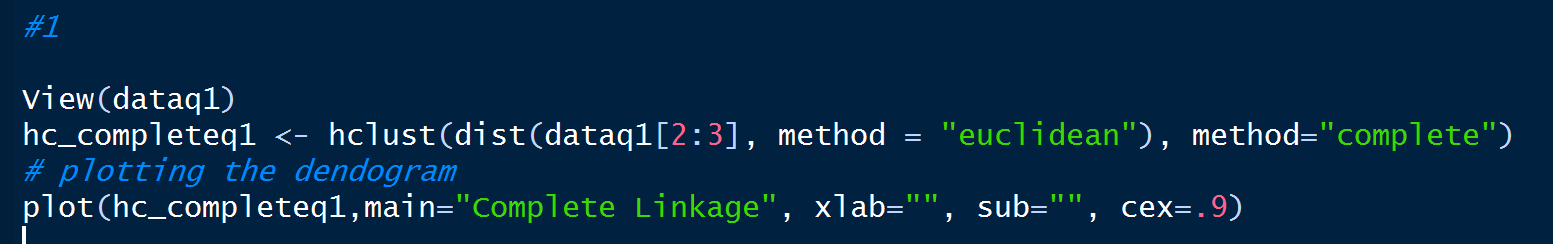


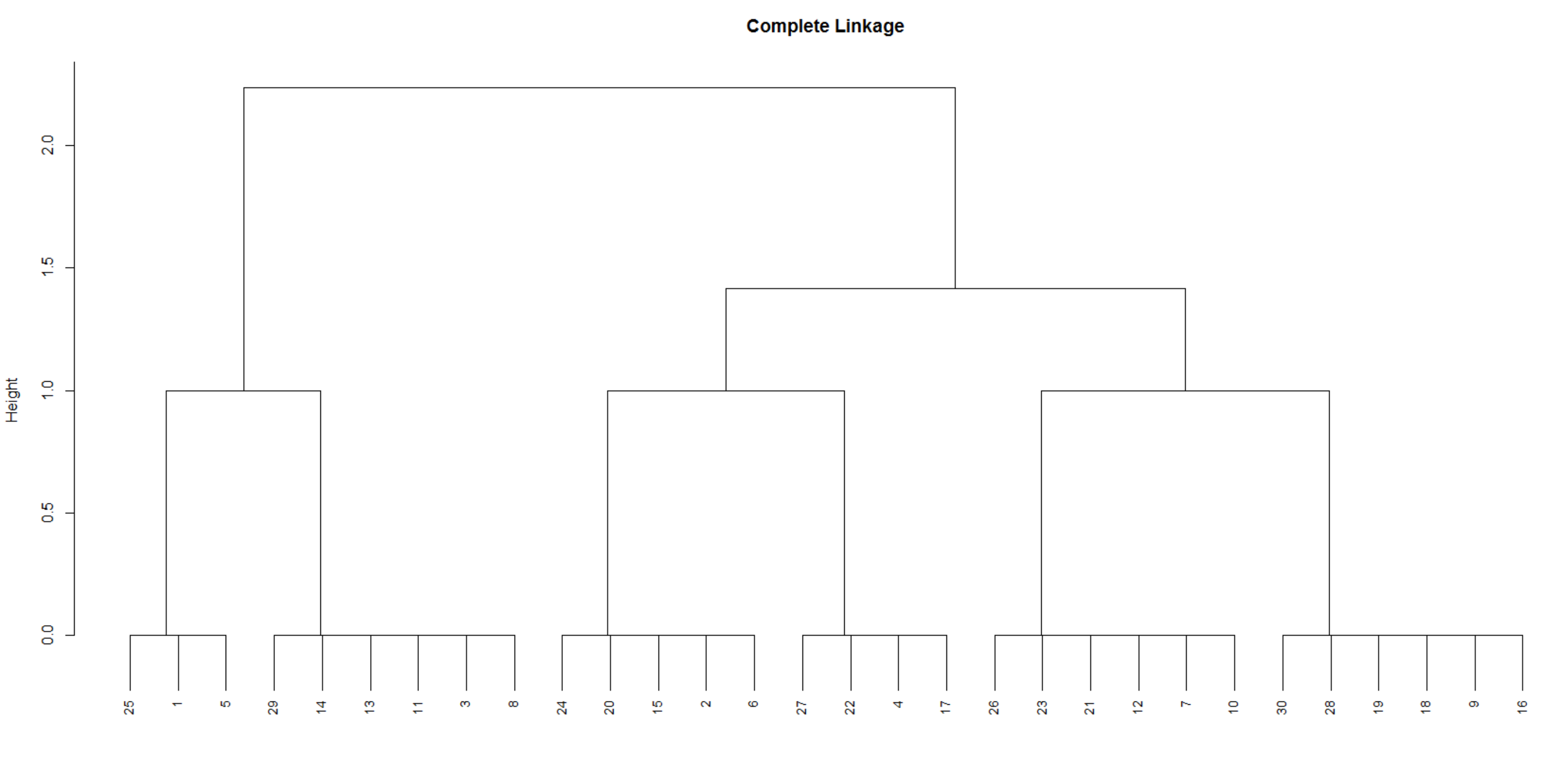
*Scientific notes & Solutions for clustering:*

**#1**

***Aim:*** *Simulation of Hierarchical clustering for 30 customers with 3 variables i.e. Name, Gender & Ethnicity*

***Exploring Data:***





***Observations & Key points:***

* In the above dendrogram, if we cut the tree at a height between 1 and 1.5 we will have 3 clusters which club with 2 clusters together reducing the homogeneity within the cluster.
* By increasing the height of the cut, Hierarchical clustering keeps clubbing the clusters within and merging them to be a part of the same cluster.
* The time complexity of most of the hierarchical clustering algorithms is quadratic i.e. O(n2).
* Therefore, for the same amount of data, hierarchical clustering will take quadratic amount of time. Imagine clustering 1 million records?
* Hence this problem can be avoided by using *K-means clustering* as it performs the on 2 important steps.

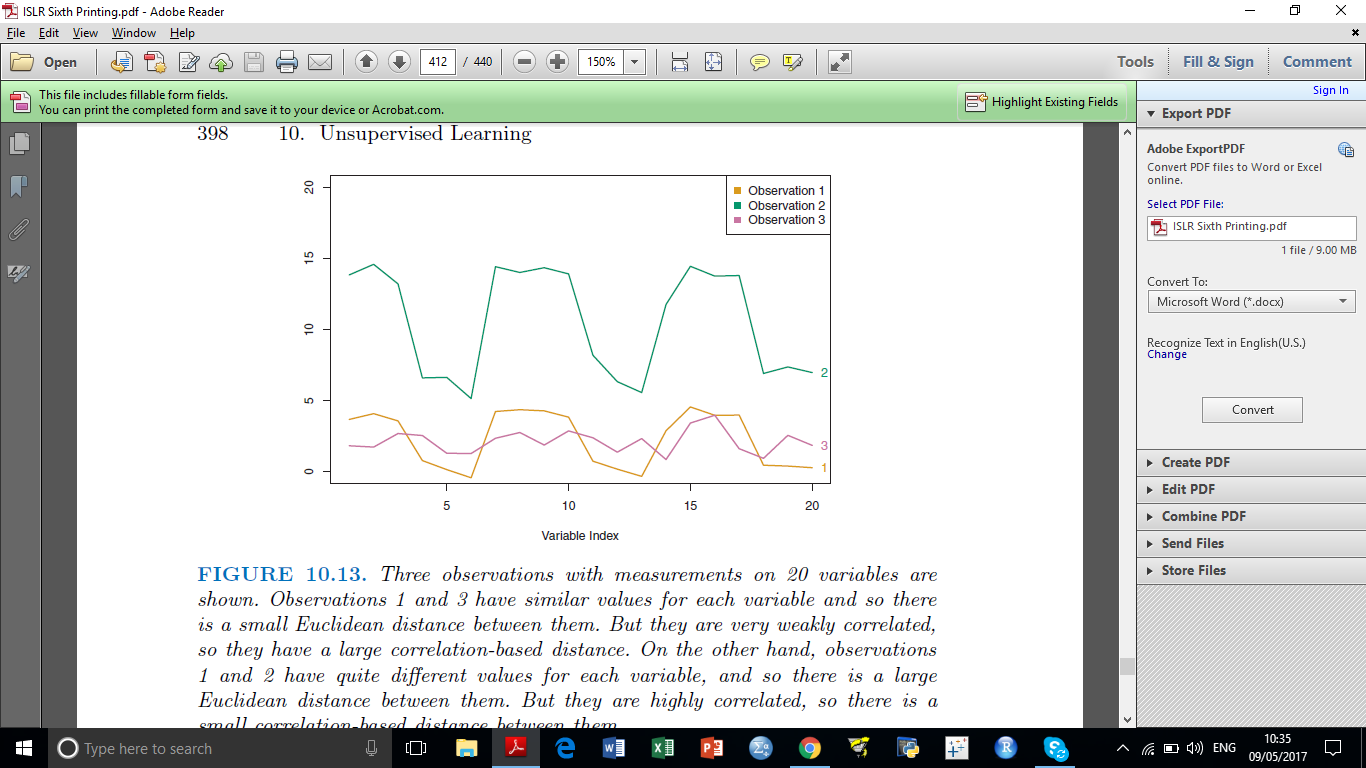
Step 1 Cluster Assignment

Step 2 Moving the Centroid and assigning the clusters [Iterative]

* In hierarchical clustering you can stop at whatever level (or clusters) you wish.

**#2**

***Aim:*** *To Elaborate and simplify the example provided about online retail*



Observation 1 and Observation 2 Patterns are similar in contrast with Observation 3. It depends upon the business problem of what we want to use, whether to look for patterns or to check Euclidean distance. Here the retailer wants to know the shoppers who have similar preferences while buying. So Co-relation based distance is a significant choice

***Analysis:***

* Computing Euclidean distance for the dissimilarity measure in case of Online retail will cluster the customers with similar shopping frequency together which is not the business problem.
* The business problem here is to find the customer similarity with respect to the preferences so that products bought by the customers of one cluster can recommend to the other customers of the same cluster.

**#3**

***Aim:*** *To check whether the proportion holds between Euclidean distance and co-relation based distance*

***Exploring Data:***

> scaled\_data <- scale(data)

> euclid\_dist <- dist(scaled\_data)^2

> cor\_dist <- as.dist(1-cor(t(scaled\_data)))

> summary(cor\_dist/euclid\_dist)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.000086 0.069135 0.133943 0.234193 0.262589 4.887686

> apply(scaled\_data,2,sd)

Murder Assault UrbanPop Rape

1 1 1 1

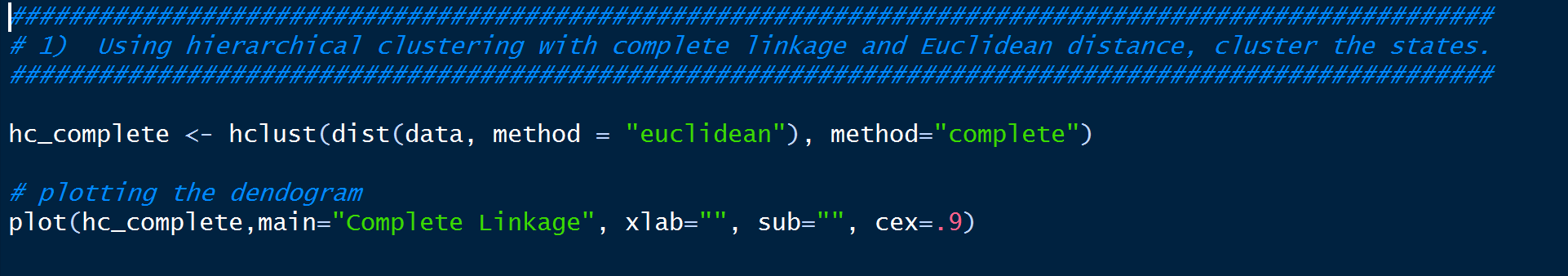
***Analysis:***

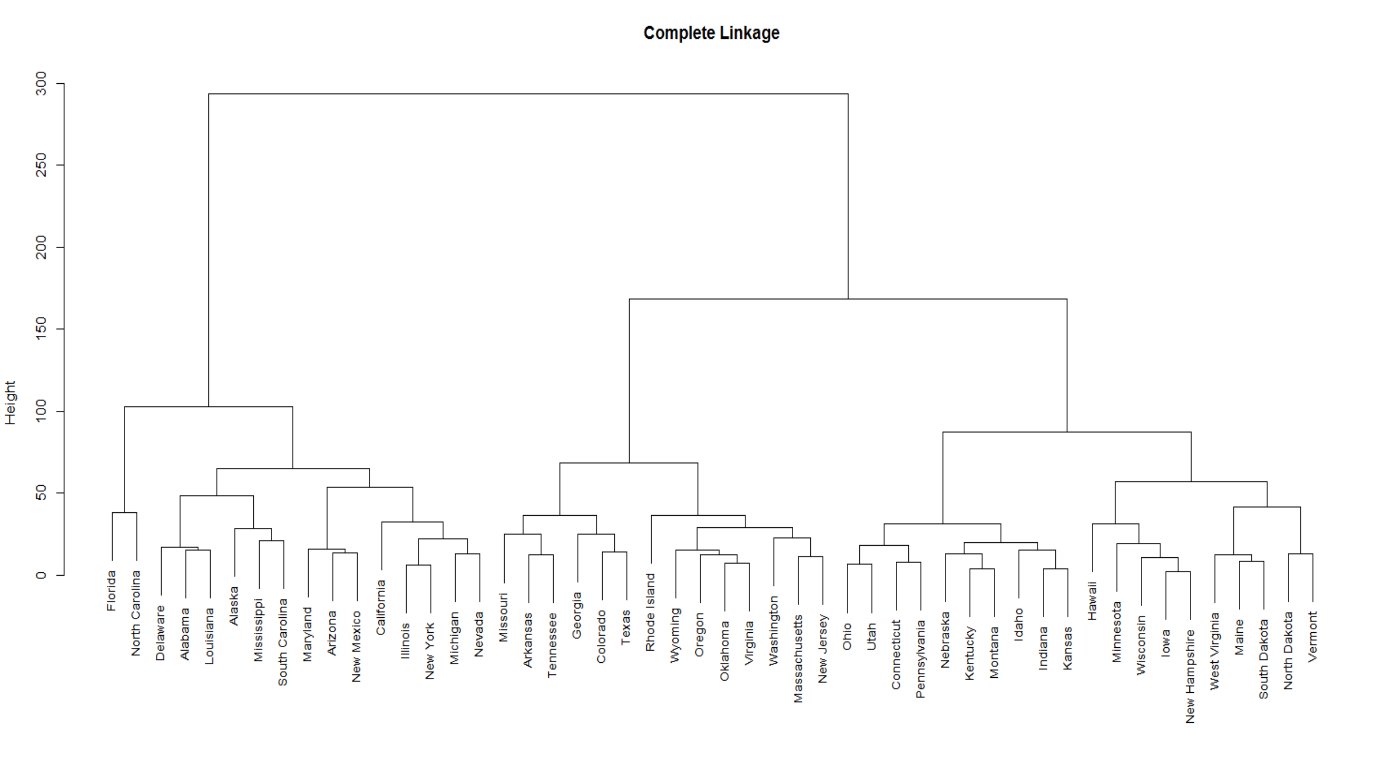
* The hierarchical cluster analysis procedure requires the specification of a distance measure, we chose the most widely used such measure, the squared Euclidean distance.
* This measure is proportional to 1-rij where rij is the correlation between the two variables. It is zero for two variables whose correlation is +1, and it is greatest for two variables correlated at -1.

**#4**

***Aim:*** *To perform Hierarchical Clustering using complete linkage and clustering the states.*

***Exploring Data and Plotting :***







> cutree(hc\_complete,3)

Alabama Alaska Arizona Arkansas California Colorado Connecticut Delaware Florida

1 1 1 2 1 2 3 1 1

Georgia Hawaii Idaho Illinois Indiana Iowa Kansas Kentucky Louisiana

2 3 3 1 3 3 3 3 1

Maine Maryland Massachusetts Michigan Minnesota Mississippi Missouri Montana Nebraska

3 1 2 1 3 1 2 3 3

Nevada New Hampshire New Jersey New Mexico New York North Carolina North Dakota Ohio Oklahoma

1 3 2 1 1 1 3 3 2

Oregon Pennsylvania Rhode Island South Carolina South Dakota Tennessee Texas Utah Vermont

2 3 2 1 3 2 2 3 3

Virginia Washington West Virginia Wisconsin Wyoming

2 2 3 3 2

> clusters <- cutree(hc\_complete,3)

> clust\_1 <- clusters[clusters == 1]

> clust\_2 <- clusters[clusters == 2]

> clust\_3 <- clusters[clusters == 3]

> coloring = c("cyan", "green", "violet")

> plot(as.phylo(hc\_complete), tip.color = clust\_color[cutree(hc\_complete, 3)], main = "Complete Linkage")

>

>

> names(clust\_1)

[1] "Alabama" "Alaska" "Arizona" "California" "Delaware" "Florida" "Illinois"

[8] "Louisiana" "Maryland" "Michigan" "Mississippi" "Nevada" "New Mexico" "New York"

[15] "North Carolina" "South Carolina"

> names(clust\_2)

[1] "Arkansas" "Colorado" "Georgia" "Massachusetts" "Missouri" "New Jersey" "Oklahoma" "Oregon"

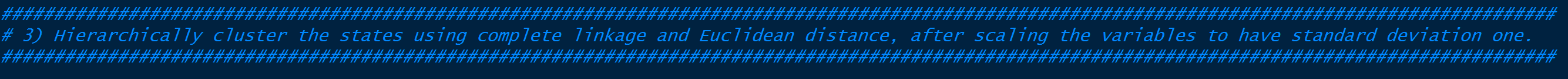
[9] "Rhode Island" "Tennessee" "Texas" "Virginia" "Washington" "Wyoming"

> names(clust\_3)

[1] "Connecticut" "Hawaii" "Idaho" "Indiana" "Iowa" "Kansas" "Kentucky" "Maine"

[9] "Minnesota" "Montana" "Nebraska" "New Hampshire" "North Dakota" "Ohio" "Pennsylvania" "South Dakota"

[17] "Utah" "Vermont" "West Virginia" "Wisconsin"



> scale\_data <- scale(data)

> hc\_scaled <- hclust(dist(scale\_data, method = "euclidean"), method="complete")

> plot(hc\_scaled, main="Hierarchical Clustering with Scaled Features")

> plot(as.phylo(hc\_scaled), tip.color = clust\_color[cutree(hc\_scaled, 3)], main = "Complete Linkage")

> scaled\_clusters <- cutree(hc\_scaled, 3)

> sclust\_1 <- scaled\_clusters[scaled\_clusters == 1]; # pull out the names of the states

> sclust\_2 <- scaled\_clusters[scaled\_clusters == 2];

> sclust\_3 <- scaled\_clusters[scaled\_clusters == 3]

> names(sclust\_1)

[1] "Alabama" "Alaska" "Georgia" "Louisiana" "Mississippi" "North Carolina" "South Carolina" "Tennessee"

> names(sclust\_2)

[1] "Arizona" "California" "Colorado" "Florida" "Illinois" "Maryland" "Michigan" "Nevada" "New Mexico" "New York" "Texas"

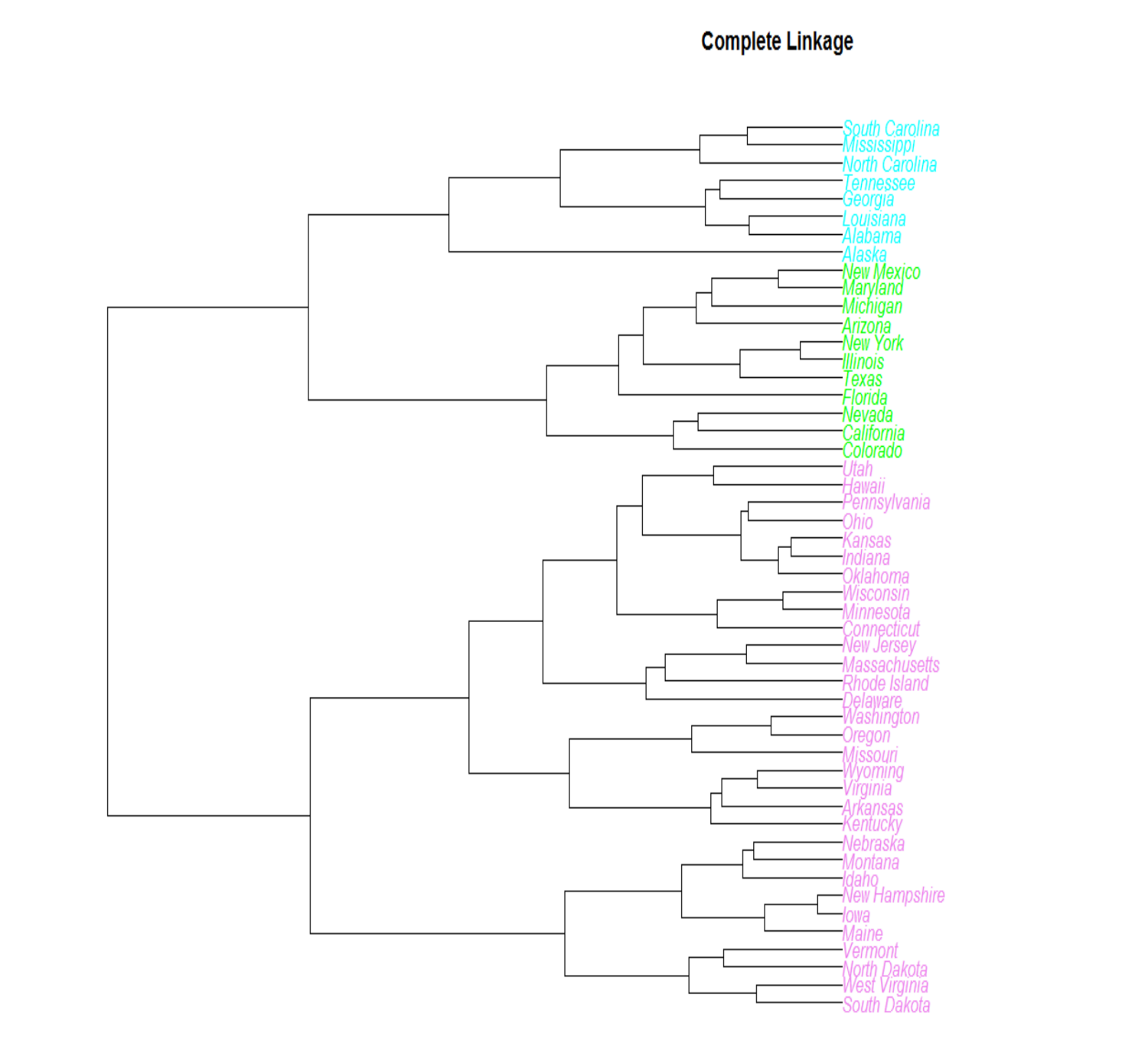
> names(sclust\_3)

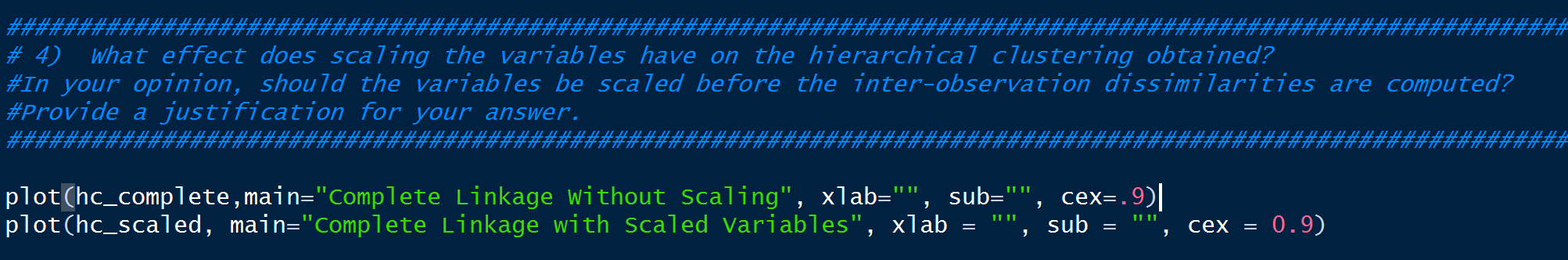
[1] "Arkansas" "Connecticut" "Delaware" "Hawaii" "Idaho" "Indiana" "Iowa" "Kansas" "Kentucky"

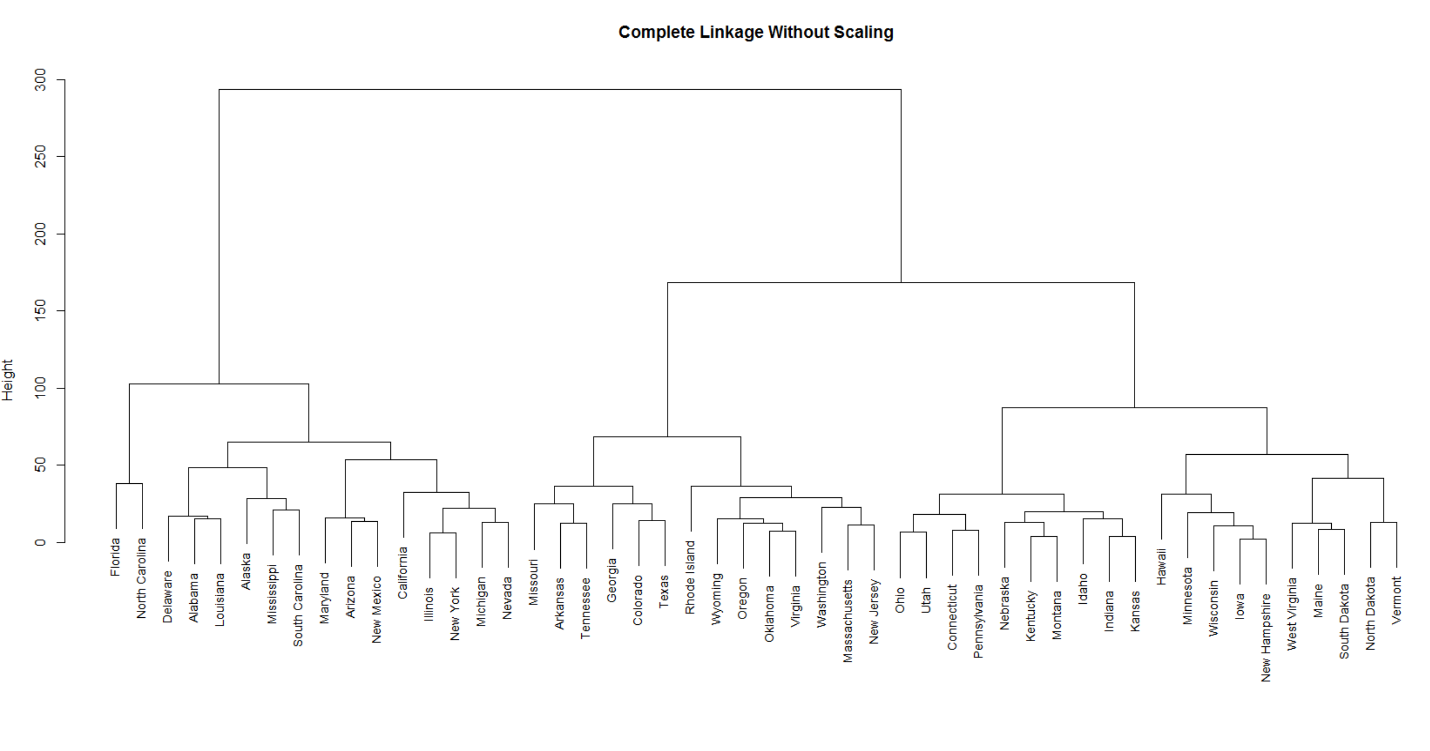
[10] "Maine" "Massachusetts" "Minnesota" "Missouri" "Montana" "Nebraska" "New Hampshire" "New Jersey" "North Dakota"

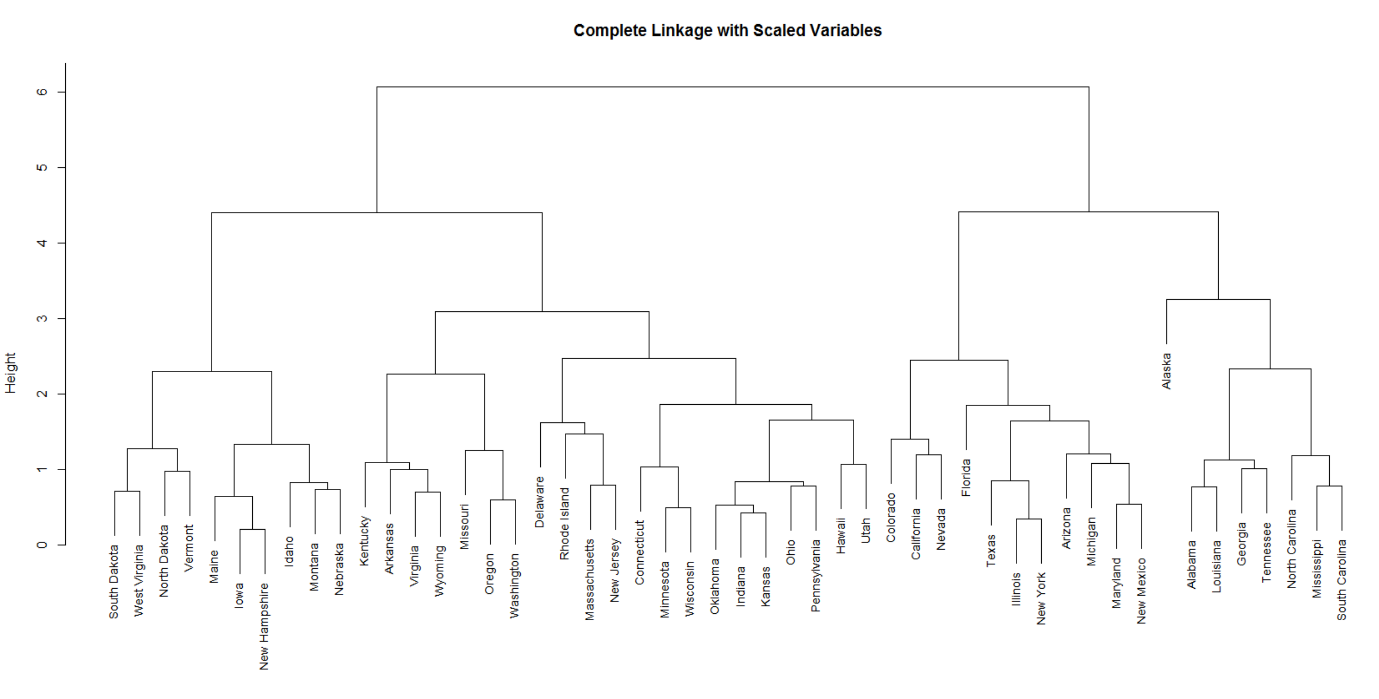
[19] "Ohio" "Oklahoma" "Oregon" "Pennsylvania" "Rhode Island" "South Dakota" "Utah" "Vermont" "Virginia"

[28] "Washington" "West Virginia" "Wisconsin" "Wyoming"









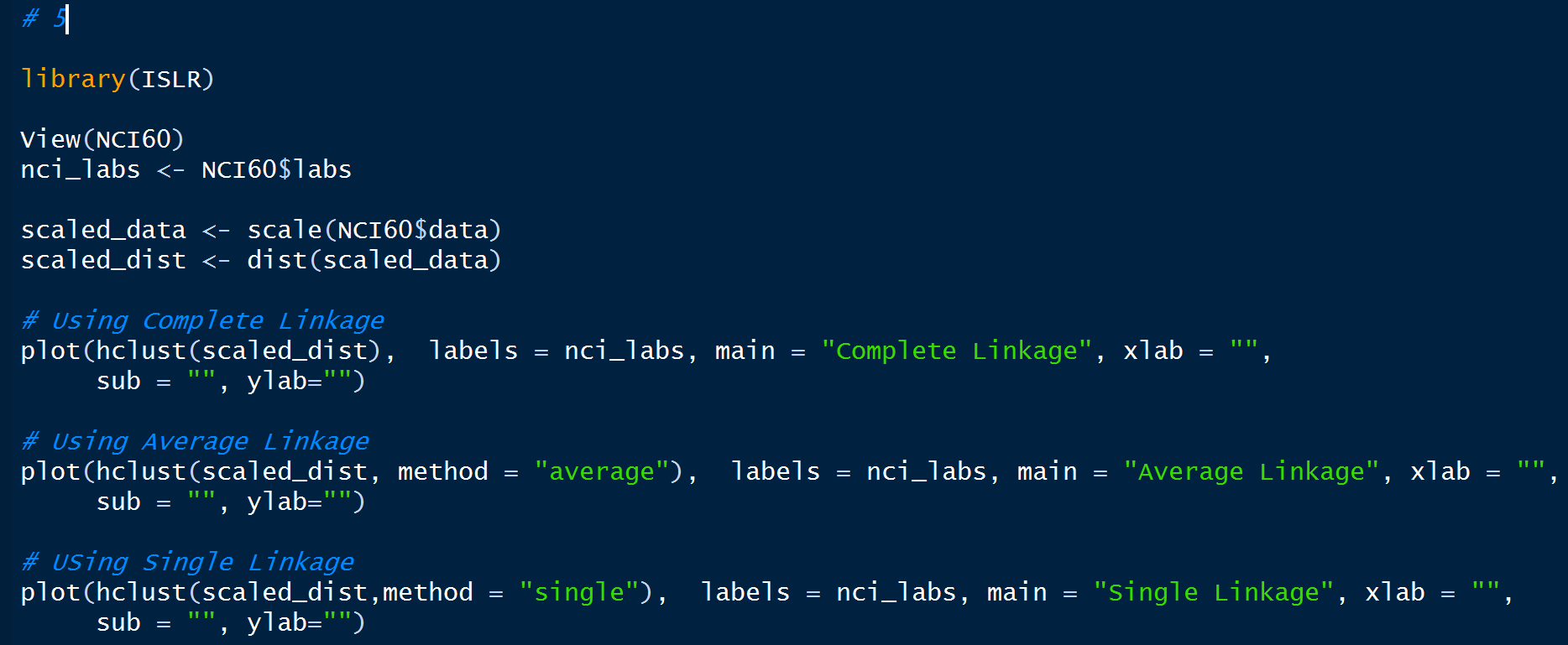
***Key Points:***

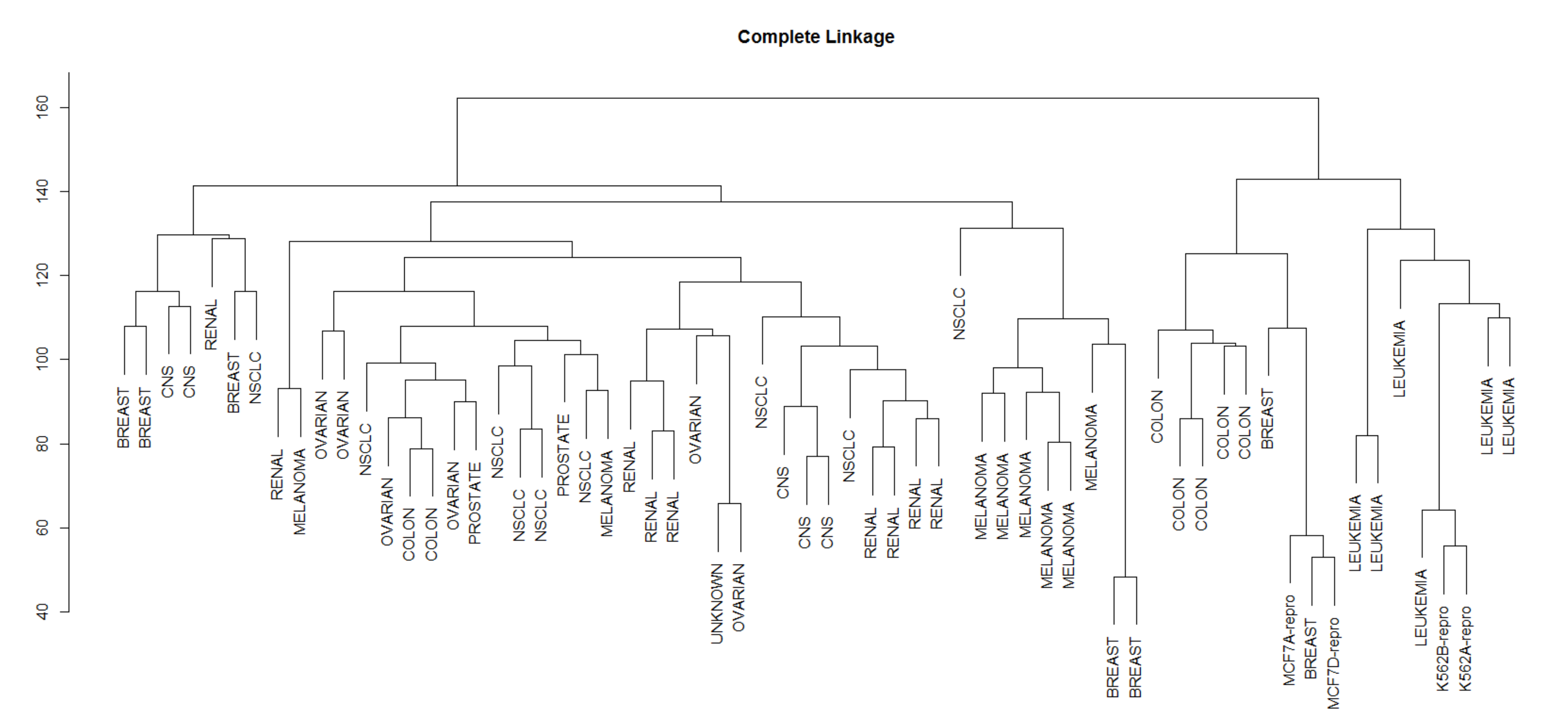
* Yes it should be scaled.
* Scaling impacts the hierarchical clustering, we don’t get the same clusters.
* For example, with scaling, Michigan clusters nearby Arizona but without scaling it clusters with Nevada.
* Some other observations are the branch for Alaska is shorter in the scaled tree.
* It is also important to scale, so that the units of urbanpop has an equal contribution to the hierarchical clustering as other variables.

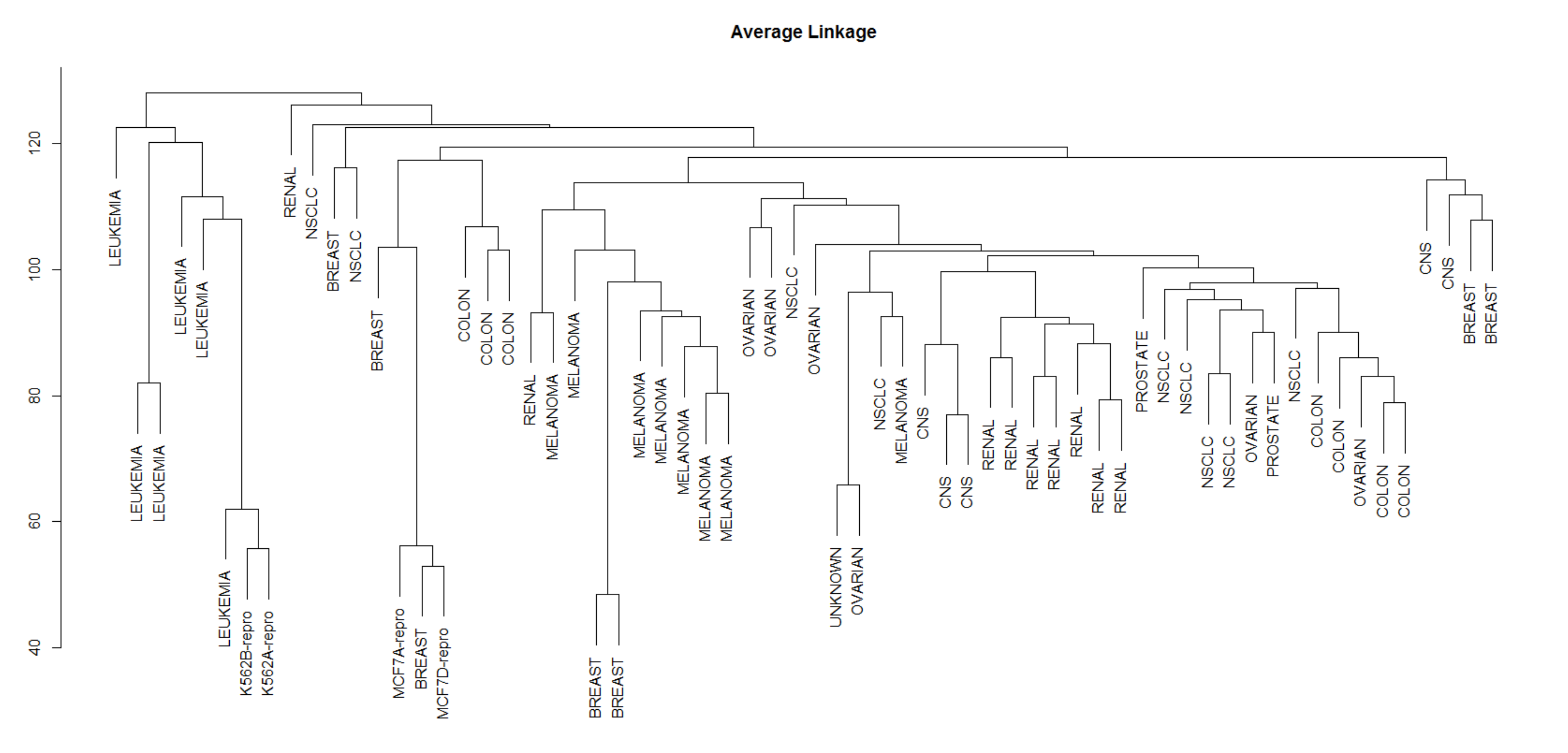
**#5**

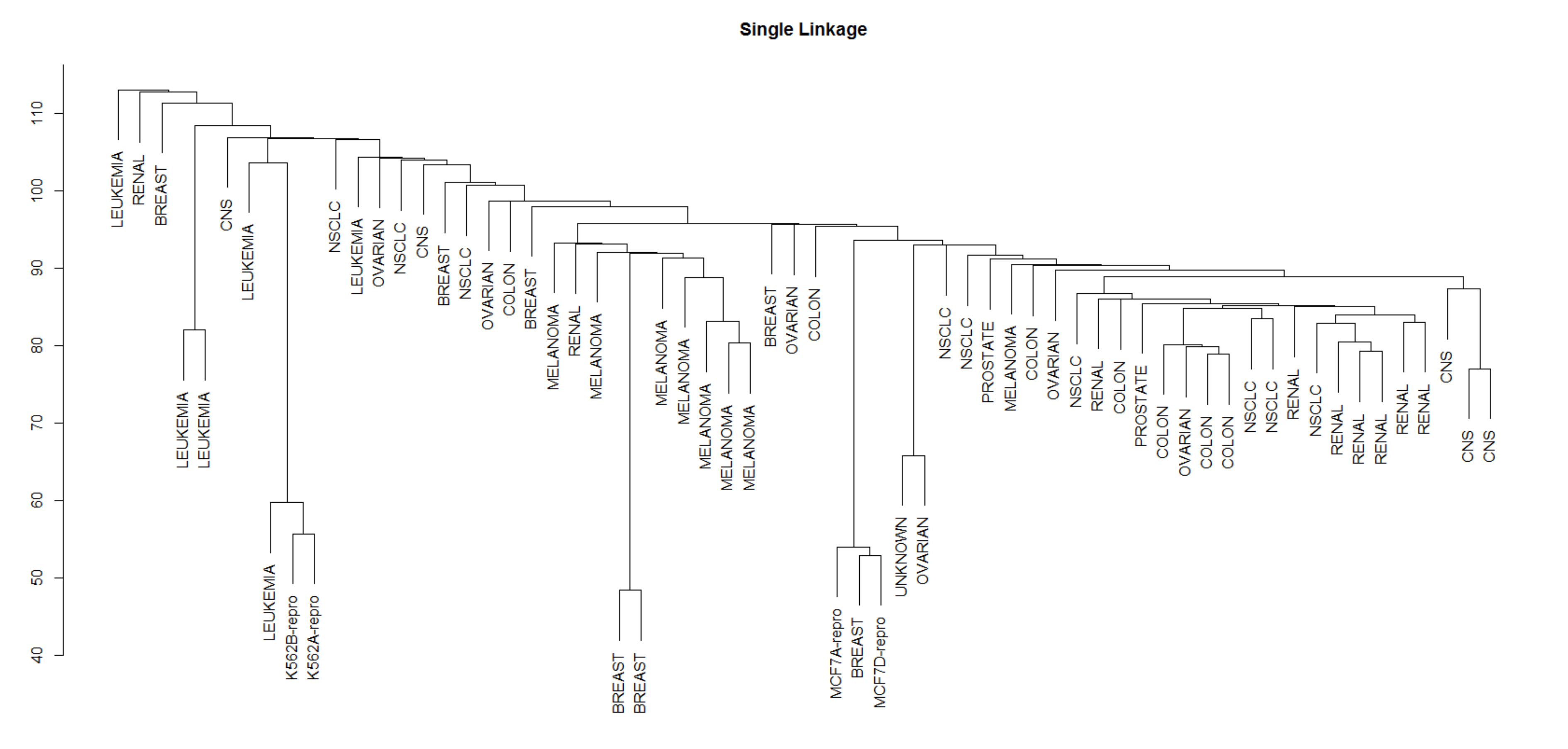
***Aim:*** *Cluster Analysis on NC160 data using Complete, Average and single linkage.*

***Exploring Data & Plotting :***









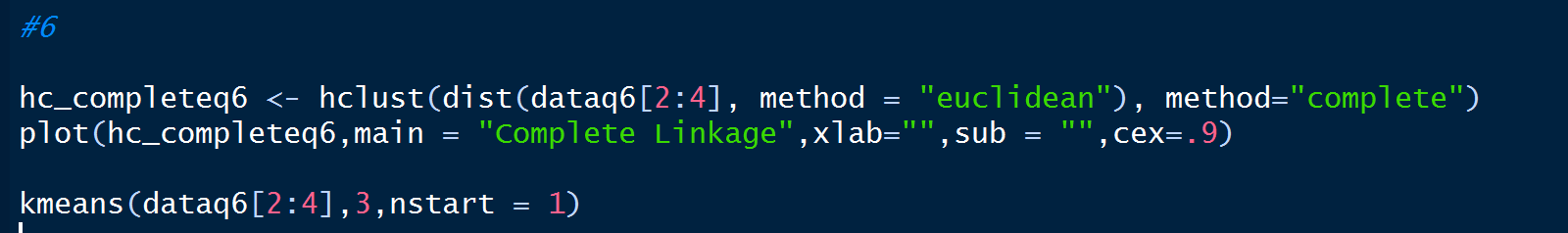
***Observations:***

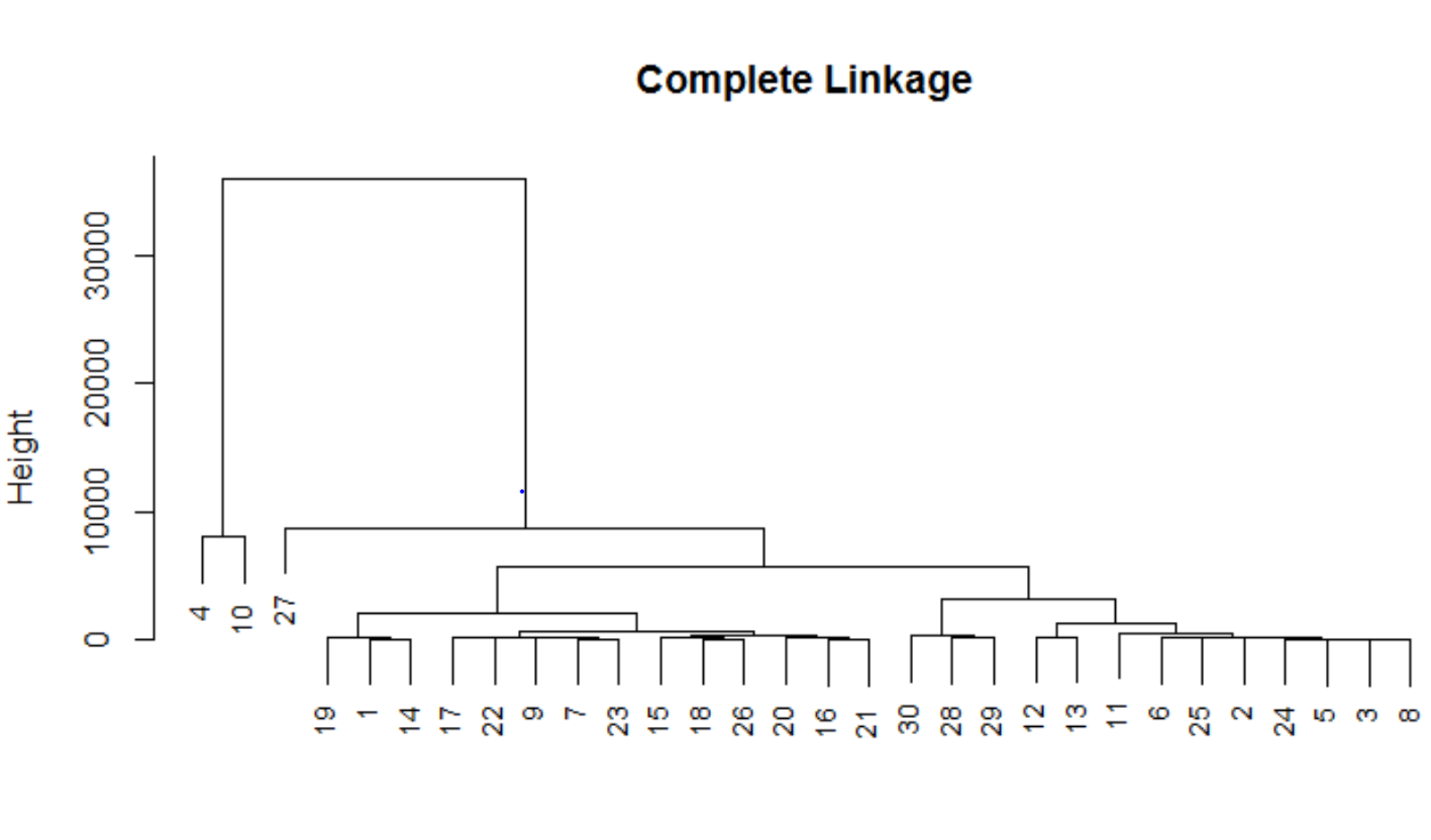
* Complete and Average linkage tend to yield more balanced clusters.
* Single linkage method tend to yield trailing clusters.

**#6**

***Aim:*** *Checking the distortion of clusters due to the presence of outliers.*

***Exploring Data & Plotting:***





> kmeans(dataq6[2:4],3,nstart = 1)

K-means clustering with 3 clusters of sizes 4, 2, 24

Cluster means:

Gender Ethnicity Income

1 0.5 2.500000 6366.250

2 0.5 2.500000 32000.000

3 0.5 1.916667 1980.042

Clustering vector:

[1] 3 3 3 2 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1

Within cluster sum of squares by cluster:

[1] 7568449 32000001 22149367

(between\_SS / total\_SS = 96.5 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" "size"

[8] "iter" "ifault"

***Analysis:***

* Employee no.4 and 10 are outliers and when using the hierarchical clustering technique, they are getting clustered into 1 cluster.
* Hence we need to use mixture models which use a technique called soft clustering which assigns a point to a data point indicating the strength of the data point to a certain cluster which is the probability of data point belonging to a cluster.
* This method of clustering is efficient and significant than the method of clustering used in hierarchical clustering and K-means.

**#7**

***Issues in Cluster Analysis:***

* **Large number of samples**

The number of samples to be processed is very high. Algorithms have to be very conscious of scaling issues. Like many interesting problems, clustering in general is NP-hard, and practical and successful data mining algorithms usually scale linear or log-linear. Quadratic and cubic scaling may also be allowable but a linear behaviour is highly desirable.

* **High dimensionality.**

The number of features is very high and may even exceed the number of samples. So one has to face the curse of dimensionality.

* **Sparsity.**

Most features are zero for most samples, i.e. the object-feature matrix is sparse. This property strongly affects the measurements of similarity and the computational complexity.

* **Strong non-Gaussian distribution of feature values.**

The data is so skewed that it cannot be safely modelled by normal distributions.

* **Significant outliers.**

Outliers may have significant importance. Finding these outliers is highly non-trivial, and removing them is not necessarily desirable.

***Application context***

* **Legacy clustering**

Previous cluster analysis results are often available. This knowledge should be reused instead of starting each analysis from scratch.

**Advantages of Hierarchical Clustering**

* No apriori information about the number of clusters required.
* Easy to implement and gives best result in some cases.

**Disadvantages of Hierarchical Clustering**

* Sensitivity to noise and outliers
* Breaking larger clusters
* The nesting of the clusters reduces homogeneity within the clusters.
* Sometimes it is difficult to identify the correct number of clusters by the dendrogram.

**K-Means Advantages :**

* If variables are huge, then  K-Means is most of the times computationally faster than hierarchical clustering, if we keep k smalls.
* K-Means produce tighter clusters than hierarchical clustering

**K-Means Disadvantages :**

* Difficult to predict K-Value.
* Different initial partitions can result in different final clusters.
* It does not work well with clusters (in the original data) of Different size and Different density

**Hierarchical Clustering vs K-means**

**Time Complexity:**

* K-means is linear in the number of data objects i.e. O(n), where *n* is the number of data objects.
* The time complexity of most of the hierarchical clustering algorithms is quadratic i.e. O(n2).
* Therefore, for the same amount of data, hierarchical clustering will take quadratic amount of time. Imagine clustering 1 million records?

**Shape of the clusters:**

* K-means works well when the shape of clusters are hyper-spherical (or circular in 2 dimensions). If the natural clusters occurring in the dataset are non-spherical then probably K-means is not a good choice.

**Repeatability:**

* In hierarchical clustering you can stop at whatever level (or clusters) you wish.
* K-means clustering requires prior knowledge of K (or number of clusters),

Eg: In Market segmentation of T-shirt sizing, using k-means is good enough to segment the T-shirt sizing to S,M,L. Here k=3

**THE END**