



UNSUPERVISED LEARNING IN R

Introduction to the case study

Objectives

- Complete analysis using unsupervised learning
 - Reinforce what you've already learned
 - Add steps not covered before (e.g. preparing data, selecting good features for supervised learning)
 - Emphasize creativity

Example use case

- Human breast mass data:
 - Ten features measured of each cell nuclei
 - Summary information is provided for each group of cells
 - Includes diagnosis: benign (not cancerous) and malignant (cancerous)
Will not use this for modeling, as it is the target variable

Overall steps

- Download data and prepare data for modeling
- Exploratory data analysis (# observations, # features, etc.)
- Perform PCA and interpret results
- Complete two types of clustering
- Understand and compare the two types
- Combine PCA and clustering

Review: PCA in R

```
> pr.iris <- prcomp(x = iris[-5], scale = FALSE, center = TRUE)
> summary(pr.iris)
```

Importance of components:

	PC1	PC2	PC3	PC4
Standard deviation	2.0563	0.49262	0.2797	0.15439
Proportion of Variance	0.9246	0.05307	0.0171	0.00521
Cumulative Proportion	0.9246	0.97769	0.9948	1.00000

Unsupervised learning is open-ended

- Steps in this use case are only one example of what can be done
- There are other approaches to analyzing this dataset



UNSUPERVISED LEARNING IN R

Let's practice!



UNSUPERVISED LEARNING IN R

PCA review and next steps

Review thus far

- Downloaded data and prepared it for modeling
- Exploratory data analysis
- Performed principal component analysis

Next steps

- Complete hierarchical clustering
- Complete k-means clustering
- Combine PCA and clustering
- Contrast results of hierarchical clustering with diagnosis
- Compare hierarchical and k-means clustering results
- PCA as a pre-processing step for clustering

Review: Hierarchical clustering in R

```
> # Calculates similarity as Euclidean distance between observations  
> s <- dist(x) x is a data matrix
```

```
> # Returns hierarchical clustering model  
> hclust(s)
```

```
Call:  
hclust(d = s)
```

```
Cluster method      : complete  
Distance           : euclidean  
Number of objects: 50
```

Review: k-means in R

```
> # k-means algorithm with 5 centers, run 20 times  
> kmeans(x, centers = 5, nstart = 20)
```

- One observation per row, one feature per column
- k-means has a random component
- Run algorithm multiple times to improve odds of the best model



UNSUPERVISED LEARNING IN R

Let's practice!



UNSUPERVISED LEARNING IN R

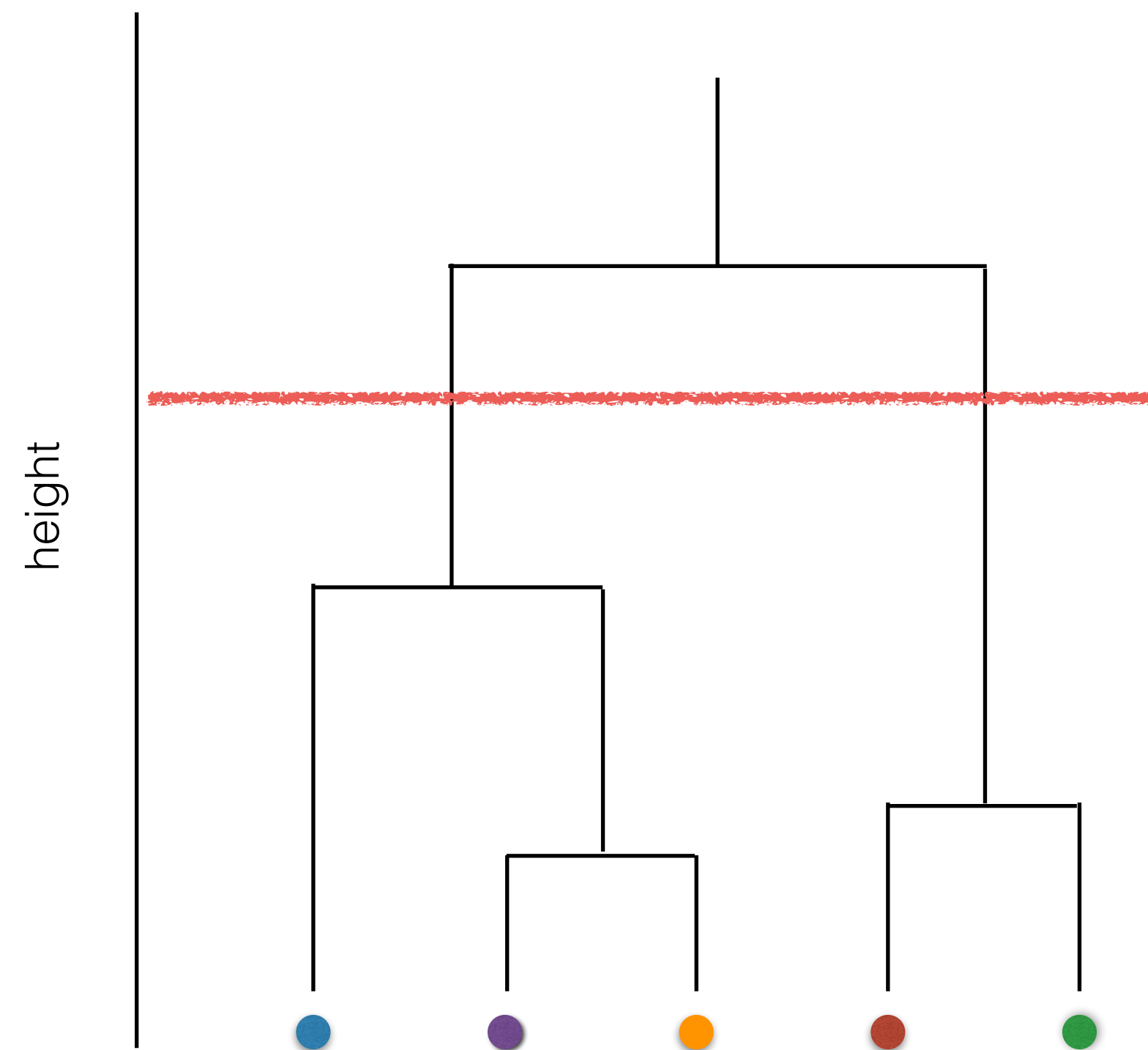
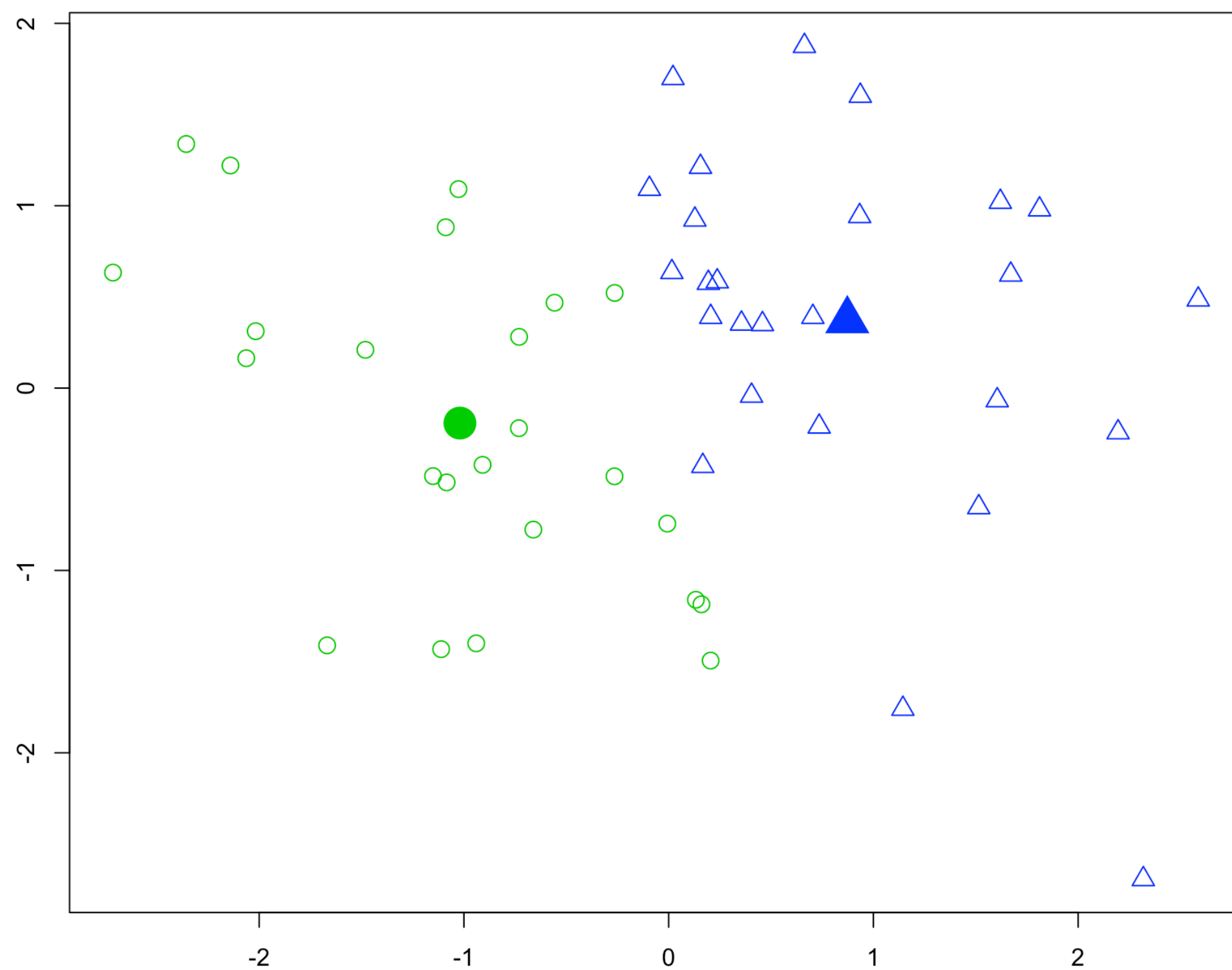
Wrap-up and review

Case study wrap-up

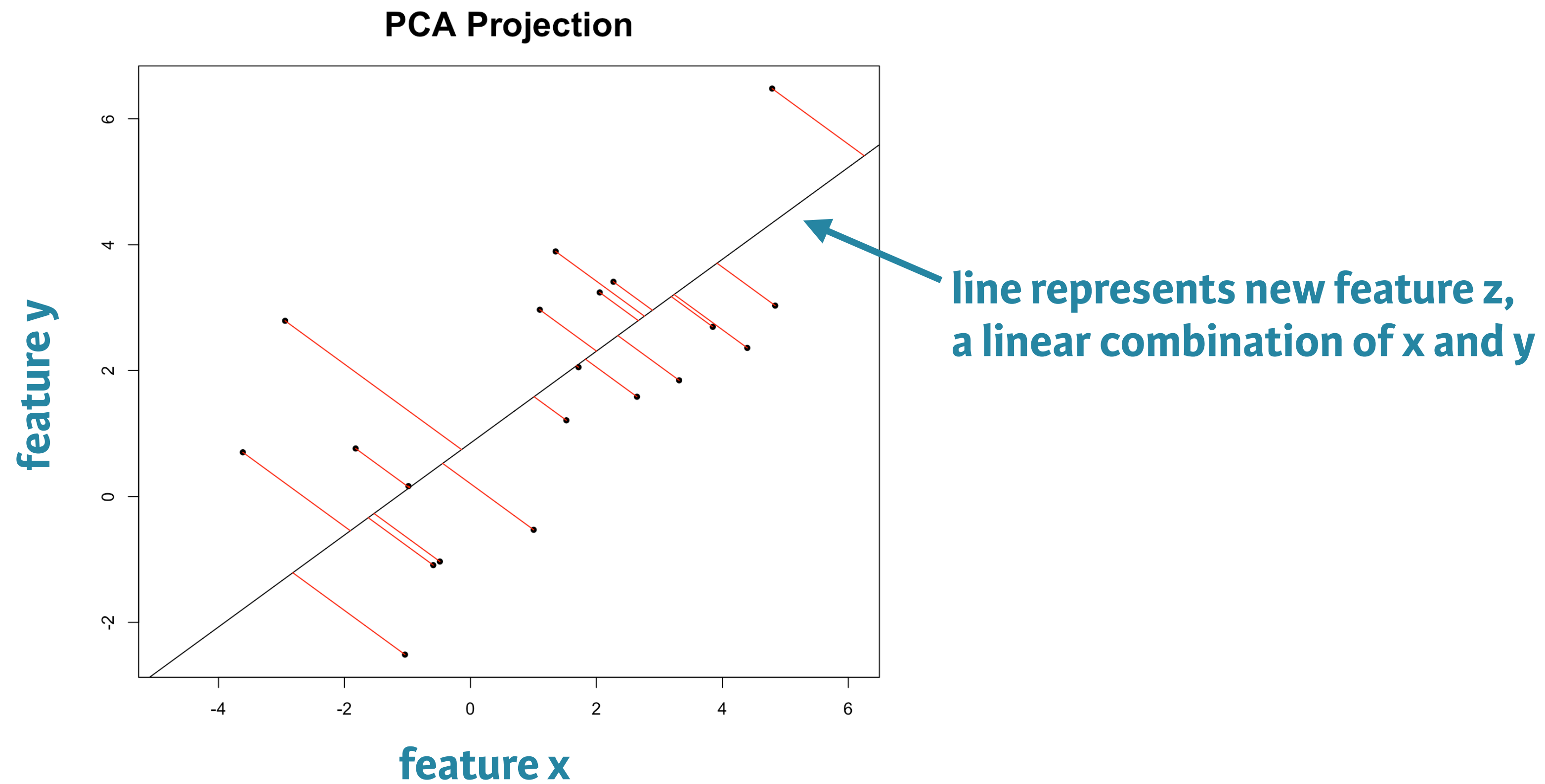
- Entire data analysis process using unsupervised learning
- Creative approach to modeling
- Prepared to tackle real world problems

Types of clustering

Iteration 5



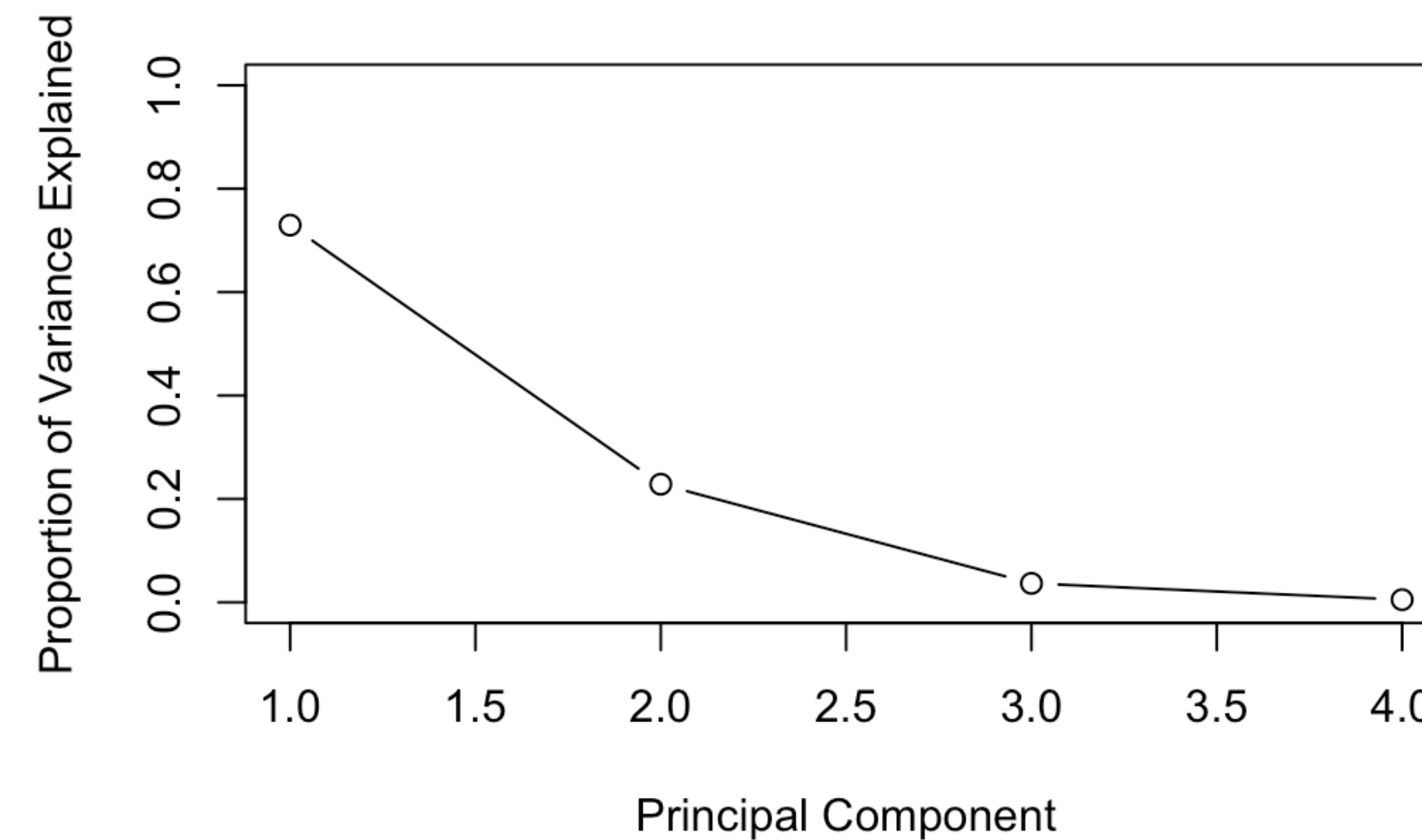
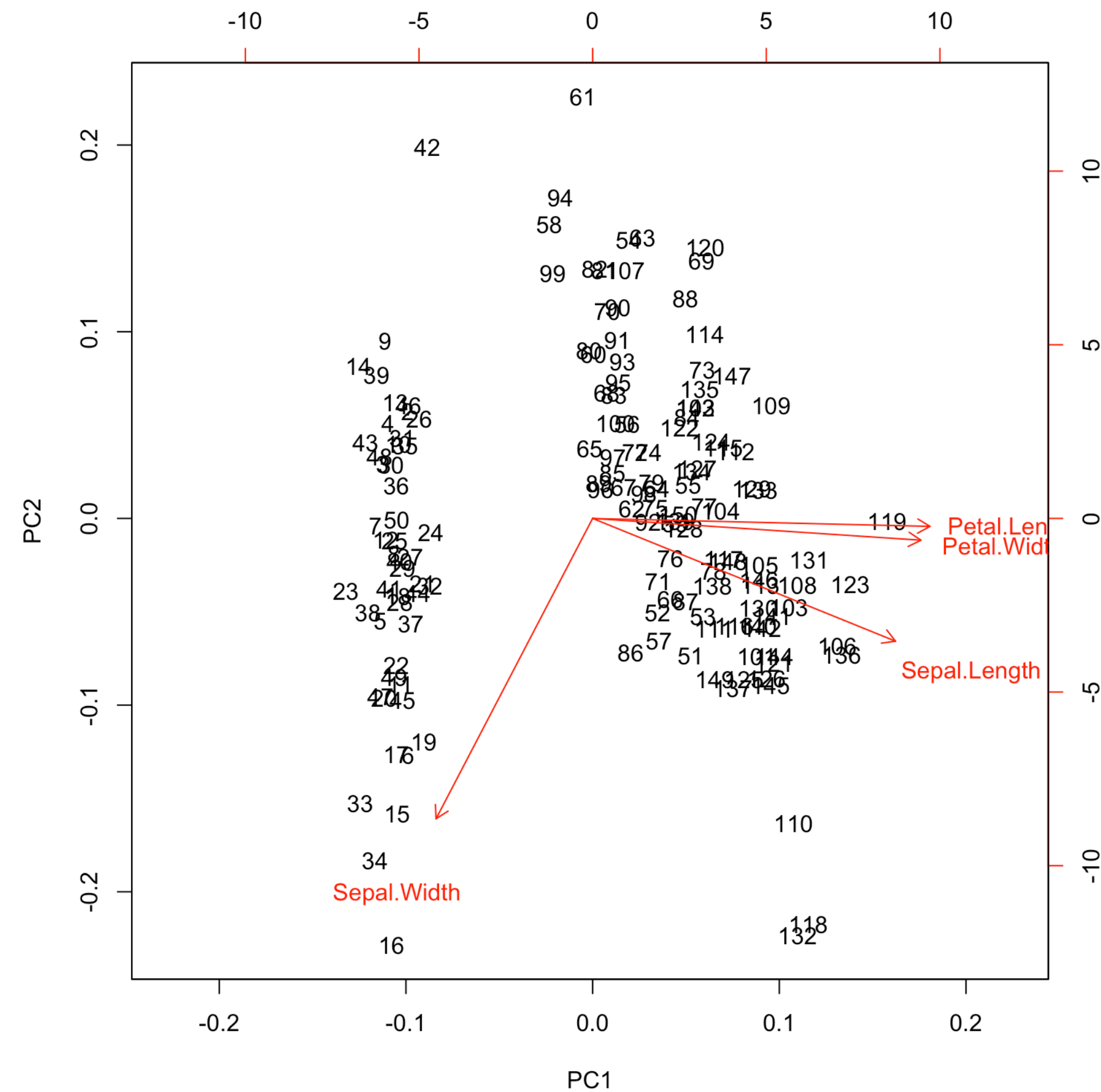
Dimensionality reduction



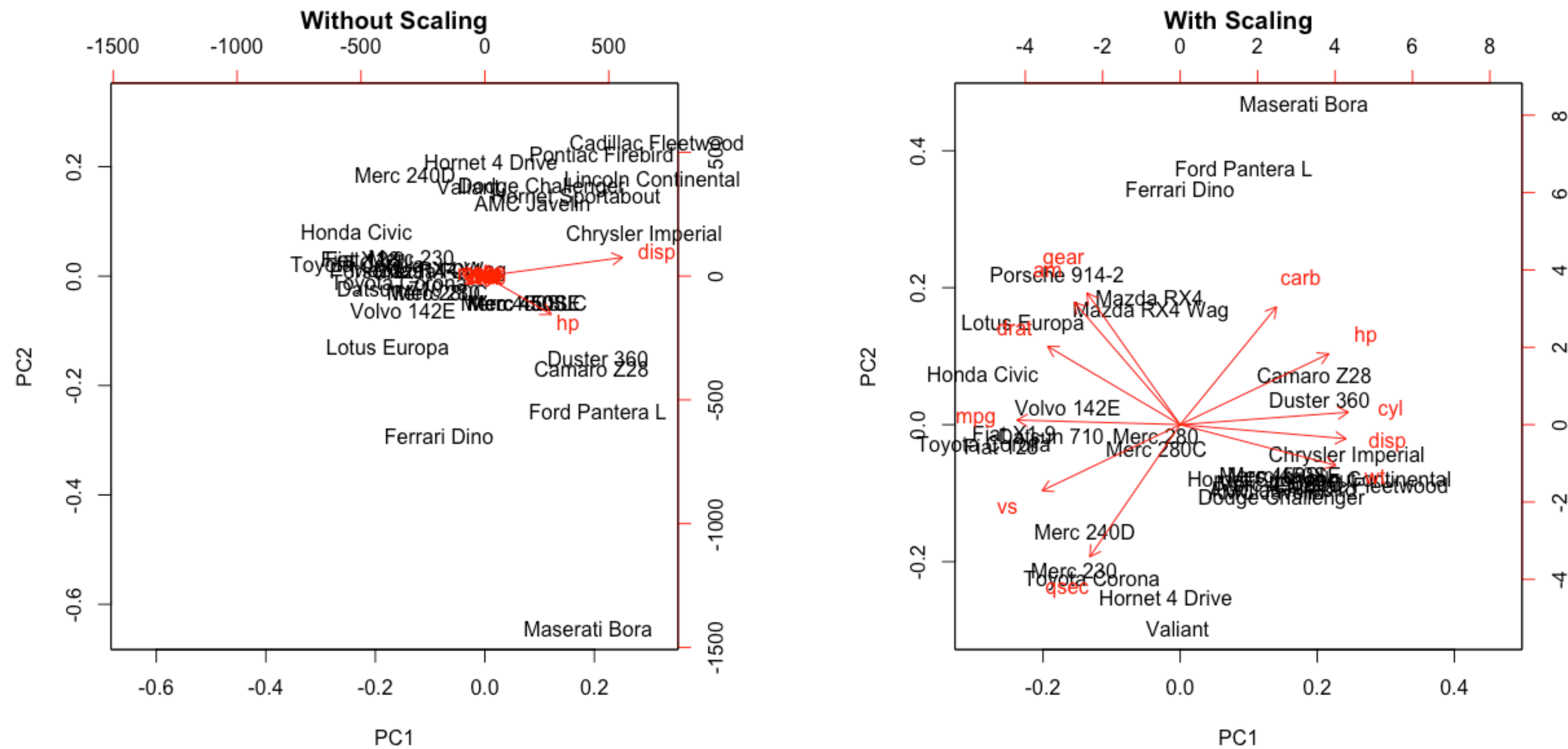
Model selection

```
# Initialize total within sum of squares error: wss
> wss <- 0
>
# Look over 1 to 15 possible clusters
> for (i in 1:15) {
  # Fit the model: km.out
  km.out <- kmeans(pokemon, centers = i, nstart = 20, iter.max = 50)
  # Save the within cluster sum of squares
  wss[i] <- km.out$tot.withinss
}
>
# Produce a scree plot
> plot(1:15, wss, type = "b",
      xlab = "Number of Clusters",
      ylab = "Within groups sum of squares")
```

Interpreting PCA results



Importance of scaling data



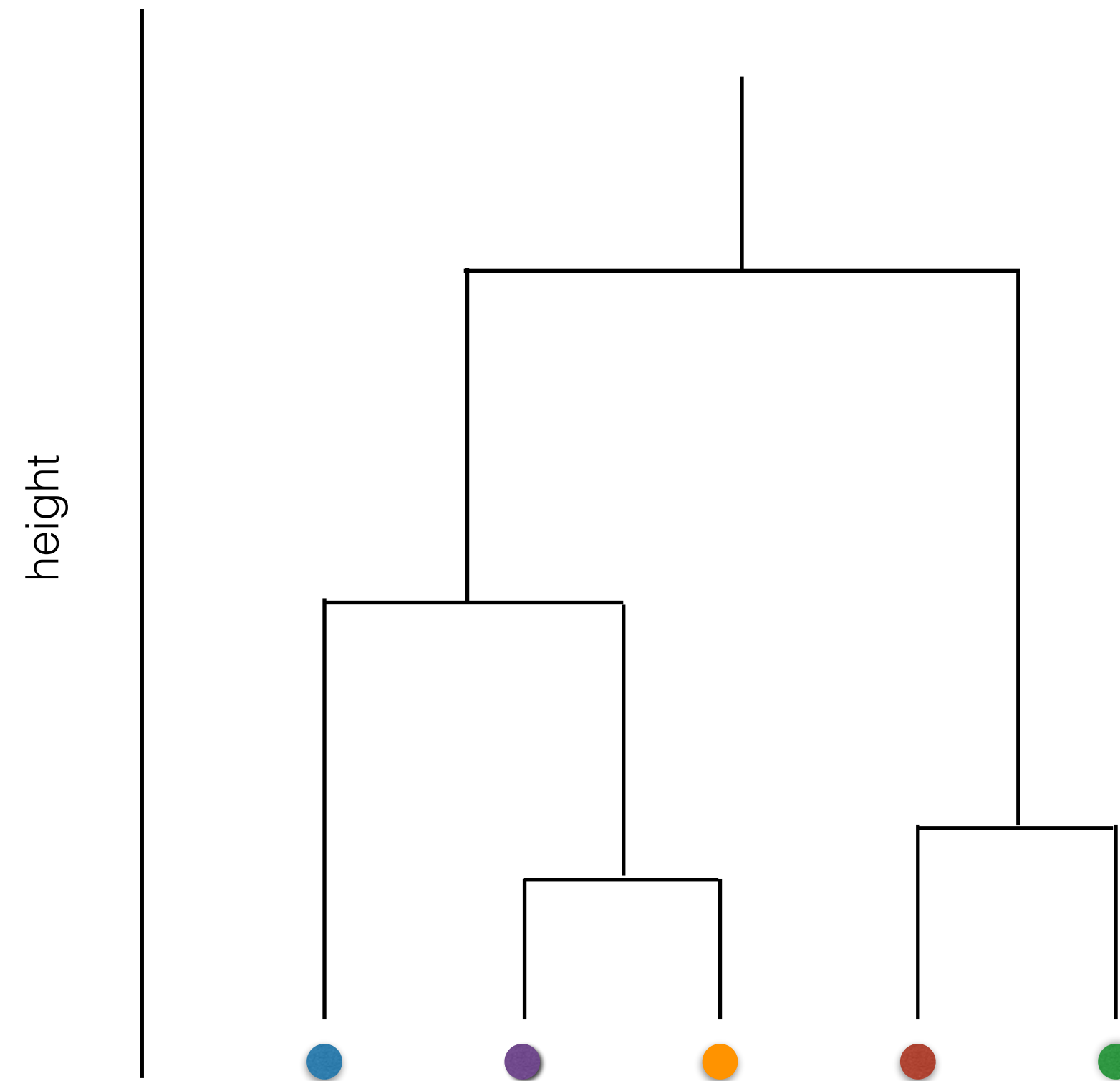
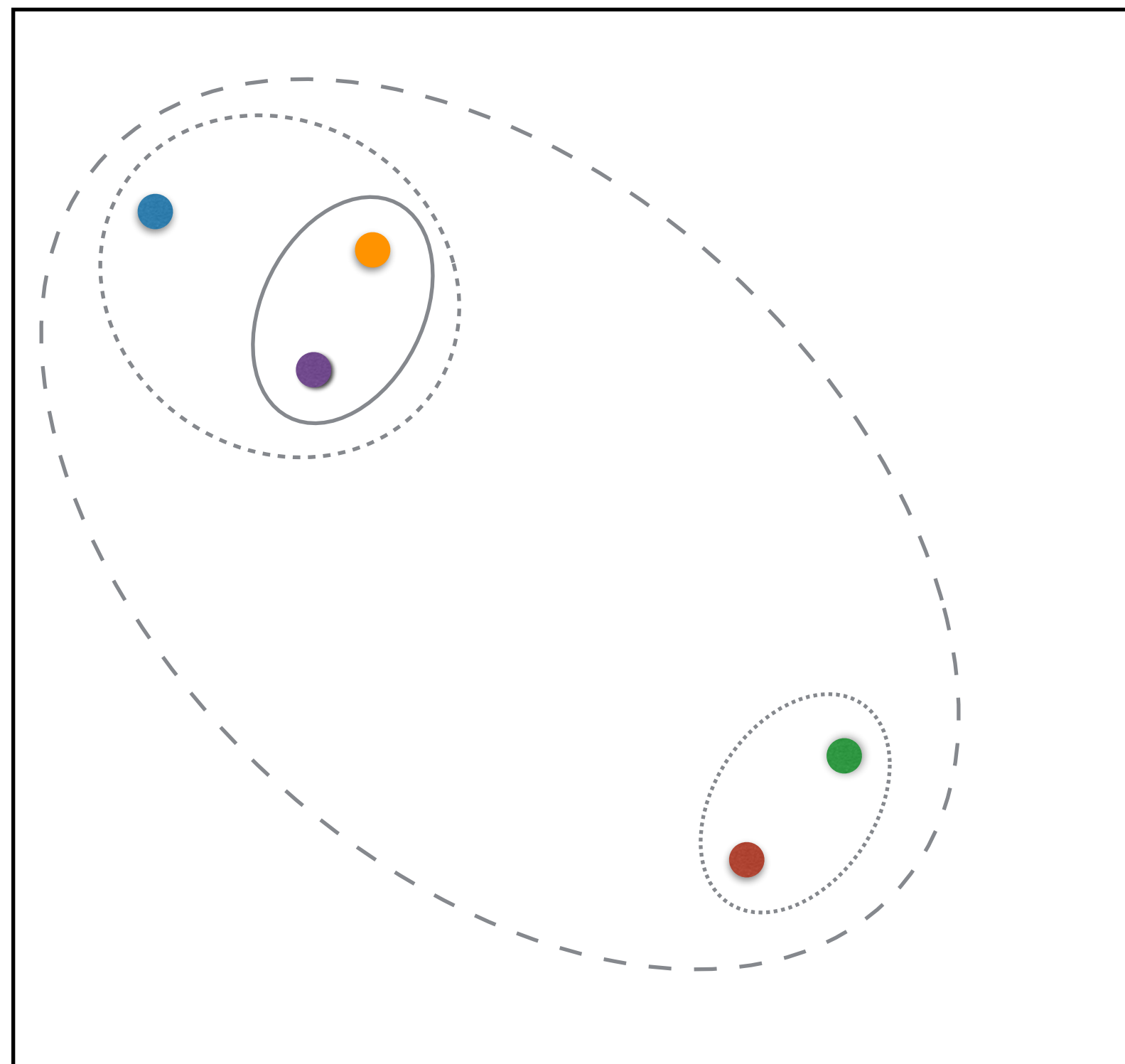
Course review

```
> pr.iris <- prcomp(x = iris[-5], scale = FALSE, center = TRUE)
> summary(pr.iris)
```

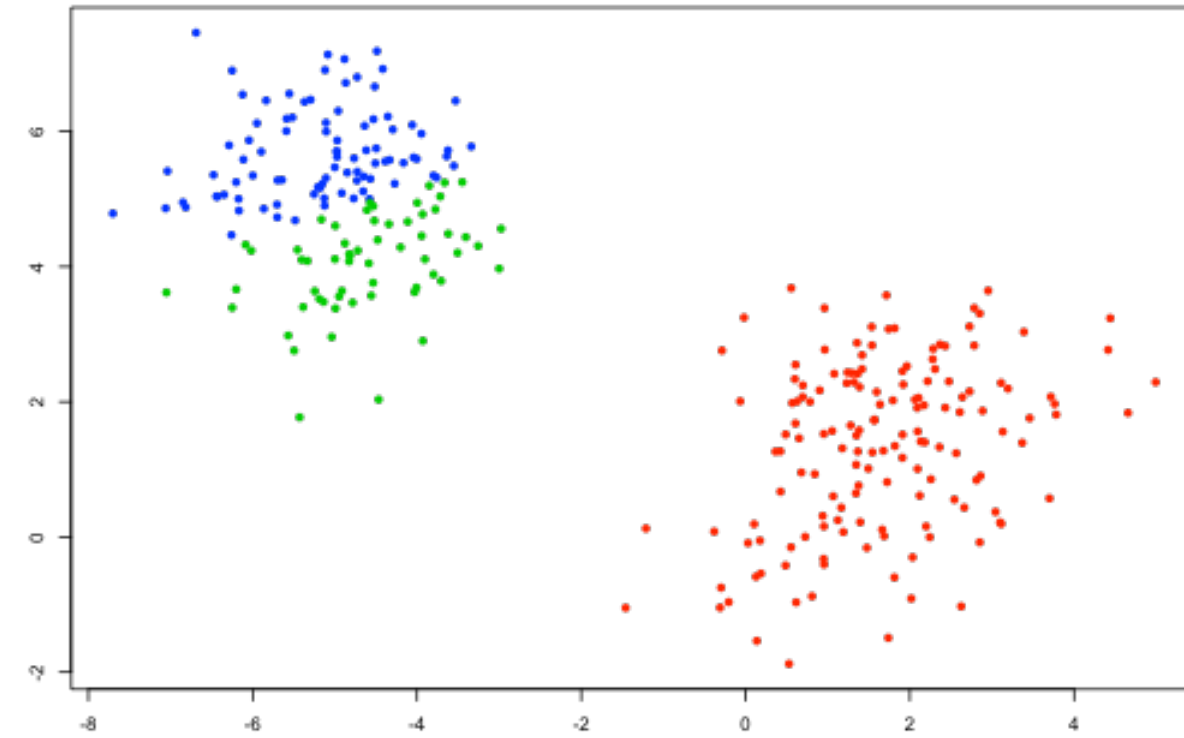
Importance of components:

	PC1	PC2	PC3	PC4
Standard deviation	2.0563	0.49262	0.2797	0.15439
Proportion of Variance	0.9246	0.05307	0.0171	0.00521
Cumulative Proportion	0.9246	0.97769	0.9948	1.00000

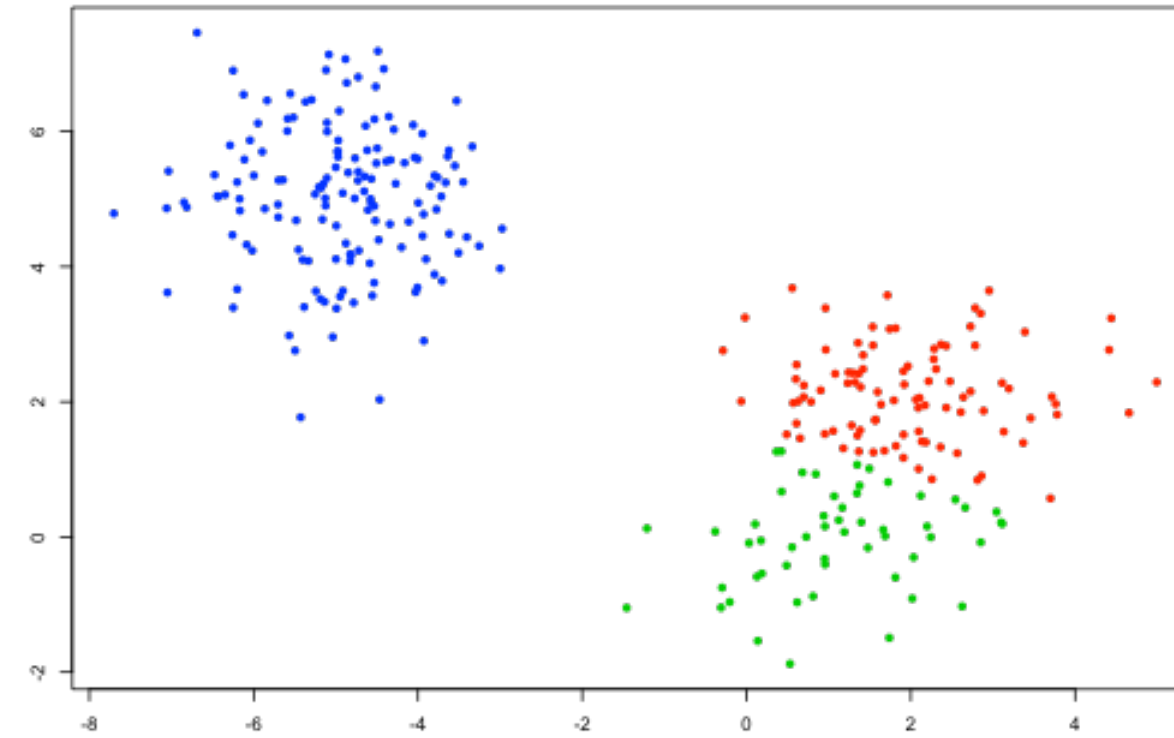
Dendrogram



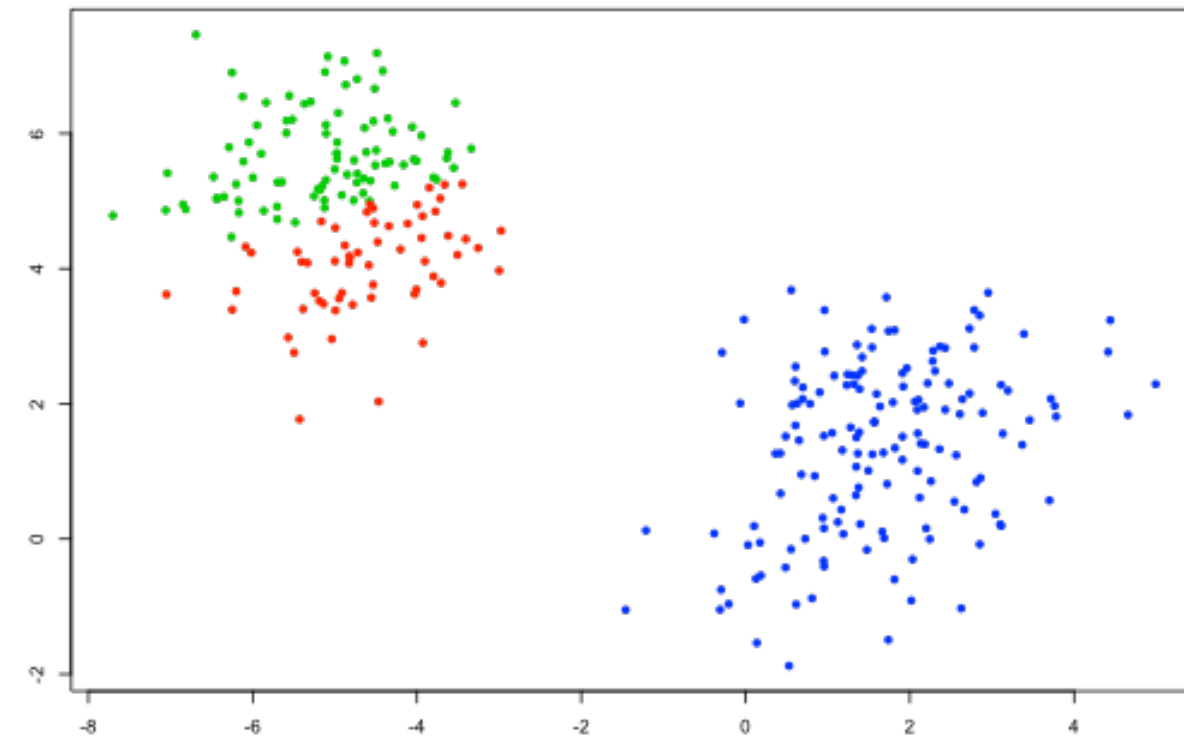
Total Within SS = 615.340155087286



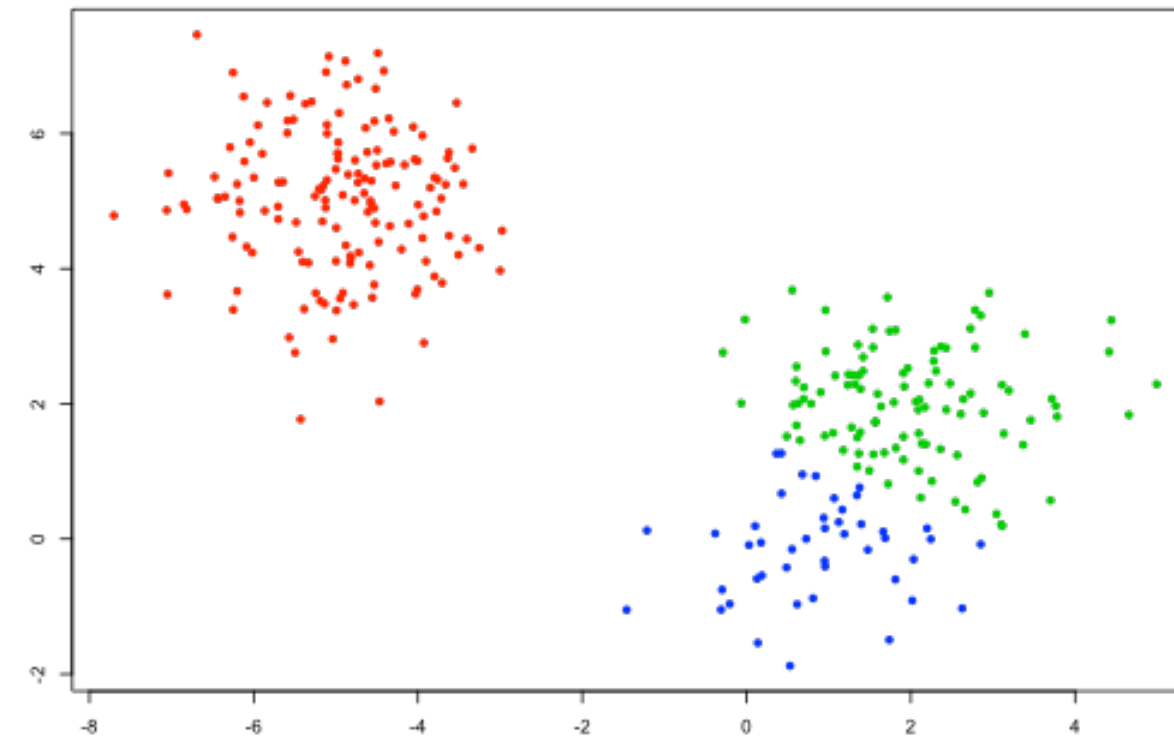
Total Within SS = 533.574674546762



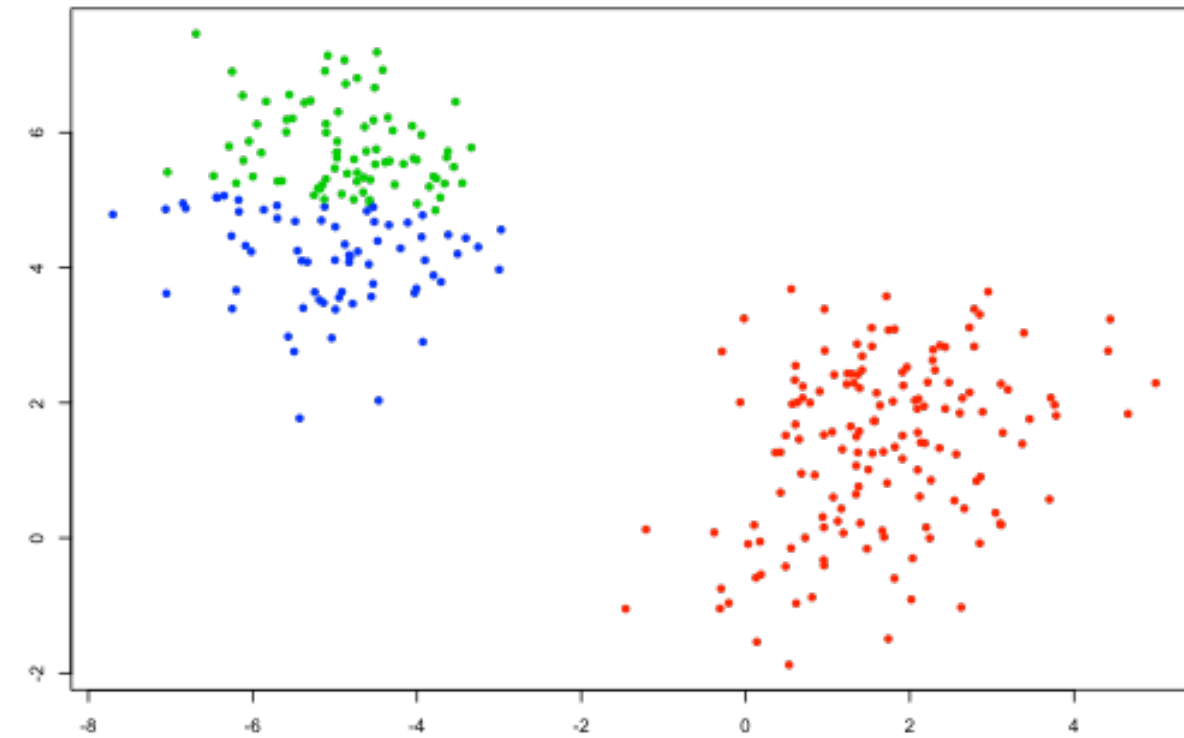
Total Within SS = 615.340155087286



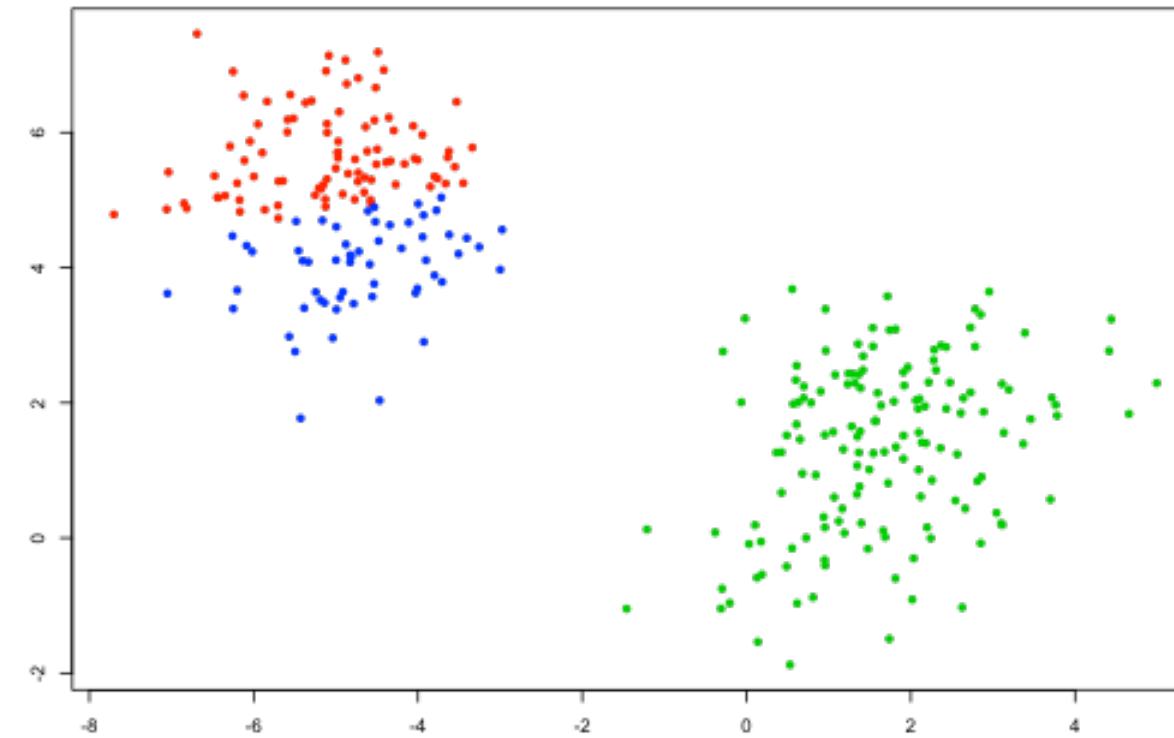
Total Within SS = 533.630315419879



Total Within SS = 620.078927563537

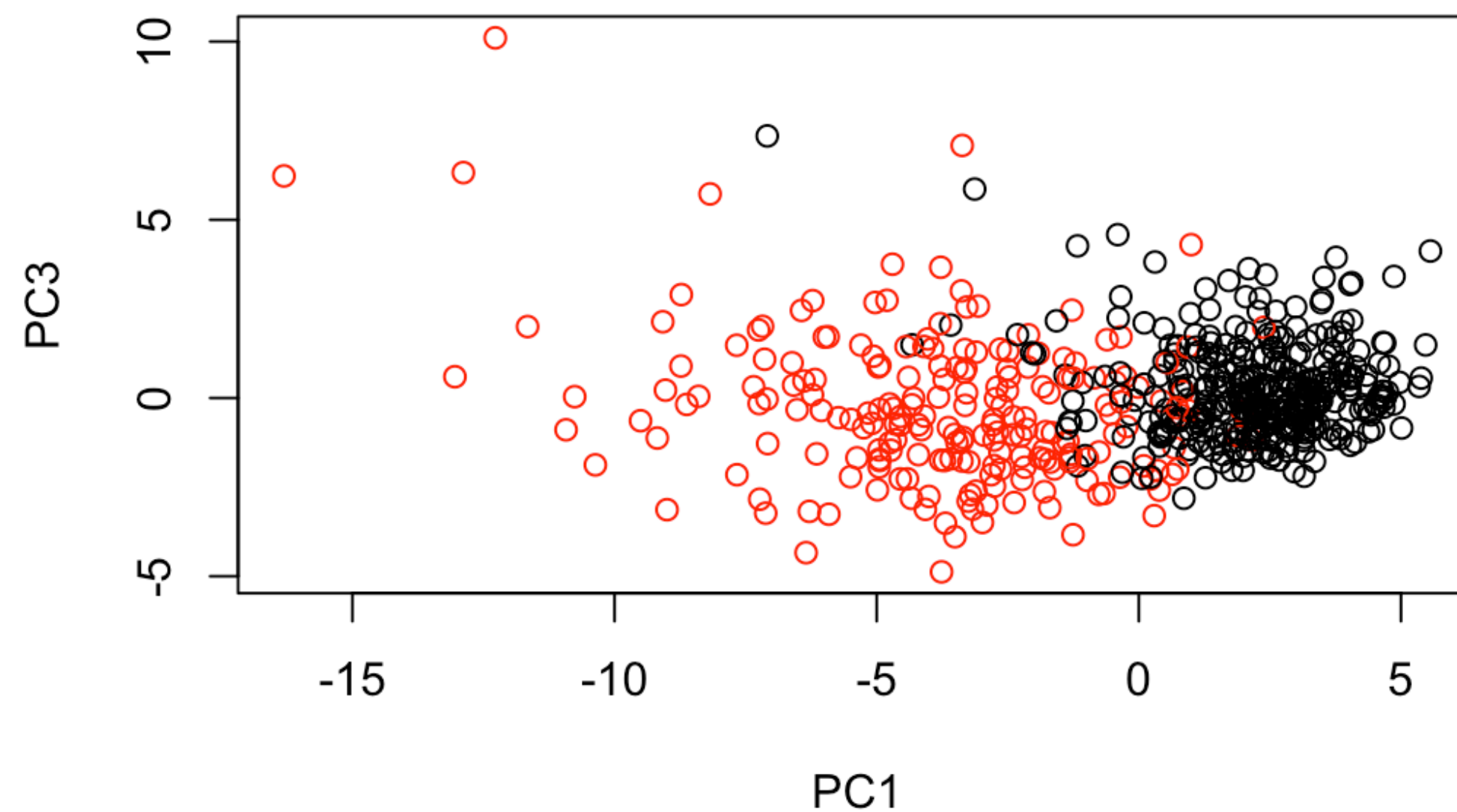


Total Within SS = 615.293089004275



Course review

```
> # Repeat for components 1 and 3  
> plot(wisc.pr$x[, c(1, 3)], col = (diagnosis + 1),  
      xlab = "PC1", ylab = "PC3")
```





UNSUPERVISED LEARNING IN R

Thanks!