

Q1.R

mirrien

2022-04-10

```
# Question 1

# Helper function to calculate L1 distances

manhattan <- function(p1,p2) {
  distance <- matrix(NA, nrow=dim(p1)[1], ncol=dim(p2)[1])
  for(i in 1:nrow(p2)) {
    distance[,i] = rowSums(abs(t(t(p1)-p2[i,])))
  }
  return(distance)
}

# Another function for calculating L1 distances

manhattan2<-function(x,K){
  distance = matrix(NA, nrow= nrow(x), ncol = nrow(K))
  for(j in 1:nrow(K)) {
    for(i in 1:nrow(x)) {
      distance[i,j]<-dist(rbind(x[i,],K[j,]), method = "manhattan")
    }
  }
  return(distance)
}

kmedian <- function(x,K,itters) {
  # convert df to matrix
  x = as.matrix(x)

  # randomly sample some centers, set a seed 100
  set.seed(100)
  K <- x[sample(nrow(x), K),]

  # empty lists to store outputs
  assignments <- vector(itters, mode = "list")
  locations <- vector(itters, mode = "list")

  for(i in 1:itters) {
    # call manhattan distance helper function
    dists = manhattan(x,K)
    # find minimum distance
```

```

clusters <- apply(dists,1,which.min)
# tapply median()
centers <- apply(x,2,tapply,clusters,median)

# store outputs
assignments[[i]] <- clusters
locations[[i]] <- centers
}

# return outputs in list
return(list(locations=locations[[1]], assignments = assignments[[1]]))
}

# Test:

# df = read.csv("parkinsons.data",row.names = 1)
# kmedian(df,3,10)

```