

Business 4720 - Class 13

Joerg Evermann

Faculty of Business Administration
Memorial University of Newfoundland
jevermann@mun.ca



Unless otherwise indicated, the copyright in this material is owned by Joerg Evermann. This material is licensed to you under the [Creative Commons by-attribution non-commercial license \(CC BY-NC 4.0\)](#)

What You Will Learn:

- ▶ Unsupervised Machine Learning
 - ▶ Dimension Reduction using Principal Components Analysis
 - ▶ Clustering

Based On

Gareth James, Daniel Witten, Trevor Hastie and Robert Tibshirani: *An Introduction to Statistical Learning with Applications in R*. 2nd edition, corrected printing, June 2023. (ISLR2)

<https://www.statlearning.com>

Chapter 12

Trevor Hastie, Robert Tibshirani, and Jerome Friedman: *The Elements of Statistical Learning*. 2nd edition, 12th corrected printing, 2017. (ESL)

<https://hastie.su.domains/ElemStatLearn/>

Chapter 14

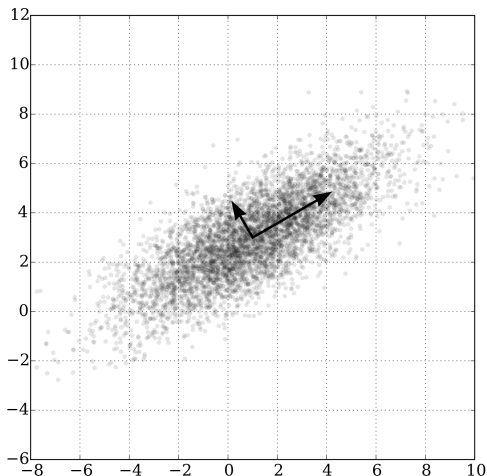
Kevin P. Murphy: *Probabilistic Machine Learning – An Introduction*. MIT Press 2022.

<https://probml.github.io/pml-book/book1.html>

Chapters 20, 21

Principal Components Analysis (PCA)

- ▶ Create linear combinations of predictors that are:
 - ▶ Maximally variable
 - ▶ Independent of each other
- ▶ Generally fewer components than predictors
- ▶ Can be used instead of original predictors in regression or classification models
- ▶ Useful when the problem dimensionality is too high (too many parameters)
 - ▶ Can be interpreted as a regularization method
- ▶ Useful for visualization to show 2D or 3D summaries of high-dimensional data



Scatterplot with **Principal Components**

(Eigenvectors of covariance matrix, scaled by the square root of the corresponding eigenvalue and shifted to mean)

<https://commons.wikimedia.org/wiki/File:GaussianScatterPCA.svg>

- ▶ First principal component (PC) for $1 \leq i \leq n$ data values and p variables:

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{ip}$$

- ▶ **Loading vector** $\phi = (\phi_{11}, \dots, \phi_{p1})$ scaled so that $\|\phi\|_2 = 1$
- ▶ Assume zero-centered variables
- ▶ Maximize:

$$\frac{1}{n} \sum_{i=1}^n z_{i1}^2 = \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j1} x_{ij} \right)^2 \quad (\text{Variance of } z_{i1})$$

- ▶ Subject to:

$$\sum_{j=1}^p \phi_{j1}^2 = 1 \quad (\text{Scaling constraint})$$

- ▶ For further components k , subtract the first $k - 1$ components from the data X (residualization), then repeat the maximization
- ▶ At most as many components as data variables p
- ▶ Each successive component explains a decreasing proportion of the variance in the data
- ▶ Information loss when using fewer components to represent data

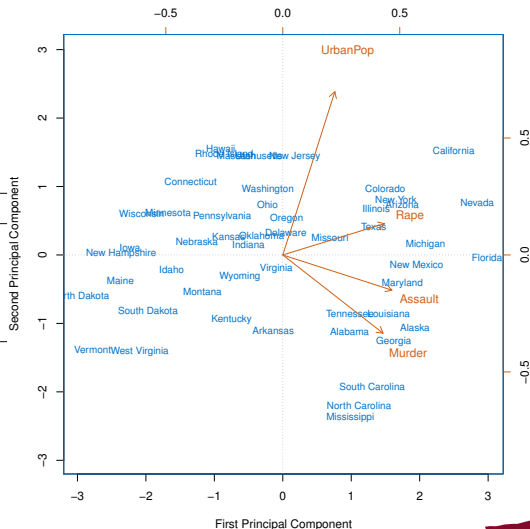
Tips

- ▶ Scale data prior to PCA
- ▶ Principle component signs can be "flipped" (arbitrarily)

PCA – Example and Biplot

	PC1	PC2
Murder	.536	-0.418
Assault	.583	-0.188
UrbanPop	.278	0.873
Rape	.543	0.167

Source: ISLR2 Table 12.1



Source: ISLR2 Figure 12.1

- ▶ Each PC is an **eigenvector** of the data correlation matrix:

$$V^{-1}CV = \Lambda$$

where V are the eigenvectors, C is the correlation matrix, and Λ is a diagonal matrix of eigenvalues

- ▶ The **proportion of variance explained** f_k by each PC k is proportional to the corresponding **eigenvalue** λ_k :

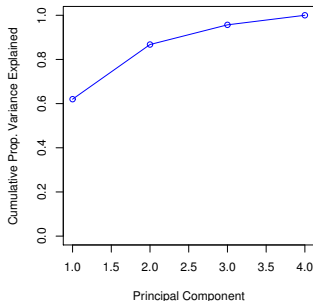
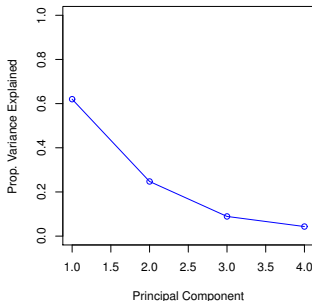
$$f_k = \frac{\lambda_k}{\sum_{j=1}^p \lambda_j}$$

- ▶ The cumulative proportion of variance F_k explained by the first k PC is then:

$$F_k = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j'=1}^p \lambda_{j'}}$$

Choosing the Number of Principal Components

- ▶ Eigenvalue $\lambda > 1$
- ▶ Cumulative explained variance greater than threshold
- ▶ Cross-validation to find optimal K (lowest test error) in a linear regression or classification model
- ▶ "Eyeballing" the screeplot



Source: ISLR2 Figure 12.3

PCA in R

Use the `USArrests` dataset that contains data on the arrests (per 100,000 residents) for various violent crimes as well as the percentage of urban population in the 50 states of the US.

R

```
library(ISLR2)
?USArrests
summary(USArrests)

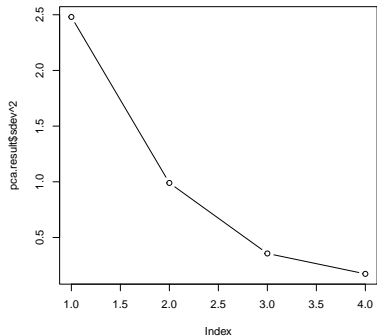
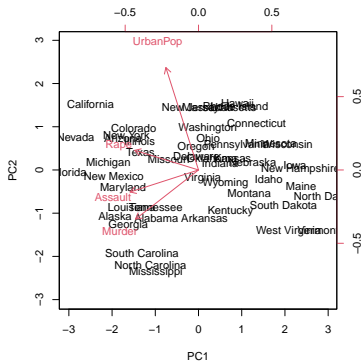
# PCA using prcomp()
# Scaling is generally a good idea
pca.result <- prcomp(USArrests, scale=TRUE)

# Print the component loadings
pca.result$rotation

# Biplot for components 1 and 2
biplot(pca.result, scale=0)

# Explained variance for each component
pca.result$sdev^2
# Scree plot (both points and lines)
plot(pca.result$sdev^2, type='b')
```

Biplot and Screeplot:



continued ...

— R —

```
# Proportion of variance explained
pve <- pca.result$sdev^2 / sum(pca.result$sdev^2)

# Cumulative sum of variance explained
plot(cumsum(pve), type='b')

# Eigen-decomposition of correlation matrix
e <- eigen(cor(USArrests))
# Compare values and vectors to prcomp results
e$values
e$vectors

# Print the component scores themselves
# For further use in regression, etc.
head(pca.result$x)
```

Hands-On Exercises – PCA

The `Boston` dataset in the `ISLR2` library describes house prices in the different suburbs of Boston. Use PCA to reduce the number of dimensions for this dataset:

- 1 Use the `prcomp` function to perform a PCA on the centered and standardized data. Limit yourself to quantitative inputs.
- 2 Produce a biplot of the first two components
- 3 Provide the proportion of variance explained by each component
- 4 How many components would you retain? Why? How much of the total variance would this explain?
- 5 Based on the loadings, can you ascribe meaning to the components? What do they represent?

Hands-On Exercises – PCA

The `Harmann74.cor` dataset in the `datasets` library contains the results of 24 psychological tests given to 145 school children. Use PCA to reduce the number of dimensions for this dataset:

- 1 Use the `prcomp` function to perform a PCA on the centered and standardized data. Limit yourself to quantitative inputs.
- 2 Produce a biplot of the first two components
- 3 Provide the proportion of variance explained by each component
- 4 How many components would you retain? Why? How much of the total variance would this explain?
- 5 Based on the loadings, can you ascribe meaning to the components? What do they represent?

Hands-On Exercises – PCA

The `Hitters` dataset in the `ISLR2` library contains the salary of 322 baseball players and season statistics. Use `salary` as the target variable and all other numerical variables as predictors.

- 1 Use PCA to reduce the number of dimensions for the predictors.
- 2 Retain the first principal component.
- 3 Estimate and cross-validate a regression model using the first PC as predictor. What is the training and validation error?
- 4 Repeat steps (1) to (3), retaining 2, 3, \dots , all components
- 5 Plot the training and validation error against the number of components. Describe and discuss your results.

Clustering

Goals

- ▶ Form homogenous subgroups of data
- ▶ Based on *similarity* of (or *distance* between) observations
- ▶ Discover "structure" in the data
- ▶ Clustering observations based on features, or clustering features based on observations (transpose of data matrix)

K-Means Clustering

- ▶ Number of clusters K is given

Hierarchical Clustering

- ▶ Unknown or variable number of clusters

K-Means Clustering

- ▶ Minimize within-cluster variation $W(C_i)$:

$$\min_{C_i} \left\{ \sum_{k=1}^K W(C_k) \right\}$$

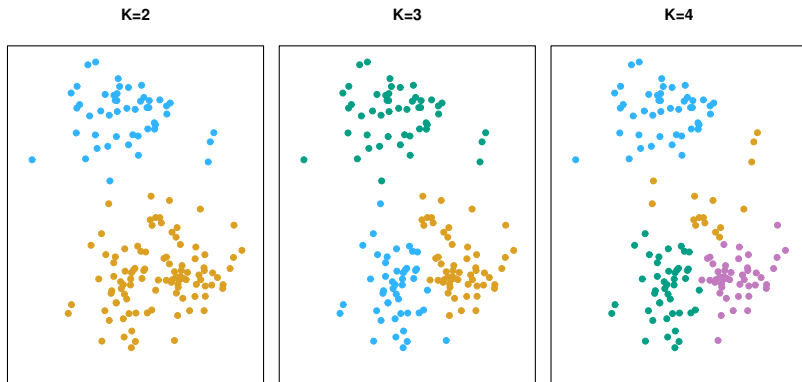
- ▶ Squared Euclidean Distance
 - ▶ Between every pair of observations in the cluster (equation 1)
 - ▶ Between every observation and the cluster **centroid** ("mean") (equation 2)

$$W(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \quad (1)$$

$$= 2 \sum_{i \in C_k} \sum_{j=1}^p (x_{ij} - \bar{\mu}_{kj})^2 \quad (2)$$

- ▶ Only for quantitative variables

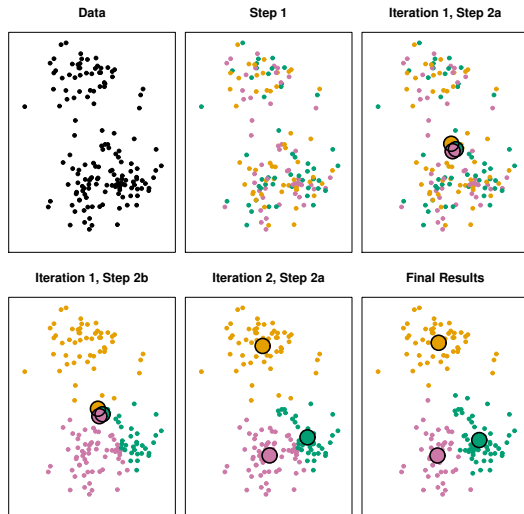
K-Means Clustering



Source: ISLR2 Figure 12.7

K-Means Clustering – Iterative Cluster Assignment

- 1 Randomly assign each observation to a cluster
- 2 Iterate until cluster assignments are stable
 - 2.1 Compute cluster means / centroid
 - 2.2 Assign each observation to cluster with closest centroid



Source: ISLR2 Figure 12.8

K-Means Clustering – Randomized Starting

- ▶ Different random initial starting clusters lead to different (suboptimal) solutions
- ▶ Run algorithm multiple times and select solution with lowest objective value



Source: ISLR2 Figure 12.9

K-Means Clustering in R

Simulated example:

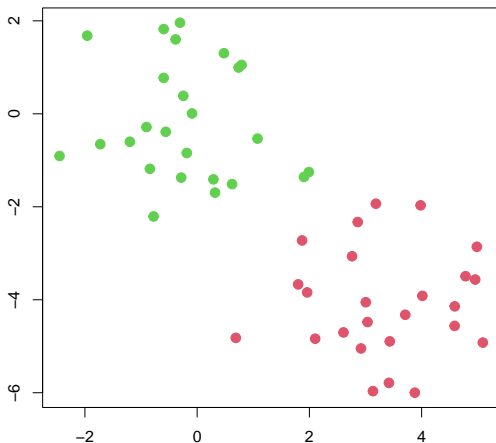
```
_____ R _____  
# Set RNG seed for replicability  
set.seed(2)  
  
# Create a 50 x 2 matrix of random variables  
# Normally distributed, with 0 mean and SD=1  
x <- matrix(rnorm(n=50*2, mean=0, sd=1), ncol=2)  
  
# Clearly separate the first 25 points by  
# shifting their coordinates  
x[1:25, 1] <- x[1:25, 1] + 3  
x[1:25, 2] <- x[1:25, 2] - 4  
  
# Cluster into 2 clusters, performing  
# 20 random starting assignments  
km.result <- kmeans(x, 2, nstart=20)
```

continued ...

```
_____ R _____  
# Results show cluster means, cluster  
# assignments, and sums of squares (distances)  
# within and between  
km.result  
# Those values are also available in  
# the result object  
names(km.result)  
  
# Plot the color-coded points  
plot(x, col=(km.result$cluster+1),  
      main = 'K-Means Clustering Results with K=2',  
      xlab = '', ylab='', pch=20, cex=2)
```

K-Means Clustering in R [cont'd]

K-Means Clustering Results with K=2



Hands-On Exercises – K-Means Clustering

The `Boston` dataset in the `ISLR2` library describes house prices in the different suburbs of Boston. Use K-Means Clustering to identify sets of similar suburbs using only the numerical variables in the data set.

- 1 Use the `kmeans` function to perform a cluster analysis, using multiple starting assignments. Limit yourself to quantitative inputs.
- 2 Use different numbers of clusters k and identify which value of k gives you the best results. Justify your choice.
- 3 Scale the data so that each variable has the same variance or standard deviation, but do not change the variable means.
- 4 Repeat the cluster analysis with the best value of k and compare results.

Hands-On Exercises – K-Means Clustering

The `Hitters` dataset in the `ISLR2` library contains the salary of 322 baseball players and season statistics. Use K-Means Clustering to identify sets of similar players, using only the numerical variables in the data set.

- 1 Use the `kmeans` function to perform a cluster analysis, using multiple starting assignments. Limit yourself to quantitative inputs.
- 2 Use different numbers of clusters k and identify which value of k gives you the best results. Justify your choice.
- 3 Scale the data so that each variable has the same variance or standard deviation, but do not change the variable means.
- 4 Repeat the cluster analysis with the best value of k and compare results.

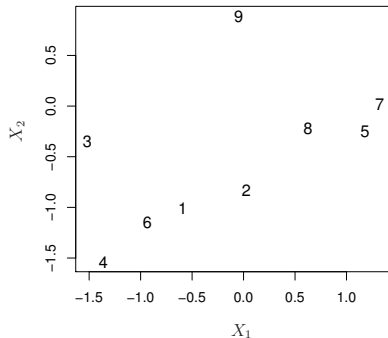
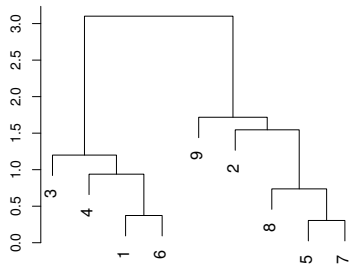
Bottom-Up / Agglomerative Clustering

- 1 Begin with n observations and a dissimilarity or distance metric
- 2 Treat each observation as its own cluster
- 3 Repeat $n - 2$ times:
 - 3.1 Calculate dissimilarities or distances between all pairs of clusters
 - 3.2 Identify the pair of clusters that are least dissimilar (most similar)
 - 3.3 "Fuse" or merge these two clusters

Hierarchical Clustering

Dendrogram

- Shows what clusters were fused at what dissimilarity



Source: ISLR2 Figure 12.12

Key Decisions




- ▶ How to measure dissimilarity/distance between observations?
- ▶ How to measure dissimilarity between clusters ("**linkage**")?
- ▶ How many clusters should we have?

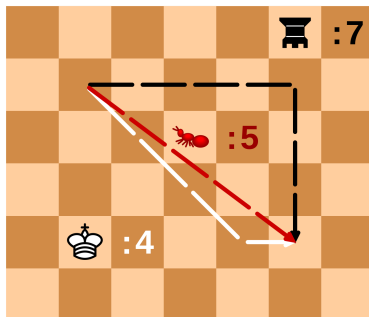
Hierarchical Clustering – Common Distance Metrics

Common Distance Metrics or "Norms"

Taxicab / Manhattan	$\ q - p\ _1$	$\sum_i q_i - p_i $
Euclidean	$\ q - p\ _2$	$\sqrt{\sum_i (q_i - p_i)^2}$
Minkowski	$\ q - p\ _p$	$\left(\sum_i q_i - p_i ^p\right)^{\frac{1}{p}}$
Chebyshev	$\ q - p\ _\infty$	$\lim_{p \rightarrow \infty} \left(\sum_i q_i - p_i ^p\right)^{\frac{1}{p}} = \max_i (q_i - p_i)$
	$\ q - p\ _{-\infty}$	$\lim_{p \rightarrow -\infty} \left(\sum_i q_i - p_i ^p\right)^{\frac{1}{p}} = \min_i (q_i - p_i)$

Hierarchical Clustering – Common Distance Metrics

1	1	1	$\sqrt{2}$	1	$\sqrt{2}$	2	1	2
1		1	1		1	1		1
1	1	1	$\sqrt{2}$	1	$\sqrt{2}$	2	1	2
Chebyshev			Euclidean			Taxicab		



https://commons.wikimedia.org/wiki/File:Minkowski_distance_examples.svg

Hierarchical Clustering – Common Linkage Criteria

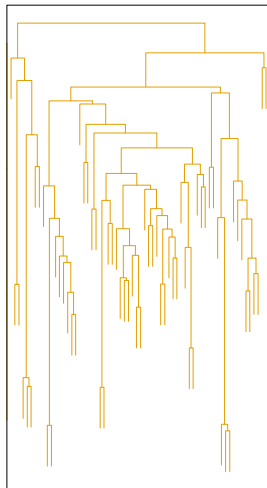
Common Linkages

Single	$d_{SL}(G, H) = \min_{i \in G, i' \in H} d_{i,i'}$
Complete	$d_{CL}(G, H) = \max_{i \in G, i' \in H} d_{i,i'}$
Average	$d_{AL}(G, H) = \text{mean}_{i \in G, i' \in H} d_{i,i'}$

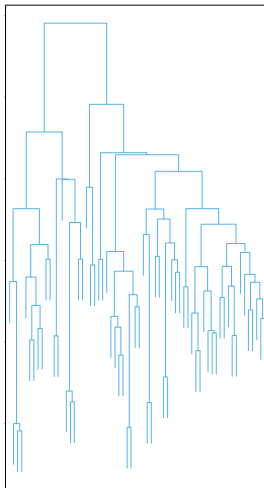
There are many other linkage functions: https://en.wikipedia.org/wiki/Hierarchical_clustering

Hierarchical Clustering – Common Linkage Criteria

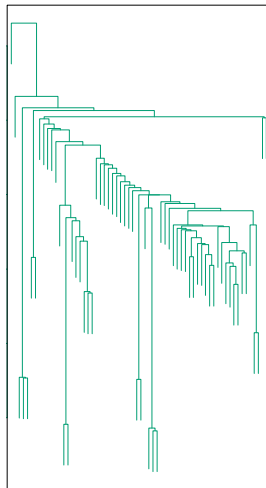
Average Linkage



Complete Linkage



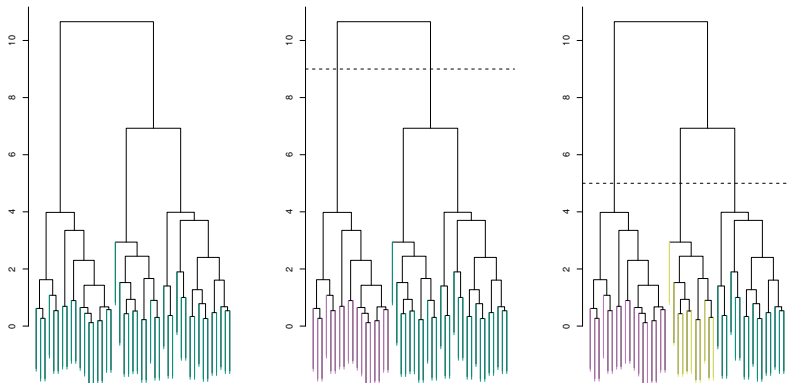
Single Linkage



Source: ISLR2 Figure 12.14

Hierarchical Clustering – How Many Clusters?

- "Cut" the dendrogram at a dissimilarity value



Source: ISLR2 Figure 12.11

Hierarchical Clustering in R

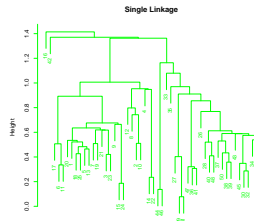
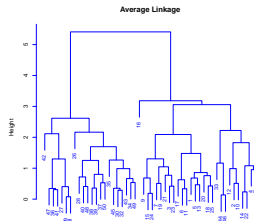
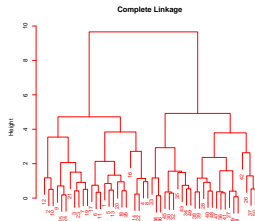
R

```
# The dist() function calculated distances  
# according to a variety of metrics/norms  
euclid.dist <- dist(x, method='euclidean')  
pnorm.dist <- dist(x, method='minkowski', p=3)  
manh.dist <- dist(x, method='manhattan')  
max.dist <- dist(x, method='maximum')  
  
# Use the hclust() function with a distance metric  
hc.complete <- hclust(euclid.dist, method='complete')  
hc.single <- hclust(euclid.dist, method='single')  
hc.average <- hclust(euclid.dist, method='average')
```

continued ...

```
_____ R _____  
# Plot the dendrograms in a single plot  
par(mfrow = c(1, 3))  
plot(hc.complete , col='red',  
     main = "Complete Linkage",  
     xlab = "", sub = "", cex = .9)  
  
plot(hc.average , col='blue',  
     main = "Average Linkage",  
     xlab = "", sub = "", cex = .9)  
  
plot(hc.single , col='green',  
     main = "Single Linkage",  
     xlab = "", sub = "", cex = .9)
```

Hierarchical Clustering in R [cont'd]



Cutting the tree and identifying clusters:

```
_____ R _____  
# Cut by number of groups/clusters  
cutree(hc.complete, k=4)  
# Cut by height (dissimilarity)  
cutree(hc.complete, h=6)
```

Hands-On Exercises – Hierarchical Clustering

The `Boston` dataset in the `ISLR2` library describes house prices in the different suburbs of Boston. Use Hierarchical Clustering to identify sets of similar suburbs using only the numerical variables in the data set.

- 1 Use the `hclust` function to perform a cluster analysis, exploring different distance metrics and linkage functions.
- 2 Examine the dendrograms and identify which combination of distance metric and linkage function gives you the "cleanest" separation of clusters.
- 3 How many factors k would you retain?
- 4 Using this value for k , perform a K-Means Clustering and compare the results.

Hands-On Exercises – Hierarchical Clustering

The `Hitters` dataset in the `ISLR2` library contains the salary of 322 baseball players and season statistics. Use Hierarchical Clustering to identify sets of similar players, using only the numerical variables in the data set.

- 1 Use the `hclust` function to perform a cluster analysis, exploring different distance metrics and linkage functions.
- 2 Examine the dendrograms and identify which combination of distance metric and linkage function gives you the "cleanest" separation of clusters.
- 3 How many factors k would you retain?
- 4 Using this value for k , perform a K-Means Clustering and compare the results.

Hands-On Exercises – Hierarchical Clustering

The `Auto` dataset in the `ISLR2` library contains information on 392 vehicles. Use Hierarchical Clustering to identify sets of similar vehicles, using only the numerical variables in the data set.

- 1 Use the `hclust` function to perform a cluster analysis, exploring different distance metrics and linkage functions.
- 2 Examine the dendrograms and identify which combination of distance metric and linkage function gives you the "cleanest" separation of clusters.
- 3 How many factors k would you retain?
- 4 Using this value for k , perform a K-Means Clustering and compare the results.