INA CNG

Chris Buddenhagen

25/11/2021

## High risk farms and CNG points

## Reading layer `Farms\_close to\_CNG' from data source   
## `C:\Projects\SleeperWeeds\Farms\_close to\_CNG.shp' using driver `ESRI Shapefile'  
## Simple feature collection with 12885 features and 136 fields  
## Geometry type: MULTIPOLYGON  
## Dimension: XY  
## Bounding box: xmin: 176.0506 ymin: -40.43129 xmax: 178.0017 ymax: -38.775  
## Geodetic CRS: WGS 84

## xmin ymin xmax ymax   
## 176.05061 -40.43129 178.00167 -38.77500

## Reading layer `river-environment-classification-new-zealand-2010-deprecated' from data source `C:\Projects\SleeperWeeds\Inputs\rivers\_mfe\river-environment-classification-new-zealand-2010-deprecated.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 576688 features and 25 fields  
## Geometry type: LINESTRING  
## Dimension: XY  
## Bounding box: xmin: 18527310 ymin: -5956976 xmax: 19875650 ymax: -4058648  
## Projected CRS: WGS 84 / World Mercator

## Warning: attribute variables are assumed to be spatially constant throughout all  
## geometries

## Reading layer `nz-coastlines-and-islands-polygons-topo-150k' from data source   
## `C:\Projects\SleeperWeeds\Inputs\coast\nz-coastlines-and-islands-polygons-topo-150k.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 9136 features and 7 fields  
## Geometry type: MULTIPOLYGON  
## Dimension: XY  
## Bounding box: xmin: 165.869 ymin: -52.62088 xmax: 183.8457 ymax: -29.23134  
## Geodetic CRS: NZGD2000

## Warning: attribute variables are assumed to be spatially constant throughout all  
## geometries

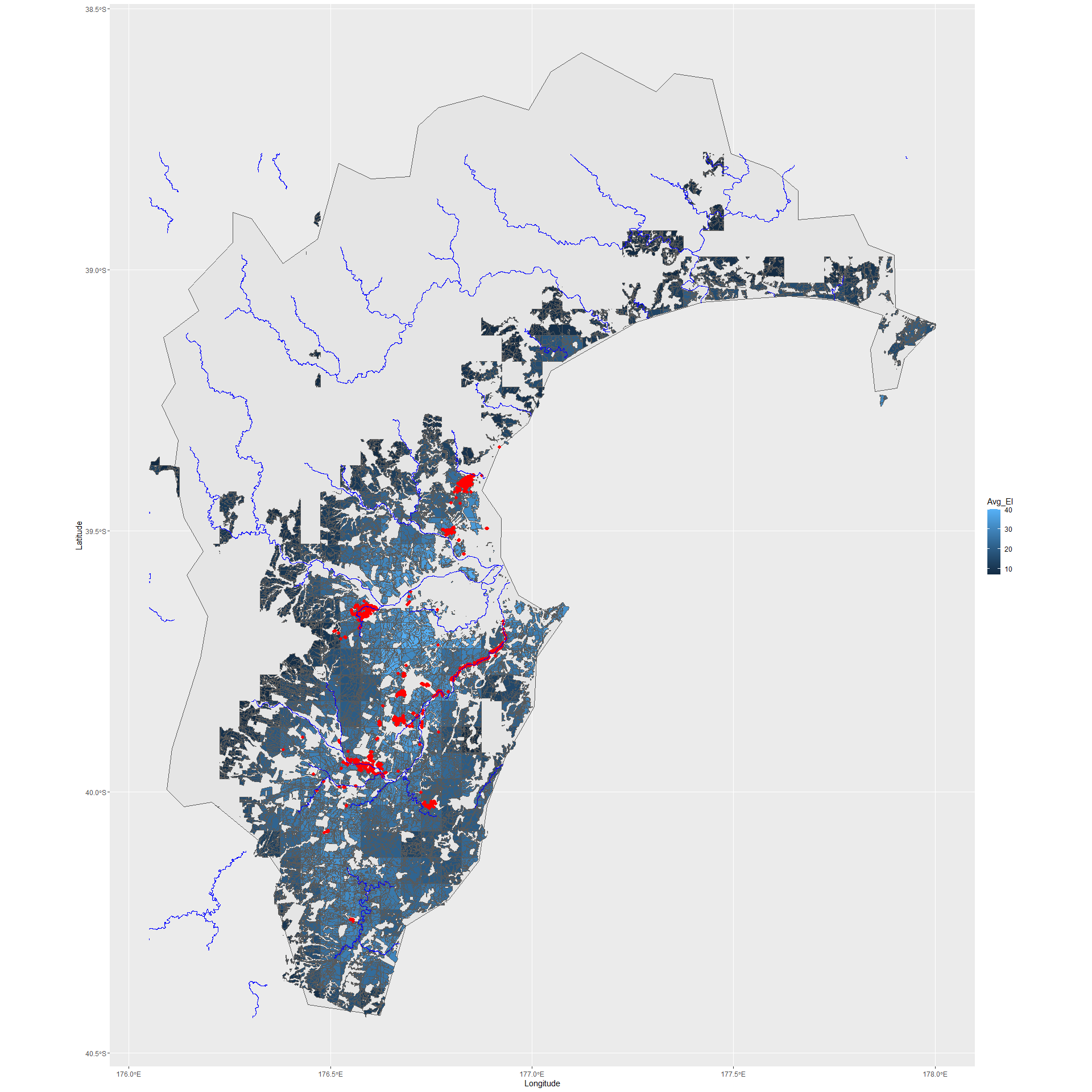
## [1] "sf" "data.frame"

## xmin ymin xmax ymax   
## 176.08088 -40.42802 178.00167 -38.58392

## xmin ymin xmax ymax   
## 176.05061 -40.43129 178.00167 -38.77500

## Plot farms and Climex average index

A plot example:



#Get centroids

cent\_HighRiskFarms<- st\_point\_on\_surface(HighRiskFarms)

## Warning in st\_point\_on\_surface.sf(HighRiskFarms): st\_point\_on\_surface assumes  
## attributes are constant over geometries of x

## Warning in st\_point\_on\_surface.sfc(st\_geometry(x)): st\_point\_on\_surface may not  
## give correct results for longitude/latitude data

HighRiskFarms<-HighRiskFarms %>%   
 mutate(has\_cng=if\_else(buffered\_1>0,1,0))  
  
xy<-st\_coordinates(cent\_HighRiskFarms)  
add\_cents<-cbind(xy,cent\_HighRiskFarms)  
  
HighRiskFarms<-cbind(xy,HighRiskFarms)  
# class(add\_cents)  
# names(add\_cents)  
#add\_cents$buffered.1  
  
#Change column header  
names(HighRiskFarms)[138]<-"area\_with\_CNG"  
  
#wrote this but you cannot over-write  
  
  
#st\_write(HighRiskFarms, "Inputs/HighRiskFarms.shp")  
#HighRiskFarms<-st\_read("Inputs/HighRiskFarms.shp")  
class(HighRiskFarms$CNG\_POINTS)

## [1] "numeric"

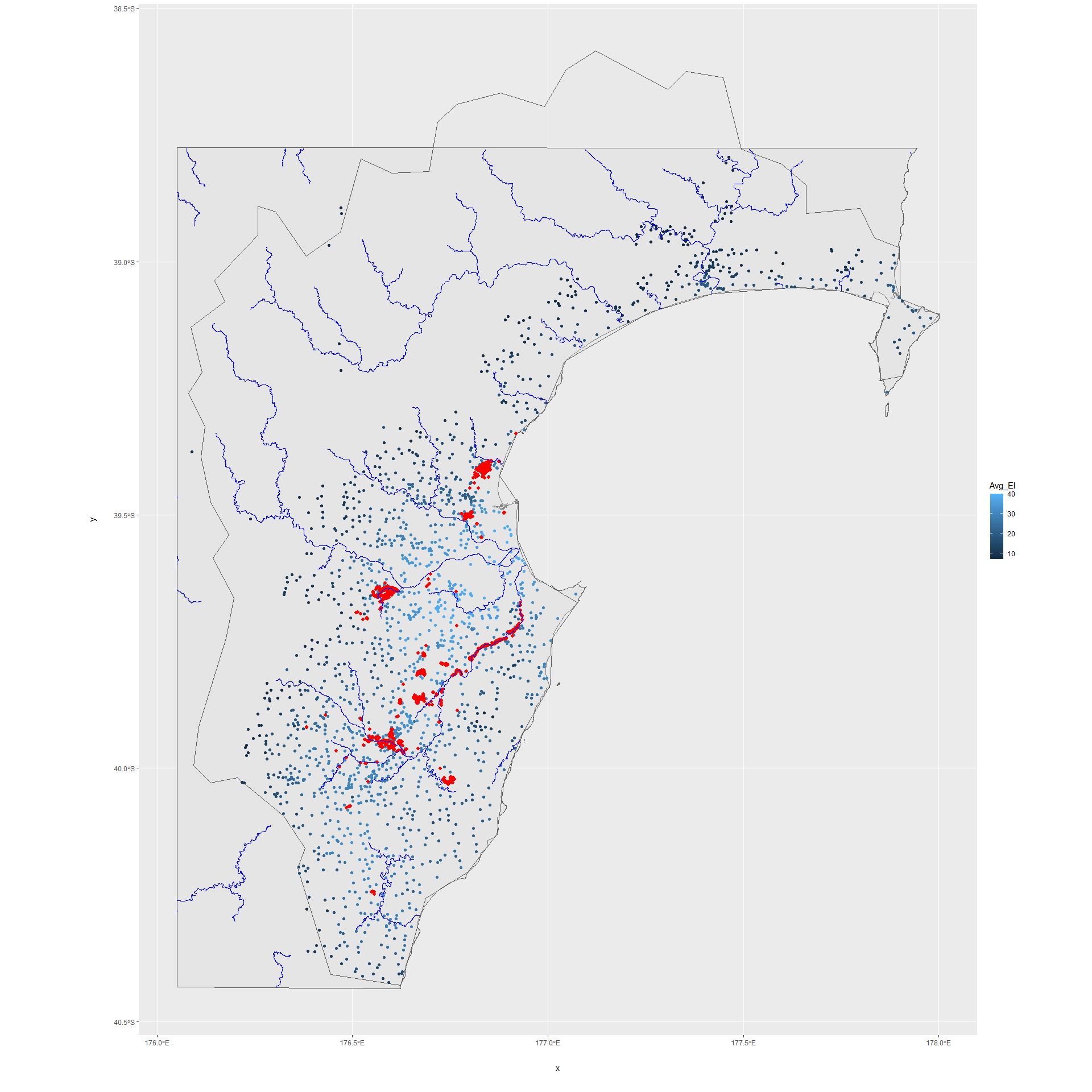
#deal with multipart farms group them and add xy for centroid  
HighRiskGrouped<-HighRiskFarms %>%   
 group\_by(FARM\_ID) %>%   
 summarize(SIZE\_HA= sum(SIZE\_HA), area\_with\_CNG=sum(area\_with\_CNG),Avg\_EI= mean(Avg\_EI), CNG\_POINTS=sum(CNG\_POINTS), POSTAL\_TWN=first(POSTAL\_TWN), LOCALITY=first(LOCALITY), POSTAL\_COD=first(POSTAL\_COD))  
  
HighRiskGrouped<-HighRiskGrouped %>%   
 mutate(has\_cng=if\_else(area\_with\_CNG>0,1,0))   
  
cent\_HighRiskGrouped<- st\_point\_on\_surface(HighRiskGrouped)

## Warning in st\_point\_on\_surface.sf(HighRiskGrouped): st\_point\_on\_surface assumes  
## attributes are constant over geometries of x  
  
## Warning in st\_point\_on\_surface.sf(HighRiskGrouped): st\_point\_on\_surface may not  
## give correct results for longitude/latitude data

xy<-st\_coordinates(cent\_HighRiskGrouped)  
HighRiskGrouped<-cbind(xy,HighRiskGrouped)  
HighRiskGrouped$area\_with\_CNG<-HighRiskGrouped$area\_CNG  
#st\_write(HighRiskGrouped, "HighRiskGrouped.shp")  
Simple\_points<-HighRiskGrouped   
Simple\_points<-as\_tibble(bind\_cols(farm\_id=Simple\_points$FARM\_ID,x=Simple\_points$X, y= Simple\_points$Y, Avg\_EI=Simple\_points$Avg\_EI, has\_cng\_near=Simple\_points$has\_cng, has\_CNG\_pt=Simple\_points$CNG\_POINTS,size\_ha=Simple\_points$SIZE\_HA,Postal\_town=Simple\_points$POSTAL\_TWN, Postal\_code=Simple\_points$POSTAL\_COD, Locality=Simple\_points$LOCALITY))  
  
  
write\_csv(Simple\_points, "Simple\_points\_for\_INA.csv")

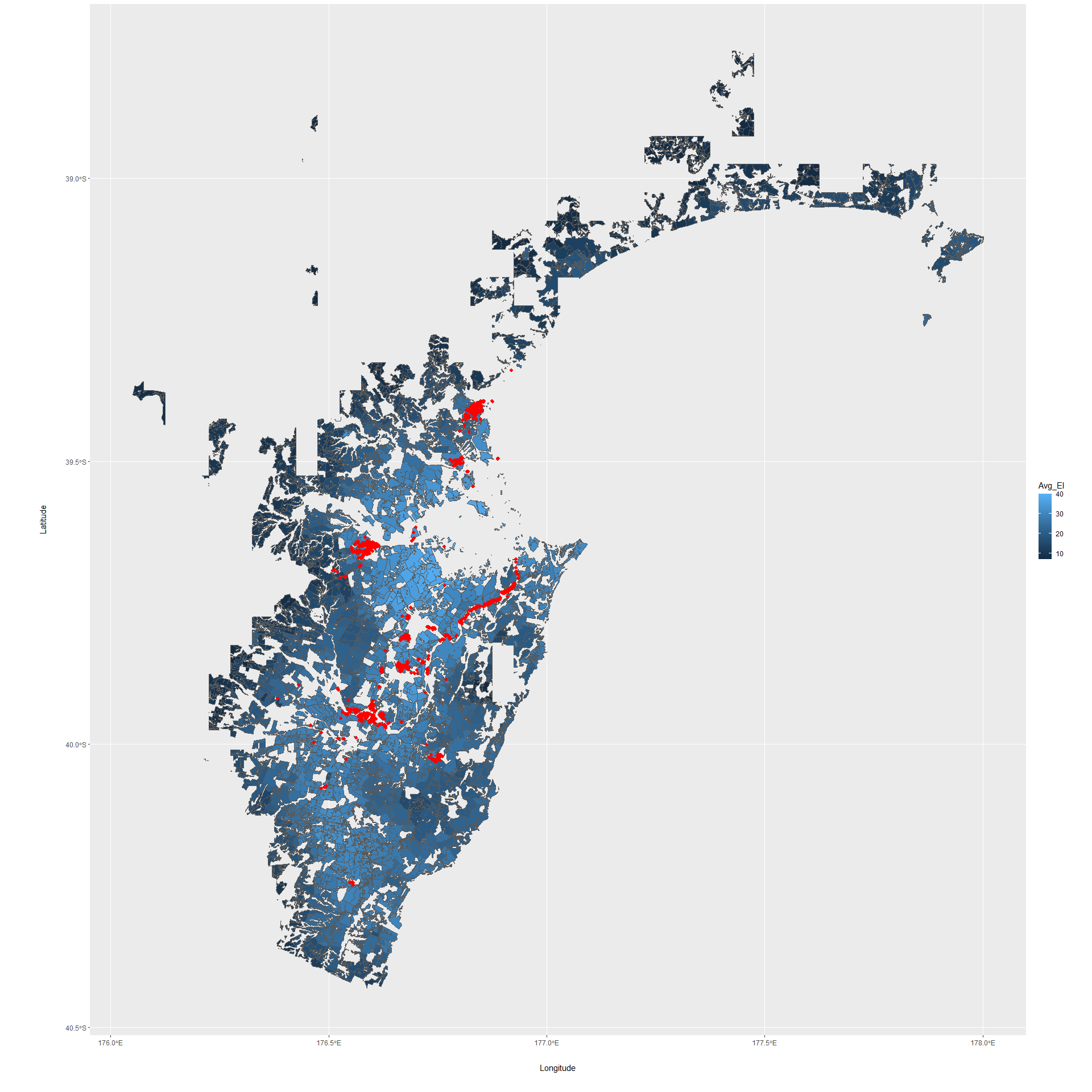
## Plot centroids.

ggplot() +  
 geom\_sf(data=HB\_coast)+  
 geom\_sf(data=HB, alpha=0.3)+  
 geom\_point(data=Simple\_points, aes(x=x, y=y, color=Avg\_EI))+  
 theme(legend.position = "right")+   
 geom\_point(data=CNGpoints, aes(x=Longitude, y=Latitude), shape=20, size=3, colour="red") +  
 geom\_sf(data=rivers4, colour="blue")



ggsave("Farm\_centroids\_CNG\_points.jpg", height=20, width=20, dpi=600)

ggplot() +  
 geom\_sf(data=HighRiskGrouped,#%>%   
 #filter(FARM\_ID=="HB00006"),  
 aes(fill=Avg\_EI))+  
 theme(legend.position = "right")+   
 geom\_point(data=CNGpoints, aes(x=Longitude, y=Latitude), shape=20, size=3, colour="red") #+



#geom\_sf(data=rivers4, colour="blue")  
ggsave("Farms\_CNG\_points\_waipawa.jpg", height=20, width=20, dpi=600)

shp\_1<- readOGR("HighRiskGrouped.shp")

## OGR data source with driver: ESRI Shapefile   
## Source: "C:\Projects\SleeperWeeds\HighRiskGrouped.shp", layer: "HighRiskGrouped"  
## with 1762 features  
## It has 10 fields

class(shp\_1)

## [1] "SpatialPolygonsDataFrame"  
## attr(,"package")  
## [1] "sp"

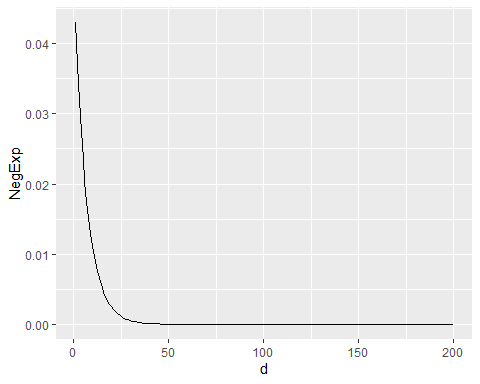
shp\_2<-spTransform(shp\_1, CRS("+init=epsg:2193") ) #NZGD2000

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer\_proj  
## = prefer\_proj): Discarded datum New\_Zealand\_Geodetic\_Datum\_2000 in Proj4  
## definition

#distance matrix code is slow  
  
#poly\_dist<-gDistance(shp\_2,byid = T)  
  
#saveRDS(poly\_dist, file="poly\_dist.rds")  
  
  
  
poly\_dist<-readRDS("poly\_dist.rds")  
poly\_dist[poly\_dist==0]<-0.01

graph.vertex.attr<-as\_tibble(bind\_cols(Number= as.numeric(seq(1:length(shp\_2$FARM\_ID))), farm\_id=as.character(shp\_2$FARM\_ID), area\_ha=as.numeric(shp\_2$SIZE\_HA), x=as.numeric(shp\_2$X), y=as.numeric(shp\_2$Y), town=shp\_2$POSTAL\_T))  
  
#CHECK THAT THE NODES ARE LABELLED GOOD  
  
  
# graph.vertex.attr%>%  
# ggplot(aes(x,y, label=Number))+  
# geom\_point()+  
# geom\_text(aes(label = farm\_id), hjust = -0.05,vjust = 0.05, size=2)  
  
  
#Basic spread with user determined start node one in Waipawa  
#?FromToDiseaseSpreadNegExp  
  
#check graph for distance by probability function need to add the threshold part to the function  
  
  
# #20 simulations with 1700 nodes 50 time steps about 15 minutes with lamda at 0.0009  
#   
# FToutNegExp<-FromToDiseaseSpreadNegExp(2, poly\_dist, lmda, 20, 50)  
#   
# small.list<-Make\_sim\_list\_for\_get.graph(FToutNegExp)  
#   
# #apply igraph generating function to the list of simulations  
#   
# #time consuming 10 minutes  
# graph.list.small<-lapply(small.list,   
# get\_graphs\_with\_extra\_attributes, df2=graph.vertex.attr)  
#   
# #started at 11 pm, finished at 11:17  
# listnodesinfected.small<-lapply(graph.list.small, get\_node\_numbers\_stepwise, MaxStep=50)  
#   
# #11:18 to   
# listnodesinfected\_df.small<-tibble::as\_tibble(do.call(rbind, listnodesinfected.small))  
# dt<-listnodesinfected\_df.small

#Try threshold version



## [1] "2021-12-17 10:34:57 NZDT"

## [1] "2021-12-17 10:40:32 NZDT"

## [1] "2021-12-17 10:40:32 NZDT"

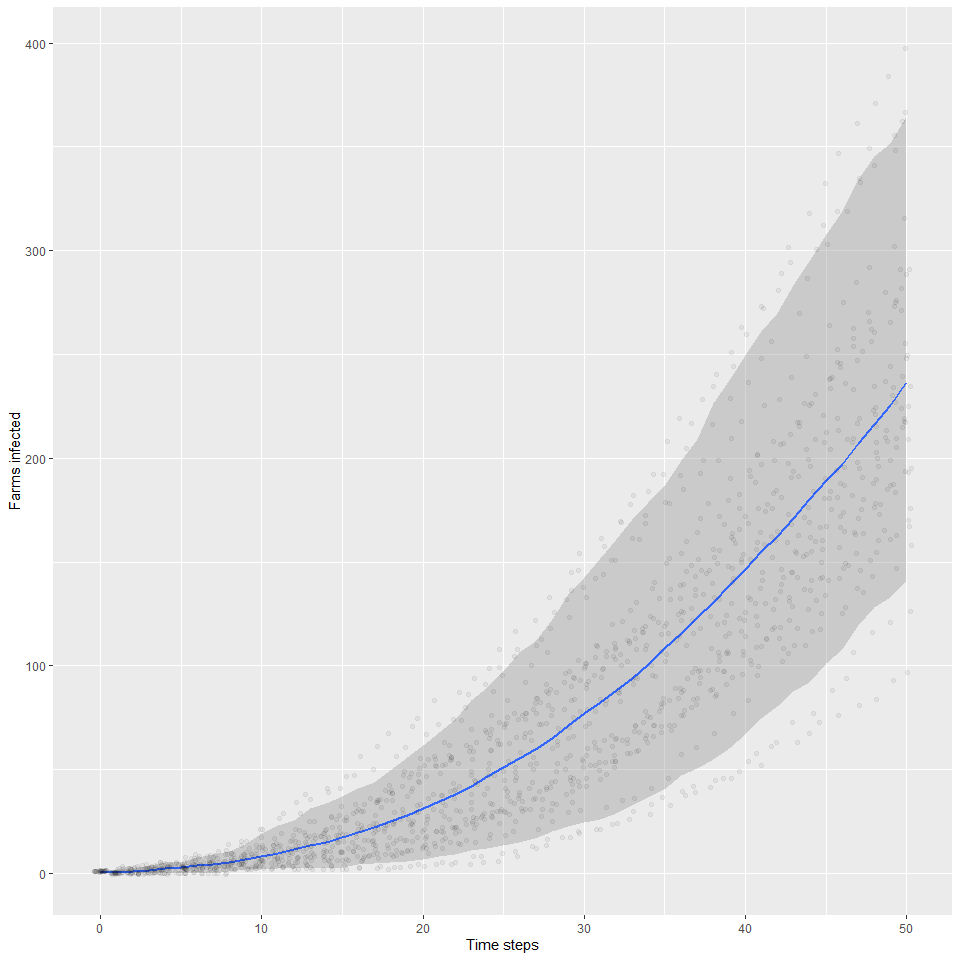
## [1] "2021-12-17 10:40:34 NZDT"

## [1] "2021-12-17 10:40:38 NZDT"

## [1] "2021-12-17 10:40:38 NZDT"

## How many nodes were infected?

p<-ggplot(dt, aes(Step, nodesinfected)) + geom\_smooth(stat = 'summary', fun.data = function(y) data.frame(ymin = quantile(y, .05), y = mean(y), ymax = quantile(y, .95)))+geom\_jitter(alpha=0.05) +ylab("Farms infected")+xlab("Time steps")  
p

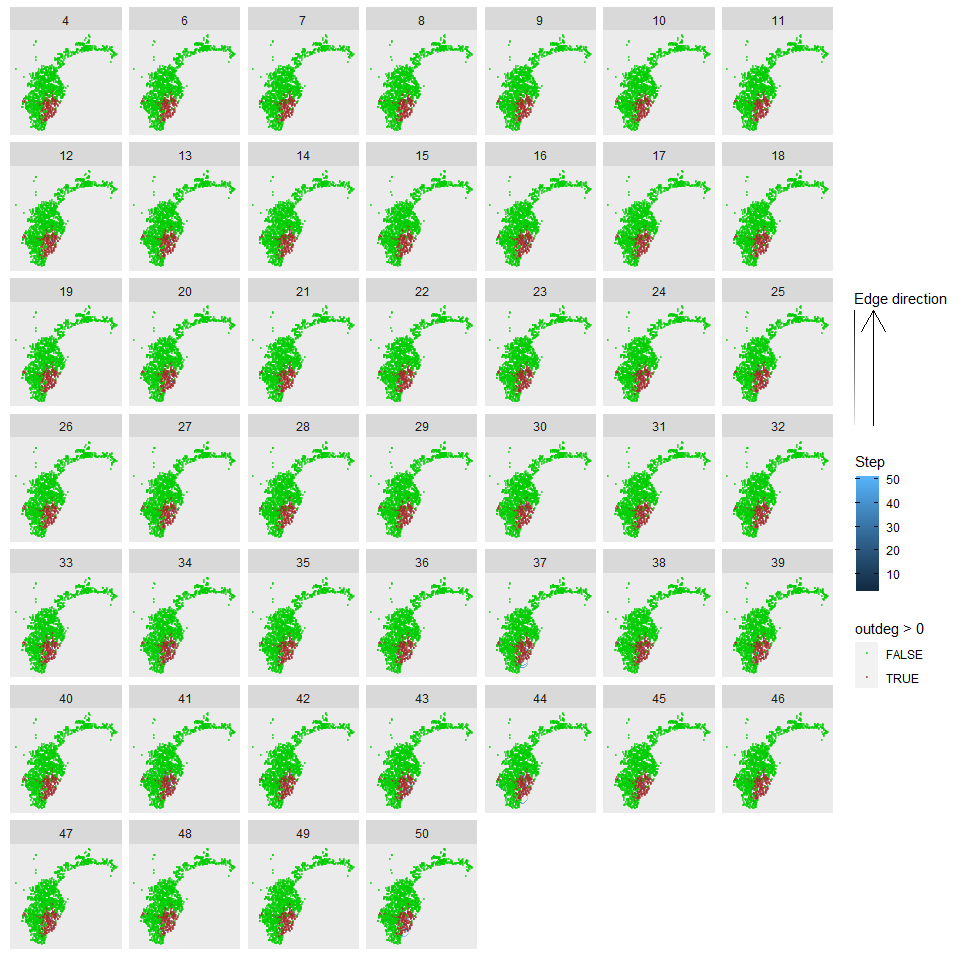


ggsave(paste0("Simulations starting at node 2 with lambda thresh timesteps ",lmda," ", thresh, " ", timesteps, ".jpg"), height=10, width=10, dpi=300)  
  
g<-graph.list.small[[2]]  
  
saveRDS(g, "simulation\_igraph.rds")  
#g  
#graph\_attr(g)  
# graph\_attr\_names(g)  
# V(g)$  
# V(g)$x  
# V(g)$y  
# V(g)$town  
  
   
g<-readRDS("simulation\_igraph.rds")  
 coords<-as.data.frame(as\_tibble(bind\_cols(x=V(g)$x, y=V(g)$y)))  
  
g2 <-as\_tbl\_graph(g)   
  
g2tibble<- as\_tibble(g2%>% activate(edges))  
  
g2\_tibble\_nodes<- as\_tibble(g2%>% activate(nodes))  
#V(g)$outdeg  
  
  
   
# vertex.attributes(g)  
# edge\_attr(g)  
library(ggraph)

##   
## Attaching package: 'ggraph'

## The following object is masked from 'package:sp':  
##   
## geometry

pg<-ggraph(g, x=coords$x, y=coords$y)+  
  
 #make gradient link  
 geom\_edge\_arc(aes(colour = Step, alpha = ..index..)) +  
 # geom\_edge\_fan(#aes(colour = Step),   
 # strength = 0.1)+  
 scale\_edge\_alpha('Edge direction', guide = 'edge\_direction')+  
 facet\_edges(~Step) + geom\_node\_point(aes(colour=outdeg>0),   
 size=0.1, alpha=0.5)+  
 scale\_colour\_manual(values=c("green3","brown"))   
#plotting graphic took longer than saving the graphic too large in both cases  
  
pg



ggsave(paste("Check network at lambda threshold timesteps ",lmda," ", thresh, " ", timesteps, ".pdf"), height=30, width=30, dpi=300)