기본 패키지	▪ base, utils, stats, graphics
ETL	▪ foreach, stringr ▪ data.table, sqldf, plyr, dplyr, reshape, reshape2 ▪ Amelia, DMwR / cvTools, doBy, sampling, caret, DMwR
Description	▪ Hmisc, caret, FSelector, rpart
데이터 타입	▪ vector : vector, factor, ordered, matrix, array ▪ list : list, data.frame ▪ 속성 : names, dim, dimnames, row.names, class, levels
데이터 속성 조회	▪ class, unclass, attributes, attr, str, mode, typeof ▪ library(help="~"), methods, args, example
인코딩 (KoNLP)	▪ localeToCharset(), Encoding(~)
I/O	▪ read.table("~.csv", header=T, sep=",", stringsAsFactors=F, na.strings=c("NIL"), comment.char="#", fileEncoding="UTF-8", encoding="CP949") ▪ write.table(data, file="~.csv", row.names=F, sep=",", append=F, quote=F, fileEncoding="UTF-8") ▪ save(data, "~.RData"), load("~.RData")
데이터 결합	▪ rbind(~, ~), cbind(~, ~), merge(~, ~, all=T)
변수 추가 (plyr)	▪ transform(data, var1 = ~), mutate(data, var2 = ~)
반복 실행 (foreach)	▪ foreach(i = 1:5, .combine=rbind) %do% { ~ }
결측값 처리 (Amelia)	▪ is.na, complete.cases, na.omit, missmap, na.rm ▪ zztemp <- amelia(data, m=5, ts="year", cs="country")' ▪ data$field <- zztemp$imputation[[5]]$field
이상값 처리 (DMwR)	▪ factor <- lofactor(data, k=5) ▪ idx <- order(factor, decreasing=T)[1:5] ▪ pairs(data[, 1:4], pch=pch, col=col) ▪ biplot(prcomp(data[, 1:4]), cex=.8, xlabs=labels)
행별 함수 (plyr)	▪ adply(data, 1, function(row) { ~ })
열별 함수 (plyr)	▪ adply(data, 2, function(col) { ~ })
그룹별 함수 적용	▪ aggregate(value ~ group, data, function(group) { ~ })
값별 함수 적용	▪ sapply(data, function(val) { ~ })
데이터 구조 변환 (reshape)	▪ melt(data, id=c("~"), na.rm=T) ▪ cast(data, y ~ x ~ z, func, subset=variable=="~") ▪ cast(data, y ~ . | z, func)
샘플링 (sampling)	▪ idx <- sample(2, nrow(data), replace=T, prob=c(0.7, 0.3)) ▪ idx <- strata(c("Species"), size=c(3, 4, 5), method="srswor", data=data) #--- srswor, srswr (복원)

Description (Hmisc)	▪ mean, median, sd, var, quantile, max, min ▪ cov, cor, rcorr(x, y, type="spearman")$r #--- pearson
변수 선택 (caret, FSelector, rpart)	▪ nearZeroVar(data, saveMetrics=T) # nzv=true인 변수 제거 ▪ findCorrelation(cor(data)) #--- 표시된 열 제거 ▪ m <- linear.correlation(y ~ x1 + x2, data=data) ▪ m[order(-m), , drop=F] #--- rank.correlation, chi.squared #--- attr_importance 값이 높은 변수부터 제거 ▪ m <- rpart(y ~ x1 + x2, data=data] ▪ varImp(m) #--- overall 값이 높은 변수부터 제거 ▪ m <- prcomp(data, scale=T) #--- 표준 점수로 주성분분석 ▪ m <- princomp(data, cor=T) #--- 상관 행렬로 주성분분석 ▪ summary(m), predict(m), plot(m, type="l"), biplot(m) ▪ m <- cmdscale(data) ▪ plot(m[, 1], m[, 2], type="l") ▪ text(m[, 1], m[, 2], rownames(m), cex=0.8)
분류 분석 (Classification) MASS, party, rparty, randomForest, ROCR, caret, lattice	▪ m <- lda(Species ~ ., data=train); #--- qda, ctree ▪ m <- cforest(Species ~ ., data=train, control=cforest_unbiased(mtry = 3)); ▪ m <- randomForest(Species ~ ., data=train, ntree=100, proximity=TRUE); ▪ pred <- predict(result$m), pred$class ▪ plot(m, type="simple) ▪ margin(m, test$cls), varImpPlot(m), importance(m)
예측 분석 (Estimation) TTR, forecast, mgcv	▪ m <- lm(left ~ upper, data = data) ▪ step(lm(left ~ 1, data=data), scope=list(lower=~ 1, upper=~ upper), direction="forward") #--- 전진선택법 ▪ step(lm(left ~ upper, data=data), scope=list(lower=~ 1, upper=~ upper), direction="backward") #--- 후진소거법 ▪ step(lm(left ~ 1, data=data), scope=list(lower=~ 1, upper=~ upper), direction="both") #--- 단계적방법
	▪ auto.arima(data) ▪ SMA(data01, n = 2) #--- 평활 ▪ diff(data01, differences = 2) #--- 차분 ▪ pacf(data01, lag.max=10) ▪ acf(data01, lag.max=10, Plot=TRUE) ▪ data02 <- arima(data, order=c(2, 0, 0)) #--- 모델 보정 ▪ data03 <- forecast.Arima(data02, h=20) #--- 예측 ▪ m <- decompose(data) ▪ mx, mtrend, m$seasonal, m$random
군집 분석 (Clustering) cluster, fpc, class	▪ m <- kmeans(data, centers = 3); ▪ m <- hclust(dist(data, method = "euclidean")^2, method = "ward"); pred <- cutree(m, k = 3) type : complete(Default), single, ward, average subtype : euclidean(D), manhattan, canberra,minkowski ▪ m <- pam(data, 3); m <- fanny(data, 2); m$cluster

군집 분석 (Clustering)

cluster, fpc, class

- 군집 개수 산정

```
findCluster = function(data = iris[, 1:4], groups = 1:10) {
  models <- 0
  for (i in groups) {
    models[i] <- sum(kmeans(data, centers = i)$withinss)
  }
  plot(groups, models, type="b", xlab="수", ylab="제곱합")
}
```

연관 분석 (Association)

arules, arulesViz, pmml

- data <- as(data, "transactions")
- length, size, summary
- inspect, transactionInfo, itemsetInfo, itemInfo
- as(data, "list");
- format(as.POSIXlt(transactionInfo(data[1])[["TimeStamp"]]), "%Y-%m-%d %H:%M:%S")
- itemFrequencyPlot(data, support = 0.1, cex.names = 0.8)
- m <- apriori(data, parameter = list (support = 0.01, confidence = 0.6)) #--- 최소 지지도, 최소 신뢰도
- summary, length
- inspect(sort(m[1:10], by = "lift"))
- small <- subset(m, subset = rhs %in% "income=small" & lift > 1.2)

최적화 (Optimization)

lpSolve, lpSolveAPI

- m <- make.lp(0, 4)
- lp.control(m, sense="max")
- set.objfn(m, c(4, 6, 7, 8))
- add.constraint(m, c(1, 1.5, 2, 3), "<=", 800)
- solve(m), get.objective(m), get.variables(m)

텍스트 마이닝 (Text Mining)

tm, wordcloud

- getReaders(), getTransformations()
- 파일 읽기

```
finp <- file("data/~.txt", encoding = "UTF-8")
lines <- readLines(finp)
close(finp)
```

- doc <- Corpus(VectorSource(lines))
- summary, inspect
- doc <- tm_map(doc, as.PlainTextDocument)
- as.PlainTextDocument, stripWhitespace, tolower, removePunctuation, removeWords (stopwords("english")), stripWhitespace, stemDocument
- m <- DocumentTermMatrix(doc)
- m <- removeSparseTerms(m, 0.70)
- m <- DocumentTermMatrix(doc, list(dictionary = dic))
- m <- t(m), inspect(m)

- findFreqTerms(m, 10, 15)
- findAssocs(m, "oil", 0.65)
- frequency <- colSums(data)
- wordcloud(names(frequency), as.numeric(frequency), colors = c("green", "red"))

소셜 네트워크 분석 (Social Network Analytics)

igraph

- m <- graph.data.frame(d = data, directed = TRUE, vertices = attrs)
- g <- graph.incidence(m, mode = c("all"))
- m <- graph.adjacency(data, weight = T, mode = "undirected")
- g <- simplify(g)
- m$layout <- layout.fruchterman.reingold(m)
- vcount, ecount, V, E
- get.vertex.attribute(m, "name")
- get.edge.attribute(m, "advice_tie")
- plot(m, layout = layouts, main = "Krackhardt High-Tech Managers", edge.arrow.size = .05)
- vertex.size : color, frame, frame.color, shape, abel, label.font, label.family, label.cex, label.dist, label.color
- edge.color : arrow.width, arrow.size, arrow.width, lty, label, label.font, label.family, label.cex, label.color
- mAdvice <- delete.edges(m, E(m)[get.edge.attribute(m, name = "advice_tie") == 0])
- mAdviceNo <- delete.vertices(mAdvice, V(mAdvice)[degree(mAdvice) == 0])

- m1 <- edge.betweenness.community(m)
- m1 <- walktrap.community(m, steps = 200, modularity = TRUE)
- plot(as.dendrogram(m1, use.modularity = TRUE))
- deg_full_in <- degree(m, mode = "in")
- m1 <- closeness(m)
- sp_full_in <- shortest.paths(m, mode = "in")
- tc_full <- triad.census(m)

Visualization

ggplot2, aplpack, vcd, googleVis, shiny, caret

- main, xlab, ylab, xlim, ylim, type = p, l, s, o, b, n
- col, col.lab, col.axis, pch, axes = F

- x, y, colour, shape, size, alpha, fill, binwidth
- ggplot(data=~, aes(x=~, y=~, colour=~, group=~))
- geom_point() : smooth, line, histogram, density, bar, pointrange
- stat_bin2d(), facet_grid(Diet ~ .)
- xlab(), ylab(), coord_cartesion(xlim, ylim)
- coord_flip(), scale_x_reverse(), scale_y_reverse()
- ..count.., ..density..
- ggsave("~.png")
- pie(table(Species))
- parcoord(iris[, 1:4], main="평행좌표", col = iris$Species)
- faces(iris[, 1:4], face.type = 1, main = "안형그림")
- stars(cpi[, 2:6], locations = c(0, 0), radius = T, key.loc = c(0, 0), main = "상품성질별 소비자 물가지수")
- mosaicplot(~ Class + Survived, data=Titanic, color=TRUE)
- plot(gvisMotionChart(Fruits, idvar="Fruit", timevar="Year"))
- gvisGeoChart(data, locationvar, colorvar, sizebar)