R Lecture #3

November 6th, 2017

Hyeokkoo Eric Kwon (KAIST)

hkkwon7@business.kaist.ac.kr

Graphical Illustrations

Scatter Plot

plot(x, y) :

- **x** : the *x* coordinates of points in the plot
- **y** : the *y* coordinates of points in the plot

1. *install.packages("mlbench")*
2. *library(mlbench)*
3. *data(Ozone)*

Install & Load Ozone Data

4. *plot(Ozone$V8, Ozone$V9)*

Scatter Plot (Cont'd)

plot(x, y, xlab, ylab, main, pch, cex, col, xlim, ylim) :

- **xlab**, **ylab** : a title for the *x*, *y* axis
- **main** : an overall title for the plot
- **pch** : symbol for scatters
- **cex** : size of scatters
- **col** : color of scatters
- **xlim**, **ylim** : range of the *x*, *y* axis

Scatter Plot (Cont'd)

1. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature")*
2. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature", **main** = "Ozone")*

3. *help(pch)*
4. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature", **main** = "Ozone", **pch** = 13)*
5. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature", **main** = "Ozone", **pch** = "X")*

Scatter Plot (Cont'd)

cex : size of scatters

- Default = 1
- The amount by which plotting text and symbols should be magnified relative to the default

1. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature", **main** = "Ozone", **pch** = 13, **cex** = 1)*
2. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature", **main** = "Ozone", **pch** = 13, **cex** = 0.1)*
3. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature", **main** = "Ozone", **pch** = 13, **cex** = 2)*

Scatter Plot (Cont'd)

col: color of scatters

- Name or RGB

1. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature", **main** = "Ozone", **pch** = 13, **cex** = 1, **col** = "blue")*
2. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature", **main** = "Ozone", **pch** = 13, **cex** = 1, **col** = "#FF0000")*
3. *plot(Ozone$V8, Ozone$V9, **xlab** = "Sandburg Temperature", **ylab** = "El Monte Temperature", **main** = "Ozone", **pch** = 13, **cex** = 1, **col** = "#FF0000", **col.axis**="blue", **col.lab**="green")*

Scatter Plot (Cont'd)

plot(x, y, type) :

- **type :** what type of plot should be drawn
  1. *"p" for points,*
  2. *"l" for lines,*
  3. *"b" for both,*
  4. *"o" for both 'overplotted',*
  5. *"h" for 'histogram' like (or 'high-density') vertical lines,*
  6. *"s" for stair steps,*
  7. *"S" for other steps, see 'Details' below,*

Scatter Plot (Cont'd)

1. *data(cars)* **Load Speed and Breaking Distance Data**
2. *str(cars)*

3. *plot(cars$speed, cars$dist)*
4. *plot(cars)*

5. *plot(cars, type = "l")*
6. *plot(cars, type = "b")*
7. *plot(cars, type = "h")*

Line Plot

1. *plot(cars, type = "l")* **Duplicate Observations (i.e., Vertical Line)**

2. *apply(cars, 2, mean)*
3. *tapply(cars$dist, cars$speed, mean)*

   Object Group

4. *plot(tapply(cars$dist, cars$speed, mean), type="l")*

Line Plot (Cont'd)

plot(x, y, lty) :

- **lty:** The line type, line types can either be specified as an integer

1. *0 = blank*
2. *1 = solid (default)*
3. *2 = dashed*
4. *3 = dotted*
5. *4 = dotdash*
6. *5 = longdash*
7. *6 = twodash*

8. *plot(cars, **type** = "l", **lty** = "dashed")*
9. *plot(cars, **type** = "l", **lty** = 5)*

Graph Arrangement

par(mfrow = c(nr, nc))

- Subsequent figures will be drawn in an **nr**-by-**nc** array on the device

1. *par(mfrow = c(2, 3))*

2. *plot(cars, **type** = "l", **lty** = 1)*
3. *plot(cars, **type** = "l", **lty** = 2)*
4. *plot(cars, **type** = "l", **lty** = 3)*
5. *plot(cars, **type** = "l", **lty** = 4)*
6. *plot(cars, **type** = "l", **lty** = 5)*
7. *plot(cars, **type** = "l", **lty** = 6)*

Add Points to Existing Graph

points(x, y, ...)

1. *par(mfrow=c(1,1))*
2. ***plot**(iris$Sepal.Width, iris$Sepal.Length, **cex** = 0.5, **pch** = 20, **col** = "black")*
3. ***plot**(iris$Petal.Width, iris$Petal.Length, **cex** = 1, **pch** = "X", **col** = "red")*

4. ***plot**(iris$Sepal.Width, iris$Sepal.Length, **cex** = 0.5, **pch** = 20, **col** = "black")*
5. ***points**(iris$Petal.Width, iris$Petal.Length, **cex** = 1, **pch** = "X", **col** = "red")*

6. ***plot**(NULL, **xlim** = c(0,5), **ylim** = c(0,10))*
7. ***points**(iris$Sepal.Width, iris$Sepal.Length, **cex** = 0.5, **pch** = 20, **col** = "black")*
8. ***points**(iris$Petal.Width, iris$Petal.Length, **cex** = 1, **pch** = "X", **col** = "red")*

Add Text to Existing Graph

text(x, y, labels, adj)

- **labels :** text to be written (default = row name)
- **adj :** one or two values which specify the *x* and *y* adjustment of the labels.

1. *plot(NULL, **xlim** = c(4,6), **ylim** = c(4,6))*
2. *text(5, 5, "X")*
3. *text(5, 5, "A", adj = c(0,0))*
4. *text(5, 5, "B", adj = c(0,1))*
5. *text(5, 5, "C", adj = c(1,0))*
6. *text(5, 5, "D", adj = c(1,1))*

7. *sample <- cars[1:3,]*
8. *rownames(sample) <- c("BMW", "Benz", "Audi")*
9. *plot(sample, xlim=c(3,8), ylim=c(0,12))*
10. *text(sample$speed, sample$dist)*

11. *plot(sample, xlim=c(3,8), ylim=c(0,12))*
12. *text(sample$speed, sample$dist, **adj=c(0,0)**)*

13. *plot(sample, xlim=c(3,8), ylim=c(0,12))*
    *text(sample$speed, sample$dist, **rownames(sample)**, adj=c(0,0))*

Legend

legend(x, y=NULL, legend, ...):

- **x** : the *x* coordinates of legend (or bottom, top, left, right, center)
- **legend** : character to appear in the legend

1. *plot(iris$Sepal.Width, iris$Sepal.Length, pch = 20, col = "black")*
2. *points(iris$Petal.Width, iris$Petal.Length, pch = 10, col = "red")*

3. *legend("topright", legend = c("Sepal", "Petal"), pch = c(20, 10), col = c("black", "red"), bg = "gray")*

Histogram

hist (x, breaks, freq,...):

- **x** : a vector of values for which the histogram is desired
- **breaks**: a single number giving the number of cells for the histogram
- **freq :** TRUE **=** frequency, FALSE = density

1. *hist(iris$Sepal.Width, freq = TRUE)*
2. *hist(iris$Sepal.Width, freq = FALSE)*

3. *hist(iris$Sepal.Width)*
4. *hist(iris$Sepal.Width, breaks = 3)*

Vulnerable to # of cells

5. ***plot(density(****iris$Sepal.Width****))***

Bar and Pie Chart

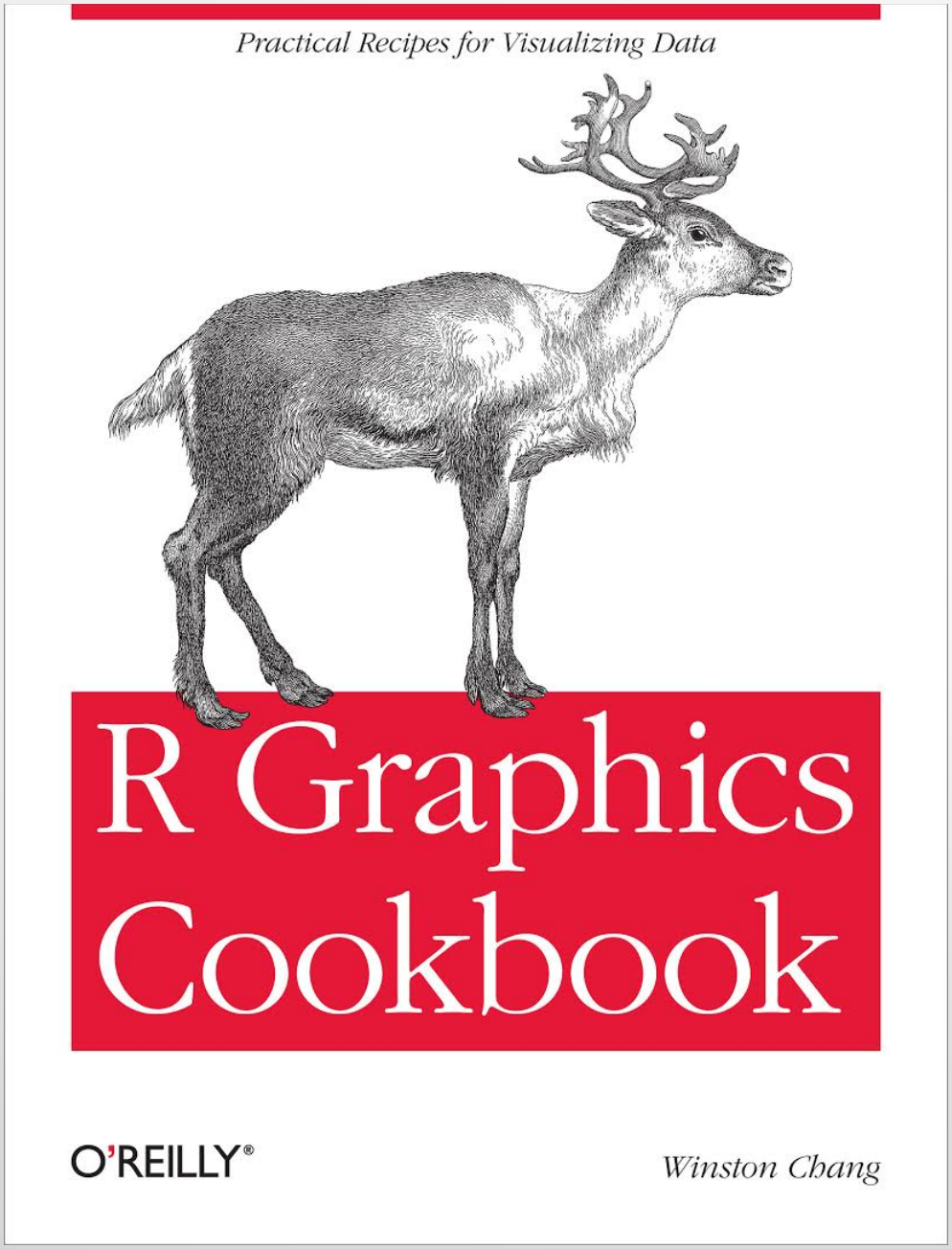
barplot(height, ...)

- **height** : either a vector or matrix of values describing the bars

pie(x)

- **x** : values displayed as the areas of pie slices

1. *sample <- c(Dog = 60, Cat = 40, Bird = 80)*
2. *barplot(sample)*
3. *pie(sample)*


Practical Recipes for Visualizing Data
R Graphics Cookbook
O'REILLY®
Winston Chang

Clustering Analysis

Cluster Analysis

- Cluster: a collection of data objects
  1. *Similar to one another within the same cluster*
  2. *Dissimilar to the objects in other clusters*
- Cluster analysis
  1. *Grouping a set of data objects into clusters*
- Clustering is unsupervised classification: no predefined classes
- A good clustering method will produce high quality clusters with
  1. *high intra-class similarity*
  2. *low inter-class similarity*

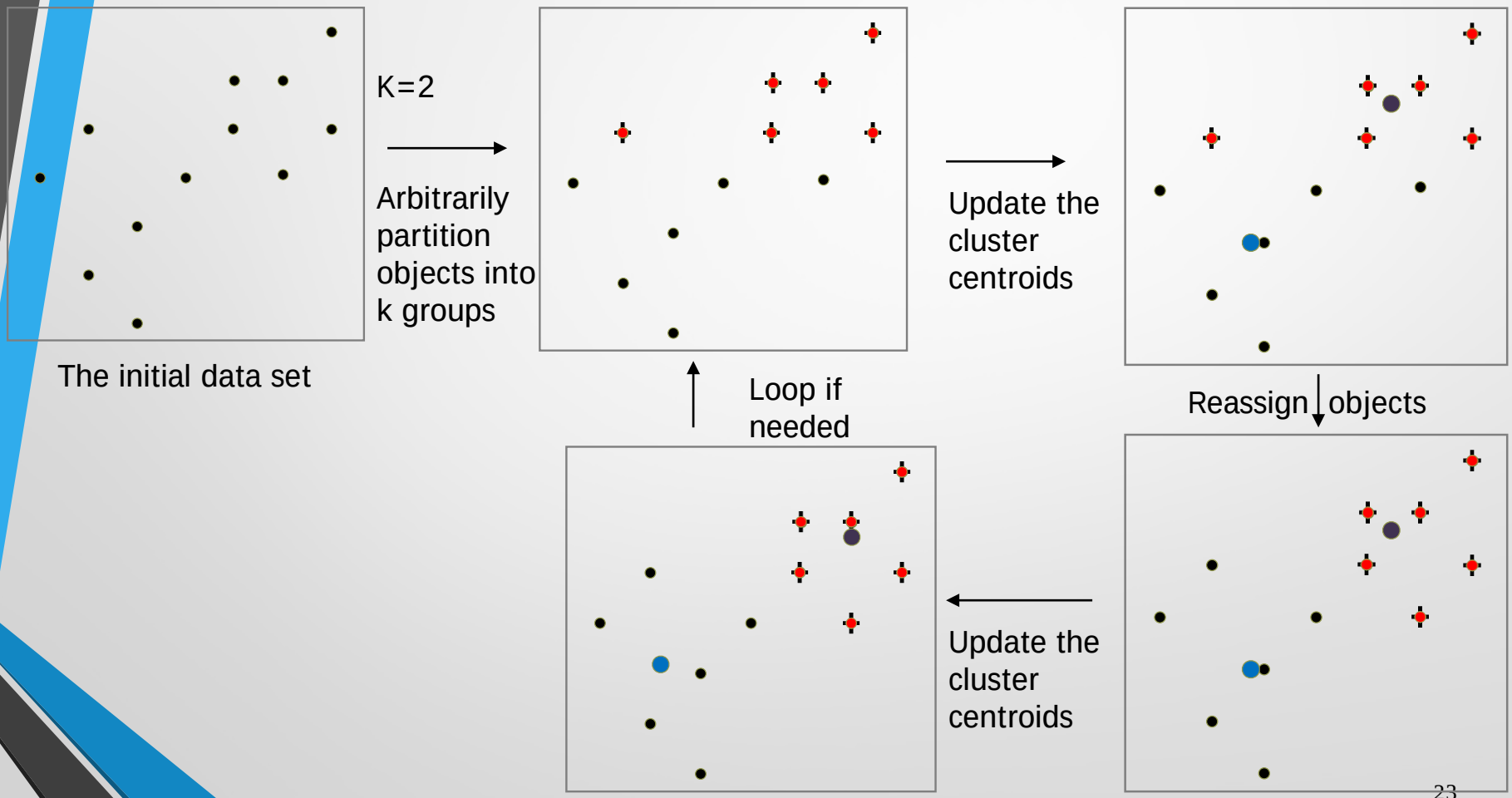
Examples of Clustering Applications

- **Marketing**: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- **Web**: Cluster groups of users based on their access patterns on webpages
- **Insurance**: Identifying groups of motor insurance policy holders with a high average claim cost
- **City-planning**: Identifying groups of houses according to their house type, value, and geographical location
- **Earth-quake studies**: Observed earth quake epicenters should be clustered along continent faults

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  1. *Partition objects into k nonempty subsets*
  2. *Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., mean point, of the cluster)*
  3. *Assign each object to the cluster with the nearest seed point*
  4. *Go back to Step 2, stop when no more new assignment*

The *K-Means* Clustering Method (Cont'd)



Exploration of Data

The iris dataset contains data about sepal length, sepal width, petal length, and petal width of flowers of different species.

1. *iris*
2. *plot(iris$Petal.Width[1:50], iris$Petal.Length[1:50], pch = 16, col = "red", xlim=c(0,3), ylim=c(0,8))*
3. *points(iris$Petal.Width[51:100], iris$Petal.Length[51:100], pch = 16, col = "blue")*
4. *points(iris$Petal.Width[101:150], iris$Petal.Length[101:150], pch = 16, col = "green")*

Petal.Length and Petal.Width were similar among the same species but varied considerably between different species

The *K-Means* Clustering in R

kmeans(x, centers, nstart)

- **x :** numeric matrix of data
- **centers :** either the number of clusters, say k, or a set of initial cluster centres. If a number, a random set of rows in x is chosen as the initial centres.
- **nstart** : if centers is a number, how many random sets should be chosen?

1. *irisCluster <- kmeans(iris[, 3:4], 3, nstart = 20)*
2. *irisCluster*
3. *table(irisCluster$cluster, iris$Species)*
4. *iris[,6] <- irisCluster$cluster*

We know the number of clusters

The *K-Means* Clustering in R (Cont'd)

```
> irisCluster
K-means clustering with 3 clusters of sizes 48, 50, 52

Cluster means:
  Petal.Length Petal.Width
1     5.595833    2.037500
2     1.462000    0.246000
3     4.269231    1.342308

Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [25] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [49] 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [73] 3 3 3 3 3 1 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3
 [97] 3 3 3 3 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 3
[121] 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1
[145] 1 1 1 1 1 1

Within cluster sum of squares by cluster:
[1] 16.29167  2.02200 13.05769
 (between_SS / total_SS =  94.3 %)

Available components:

[1] "cluster"      "centers"      "totss"
[4] "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

The 94.3 % is a measure of the total variance in your data set that is explained by the clustering.

k-means minimize the within group dispersion and maximize the between-group dispersion.

When We Don’t Know the Number of Clusters

```
1. mydata <- iris[,3:4]
2. wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))

3. for (i in 2:15) wss[i] <- sum(kmeans(mydata, centers=i)$withinss)
4. plot(1:15, wss, type="b", xlab="Number of Clusters",
5.     ylab="Within groups sum of squares",
6.     main="Assessing the Optimal Number of Clusters with the Elbow Method",
7.     pch=20, cex=2)
```