# Prediction of sale prices of house

## Problem Statement

To predict the sale prices of houses

## Project Team

1. Tejaswini Nutalapati
2. Aditi Bhargava

## About the Data

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence. With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this dataset allows us to predict the final price of each home.

The Ames Housing dataset was compiled by Dean De Cock for use in data science education. It's an incredible alternative for data scientists looking for a modernized and expanded version of the often-cited Boston Housing dataset.

### Data Souce: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

## Clustering

```
#Loading the libraries

library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
library(VIM)
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## Loading required package: data.table
```

```
## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/alexko
wa/VIM/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:Hmisc':
##
##     src, summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(plyr)
```

```
## ------------------------------------------------------------------------
----
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, th
en dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------
----

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:Hmisc':
##
##     is.discrete, summarize

library(ggplot2)
library(RColorBrewer)
library(lubridate)

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
# Loading the Dataset
list.files("../input")

## character(0)

train<-read.csv("C:/Users/aditi/OneDrive/Desktop/MVA/train.csv")
```

```
# Missing Imputation and Missing Indicators Creating
#Before everything, we should deal with missing values.
sort(sapply(train, function(x) sum(is.na(x))), decreasing = TRUE)

##        PoolQC   MiscFeature         Alley         Fence   FireplaceQu
##          1453          1406          1369          1179           690
##   LotFrontage    GarageType   GarageYrBlt  GarageFinish    GarageQual
##           259            81            81            81            81
##    GarageCond  BsmtExposure  BsmtFinType2      BsmtQual      BsmtCond
##            81            38            38            37            37
```

```
##  BsmtFinType1    MasVnrType    MasVnrArea    Electrical            Id
##            37             8             8             1             0
##    MSSubClass      MSZoning       LotArea        Street      LotShape
##             0             0             0             0             0
##   LandContour     Utilities     LotConfig     LandSlope  Neighborhood
##             0             0             0             0             0
##    Condition1    Condition2      BldgType    HouseStyle   OverallQual
##             0             0             0             0             0
##   OverallCond     YearBuilt  YearRemodAdd     RoofStyle      RoofMatl
##             0             0             0             0             0
##   Exterior1st   Exterior2nd     ExterQual     ExterCond    Foundation
##             0             0             0             0             0
##    BsmtFinSF1    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating
##             0             0             0             0             0
##     HeatingQC    CentralAir     X1stFlrSF     X2ndFlrSF  LowQualFinSF
##             0             0             0             0             0
##     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath      HalfBath
##             0             0             0             0             0
##  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd    Functional
##             0             0             0             0             0
##    Fireplaces    GarageCars    GarageArea    PavedDrive    WoodDeckSF
##             0             0             0             0             0
##   OpenPorchSF EnclosedPorch    X3SsnPorch   ScreenPorch      PoolArea
##             0             0             0             0             0
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##             0             0             0             0             0
##     SalePrice
##             0
```

```
#From the table, we found some variables that has amount of missing values: A
lley(1369), PoolQC(1453), Fence(1179)
#Before we dump those variables, we need to creat missing indicator first. Fo
r example, the missing indicators of the PoolQC will tell us if the house has
a pool or not.
train <- train %>% mutate(PoolQC_imp = if_else(is.na(PoolQC), 1, 0),
                          MiscFeature_imp = if_else(is.na(MiscFeature), 1, 0),
                          Alley_imp = if_else(is.na(Alley), 1, 0),
                          Fence_imp = if_else(is.na(Fence), 1, 0))
drop1 <- names(train) %in% c("PoolQC", "MiscFeature", "Alley", "Fence", "Id")
train <- train[!drop1]

#Using kNN imputation for missing imputation.
train <- kNN(train, k = 5)

#Now we need to drop some unnessary missing indicators.
missing_ind <- train[81:160]
drop2 <- c("PoolQC_imp_imp", "MiscFeature_imp_imp", "Alley_imp_imp", "Fence_i
mp_imp")
```

```r
for (i in 1:80)
{
  if (sum(missing_ind[i] == "TRUE") < 20)
  {
    drop2 <- append(drop2, names(missing_ind)[i])
  }
  else
  {
    next
  }
}
drop2a <- names(train) %in% drop2
train1 <- train[!drop2a]

#As clustering can only recognise the numerical vectors, we have to transfer
all the factors to integers.
train1$MSZoning = as.integer(train1$MSZoning)
train1$Street = as.integer(train1$Street)
train1$LotShape = as.integer(train1$LotShape)
train1$LandContour = as.integer(train1$LandContour)
train1$Utilities = as.integer(train1$Utilities)
train1$LotConfig = as.integer(train1$LotConfig)
train1$LandSlope = as.integer(train1$LandSlope)
train1$Neighborhood = as.integer(train1$Neighborhood)
train1$Condition1 = as.integer(train1$Condition1)
train1$Condition2 = as.integer(train1$Condition2)
train1$BldgType = as.integer(train1$BldgType)
train1$HouseStyle = as.integer(train1$HouseStyle)
train1$RoofStyle = as.integer(train1$RoofStyle)
train1$RoofMatl = as.integer(train1$RoofMatl)
train1$Exterior1st = as.integer(train1$Exterior1st)
train1$Exterior2nd = as.integer(train1$Exterior2nd)
train1$MasVnrType = as.integer(train1$MasVnrType)
train1$ExterQual = as.integer(train1$ExterQual)
train1$ExterCond = as.integer(train1$ExterCond)
train1$Foundation = as.integer(train1$Foundation)
train1$BsmtQual = as.integer(train1$BsmtQual)
train1$BsmtCond = as.integer(train1$BsmtCond)
train1$BsmtExposure = as.integer(train1$BsmtExposure)
train1$BsmtFinType1 = as.integer(train1$BsmtFinType1)
train1$BsmtFinType2 = as.integer(train1$BsmtFinType2)
train1$Heating = as.integer(train1$Heating)
train1$HeatingQC = as.integer(train1$HeatingQC)
train1$CentralAir = as.integer(train1$CentralAir)
train1$Electrical = as.integer(train1$Electrical)
train1$KitchenQual = as.integer(train1$KitchenQual)
train1$Functional = as.integer(train1$Functional)
train1$FireplaceQu = as.integer(train1$FireplaceQu)
train1$GarageType = as.integer(train1$GarageType)
train1$GarageFinish = as.integer(train1$GarageFinish)
```

```
train1$GarageQual = as.integer(train1$GarageQual)
train1$GarageQual = as.integer(train1$GarageQual)
train1$GarageCond = as.integer(train1$GarageCond)
train1$PavedDrive = as.integer(train1$PavedDrive)
train1$SaleType = as.integer(train1$SaleType)
train1$SaleCondition = as.integer(train1$SaleCondition)
str(train1)
```

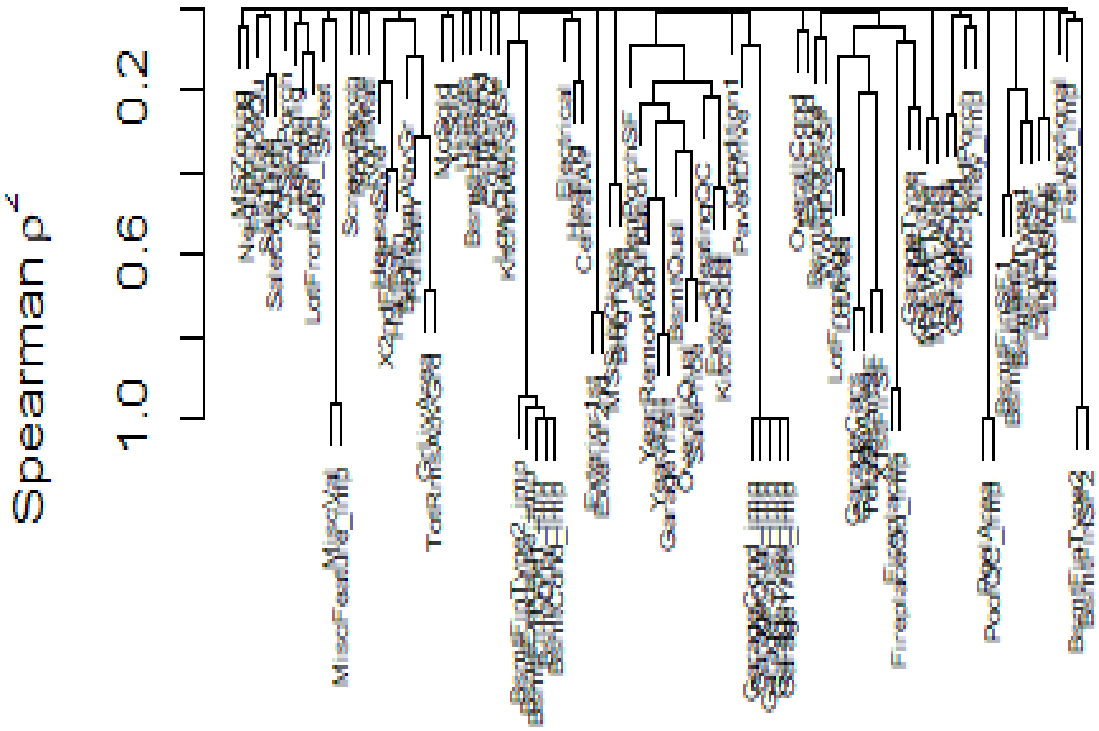
```
## 'data.frame':    1460 obs. of  92 variables:
##  $ MSSubClass      : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning        : int  4 4 4 4 4 4 4 4 5 4 ...
##  $ LotFrontage     : int  65 80 68 60 84 85 75 90 51 50 ...
##  $ LotArea         : int  8450 9600 11250 9550 14260 14115 10084 10382 612
0 7420 ...
##  $ Street          : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ LotShape        : int  4 4 1 1 1 1 4 1 4 4 ...
##  $ LandContour     : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Utilities       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ LotConfig       : int  5 3 5 1 3 5 5 1 5 1 ...
##  $ LandSlope       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood    : int  6 25 6 7 14 12 21 17 18 4 ...
##  $ Condition1      : int  3 2 3 3 3 3 3 5 1 1 ...
##  $ Condition2      : int  3 3 3 3 3 3 3 3 3 1 ...
##  $ BldgType        : int  1 1 1 1 1 1 1 1 1 2 ...
##  $ HouseStyle      : int  6 3 6 6 6 1 3 6 1 2 ...
##  $ OverallQual     : int  7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond     : int  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt       : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 193
9 ...
##  $ YearRemodAdd    : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 195
0 ...
##  $ RoofStyle       : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ RoofMatl        : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Exterior1st     : int  13 9 13 14 13 13 13 7 4 9 ...
##  $ Exterior2nd     : int  14 9 14 16 14 14 14 7 16 9 ...
##  $ MasVnrType      : int  2 3 2 3 2 3 4 4 3 3 ...
##  $ MasVnrArea      : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual       : int  3 4 3 4 3 4 3 4 4 4 ...
##  $ ExterCond       : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Foundation      : int  3 2 3 1 3 6 3 2 1 1 ...
##  $ BsmtQual        : int  3 3 3 4 3 3 1 3 4 4 ...
##  $ BsmtCond        : int  4 4 4 2 4 4 4 4 4 4 ...
##  $ BsmtExposure    : int  4 2 3 4 1 4 1 3 4 4 ...
##  $ BsmtFinType1    : int  3 1 3 1 3 3 3 1 6 3 ...
##  $ BsmtFinSF1      : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ BsmtFinType2    : int  6 6 6 6 6 6 6 2 6 6 ...
##  $ BsmtFinSF2      : int  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF       : int  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF     : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ Heating         : int  2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ HeatingQC       : int  1 1 1 3 1 1 1 1 3 1 ...
##  $ CentralAir      : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical      : int  5 5 5 5 5 5 5 5 2 5 ...
##  $ X1stFlrSF       : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ..
.
##  $ X2ndFlrSF       : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea       : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 107
7 ...
##  $ BsmtFullBath    : int  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath    : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath        : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath        : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr    : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr    : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual     : int  3 4 3 3 3 4 3 4 4 4 ...
##  $ TotRmsAbvGrd    : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional      : int  7 7 7 7 7 7 7 7 3 7 ...
##  $ Fireplaces      : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu     : int  5 5 5 3 5 5 3 5 5 5 ...
##  $ GarageType      : int  2 2 2 6 2 2 2 2 6 2 ...
##  $ GarageYrBlt     : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 193
9 ...
##  $ GarageFinish    : int  2 2 2 3 2 3 2 2 3 2 ...
##  $ GarageCars      : int  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea      : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual      : int  5 5 5 5 5 5 5 5 2 3 ...
##  $ GarageCond      : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ PavedDrive      : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ WoodDeckSF      : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF     : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch   : int  0 0 0 272 0 0 0 228 205 0 ...
##  $ X3SsnPorch      : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MiscVal         : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold          : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold          : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 200
8 ...
##  $ SaleType        : int  9 9 9 9 9 9 9 9 9 9 ...
##  $ SaleCondition   : int  5 5 5 1 5 5 5 5 1 5 ...
##  $ SalePrice       : int  208500 181500 223500 140000 250000 143000 307000
200000 129900 118000 ...
##  $ PoolQC_imp      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ MiscFeature_imp : num  1 1 1 1 1 0 1 0 1 1 ...
##  $ Alley_imp       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Fence_imp       : num  1 1 1 1 1 0 1 1 1 1 ...
##  $ LotFrontage_imp : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ BsmtQual_imp    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ BsmtCond_imp    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

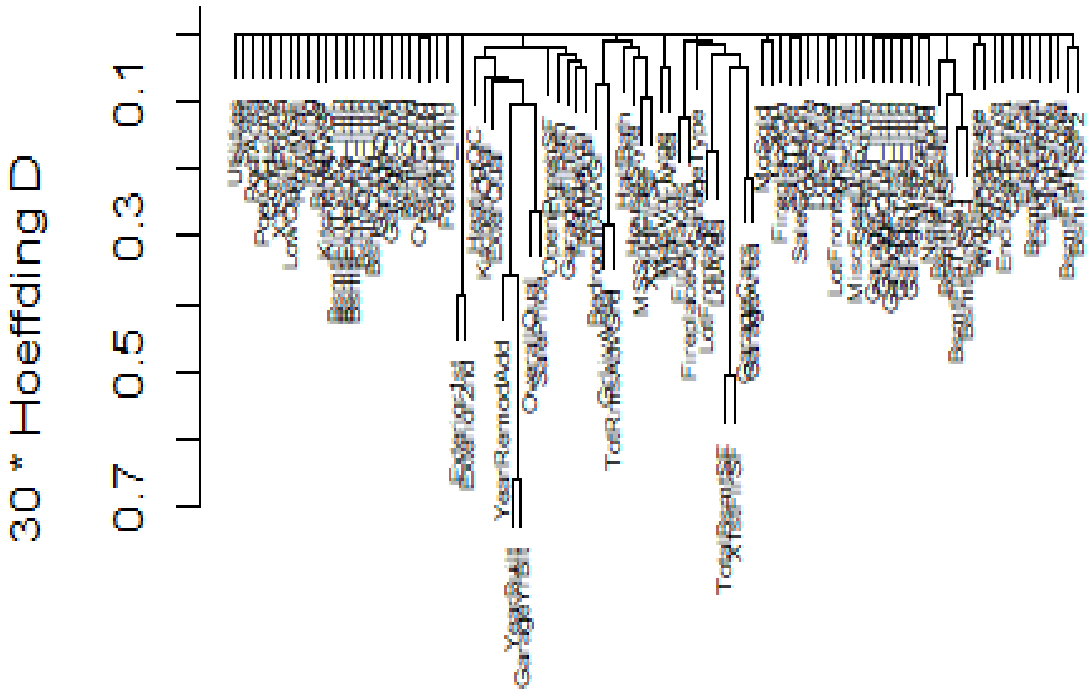
```
##  $ BsmtExposure_imp: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ BsmtFinType1_imp: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ BsmtFinType2_imp: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ FireplaceQu_imp : logi  TRUE FALSE FALSE FALSE FALSE TRUE ...
##  $ GarageType_imp  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ GarageYrBlt_imp : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ GarageFinish_imp: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ GarageQual_imp  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ GarageCond_imp  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

Clustering

```
# Variable Clustering


clus_var <- varclus(as.matrix(train1), similarity = "spearman", minlev = 0.05
)
plot(clus_var, cex = 0.5)
```




```
clus_var2 <- varclus(as.matrix(train1), similarity = "hoeffding", minlev = 0.
05)
plot(clus_var2, cex = 0.5)
```




#Non-hirerarchical Method
#K-Means

```
# Centers (k's) are numbers thus, 10 random sets are chosen
(kmeans2.dataset <- kmeans(train1,2,nstart = 10))
```

```
## K-means clustering with 2 clusters of sizes 335, 1125
##
## Cluster means:
##   MSSubClass MSZoning LotFrontage   LotArea   Street LotShape LandContour
## 1   49.46269 3.868657    82.49254 14940.206 1.997015 2.429851    3.776119
## 2   59.11111 4.076444    66.00978  9199.644 1.995556 3.095111    3.777778
##   Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgTyp
e
## 1  1.000000  3.817910  1.104478     14.66567   3.116418   3.008955 1.28656
7
## 2  1.000889  4.079111  1.049778     12.69867   3.006222   3.008000 1.55466
7
##   HouseStyle OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle Roof
Matl
## 1   4.408955    7.725373    5.405970  1992.854     1999.851  2.680597 2.20
0000
## 2   3.928000    5.615111    5.625778  1964.840     1980.404  2.329778 2.03
8222
##   Exterior1st Exterior2nd MasVnrType MasVnrArea ExterQual ExterCond Founda
tion
```

```
## 1    11.06567    11.86269   2.805970  226.62985  2.853731  4.850746   2.81
7910
## 2    10.49333    11.18400   2.750222   67.18311  3.744000  4.698667   2.27
1111
##   BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2
## 1 2.501493 3.841791     2.811940     3.710448   675.5642     5.785075
## 2 3.510222 3.809778     3.420444     3.741333   374.5778     5.691556
##   BsmtFinSF2 BsmtUnfSF TotalBsmtSF  Heating HeatingQC CentralAir Electrica
l
## 1   40.50149  731.9284    1447.994 2.014925  1.465672   1.994030   4.98806
0
## 2   48.35022  518.2000     941.128 2.042667  2.857778   1.917333   4.59111
1
##   X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath Ful
lBath
## 1  1508.725  549.3821     2.877612  2060.985    0.5940299   0.03283582 1.9
94030
## 2  1059.566  286.7253     6.728000  1353.020    0.3751111   0.06488889 1.4
37333
##    HalfBath BedroomAbvGr KitchenAbvGr KitchenQual TotRmsAbvGrd Functional
## 1 0.5432836     3.035821     1.000000    2.558209     7.746269   6.943284
## 2 0.3351111     2.816000     1.060444    3.572444     6.152000   6.691556
##   Fireplaces FireplaceQu GarageType GarageYrBlt GarageFinish GarageCars
## 1  1.0447761    3.543284   2.465672    1994.391     1.611940   2.453731
## 2  0.4844444    3.765333   3.632000    1972.109     2.405333   1.562667
##   GarageArea GarageQual GarageCond PavedDrive WoodDeckSF OpenPorchSF
## 1   676.7284   4.919403   4.988060   2.988060  148.23881    82.00597
## 2   412.3084   4.847111   4.878222   2.816889   78.16622    36.13511
##   EnclosedPorch X3SsnPorch ScreenPorch PoolArea  MiscVal   MoSold   YrSold
## 1      12.86269   5.235821    22.78209 6.937313 26.41791 6.734328 2007.740
## 2      24.66133   2.865778    12.76178 1.514667 48.57244 6.199111 2007.838
##   SaleType SaleCondition SalePrice PoolQC_imp MiscFeature_imp Alley_imp
## 1 8.405970      5.092537  294385.5  0.9880597       0.9820896 0.9910448
## 2 8.540444      4.674667  147134.1  0.9973333       0.9573333 0.9217778
##   Fence_imp LotFrontage_imp BsmtQual_imp BsmtCond_imp BsmtExposure_imp
## 1 0.9223881       0.1701493   0.00000000   0.00000000       0.00000000
## 2 0.7733333       0.1795556   0.03288889   0.03288889       0.03377778
##   BsmtFinType1_imp BsmtFinType2_imp FireplaceQu_imp GarageType_imp
## 1       0.00000000      0.002985075       0.1194030          0.000
## 2       0.03288889      0.032888889       0.5777778          0.072
##   GarageYrBlt_imp GarageFinish_imp GarageQual_imp GarageCond_imp
## 1           0.000            0.000          0.000          0.000
## 2           0.072            0.072          0.072          0.072
##
## Clustering vector:
##    [1] 2 2 1 2 1 2 1 2 2 2 2 1 2 1 2 2 2 2 2 2 1 2 1 2 2 1 2 1 2 2 2 2 2 2
1 1 2
##   [38] 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 1 2 1 2 1 1
2 2 2
##   [75] 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
```

```
2 2 2
##  [112] 2 1 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1 1 2 1 2 2 2
2 2 1
##  [149] 2 2 2 1 2 1 2 2 2 1 1 1 2 1 2 2 2 2 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2 2
2 2 2
##  [186] 1 2 2 2 1 1 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 1
2 2 2
##  [223] 2 2 1 2 1 2 2 2 2 1 2 2 2 2 2 2 1 2 1 2 2 2 2 1 2 2 2 1 2 1 2 2 2 1
2 2 1
##  [260] 2 2 1 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 1 2 1 2 2 1 2 2 2 2 2 2 1 2 2
1 2 2
##  [297] 2 1 2 2 2 1 2 2 1 1 1 2 2 1 2 2 2 1 2 2 1 1 1 2 1 1 1 2 1 2 1 2 2 2
2 2 1
##  [334] 2 2 1 1 2 2 2 2 2 2 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
2 2 2
##  [371] 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 1 2 2 1
2 2 2
##  [408] 2 1 1 2 2 1 2 1 2 2 1 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1
2 2 2
##  [445] 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 1 1 2 2 1
1 2 1
##  [482] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2
1 2 1
##  [519] 2 1 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 1 2 2 2 1 2 2
1 2 1
##  [556] 2 2 2 2 2 1 2 2 2 2 1 2 1 2 1 2 2 2 1 2 2 2 2 2 2 2 2 1 2 1 2 1 2 2 2
2 2 1
##  [593] 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 1 1 2 1 2 1 2 2 2 2 2 1 1 2 1 2 2 2 2
2 2 2
##  [630] 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2
2 1 1
##  [667] 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 1 1 2 1 2 1 2 2 2 1 1 2 2 2 2 2 2 2
1 2 1
##  [704] 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 1 2 1 2 2 2 1 2 1 2 2 2 2
1 2 2
##  [741] 2 2 2 2 2 2 1 1 1 1 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 1 1 1 2 2 2 1 2 2 2
1 2 1
##  [778] 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 1 2 2 2 2 2 1 2 2 2 2 1 2 1 2 1 2 2 2
2 2 2
##  [815] 2 1 2 1 2 1 2 2 1 2 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2
##  [852] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 1 2 2 2 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2
1 2 2
##  [889] 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2
2 2 2
##  [926] 2 1 2 1 1 2 2 1 2 1 2 2 2 1 1 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
2 2 1
##  [963] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 1 2 2 1 2 2 2 1 2
2 2 2
## [1000] 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 1
```

```
1 2 2
## [1037] 1 1 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2 1 2 2 2 2
2 2 2
## [1074] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
1 2 1
## [1111] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2
2 2 2
## [1148] 2 2 2 2 2 2 1 2 2 2 2 1 1 2 2 1 2 2 2 1 1 2 1 1 2 2 2 2 2 1 1 2 2 2 2 1
1 1 2
## [1185] 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 1
2 2 2
## [1222] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 1 1 2 2 2 2 2 2 1 2 2 1 2
2 1 2
## [1259] 2 2 2 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 1 1 2 2
2 2 2
## [1296] 2 2 2 2 2 2 1 2 1 1 2 1 2 2 2 2 1 2 1 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1
2 1 2
## [1333] 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 2 2 2
2 2 2
## [1370] 1 2 2 1 1 1 1 2 2 2 2 2 1 2 2 2 2 1 2 1 2 1 2 2 2 1 1 2 2 2 2 2 2 2
1 2 1
## [1407] 2 2 2 2 2 1 2 2 1 2 2 2 1 2 1 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2
2 2 1
## [1444] 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 1 2 2
##
## Within cluster sum of squares by cluster:
## [1] 2.188516e+12 1.561143e+12
##  (between_SS / total_SS =  59.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withi
nss"
## [6] "betweenss"    "size"         "iter"         "ifault"



# Computing the percentage of variation accounted for. Two clusters
perc.var.2 <-
round(100*(1 - kmeans2.dataset$betweenss/kmeans2.dataset$totss),1)
names(perc.var.2) <- "Perc. 2 clus"
perc.var.2

## Perc. 2 clus
##         40.1

Distance <- dist(train1, method="euclidean")
Distance

##                 1            2            3            4            5
## 2     27052.12165
```

```
## 3     15263.95650  42052.21010
## 4     68512.87375  41523.83236  83518.73883
## 5     41913.16883  68676.12705  26679.09427 110105.25189
## 6     65751.04925  38783.64658  80558.01387   5587.95633 107010.54190
## 7     98527.06744 125504.04160  83524.94080 167011.45148  57175.24855
## 8      8749.76143  18570.44429  23529.71978  60015.35661  50154.81191
## 9     78642.82744  51741.14930  93743.89049  10681.14118 120380.91942
## 10    90513.62692  63540.29203 105577.64340  22145.76720 132189.00328
## 11    79056.46073  52027.21112  94008.97980  10718.14168 120551.44819
## 12   136547.20765 163524.59912 121505.49819 205017.84760  95030.09446
## 13    64669.73369  37657.65946  79529.84489   5435.99963 106023.81912
## 14    71062.43583  98020.15246  56029.58199 139513.85932  29775.16084
## 15    51573.65792  24542.43474  66511.79144  17100.88486  93072.72495
## 16    76551.65174  49640.99747  91654.96482   8793.12129 118296.75270
## 17    59581.10256  32556.63862  74512.73474   9261.62480 101061.33088
## 18   118534.28455  91524.02893 133510.87703  50036.55237 160051.49997
## 19    49790.81394  22876.71657  64556.67546  19482.10941  91015.57555
## 20    69515.95336  42554.41290  84588.09312   2485.52590 111212.76395
## 21   116952.73781 143890.70694 101850.65482 185363.08992  75306.27683
## 22    69122.50458  42174.94045  84196.50190   2439.52167 110825.48303
## 23    21669.78039  48542.94470   6991.30017  90023.05283  20608.12250
## 24    78721.81818  51881.53602  93871.53061  11492.46327 120529.99727
## 25    54520.91350  27556.10990  69580.81600  14133.42814  96207.19915
## 26    48196.15044  74967.24767  32988.09050 116408.34178   6576.89532
## 27    73723.77829  46774.63674  88803.56673   5858.72051 115431.82085
## 28    97560.31259 124517.84424  82515.80263 166020.75723  56090.11509
## 29     8074.05338  26863.24586  16858.31427  67861.43743  42581.55848
## 30   140027.46149 113059.81293 155088.26860  71590.26870 181689.00519
## 31   168503.55716 141512.99117 183523.02763 100008.04277 210084.74122
## 32    59175.58753  32198.11723  74214.60342   9501.56508 100829.63640
## 33    28769.35055   2571.88705  43625.36332  39950.03463  70197.18795
## 34    43073.90199  16044.03169  58020.91725  25565.04463  84594.27838
## 35    69024.28957  96029.36340  54162.92811 137527.99319  28402.84017
## 36   100634.57046 127576.58473  85536.00865 169048.74243  59014.79527
## 37    63567.92683  36545.78790  78514.67509   5329.41845 105070.84433
## 38    55515.99884  28530.75094  70567.84107  13149.79099  97182.85490
## 39    99508.45885  72521.68152 114555.13974  31069.38969 141152.91057
## 40   126533.74656  99578.53413 141606.05422  58126.13491 168216.00130
## 41    48513.59116  21527.48501  63562.52955  20052.36916  90186.23232
## 42    39437.41465  13629.53763  53813.60166  30920.78398  80059.30551
## 43    64516.14042  37510.56882  79539.10802   4260.89955 106138.13661
## 44    78265.64455  51262.33058  93283.07453   9848.65448 119872.21673
## 45    67516.40606  40547.12677  82576.71949   2263.10738 109197.20885
## 46   111419.88117 138421.68069  96482.37898 179918.62090  70231.09115
## 47    31515.67669  58288.57458  16316.74419  99752.82278  10473.42800
## 48    41342.95819  68241.61367  26268.79110 109725.61658   3727.35389
## 49    95590.46638  68711.86144 110713.01455  27488.48335 137359.07448
## 50    81511.65314  54535.62558  96572.34244  13191.97821 123185.57328
## 51    31972.87899   6344.85327  46577.64368  37254.59176  73011.44360
## 52    94038.27283  67098.94162 109123.64715  25741.20603 135749.78576
```

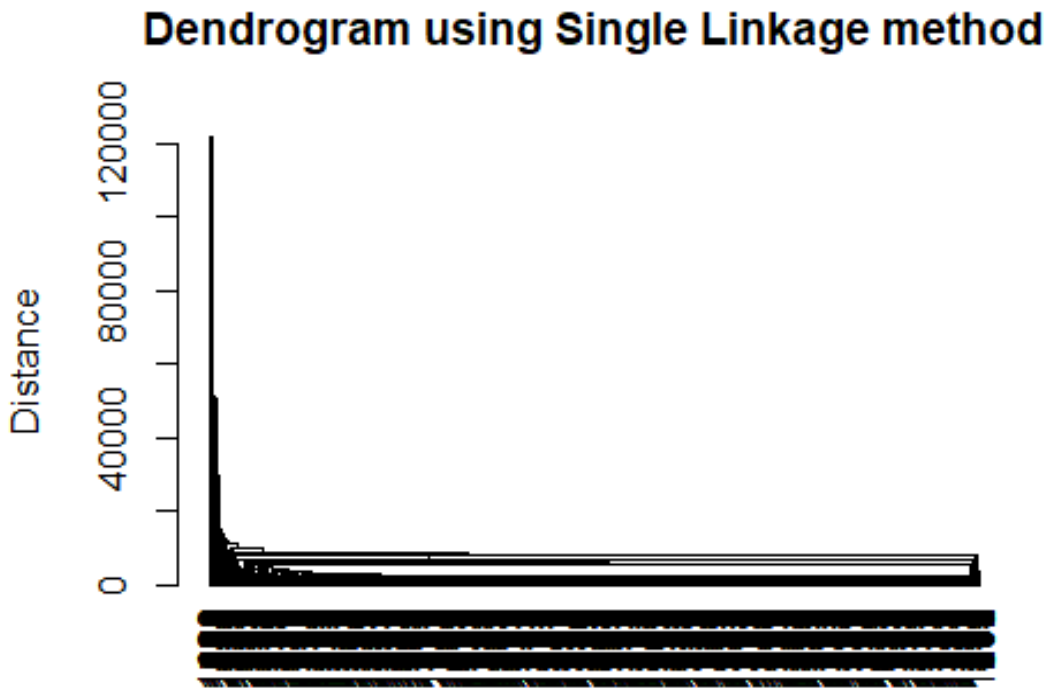
```
## 53    98512.62040  71522.84494 113545.47963  30058.11207 140135.76293
## 54   181400.29219 207529.88155 166162.88019 248373.55497 139735.87331
## 55    78520.58749  51571.92963  93599.34128  10355.48087 120223.81945
## 56    28094.12827   1554.09105  43034.68876  40527.99629  69637.02125
## 57    36467.40069  11432.85201  51722.68565  33234.79516  78369.04560
## 58    12459.94266  15226.69042  27011.38029  56541.23486  53575.32570
## 59   230351.22266 257328.94737 215303.60953 298816.44190 188788.06378
## 60    83619.71862  56658.43392  98693.48098  15347.52117 125314.50740
## 61    50724.05970  23758.17211  65537.09409  18390.46927  92021.67588
## 62   107513.92209  80551.09176 122572.00344  39081.08038 149177.49431
## 63     6582.10954  21286.76739  21599.17892  62593.85160  48172.22165
## 64    68530.97904  41537.08509  83508.04990    848.62713 110077.70135
## 65    11071.12795  38028.78244   4495.18342  79508.45502  30893.92447
## 66   108518.06053 135519.43949  93523.63084 177005.06318  67170.02533
## 67    30794.12092  10547.98355  44400.07436  41375.04839  70250.59971
## 68    17688.58912  44515.22422   2887.18669  86019.83531  24310.59185
## 69   128567.90829 101633.09953 143662.02989  60219.02275 170287.24003
##                6            7            8            9           10
## 2
## 3
## 4
## 5
## 6
## 7    164059.28077
## 8     57133.85412 107011.17008
## 9     15420.22325 177155.61983  70241.64896
.

.
## 67
## 68
## 69
##  [ reached getOption("max.print") -- omitted 1391 rows ]
```

```
#Hirerarchical Methods

#1. Single Linkage Method
# Invoking hclust command (cluster analysis by single linkage method)
clus_sales_prediction.nn <- hclust(Distance, method = "single")
clus_sales_prediction.nn
```

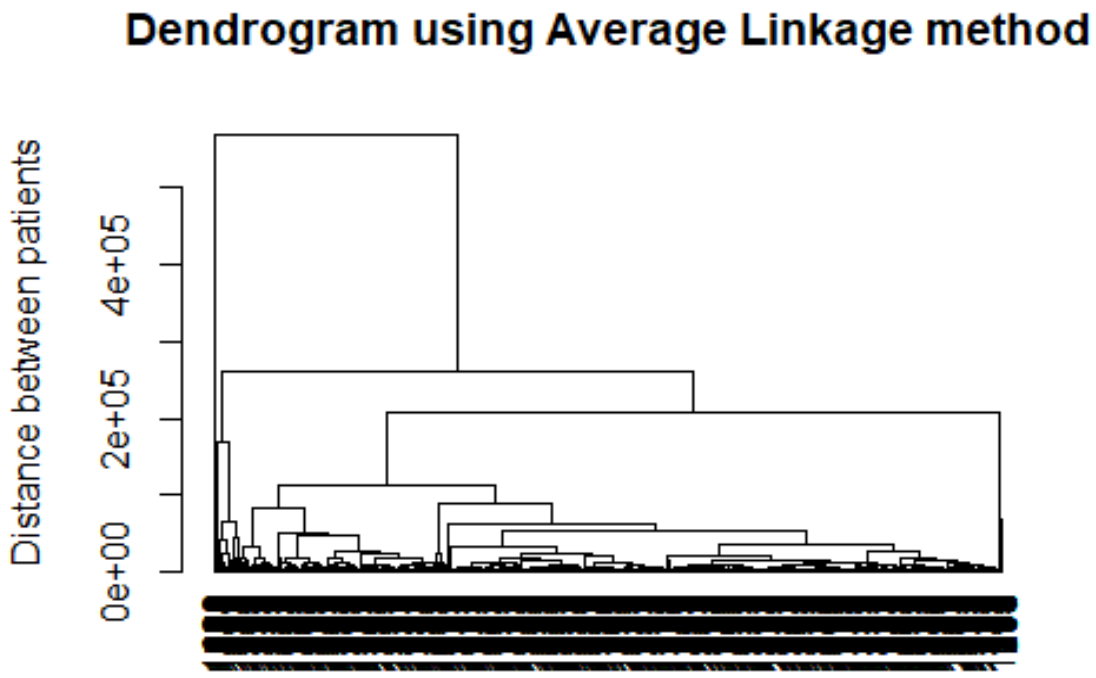
```
##
## Call:
## hclust(d = Distance, method = "single")
##
## Cluster method   : single
## Distance         : euclidean
## Number of objects: 1460
```

```
#Plotting of dendrogram using Single Linkage method
plot(as.dendrogram(clus_sales_prediction.nn),
```

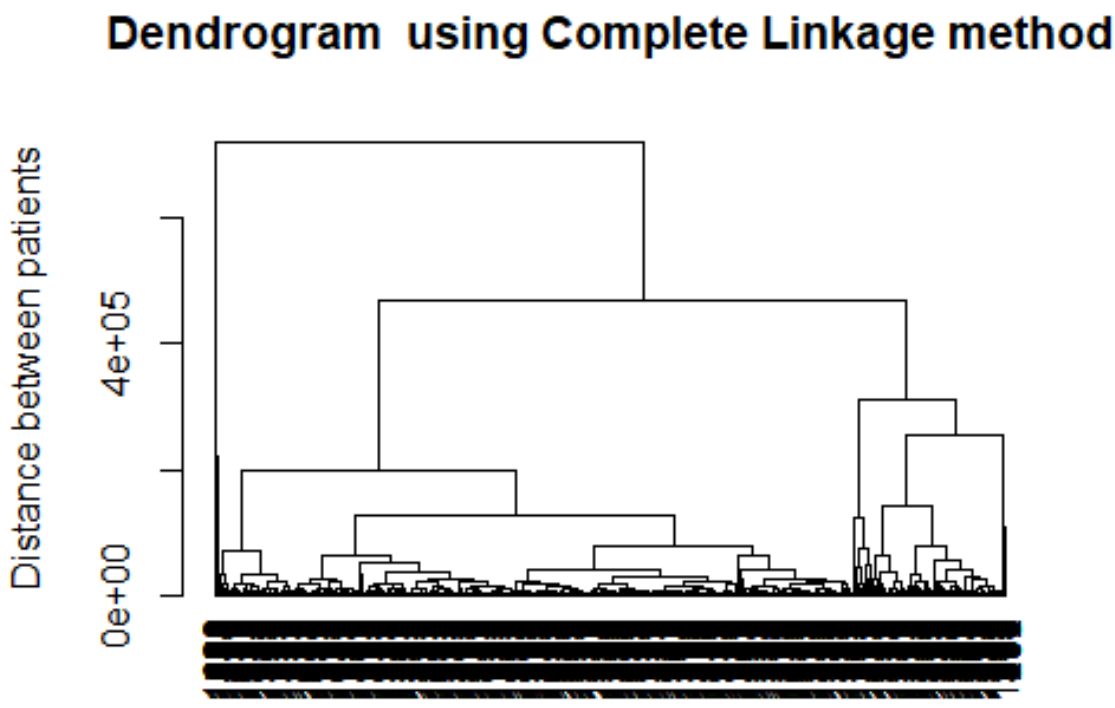
```
     ylab="Distance",
     main="Dendrogram using Single Linkage method")
```




```
#2. Average Linkage Method
clus_sales_prediction.avl <- hclust(Distance, method = "average")
clus_sales_prediction.avl

##
## Call:
## hclust(d = Distance, method = "average")
##
## Cluster method   : average
## Distance         : euclidean
## Number of objects: 1460

#Plotting of dendrogram using Average Linkage method
plot(as.dendrogram(clus_sales_prediction.avl),
     ylab="Distance between patients",
     main="Dendrogram using Average Linkage method")
```




```
#3. Complete Linkage Method
clus_sales_prediction.fn <- hclust(Distance)
clus_sales_prediction.fn

##
## Call:
## hclust(d = Distance)
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 1460

plot(as.dendrogram(clus_sales_prediction.fn),
     ylab="Distance between patients",
     main="Dendrogram  using Complete Linkage method")
```


Dendrogram using Complete Linkage method
Distance between patients
0e+00
4e+05