**(UBIT) DEPARTMENT OF COMPUTER SCIENCE**



**Critical Temperature of Superconductivity**

**MCS – Batch 2020-2021**

**Group Members:**

* **Ali Askari Zaidi (P19101004)**
* **Syeda Rida Batool Zaidi. (P19101074)**
* **Hafiz Faseeh Uddin. (P19101020)**
* **Muhammed Maaz Siddiqui. (P19101038)**

**Submitted to: Dr. Tehseen Jillani.**

**University of Karachi**

**Index**

Chapter -------------- page no

Introduction------------------??

Methodology & Visualization----------------??

* Heat Map
* Normalization
* Principal Component Analysis
* Factor Analysis
* Scree Plot
* K-Means Clustering
* Decision Regression Tree

Conclusion-------------------------??

**Superconductors**

**Introduction:**

**Superconductivity** is a set of physical properties observed in certain materials where [electrical resistance](https://en.wikipedia.org/wiki/Electrical_resistance_and_conductance) vanishes and [magnetic flux fields](https://en.wikipedia.org/wiki/Magnetic_field) are expelled from the material. Any material exhibiting these properties is a **superconductor**. Unlike an ordinary metallic [conductor](https://en.wikipedia.org/wiki/Electrical_conductor), whose resistance decreases gradually as its temperature is lowered even down to near [absolute zero](https://en.wikipedia.org/wiki/Absolute_zero), a superconductor has a characteristic [critical temperature](https://en.wikipedia.org/wiki/Phase_transition) below which the resistance drops abruptly to zero. An [electric current](https://en.wikipedia.org/wiki/Electric_current) through a loop of [superconducting wire](https://en.wikipedia.org/wiki/Superconducting_wire) can persist indefinitely with no power source temperatures

**Problem Statement:**

There are a number of superconducting materials which have critical temperatures on which they behave as superconductors. In order to understand the factors that affect these critical temperatures we use data mining techniques based on features of atoms and molecules in each superconducting material.

**List of main features:**

* **atomic\_mass**, total proton and neutron rest masses, in Atomic Mass Units (AMU).
* **fie**, First Ionization Energy, energy required to remove a valence electron, in kilo-Joules per mole (kJ/mol).
* **atomic\_radius**, calculated atomic radius, in picometer (pm).
* **density**, density at standard temperature and pressure, in kilograms per meters cubed (kg/m3).
* **electron\_affinity**, energy required to add an electron to a neutral atom, in kilo-Joules per mole (kJ/mol).
* **fusion\_heat**, energy to change from solid to liquid without temperature change, in kilo-Joules per mole (kJ/mol).
* **thermal\_conductivity**, thermal conductivity coefficient k, in watts per meter-kelvin (W/(m ?? K)).
* **valence**, typical number of chemical bonds formed by the element, no units.
* **critical\_temp**, superconductor critical temperature, in Kelvin.

**R Markdown**

**This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see** [**http://rmarkdown.rstudio.com**](http://rmarkdown.rstudio.com/)**.**

**When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:**

summary(train4)

**Including Plots**

**Heat Map and data loading**

The heat map shows the correlation among each feature and show how much one feature is correlated to the other. This will help us to use most correlated features in our data mining models. The correlation is among 40 x 40 features. This is done by dividing the data into two halves as row and column of features in corrplot or heatmap.

library(readxl)

## Warning: package 'readxl' was built under R version 4.1.2

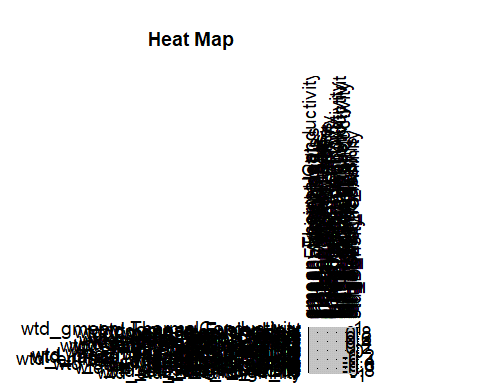
train4 <- read\_excel("train4.xlsx")  
View(train4)  
library("corrplot")

## Warning: package 'corrplot' was built under R version 4.1.2

## corrplot 0.92 loaded

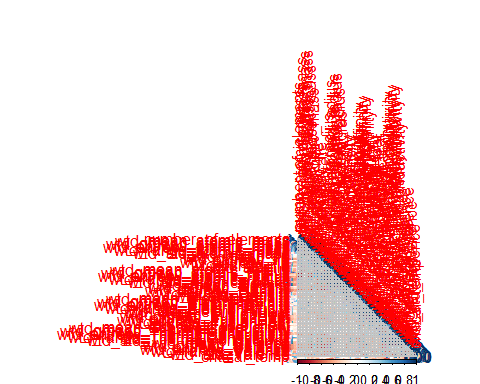
corr\_matrix <- cor(train4)  
  
*# create a new df and order columns alphabetically*  
train4\_new <- train4[, order(colnames(train4))]  
  
*# correlation*  
train4\_new.mat <- cor(train4\_new)  
  
*# remove unwanted rows and columns from the correlation matrix*  
train4\_new\_mat <- train4\_new.mat[42:82, 1:41] *#41 x 41 square correlation*  
  
*# run plot*  
corrplot(  
 train4\_new\_mat,  
 title = "Heat Map",  
 method = "circle",  
 type = "full",  
 tl.col = "black",  
 order = "hclust",  
 hclust.method = "ward.D2",  
 tl.cex = 1.2,  
 cl.cex=1.2,  
 outline = T,  
 mar=c(0,0,4,5),  
 sig.level = 0.05,  
  
)

## Warning in corrplot(train4\_new\_mat, title = "Heat Map", method = "circle", :  
## Not been able to calculate text margin, please try again with a clean new empty  
## window using {plot.new(); dev.off()} or reduce tl.cex



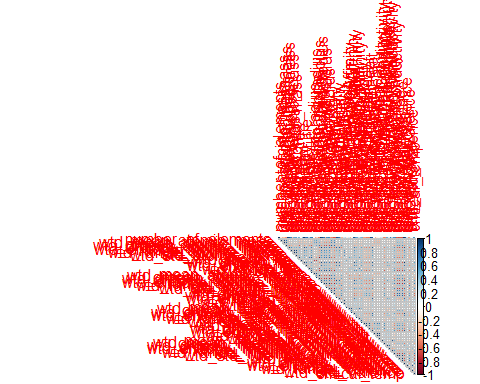
Here, 80 x 80 features are used in corrplot as heatmap.

*# with numbers and lower*  
corrplot(corr\_matrix,  
 method = 'number',  
 type = "lower")



Here, 80 x 80 corrplot as heatmap

library("corrplot")  
cr <- cor(train4)  
*#corrplot(cr)*  
corrplot(cr, type="upper")



Normalization:

Min-max normalization is done to normalize data from 0-1. This is done because the scale of features is different. Then we found the statistics using summary of the normalized data.

min\_max <- **function**(x){  
 res <- (x - min(x))/(max(x)-min(x))  
 return(res)  
}  
newd<- as.data.frame(sapply(train4[,1:82],min\_max))  
  
  
summary(newd)

## number\_of\_elements mean\_atomic\_mass wtd\_mean\_atomic\_mass gmean\_atomic\_mass  
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2500 1st Qu.:0.3243 1st Qu.:0.2257 1st Qu.:0.2589   
## Median :0.3750 Median :0.3860 Median :0.2679 Median :0.2997   
## Mean :0.3894 Mean :0.3990 Mean :0.3286 Mean :0.3239   
## 3rd Qu.:0.5000 3rd Qu.:0.4626 3rd Qu.:0.3934 3rd Qu.:0.3574   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## wtd\_gmean\_atomic\_mass entropy\_atomic\_mass wtd\_entropy\_atomic\_mass  
## Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1608 1st Qu.:0.4873 1st Qu.:0.3960   
## Median :0.1834 Median :0.6047 Median :0.5856   
## Mean :0.2733 Mean :0.5876 Mean :0.5433   
## 3rd Qu.:0.3437 3rd Qu.:0.7282 3rd Qu.:0.6942   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## range\_atomic\_mass wtd\_range\_atomic\_mass std\_atomic\_mass wtd\_std\_atomic\_mass  
## Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3775 1st Qu.:0.08183 1st Qu.:0.3256 1st Qu.:0.2825   
## Median :0.5910 Median :0.12956 Median :0.4467 Median :0.4384   
## Mean :0.5558 Mean :0.16161 Mean :0.4394 Mean :0.4103   
## 3rd Qu.:0.7411 3rd Qu.:0.18657 3rd Qu.:0.5872 3rd Qu.:0.5309   
## Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.0000   
## mean\_fie wtd\_mean\_fie gmean\_fie wtd\_gmean\_fie   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3714 1st Qu.:0.3737 1st Qu.:0.3381 1st Qu.:0.3619   
## Median :0.4153 Median :0.5290 Median :0.3759 Median :0.5049   
## Mean :0.4203 Mean :0.5089 Mean :0.3861 Mean :0.4803   
## 3rd Qu.:0.4488 3rd Qu.:0.6464 3rd Qu.:0.4162 3rd Qu.:0.5904   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## entropy\_fie wtd\_entropy\_fie range\_fie wtd\_range\_fie   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.5032 1st Qu.:0.3697 1st Qu.:0.2011 1st Qu.:0.2325   
## Median :0.6285 Median :0.4498 Median :0.5857 Median :0.4077   
## Mean :0.6021 Mean :0.4546 Mean :0.4387 Mean :0.3862   
## 3rd Qu.:0.7189 3rd Qu.:0.5208 3rd Qu.:0.6214 3rd Qu.:0.5517   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## std\_fie wtd\_std\_fie mean\_atomic\_radius wtd\_mean\_atomic\_radius  
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2284 1st Qu.:0.1941 1st Qu.:0.4053 1st Qu.:0.2565   
## Median :0.5331 Median :0.5394 Median :0.4490 Median :0.3119   
## Mean :0.4315 Mean :0.4676 Mean :0.4399 Mean :0.3469   
## 3rd Qu.:0.5958 3rd Qu.:0.7151 3rd Qu.:0.4874 3rd Qu.:0.4411   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## gmean\_atomic\_radius wtd\_gmean\_atomic\_radius entropy\_atomic\_radius  
## Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3422 1st Qu.:0.1648 1st Qu.:0.4979   
## Median :0.3792 Median :0.2607 Median :0.6213   
## Mean :0.3858 Mean :0.2920 Mean :0.5919   
## 3rd Qu.:0.4318 3rd Qu.:0.4120 3rd Qu.:0.7061   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## wtd\_entropy\_atomic\_radius range\_atomic\_radius wtd\_range\_atomic\_radius  
## Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.4476 1st Qu.:0.3125 1st Qu.:0.1191   
## Median :0.6529 Median :0.6680 Median :0.1790   
## Mean :0.5942 Mean :0.5442 Mean :0.2139   
## 3rd Qu.:0.7489 3rd Qu.:0.8008 3rd Qu.:0.2508   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## std\_atomic\_radius wtd\_std\_atomic\_radius mean\_Density wtd\_mean\_Density  
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3040 1st Qu.:0.3296 1st Qu.:0.1998 1st Qu.:0.1327   
## Median :0.5079 Median :0.6170 Median :0.2359 Median :0.1905   
## Mean :0.4468 Mean :0.5388 Mean :0.2705 Mean :0.2331   
## 3rd Qu.:0.6011 3rd Qu.:0.7595 3rd Qu.:0.2978 3rd Qu.:0.2840   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## gmean\_Density wtd\_gmean\_Density entropy\_Density wtd\_entropy\_Density  
## Min. :0.00000 Min. :0.000000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.03903 1st Qu.:0.002924 1st Qu.:0.4677 1st Qu.:0.4043   
## Median :0.05926 Median :0.067053 Median :0.5581 Median :0.5182   
## Mean :0.15314 Mean :0.137966 Mean :0.5488 Mean :0.5025   
## 3rd Qu.:0.25648 3rd Qu.:0.255224 3rd Qu.:0.6774 3rd Qu.:0.6346   
## Max. :1.00000 Max. :1.000000 Max. :1.0000 Max. :1.0000   
## range\_Density wtd\_range\_Density std\_Density wtd\_std\_Density   
## Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2943 1st Qu.:0.07385 1st Qu.:0.2629 1st Qu.:0.2463   
## Median :0.3966 Median :0.09285 Median :0.3079 Median :0.3483   
## Mean :0.3836 Mean :0.12939 Mean :0.3186 Mean :0.3188   
## 3rd Qu.:0.4329 3rd Qu.:0.15196 3rd Qu.:0.3734 3rd Qu.:0.3803   
## Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.0000   
## mean\_ElectronAffinity wtd\_mean\_ElectronAffinity gmean\_ElectronAffinity  
## Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1867 1st Qu.:0.2213 1st Qu.:0.0992   
## Median :0.2206 Median :0.3123 Median :0.1539   
## Mean :0.2322 Mean :0.2810 Mean :0.1628   
## 3rd Qu.:0.2588 3rd Qu.:0.3365 3rd Qu.:0.2033   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## wtd\_gmean\_ElectronAffinity entropy\_ElectronAffinity  
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1518 1st Qu.:0.5038   
## Median :0.2208 Median :0.6439   
## Mean :0.2185 Mean :0.6054   
## 3rd Qu.:0.2726 3rd Qu.:0.7614   
## Max. :1.0000 Max. :1.0000   
## wtd\_entropy\_ElectronAffinity range\_ElectronAffinity wtd\_range\_ElectronAffinity  
## Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3943 1st Qu.:0.2484 1st Qu.:0.1556   
## Median :0.4663 Median :0.3640 Median :0.3254   
## Mean :0.4600 Mean :0.3459 Mean :0.2713   
## 3rd Qu.:0.5238 3rd Qu.:0.3972 3rd Qu.:0.3507   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## std\_ElectronAffinity wtd\_std\_ElectronAffinity mean\_FusionHeat   
## Min. :0.0000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.2356 1st Qu.:0.1978 1st Qu.:0.07031   
## Median :0.3139 Median :0.2841 Median :0.08668   
## Mean :0.3003 Mean :0.2627 Mean :0.13432   
## 3rd Qu.:0.3451 3rd Qu.:0.3154 3rd Qu.:0.16122   
## Max. :1.0000 Max. :1.0000 Max. :1.00000   
## wtd\_mean\_FusionHeat gmean\_FusionHeat wtd\_gmean\_FusionHeat entropy\_FusionHeat  
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:0.04592 1st Qu.:0.03711 1st Qu.:0.01050 1st Qu.:0.4096   
## Median :0.07739 Median :0.04802 Median :0.04493 Median :0.5466   
## Mean :0.13005 Mean :0.09463 Mean :0.09467 Mean :0.5374   
## 3rd Qu.:0.17458 3rd Qu.:0.12768 3rd Qu.:0.15468 3rd Qu.:0.6774   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.0000   
## wtd\_entropy\_FusionHeat range\_FusionHeat wtd\_range\_FusionHeat std\_FusionHeat   
## Min. :0.0000 Min. :0.0000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.3850 1st Qu.:0.1229 1st Qu.:0.02269 1st Qu.:0.08253   
## Median :0.5695 Median :0.1229 Median :0.03347 Median :0.09583   
## Mean :0.5232 Mean :0.2018 Mean :0.08004 Mean :0.16120   
## 3rd Qu.:0.6624 3rd Qu.:0.2214 3rd Qu.:0.10225 3rd Qu.:0.17510   
## Max. :1.0000 Max. :1.0000 Max. :1.00000 Max. :1.00000   
## wtd\_std\_FusionHeat mean\_ThermalConductivity wtd\_mean\_ThermalConductivity  
## Min. :0.00000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.08908 1st Qu.:0.1834 1st Qu.:0.1331   
## Median :0.10644 Median :0.2902 Median :0.1801   
## Mean :0.14933 Mean :0.2697 Mean :0.2003   
## 3rd Qu.:0.15514 3rd Qu.:0.3338 3rd Qu.:0.2434   
## Max. :1.00000 Max. :1.0000 Max. :1.0000   
## gmean\_ThermalConductivity wtd\_gmean\_ThermalConductivity  
## Min. :0.00000 Min. :0.000000   
## 1st Qu.:0.02615 1st Qu.:0.002831   
## Median :0.04487 Median :0.016152   
## Mean :0.09380 Mean :0.072565   
## 3rd Qu.:0.13322 3rd Qu.:0.125755   
## Max. :1.00000 Max. :1.000000   
## entropy\_ThermalConductivity wtd\_entropy\_ThermalConductivity  
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2802 1st Qu.:0.1554   
## Median :0.4521 Median :0.3384   
## Mean :0.4453 Mean :0.3348   
## 3rd Qu.:0.5889 3rd Qu.:0.4819   
## Max. :1.0000 Max. :1.0000   
## range\_ThermalConductivity wtd\_range\_ThermalConductivity  
## Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.2009 1st Qu.:0.07311   
## Median :0.9298 Median :0.14088   
## Mean :0.5835 Mean :0.15453   
## 3rd Qu.:0.9302 3rd Qu.:0.22885   
## Max. :1.0000 Max. :1.00000   
## std\_ThermalConductivity wtd\_std\_ThermalConductivity mean\_Valence   
## Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1764 1st Qu.:0.1500 1st Qu.:0.2222   
## Median :0.6315 Median :0.5324 Median :0.3056   
## Mean :0.4602 Mean :0.4512 Mean :0.3664   
## 3rd Qu.:0.7154 3rd Qu.:0.7628 3rd Qu.:0.5000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## wtd\_mean\_Valence gmean\_Valence wtd\_gmean\_Valence entropy\_Valence   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1861 1st Qu.:0.2133 1st Qu.:0.1819 1st Qu.:0.4953   
## Median :0.2697 Median :0.2692 Median :0.2390 Median :0.6391   
## Mean :0.3589 Mean :0.3428 Mean :0.3426 Mean :0.6049   
## 3rd Qu.:0.5044 3rd Qu.:0.4547 3rd Qu.:0.4858 3rd Qu.:0.7419   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## wtd\_entropy\_Valence range\_Valence wtd\_range\_Valence std\_Valence   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3978 1st Qu.:0.1667 1st Qu.:0.1318 1st Qu.:0.1506   
## Median :0.5983 Median :0.3333 Median :0.1520 Median :0.2667   
## Mean :0.5400 Mean :0.3402 Mean :0.2121 Mean :0.2798   
## 3rd Qu.:0.6826 3rd Qu.:0.5000 3rd Qu.:0.2744 3rd Qu.:0.4000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## wtd\_std\_Valence critical\_temp   
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1023 1st Qu.:0.0290   
## Median :0.1667 Median :0.1081   
## Mean :0.2247 Mean :0.1861   
## 3rd Qu.:0.3401 3rd Qu.:0.3405   
## Max. :1.0000 Max. :1.0000

Principal Component Analysis(PCA):

The features are too many causing too much problems without having importance. So those features that contribute least are removed using a dimensionality reduction technique called Principal Component Analysis. In this technique, Components are made on projections of the features from the original data points resulting in components. These components tell us how much does a particular feature contributes in obtaining the target or dependent variable outcome. These components are in dimensions where each feature represent single dimension. The first principal component is the projection of data points projecting most of the information then the second principal component has the second most information as projection of data points. Here the value of 5.6156 shows that PC1 has the most standard deviation covering the data points. Proportion is there of each component and cumulative proportion is telling that how much components from PC1 till the specific PC contributes in projecting information to produce data points. Here PC17 has cumulative proportion 0.94810. which means 94.810% informations has been projected by PC1 to PC17.

library("factoextra")

## Warning: package 'factoextra' was built under R version 4.1.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.1.2

## Welcome! Want to learn more? See two factoextra-related books at <https://goo.gl/ve3WBa>

library("FactoMineR")

## Warning: package 'FactoMineR' was built under R version 4.1.2

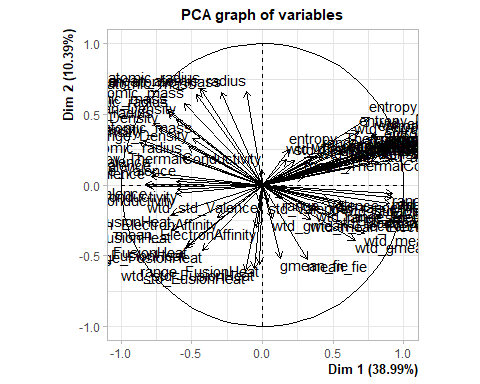
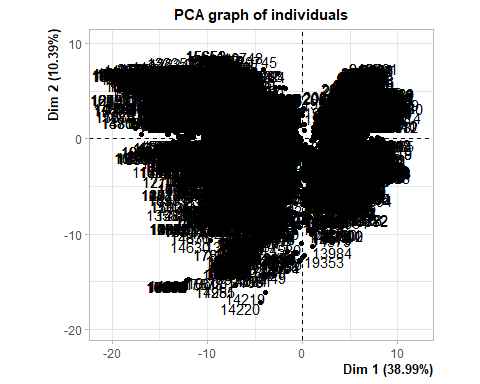
*# before the PCA analysis*  
res.pca <- prcomp(train4[,c(1:81)], scale = TRUE)  
summary(res.pca)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 5.6156 2.9139 2.77708 2.53086 2.18279 1.75174 1.71290  
## Proportion of Variance 0.3893 0.1048 0.09521 0.07908 0.05882 0.03788 0.03622  
## Cumulative Proportion 0.3893 0.4941 0.58935 0.66843 0.72725 0.76513 0.80136  
## PC8 PC9 PC10 PC11 PC12 PC13 PC14  
## Standard deviation 1.58643 1.38293 1.26573 1.21695 1.08695 0.97701 0.89935  
## Proportion of Variance 0.03107 0.02361 0.01978 0.01828 0.01459 0.01178 0.00999  
## Cumulative Proportion 0.83243 0.85604 0.87582 0.89410 0.90869 0.92047 0.93046  
## PC15 PC16 PC17 PC18 PC19 PC20 PC21  
## Standard deviation 0.89208 0.79563 0.76303 0.66348 0.62570 0.55602 0.49481  
## Proportion of Variance 0.00982 0.00782 0.00719 0.00543 0.00483 0.00382 0.00302  
## Cumulative Proportion 0.94028 0.94810 0.95529 0.96072 0.96555 0.96937 0.97239  
## PC22 PC23 PC24 PC25 PC26 PC27 PC28  
## Standard deviation 0.48177 0.45580 0.40959 0.39968 0.38845 0.3711 0.33985  
## Proportion of Variance 0.00287 0.00256 0.00207 0.00197 0.00186 0.0017 0.00143  
## Cumulative Proportion 0.97526 0.97782 0.97989 0.98187 0.98373 0.9854 0.98686  
## PC29 PC30 PC31 PC32 PC33 PC34 PC35  
## Standard deviation 0.31984 0.30537 0.28801 0.27892 0.27287 0.24129 0.23554  
## Proportion of Variance 0.00126 0.00115 0.00102 0.00096 0.00092 0.00072 0.00068  
## Cumulative Proportion 0.98812 0.98927 0.99029 0.99125 0.99217 0.99289 0.99358  
## PC36 PC37 PC38 PC39 PC40 PC41 PC42  
## Standard deviation 0.22426 0.21502 0.19983 0.18811 0.18503 0.16253 0.15744  
## Proportion of Variance 0.00062 0.00057 0.00049 0.00044 0.00042 0.00033 0.00031  
## Cumulative Proportion 0.99420 0.99477 0.99526 0.99570 0.99612 0.99645 0.99675  
## PC43 PC44 PC45 PC46 PC47 PC48 PC49  
## Standard deviation 0.14428 0.13868 0.13456 0.13208 0.1262 0.12326 0.12100  
## Proportion of Variance 0.00026 0.00024 0.00022 0.00022 0.0002 0.00019 0.00018  
## Cumulative Proportion 0.99701 0.99725 0.99747 0.99769 0.9979 0.99807 0.99825  
## PC50 PC51 PC52 PC53 PC54 PC55 PC56  
## Standard deviation 0.11914 0.11255 0.11162 0.10131 0.09863 0.09771 0.09242  
## Proportion of Variance 0.00018 0.00016 0.00015 0.00013 0.00012 0.00012 0.00011  
## Cumulative Proportion 0.99843 0.99858 0.99874 0.99886 0.99898 0.99910 0.99921  
## PC57 PC58 PC59 PC60 PC61 PC62 PC63  
## Standard deviation 0.08503 0.08118 0.08045 0.07627 0.07243 0.06789 0.05995  
## Proportion of Variance 0.00009 0.00008 0.00008 0.00007 0.00006 0.00006 0.00004  
## Cumulative Proportion 0.99930 0.99938 0.99946 0.99953 0.99959 0.99965 0.99970  
## PC64 PC65 PC66 PC67 PC68 PC69 PC70  
## Standard deviation 0.05968 0.05650 0.05349 0.05107 0.04773 0.04298 0.04084  
## Proportion of Variance 0.00004 0.00004 0.00004 0.00003 0.00003 0.00002 0.00002  
## Cumulative Proportion 0.99974 0.99978 0.99981 0.99985 0.99988 0.99990 0.99992  
## PC71 PC72 PC73 PC74 PC75 PC76 PC77  
## Standard deviation 0.03832 0.03676 0.03457 0.02734 0.02483 0.02113 0.01819  
## Proportion of Variance 0.00002 0.00002 0.00001 0.00001 0.00001 0.00001 0.00000  
## Cumulative Proportion 0.99994 0.99995 0.99997 0.99998 0.99999 0.99999 0.99999  
## PC78 PC79 PC80 PC81  
## Standard deviation 0.01366 0.01096 0.008603 0.007042  
## Proportion of Variance 0.00000 0.00000 0.000000 0.000000  
## Cumulative Proportion 1.00000 1.00000 1.000000 1.000000

*# Extract the eigenvalues/variances*  
  
res.pca<-prcomp(train4,scale. = TRUE)

pca <- PCA(train4, scale.unit=TRUE, ncp=5, graph=T)

The below PCA graph represent what data points are represented by which component or dimension the most.



The command below prints the dimensions which have features and each dimension shows the features it represents. This is shown by correlation on each dimension of a particular feature.

dimdesc(pca, proba = 0.05)[1:3]

## $Dim.1  
## $quanti  
## correlation p.value  
## range\_fie 0.92046417 0.000000e+00  
## range\_atomic\_radius 0.91904268 0.000000e+00  
## wtd\_entropy\_atomic\_radius 0.91778556 0.000000e+00  
## wtd\_std\_fie 0.91748375 0.000000e+00  
## wtd\_std\_atomic\_radius 0.90113776 0.000000e+00  
## entropy\_Valence 0.89176439 0.000000e+00  
## wtd\_entropy\_atomic\_mass 0.87973401 0.000000e+00  
## entropy\_fie 0.87923400 0.000000e+00  
## wtd\_entropy\_Valence 0.87629309 0.000000e+00  
## number\_of\_elements 0.87535434 0.000000e+00  
## std\_fie 0.87131505 0.000000e+00  
## entropy\_atomic\_radius 0.86123082 0.000000e+00  
## std\_atomic\_radius 0.84056075 0.000000e+00  
## entropy\_atomic\_mass 0.82243946 0.000000e+00  
## range\_ThermalConductivity 0.80867436 0.000000e+00  
## wtd\_entropy\_FusionHeat 0.80430780 0.000000e+00  
## entropy\_ElectronAffinity 0.78446897 0.000000e+00  
## wtd\_std\_ThermalConductivity 0.78436087 0.000000e+00  
## std\_ThermalConductivity 0.75894547 0.000000e+00  
## entropy\_FusionHeat 0.75441643 0.000000e+00  
## wtd\_mean\_fie 0.72227709 0.000000e+00  
## range\_atomic\_mass 0.69955467 0.000000e+00  
## entropy\_Density 0.68480768 0.000000e+00  
## critical\_temp 0.66902484 0.000000e+00  
## wtd\_gmean\_fie 0.65693036 0.000000e+00  
## wtd\_entropy\_Density 0.65074563 0.000000e+00  
## wtd\_std\_atomic\_mass 0.63176259 0.000000e+00  
## wtd\_entropy\_fie 0.61762155 0.000000e+00  
## range\_ElectronAffinity 0.58416110 0.000000e+00  
## wtd\_std\_ElectronAffinity 0.57589779 0.000000e+00  
## std\_atomic\_mass 0.55321568 0.000000e+00  
## wtd\_entropy\_ElectronAffinity 0.53933799 0.000000e+00  
## std\_ElectronAffinity 0.51424829 0.000000e+00  
## wtd\_range\_fie 0.47779881 0.000000e+00  
## range\_Density 0.42402538 0.000000e+00  
## wtd\_std\_Density 0.40022557 0.000000e+00  
## wtd\_range\_ElectronAffinity 0.38367867 0.000000e+00  
## wtd\_range\_ThermalConductivity 0.34996606 0.000000e+00  
## wtd\_mean\_ElectronAffinity 0.33809853 0.000000e+00  
## mean\_fie 0.31701654 0.000000e+00  
## mean\_ThermalConductivity 0.29620194 0.000000e+00  
## std\_Density 0.22826110 2.178196e-249  
## entropy\_ThermalConductivity 0.18940244 6.279825e-171  
## wtd\_mean\_ThermalConductivity 0.17076247 7.570906e-139  
## range\_Valence 0.14129766 3.023857e-95  
## gmean\_fie 0.12860001 4.338743e-79  
## wtd\_gmean\_ElectronAffinity 0.06980364 2.186663e-24  
## std\_Valence 0.04090541 2.417922e-09  
## range\_FusionHeat -0.02055268 2.725618e-03  
## wtd\_std\_Valence -0.02747752 6.143021e-05  
## mean\_ElectronAffinity -0.04129886 1.697919e-09  
## wtd\_std\_FusionHeat -0.06075596 7.513841e-19  
## mean\_atomic\_radius -0.11342657 7.926033e-62  
## std\_FusionHeat -0.11765823 2.030178e-66  
## mean\_atomic\_mass -0.28868269 0.000000e+00  
## gmean\_ElectronAffinity -0.31767723 0.000000e+00  
## wtd\_range\_FusionHeat -0.42532385 0.000000e+00  
## gmean\_atomic\_radius -0.43730065 0.000000e+00  
## gmean\_atomic\_mass -0.46554120 0.000000e+00  
## wtd\_range\_atomic\_mass -0.49355092 0.000000e+00  
## wtd\_range\_Density -0.52343837 0.000000e+00  
## mean\_FusionHeat -0.52899113 0.000000e+00  
## wtd\_mean\_FusionHeat -0.54398504 0.000000e+00  
## wtd\_mean\_atomic\_mass -0.55531207 0.000000e+00  
## wtd\_range\_atomic\_radius -0.56486605 0.000000e+00  
## wtd\_gmean\_ThermalConductivity -0.60308214 0.000000e+00  
## wtd\_range\_Valence -0.60708527 0.000000e+00  
## mean\_Density -0.61054661 0.000000e+00  
## gmean\_ThermalConductivity -0.61551155 0.000000e+00  
## wtd\_gmean\_FusionHeat -0.64409026 0.000000e+00  
## gmean\_FusionHeat -0.64829636 0.000000e+00  
## wtd\_mean\_atomic\_radius -0.66938644 0.000000e+00  
## wtd\_gmean\_atomic\_mass -0.67216169 0.000000e+00  
## wtd\_mean\_Density -0.73433523 0.000000e+00  
## wtd\_gmean\_atomic\_radius -0.77268185 0.000000e+00  
## mean\_Valence -0.78662037 0.000000e+00  
## gmean\_Valence -0.79361492 0.000000e+00  
## wtd\_mean\_Valence -0.81445591 0.000000e+00  
## wtd\_gmean\_Valence -0.82539165 0.000000e+00  
## gmean\_Density -0.84446153 0.000000e+00  
## wtd\_gmean\_Density -0.87301402 0.000000e+00  
##   
## attr(,"class")  
## [1] "condes" "list"   
##   
## $Dim.2  
## $quanti  
## correlation p.value  
## gmean\_atomic\_radius 0.68167713 0.000000e+00  
## mean\_atomic\_radius 0.66504058 0.000000e+00  
## mean\_atomic\_mass 0.65644178 0.000000e+00  
## gmean\_atomic\_mass 0.63929439 0.000000e+00  
## wtd\_mean\_atomic\_mass 0.57529754 0.000000e+00  
## wtd\_gmean\_atomic\_mass 0.53587674 0.000000e+00  
## wtd\_mean\_atomic\_radius 0.51633720 0.000000e+00  
## entropy\_FusionHeat 0.47673743 0.000000e+00  
## mean\_Density 0.46398703 0.000000e+00  
## wtd\_gmean\_atomic\_radius 0.43750465 0.000000e+00  
## wtd\_mean\_Density 0.40344396 0.000000e+00  
## entropy\_Density 0.38191494 0.000000e+00  
## wtd\_entropy\_FusionHeat 0.36005144 0.000000e+00  
## entropy\_atomic\_mass 0.34484284 0.000000e+00  
## wtd\_range\_atomic\_mass 0.33231029 0.000000e+00  
## wtd\_entropy\_Density 0.32310756 0.000000e+00  
## gmean\_Density 0.31649483 0.000000e+00  
## wtd\_gmean\_Density 0.29871581 0.000000e+00  
## entropy\_atomic\_radius 0.28634199 0.000000e+00  
## wtd\_range\_Density 0.26996354 0.000000e+00  
## number\_of\_elements 0.25460952 8.013862e-312  
## entropy\_ElectronAffinity 0.25206096 1.899392e-305  
## entropy\_Valence 0.25103947 6.505063e-303  
## entropy\_ThermalConductivity 0.24904279 5.420126e-298  
## entropy\_fie 0.24348872 1.544223e-284  
## wtd\_entropy\_atomic\_mass 0.23132177 2.993597e-256  
## range\_Density 0.22192520 1.666650e-235  
## wtd\_entropy\_Valence 0.21726943 1.411601e-225  
## wtd\_std\_ThermalConductivity 0.21319731 4.404216e-217  
## range\_atomic\_mass 0.20607990 1.173004e-202  
## wtd\_entropy\_fie 0.20530750 4.006428e-201  
## range\_ThermalConductivity 0.19517664 1.377324e-181  
## wtd\_range\_ThermalConductivity 0.19479479 7.153866e-181  
## wtd\_entropy\_ElectronAffinity 0.19077407 1.982001e-173  
## wtd\_range\_atomic\_radius 0.18899701 3.415364e-170  
## critical\_temp 0.18711175 8.538577e-167  
## wtd\_entropy\_atomic\_radius 0.18283110 3.247462e-159  
## wtd\_mean\_ThermalConductivity 0.17936794 3.215705e-153  
## std\_Density 0.17330578 5.122922e-143  
## std\_atomic\_mass 0.16591035 4.502477e-131  
## std\_ThermalConductivity 0.15904763 1.765521e-120  
## wtd\_std\_Density 0.14830786 7.688343e-105  
## wtd\_std\_atomic\_mass 0.12588009 7.848046e-76  
## wtd\_entropy\_ThermalConductivity 0.11141295 1.056165e-59  
## gmean\_Valence 0.08941303 5.324612e-39  
## mean\_Valence 0.06219686 1.108918e-19  
## mean\_ThermalConductivity 0.06037699 1.233762e-18  
## wtd\_range\_Valence 0.02332677 6.695946e-04  
## wtd\_gmean\_ThermalConductivity -0.03475167 4.006161e-07  
## gmean\_ThermalConductivity -0.05864941 1.138246e-17  
## range\_atomic\_radius -0.06176945 1.965100e-19  
## range\_fie -0.08570388 5.839791e-36  
## wtd\_std\_fie -0.09058521 5.478348e-40  
## range\_Valence -0.09275429 7.532435e-42  
## wtd\_range\_fie -0.10904607 2.962440e-57  
## wtd\_std\_Valence -0.11033541 1.395723e-58  
## std\_Valence -0.11799068 8.700596e-67  
## wtd\_std\_atomic\_radius -0.13096386 5.600872e-82  
## wtd\_std\_ElectronAffinity -0.13130401 2.128488e-82  
## wtd\_range\_ElectronAffinity -0.17236895 1.790575e-141  
## range\_ElectronAffinity -0.17939302 2.912712e-153  
## std\_fie -0.18091637 6.943638e-156  
## gmean\_FusionHeat -0.20667041 7.811820e-204  
## std\_atomic\_radius -0.21271336 4.381935e-216  
## std\_ElectronAffinity -0.21438203 1.551538e-219  
## gmean\_ElectronAffinity -0.23137046 2.323850e-256  
## wtd\_gmean\_ElectronAffinity -0.23833055 2.393993e-272  
## wtd\_mean\_ElectronAffinity -0.24087723 2.495171e-278  
## mean\_ElectronAffinity -0.30386218 0.000000e+00  
## wtd\_gmean\_FusionHeat -0.32428766 0.000000e+00  
## wtd\_mean\_fie -0.34592732 0.000000e+00  
## wtd\_gmean\_fie -0.39372832 0.000000e+00  
## mean\_FusionHeat -0.42535096 0.000000e+00  
## wtd\_mean\_FusionHeat -0.44636614 0.000000e+00  
## wtd\_range\_FusionHeat -0.46339928 0.000000e+00  
## gmean\_fie -0.51908230 0.000000e+00  
## mean\_fie -0.52408945 0.000000e+00  
## range\_FusionHeat -0.56051346 0.000000e+00  
## wtd\_std\_FusionHeat -0.59069348 0.000000e+00  
## std\_FusionHeat -0.61134638 0.000000e+00  
##   
## attr(,"class")  
## [1] "condes" "list"   
##   
## $Dim.3  
## $quanti  
## correlation p.value  
## wtd\_entropy\_ThermalConductivity 0.68321723 0.000000e+00  
## wtd\_std\_Valence 0.64580114 0.000000e+00  
## range\_Valence 0.63133780 0.000000e+00  
## std\_Valence 0.58536906 0.000000e+00  
## wtd\_entropy\_fie 0.58200794 0.000000e+00  
## entropy\_ThermalConductivity 0.57343410 0.000000e+00  
## wtd\_entropy\_ElectronAffinity 0.54880498 0.000000e+00  
## range\_FusionHeat 0.47445960 0.000000e+00  
## wtd\_std\_FusionHeat 0.46712185 0.000000e+00  
## std\_Density 0.44986330 0.000000e+00  
## std\_FusionHeat 0.44016177 0.000000e+00  
## wtd\_entropy\_Density 0.42136393 0.000000e+00  
## range\_Density 0.41615003 0.000000e+00  
## range\_ElectronAffinity 0.40041365 0.000000e+00  
## wtd\_std\_Density 0.37209554 0.000000e+00  
## mean\_FusionHeat 0.36381373 0.000000e+00  
## std\_ElectronAffinity 0.36302687 0.000000e+00  
## wtd\_std\_ElectronAffinity 0.35406723 0.000000e+00  
## wtd\_mean\_Valence 0.35025512 0.000000e+00  
## wtd\_mean\_FusionHeat 0.34661815 0.000000e+00  
## mean\_Valence 0.34157900 0.000000e+00  
## gmean\_fie 0.33900866 0.000000e+00  
## mean\_ElectronAffinity 0.33585141 0.000000e+00  
## std\_atomic\_mass 0.33498785 0.000000e+00  
## range\_atomic\_mass 0.30696842 0.000000e+00  
## wtd\_gmean\_Valence 0.30195072 0.000000e+00  
## wtd\_entropy\_Valence 0.30191882 0.000000e+00  
## wtd\_gmean\_FusionHeat 0.29789638 0.000000e+00  
## wtd\_std\_atomic\_mass 0.29338235 0.000000e+00  
## gmean\_FusionHeat 0.28887915 0.000000e+00  
## wtd\_mean\_Density 0.28687515 0.000000e+00  
## mean\_fie 0.28250884 0.000000e+00  
## entropy\_fie 0.27933930 0.000000e+00  
## entropy\_atomic\_radius 0.27573143 0.000000e+00  
## gmean\_Valence 0.27392631 0.000000e+00  
## entropy\_ElectronAffinity 0.27274639 0.000000e+00  
## entropy\_Density 0.24890423 1.185366e-297  
## wtd\_entropy\_FusionHeat 0.23744745 2.734582e-270  
## wtd\_range\_FusionHeat 0.23162605 6.144149e-257  
## mean\_Density 0.22689063 2.386834e-246  
## number\_of\_elements 0.22335091 1.365638e-238  
## wtd\_entropy\_atomic\_radius 0.22119393 6.259242e-234  
## gmean\_ElectronAffinity 0.22060879 1.128279e-232  
## entropy\_Valence 0.21270633 4.530384e-216  
## wtd\_gmean\_atomic\_radius 0.19422870 8.175360e-180  
## wtd\_gmean\_Density 0.19253300 1.152796e-176  
## entropy\_atomic\_mass 0.18581969 1.734197e-164  
## wtd\_range\_Density 0.18299026 1.710200e-159  
## wtd\_mean\_atomic\_mass 0.18078365 1.177680e-155  
## entropy\_FusionHeat 0.17742541 6.561092e-150  
## wtd\_gmean\_ThermalConductivity 0.17241498 1.504371e-141  
## wtd\_mean\_atomic\_radius 0.17026823 4.806763e-138  
## gmean\_Density 0.16331523 5.183589e-127  
## wtd\_range\_atomic\_mass 0.15842074 1.552969e-119  
## wtd\_entropy\_atomic\_mass 0.15019643 1.653360e-107  
## gmean\_ThermalConductivity 0.12945478 3.971103e-80  
## wtd\_range\_Valence 0.12738383 1.266825e-77  
## wtd\_gmean\_atomic\_mass 0.10370126 6.355040e-52  
## mean\_atomic\_mass 0.05446844 1.896493e-15  
## wtd\_range\_atomic\_radius 0.03674694 8.327960e-08  
## wtd\_mean\_ElectronAffinity 0.02758685 5.741283e-05  
## range\_fie 0.01558255 2.307284e-02  
## std\_fie -0.02126057 1.932922e-03  
## gmean\_atomic\_mass -0.03798606 3.011970e-08  
## range\_atomic\_radius -0.04130919 1.682152e-09  
## std\_atomic\_radius -0.04469708 7.005865e-11  
## wtd\_mean\_ThermalConductivity -0.05505559 9.454669e-16  
## wtd\_gmean\_ElectronAffinity -0.06788064 3.800818e-23  
## wtd\_std\_atomic\_radius -0.08810064 6.557393e-38  
## wtd\_range\_ElectronAffinity -0.13351733 3.685527e-85  
## wtd\_std\_fie -0.13800671 6.588525e-91  
## wtd\_gmean\_fie -0.16612006 2.100548e-131  
## gmean\_atomic\_radius -0.17605024 1.370420e-147  
## wtd\_mean\_fie -0.19096452 8.878314e-174  
## mean\_ThermalConductivity -0.19512477 1.723152e-181  
## critical\_temp -0.19900969 7.464744e-189  
## wtd\_range\_ThermalConductivity -0.20451624 1.469433e-199  
## mean\_atomic\_radius -0.26974046 0.000000e+00  
## range\_ThermalConductivity -0.27440666 0.000000e+00  
## wtd\_std\_ThermalConductivity -0.29175148 0.000000e+00  
## std\_ThermalConductivity -0.32058964 0.000000e+00  
## wtd\_range\_fie -0.40641062 0.000000e+00  
##   
## attr(,"class")  
## [1] "condes" "list"

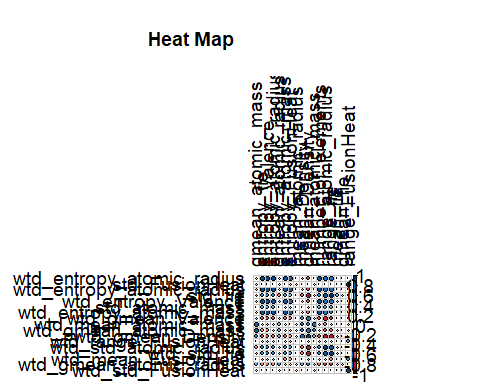
A new dataframe has been created based on the most contributing features by checking the above PCA analysis.

*# creating new df with reduced features*  
dfnew = data.frame()  
dfnew <- train4[,c("range\_fie" , "range\_atomic\_radius" , "wtd\_entropy\_atomic\_radius" , "wtd\_std\_fie"   
 , "wtd\_std\_atomic\_radius" , "entropy\_Valence" , "wtd\_entropy\_atomic\_mass" , "entropy\_fie"   
 , "wtd\_entropy\_Valence"  
 , "number\_of\_elements" , "wtd\_gmean\_Density" , "std\_fie" , "entropy\_atomic\_radius" , "gmean\_Density"   
 , "std\_atomic\_mass" , "wtd\_gmean\_Valence" , "entropy\_atomic\_mass" , "mean\_atomic\_radius"   
 , "gmean\_atomic\_radius"   
 , "mean\_atomic\_mass" , "gmean\_atomic\_mass" , "std\_FusionHeat" , "wtd\_std\_FusionHeat"   
 , "wtd\_mean\_atomic\_mass"   
 , "range\_FusionHeat" , "wtd\_gmean\_atomic\_mass" , "mean\_fie" , "gmean\_fie" , "wtd\_entropy\_atomic\_radius"   
 , "entropy\_FusionHeat" , "mean\_Density" , "wtd\_range\_FusionHeat" , "wtd\_mean\_FusionHeat"   
 , "wtd\_gmean\_atomic\_radius")]  
head(dfnew)

## # A tibble: 6 x 34  
## range\_fie range\_atomic\_radius wtd\_entropy\_atomi~ wtd\_std\_fie wtd\_std\_atomic\_r~  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 811. 205 1.21 356. 69.2  
## 2 811. 205 1.20 355. 68.0  
## 3 811. 205 1.13 355. 67.8  
## 4 811. 205 1.17 355. 68.5  
## 5 811. 205 1.26 356. 70.6  
## 6 811. 205 1.33 358. 73.3  
## # ... with 29 more variables: entropy\_Valence <dbl>,  
## # wtd\_entropy\_atomic\_mass <dbl>, entropy\_fie <dbl>,  
## # wtd\_entropy\_Valence <dbl>, number\_of\_elements <dbl>,  
## # wtd\_gmean\_Density <dbl>, std\_fie <dbl>, entropy\_atomic\_radius <dbl>,  
## # gmean\_Density <dbl>, std\_atomic\_mass <dbl>, wtd\_gmean\_Valence <dbl>,  
## # entropy\_atomic\_mass <dbl>, mean\_atomic\_radius <dbl>,  
## # gmean\_atomic\_radius <dbl>, mean\_atomic\_mass <dbl>, ...

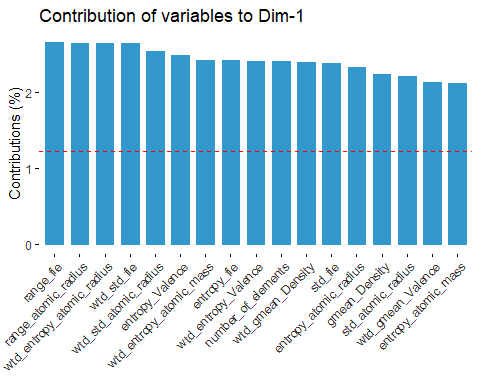
A new corrplot has been constructed based on the new dataset.

dfnew\_matrix <- cor(dfnew)  
dfnew\_new <- dfnew[, order(colnames(dfnew))]  
  
*# correlation*  
dfnew\_new.mat <- cor(dfnew\_new)  
  
*# remove unwanted rows and columns from the correlation matrix*  
dfnew\_new\_mat <- dfnew\_new.mat[18:34, 1:17] *#41 x 41 square correlation*  
  
*# run plot*  
corrplot(  
 dfnew\_new\_mat,  
 title = "Heat Map",  
 method = "circle",  
 type = "full",  
 tl.col = "black",  
 order = "hclust",  
 hclust.method = "ward.D2",  
 tl.cex = 1.2,  
 cl.cex=1.2,  
 outline = T,  
 mar=c(0,0,4,5),  
 sig.level = 0.05,  
  
)

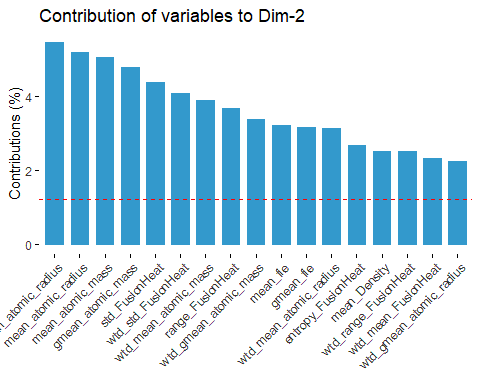


The below barplot shows the contribution by percentage of each component.

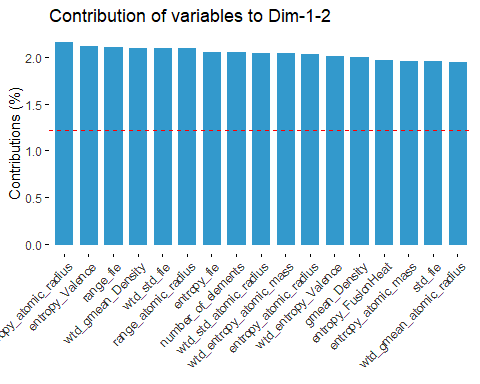
fviz\_contrib(res.pca, choice = "var", axes = 1, top = 17)



*# Contributions of variables to PC2*  
fviz\_contrib(res.pca, choice = "var", axes = 2, top = 17)

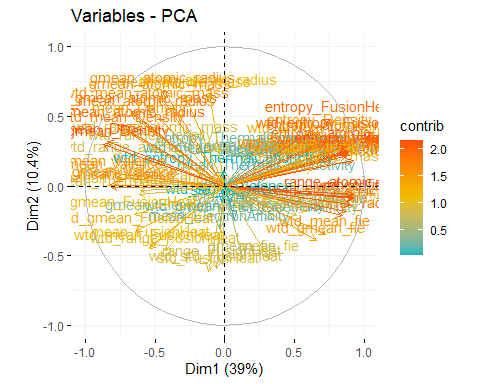


fviz\_contrib(res.pca, choice = "var", axes = 1:2, top = 17)



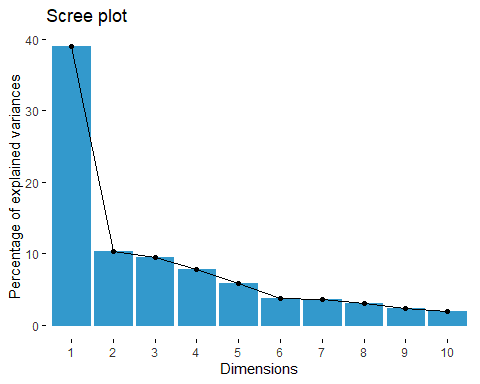
The below graph shows the factor plot which represent each feature as vector in 2 dimensions of principal component. The larger the vector, the more it contributes in particular dimension.

fviz\_pca\_var(res.pca, col.var = "contrib",  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07")  
 )



Next is scree plot. The screeplot shows the explained variances vs each dimension taken in explaining it.

library("factoextra")  
fviz\_eig(res.pca)

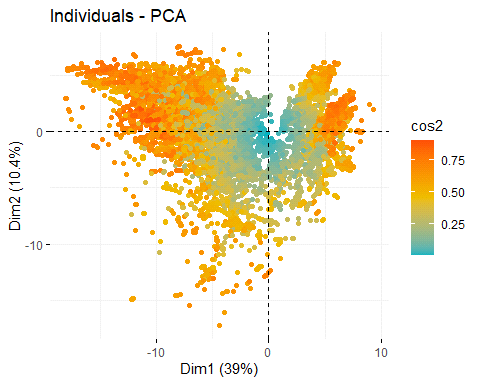


get\_eig(res.pca)

## eigenvalue variance.percent cumulative.variance.percent  
## Dim.1 3.197402e+01 3.899271e+01 38.99271  
## Dim.2 8.522273e+00 1.039302e+01 49.38573  
## Dim.3 7.750981e+00 9.452416e+00 58.83814  
## Dim.4 6.414113e+00 7.822089e+00 66.66023  
## Dim.5 4.793358e+00 5.845559e+00 72.50579  
## Dim.6 3.103236e+00 3.784435e+00 76.29022  
## Dim.7 2.935335e+00 3.579677e+00 79.86990  
## Dim.8 2.517540e+00 3.070171e+00 82.94007  
## Dim.9 1.919869e+00 2.341304e+00 85.28138  
## Dim.10 1.602099e+00 1.953779e+00 87.23516  
## Dim.11 1.496689e+00 1.825231e+00 89.06039  
## Dim.12 1.186767e+00 1.447277e+00 90.50766  
## Dim.13 9.562231e-01 1.166126e+00 91.67379  
## Dim.14 8.168746e-01 9.961886e-01 92.66998  
## Dim.15 8.081725e-01 9.855762e-01 93.65555  
## Dim.16 6.331548e-01 7.721400e-01 94.42769  
## Dim.17 5.823045e-01 7.101274e-01 95.13782  
## Dim.18 4.948232e-01 6.034430e-01 95.74126  
## Dim.19 4.083654e-01 4.980066e-01 96.23927  
## Dim.20 3.772613e-01 4.600747e-01 96.69935  
## Dim.21 2.821501e-01 3.440855e-01 97.04343  
## Dim.22 2.432141e-01 2.966026e-01 97.34003  
## Dim.23 2.313789e-01 2.821694e-01 97.62220  
## Dim.24 1.840713e-01 2.244772e-01 97.84668  
## Dim.25 1.676687e-01 2.044740e-01 98.05115  
## Dim.26 1.522261e-01 1.856416e-01 98.23680  
## Dim.27 1.377349e-01 1.679694e-01 98.40477  
## Dim.28 1.337016e-01 1.630507e-01 98.56782  
## Dim.29 1.152702e-01 1.405734e-01 98.70839  
## Dim.30 1.022925e-01 1.247469e-01 98.83314  
## Dim.31 9.281349e-02 1.131872e-01 98.94632  
## Dim.32 8.073964e-02 9.846298e-02 99.04479  
## Dim.33 7.778120e-02 9.485513e-02 99.13964  
## Dim.34 7.414602e-02 9.042198e-02 99.23006  
## Dim.35 5.813544e-02 7.089688e-02 99.30096  
## Dim.36 5.540346e-02 6.756520e-02 99.36853  
## Dim.37 4.931143e-02 6.013589e-02 99.42866  
## Dim.38 4.619792e-02 5.633893e-02 99.48500  
## Dim.39 3.990563e-02 4.866540e-02 99.53367  
## Dim.40 3.538367e-02 4.315082e-02 99.57682  
## Dim.41 3.423373e-02 4.174845e-02 99.61857  
## Dim.42 2.605810e-02 3.177817e-02 99.65034  
## Dim.43 2.476183e-02 3.019736e-02 99.68054  
## Dim.44 2.057096e-02 2.508653e-02 99.70563  
## Dim.45 1.918792e-02 2.339990e-02 99.72903  
## Dim.46 1.810256e-02 2.207629e-02 99.75110  
## Dim.47 1.739759e-02 2.121657e-02 99.77232  
## Dim.48 1.591674e-02 1.941066e-02 99.79173  
## Dim.49 1.516501e-02 1.849391e-02 99.81022  
## Dim.50 1.463569e-02 1.784841e-02 99.82807  
## Dim.51 1.418925e-02 1.730396e-02 99.84538  
## Dim.52 1.266441e-02 1.544440e-02 99.86082  
## Dim.53 1.245629e-02 1.519060e-02 99.87601  
## Dim.54 1.026163e-02 1.251419e-02 99.88853  
## Dim.55 9.551321e-03 1.164795e-02 99.90017  
## Dim.56 9.534663e-03 1.162764e-02 99.91180  
## Dim.57 8.529773e-03 1.040216e-02 99.92220  
## Dim.58 7.219275e-03 8.803994e-03 99.93101  
## Dim.59 6.588780e-03 8.035097e-03 99.93904  
## Dim.60 6.348146e-03 7.741642e-03 99.94678  
## Dim.61 5.704030e-03 6.956134e-03 99.95374  
## Dim.62 5.244311e-03 6.395501e-03 99.96014  
## Dim.63 4.600197e-03 5.609996e-03 99.96575  
## Dim.64 3.579661e-03 4.365440e-03 99.97011  
## Dim.65 3.561120e-03 4.342829e-03 99.97445  
## Dim.66 3.186982e-03 3.886563e-03 99.97834  
## Dim.67 2.848705e-03 3.474031e-03 99.98181  
## Dim.68 2.582506e-03 3.149397e-03 99.98496  
## Dim.69 2.249312e-03 2.743064e-03 99.98771  
## Dim.70 1.843828e-03 2.248571e-03 99.98996  
## Dim.71 1.657275e-03 2.021067e-03 99.99198  
## Dim.72 1.467499e-03 1.789633e-03 99.99377  
## Dim.73 1.350994e-03 1.647553e-03 99.99541  
## Dim.74 1.191890e-03 1.453524e-03 99.99687  
## Dim.75 7.460413e-04 9.098065e-04 99.99778  
## Dim.76 6.161272e-04 7.513747e-04 99.99853  
## Dim.77 4.455887e-04 5.434008e-04 99.99907  
## Dim.78 3.309852e-04 4.036405e-04 99.99948  
## Dim.79 1.866172e-04 2.275820e-04 99.99970  
## Dim.80 1.195923e-04 1.458443e-04 99.99985  
## Dim.81 7.397678e-05 9.021558e-05 99.99994  
## Dim.82 4.949803e-05 6.036345e-05 100.00000

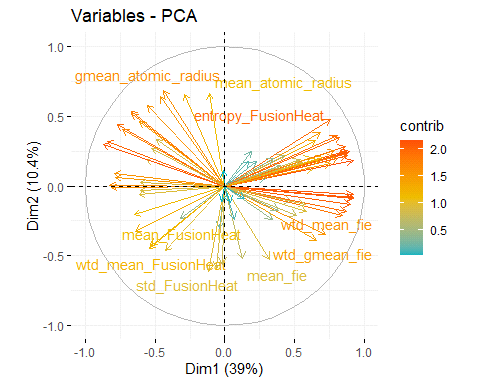
fviz\_pca\_ind(res.pca,  
 col.ind = "cos2", *# Color by the quality of representation*  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE *# Avoid text overlapping*  
)

## Warning: ggrepel: 21263 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps

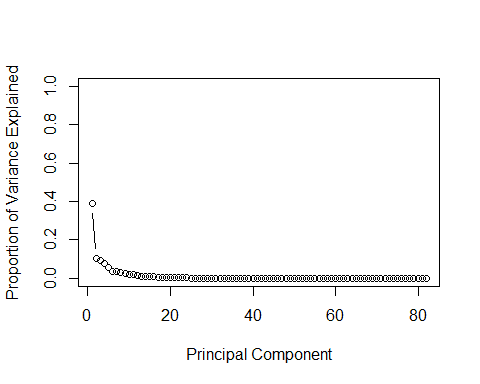


fviz\_pca\_var(res.pca,  
 col.var = "contrib", *# Color by contributions to the PC*  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE *# Avoid text overlapping*  
)

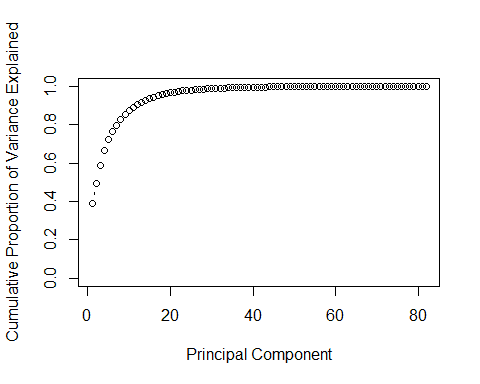
## Warning: ggrepel: 73 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps



pr\_var <- res.pca$sdev ^ 2  
pve <- pr\_var / sum(pr\_var)  
plot(pve, xlab = "Principal Component", ylab = "Proportion of Variance Explained", ylim = c(0,1), type = 'b')

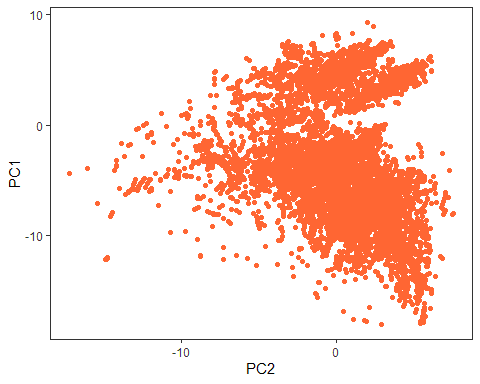


*# Cumulative PVE plot*  
plot(cumsum(pve), xlab = "Principal Component", ylab = "Cumulative Proportion of Variance Explained", ylim =c(0,1), type = 'b')

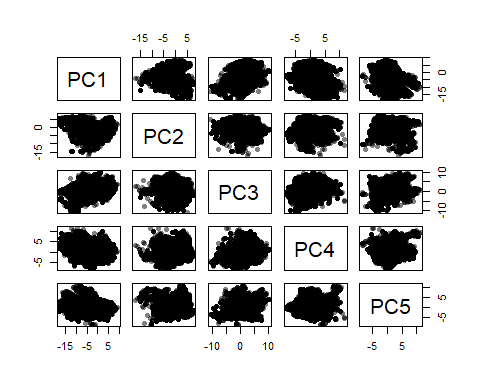


*# Rotate loadings*

library(ggplot2)  
theme\_set(theme\_bw())  
  
homes.pca <- data.frame(res.pca$x[, 1:2]) *# we only need the first two principal components*  
ggplot(homes.pca, aes(y = PC1, x = PC2)) + geom\_point(col = 'tomato2')



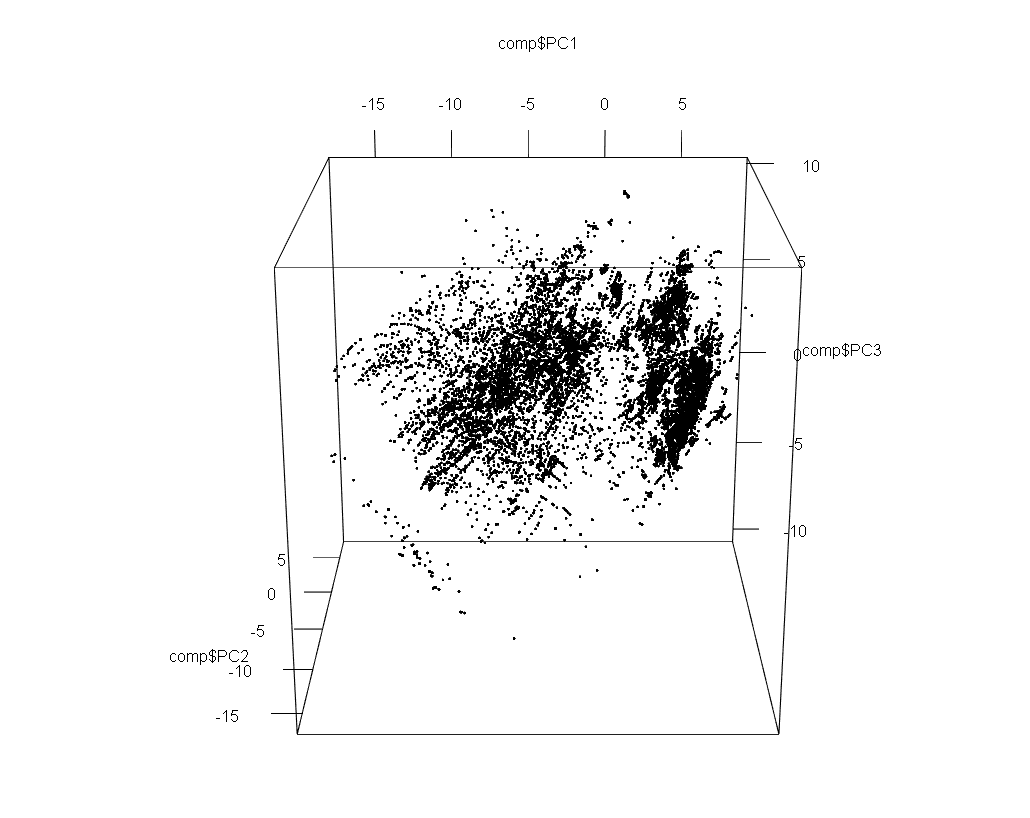
comp <- data.frame(res.pca$x[,1:5])  
*# Plot*  
plot(comp, pch=16, col=rgb(0,0,0,0.5))



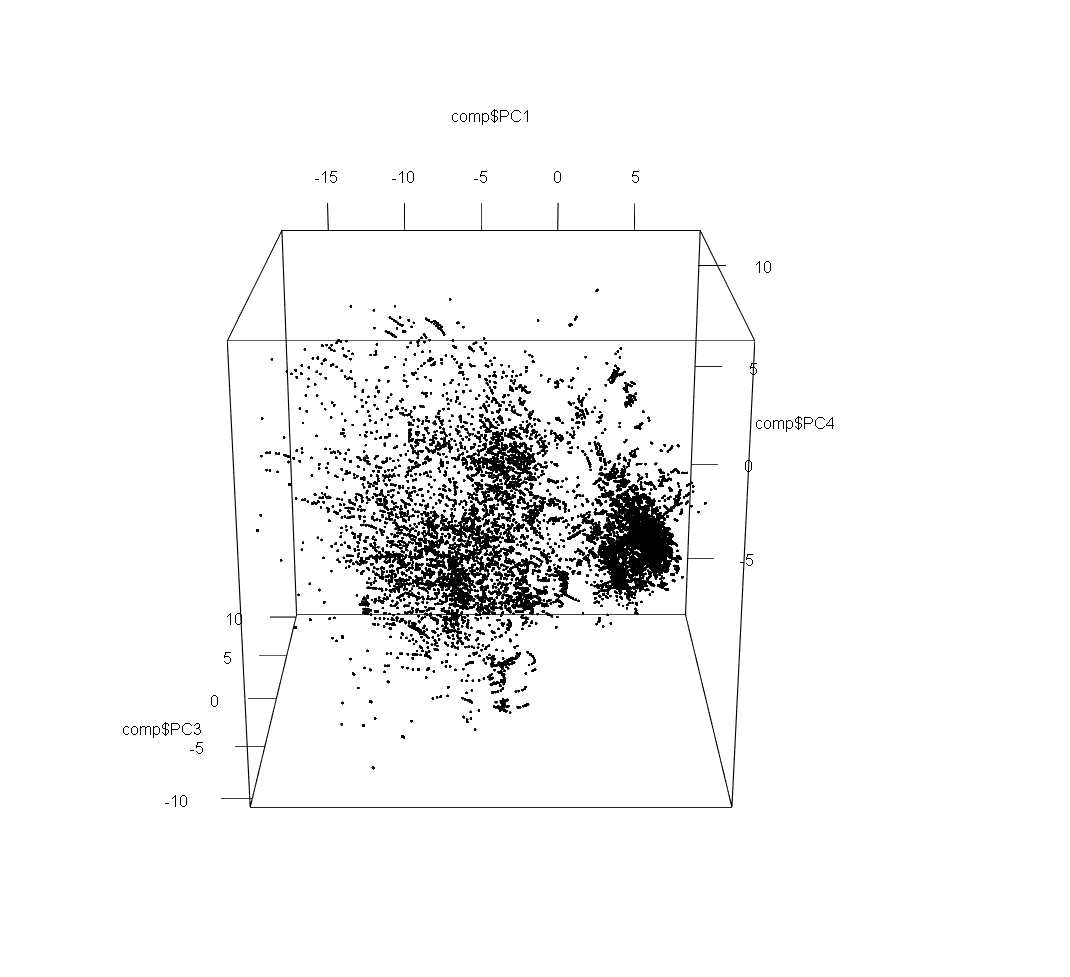
library(rgl)

## Warning: package 'rgl' was built under R version 4.1.2

*# Multi 3D plot*  
plot3d(comp$PC1, comp$PC2, comp$PC3)



plot3d(comp$PC1, comp$PC3, comp$PC4)



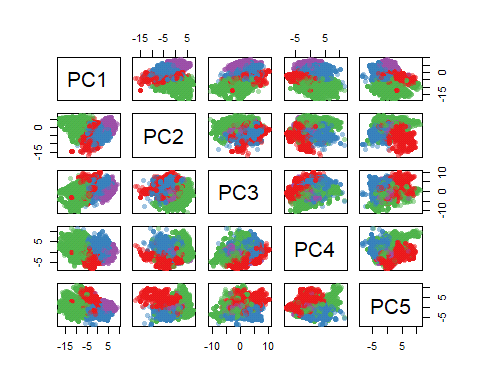
K-Means Clustering:

In k-means clustering the mean is used to calculate the centroid of each cluster. First random points are taken according to the number of k values entered. In our case we have used k=4 forming 4 clusters. Distances between random points selected as centroids and the data points will be taken, the data point nearer to the random point will be taken into the cluster of that centroid/random point. Then means are taken between the each random point and all the datapoints. This will result into a new centroid. Then in the next iteration the same procedure will repeat. And finally when no further data points will change their cluster, the algorithm will stop forming 4 cluster each having different data points.

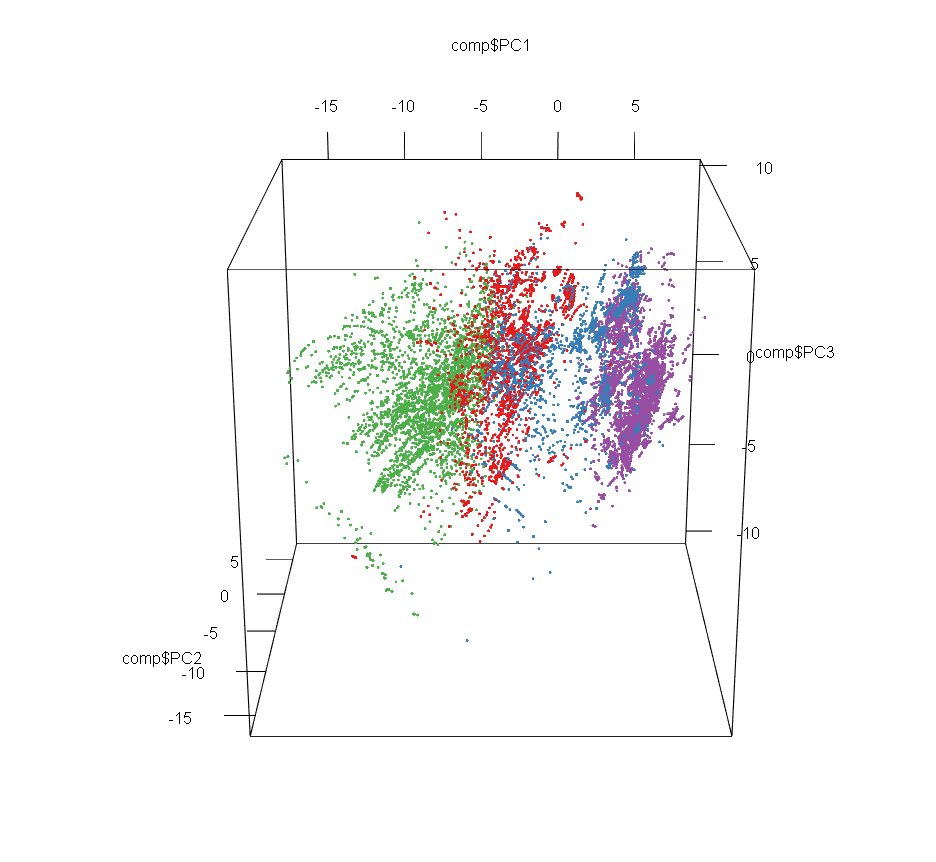
*# From scree plot elbow occurs at k = 4*  
*# Apply k-means with k=4*  
k <- kmeans(comp, 4, nstart=25, iter.max=1000)  
library(RColorBrewer)  
library(scales)

## Warning: package 'scales' was built under R version 4.1.2

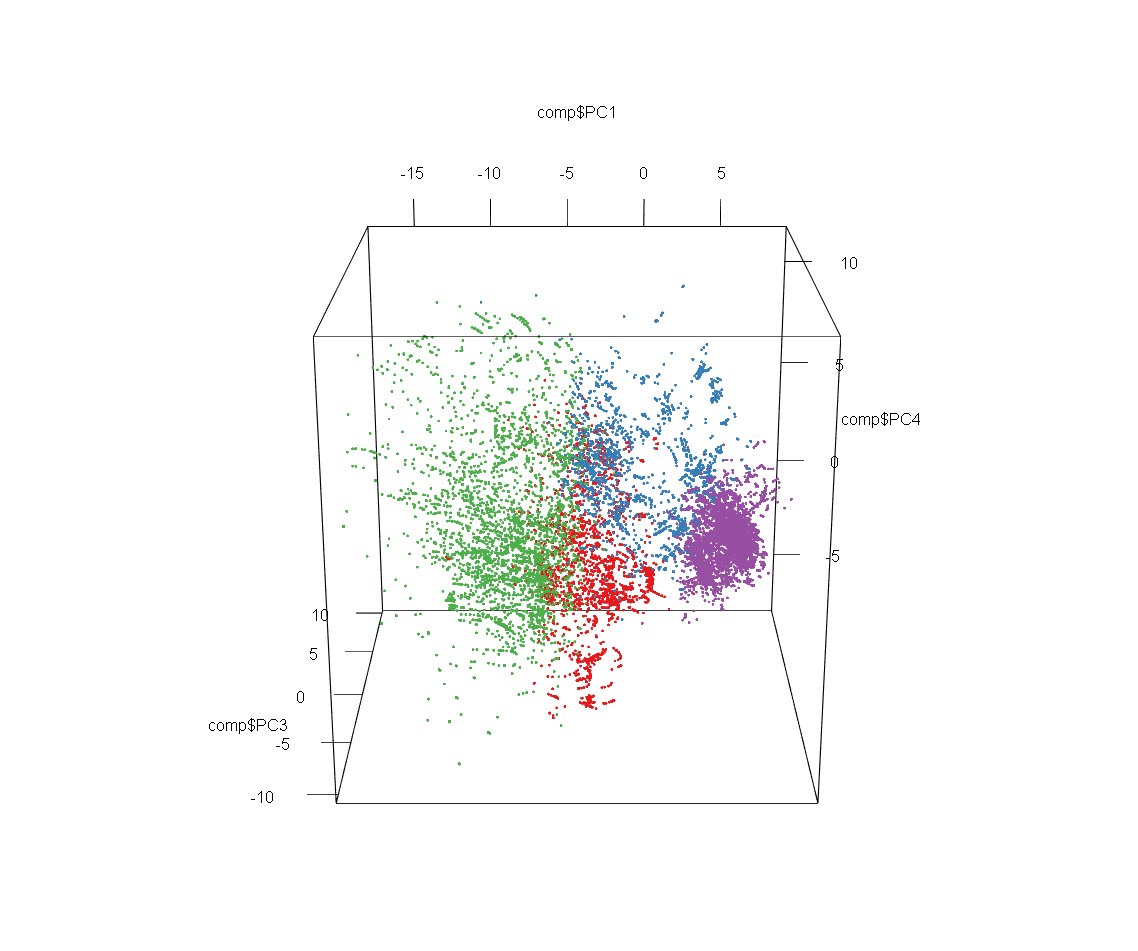
palette(alpha(brewer.pal(9,'Set1'), 0.5))  
plot(comp, col=k$clust, pch=16)



*# 3D plot*  
plot3d(comp$PC1, comp$PC2, comp$PC3, col=k$clust)



plot3d(comp$PC1, comp$PC3, comp$PC4, col=k$clust)

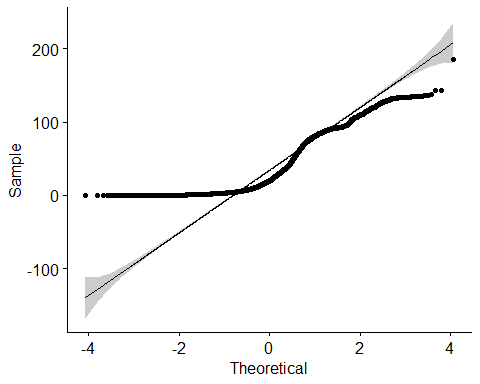


The normality test is done to check whether linear regression would be applicable or not. The test below showed that the data is not nomally skewed on qqplot. Therefore, results of multiple linear regression will not be accurate.

library(ggpubr)

## Warning: package 'ggpubr' was built under R version 4.1.2

ggqqplot(train4$critical\_temp)



Although the normality test has been failed but we are applying multiple linear regression model based on the features selected from PCA above. The linear regression model shows coefficients of each feature which represents an increment of 1 unit in that feature will cause and increase with that coefficient in the overall outcome i.e. in our case critical temperature. The R squared value shows the accuracy of the model. Our model has R=0.6344 which means only 63.44% accuracy of our model is there. The adjusted R square shows the R based on removing the extra terms that are very minimal in contributing to the model. R adjusted = 0.6338.

lmcriticaltemp = lm(critical\_temp~range\_fie + range\_atomic\_radius + wtd\_entropy\_atomic\_radius + wtd\_std\_fie   
 + wtd\_std\_atomic\_radius + entropy\_Valence + wtd\_entropy\_atomic\_mass + entropy\_fie + wtd\_entropy\_Valence  
 + number\_of\_elements + wtd\_gmean\_Density + std\_fie + entropy\_atomic\_radius + gmean\_Density   
 + std\_atomic\_mass + wtd\_gmean\_Valence + entropy\_atomic\_mass + mean\_atomic\_radius + gmean\_atomic\_radius   
 + mean\_atomic\_mass + gmean\_atomic\_mass + std\_FusionHeat + wtd\_std\_FusionHeat + wtd\_mean\_atomic\_mass   
 + range\_FusionHeat + wtd\_gmean\_atomic\_mass + mean\_fie + gmean\_fie + wtd\_entropy\_atomic\_radius   
 + entropy\_FusionHeat + mean\_Density + wtd\_range\_FusionHeat + wtd\_mean\_FusionHeat   
 + wtd\_gmean\_atomic\_radius, data = train4)  
summary(lmcriticaltemp)

##   
## Call:  
## lm(formula = critical\_temp ~ range\_fie + range\_atomic\_radius +   
## wtd\_entropy\_atomic\_radius + wtd\_std\_fie + wtd\_std\_atomic\_radius +   
## entropy\_Valence + wtd\_entropy\_atomic\_mass + entropy\_fie +   
## wtd\_entropy\_Valence + number\_of\_elements + wtd\_gmean\_Density +   
## std\_fie + entropy\_atomic\_radius + gmean\_Density + std\_atomic\_mass +   
## wtd\_gmean\_Valence + entropy\_atomic\_mass + mean\_atomic\_radius +   
## gmean\_atomic\_radius + mean\_atomic\_mass + gmean\_atomic\_mass +   
## std\_FusionHeat + wtd\_std\_FusionHeat + wtd\_mean\_atomic\_mass +   
## range\_FusionHeat + wtd\_gmean\_atomic\_mass + mean\_fie + gmean\_fie +   
## wtd\_entropy\_atomic\_radius + entropy\_FusionHeat + mean\_Density +   
## wtd\_range\_FusionHeat + wtd\_mean\_FusionHeat + wtd\_gmean\_atomic\_radius,   
## data = train4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -79.63 -13.29 0.12 13.09 162.75   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.673e+01 4.570e+00 -5.850 5.00e-09 \*\*\*  
## range\_fie 1.144e-01 5.578e-03 20.513 < 2e-16 \*\*\*  
## range\_atomic\_radius 2.608e-03 1.597e-02 0.163 0.87030   
## wtd\_entropy\_atomic\_radius 6.552e+01 3.605e+00 18.175 < 2e-16 \*\*\*  
## wtd\_std\_fie 4.413e-02 8.319e-03 5.305 1.14e-07 \*\*\*  
## wtd\_std\_atomic\_radius 3.080e-01 3.935e-02 7.828 5.19e-15 \*\*\*  
## entropy\_Valence 1.593e+02 4.295e+00 37.083 < 2e-16 \*\*\*  
## wtd\_entropy\_atomic\_mass 8.469e+00 3.062e+00 2.766 0.00568 \*\*   
## entropy\_fie -4.211e+02 1.539e+01 -27.359 < 2e-16 \*\*\*  
## wtd\_entropy\_Valence -4.740e+01 2.444e+00 -19.397 < 2e-16 \*\*\*  
## number\_of\_elements 5.755e+00 6.196e-01 9.289 < 2e-16 \*\*\*  
## wtd\_gmean\_Density 1.663e-03 2.182e-04 7.623 2.57e-14 \*\*\*  
## std\_fie -3.602e-01 1.619e-02 -22.246 < 2e-16 \*\*\*  
## entropy\_atomic\_radius 3.075e+02 1.596e+01 19.267 < 2e-16 \*\*\*  
## gmean\_Density -2.680e-03 3.049e-04 -8.789 < 2e-16 \*\*\*  
## std\_atomic\_mass 1.999e-01 2.283e-02 8.757 < 2e-16 \*\*\*  
## wtd\_gmean\_Valence -8.131e+00 2.779e-01 -29.258 < 2e-16 \*\*\*  
## entropy\_atomic\_mass -7.323e+01 4.331e+00 -16.908 < 2e-16 \*\*\*  
## mean\_atomic\_radius 2.684e+00 1.008e-01 26.617 < 2e-16 \*\*\*  
## gmean\_atomic\_radius -3.115e+00 1.100e-01 -28.314 < 2e-16 \*\*\*  
## mean\_atomic\_mass 5.459e-03 6.632e-02 0.082 0.93439   
## gmean\_atomic\_mass 9.992e-02 6.828e-02 1.463 0.14341   
## std\_FusionHeat 2.781e+00 1.930e-01 14.406 < 2e-16 \*\*\*  
## wtd\_std\_FusionHeat -1.498e+00 7.375e-02 -20.318 < 2e-16 \*\*\*  
## wtd\_mean\_atomic\_mass -1.215e+00 6.087e-02 -19.965 < 2e-16 \*\*\*  
## range\_FusionHeat -8.512e-01 7.007e-02 -12.149 < 2e-16 \*\*\*  
## wtd\_gmean\_atomic\_mass 1.084e+00 5.807e-02 18.661 < 2e-16 \*\*\*  
## mean\_fie -5.577e-01 4.192e-02 -13.305 < 2e-16 \*\*\*  
## gmean\_fie 6.236e-01 4.194e-02 14.870 < 2e-16 \*\*\*  
## entropy\_FusionHeat -6.185e+00 2.102e+00 -2.943 0.00325 \*\*   
## mean\_Density 1.631e-03 2.255e-04 7.233 4.88e-13 \*\*\*  
## wtd\_range\_FusionHeat 3.559e-01 3.940e-02 9.033 < 2e-16 \*\*\*  
## wtd\_mean\_FusionHeat -2.487e-01 3.494e-02 -7.116 1.14e-12 \*\*\*  
## wtd\_gmean\_atomic\_radius 5.227e-01 1.845e-02 28.337 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 20.73 on 21229 degrees of freedom  
## Multiple R-squared: 0.6344, Adjusted R-squared: 0.6338   
## F-statistic: 1116 on 33 and 21229 DF, p-value: < 2.2e-16

The decision tree is used to split the data based on criteria for each feature splitting. We use regression tree which is used for continuous variable. The data is splitted based on information gain methods such as c4.5,gini index or id3. This will then decide the root and the upper nodes of the tree based on information gain of a feature. Then relational operators are used to split data on each feature. This will produce a tree splitting all data based on features. We used features obtained from PCA.

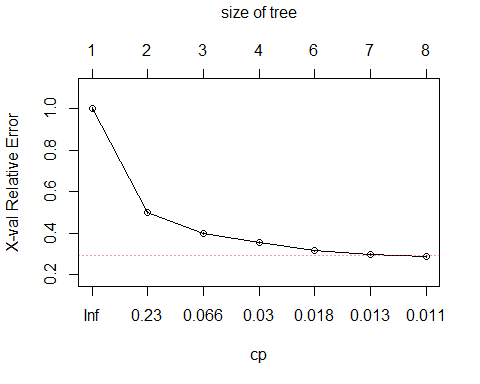
library(rpart)

## Warning: package 'rpart' was built under R version 4.1.2

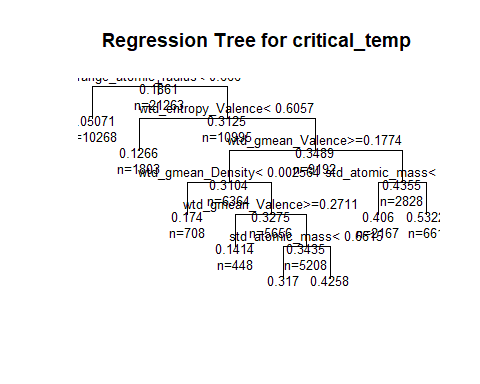
fit <- rpart(critical\_temp~range\_fie + range\_atomic\_radius + wtd\_entropy\_atomic\_radius + wtd\_std\_fie   
 + wtd\_std\_atomic\_radius + entropy\_Valence + wtd\_entropy\_atomic\_mass + entropy\_fie + wtd\_entropy\_Valence  
 + number\_of\_elements + wtd\_gmean\_Density + std\_fie + entropy\_atomic\_radius + gmean\_Density   
 + std\_atomic\_mass + wtd\_gmean\_Valence + entropy\_atomic\_mass + mean\_atomic\_radius + gmean\_atomic\_radius   
 + mean\_atomic\_mass + gmean\_atomic\_mass + std\_FusionHeat + wtd\_std\_FusionHeat + wtd\_mean\_atomic\_mass   
 + range\_FusionHeat + wtd\_gmean\_atomic\_mass + mean\_fie + gmean\_fie + wtd\_entropy\_atomic\_radius   
 + entropy\_FusionHeat + mean\_Density + wtd\_range\_FusionHeat + wtd\_mean\_FusionHeat   
 + wtd\_gmean\_atomic\_radius,  
 method="anova", data=newd)  
printcp(fit)

##   
## Regression tree:  
## rpart(formula = critical\_temp ~ range\_fie + range\_atomic\_radius +   
## wtd\_entropy\_atomic\_radius + wtd\_std\_fie + wtd\_std\_atomic\_radius +   
## entropy\_Valence + wtd\_entropy\_atomic\_mass + entropy\_fie +   
## wtd\_entropy\_Valence + number\_of\_elements + wtd\_gmean\_Density +   
## std\_fie + entropy\_atomic\_radius + gmean\_Density + std\_atomic\_mass +   
## wtd\_gmean\_Valence + entropy\_atomic\_mass + mean\_atomic\_radius +   
## gmean\_atomic\_radius + mean\_atomic\_mass + gmean\_atomic\_mass +   
## std\_FusionHeat + wtd\_std\_FusionHeat + wtd\_mean\_atomic\_mass +   
## range\_FusionHeat + wtd\_gmean\_atomic\_mass + mean\_fie + gmean\_fie +   
## wtd\_entropy\_atomic\_radius + entropy\_FusionHeat + mean\_Density +   
## wtd\_range\_FusionHeat + wtd\_mean\_FusionHeat + wtd\_gmean\_atomic\_radius,   
## data = newd, method = "anova")  
##   
## Variables actually used in tree construction:  
## [1] range\_atomic\_radius std\_atomic\_mass wtd\_entropy\_Valence  
## [4] wtd\_gmean\_Density wtd\_gmean\_Valence   
##   
## Root node error: 728.94/21263 = 0.034282  
##   
## n= 21263   
##   
## CP nsplit rel error xerror xstd  
## 1 0.499047 0 1.00000 1.00007 0.0083088  
## 2 0.102224 1 0.50095 0.50109 0.0054271  
## 3 0.042031 2 0.39873 0.39930 0.0051075  
## 4 0.021732 3 0.35670 0.35752 0.0046578  
## 5 0.015561 5 0.31323 0.31543 0.0044826  
## 6 0.011056 6 0.29767 0.29918 0.0043998  
## 7 0.010000 7 0.28662 0.28769 0.0042865

plotcp(fit)

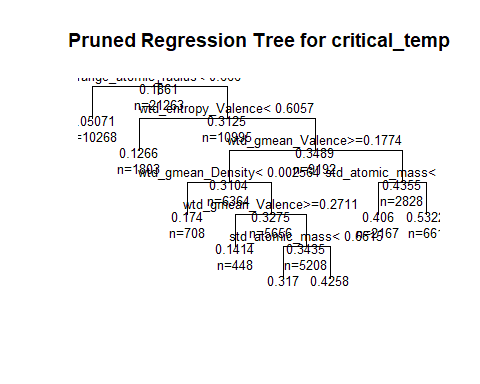


plot(fit, uniform=TRUE,  
 main="Regression Tree for critical\_temp ")  
text(fit, use.n=TRUE, all=TRUE, cex=.8)



The tree is pruned for irrelevant features which do not provide further splitting. But in our case both trees are same i.e with pruning or without pruning.

*# prune the tree*  
pfit<- prune(fit, cp=0.011)   
plot(pfit, uniform=TRUE,  
 main="Pruned Regression Tree for critical\_temp")  
text(pfit, use.n=TRUE, all=TRUE, cex=.8)

-

**Conclusion:**

Our critical temperature depends upon range\_fie , range\_atomic\_radius , wtd\_entropy\_atomic\_radius , wtd\_std\_fie , wtd\_std\_atomic\_radius , entropy\_Valence wtd\_entropy\_atomic\_mass , entropy\_fie , wtd\_entropy\_Valence , number\_of\_elements , wtd\_gmean\_Density , std\_fie , entropy\_atomic\_radius , gmean\_Density , std\_atomic\_mass , wtd\_gmean\_Valence, entropy\_atomic\_mass , mean\_atomic\_radius , gmean\_atomic\_radius , mean\_atomic\_mass , gmean\_atomic\_mass , std\_FusionHeat, wtd\_std\_FusionHeat , wtd\_mean\_atomic\_mass , range\_FusionHeat , wtd\_gmean\_atomic\_mass , mean\_fie , gmean\_fie, wtd\_entropy\_atomic\_radius , entropy\_FusionHeat , mean\_Density , wtd\_range\_FusionHeat , wtd\_mean\_FusionHeat , wtd\_gmean\_atomic\_radius. These factors cause superconductors to behave on critical temperature.