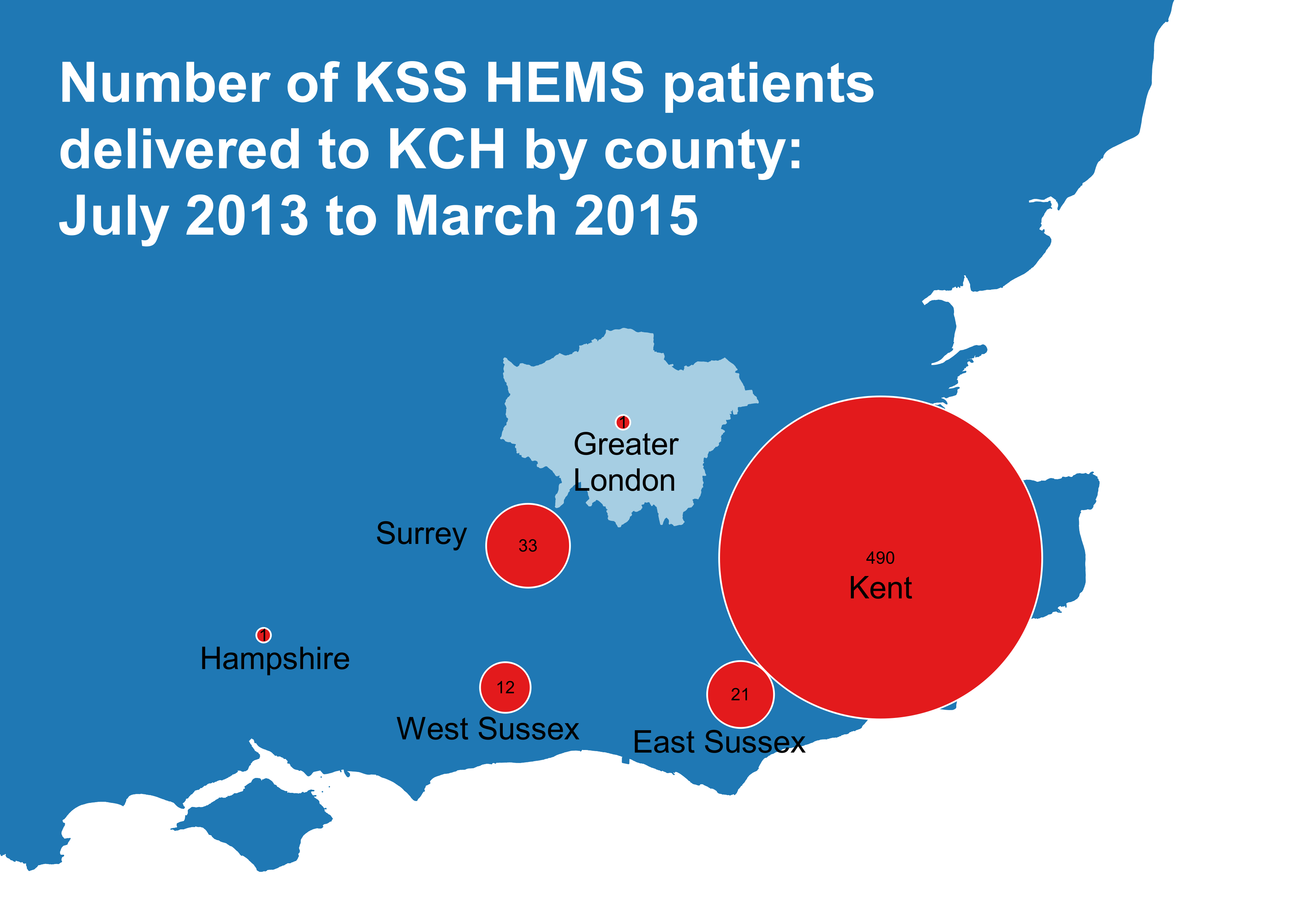
#Plot a histogram of the distances  
hist(data$distance, breaks = 25, prob=TRUE, col="grey", xlab = "Distance of transfers (km)", ylab = "Relative Frequency", main = "Histogram of Transfer Distances with Kernel Density Curve")  
  
#Overlay the kernel density plot of the distances  
lines(density(data$distance))

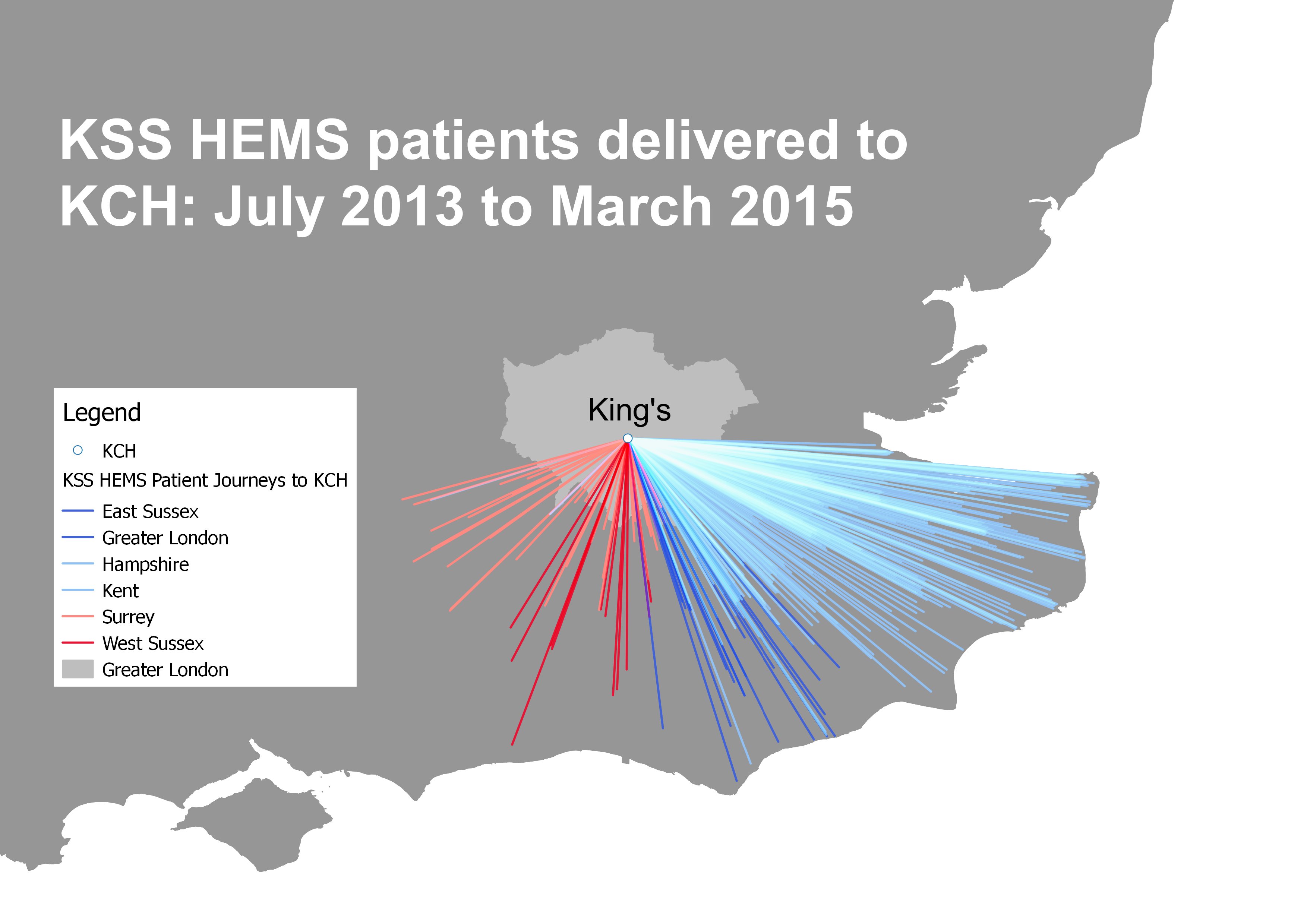


We will create a new field for drawing lines between the Head and Tail sites on QGIS in a Well-Known-Text (WKT) LINESTRING format

#Text formatting for WKT LINESTRING  
data$wkt <- paste0("LINESTRING(", data$lon, " ", data$lat, ",", KCH$lon, " ", KCH$lat, ")")  
  
#Save the dataframe as a .csv  
write.csv(data, file = "data/cleaned.csv")  
write.csv(KCH, "data/KCH.csv")

We can now see the QGIS mapping results below:





We want to see whether there is a temporal relationship with HEMS cases, and whether there St George's building their Helipad has had any effect on KCH's workload.

#Load the required libraries  
require(lubridate)  
require(dplyr)  
  
#We use the Lubridate package to manipulate the dates and time strings into POSIXct format so we can perform stats on them  
timestamp <- paste(data$Date, " ", data$Time)  
  
data$Date <- dmy(data$Date)  
data$Day <- wday(data$Date, label = TRUE)  
  
#Let's see if there is a difference between the cases depending of which day of the week they arrive  
count(data, Day)

## Source: local data frame [7 x 2]  
##   
## Day n  
## 1 Sun 90  
## 2 Mon 69  
## 3 Tues 72  
## 4 Wed 78  
## 5 Thurs 90  
## 6 Fri 89  
## 7 Sat 75

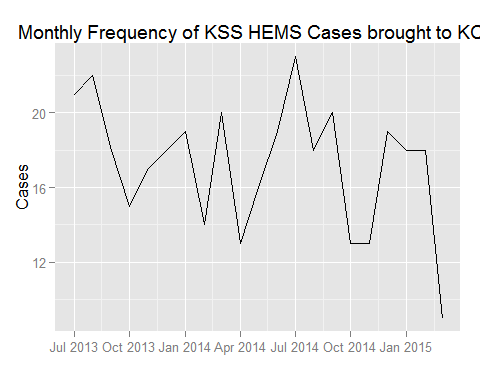
#Let's select the cases 6 months before St George's Helipad was constructed and 6 months after  
data\_before <- data %>% filter(Date >= dmy("01/10/2013") & Date < dmy("01/04/2014"))  
data\_after <- data %>% filter(Date >= dmy("01/04/2014") & Date < dmy("01/10/2014"))   
  
count(data\_before, County)

## Source: local data frame [6 x 2]  
##   
## County n  
## 1 East Sussex 3  
## 2 Greater London 1  
## 3 Kent 132  
## 4 Surrey 12  
## 5 West Sussex 5  
## 6 NA 5

count(data\_after, County)

## Source: local data frame [4 x 2]  
##   
## County n  
## 1 East Sussex 10  
## 2 Kent 150  
## 3 Surrey 4  
## 4 West Sussex 3

#Let's draw a time series of the cases, by monthly frequency  
case\_count <- count(data, Date)  
case\_count$Month <- as.Date(cut(case\_count$Date, breaks = "month"))  
monthly\_count <- count(case\_count, Month)  
  
#Load ggplot2 to graph  
require(ggplot2)  
  
#Plot  
ggplot(monthly\_count, aes(Month, n)) +   
 geom\_line() +   
 xlab("") +   
 ylab("Cases") +   
 ggtitle("Monthly Frequency of KSS HEMS Cases brought to KCH")



#Let's now draw a time series of the cases, by weekly frequency this time  
case\_count$Week <- as.Date(cut(case\_count$Date, breaks = "week"))  
weekly\_count <- count(case\_count, Week)  
  
#Plot  
ggplot(weekly\_count, aes(Week, n)) +   
 geom\_line() +   
 xlab("") +   
 ylab("Cases") +  
 ggtitle("Weekly Frequency of KSS HEMS Cases brought to KCH")

