Cluster Analysis - Employee Satisfaction Survey

Joseph O'Malley

11 June, 2019

Contents

Data Overview

This data contains 1,470 records with information about the employees and feedback pertaining to their job satisfaction. The data contains 28 feature columns with information such as Department, Job, Education, income and performance rating. While this data does not have a lot of records, the information is rich with information.

```
#The dataset of employee satisfaction
emp_sat_data <- read.csv(filepath,sep=",",header = T)
```

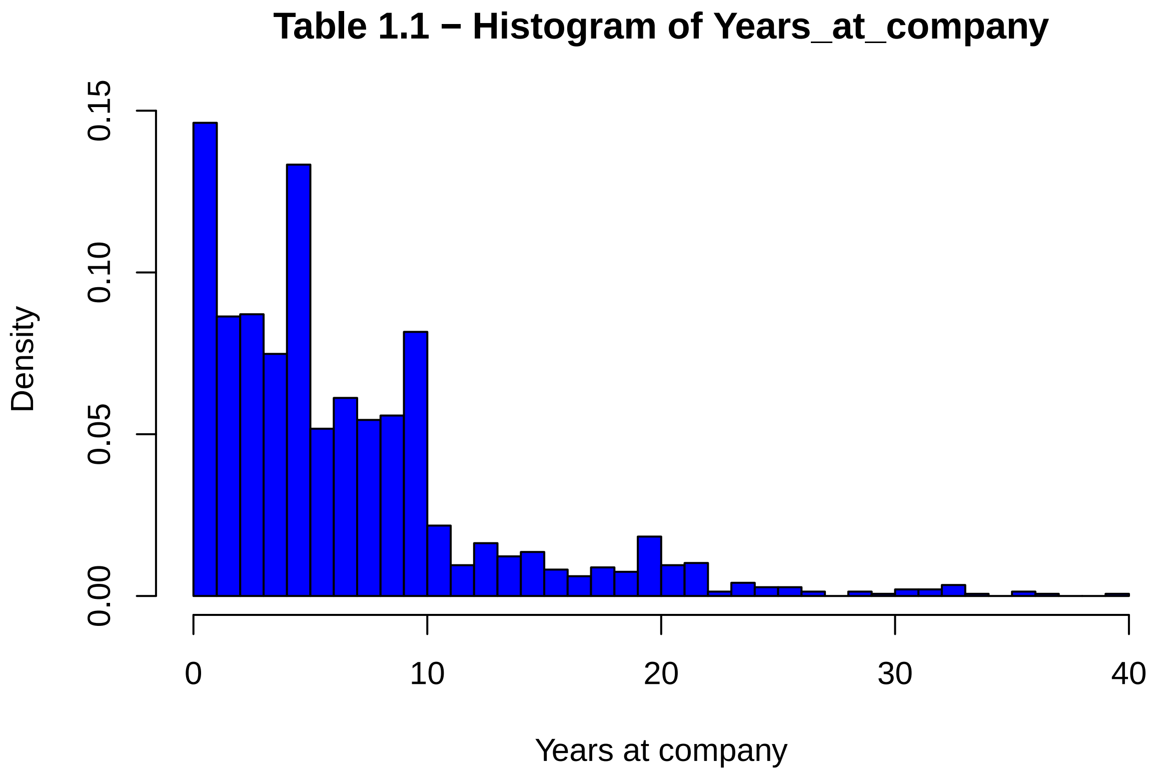
Exploratory Data Analysis

```
require(data.table)
require(cluster)
require(NbClust)
require(factoextra)

#Check basic info
names(emp_sat_data)
## return summary statistics for numeric, counts for catagorical (by column)
summary(emp_sat_data)
## return columns: null counts, datatypes
```

```
str(emp_sat_data)
##return
dim(emp_sat_data)
```

plot histogram distributions

```
## histogram of continuous columns
hist(emp_sat_data$Years.At.Company, breaks = 40, probability = TRUE, col = 'blue', xlab="Years at compan
```



```
hist(emp_sat_data$Salary.Hike...., breaks = 12, probability = TRUE, col = 'beige', xlab="Salary Increas
```

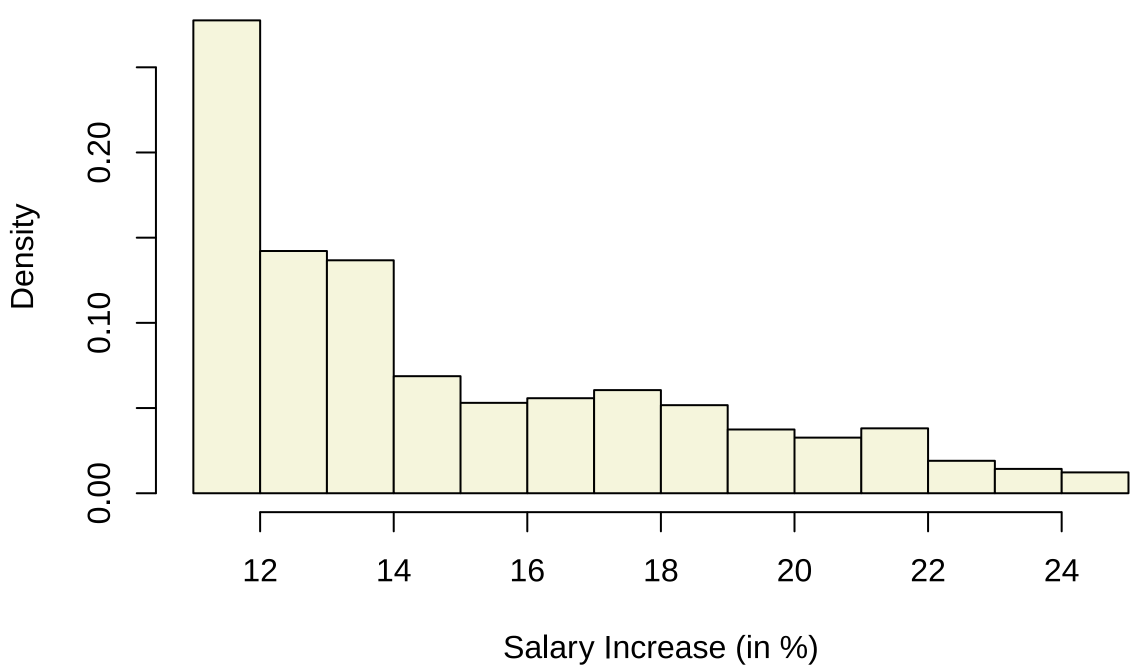


Table 1.2 − Histogram of Salary_increase

```
## look at catagorical columns
### make data.table for ease
require(data.table)
emp_sat_dt <- as.data.table(emp_sat_data)

## subset data to remove unwanted columns
drops <- c("X","X.1","X.2","Over.18")
emp_sat_data_sub <- emp_sat_data[ , !(names(emp_sat_data) %in% drops)]
```

EDA findings

My initial exploratory analysis showed a good distribution of continuous variables. Only 25 percent of the company had less than 6 years total work experience. I took a look at a histogram of years worked at the company (see Table 1.1) to find that most the most employees had been with the company 1 or 5 years. Salary Hike was another column that was interesting, with employees receiving a minimum raise of 11 percent (see Table 1.2). By looking at the structure & summary, we see that the continuous columns most had only a few levels (i.e. – bad, best, better, good) presumably from a multiple choice survey. Males make up slightly over 60 percent of this company and there are no employees under the age of 18 so we can drop that column before we start or cluster analysis.

Cluster Analysis

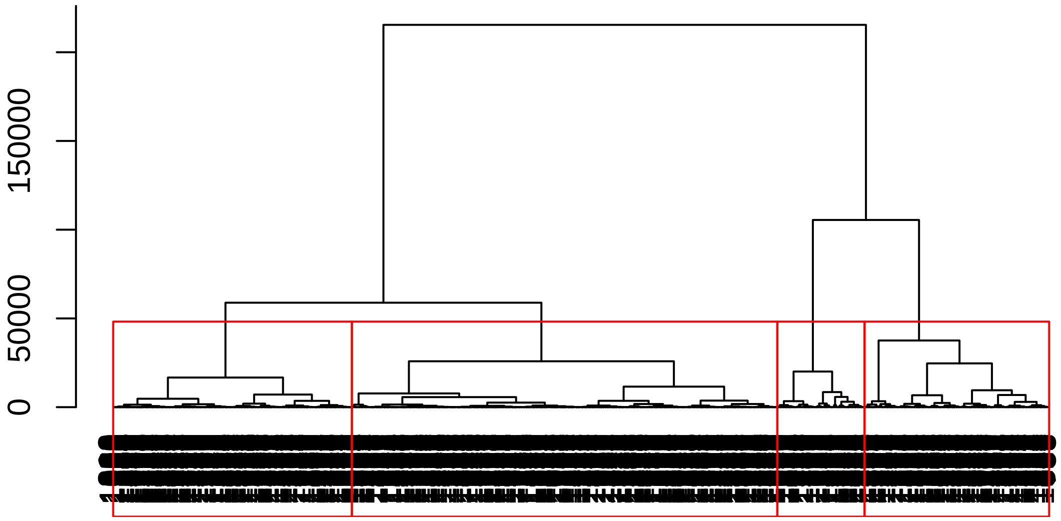
scale data:

```
## 1) subset numeric columns
num_cols <- c("Will.consider.switch","Age","Distance.From.Home..kms.","Job.Level","Monthly.Income..USD.
              ,"Salary.Hike....","Stock.Option.Level","No..of.Companies.Worked","Total.Working.Years"
              ,"Years.At.Company","Years.In.Current.Role","Years.Since.Last.Promotion","Years.With.Curr
              ,"Training.Times.Last.Year")
emp_sat_data_numeric <- emp_sat_data[ , (names(emp_sat_data) %in% num_cols)]

## 2) convert all to proper datatype
for(i in 1:ncol(emp_sat_data_numeric)) emp_sat_data_numeric[[i]] <- as.numeric(emp_sat_data_numeric[[i]
str(emp_sat_data_numeric)

## 3) seperate target column
cols = c("Will.consider.switch")
scale(emp_sat_data_numeric[ , !(names(emp_sat_data_numeric) %in% cols)])

# z-score normalize numeric data
emp_sat_data_numeric_z <- emp_sat_data_numeric[ , !(names(emp_sat_data_numeric) %in% cols)]

## check structure of scaled data
str(emp_sat_data_numeric_z)
```

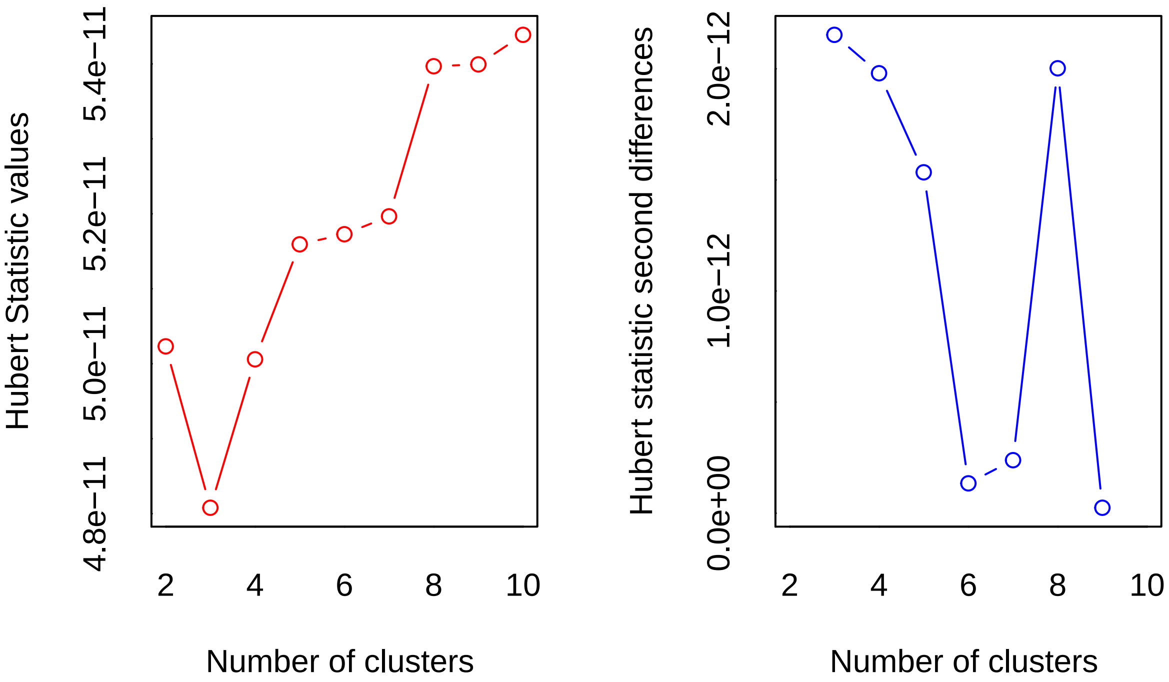
Hierarchical Clustering

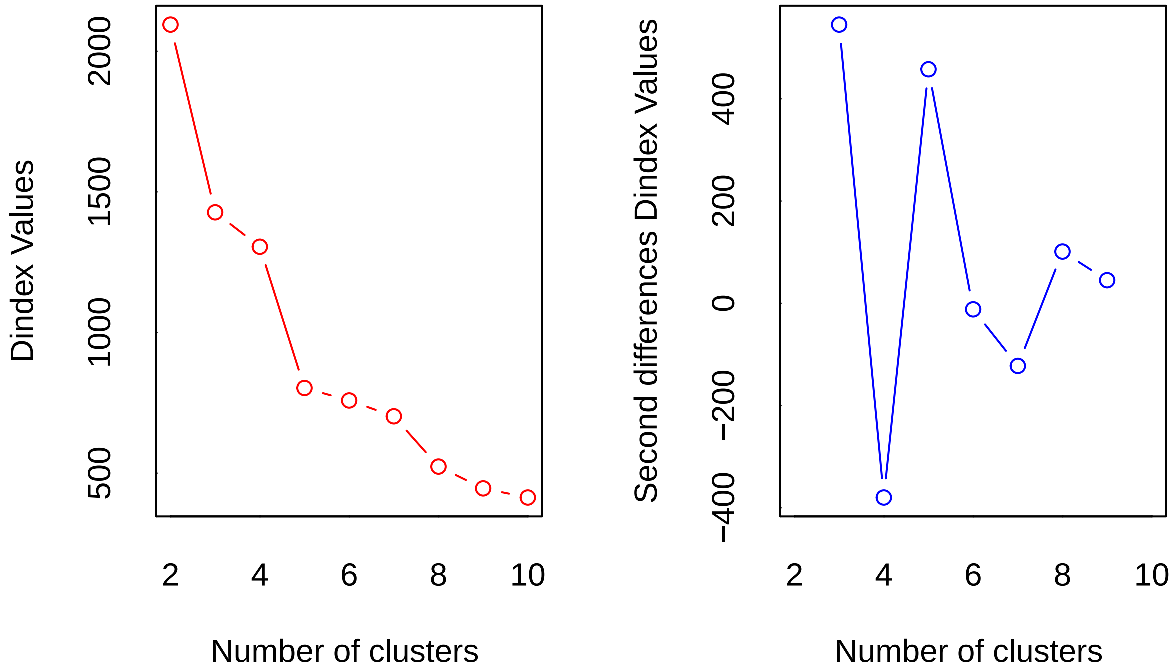
visualize dendogram:

```
#create a matrix with Euclidean distances for all observations
dis.matrix <- dist(emp_sat_data_numeric_z)
#convert the above into a matrix object
dis.matrix_view <- as.matrix(dis.matrix)

# optional - print closest by Euclidean distances
# sort(dis.matrix_view[1,])
# sort(dis.matrix_view[10,])

set.seed(123)
### create dendogram - using ward.D2 linkage method
crash_hiearchical_clusters <- hclust(dis.matrix, method="ward.D2")
# plot dendogram
plot(crash_hiearchical_clusters, main = "Dendrogram: Ward's Method", hang=-1, ann=FALSE)
##cut dendogram
rect.hclust(crash_hiearchical_clusters, k=4, border="red")
```

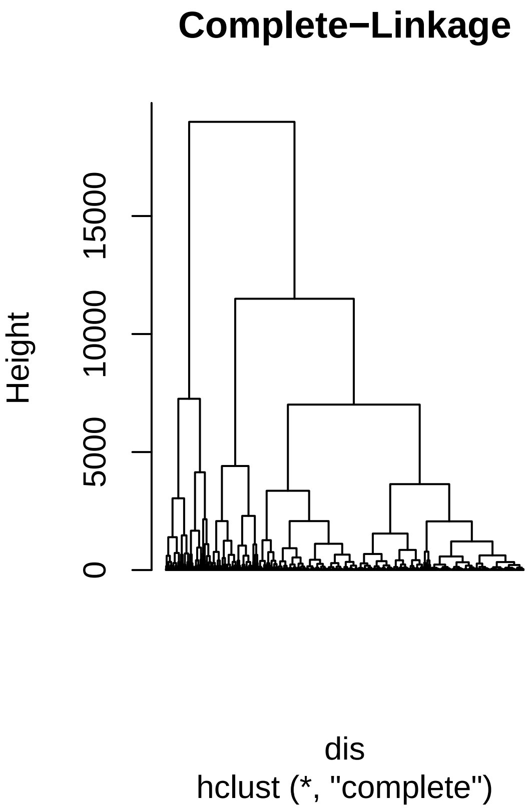


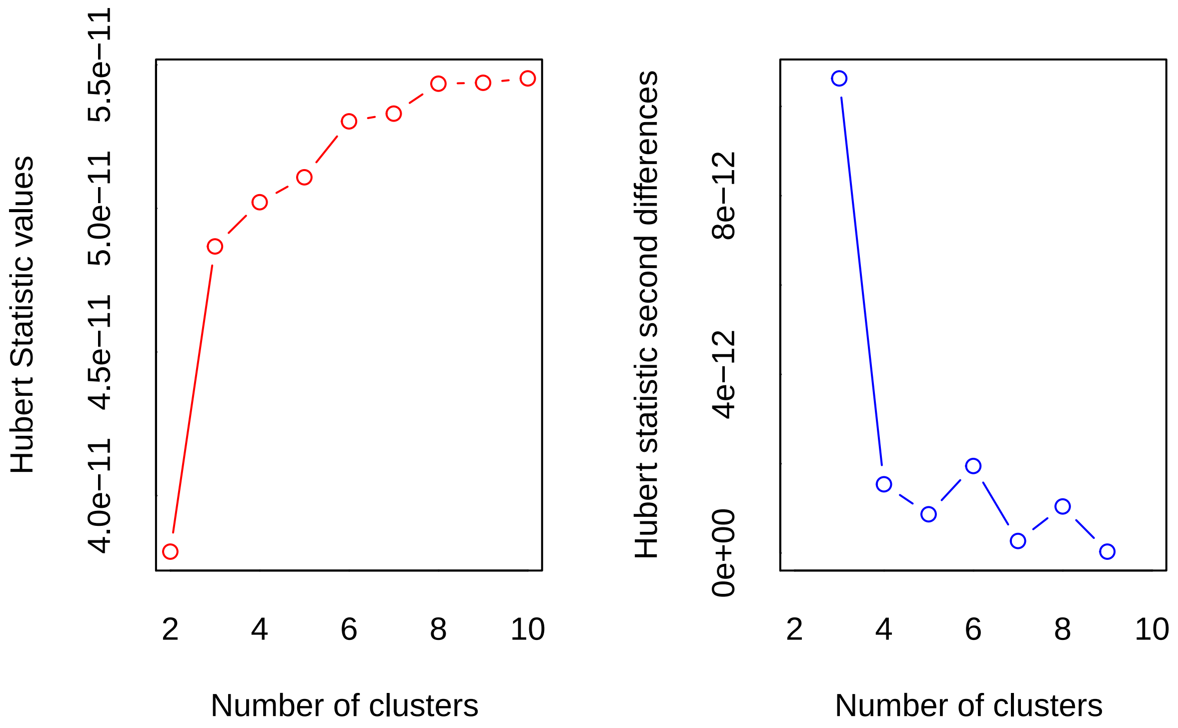
```
## alias scaled data
w2 <- emp_sat_data_numeric_z

#Find the best number of clusters using NbClust
numComplete <- NbClust(w2, distance="euclidean", min.nc=2, max.nc=10,
                       method="complete", index="all")
```

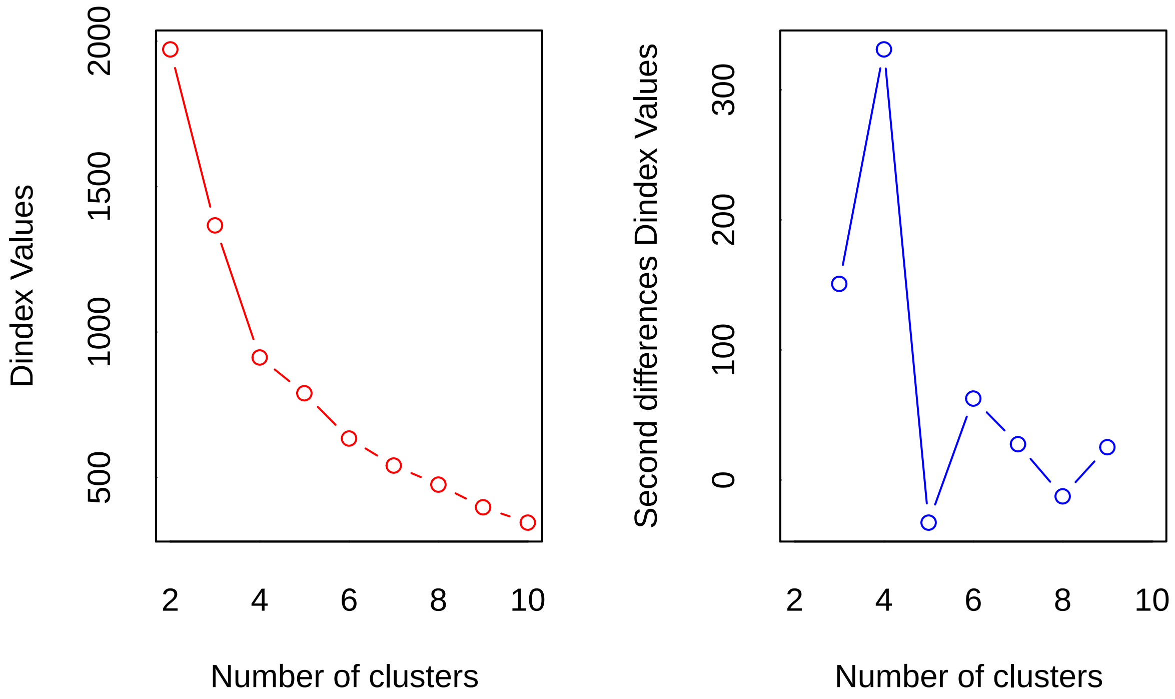

Hubert Statistic values
4.8e−11
5.0e−11
5.2e−11
5.4e−11
2
4
6
8
10
Number of clusters
Hubert statistic second differences
0.0e+00
1.0e−12
2.0e−12
2
4
6
8
10
Number of clusters



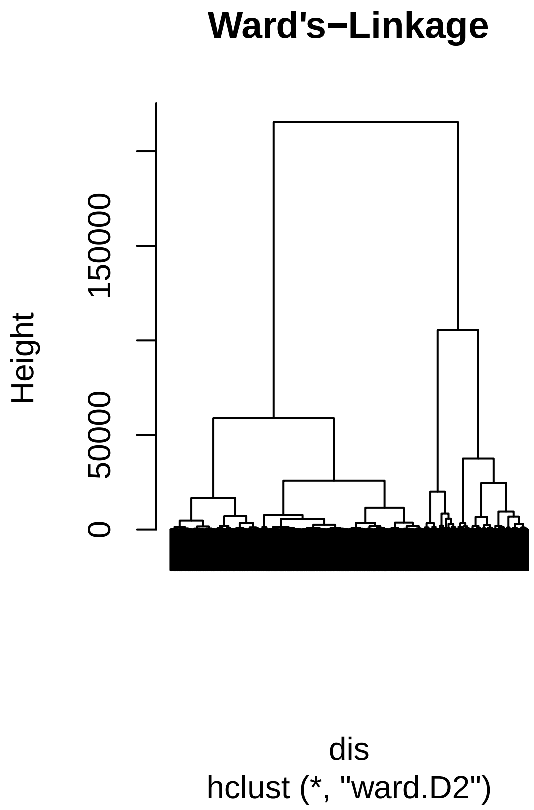
```
#The two figures "Dindex value" (elbow rule), and the second differences Dindex Values (highest point r
names(numComplete)

#See all the indices
numComplete$Best.nc

# calculate distance matrix
dis = dist(w2, method="euclidean")

# create hierarchical cluster
hc = hclust(dis, method="complete")

## try "complete linkage option"
plot(hc, hang=-1,labels=FALSE, main="Complete-Linkage")

#We cut the tree into 5 clusters
comp5 <- cutree(hc, 5)
table(comp5)

# Ward's linkage
library(NbClust)
NbClust(w2, distance="euclidean", min.nc=2, max.nc=10,
        method="ward.D2", index="all")
```


Complete−Linkage
Height
0
5000
10000
15000
dis
hclust (*, "complete")


Hubert Statistic values
4.0e−11
4.5e−11
5.0e−11
5.5e−11
2
4
6
8
10
Number of clusters
Hubert statistic second differences
0e+00
4e−12
8e−12
2
4
6
8
10
Number of clusters



```
#"* According to the majority rule, the best number of clusters is  3
hcWard <- hclust(dis, method="ward.D2")

plot(hcWard, labels=FALSE, main="Ward's-Linkage")

ward3 <- cutree(hcWard, 3)

table(ward3)

#Check the cross table of the two solutions
table(comp5, ward3)

## compare using the aggregate() summarizing on a statistic such as the mean or median
## We check group means by each cluster
aggregate(w2,list(comp5),mean)

par(mfrow=c(1,2))
```



```
w3 <- w2

#merge cluster ID to the original dataset
w4<-as.data.frame(cbind(emp_sat_data, comp5, ward3))
##check structure of new merged tabele
#str(w4)

## check dimensions of new df
dim(w4)

table(w4$comp5,w4$ward3)
```

crosstab clusters

```
## crosstab distribution of the clusters to Will.consider.switch
## analyze switches
prop.table(table(w4$Will.consider.switch))
```

```
##
##        No       Yes
## 0.8387755 0.1612245
```

```
t(table(w4$Will.consider.switch,w4$comp5))
```

```
##
##       No Yes
##    1 366  44
##    2 528 153
##    3 160  30
##    4  86   5
##    5  93   5
```

```
t(table(w4$Will.consider.switch,w4$ward3))
```

```
##
##       No Yes
##    1 853 190
##    2 248  42
##    3 132   5
```

```
## analyze performance ratings
prop.table(table(w4$Performance.Rating))
```

```
##
##    Excellent Outstanding
##    0.8462585   0.1537415
```

```
t(table(w4$Performance.Rating,w4$comp5))
```

```
##
##      Excellent Outstanding
##    1       353          57
##    2       570         111
##    3       159          31
##    4        77          14
##    5        85          13
```

```
t(table(w4$Performance.Rating,w4$ward3))
```

```
##
##      Excellent Outstanding
##    1       882         161
##    2       246          44
##    3       116          21
```

verify clusters

```
#To check the quality of tree-cutting, we can also use the correlation between cophenetic distance and

#First we calculate the distance and also conduct clustering
dis = dist(w2, method="euclidean")
hc = hclust(dis, method="complete")

# Compute cophentic distance
```

```
res.coph <- cophenetic(hc)

# Correlation between cophenetic distance and the original distance. The higher, the better.
cor(dis, res.coph)
```

```
## [1] 0.888408
```

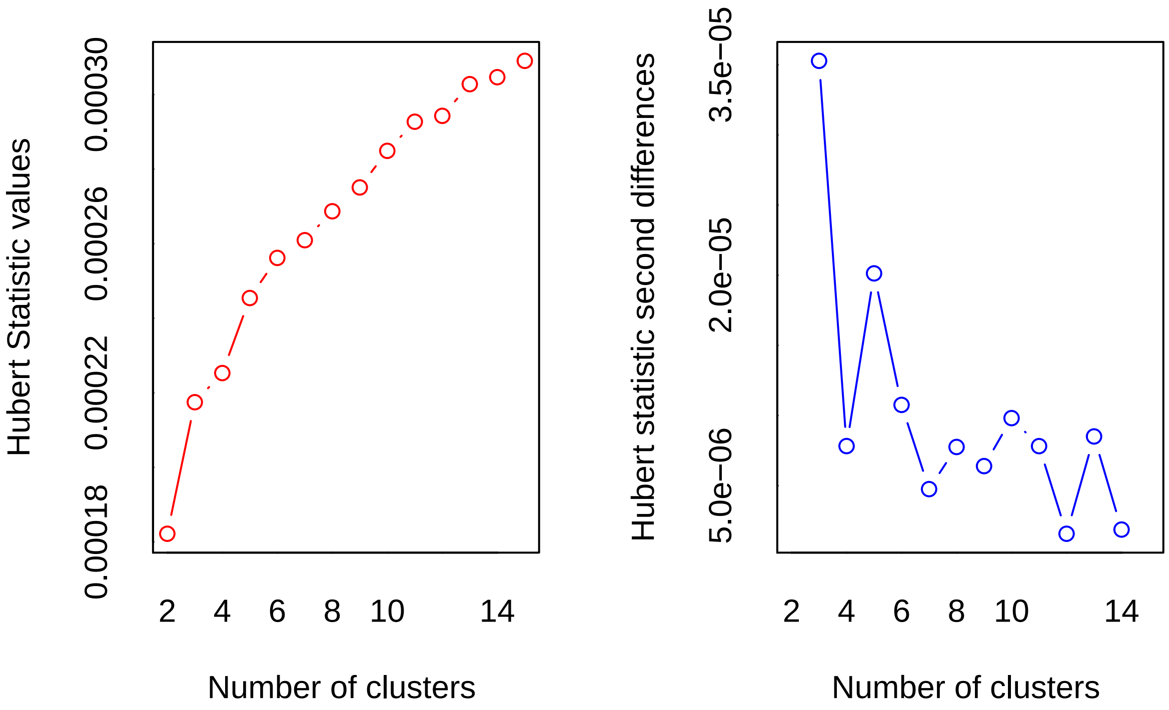
```
#additional linkage options: "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median"
```

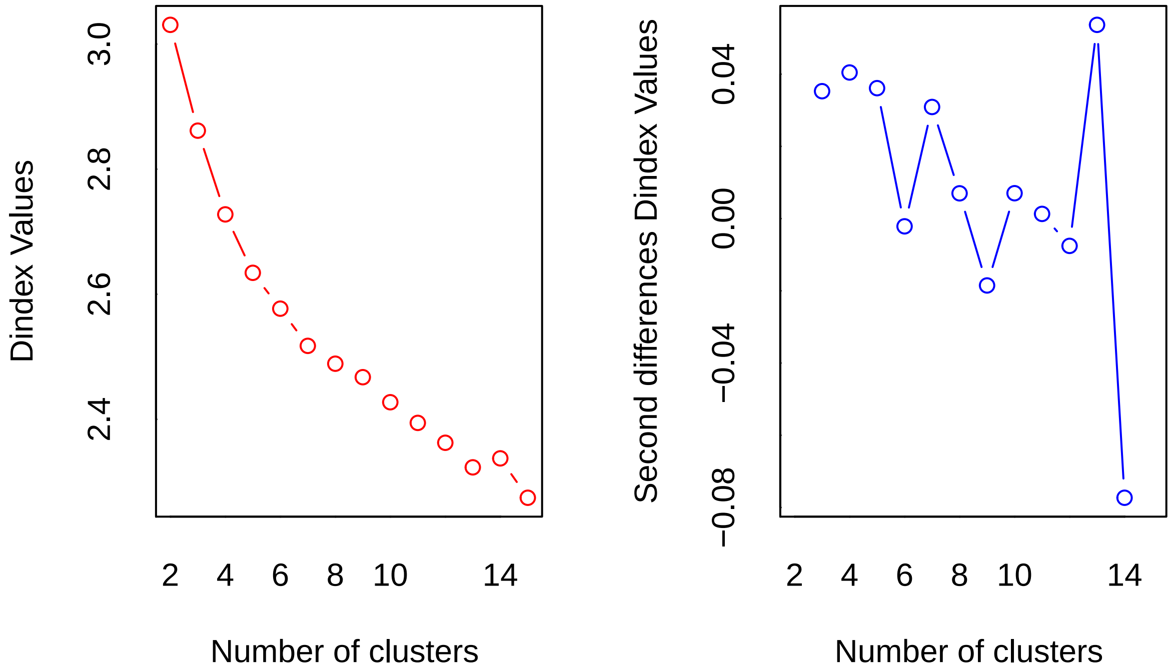
Hierarhical Clustering - results

I started my cluster analysis by first examining just the numeric columns (13 total – Age, Distance from home, Job level, Monthly Income, etc.) using hierarchical clustering with "Ward's linkage". I first had to normalize these data so they could work with the K-means methods (and would not favor continuous variable with more cut points). After looking at the dendogram for "Ward's linkage", I could tell there were 3-5 well defined groupings. I then tried "complete linkage" and found a majority vote at 5. The elbow chart also looked optimal around 4-5 clusters. I then compared the number of observation in each respective group of the two clusters and found that clusters 1 and 3 of the "Ward's" linkage were in agreement with groups 1 and 4. Group 2 was dispersed amongst several "complete linkage" clusters, but it could be either more granular information or noise. After joining the cluster assignments back to the full set, you could see a clear relationship between the cluster assignments and features that pertained to age, number of years worked, number of years with the company, etc. This makes sense given that nearly all of these columns could be tied to tenure.

K-Means clustering

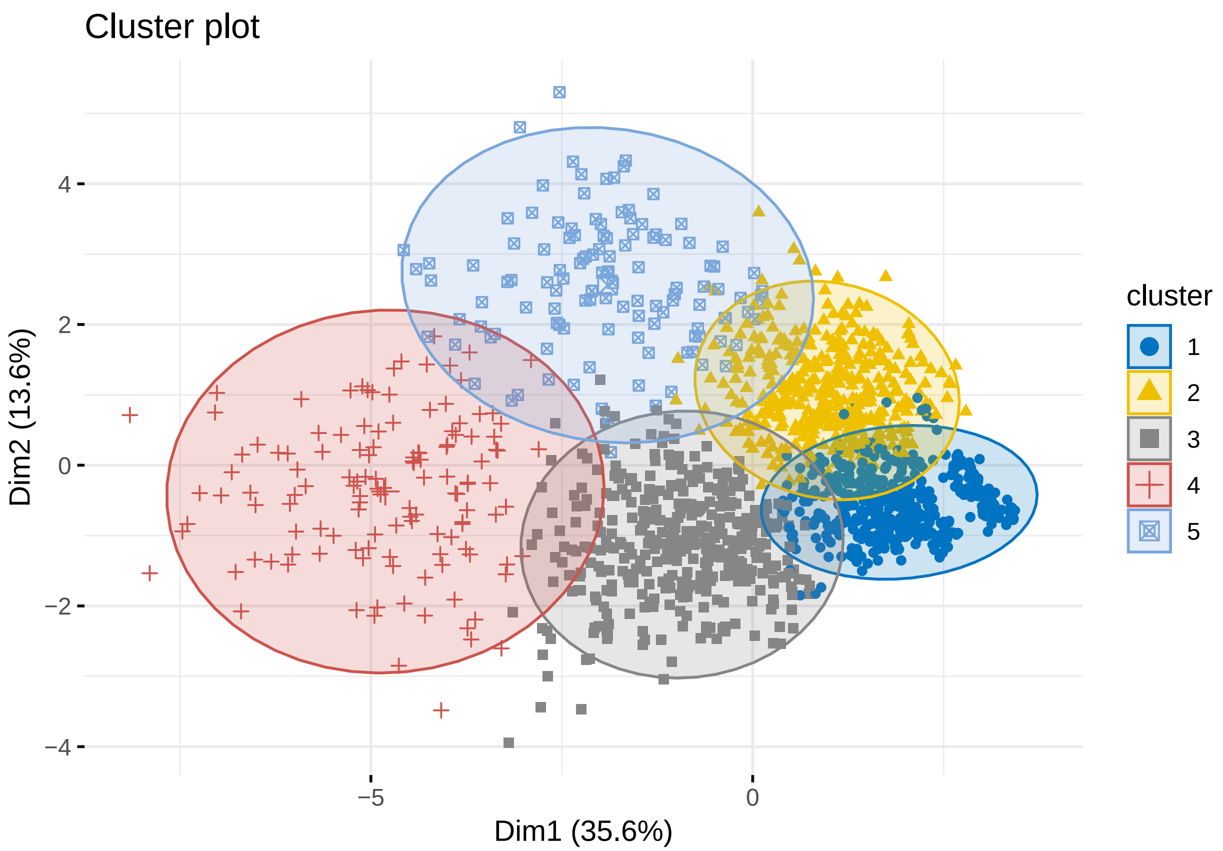
```
#Remove the quality varaible
emp_sat_data_numeric <- emp_sat_data[ , (names(emp_sat_data) %in% num_cols)]
cols = c("Will.consider.switch")
w1 <- emp_sat_data_numeric[ , !(names(emp_sat_data_numeric) %in% cols)]
str(w1)

#Standardize all the input variables
w2<- scale(w1)

# find the best number of clusters using NbClust
NbClust(w2, min.nc=2, max.nc=15, method="kmeans")
```


Hubert Statistic values
0.00018
0.00022
0.00026
0.00030
2
4
6
8
10
14
Number of clusters
Hubert statistic second differences
5.0e−06
2.0e−05
3.5e−05
2
4
6
8
10
14
Number of clusters

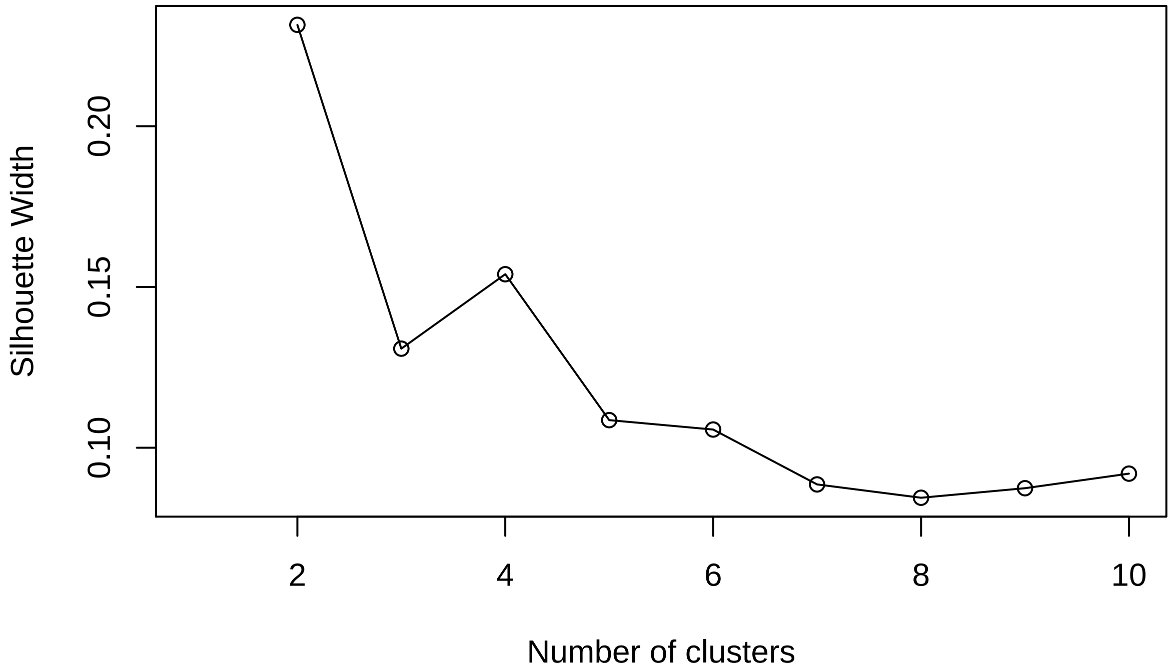


Based on the Hubert and Dindex, 4-10 clusters appears to be the optimal range.

```r
set.seed(1234)
km<- kmeans(w2,5,nstart=25)
#It is recommended to have a relative large nstart number such as 25 or 50 for initial random centroids
table(km$cluster)

#cluster centers for each variable in each cluster
km$centers

#merge cluster ID to the original dataset
w3<-as.data.frame(cbind(w1, km$cluster))
colnames(w3)[]<-"clusters"
str(w3)
#Group means by each cluster. Then you can do many further analysis based on groups
aggregate(w3[], by=list(w3$clusters), FUN=mean)

#visually show clusters
library(factoextra)
set.seed(1234)
km.res <- eclust(w2, "kmeans", k = 5, nstart = 25, graph = FALSE)

table(km.res$cluster)
fviz_cluster(km.res, geom = "point", ellipse.type = "norm",
             palette = "jco", ggtheme = theme_minimal())
```



```
# optional - Compare clustering algorithms using "clvalid"
# install.packages("clValid")
# library(clValid)

#clmethods <- c("hierarchical","kmeans","pam")
#intern <- clValid(w2, nClust = 2:6, clMethods = clmethods, validation = "internal")
# Summary, it shows that Hierarchical" may be the better choice.
#summary(intern)


library(cluster)
#We use daisy function to calcuate distance for mixed dataset
disMat = daisy(w2, metric="gower")

#You can find the best number of clusters using Silhouette approach: a measure to estimate the dissimil
sil_width <- c(NA)
for(i in 2:10){
  pam_fit <- pam(disMat, diss = TRUE, k = i)
  sil_width[i] <- pam_fit$silinfo$avg.width
}

plot(1:10, sil_width,
     xlab = "Number of clusters",
     ylab = "Silhouette Width")
lines(1:10, sil_width)
```



```
set.seed(123)
pamFit = pam(disMat, k=3)
table(pamFit$clustering)


#install.packages("compareGroups", "haplo.stats", "mvtnorm", "zoo")
#install.packages("rlang", dep=TRUE)
#R 3.5.3 or newer version (use when number of clusters less than or equal to 5.
require(multcomp)
library(haplo.stats)
require(ggplot2)
library(compareGroups)

w4<-as.data.frame(cbind(w2, pamFit$clustering))
colnames(w4)[14]<-"clusters"

group <- compareGroups(clusters~., data=w4)
#We can check the group means and standard deviations for each cluster on each variable
clustab = createTable(group, digits=3, show.p.overall = FALSE)
clustab


#Also we can visually check the variations of the clusters regarding the means
str(w4)

#For numeric variables
```

```
p1<-aggregate(w4[-c(9,12)], by=list(w4$cluster), mean)
p1
```

Cluster Analysis - results

I then used K-means clustering to analyze the employees. I tried several different cluster sizes, but settled on using 6 (Appendix – B4). As you can see from the visualization, which attempts to plot multidimensional data onto a 2-d surface, that 3 of the groupings appear to stand out (1,2 & 6). After acquiring the cluster assignments for both methods, I compared the 2 and found that k-means had higher connectivity as the number of clusters increased. Hierarchical had higher "Dunn" and "Silouhette" metrics throughout. When comparing the cluster assignments to the target variables (performance rating and will consider switch) a majority were in within a ~3 percent of the prior distribution for "Will you consider switch", which had a prior distribution of 16.7 percent positive cases. The hierarchical cluster 3 (with "Ward's") had a much lower rate with only 3.7 percent. Performance Rating also had in unbalanced target of only 15.3 percent "Outstanding" ratings. Cluster 2 from the "complete linkage" hierarchical cluster had nearly double with 28.1 percent "Outstanding" ratings.

Reccomendations

This analysis should inform managers which employees may need special attention to meet their needs and hopefully help to retain them. Specifically, the finding from cluster 3 rates of "Will you consider switch" should be a driving factor in targeting that ~10 percent of employees for review and see what is common among them that may or may not be addressable.

“‘