A Quick Introduction to Machine Learning Using R to Analyze & Visualize Geodemographic Clustering & Identify Neighbourhood-Level Socio-Economic Differences

John Cunningham
Epidemiologist

Objectives

- Demonstrate R programming language in a public health application
- Demonstrate using a machine learning algorithm to identify socio-demographic patterns in a defined population
- Demonstrate some basic mapping capabilities of R
- Increase interest in spatial data sciences and geo-analytics amongst public health epidemiologists/analysts/geographers

Introduction & Background

Introduction – What is R

- R is a free software environment for statistical computing and graphics
- Is a derivative of the S computing language and developed at U-Auckland
- Developed specifically for statistical analysis (calls Fortran, C, C++ subroutines)
- Is a Free Software that allows users to add functionality by defining new functions
- Everything you need to become proficient available online
- Learning curve is steep so be patient with yourself
- https://www.r-project.org/

Introduction - Geodemographics

- Neighbourhood scale analyses of people by where they live
- Classify neighbourhoods based on shared socio-economic and demographic characteristics
- Tobler's law of geography (proximity = similarity)
- Provide summary indicators of commonality in social structure that link locations geographically (birds of a feather…)
- Everyday use: marketing, retail, service planning
- K-means cluster algorithm common methodology

Introduction – Cluster Analysis

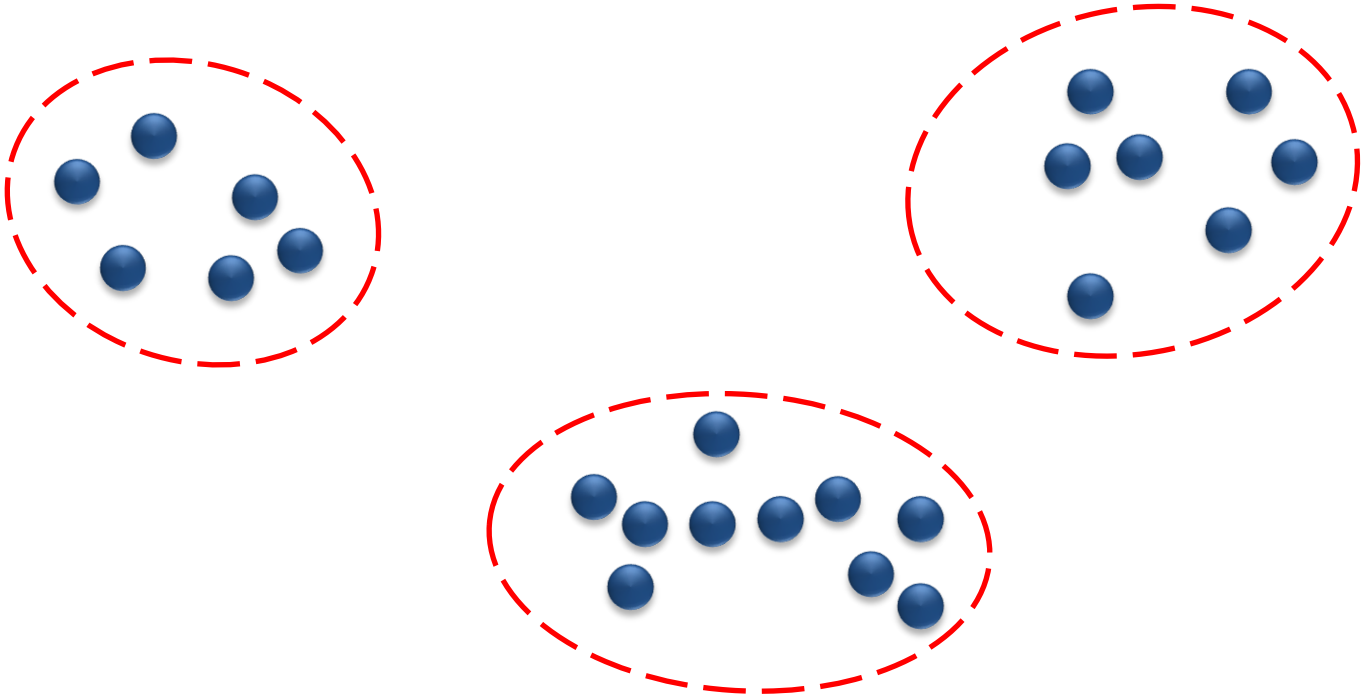
- Cluster analysis is a broad set of techniques for finding subgroups of observations within a data set (r x c matrix)
- K-means clustering is the most commonly used unsupervised machine learning algorithm for partitioning a given data set into a set of k groups
- K-means assigns observations by high intra-class similarity and low inter-class similarity (k groups or classes)
- Similarity is often hard to define. Different criteria will likely result in different cluster outcomes

Background – K-Means

- So how does it work?

Background – K-Means

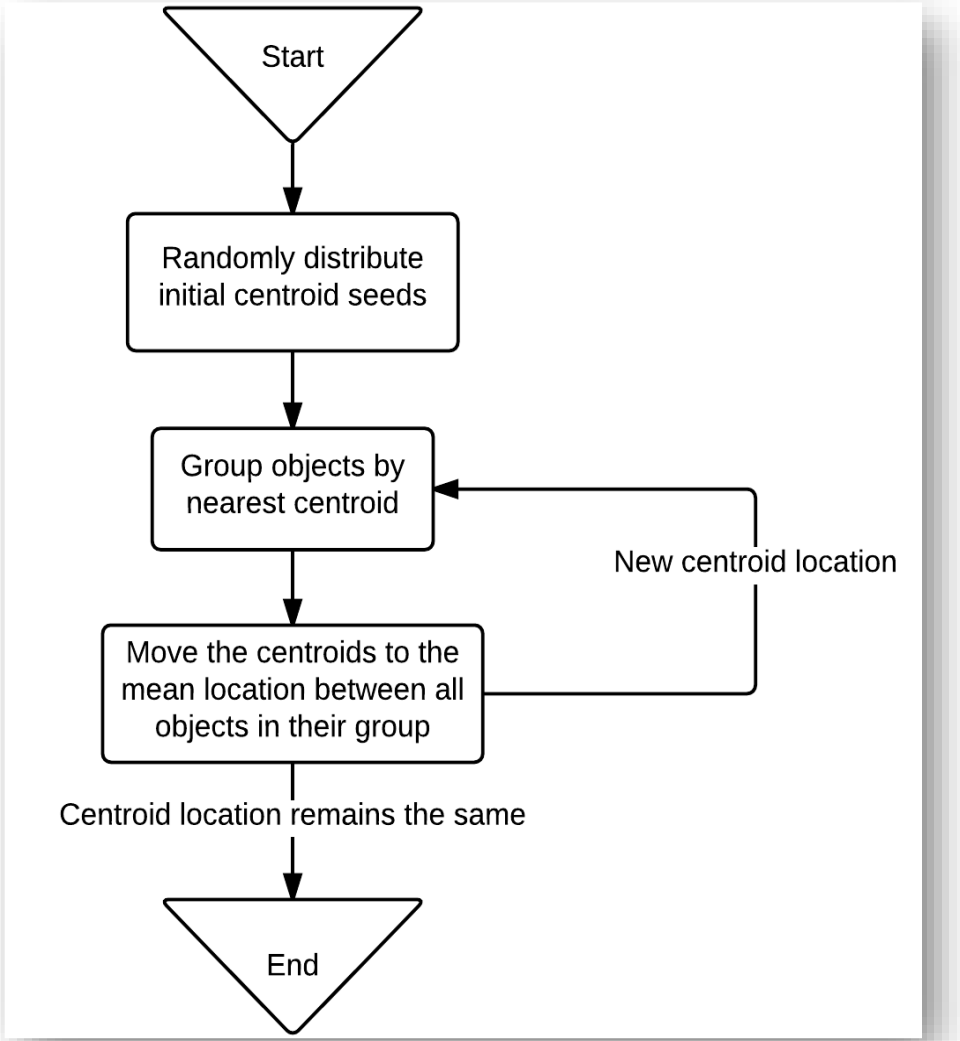
- The human brain can quickly discern visual patterns/clusters. Computers cannot



Background – K-Means

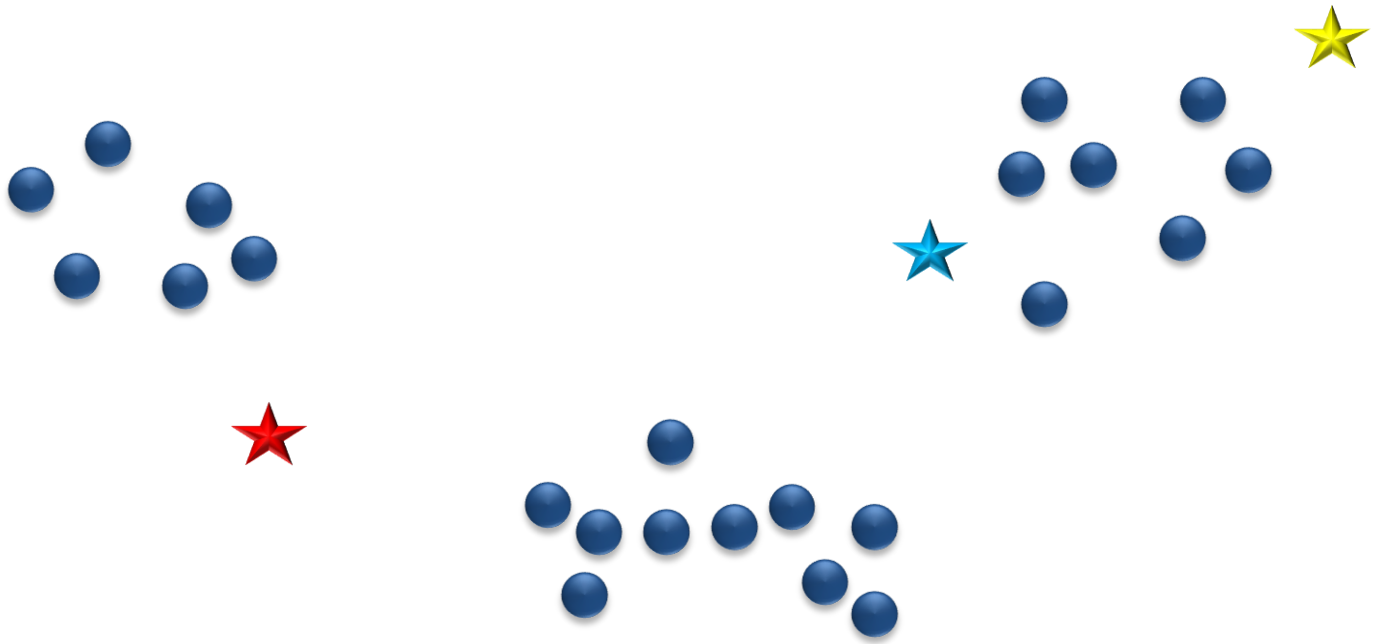
- The k-means algorithm process

$$S_i^t = \left\{ x_p : \left\| x_p - m_i^t \right\|^2 \le \left\| x_p - m_j^t \right\|^2 \ \forall j, 1 \le j \le k \right\}$$



Background – K-Means

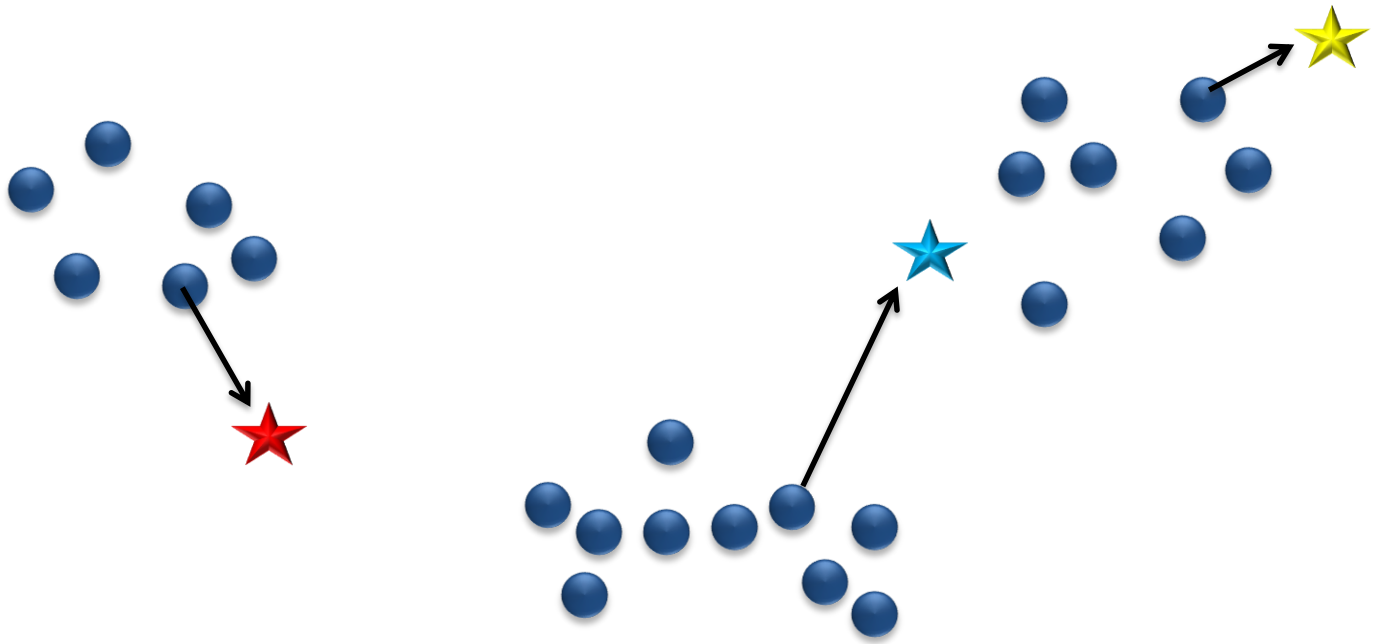
- Machine learning starts at random points and gains precision iteratively
- Instruct algorithm to start with 3 random centroid points in this example



Background – K-Means

- Apply 2 logic statements:

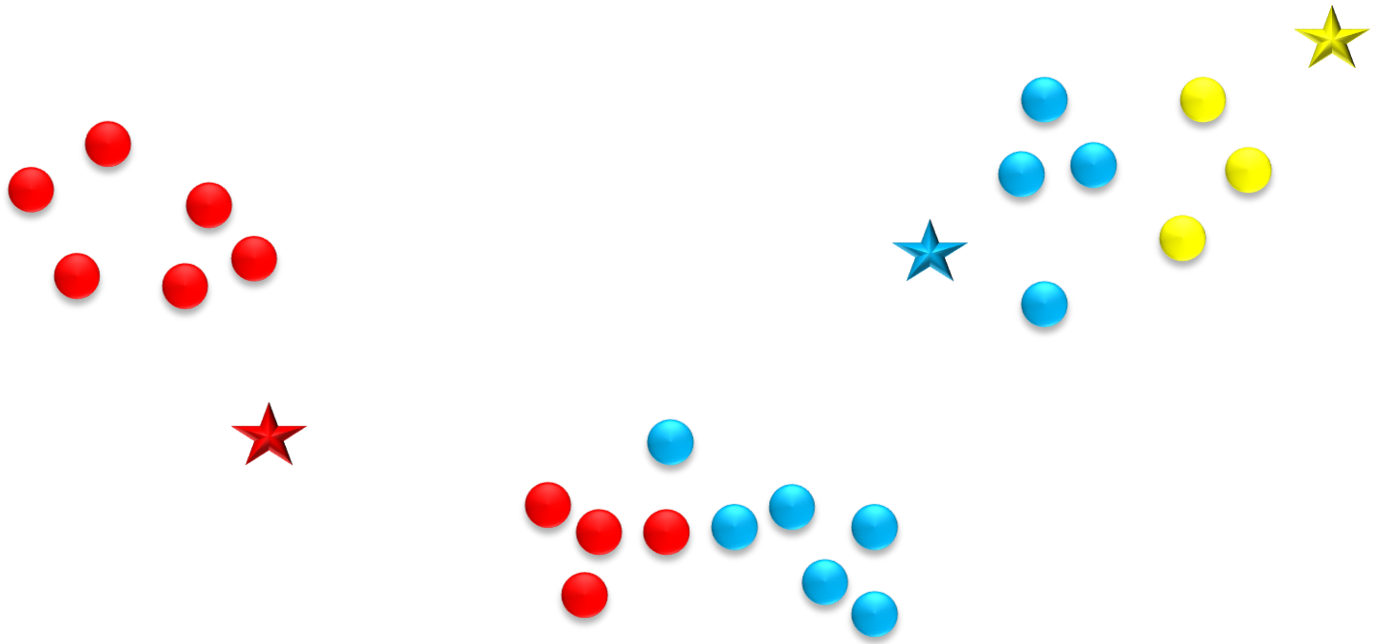
1. Each observation assigned to closest centroid by Euclidean distance



Background – K-Means

- Apply 2 logic statements:

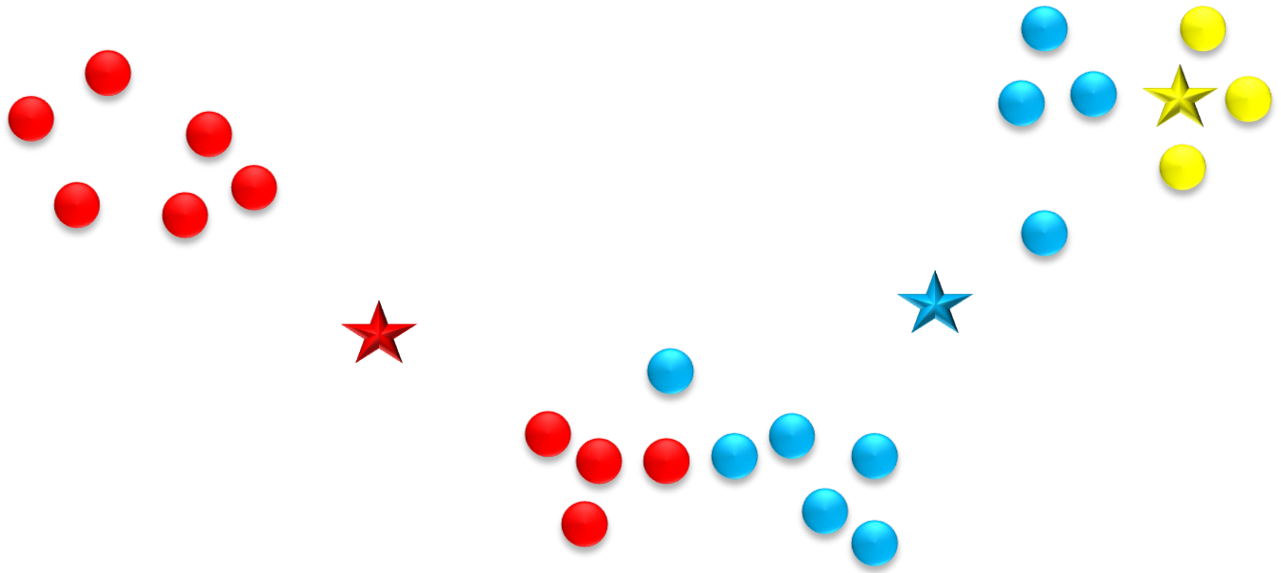
2. Each centroid moves to centre of observations assigned to it with each iteration



Background – K-Means

- Apply 2 logic statements:

2. Each centroid moves to centre of observations assigned to it with each iteration

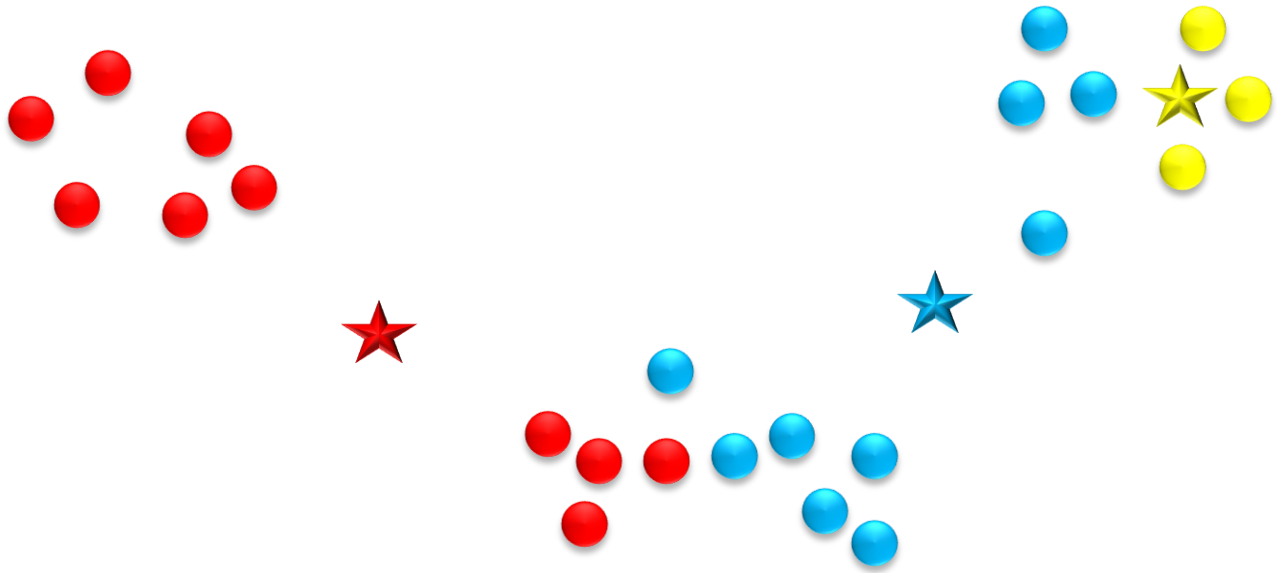


Background – K-Means

- Apply both logic statements again:

1. Each observation assigned to the closest centroid by Euclidean distance

2. Each centroid moves to centre of observations assigned to it with each iteration

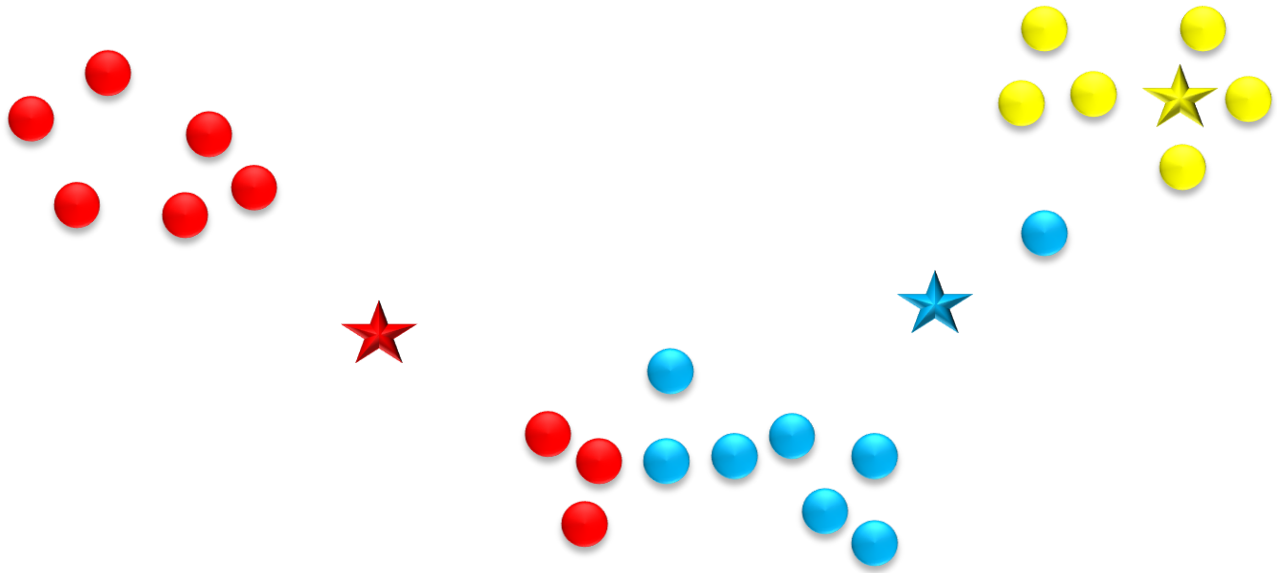


Background – K-Means

- Apply both logic statements again:

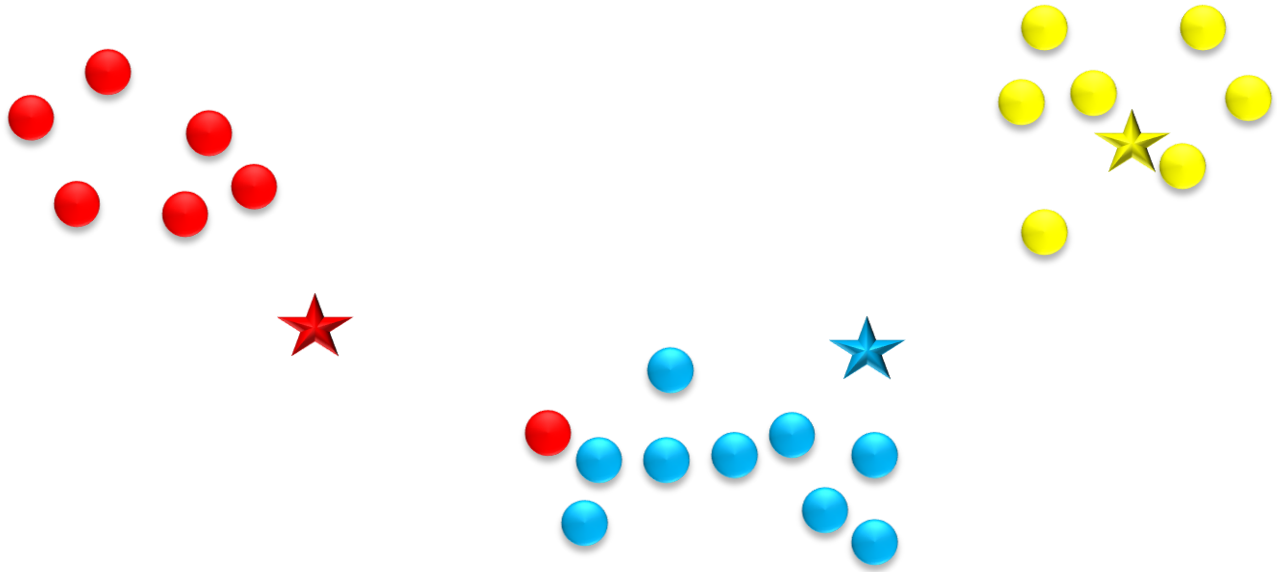
1. Each observation assigned to the closest centroid by Euclidean distance

2. Each centroid moves to centre of observations assigned to it with each iteration



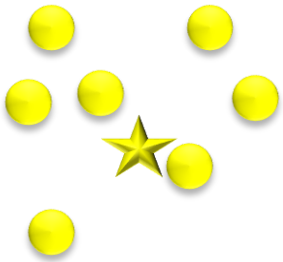
Background – K-Means

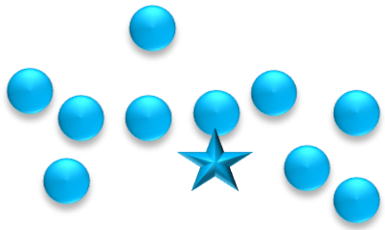
- …and again



Background – K-Means

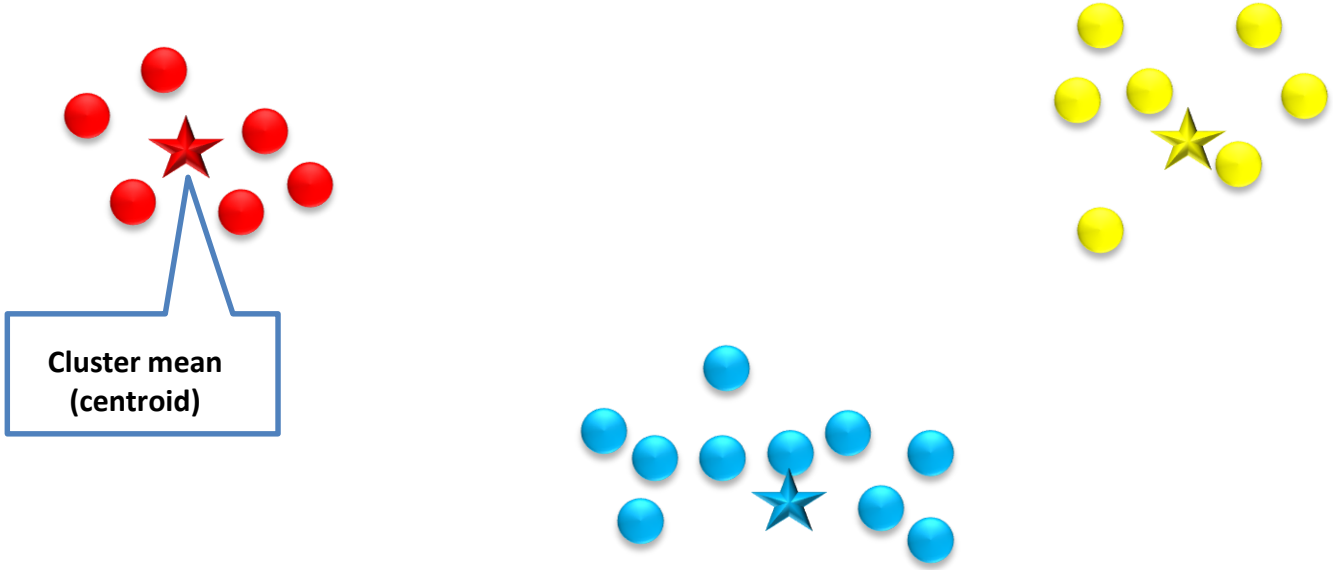
- …and again until convergence is attained





Background – K-Means

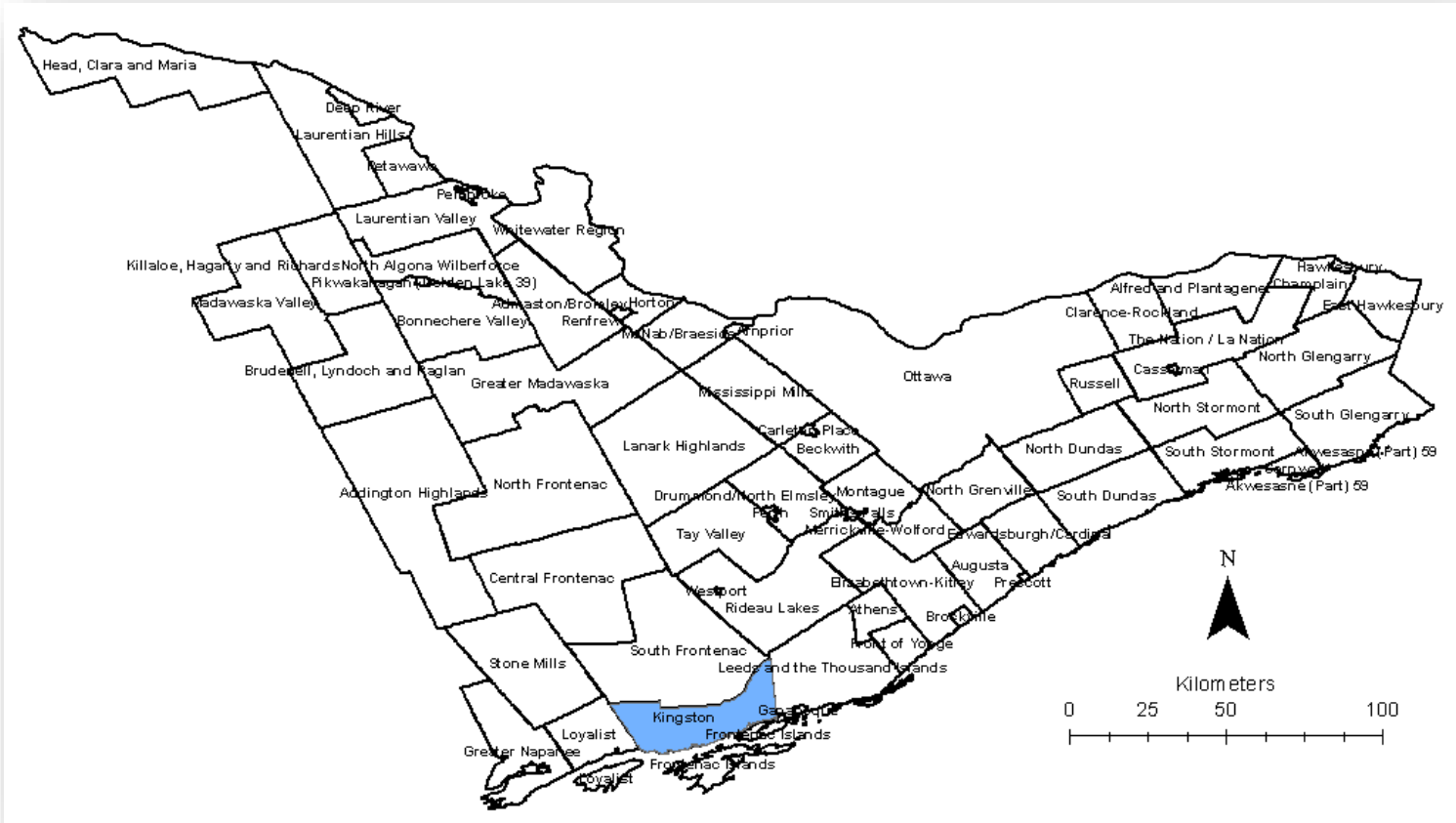
- The logic allowed the computer to come to the same conclusion as your eye
- Each cluster is represented by the mean centre (centroid) of the observations belonging to it
- The process maximizes the between cluster variance and minimizes the within cluster variance in n-dimensions (minimize sum of squared error distances)



Methods & Results

Methods – Study Site

- Chose City of Kingston (194 DA)



Methods/Results – Step 1

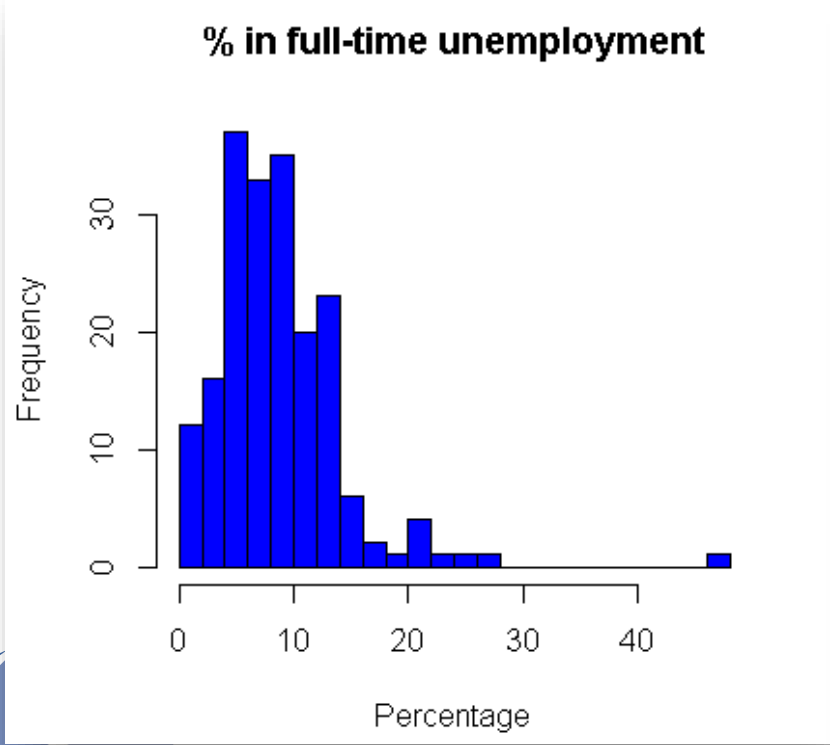
- 2016 Census of Canada data for Kingston (194 DA)
- Chose: Marital status, LIM-AT, Post-secondary education, Employment status
- Chose variables known to be associated with socio-economic differences
- Choose only continuous variables (means) and transform into percentages
- Best to examine the distribution of the variables before proceeding
- Best to clean and format the data in Excel prior to importing into R
- Best to only keep variables of interest

Methods/Results – Step 2

Examine the Distributions of the Variables:

- Always good practice to examine the data distributions before beginning

```
# creates histograms to view data distribution:
hist(kd$WorkUnemp, col="blue")
```




```
# creates boxplots to view data distribution:
boxplot(kd$WorkUnemp, col="blue")
```

Methods/Results – Step 2

Examine the Distributions of the Variables:

- Are there relationships between your chosen variables

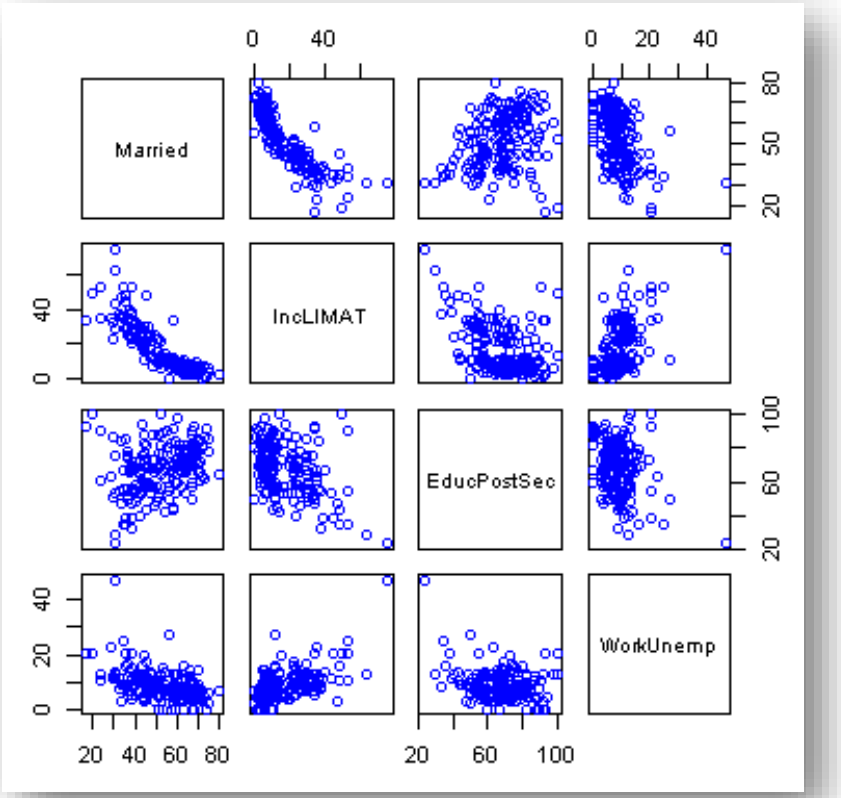
 # create correlation matrix to view potential relationships in the data:

```
data1 <- kd[ ,2:5]
cor (data1)
round (cor(data1), 2)
```

	Married	IncLIMAT	EducPostSec	WorkUnemp
Married	1.00	-0.86	0.31	-0.49
IncLIMAT	-0.86	1.00	-0.45	0.59
EducPostSec	0.31	-0.45	1.00	-0.31
WorkUnemp	-0.49	0.59	-0.31	1.00

 # visualize the correlation matrix:

```
Plot (data1, col="blue")
```



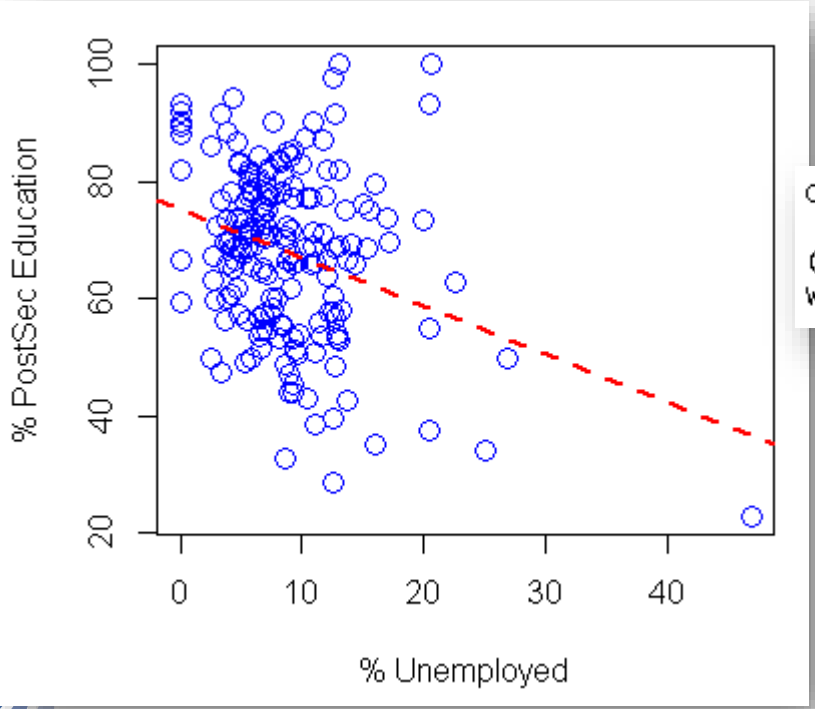
Methods/Results – Step 2

Examine the Distributions of the Variables:

- Are there relationships between your chosen variables

 # create simple linear regressions:

Model_1 <- lm(EducPostSec ~WorkUnemp, kd)




```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  75.4070     1.8923  39.850  < 2e-16 ***
WorkUnemp    -0.8286     0.1843  -4.496 1.22e-05 ***
```

Methods/Results – Step 3

Standardize the Data:

- Scaling the data allows for comparison between observations by giving them equal weighting

- Z-score (mean = 0, SD = 1) $\mathbf{Z = (x - \mu)/\sigma}$

- Remove or interpolate empty cells before scaling!!!

 # standardize data set:
Stand_data <- scale(kd)

Married (%)
40.8
80.8
55.0
73.0

Married (Z)
-1.89
0.87
-0.92
0.33

Methods/Results – Step 4

Assess Cluster Tendency:

- Assess data for non-random structuring before starting by comparing actual data to a randomized version of itself

- Why? Because cluster detection algorithms will create clusters of observations even when they don't exist

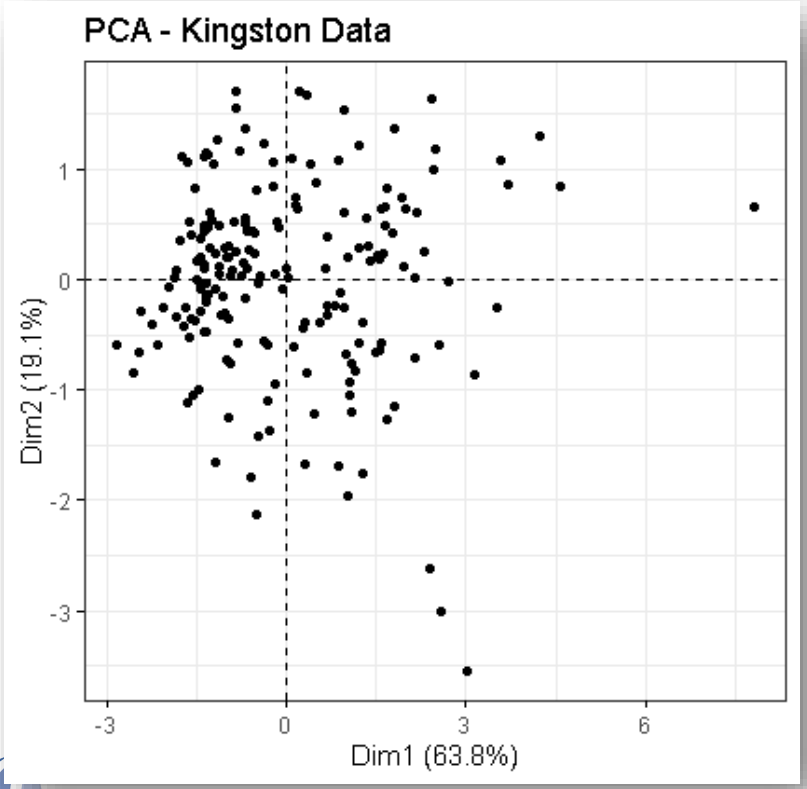
- Data contains more than 2 variables so reduce dimensionality to plot (PCA)

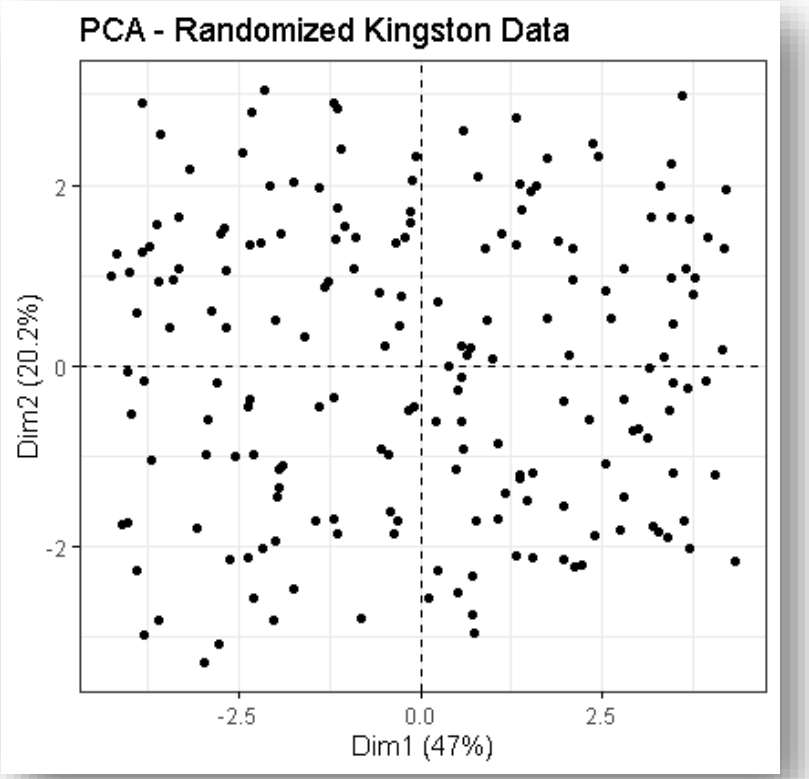
```
# plot original and randomized datasets:
fviz_pca_ind(prcomp(sk), title = "PCA - Kingston Data",
             geom = "point", ggtheme = theme_bw())

fviz_pca_ind(prcomp(random_sk), title = "PCA - Randomized Kingston Data",
             geom = "point", ggtheme = theme_bw())
```

Methods/Results – Step 4

Assess Cluster Tendency:

- Our Kingston data is showing definite structure

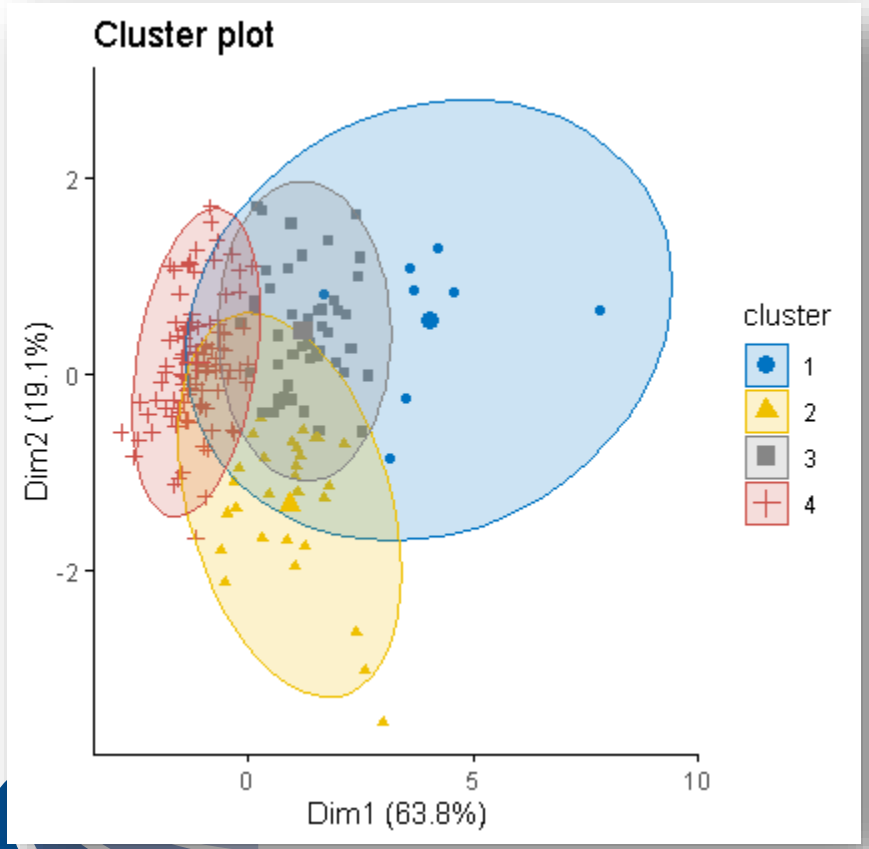


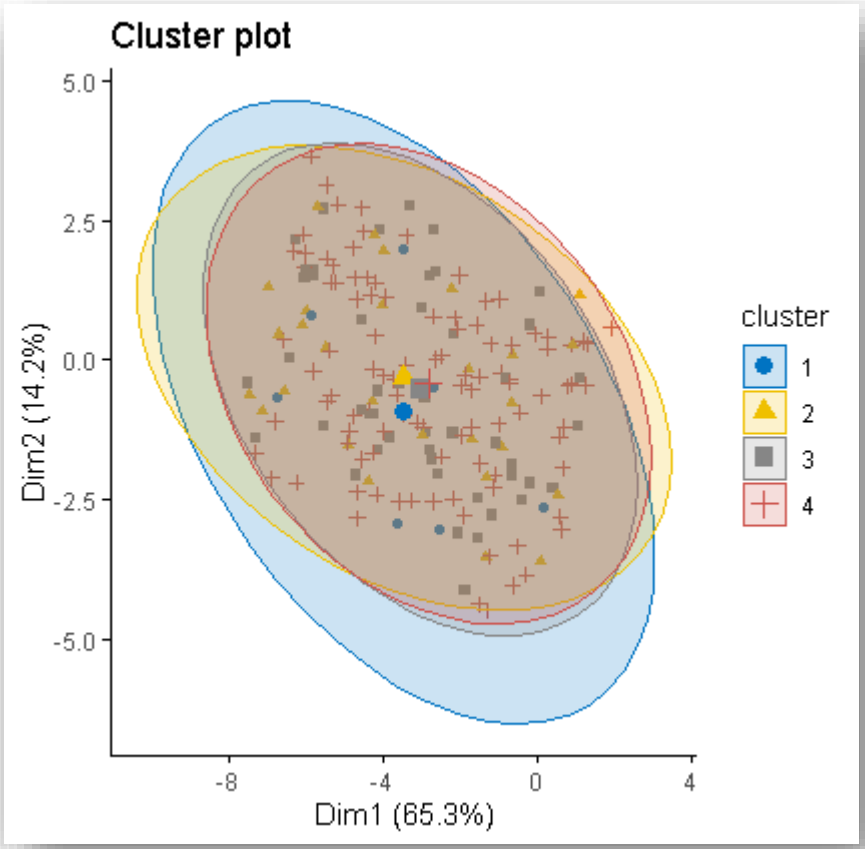


Methods/Results – Step 4

Assess Cluster Tendency:

- Adding ellipses helps to visualize the data better. Actual vs. randomized data. Some overlap likely due to co-linear nature of data





Methods/Results – Step 5

Compute K-means Clustering:

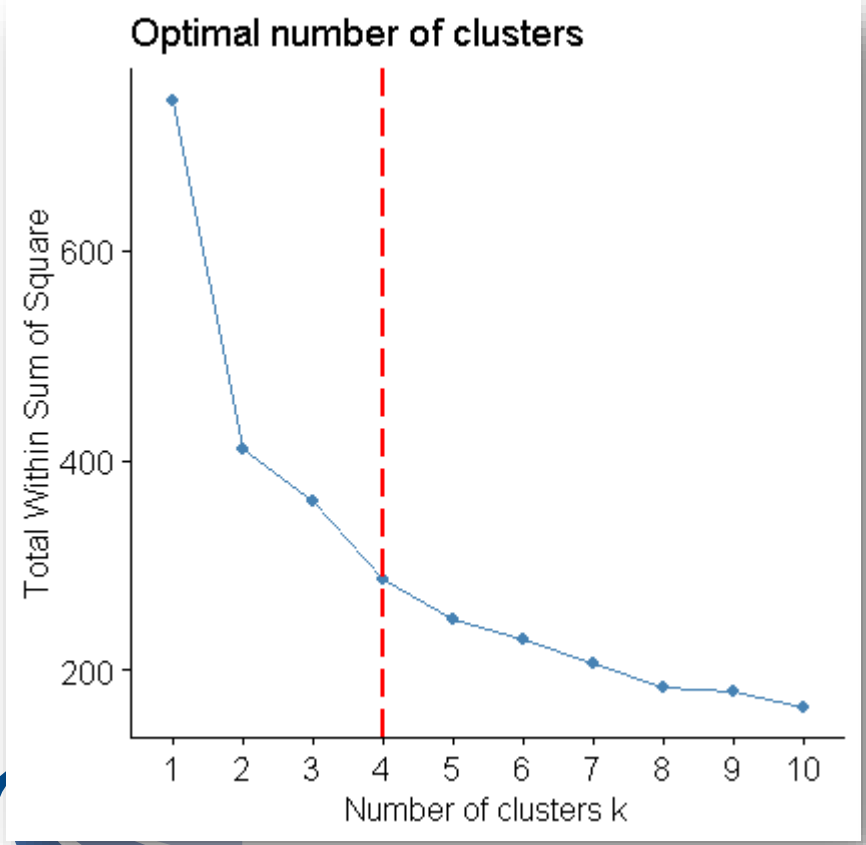
- Unsupervised machine learning process. Must specify number of clusters (k) and type of distance measure (Euclidean vs. Manhattan)
- Best to compute several versions of the algorithm (e.g. K = 2, 3,…n)
- Can use output to compare total within/between cluster sum of squares. Beware of overfitting model

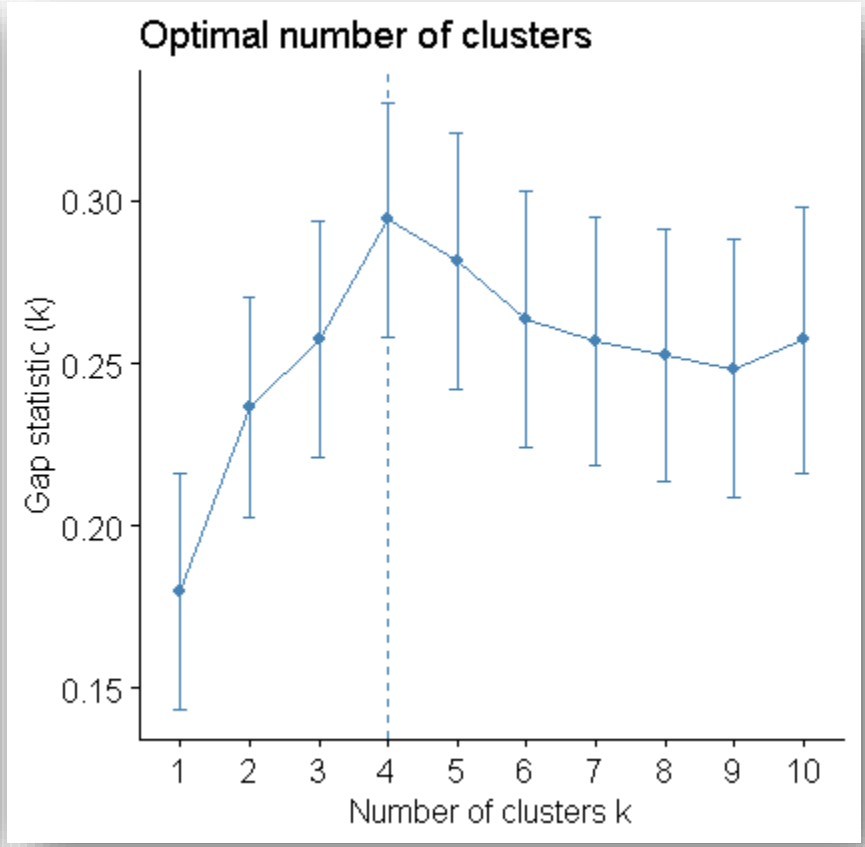
```
# runs k-means clustering
set.seed(123)
Km <- kmeans(kd, 5, start = 25, iter.max = 1000)
```

Methods/Results – Step 6

Determine Optimal Number of Clusters:

- After computing the k-means with different values for k make a determination of the ideal number of clusters by plotting the within cluster sum of squares (wss)





Methods/Results – Step 7

Final Model Results:

- Chose k=4. Values are means for each variable within each cluster centre

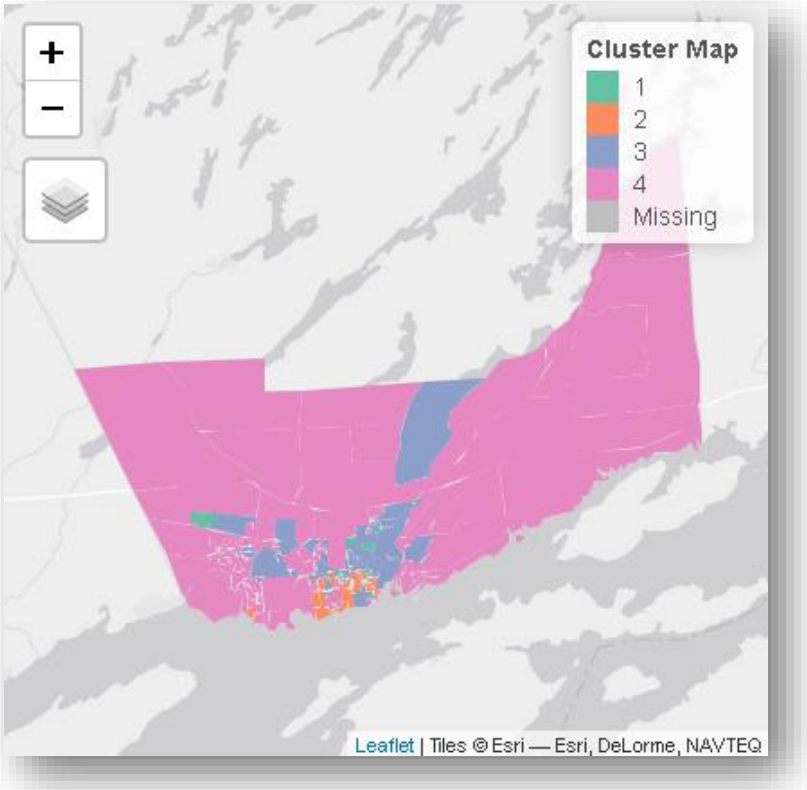
```
# print out cluster centre data
Km.res$centers
```

Cluster	Married (%)	Below LIM-AT (%)	Post-Sec (%)	Unemployed (%)
1	36.4	47.7	40.8	23.9
2	40.2	25.5	80.8	12.2
3	42.8	26.2	55.0	9.2
4	63.2	5.9	73.0	6.3

Methods/Results – Step 8

Map the Results:

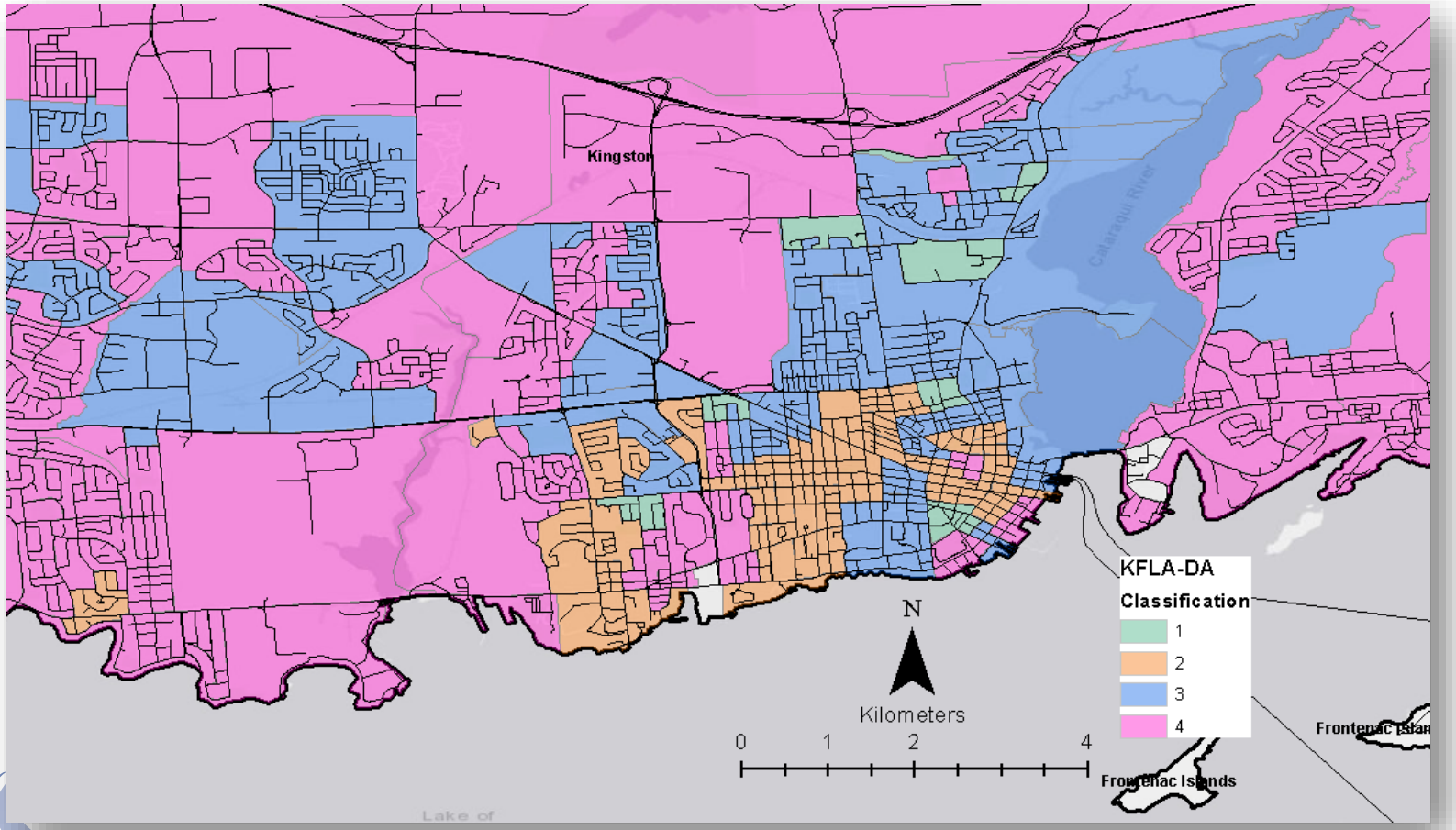
- Best way to understand date is to visualize it…well for me anyway
- Extract and join the cluster labels to the original KFLA DA file
- Map it (interactive mapping in R)
- Embed the .HTML to a website…



Methods/Results – Step 8

Map the Results:

- Recreated the map using ArcGIS

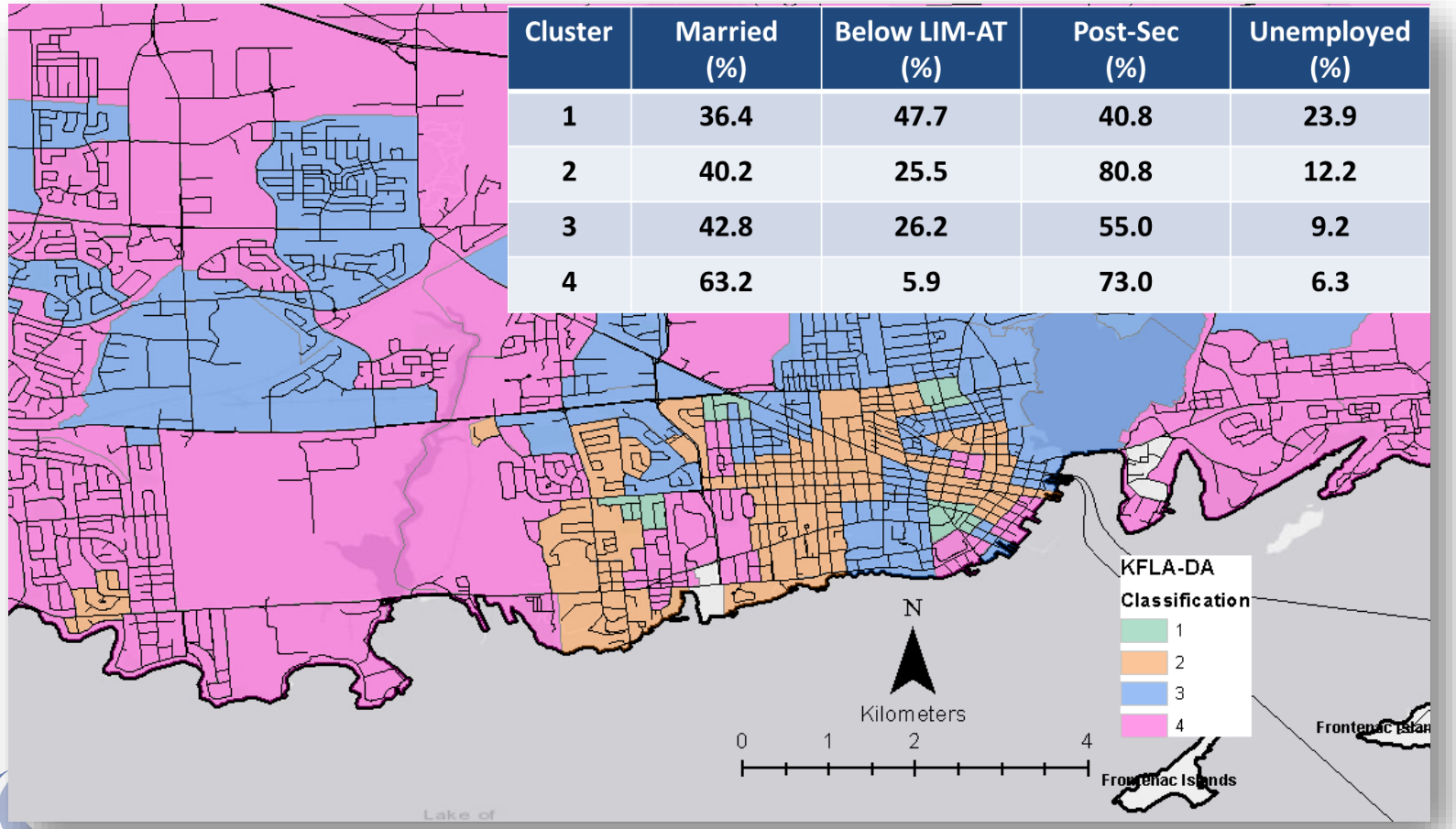


Methods/Results – Step 8

Map the Results:

- So lets look at the clusters spatially…

Cluster	Married (%)	Below LIM-AT (%)	Post-Sec (%)	Unemployed (%)
1	36.4	47.7	40.8	23.9
2	40.2	25.5	80.8	12.2
3	42.8	26.2	55.0	9.2
4	63.2	5.9	73.0	6.3



Discussion

Discussion/Observations

- Interesting that the variables clustered from a socio-economic perspective and a geographical one as well (Tobler's Law, autocorrelation)
- Careful evaluation of the data, process and output needed
- Although this example is geographic; can be non-geographic as well
- Is a behaviour based process (vulnerable to ecological issues)
- Algorithm creates groups; you have to give them meaning
- Try imputation or create dummy variables instead of deleting missing data obs.
- There are many ways of accomplishing this analysis

Discussion

Strengths:

- K-means is a simple and very fast algorithm that can deal with large amounts of data very efficiently in R
- Fairly easy to visualize and interpret the results
- R is incredibly fast at running algorithm

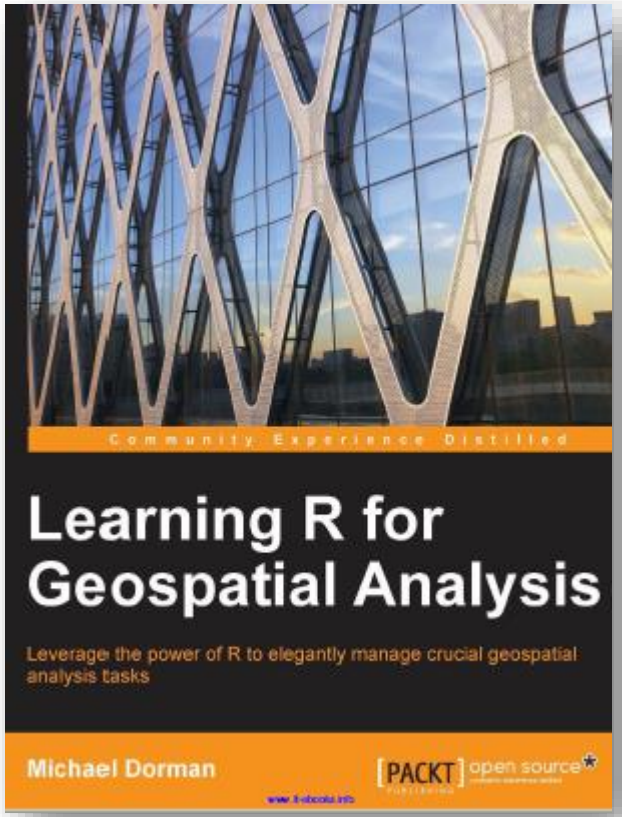
Limitations:

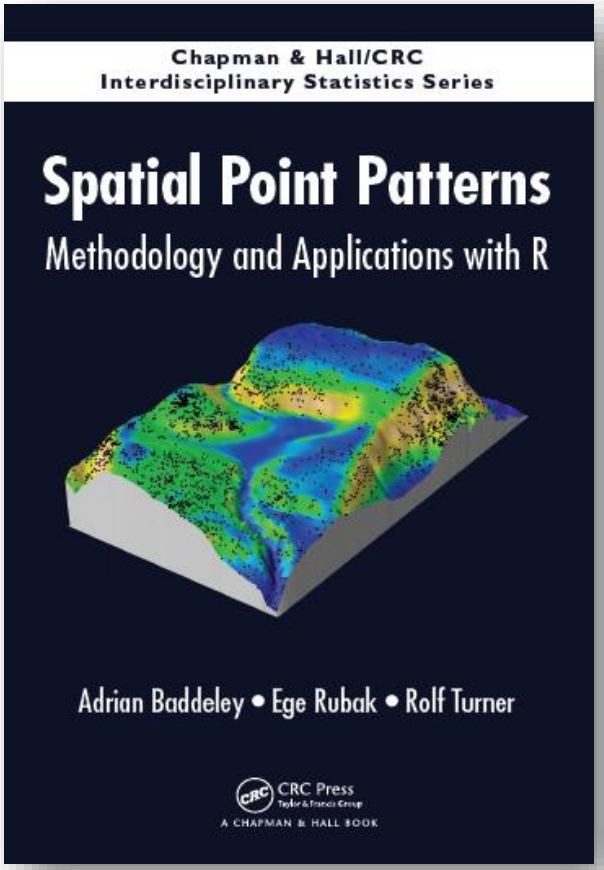
- Pre-specify number of clusters (Hierarchical Clustering a solution)
- Is susceptible to outliers being a means-based process (Need to examine data thoroughly before beginning. Alternate is a K-medians process)
- Slight differences in multiple runs of same data and model specification (random centre selection)

Appendix 1 – R Packages Used

Package	Description
cluster	Clustering algorithms
factoextra	Factor analysis, clustering algorithms & visualization
ggplot2	Refined visualization
leaflet	Interactive web-mapping
rgdal	Raster data processing
rgeos	Mapping interface engine
sp	Spatial analysis engine
tidyverse	Data manipulation
tmap	Interactive map creation

Appendix 2 – Cool Books



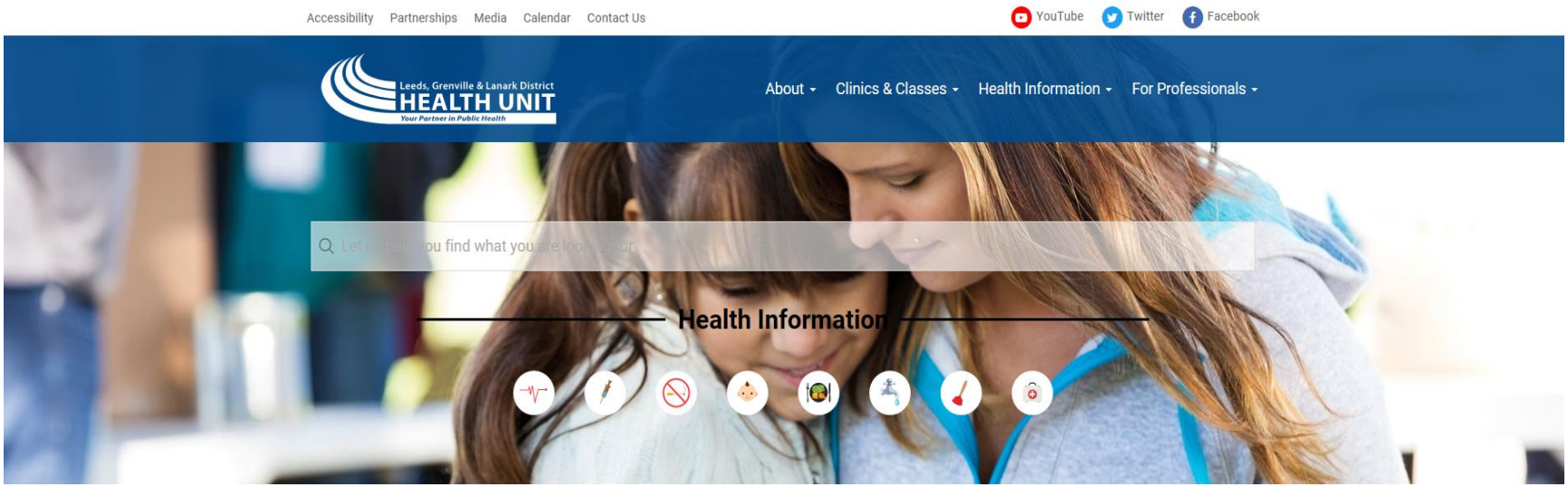


Questions?

john.cunningham@healthunit.org

epi@healthunit.org

Contact Us!



Visit our website:

www.healthunit.org

Email us at:

contact@healthunit.org

Call us at:

1-800-660-5853

FACEBOOK:
LGLHealthUnit

TWITTER:
@LGLHealthUnit