spatial_descriptive_statistics

LingruFeng

2020/12/4

Getting Started

```
library(highcharter)
library(tidyverse)
library(downloader)
library(rgdal)
library(sf)
library(ggplot2)
library(reshape2)
library(plotly)
library(raster)
library(downloader)
library(rgdal)
```

read data

```
LondonWards <- st_read(here::here("prac8_data",
                                  "New_ward_data",
                                  "NewLondonWard.shp"))
```

```
## Reading layer `NewLondonWard' from data source `E:\STUDY\UCL\postgraduate\module\GIS\GIS_repo\week8\s
## Simple feature collection with 625 features and 76 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 503575 ymin: 155850.8 xmax: 561956.7 ymax: 200933.6
## projected CRS:  OSGB 1936 / British National Grid
```

add extra data

```
extradata <- read_csv(here::here("prac8_data", "LondonAdditionalDataFixed.csv"))
```

```
## Parsed with column specification:
## cols(
##   WardName = col_character(),
##   WardCode = col_character(),
##   Wardcode = col_character(),
##   PctSharedOwnership2011 = col_double(),
##   PctRentFree2011 = col_double(),
##   Candidate = col_character(),
##   InnerOuter = col_character(),
##   x = col_double(),
##   y = col_double(),
##   AvgGCSE2011 = col_double(),
##   UnauthAbsenceSchools11 = col_double()
## )
```

```
LondonWardsleftjoin <- LondonWards %>%
  left_join(.,extradata,
            by = c("WD11CD" = "Wardcode"))

#LondonWardsSF <- merge(LondonWards, extradata, by.x = "WD11CD", by.y = "Wardcode")
```

Task 1 - Descriptive Statistics

```
summary(extradata$AvgGCSE2011)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   245.0   332.3   343.7   345.8   358.3   409.1
```

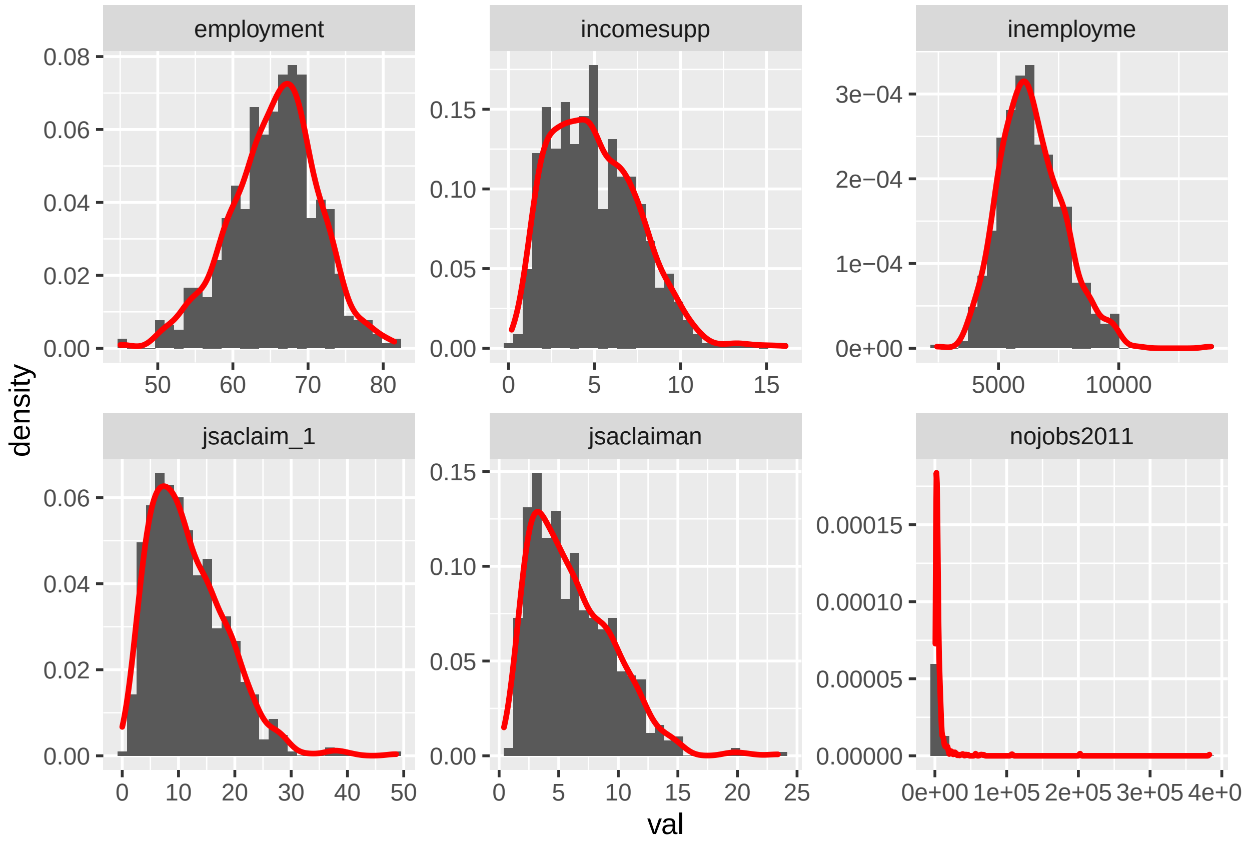
```
#check which variables are numeric first

Datatypelist <- LondonWardsleftjoin %>%
  st_drop_geometry()%>%
  summarise_all(class) %>%
  pivot_longer(everything(),
               names_to="All_variables",
               values_to="Variable_class")

#make groups based on types of variables
Groups <- LondonWardsleftjoin %>%
  st_drop_geometry()%>%
  dplyr::select(is.numeric)%>%
  pivot_longer(everything(),
               names_to="All_variables",
               values_to="val")%>%
  mutate(All_variables = tolower(All_variables))%>%
  mutate(group = case_when(str_detect(All_variables, "age") ~ "Age",
                           str_detect(All_variables, "employ|income|job|jsa") ~ "Employment",
                           str_detect(All_variables, "house|rent|detatched|flat|terrace|owned|social|pr
```

```
## Warning: Predicate functions must be wrapped in `where()`.
##
##   # Bad
##   data %>% select(is.numeric)
##
```

```
##   # Good
##   data %>% select(where(is.numeric))
##
## i Please update your code.
## This message is displayed once per session.
```

```
Employmenthist <- Groups%>%
  filter(group=="Employment")%>%
  ggplot(., aes(x=val)) +
  geom_histogram(aes(x = val, y = ..density..))+
  geom_density(colour="red", size=1, adjust=1)+
  facet_wrap(~All_variables, scales = 'free')

print(Employmenthist)
```

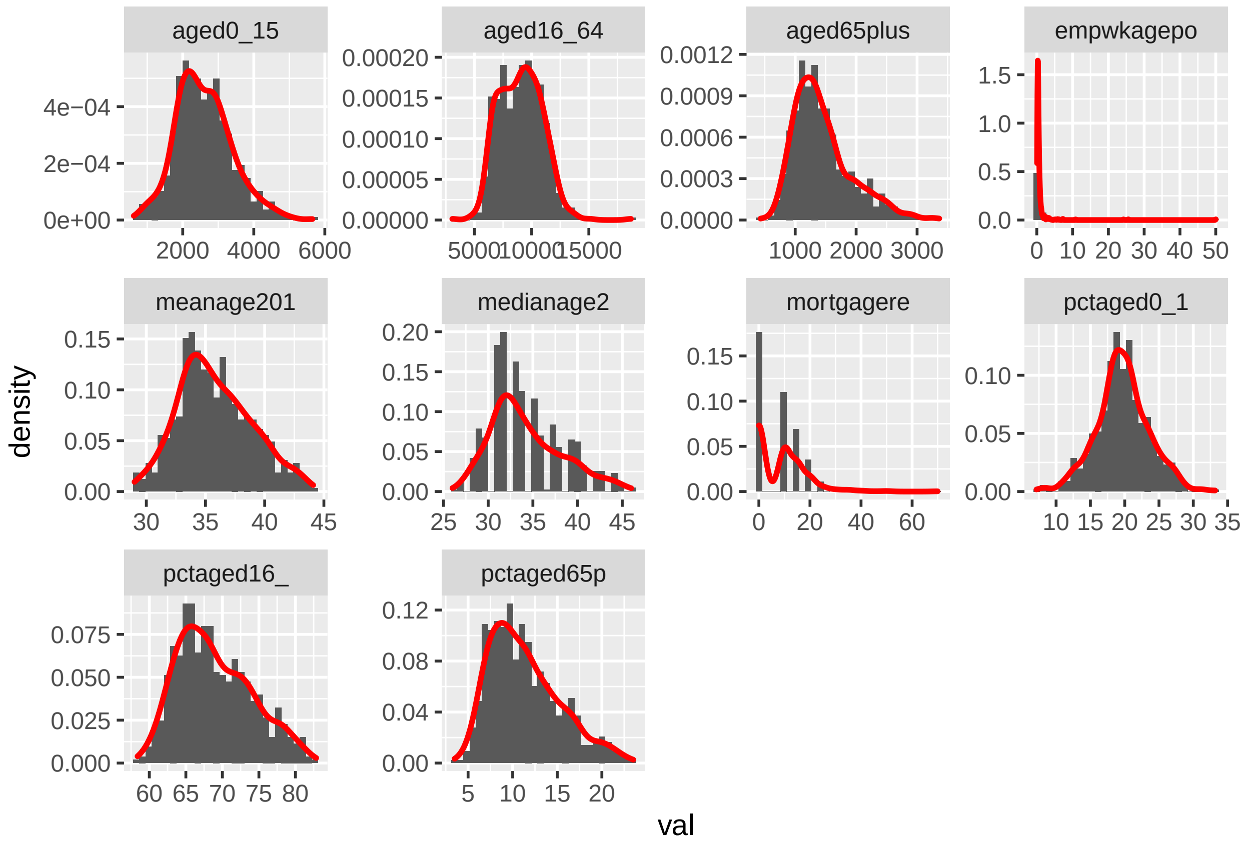
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



Plot Histogram by age

```
Agehist1 <- Groups%>%
  filter(group=="Age")%>%
  ggplot(., aes(x=val)) +
  geom_histogram(aes(x = val, y = ..density..))+
  geom_density(colour="red", size=1, adjust=1)+
  facet_wrap(~All_variables, scales = 'free')
Agehist1
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



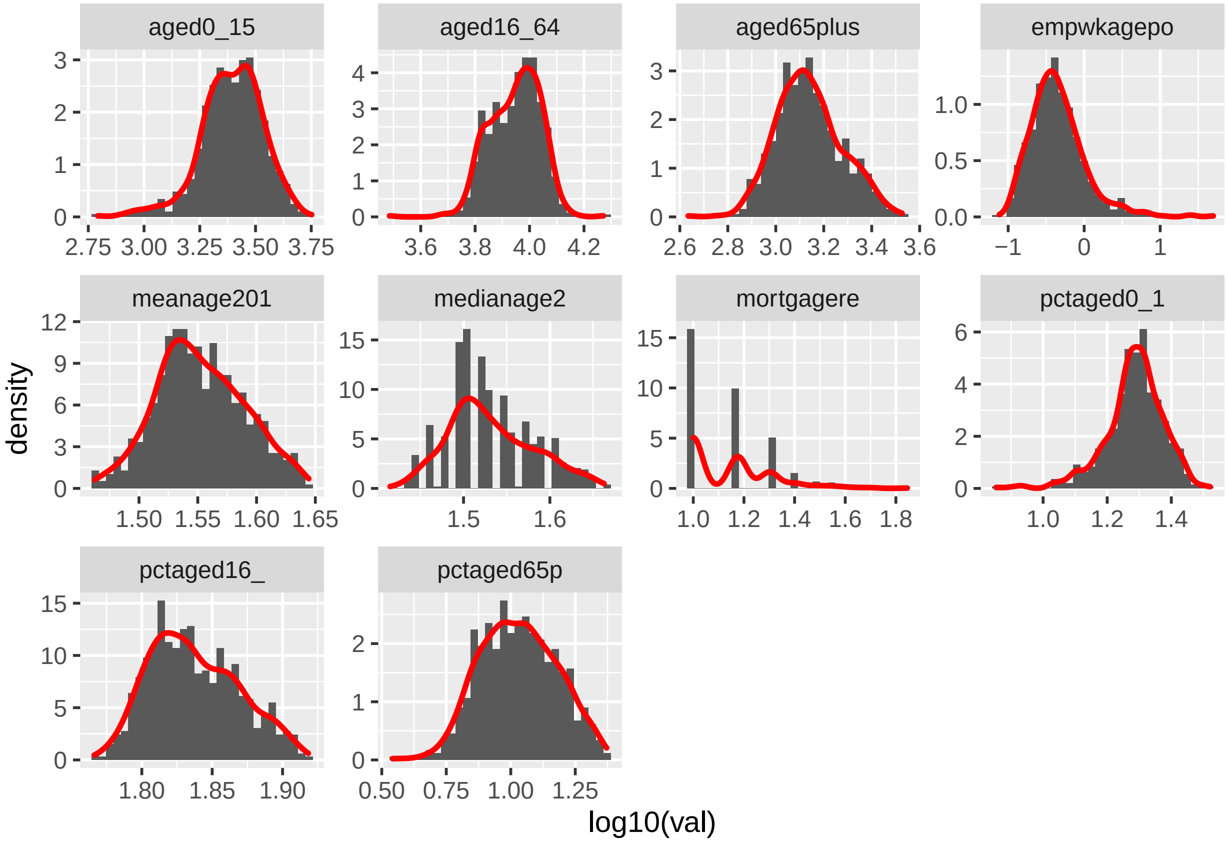
Log the age data

```
Agehist <- Groups%>%
  filter(group=="Age")%>%
  ggplot(., aes(x=log10(val))) +
  geom_histogram(aes(x = log10(val), y = ..density..))+
  geom_density(colour="red", size=1, adjust=1)+
  facet_wrap(~All_variables, scales = 'free')
Agehist
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

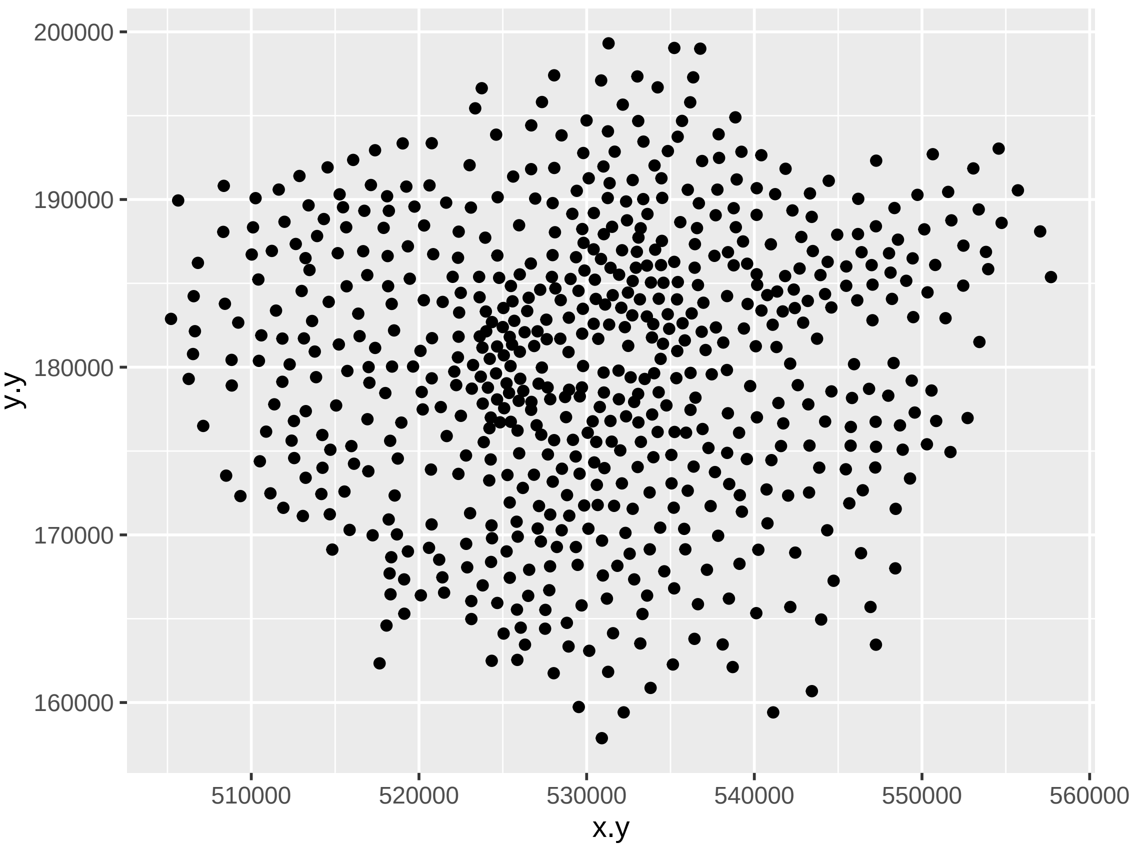
```
## Warning: Removed 266 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 266 rows containing non-finite values (stat_density).
```

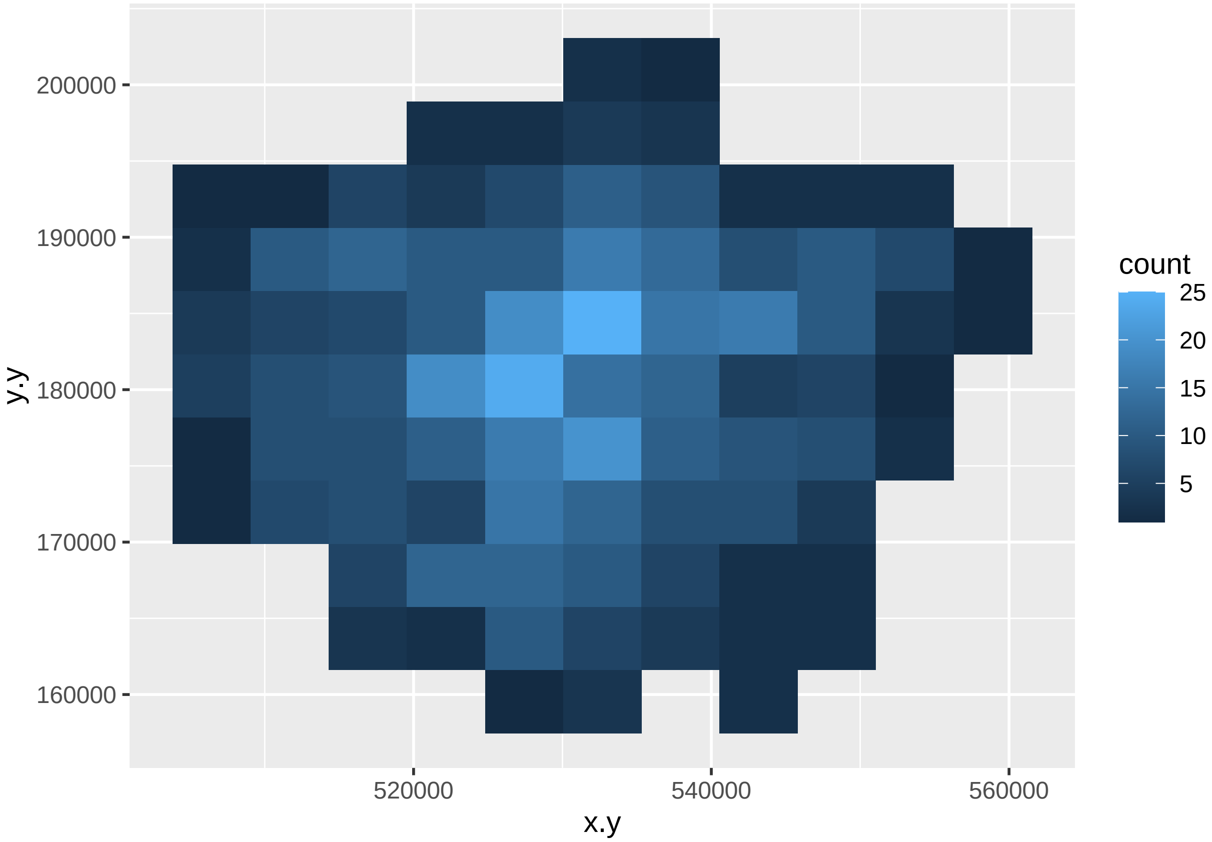


Using Eastings and Northings data in the X and Y columns of the dataset, a 2D histogram and a 2D core density estimate of the Ward Centroids in London were created

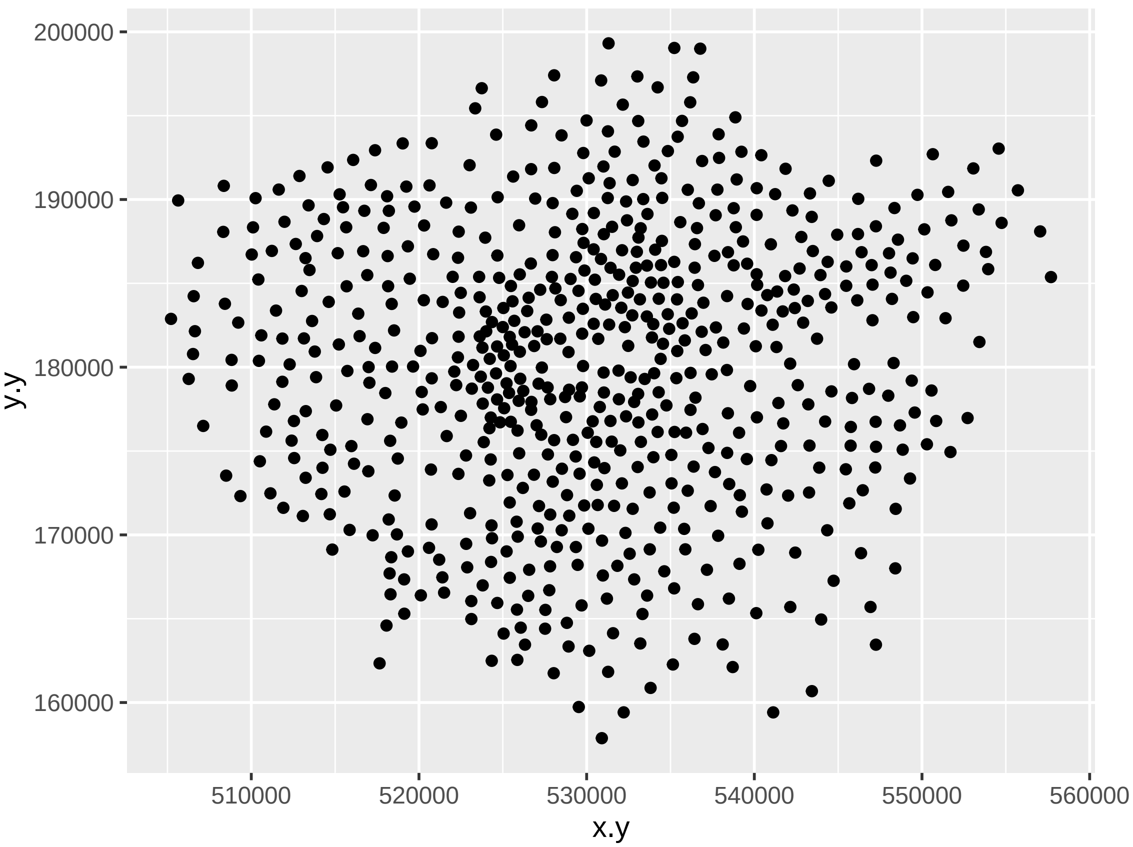
```
Londonpoint <- ggplot(LondonWardsleftjoin, aes(x=x.y,y=y.y))+geom_point()+coord_equal()
Londonpoint
```



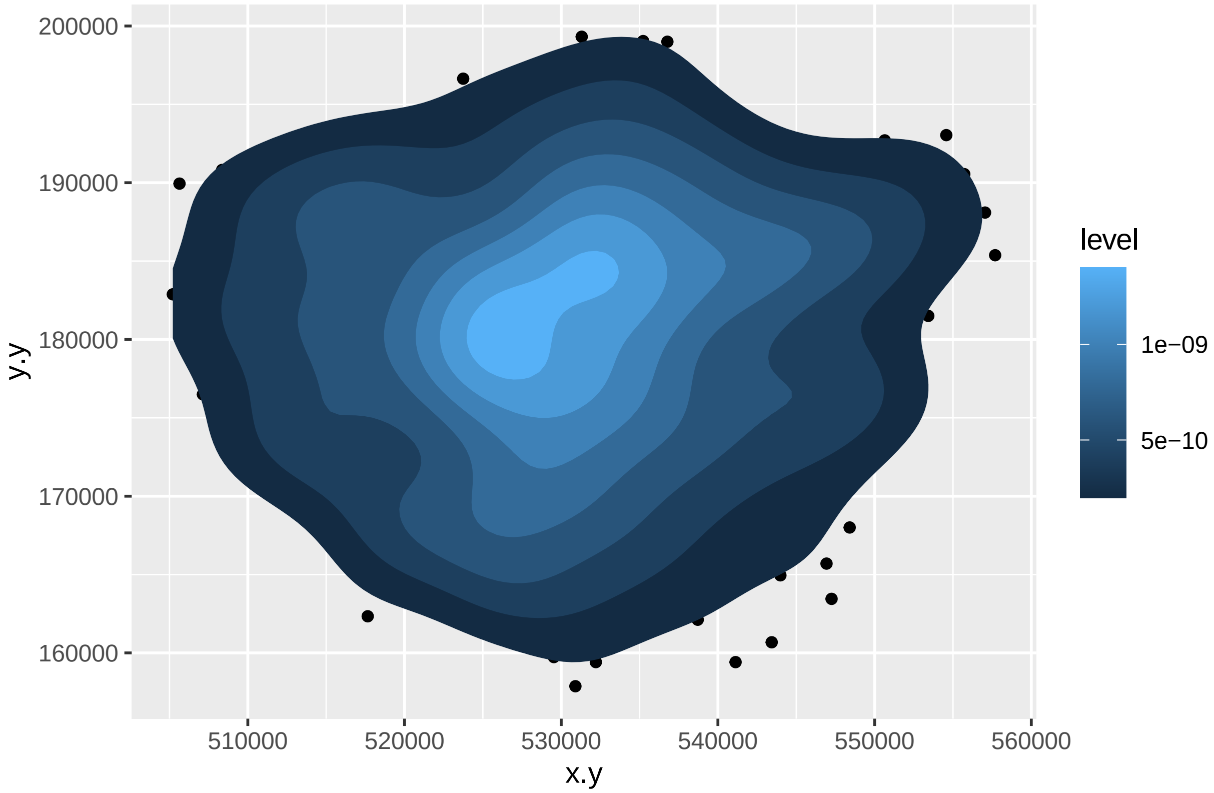
```
Londonpoint<-ggplot(LondonWardsleftjoin, aes(x=x.y,y=y.y))+stat_bin2d(bins=10)
Londonpoint
```



```
Londonpoint<-ggplot(LondonWardsleftjoin, aes(x=x.y,y=y.y))+geom_point()+coord_equal()
Londonpoint
```



```
Londonpoint+stat_density2d(aes(fill = ..level..), geom="polygon")
```



Task 2 - Function to recode data

```
newvar<-0
recode<-function(variable,high,medium,low){
  newvar[variable<=high]<-"High"
  newvar[variable<=medium]<-"Medium"
  newvar[variable<=low]<-"Low"
  return(newvar)
}
```

```
attach(LondonWards)
#Check the name of your column, there could be a slight error and it might be called 'AvgGCSED201'
summary(LondonWards$AvgGCSE201)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   245.0   332.3   343.7   345.8   358.3   409.1
```

```
LondonWards$GCSE_recode <- recode(AvgGCSE201,409.1,358.3,332.3)
```

```
#Location Quotient function 1
LQ1<-function(pctVariable){
  pctVariable /mean(pctVariable)
}
#Location Quotient function 2
LQ2<-function(variable,rowtotal){
  localprop<-variable/rowtotal
  globalprop<-sum(variable)/sum(rowtotal)
  return(localprop/globalprop)
}
```

```
head(LondonWards[,1:7])
```

```
## Simple feature collection with 6 features and 7 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 507996.8 ymin: 170317.9 xmax: 533838 ymax: 182206.1
## projected CRS:  OSGB 1936 / British National Grid
##      WD11CD WD11CDO         WD11NM WD11NMW                  WardName WardCode
## 1 E09000001    00AA City of London    <NA>            City of London     00AA
## 2 E05000352  00ATGE   Feltham West    <NA>   Hounslow - Feltham West   00ATGE
## 3 E05000353  00ATGF       Hanworth    <NA>       Hounslow - Hanworth   00ATGF
## 4 E05000354  00ATGG  Hanworth Park    <NA>  Hounslow - Hanworth Park   00ATGG
## 5 E05000355  00ATGH Heston Central    <NA> Hounslow - Heston Central   00ATGH
## 6 E05000356  00ATGJ    Heston East    <NA>    Hounslow - Heston East   00ATGJ
##   Wardcode1                       geometry
## 1      <NA> MULTIPOLYGON (((532134.9 18...
## 2 E05000352 MULTIPOLYGON (((509740 1736...
## 3 E05000353 MULTIPOLYGON (((513585.1 17...
## 4 E05000354 MULTIPOLYGON (((512142.6 17...
## 5 E05000355 MULTIPOLYGON (((513098.5 17...
## 6 E05000356 MULTIPOLYGON (((513467.2 17...
```

use function

```
#this is pseudo code, but you should see how this works
LondonWards$LQ_PctAged0_15 <- LQ1(PctAged0_1)
#or
LondonWards$LQ_Aged0_15 <- LQ2(Aged0_15,PopCensus2)

LondonWards <- LondonWards %>%
  mutate(LQ_Aged16_65=LQ1(PctAged16_))

summary(LondonWards$LQ_Aged0_15)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3589  0.8815  0.9828  0.9905  1.1034  1.6696
```

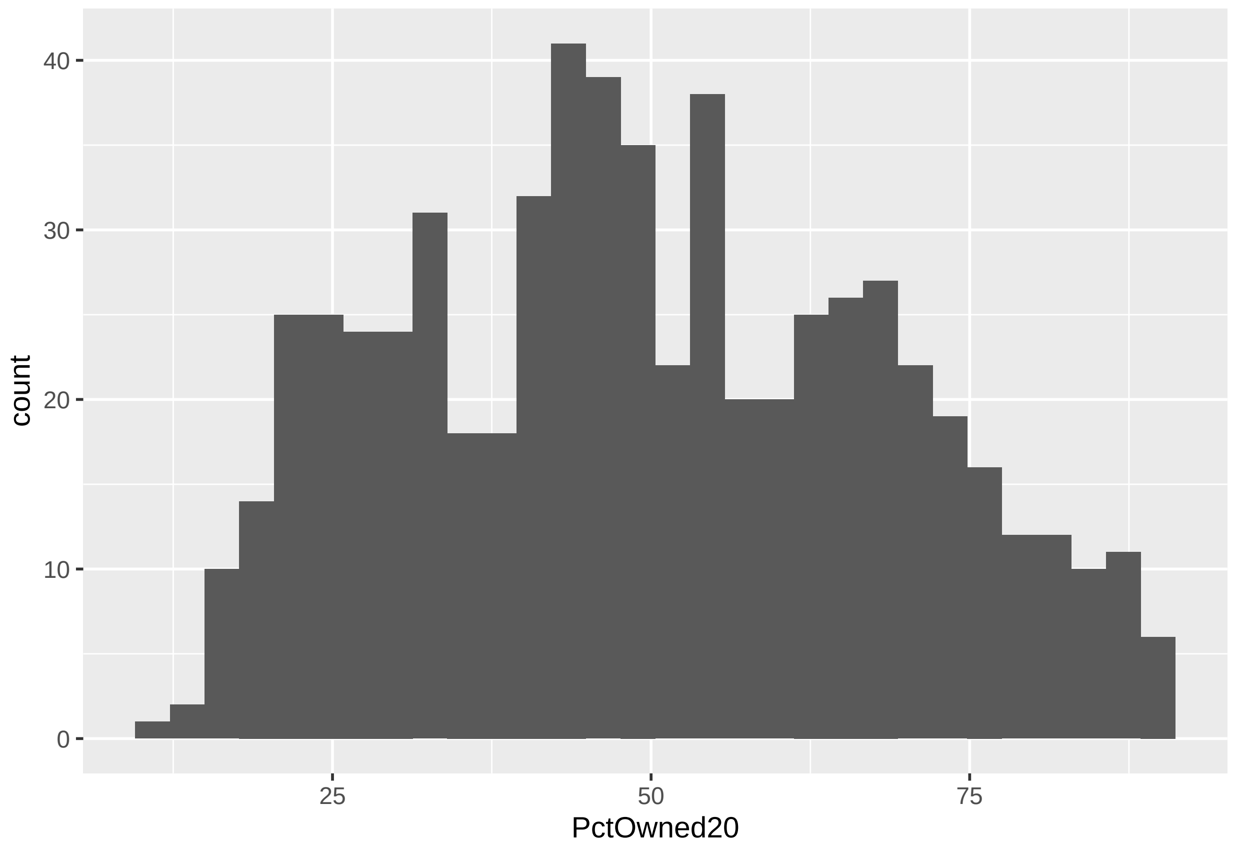
Creating a Basic Geodemographic Classification

```
LondonWardsData <- LondonWards %>%
  #drop geometry
  st_drop_geometry()%>%
  #display list of variables
  summarise_all(class) %>%
  pivot_longer(everything(),
               names_to="All_variables",
               values_to="Variable_class")

slice_head(LondonWardsData, n=5)
```

```
## # A tibble: 5 x 2
##   All_variables Variable_class
##   <chr>         <chr>
## 1 WD11CD        character
## 2 WD11CDO       character
## 3 WD11NM        character
## 4 WD11NMW       character
## 5 WardName      character
```

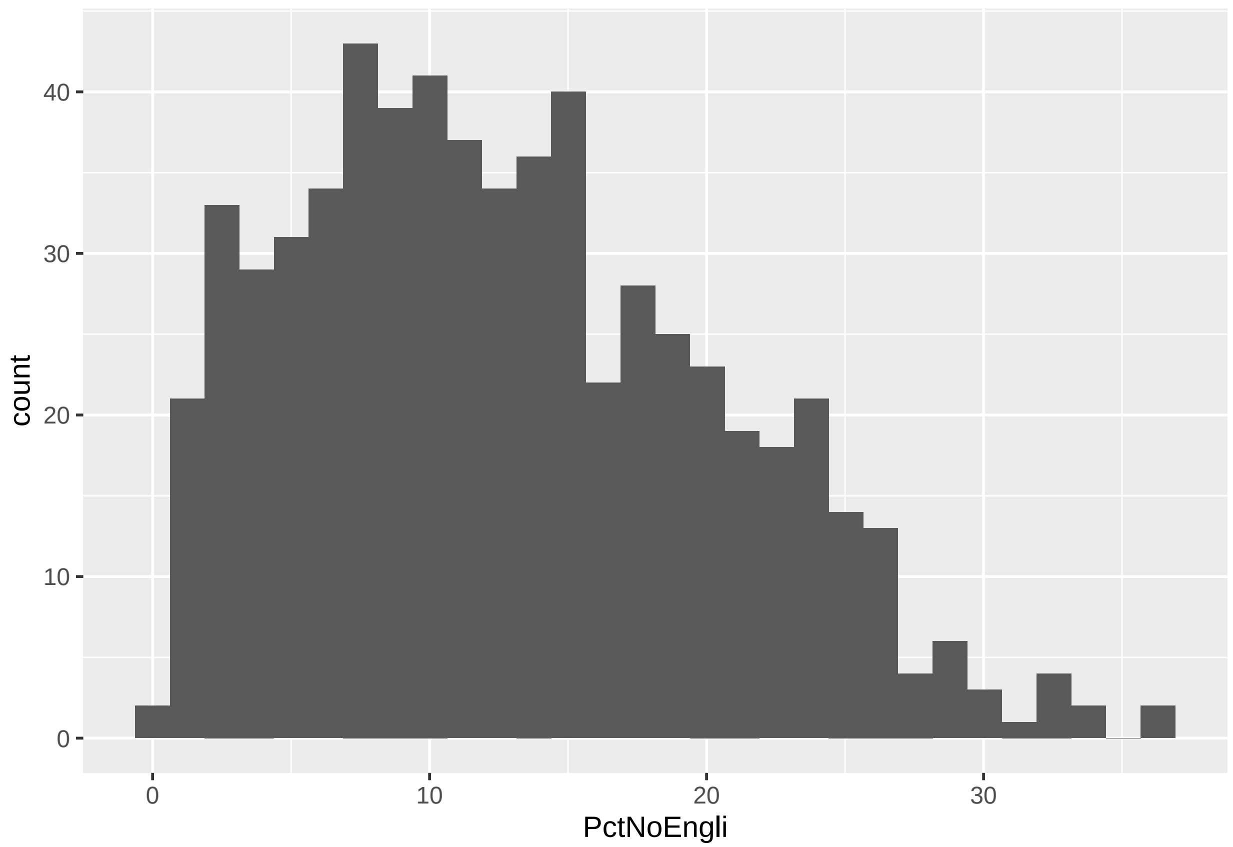
```
# Create a new data frame just containing the two variables we are interested in
mydata <- LondonWards %>%
      st_drop_geometry()%>%
      dplyr::select(c(PctOwned20, PctNoEngli))

#- check variable distributions first
histplot <- ggplot(data=mydata, aes(x=PctOwned20))
histplot +geom_histogram()
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
histplot <- ggplot(data=mydata, aes(x= PctNoEngli))
histplot +geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
fit <- mydata %>%
  kmeans(., 3, nstart=25)
```

```
# get cluster means
library(tidymodels)
```

```
## Warning: package 'tidymodels' was built under R version 4.0.3

## -- Attaching packages -------------------------------------- tidymodels 0.1.2 --

## v broom     0.7.2      v recipes   0.1.15
## v dials     0.0.9      v rsample   0.0.8
## v infer     0.5.3      v tune      0.1.2
## v modeldata 0.1.0      v workflows 0.2.1
## v parsnip   0.1.4      v yardstick 0.0.7

## Warning: package 'broom' was built under R version 4.0.3

## Warning: package 'dials' was built under R version 4.0.3

## Warning: package 'infer' was built under R version 4.0.3

## Warning: package 'modeldata' was built under R version 4.0.3
## Warning: package ’parsnip’ was built under R version 4.0.3
```

```
## Warning: package ’recipes’ was built under R version 4.0.3
```

```
## Warning: package ’rsample’ was built under R version 4.0.3
```

```
## Warning: package ’tune’ was built under R version 4.0.3
```

```
## Warning: package ’workflows’ was built under R version 4.0.3
```

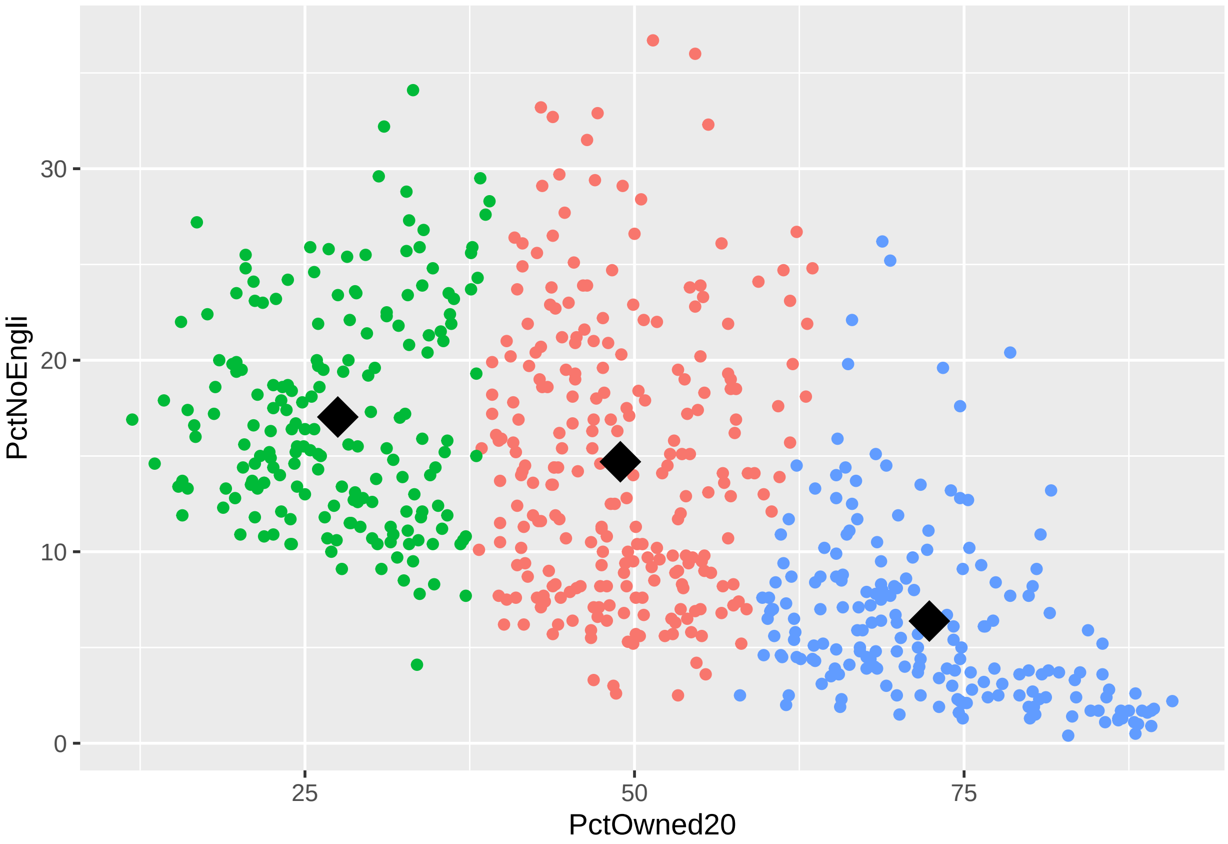
```
## Warning: package ’yardstick’ was built under R version 4.0.3
```

```
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard()    masks purrr::discard()
## x raster::extract()    masks tidyr::extract()
## x plotly::filter()     masks dplyr::filter(), stats::filter()
## x recipes::fixed()     masks stringr::fixed()
## x dplyr::lag()         masks stats::lag()
## x .GlobalEnv::recode() masks dplyr::recode()
## x raster::select()     masks plotly::select(), dplyr::select()
## x yardstick::spec()    masks readr::spec()
## x recipes::step()      masks stats::step()
## x recipes::update()    masks raster::update(), stats::update()
```

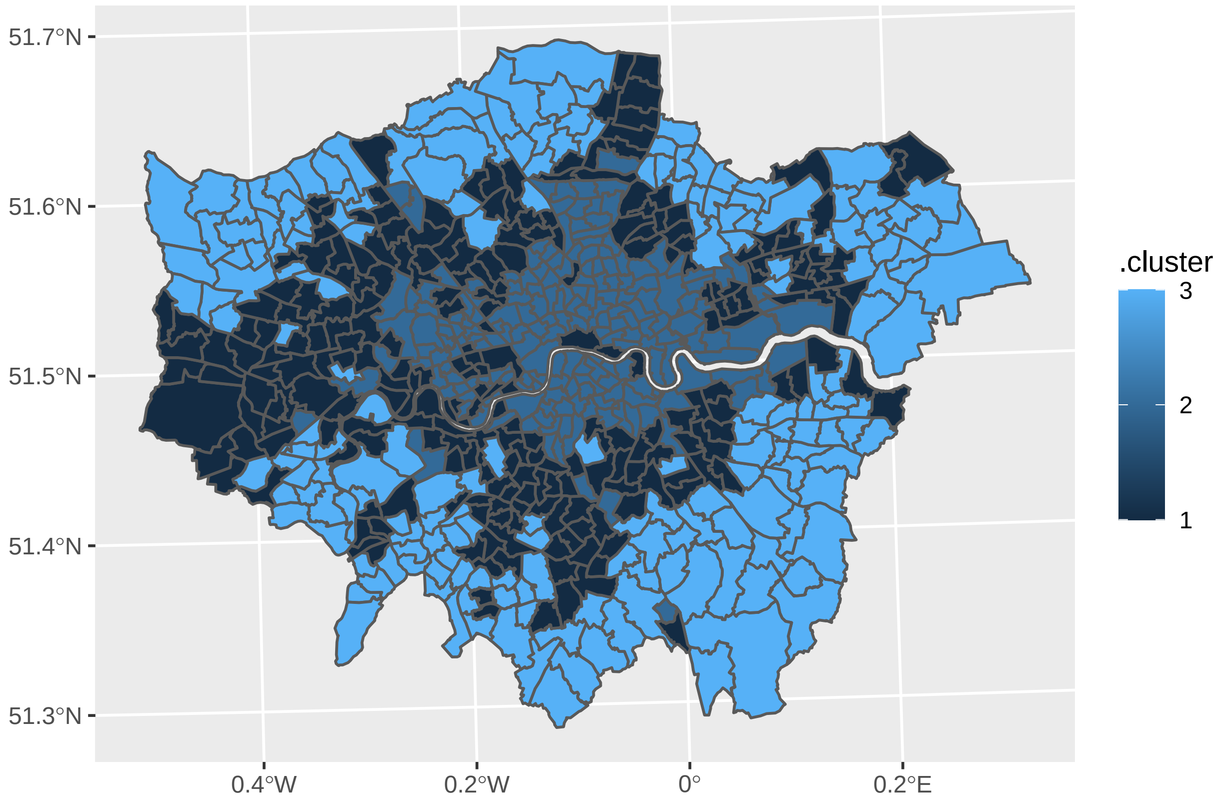
```
centroid <- tidy(fit)%>%
  #print the results of the cluster groupings
  print()%>%
  dplyr::select(PctOwned20, PctNoEngli)
```

```
## # A tibble: 3 x 5
##   PctOwned20 PctNoEngli  size withinss cluster
##        <dbl>      <dbl> <int>    <dbl> <fct>
## 1       48.9      14.7    247   21793. 1
## 2       27.5      17.0    187   13365. 2
## 3       72.4       6.38   191   17086. 3
```

```
# as we only have variable two dimensions we can plot the clusters on a graph
p <- ggplot(mydata,aes(PctOwned20, PctNoEngli))+
  geom_point(aes(colour=factor(fit$cluster)))+
  geom_point(data=centroid,aes(PctOwned20, PctNoEngli), size=7, shape=18)+ theme(legend.position="none"
p
```



```
LondonWards <- fit %>%
  #
  augment(., LondonWards)%>%
  dplyr::select(WD11CD, .cluster)%>%
  #make sure the .cluster column is numeric
  mutate(across(.cluster, as.numeric))%>%
  # join the .cluster to our sf layer
  left_join(LondonWards,
            .,
            by = c("WD11CD" = "WD11CD"))


#now map our geodeomographic classification
map <- ggplot(LondonWards) +
  geom_sf(mapping = aes(fill=.cluster))+
  scale_fill_continuous(breaks=c(1,2,3))
map
```


51.7°N
51.6°N
51.5°N
51.4°N
51.3°N
0.4°W
0.2°W
0°
0.2°E
.cluster
3
2
1