

```

#consider na value for each round, update centroid at the end
Partial_km<-function(m,k,initCentroids,nIters){
  iter <- 0
  initC<-matrix(initCentroids,byrow = TRUE,nrow=k)
  #creating temporary centroid
  temp_Centroid <- matrix(0, nrow = k, ncol = ncol(m))
  #creating distance matrix
  d <- matrix(0, nrow = nrow(m), ncol = k)

  while(iter<=as.integer(nIters)){
    iter <- iter + 1
    for (j in 1:k)#each centroid
      {
        d[,j] = apply(m,1,function(x){sum(x[which(!is.na(x))]-initC[j,][which(!is.na(x))])^2})
      }
    ##assign cluster number to each observation
    v<-apply(d,1,which.min)
    ##calculate the new centroid
    for (i in 1:k){
      m_i<-m[which(v==i),]
      temp_Centroid[i,] <- apply(m_i,2,function(x){sum(x[which(!is.na(x))])/sum(!is.na(x))})
      #use initC to make up na values
      temp_Centroid[i,][which(is.na(temp_Centroid[i,]))]<-initC[i,][which(is.na(temp_Centroid[i,]))]
    }
    #update the current centroid
    if(all(temp_Centroid == initC)){
      break
    }
    else{
      initC <- temp_Centroid
    }
  }
  return(v)
}

test_accuracy<-function(actual_y,fit_y){
  acc_rate<-sum(fit_y==actual_y)/length(actual_y)
  return(acc_rate)
}

```

## test with house.votes

```

library(dplyr)

##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##   filter, lag
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

```

```

house.votes <- read.csv("~/Desktop/research/research/house-votes-84.data", header=FALSE, na.strings="?")
encode<-function(x){
  factor(x,levels = c('n', 'y'),labels = c(1, 2))>%as.numeric()
}
test_na<-apply(house.votes[2:17],2,encode)
initCentroids=c(1,2,1,2,2,2,1,1,1,1,1,2,2,2,1,2,
                2,2,2,1,1,1,2,2,2,2,1,1,2,1,2,2)
initC<-matrix(initCentroids,byrow = TRUE,nrow=2)
fitted_y<-Partial_km(m=test_na,k=2,initCentroids,nIters=100)
actual_y<-factor(house.votes[,1],levels = c( 'democrat','republican'),labels = c(1, 2))>%as.numeric()
test_accuracy(actual_y,fitted_y)

```

```
## [1] 0.5402299
```

## test with penguins

```

library(palmerpenguins)
penguins[,c(2,7:8)]<-apply(penguins[,c(2,7:8)],2,function(x){as.numeric(as.factor(x))})
fitted_y<-Partial_km(m=penguins[,2:8],k=3,initCentroids=cbind(penguins[5,-1],penguins[150,-1],penguins[
actual_y<-penguins$species>%as.numeric()
test_accuracy(actual_y,fitted_y)

```

```
## [1] 0.5901163
```