2. 코드 및 내용(텍스트 형식)

|  |  |
| --- | --- |
| 페이지 | 코드 및 내용 |
| 34 | > install.packages("ggthemes")  > library(ggthemes) |
| 36 | #creating '객체'  > a <- 1 |
| 37 | > a  [1] 1 |
|  | > a <- 1  > b <- 2 |
|  | > a  [1] 1  > b  [1] 2 |
| 38 | > a+b  [1] 3  > a-b  [1] -1  > a\*b  [1] 2  > a/b  [1] 0.5 |
|  | > c <- "문자"  > d <- TRUE  > e <- iris |
| 39 | > c  > d  > e |
|  | > c  [1] "문자"  > d  [1] TRUE  > e  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa |
| 40 | > c <- "문자"  > d <- TRUE  > e <- iris |
| 41 | > f <- c(1,2,3) |
|  | > f  [1] 1 2 3 |
|  | > is(f)  [1] “numeric" "vector" |
| 42 | > is.vector(f)  [1] TRUE |
| 44 | > a <- c(1, 2, 3)  > a  [1] 1 2 3 |
|  | > is(a)  [1] "numeric" "vector" |
| 45 | > a <- as.integer(a)  > a  [1] 1 2 3 |
|  | > is(a)  [1] "integer" "numeric" "vector"  [4] “data.frameRowLabels" |
|  | > b <- c("문자", "형", "벡터", "입니다", "참", "쉽죠잉")  > b  [1] "문자" "형" "벡터" "입니다" "참" "쉽죠잉" |
| 46 | > is(b)  [1] "character" "vector" "data.frameRowLabels"  [4] "SuperClassMethod" |
| 48 | > factor.a <- factor(c("경영학과", "심리학과", "컴퓨터공학과"))  > factor.a  [1] 경영학과 심리학과 컴퓨터공학과  Levels: 경영학과 심리학과 컴퓨터공학과 |
|  | > is(factor.a)  [1] "factor" "integer" "oldClass"  [4] "numeric" "vector" “data.frameRowLabels" |
| 49 | > levels(factor.a) <- c(levels(factor.a), “영문학과”, “미디어학과”)  > factor.a  [1] 경영학과 심리학과 컴퓨터공학과  Levels: 경영학과 심리학과 컴퓨터공학과 영문학과 미디어학과 |
| 50 | > factor.b  [1] 경영학과 심리학과 컴퓨터공학과  Levels: 경영학과 < 심리학과 < 컴퓨터공학과 |
|  | > is(factor.b)  [1] "ordered" “factor" "oldClass" |
| 51 | > factor.c <- ordered(c("이것이", “순서형","벡터"))  > factor.c  [1] 이것이 순서형 벡터  Levels: 벡터 < 순서형 < 이것이 |
| 52 | > factor.d <- ordered(c(3,1,2))  > factor.d  [1] 3 1 2  Levels: 1 < 2 < 3 |
|  | > factor.e <- factor(c(1, 2, 3), ordered = T)  > factor.e  [1] 1 2 3  Levels: 1 < 2 < 3 |
| 53 | > factor.f <- factor(c(1,2,3), ordered = T, levels = c(3,2,1))  > factor.f  [1] 1 2 3  Levels: 3 < 2 < 1 |
| 56 | > a <- c(1, 2, 3, 4)  > b <- c(3.2, 4.5, 8.9, 1.3)  > c <- c("aa", 1, 5, 7) |
|  | > is.vector(a)  > is.vector(b)  > is.vector(c) |
|  | > is.vector(a)  [1] TRUE  > is.vector(b)  [1] TRUE  > is.vector(c)  [1] TRUE |
| 57 | > data.frame(a,b,c)  a b c  1 1 3.2 aa  2 2 4.5 1  3 3 8.9 5  4 4 1.3 7 |
|  | > df <- data.frame(a,b,c)  > df  a b c  1 1 3.2 aa  2 2 4.5 1  3 3 8.9 5  4 4 1.3 7 |
| 59 | > matrix(c(1, 2, 3, 4, 5, 6, 7, 8), nrow = 2, ncol = 4)  [,1] [,2] [,3] [,4]  [1,] 1 3 5 7  [2,] 2 4 6 8 |
|  | > matrix(c(1, 2, 3, 4, 5, 6, 7, 8), nrow = 2, ncol = 4, byrow = TRUE)  [,1] [,2] [,3] [,4]  [1,] 1 2 3 4  [2,] 5 6 7 8 |
| 60 | > a <- matrix(c(1, 2, 3, 4), nrow = 2, ncol = 2)  > b <- matrix(c(4, 3, 2, 1), nrow = 2, ncol = 2) |
|  | > a  [ ,1] [ ,2]  [1, ] 1 3  [2, ] 2 4  > b  [ ,1] [ ,2]  [1, ] 4 2  [2, ] 3 1 |
| 61 | > a+b  [ , 1] [ , 2]  [1, ] 5 5  [2, ] 5 5 |
|  | a <- c(1, 2, 3, 4, 5)  b <- c("간짜장", "간짬뽕", "탕수육", "양장피")  c <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 3, ncol = 2)  d <- head(iris) |
| 62 | > a  [1] 1 2 3 4 5  > b  [1] “간짜장" “간짬뽕" "탕수육" "양장피"  > c  [ ,1] [ ,2]  [1, ] 1 4  [2, ] 2 5  [3, ] 3 6  > d  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa |
|  | > is(a)  [1] "numeric" "vector"  > is(b)  [1] "character" "vector" “data.frameRowLabels”  [4] “SuperClassMethod”  > is(c)  [1] "matrix" "array" “structure" "vector"  > is(d)  [1] "data.frame" "list" "oldClass" "vector" |
| 63 | > list(a, b, c, d)  [[1]]  [1] 1 2 3 4 5  [[2]]  [1] “간짜장" "간짬뽕" "탕수육" "양장피"  [[3]]  [ ,1] [ ,2]  [1, ] 1 4  [2, ] 2 5  [3, ] 3 6  [[4]]  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa |
| 66 | > iris  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa |
| 67 | > install.packages(“xlsx”)  > library(xlsx) |
| 68 | > write.csv(df, "df\_csv.csv")  > write.table(df, "df\_txt.txt", sep = "\t")  > write.xlsx(df, "df\_xlsx.xlsx") |
| 70 | > df\_csv <- read.csv("df\_csv.csv", header = T)  > head(df\_csv)  X Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 1 5.1 3.5 1.4 0.2 setosa  2 2 4.9 3.0 1.4 0.2 setosa  3 3 4.7 3.2 1.3 0.2 setosa  4 4 4.6 3.1 1.5 0.2 setosa  5 5 5.0 3.6 1.4 0.2 setosa  6 6 5.4 3.9 1.7 0.4 setosa |
|  | > df\_txt <- read.table("df\_txt.txt", header = T, sep = "\t")  > head(df\_txt)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa |
| 71 | > df\_xlsx <- read.xlsx("df\_xlsx.xlsx", 1, header = T, stringsAsFactors = FALSE)  > head(df\_xlsx)  X Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 1 5.1 3.5 1.4 0.2 setosa  2 2 4.9 3.0 1.4 0.2 setosa  3 3 4.7 3.2 1.3 0.2 setosa  4 4 4.6 3.1 1.5 0.2 setosa  5 5 5.0 3.6 1.4 0.2 setosa  6 6 5.4 3.9 1.7 0.4 setosa |
| 72 | > df\_xlsx2 <- read.xlsx("df\_xlsx.xlsx", 1, header = F, stringsAsFactors = FALSE)  > head(df\_xlsx2)  X1 X2 X3 X4 X5 X6  X Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 1 5.1 3.5 1.4 0.2 setosa  2 2 4.9 3.0 1.4 0.2 setosa  3 3 4.7 3.2 1.3 0.2 setosa  4 4 4.6 3.1 1.5 0.2 setosa  5 5 5.0 3.6 1.4 0.2 setosa  6 6 5.4 3.9 1.7 0.4 setosa |
| 74 | > df <- data.frame(var1 = c(1,4,7), var2 = c(2,5,8), var3 = c(3, 6, 9))  > df  var1 var2 var3  1 1 2 3  2 4 5 6  3 7 8 9 |
| 77 | > str(iris)  'data.frame': 150 obs. of 5 variables:  $ Sepal.Length : num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  $ Petal.Length : num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ... |
| 78 | > head(iris)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa |
| 79 | > head(iris, 10)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa  7 4.6 3.4 1.4 0.3 setosa  8 5.0 3.4 1.5 0.2 setosa  9 4.4 2.9 1.4 0.2 setosa  10 4.9 3.1 1.5 0.1 setosa |
|  | > tail(iris)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  145 6.7 3.3 5.7 2.5 virginica  146 6.7 3.0 5.2 2.3 virginica  147 6.3 2.5 5.0 1.9 virginica  148 6.5 3.0 5.2 2.0 virginica  149 6.2 3.4 5.4 2.3 virginica  150 5.9 3.0 5.1 1.8 virginica |
| 80 | > tail(iris, 10)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  145 6.7 3.3 5.7 2.5 virginica  146 6.7 3.0 5.2 2.3 virginica  147 6.3 2.5 5.0 1.9 virginica  148 6.5 3.0 5.2 2.0 virginica  149 6.2 3.4 5.4 2.3 virginica  150 5.9 3.0 5.1 1.8 virginica  147 6.3 2.5 5.0 1.9 virginica  148 6.5 3.0 5.2 2.0 virginica  149 6.2 3.4 5.4 2.3 virginica  150 5.9 3.0 5.1 1.8 virginica |
| 82 | > iris\_1 <- iris[1:100, ]  > iris\_2 <- iris[101:150, ] |
| 83 | > str(iris\_1)  'data.frame': 100 obs. of 5 variables:  $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 …  > str(iris\_2)  'data.frame': 50 obs. of 5 variables:  $ Sepal.Length: num 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 ...  $ Sepal.Width : num 3.3 2.7 3 2.9 3 3 2.5 2.9 2.5 3.6 ...  $ Petal.Length: num 6 5.1 5.9 5.6 5.8 6.6 4.5 6.3 5.8 6.1 ...  $ Petal.Width : num 2.5 1.9 2.1 1.8 2.2 2.1 1.7 1.8 1.8 2.5 ...  $ Species : Factor w/ 3 levels "setosa","versicolor",..: 3 3 3 3 3 3 3 3 3 3 ... |
| 84 | > iris\_3 <- rbind(iris\_1, iris\_2)  > str(iris\_3)  'data.frame': 150 obs. of 5 variables:  $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ... |
| 85 | > iris\_1 <- iris[ ,1:3]  > iris\_2 <- iris[ ,4:5] |
|  | > head(iris\_1)  Sepal.Length Sepal.Width Petal.Length  1 5.1 3.5 1.4  2 4.9 3.0 1.4  3 4.7 3.2 1.3  4 4.6 3.1 1.5  5 5.0 3.6 1.4  6 5.4 3.9 1.7  > head(iris\_2)  Petal.Width Species  1 0.2 setosa  2 0.2 setosa  3 0.2 setosa  4 0.2 setosa  5 0.2 setosa  6 0.4 setosa |
| 86 | > iris\_3 <- cbind(iris\_1, iris\_2)  > head(iris\_3)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa |
| 88 | > df1 <- data.frame(과일 = c("사과","배","수박","바나나"), 판매량 = c(20, 5, 10, 5))  > df2 <- data.frame(과일 = c("사과", "바나나", "수박", "배"), 가격 = c(500, 700, 15000, 1200)) |
|  | > df3 <- merge(df1, df2, by= "과일")  > df3  과일 판매량 가격  1 바나나 5 700  2 배 5 1200  3 사과 20 500  4 수박 10 15000 |
| 89 | > df <- iris |
| 90 | > head(df)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa |
| 91 | > df[1:6, ]  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa |
|  | > df[130:140, ]  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  130 7.2 3.0 5.8 1.6 virginica  131 7.4 2.8 6.1 1.9 virginica  132 7.9 3.8 6.4 2.0 virginica  (중략)  138 6.4 3.1 5.5 1.8 virginica  139 6.0 3.0 4.8 1.8 virginica  140 6.9 3.1 5.4 2.1 virginica |
| 92 | > df[27, ]  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  27 5 3.4 1.6 0.4 setosa |
|  | > df[c(20, 1, 7), ]  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  20 5.1 3.8 1.5 0.3 setosa  1 5.1 3.5 1.4 0.2 setosa  7 4.6 3.4 1.4 0.3 setosa |
| 93 | > head(df[ , 1:4])  Sepal.Length Sepal.Width Petal.Length Petal.Width  1 5.1 3.5 1.4 0.2  2 4.9 3.0 1.4 0.2  3 4.7 3.2 1.3 0.2  4 4.6 3.1 1.5 0.2  5 5.0 3.6 1.4 0.2  6 5.4 3.9 1.7 0.4 |
|  | > head(df[ , 4:5])  Petal.Width Species  1 0.2 setosa  2 0.2 setosa  3 0.2 setosa  4 0.2 setosa  5 0.2 setosa  6 0.4 setosa |
| 94 | > head(df[ , c(1, 3, 5)])  Sepal.Length Petal.Length Species  1 5.1 1.4 setosa  2 4.9 1.4 setosa  3 4.7 1.3 setosa  4 4.6 1.5 setosa  5 5.0 1.4 setosa  6 5.4 1.7 setosa |
|  | > head(df[ , “Sepal.Length”])  [1] 5.1 4.9 4.7 4.6 5.0 5.4 |
| 95 | > data.frame(head(df[ , "Sepal.Length"]))  head.df....Sepal.Length...  1 5.1  2 4.9  3 4.7  4 4.6  5 5.0  6 5.4 |
|  | > df$Sepal.Length  [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1 5.7 5.1 5.4 5.1  [23] 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0  [45] 5.1 4.8 5.1 4.6 5.3 5.0 7.0 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7  [67] 5.6 5.8 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4 6.0 6.7 6.3  [89] 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2  [111] 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9  [133] 6.4 6.3 6.1 7.7 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9 |
| 96 | > df[3, 4]  [1] 0.2 |
|  | > df[3, 1:2]  Sepal.Length Sepal.Width  3 4.7 3.2 |
| 97 | > df[c(1,3,5), c(1,3,5)]  Sepal.Length Petal.Length Species  1 5.1 1.4 setosa  3 4.7 1.3 setosa  5 5.0 1.4 setosa |
| 98 | > summary(df)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100 setosa :50  1st Qu. :5.100 1st Qu. :2.800 1st Qu. :1.600 1st Qu. :0.300 versicolor:50  Median :5.800 Median :3.000 Median :4.350 Median :1.300 virginica :50  Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199  3rd Qu. :6.400 3rd Qu. :3.300 3rd Qu. :5.100 3rd Qu. :1.800  Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500 |
|  | > df[df$Sepal.Length >= 7.7, ]  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  118 7.7 3.8 6.7 2.2 virginica  119 7.7 2.6 6.9 2.3 virginica  123 7.7 2.8 6.7 2.0 virginica  132 7.9 3.8 6.4 2.0 virginica  136 7.7 3.0 6.1 2.3 virginica |
| 99 | > df[df$Sepal.Length >= 6 & df$Petal.Length >= 6, 1:4]  Sepal.Length Sepal.Width Petal.Length Petal.Width  101 6.3 3.3 6.0 2.5  106 7.6 3.0 6.6 2.1  108 7.3 2.9 6.3 1.8  110 7.2 3.6 6.1 2.5  118 7.7 3.8 6.7 2.2  119 7.7 2.6 6.9 2.3  123 7.7 2.8 6.7 2.0  126 7.2 3.2 6.0 1.8  131 7.4 2.8 6.1 1.9  132 7.9 3.8 6.4 2.0  136 7.7 3.0 6.1 2.3 |
| 100 | > head(df[df$Sepal.Length >= 6 | df$Petal.Length >= 6, 1:4])  Sepal.Length Sepal.Width Petal.Length Petal.Width  51 7.0 3.2 4.7 1.4  52 6.4 3.2 4.5 1.5  53 6.9 3.1 4.9 1.5  55 6.5 2.8 4.6 1.5  57 6.3 3.3 4.7 1.6  59 6.6 2.9 4.6 1.3 |
| 101 | > df <- data.frame(v1 = c(1,4,7), v2 = c(2,5,8), v3 = c(3, 6, 9))  > df  v1 v2 v3  1 1 2 3  2 4 5 6  3 7 8 9 |
| 102 | > data.frame(df, v4 = c(10,11,12))  v1 v2 v3 v4  1 1 2 3 10  2 4 5 6 11  3 7 8 9 12 |
|  | > df <- data.frame(sales = c(1000, 1100, 1200, 1500),  + cost = c(1200, 1000, 1000, 1100),  + row.names = c(2013, 2014, 2015, 2016)) |
|  | > df  sales cost  2013 1000 1200  2014 1100 1000  2015 1200 1000  2016 1500 1100 |
| 103 | > df2 <- data.frame(df, profit = df$sales - df$cost)  > df2  sales cost profit  2013 1000 1200 -200  2014 1100 1000 100  2015 1200 1000 200  2016 1500 1100 400 |
| 104 | > df2$profitRate <- df2$profit / df2$sales  > df2  sales cost profit profitRate  2013 1000 1200 -200 -0.20000000  2014 1100 1000 100 0.09090909  2015 1200 1000 200 0.16666667  2016 1500 1100 400 0.26666667 |
|  | > df2[1, 4] <- '-' |
|  | > df2  sales cost profit profitRate  2013 1000 1200 -200 -  2014 1100 1000 100 0.09090909  2015 1200 1000 200 0.16666667  2016 1500 1100 400 0.26666667 |
| 106 | > df2.a <- df2[-1, ]  > df2.a  sales cost profit profitRate  2014 1100 1000 100 0.09090909  2015 1200 1000 200 0.16666667  2016 1500 1100 400 0.26666667 |
|  | > df2.b <- df2[2:4, ]  > df2.b  sales cost profit profitRate  2014 1100 1000 100 0.09090909  2015 1200 1000 200 0.16666667  2016 1500 1100 400 0.26666667 |
| 107 | > df3 <- df2.b[, c(1, 2, 4)]  > df3  sales cost profitRate  2014 1100 1000 0.09090909  2015 1200 1000 0.16666667  2016 1500 1100 0.26666667 |
| 110 | > df <- data.frame(a = c(1,2,3), b = c(4,5,6), c = c(7,8,9))  > df  a b c  1 1 4 7  2 2 5 8  3 3 6 9 |
|  | # 행 기준 apply  > apply(df, 1, sum)  [1] 12 15 18  # 열 기준 apply  > apply(df, 2, sum)  a b c  6 15 24 |
| 111 | 첫번째 코드 이미지 삭제(110쪽 마지막 코드와 동일) |
|  | > df\_mobile <- data.frame("Y2013" = c(20, 5, 40, 5),  + "Y2014" = c(30, 15, 50, 10),  + "Y2015" = c(50, 30, 60, 30),  + "Y2016" = c(80, 50, 60, 40),  + row.names = c("SNS", "video", "message", "shopping")) |
| 112 | > df\_mobile  Y2013 Y2014 Y2015 Y2016  SNS 20 30 50 80  video 5 15 30 50  message 40 50 60 60  shopping 5 10 30 40 |
|  | > apply(df\_mobile, 1, sum)  SNS video message shopping  180 100 210 85  > apply(df\_mobile, 2, sum)  Y2013 Y2014 Y2015 Y2016  70 105 170 230 |
| 113 | > lapply(c(1:3), mean)  [[1]]  [1] 1  [[2]]  [1] 2  [[3]]  [1] 3 |
| 114 | > list\_df <- list(a = c(1:2), b = c(3:4)) |
|  | > lapply(list\_df, mean)  $a  [1] 1.5  $b  [1] 3.5 |
|  | > lapply(df\_mobile, sum)  $Y2013  [1] 70  $Y2014  [1] 105  $Y2015  [1] 170  $Y2016  [1] 230 |
| 115 | > sapply(c(1:3), mean)  [1] 1 2 3  > sapply(list\_df, mean)  a b  1.5 3.5  > sapply(df\_mobile, sum)  Y2013 Y2014 Y2015 Y2016  70 105 170 230 |
| 116 | > sapply(df\_mobile[ ,1:4], function(x) {x\*2})  Y2013 Y2014 Y2015 Y2016  [1,] 40 60 100 160  [2,] 10 30 60 100  [3,] 80 100 120 120  [4,] 10 20 60 80 |
| 117 | > df\_iris <- iris |
|  | > tapply(df\_iris$Sepal.Width, df\_iris$Species, sd)  setosa versicolor virginica  0.3790644 0.3137983 0.3224966 |
| 118 | > mapply(mean, df\_iris[, 1:4])  Sepal.Length Sepal.Width Petal.Length Petal.Width  5.843333 3.057333 3.758000 1.199333 |
| 120 | > subset(iris, subset = (Sepal.Length > 7.5))  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  106 7.6 3.0 6.6 2.1 virginica  118 7.7 3.8 6.7 2.2 virginica  119 7.7 2.6 6.9 2.3 virginica  123 7.7 2.8 6.7 2.0 virginica  132 7.9 3.8 6.4 2.0 virginica  136 7.7 3.0 6.1 2.3 virginica |
| 121 | > iris[iris$Sepal.Length > 7.5, ]  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  106 7.6 3.0 6.6 2.1 virginica  118 7.7 3.8 6.7 2.2 virginica  119 7.7 2.6 6.9 2.3 virginica  123 7.7 2.8 6.7 2.0 virginica  132 7.9 3.8 6.4 2.0 virginica  136 7.7 3.0 6.1 2.3 virginica |
|  | > subset(iris, Sepal.Length > 7.5)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  106 7.6 3.0 6.6 2.1 virginica  118 7.7 3.8 6.7 2.2 virginica  119 7.7 2.6 6.9 2.3 virginica  123 7.7 2.8 6.7 2.0 virginica  132 7.9 3.8 6.4 2.0 virginica  136 7.7 3.0 6.1 2.3 virginica |
|  | > subset(iris, subset = (Sepal.Length > 7.5 & Petal.Length >= 6.5))  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  106 7.6 3.0 6.6 2.1 virginica  118 7.7 3.8 6.7 2.2 virginica  119 7.7 2.6 6.9 2.3 virginica  123 7.7 2.8 6.7 2.0 virginica |
| 122 | > iris[iris$Sepal.Length > 7.5 & iris$Petal.Length >= 6.5, ]  106 7.6 3.0 6.6 2.1 virginica  118 7.7 3.8 6.7 2.2 virginica  119 7.7 2.6 6.9 2.3 virginica  123 7.7 2.8 6.7 2.0 virginica |
| 123 | > subset(iris, select = c(1:3))  Sepal.Length Sepal.Width Petal.Length  1 5.1 3.5 1.4  2 4.9 3.0 1.4  3 4.7 3.2 1.3  4 4.6 3.1 1.5  5 5.0 3.6 1.4  6 5.4 3.9 1.7  (중략) |
|  | > subset(iris, subset = (Petal.Width > 2.3), select = c(1,3,5))  Sepal.Length Petal.Length Species  101 6.3 6.0 virginica  110 7.2 6.1 virginica  115 5.8 5.1 virginica  137 6.3 5.6 virginica  141 6.7 5.6 virginica  145 6.7 5.7 virginica |
| 127 | > install.packages(“ggplot2")  > library(ggplot2) |
|  | > install.packages(“dplyr”)  > library(dplyr) |
| 128 | > df <- mpg  > df  # A tibble: 234 x 11  manufacturer model displ year cyl trans drv cty hwy fl class  <chr> <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>  1 audi a4 1.8 1999 4 auto(l5) f 18 29 p compact  2 audi a4 1.8 1999 4 manual(m5) f 21 29 p compact  3 audi a4 2.0 2008 4 manual(m6) f 20 31 p compact  4 audi a4 2.0 2008 4 auto(av) f 21 30 p compact  5 audi a4 2.8 1999 6 auto(l5) f 16 26 p compact  6 audi a4 2.8 1999 6 manual(m5) f 18 26 p compact  7 audi a4 3.1 2008 6 auto(av) f 18 27 p compact  8 audi a4 quattro 1.8 1999 4 manual(m5) 4 18 26 p compact  9 audi a4 quattro 1.8 1999 4 auto(l5) 4 16 25 p compact  10 audi a4 quattro 2.0 2008 4 manual(m6) 4 20 28 p compact  # ... with 224 more rows |
| 129 | > select(df, model)  # A tibble: 234 x 1  model  <chr>  1 a4  2 a4  3 a4  4 a4  5 a4  6 a4  7 a4  8 a4 quattro  9 a4 quattro  10 a4 quattro  # ... with 224 more rows |
| 130 | > select(df, model, year)  # A tibble: 234 x 2  model year  <chr> <int>  1 a4 1999  2 a4 1999  3 a4 2008  4 a4 2008  5 a4 1999  6 a4 1999  7 a4 2008  8 a4 quattro 1999  9 a4 quattro 1999  10 a4 quattro 2008  # ... with 224 more rows |
| 131 | > filter(df, manufacturer == 'audi')  # A tibble: 18 x 11  manufacturer model displ year cyl trans drv cty hwy fl class  <chr> <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>  1 audi a4 1.8 1999 4 auto(l5) f 18 29 p compact  2 audi a4 1.8 1999 4 manual(m5) f 21 29 p compact  3 audi a4 2.0 2008 4 manual(m6) f 20 31 p compact  4 audi a4 2.0 2008 4 auto(av) f 21 30 p compact  5 audi a4 2.8 1999 6 auto(l5) f 16 26 p compact  6 audi a4 2.8 1999 6 manual(m5) f 18 26 p compact  7 audi a4 3.1 2008 6 auto(av) f 18 27 p compact  8 audi a4 quattro 1.8 1999 4 manual(m5) 4 18 26 p compact  9 audi a4 quattro 1.8 1999 4 auto(l5) 4 16 25 p compact  10 audi a4 quattro 2.0 2008 4 manual(m6) 4 20 28 p compact  (중략) |
| 132 | > filter(df, manufacturer == 'audi' & drv == ‘f')  manufacturer model displ year cyl trans drv cty hwy fl class  <chr> <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>  1 audi a4 1.8 1999 4 auto(l5) f 18 29 p compact  2 audi a4 1.8 1999 4 manual(m5) f 21 29 p compact  3 audi a4 2.0 2008 4 manual(m6) f 20 31 p compact  4 audi a4 2.0 2008 4 auto(av) f 21 30 p compact  5 audi a4 2.8 1999 6 auto(l5) f 16 26 p compact  6 audi a4 2.8 1999 6 manual(m5) f 18 26 p compact  7 audi a4 3.1 2008 6 auto(av) f 18 27 p compact |
| 133 | > mutate(df, rate\_eff = hwy/cty)  # A tibble: 234 x 12  manufacturer model displ year cyl trans drv cty hwy fl class rate\_eff  <chr> <chr> <dbl> <int> <int> <chr> <chr><int> <int> <chr> <chr> <dbl>  1 audi a4 1.8 1999 4 auto(l5) f 18 29 p compact 1.611111  2 audi a4 1.8 1999 4 manual(m5) f 21 29 p compact 1.380952  3 audi a4 2.0 2008 4 manual(m6) f 20 31 p compact 1.550000  4 audi a4 2.0 2008 4 auto(av) f 21 30 p compact 1.428571  5 audi a4 2.8 1999 6 auto(l5) f 16 26 p compact 1.625000  6 audi a4 2.8 1999 6 manual(m5) f 18 26 p compact 1.444444  7 audi a4 3.1 2008 6 auto(av) f 18 27 p compact 1.500000  8 audi a4 quattro 1.8 1999 4 manual(m5) 4 18 26 p compact 1.444444  9 audi a4 quattro 1.8 1999 4 auto(l5) 4 16 25 p compact 1.562500  10 audi a4 quattro 2.0 2008 4 manual(m6) 4 20 28 p compact 1.400000 |
| 134 | > arrange(mutate(df, rate\_eff = hwy/cty), desc(rate\_eff))  # A tibble: 234 x 12  manufacturer model displ year cyl trans drv cty hwy fl class rate\_eff  <chr> <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr> <dbl>  1 audi a4 quattro 2.8 1999 6 auto(l5) 4 15 25 p compact 1.666667  2 audi a4 quattro 3.1 2008 6 manual(m6) 4 15 25 p compact 1.666667  3 chevrolet corvette 6.2 2008 8 auto(s6) r 15 25 p 2seater 1.666667  4 audi a4 2.8 1999 6 auto(l5) f 16 26 p compact 1.625000  5 chevrolet corvette 5.7 1999 8 manual(m6) r 16 26 p 2seater 1.625000  6 chevrolet corvette 6.2 2008 8 manual(m6) r 16 26 p 2seater 1.625000  7 pontiac grand prix 3.8 1999 6 auto(l4) f 16 26 p midsize 1.625000  8 volkswagen passat 2.8 1999 6 auto(l5) f 16 26 p midsize 1.625000  9 audi a4 1.8 1999 4 auto(l5) f 18 29 p compact 1.611111  10 chevrolet malibu 3.5 2008 6 auto(l4) f 18 29 r midsize 1.611111  # ... with 224 more rows |
| 136 | > summarise(df, sum\_cty = sum(cty))  # A tibble: 1 x 1  sum\_cty  <int>  1 3945 |
|  | > summarise(df, sum\_cty = sum(cty), sum\_hwy = sum(hwy),  + mean\_cty = mean(cty), mean\_hwy = mean(hwy))  # A tibble: 1 x 4  sum\_cty sum\_hwy mean\_cty mean\_hwy  <int> <int> <dbl> <dbl>  1 3945 5485 16.85897 23.44017 |
| 138 | > arrange(summarise(group\_by(df, manufacturer), mean\_cty = mean(cty),  + mean\_hwy = mean(hwy)), desc(mean\_hwy))  # A tibble: 15 x 3  manufacturer mean\_cty mean\_hwy  <chr> <dbl> <dbl>  1 honda 24.44444 32.55556  2 volkswagen 20.92593 29.22222  3 hyundai 18.64286 26.85714  4 audi 17.61111 26.44444  5 pontiac 17.00000 26.40000  6 subaru 19.28571 25.57143  7 toyota 18.52941 24.91176  8 nissan 18.07692 24.61538  9 chevrolet 15.00000 21.89474  10 ford 14.00000 19.36000  11 mercury 13.25000 18.00000  12 dodge 13.13514 17.94595  13 jeep 13.50000 17.62500  14 lincoln 11.33333 17.00000  15 land rover 11.50000 16.50000 |
| 141 | > df %>% select(model, cty)  # A tibble: 234 x 2  model cty  <chr> <int>  1 a4 18  2 a4 21  3 a4 20  4 a4 21  5 a4 16  6 a4 18  7 a4 18  8 a4 quattro 18  9 a4 quattro 16  10 a4 quattro 20  # ... with 224 more rows |
| 142 | > df %>% select(model, cty) %>% filter(model == 'a4')  # A tibble: 7 x 2  model cty  <chr> <int>  1 a4 18  2 a4 21  3 a4 20  4 a4 21  5 a4 16  6 a4 18  7 a4 18 |
|  | > df %>% select(model, cty) %>% filter(model == 'a4') %>% summarise(mean(cty))  # A tibble: 1 x 1  `mean(cty)`  <dbl>  1 18.85714 |
| 145 | if(조건)  '조건이 참일때 출력'  else  '조건이 거짓일 때 출력' |
|  | ifelse(조건, '조건이 참일 때', ‘거짓일 때' ) |
| 146 | 첫번 째 코드 이미지 삭제 |
|  | > if(1>2) {  + print("1은 2보다 크다")  + } else {  + print("넌 너무 비논리적이야!")  + }  [1] "넌 너무 비논리적이야!" |
| 147 | > if(rnorm(1) > 0) {  + print("0보다 크다")  + } else {  + print("0보다 작다")  + } |
|  | 두번째 코드 이미지 삭제 |
|  | > ifelse(rnorm(50) > 0, "0보다 크다", "0보다 작다")  [1] "0보다 작다" "0보다 작다" "0보다 크다" "0보다 크다" "0보다 작다" "0보다 작다" "0보다 크다"  [8] "0보다 작다" "0보다 크다" "0보다 크다" "0보다 작다" "0보다 크다" "0보다 크다" "0보다 크다"  [15] "0보다 크다" "0보다 크다" "0보다 크다" "0보다 크다" "0보다 크다" "0보다 작다" "0보다 크다"  [22] "0보다 작다" "0보다 작다" "0보다 작다" "0보다 작다" "0보다 작다" "0보다 작다" "0보다 크다"  [29] "0보다 작다" "0보다 크다" "0보다 작다" "0보다 크다" "0보다 작다" "0보다 작다" "0보다 작다"  [36] "0보다 크다" "0보다 크다" "0보다 크다" "0보다 작다" "0보다 크다" "0보다 크다" "0보다 작다"  [43] "0보다 작다" "0보다 작다" "0보다 작다" "0보다 작다" "0보다 크다" "0보다 작다" "0보다 작다"  [50] "0보다 크다" |
| 148 | for(기준 in 반복횟수) {  ‘반복할 코드'  } |
|  | for(i in 1:10) {  print(i)  } |
| 149 | > for(i in 1:10) {  + print(i)  + }  [1] 1  [1] 2  [1] 3  [1] 4  [1] 5  [1] 6  [1] 7  [1] 8  [1] 9  [1] 10 |
|  | > df <- as.data.frame(matrix(ncol=9, nrow=9)) |
|  | for (i in 1:9) {  for (j in 1:9) {  df[i, j] <- i\*j  }  } |
| 150 | > for (i in 1:9) {  + for (j in 1:9) {  + df[i,j] <- i\*j  + }   * }   > df  V1 V2 V3 V4 V5 V6 V7 V8 V9  1 1 2 3 4 5 6 7 8 9  2 2 4 6 8 10 12 14 16 18  3 3 6 9 12 15 18 21 24 27  4 4 8 12 16 20 24 28 32 36  5 5 10 15 20 25 30 35 40 45  6 6 12 18 24 30 36 42 48 54  7 7 14 21 28 35 42 49 56 63  8 8 16 24 32 40 48 56 64 72  9 9 18 27 36 45 54 63 72 81 |
| 188 | > table(iris$Species)  setosa versicolor virginica  50 50 50 |
| 195 | > df\_pie <- data.frame(a = rep(1:5))  > df\_pie  a  1 1  2 2  3 3  4 4  5 5 |
| 206 | > install.packages(“vcd”)  > library(vcd) |
|  | > df <- JointSports  > str(df)  'data.frame': 40 obs. of 5 variables:  $ Freq : num 31 103 23 61 41 77 31 52 51 67 ...  $ opinion: Factor w/ 5 levels "very good","good",..: 1 1 1 1 1 1 1 1 2 2 ...  $ year : Factor w/ 2 levels "1983","1985": 1 1 1 1 2 2 2 2 1 1 ...  $ grade : Factor w/ 2 levels "1st","3rd": 1 1 2 2 1 1 2 2 1 1 ...  $ gender : Factor w/ 2 levels "Boy","Girl": 1 2 1 2 1 2 1 2 1 2 …  > head(df)  Freq opinion year grade gender  1 31 very good 1983 1st Boy  2 103 very good 1983 1st Girl  3 23 very good 1983 3rd Boy  4 61 very good 1983 3rd Girl  5 41 very good 1985 1st Boy  6 77 very good 1985 1st Girl |
| 214 | > install.packages(“ggplot2”)  > library(ggplot2) |
| 217 | > install.packages(“gridExtra”)  > library(gridExtra) |
| 220 | > install.packages(“ggthemes”)  > library(ggthemes) |
| 230 | > str(df)  'data.frame': 533 obs. of 19 variables:  $ 충전소명 : chr "목포관광안내소" "목포시청" "제천시 차량지원센터" "충청북도 제천시 봉양읍사무소" ...  $ 충전소위치상세 : chr "전라남도 목포시 영산로 843 목포관광안내소 옆(무안에서 목포로진입해서 오른쪽 목포관광안내소옆)" "전라남도 목포시청 의회 뒷편주차장" "차고" "외부 주차장" ...  $ 설치시도명 : chr "전라남도" "전라남도" "충청북도" "충청북도" ...  $ 휴점일 : chr "연중무휴" "연중무휴" "연중무휴" "연중무휴" ...  $ 이용가능시작시각: chr "00:00" "00:00" "00:00" "00:00" ...  $ 이용가능종료시각: chr "00:00" "00:00" "24:00" "24:00" ...  $ 완속충전가능여부: chr "N" "Y" "Y" "Y" ...  $ 급속충전가능여부: chr "Y" "N" "N" "N" ...  $ 급속충전타입구분: chr "" "" "" "" ...  $ 완속충전기대수 : chr "0" "1" "1" "1" ...  $ 급속충전기대수 : chr "1" "0" "" "" ...  $ 주차료부과여부 : chr "N" "N" "N" "N" ...  $ 소재지도로명주소: chr "전라남도 목포시 영산로 843" "전라남도 목포시 양을로 203" "충청북도 제천시 내토로 295(천남동)" "충청북도 제천시 봉양읍 주포로 83" ...  $ 소재지지번주소 : chr "전라남도 목포시 대양동 276" "전라남도 목포시 용당동1188-2" "충청북도 제천시 천남동 12-2" "충청북도 제천시 봉양읍 주포리 164-2" ...  $ 관리업체명 : chr "한국자동차환경협회" "한국자동차환경협회" "충청북도 제천시청" "" ...  $ 관리업체전화번호: chr "1661-9408" "1661-9409" "043-641-5723" "043-641-4023" ...  $ 위도 : chr "34.838268" "34.811622" "37.1345172" "37.1366945" ...  $ 경도 : chr "126.422267" "126.391975" "128.2000914" "128.1291204" ...  $ 데이터기준일자 : chr "2016-10-20" "2016-10-20" "2015-09-01" "2015-09-01" ... |
| 233 | > head(mut\_df\_add)  address lon lat  1 전라남도 목포시 대양동 276 NA NA  2 전라남도 목포시 용당동1188-2 1 26.3920 34.81161  3 충청북도 제천시 천남동 12-2 NA NA  4 충청북도 제천시 봉양읍 주포리 164-2 128.1215 37.13266  5 충청북도 제천시 하소동 323-7 NA NA  6 충청북도 제천시 송학면 시곡리 164-2 128.2930 37.18174 |
| 240 | > df <- data.frame(from = c("Mike", "Mike", "Mike", "Mike", "Mike",  + "Harvey", "Harvey", "Harvey", "Harvey",  + "Jessica", "Louis", "Louis",  + "Donna", "Rachel", "traver",  + "Hardman", "Hardman", "Hardman"),  + to = c("Harvey", "Jessica", "Louis", "Donna", "Rachel",  + "Jessica", "Louis", "Donna", "Rachel",  + "Louis", "Donna", "Christine",  + "Rachel", "Jessica", "Mike",  + "Jessica", "Harvey", "Louis")) |
| 241 | > str(df)  'data.frame': 18 obs. of 2 variables:  $ from: Factor w/ 8 levels "Donna","Hardman",..: 6 6 6 6 6 3 3 3 3 4 ...  $ to : Factor w/ 7 levels "Christine","Donna",..: 3 4 5 2 7 4 5 2 7 5 …  > head(df)  from to  1 Mike Harvey  2 Mike Jessica  3 Mike Louis  4 Mike Donna  5 Mike Rachel  6 Harvey Jessica |
| 242 | > df <- graph.data.frame(df, directed = FALSE)  > df  IGRAPH 8391fbf UN-- 9 18 --  + attr: name (v/c)  + edges from 8391fbf (vertex names):  [1] Mike --Harvey Mike --Jessica Mike --Louis Mike --Donna  [5] Mike --Rachel Harvey --Jessica Harvey --Louis Harvey --Donna  [9] Harvey --Rachel Jessica --Louis Louis --Donna Louis --Christine  [13] Donna --Rachel Jessica --Rachel Mike --traver Jessica --Hardman  [17] Harvey --Hardman Louis --Hardman |
|  | > head(df)  6 x 9 sparse Matrix of class "dgCMatrix"  Mike Harvey Jessica Louis Donna Rachel traver Hardman Christine  Mike . 1 1 1 1 1 1 . .  Harvey 1 . 1 1 1 1 . 1 .  Jessica 1 1 . 1 . 1 . 1 .  Louis 1 1 1 . 1 . . 1 1  Donna 1 1 . 1 . 1 . . .  Rachel 1 1 1 . 1 . . . . |
| 246 | > degree(df)  Mike Harvey Jessica Louis Donna Rachel traver  6 6 5 6 4 4 1  Hardman Christine  3 1 |
| 265 | > str(diamonds)  Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 53940 obs. of 10 variables:  $ carat : num 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...  $ cut : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...  $ color : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...  $ clarity : Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...  $ depth : num 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...  $ table : num 55 61 65 58 58 57 57 55 61 61 ...  $ price : int 326 326 327 334 335 336 336 337 337 338 ...  $ x : num 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...  $ y : num 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...  $ z : num 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ... |
| 266 | > head(diamonds)  # A tibble: 6 x 10  carat cut color clarity depth table price x y z  <dbl> <ord> <ord> <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43  2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31  3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31  4 0.29 Premium I VS2 62.4 58 334 4.20 4.23 2.63  5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75  6 0.24 Very Good J VVS2 62.8 57 336 3.94 3.96 2.48 |
| 267 | > lmDiamond <- lm(price ~ carat , data = diamonds)  > lmDiamond  Call:  lm(formula = price ~ carat, data = diamonds)  Coefficients:  (Intercept) carat  -2256 7756 |
| 268 | > summary(lmDiamond)  Call:  lm(formula = price ~ carat, data = diamonds)  Residuals:  Min 1Q Median 3Q Max  -18585.3 -804.8 -18.9 537.4 12731.7  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) -2256.36 13.06 -172.8 <2e-16 \*\*\*  carat 7756.43 14.07 551.4 <2e-16 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 1549 on 53938 degrees of freedom  Multiple R-squared: 0.8493, Adjusted R-squared: 0.8493  F-statistic: 3.041e+05 on 1 and 53938 DF, p-value: < 2.2e-16 |
| 275 | > str(mtcars)  'data.frame': 32 obs. of 11 variables:  $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...  $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...  $ disp : num 160 160 108 258 360 ...  $ hp : num 110 110 93 110 175 105 245 62 95 123 ...  $ drat : num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...  $ wt : num 2.62 2.88 2.32 3.21 3.44 ...  $ qsec : num 16.5 17 18.6 19.4 17 ...  $ vs : num 0 0 1 1 0 1 0 1 1 1 ...  $ am : num 1 1 1 0 0 0 0 0 0 0 ...  $ gear : num 4 4 4 3 3 3 3 4 4 4 ...  $ carb : num 4 4 1 1 2 1 4 2 2 4 …  > head(mtcars)  mpg cyl disp hp drat wt qsec vs am gear carb  Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1 |
| 276 | > summary(mtcars)  mpg cyl disp hp  Min. :10.40 Min. :4.000 Min. : 71.1 Min. : 52.0  1st Qu. :15.43 1st Qu. :4.000 1st Qu. :120.8 1st Qu. : 96.5  Median :19.20 Median :6.000 Median :196.3 Median :123.0  Mean :20.09 Mean :6.188 Mean :230.7 Mean :146.7  3rd Qu. :22.80 3rd Qu.:8.000 3rd Qu. :326.0 3rd Qu. :180.0  Max. :33.90 Max. :8.000 Max. :472.0 Max. :335.0    drat wt qsec vs  Min. :2.760 Min. :1.513 Min. :14.50 Min. :0.0000  1st Qu. :3.080 1st Qu. :2.581 1st Qu. :16.89 1st Qu. :0.0000  Median :3.695 Median :3.325 Median :17.71 Median :0.0000  Mean :3.597 Mean :3.217 Mean :17.85 Mean :0.4375  3rd Qu. :3.920 3rd Qu. :3.610 3rd Qu.:18.90 3rd Qu. :1.0000  Max. :4.930 Max. :5.424 Max. :22.90 Max. :1.0000    am gear carb  Min. :0.0000 Min. :3.000 Min. :1.000  1st Qu. :0.0000 1st Qu. :3.000 1st Qu. :2.000  Median :0.0000 Median :4.000 Median :2.000  Mean :0.4062 Mean :3.688 Mean :2.812  3rd Qu. :1.0000 3rd Qu. :4.000 3rd Qu. :4.000  Max. :1.0000 Max. :5.000 Max. :8.000 |
| 278 | > mlmData <- lm(mpg ~ cyl + disp + hp + drat + wt +qsec, data = mtcars)  > mlmData  Call:  lm(formula = mpg ~ cyl + disp + hp + drat + wt + qsec, data = mtcars)  Coefficients:  (Intercept) cyl disp hp drat wt qsec  26.30736 -0.81856 0.01320 -0.01793 1.32041 -4.19083 0.40146 |
| 279 | > summary(mlmData)  Call:  lm(formula = mpg ~ cyl + disp + hp + drat + wt + qsec, data = mtcars)  Residuals:  Min 1Q Median 3Q Max  -3.9682 -1.5795 -0.4353 1.1662 5.5272  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 26.30736 14.62994 1.798 0.08424 .  cyl -0.81856 0.81156 -1.009 0.32282  disp 0.01320 0.01204 1.097 0.28307  hp -0.01793 0.01551 -1.156 0.25846  drat 1.32041 1.47948 0.892 0.38065  wt -4.19083 1.25791 -3.332 0.00269 \*\*  qsec 0.40146 0.51658 0.777 0.44436  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 2.557 on 25 degrees of freedom  Multiple R-squared: 0.8548, Adjusted R-squared: 0.82  F-statistic: 24.53 on 6 and 25 DF, p-value: 2.45e-09 |
| 286 | > str(ex\_diamonds)  Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 53940 obs. of 7 variables:  $ carat : num 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...  $ depth : num 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...  $ table : num 55 61 65 58 58 57 57 55 61 61 ...  $ price : int 326 326 327 334 335 336 336 337 337 338 ...  $ x : num 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...  $ y : num 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...  $ z : num 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ... |
|  | > lm\_ex\_dia <- lm(price ~., data = ex\_diamonds)  > lm\_ex\_dia  Call:  lm(formula = price ~ ., data = ex\_diamonds)  Coefficients:  (Intercept) carat depth table x y z  20849.32 10686.31 -203.15 -102.45 -1315.67 66.32 41.63 |
| 287 | > step\_lm\_ex\_dia <- step(lm\_ex\_dia, direction = "backward")  Start: AIC=788737.9  price ~ carat + depth + table + x + y + z  Df Sum of Sq RSS AIC  - z 1 1.9783e+06 1.2086e+11 788737  <none> 1.2086e+11 788738  - y 1 1.5131e+07 1.2087e+11 788743  - x 1 2.0910e+09 1.2295e+11 789661  - table 1 2.4724e+09 1.2333e+11 789828  - depth 1 3.0529e+09 1.2391e+11 790082  - carat 1 6.4066e+10 1.8492e+11 811678  Step: AIC=788736.8  price ~ carat + depth + table + x + y  Df Sum of Sq RSS AIC  <none> 1.2086e+11 788737  - y 1 1.6964e+07 1.2088e+11 788742  - table 1 2.4751e+09 1.2333e+11 789828  - x 1 2.8831e+09 1.2374e+11 790006  - depth 1 3.8304e+09 1.2469e+11 790418  - carat 1 6.4074e+10 1.8493e+11 811679 |
| 288 | > summary(step\_lm\_ex\_dia)  Call:  lm(formula = price ~ carat + depth + table + x + y, data = ex\_diamonds)  Residuals:  Min 1Q Median 3Q Max  -23872.1 -614.8 -50.5 347.5 12759.4  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 20702.947 419.575 49.343 < 2e-16 \*\*\*  carat 10686.707 63.199 169.095 < 2e-16 \*\*\*  depth -200.718 4.855 -41.344 < 2e-16 \*\*\*  table -102.490 3.084 -33.234 < 2e-16 \*\*\*  x -1293.542 36.063 -35.869 < 2e-16 \*\*\*  y 69.575 25.287 2.751 0.00594 \*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 1497 on 53934 degrees of freedom  Multiple R-squared: 0.8592, Adjusted R-squared: 0.8592  F-statistic: 6.583e+04 on 5 and 53934 DF, p-value: < 2.2e-16 |
| 290 | > lm\_mtcars <- lm(mpg ~., data = mtcars)  > lm\_mtcars  Call:  lm(formula = mpg ~ ., data = mtcars)  Coefficients:  (Intercept) cyl disp hp drat wt qsec  12.30337 -0.11144 0.01334 -0.02148 0.78711 -3.71530 0.82104  vs am gear carb  0.31776 2.52023 0.65541 -0.19942 |
|  | > step\_lm\_mtcars <- step(lm\_mtcars, direction = "backward")  Start: AIC=70.9  mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb    Df Sum of Sq RSS AIC  - cyl 1 0.0799 147.57 68.915  - vs 1 0.1601 147.66 68.932  - carb 1 0.4067 147.90 68.986  - gear 1 1.3531 148.85 69.190  - drat 1 1.6270 149.12 69.249  - disp 1 3.9167 151.41 69.736  - hp 1 6.8399 154.33 70.348  - qsec 1 8.8641 156.36 70.765  <none> 147.49 70.898  - am 1 10.5467 158.04 71.108   * wt 1 27.0144 174.51 74.280   (중략)  Step: AIC=61.31  mpg ~ wt + qsec + am  Df Sum of Sq RSS AIC  <none> 169.29 61.307  - am 1 26.178 195.46 63.908  - qsec 1 109.034 278.32 75.217  - wt 1 183.347 352.63 82.790 |
| 291 | > summary(step\_lm\_mtcars)  Call:  lm(formula = mpg ~ wt + qsec + am, data = mtcars)  Residuals:  Min 1Q Median 3Q Max  -3.4811 -1.5555 -0.7257 1.4110 4.6610  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 9.6178 6.9596 1.382 0.177915  wt -3.9165 0.7112 -5.507 6.95e-06 \*\*\*  qsec 1.2259 0.2887 4.247 0.000216 \*\*\*  am 2.9358 1.4109 2.081 0.046716 \*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 2.459 on 28 degrees of freedom  Multiple R-squared: 0.8497, Adjusted R-squared: 0.8336  F-statistic: 52.75 on 3 and 28 DF, p-value: 1.21e-11 |
| 297 | > df <- iris  > train\_df <- df[c(1:35, 51:85, 101:135), ]  > test\_df <- df[c(36:50, 86:100, 136:150), ] |
| 298 | > str(train\_df)  'data.frame': 105 obs. of 5 variables:  $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  $ Petal.Length : num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 …  > str(test\_df)  'data.frame': 45 obs. of 5 variables:  $ Sepal.Length : num 5 5.5 4.9 4.4 5.1 5 4.5 4.4 5 5.1 …  $ Sepal.Width : num 3.2 3.5 3.6 3 3.4 3.5 2.3 3.2 3.5 3.8 ...  $ Petal.Length : num 1.2 1.3 1.4 1.3 1.5 1.3 1.3 1.3 1.6 1.9 ...  $ Petal.Width : num 0.2 0.2 0.1 0.2 0.2 0.3 0.3 0.2 0.6 0.4 ...  $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ... |
| 299 | > tree\_df  node), split, n, deviance, yval, (yprob)  \* denotes terminal node  1) root 110 205.60 setosa ( 0.45455 0.45455 0.09091 )  2) Petal.Length < 2.45 50 0.00 setosa ( 1.00000 0.00000 0.00000 ) \*  3) Petal.Length > 2.45 60 54.07 versicolor ( 0.00000 0.83333 0.16667 )  6) Petal.Width < 1.65 48 0.00 versicolor ( 0.00000 1.00000 0.00000 ) \*  7) Petal.Width > 1.65 12 10.81 virginica ( 0.00000 0.16667 0.83333 )  14) Petal.Length < 5.7 5 6.73 virginica ( 0.00000 0.40000 0.60000 ) \*  15) Petal.Length > 5.7 7 0.00 virginica ( 0.00000 0.00000 1.00000 ) \* |
| 300 | > summary(tree\_df)  Classification tree:  tree(formula = Species ~ ., data = train\_df)  Variables actually used in tree construction:  [1] "Petal.Length" "Petal.Width"  Number of terminal nodes: 4  Residual mean deviance: 0.06349 = 6.73 / 106  Misclassification error rate: 0.01818 = 2 / 110 |
| 302 | > pred\_tree <- predict(tree\_df, test\_df, type = "class")  > pred\_tree  [1] setosa setosa setosa setosa setosa setosa setosa  [8] setosa setosa setosa setosa setosa setosa setosa  [15] setosa versicolor versicolor versicolor versicolor versicolor versicolor  [22] versicolor versicolor versicolor versicolor versicolor versicolor versicolor  [29] versicolor versicolor virginica virginica virginica virginica virginica  [36] virginica virginica virginica virginica virginica virginica virginica  [43] virginica virginica virginica  Levels: setosa versicolor virginica |
| 303 | > install.packages("e1071")  > library(e1071)  > confusionMatrix(pred\_tree, test\_df$Species)  Confusion Matrix and Statistics  Reference  Prediction setosa versicolor virginica  setosa 15 0 0  versicolor 0 15 0  virginica 0 0 15  Overall Statistics    Accuracy : 1  95% CI : (0.9213, 1)  No Information Rate : 0.3333  P-Value [Acc > NIR] : 2.2e-16    Kappa : 1  Mcnemar's Test P-Value : NA  Statistics by Class:  Class: setosa Class: versicolor Class: virginica  Sensitivity 1.0000 1.0000 1.0000  Specificity 1.0000 1.0000 1.0000  Pos Pred Value 1.0000 1.0000 1.0000  Neg Pred Value 1.0000 1.0000 1.0000  Prevalence 0.3333 0.3333 0.3333  Detection Rate 0.3333 0.3333 0.3333  Detection Prevalence 0.3333 0.3333 0.3333  Balanced Accuracy 1.0000 1.0000 1.0000 |
| 305 | > pred\_tree\_tune <- predict(tree\_df\_tune, test\_df, type = "class")  > confusionMatrix(pred\_tree\_tune, test\_df$Species)  Confusion Matrix and Statistics  Reference  Prediction setosa versicolor virginica  setosa 15 0 0  versicolor 0 15 0  virginica 0 0 15  Overall Statistics    Accuracy : 1  95% CI : (0.9213, 1)  No Information Rate : 0.3333  P-Value [Acc > NIR] : 2.2e-16    Kappa : 1  Mcnemar's Test P-Value : NA  Statistics by Class:  Class: setosa Class: versicolor Class: virginica  Sensitivity 1.0000 1.0000 1.0000  Specificity 1.0000 1.0000 1.0000  Pos Pred Value 1.0000 1.0000 1.0000  Neg Pred Value 1.0000 1.0000 1.0000  Prevalence 0.3333 0.3333 0.3333  Detection Rate 0.3333 0.3333 0.3333  Detection Prevalence 0.3333 0.3333 0.3333  Balanced Accuracy 1.0000 1.0000 1.0000 |
| 311 | > str(spam7)  'data.frame': 4601 obs. of 7 variables:  $ crl.tot : num 278 1028 2259 191 191 ...  $ dollar : num 0 0.18 0.184 0 0 0 0.054 0 0.203 0.081 ...  $ bang : num 0.778 0.372 0.276 0.137 0.135 0 0.164 0 0.181 0.244 ...  $ money : num 0 0.43 0.06 0 0 0 0 0 0.15 0 ...  $ n000 : num 0 0.43 1.16 0 0 0 0 0 0 0.19 ...  $ make : num 0 0.21 0.06 0 0 0 0 0 0.15 0.06 ...  $ yesno : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 2 2 2 2 ...  > head(spam7)  crl.tot dollar bang money n000 make yesno  1 278 0.000 0.778 0.00 0.00 0.00 y  2 1028 0.180 0.372 0.43 0.43 0.21 y  3 2259 0.184 0.276 0.06 1.16 0.06 y  4 191 0.000 0.137 0.00 0.00 0.00 y  5 191 0.000 0.135 0.00 0.00 0.00 y  6 54 0.000 0.000 0.00 0.00 0.00 y |
| 312 | > rf\_df\_train <- randomForest(yesno ~., data = train\_df, importance = TRUE)  > rf\_df\_train  Call:  randomForest(formula = yesno ~ ., data = train\_df, importance = TRUE)  Type of random forest: classification  Number of trees: 500  No. of variables tried at each split: 2  OOB estimate of error rate: 11.3%  Confusion matrix:  n y class.error  n 1843 109 0.05584016  y 255 1015 0.20078740 |
| 313 | > pred\_rf\_df <- predict(rf\_df\_train, test\_df)  > confusionMatrix(pred\_rf\_df , test\_df$yesno)  Confusion Matrix and Statistics  Reference  Prediction n y  n 782 135  y 54 408    Accuracy : 0.8629  95% CI : (0.8437, 0.8807)  No Information Rate : 0.6062  P-Value [Acc > NIR] : < 2.2e-16    Kappa : 0.7052  Mcnemar's Test P-Value : 5.915e-09    Sensitivity : 0.9354  Specificity : 0.7514  Pos Pred Value : 0.8528  Neg Pred Value : 0.8831  Prevalence : 0.6062  Detection Rate : 0.5671  Detection Prevalence : 0.6650  Balanced Accuracy : 0.8434    'Positive' Class : n |
| 314 | > importance(rf\_df\_train)  n y MeanDecreaseAccuracy MeanDecreaseGini  crl.tot 47.43758 47.218993 65.58510 178.20642  dollar 60.50621 51.960459 77.74324 316.36533  bang 89.61389 87.708136 108.76642 425.69241  money 28.00734 43.360603 44.98379 143.64712  n000 46.86500 8.719742 49.88695 90.47921  make 12.15037 17.185885 21.24163 29.48143 |
| 320 | > df\_kmeans  K-means clustering with 3 clusters of sizes 46, 50, 54  Cluster means:  Petal.Length Petal.Width  1 5.626087 2.047826  2 1.462000 0.246000  3 4.292593 1.359259  Clustering vector:  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  [47] 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 1 3 3 3 3 3 3 3 3  [93] 3 3 3 3 3 3 3 3 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 3 1 1 3 3 1 1 1 1 1 1 1 1 1 1  [139] 3 1 1 1 1 1 1 1 1 1 1 1  Within cluster sum of squares by cluster:  [1] 15.16348 2.02200 14.22741  (between\_SS / total\_SS = 94.3 %)  Available components:  [1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss"  [7] "size" "iter" "ifault" |
| 321 | > table(df\_kmeans$cluster, df$Species)    setosa versicolor virginica  1 0 2 44  2 50 0 0  3 0 48 6 |
|  | > accurancy\_df <- (50 + 48 + 44) / 150  > accurancy\_df  [1] 0.9466667 |
| 323 | > df\_kmeans$centers  Petal.Length Petal.Width  1 5.626087 2.047826  2 1.462000 0.246000  3 4.292593 1.359259  > df\_kmeans\_center <- as.data.frame(df\_kmeans$centers) |
| 325 | > df\_table <- as.matrix.data.frame(table(df$Species, df$cluster))  > df\_table  > which\_setosa <- which.max(df\_table[1, ])  > which\_versicolor <- which.max(df\_table[2, ])  > which\_virginica <- which.max(df\_table[3, ])  > for( i in 1:150) {  + if (df[i,5] == 'setosa' && df[i,6] == which\_setosa) {  + df[i,7] = 'true'  + } else if (df[i,5] == 'versicolor' && df[i,6] == which\_versicolor) {  + df[i,7] = 'true'  + } else if (df[i,5] == 'virginica' && df[i,6] == which\_virginica) {  + df[i,7] = 'true'  + } else {df[i,7] = 'false'}  + } |
|  | 위 코드 하단의 텍스트를 아래 문장으로 일괄수정 해주세요.  ——————  우선 Species와 cluster의 정확도 검증을 위해 사용한 table을 객체화해주세요. table을 기준으로, ‘which.max’ 함수를 사용하여 Species와 맵핑 된 cluster번호(1~3)을 찾습니다. 정확도가 94.6%였기때문에, Species별로 맵핑된 cluster 숫자가 가장 높은 거이 해당 Species의 cluster번호입니다.  그리고 오분류 맵핑값을 찾기 위해 for문과 if문을 사용했습니다. for문의 내부를 보면 if-else의 반복입니다. if문을 활용해, 5번째 열인 Species와 6번째열인 cluster의 번호가 조건과 같으면 true, 그렇지않으면 false를 7번째 열에 입력합니다. |
| 326 | > df[df$V7 == 'false', ]  Sepal.Length Sepal.Width Petal.Length Petal.Width Species cluster V7  78 6.7 3.0 5.0 1.7 versicolor 2 false  84 6.0 2.7 5.1 1.6 versicolor 2 false  107 4.9 2.5 4.5 1.7 virginica 1 false  120 6.0 2.2 5.0 1.5 virginica 1 false  127 6.2 2.8 4.8 1.8 virginica 1 false  139 6.0 3.0 4.8 1.8 virginica 1 false |
| 339 | > str(Orange)  Classes ‘nfnGroupedData’, ‘nfGroupedData’, ‘groupedData’ and 'data.frame': 35 obs. of 3 variables:  $ Tree : Ord.factor w/ 5 levels "3"<"1"<"5"<"2"<..: 2 2 2 2 2 2 2 4 4 4 ...  $ age : num 118 484 664 1004 1231 ...  $ circumference: num 30 58 87 115 120 142 145 33 69 111 ...  - attr(\*, "formula")=Class 'formula' language circumference ~ age | Tree  .. ..- attr(\*, ".Environment")=<environment: R\_EmptyEnv>  - attr(\*, "labels")=List of 2  ..$ x: chr "Time since December 31, 1968"  ..$ y: chr "Trunk circumference"  - attr(\*, "units")=List of 2  ..$ x: chr "(days)"  ..$ y: chr “(mm)"  > head(Orange)  Grouped Data: circumference ~ age | Tree  Tree age circumference  1 1 118 30  2 1 484 58  3 1 664 87  4 1 1004 115  5 1 1231 120  6 1 1372 142 |
|  | sqldf(  ' SELECT(뽑고 싶은 컬럼)  FROM(데이터셋) '  ) |
| 340 | > sqldf('select Tree from Orange')  Tree  1 1  2 1  3 1  4 1  5 1  6 1  (하략) |
|  | > sqldf('select Tree, age from Orange')  Tree age  1 1 118  2 1 484  3 1 664  4 1 1004  5 1 1231  6 1 1372  (하략) |
| 341 | > sqldf('select Tree, age, circumference from Orange')  Tree age circumference  1 1 118 30  2 1 484 58  3 1 664 87  4 1 1004 115  5 1 1231 120  6 1 1372 142  (하략) |
|  | > sqldf('select \* from Orange')  Tree age circumference  1 1 118 30  2 1 484 58  3 1 664 87  4 1 1004 115  5 1 1231 120  6 1 1372 142  (하략) |
| 342 | > sqldf('SELECT \* FROM Orange WHERE age > 1500')  Tree age circumference  1 1 1582 145  2 2 1582 203  3 3 1582 140  4 4 1582 214  5 5 1582 177 |
|  | > sqldf('SELECT \* FROM Orange WHERE age > 1500 AND Tree = 5')  Tree age circumference  1 5 1582 177 |
| 343 | > sqldf('SELECT \* FROM Orange WHERE Tree IN ("1","2")')  Tree age circumference  1 1 118 30  2 1 484 58  3 1 664 87  4 1 1004 115  5 1 1231 120  6 1 1372 142  7 1 1582 145  8 2 118 33  9 2 484 69  10 2 664 111  11 2 1004 156  12 2 1231 172  13 2 1372 203  14 2 1582 203 |
| 344 | > sqldf('SELECT \* FROM Orange WHERE age LIKE "10%" ')  Tree age circumference  1 1 1004 115  2 2 1004 156  3 3 1004 108  4 4 1004 167  5 5 1004 125 |
|  | > sqldf('SELECT \* FROM Orange WHERE Tree = 3 AND circumference LIKE "3%" ')  Tree age circumference  1 3 118 30 |
| 345 | > sqldf('SELECT Tree FROM Orange GROUP BY Tree')  Tree  1 1  2 2  3 3  4 4  5 5 |
| 346 | > sqldf('SELECT Tree, SUM(age), SUM(circumference), AVG(age), AVG(circumference)  + FROM Orange GROUP BY Tree')  Tree SUM(age) SUM(circumference) AVG(age) AVG(circumference)  1 1 6455 697 922.1429 99.57143  2 2 6455 947 922.1429 135.28571  3 3 6455 658 922.1429 94.00000  4 4 6455 975 922.1429 139.28571  5 5 6455 778 922.1429 111.14286 |
| 347 | > sqldf('SELECT Tree, SUM(age), SUM(circumference), AVG(age), AVG(circumference)  + FROM Orange GROUP BY Tree  + HAVING SUM(circumference) > 900')  Tree SUM(age) SUM(circumference) AVG(age) AVG(circumference)  1 2 6455 947 922.1429 135.2857  2 4 6455 975 922.1429 139.2857 |
| 348 | > sqldf('SELECT \* FROM Orange WHERE Tree = 1')  Tree age circumference  1 1 118 30  2 1 484 58  3 1 664 87  4 1 1004 115  5 1 1231 120  6 1 1372 142  7 1 1582 145  > sqldf('SELECT \* FROM Orange WHERE Tree = 1 ORDER BY age desc')  Tree age circumference  1 1 1582 145  2 1 1372 142  3 1 1231 120  4 1 1004 115  5 1 664 87  6 1 484 58  7 1 118 30 |
| 349 | > sqldf('SELECT SUM(circumference) AS SUM\_cir FROM Orange')  SUM\_cir  1 4055 |
| 362 | > str(df)  'data.frame ': 5043 obs. of 28 variables:  $ color : chr "Color" "Color" "Color" "Color" ...  $ director\_name : chr "James Cameron" "Gore Verbinski" "Sam Mendes" "Christopher Nolan" ...  $ num\_critic\_for\_reviews : int 723 302 602 813 NA 462 392 324 635 375 ...  $ duration : int 178 169 148 164 NA 132 156 100 141 153 ...  $ director\_facebook\_likes : int 0 563 0 22000 131 475 0 15 0 282 ...  $ actor\_3\_facebook\_likes : int 855 1000 161 23000 NA 530 4000 284 19000 10000 ...  $ actor\_2\_name : chr "Joel David Moore" "Orlando Bloom" "Rory Kinnear" "Christian Bale" ...  $ actor\_1\_facebook\_likes : int 1000 40000 11000 27000 131 640 24000 799 26000 25000 ...  (하략) |
| 363 | > head(df[ ,1:5])  color director\_name num\_critic\_for\_reviews duration director\_facebook\_likes  1 Color James Cameron 723 178 0  2 Color Gore Verbinski 302 169 563  3 Color Sam Mendes 602 148 0  4 Color Christopher Nolan 813 164 22000  5 Doug Walker NA NA 131  6 Color Andrew Stanton 462 132 475 |
| 364 | # maximum movie making  > head(sort(table(df$director\_name), decreasing = TRUE), 10)  Steven Spielberg Woody Allen Clint Eastwood Martin Scorsese  104 26 22 20 20  Ridley Scott Spike Lee Steven Soderbergh Tim Burton Renny Harlin  17 16 16 16 15 |
| 365 | # maximum movie making  > install.packages("dplyr")  > library(dplyr)  >df %>% count(director\_name) %>% filter(director\_name != '') %>% arrange(desc(n))  # A tibble: 2,398 x 2  director\_name n  <chr> <int>  1 Steven Spielberg 26  2 Woody Allen 22  3 Clint Eastwood 20  4 Martin Scorsese 20  5 Ridley Scott 17  6 Spike Lee 16  7 Steven Soderbergh 16  8 Tim Burton 16  9 Renny Harlin 15  10 Oliver Stone 14  # ... with 2,388 more rows |
| 366 | # maximum gross  > max\_gross <- head(df[order(-df$gross),], 10)  > max\_gross[ , c("director\_name", "movie\_title", "actor\_1\_name", "gross", "imdb\_score")]  director\_name movie\_title actor\_1\_name  1 James Cameron Avatar CCH Pounder  27 James Cameron Titanic Leonardo DiCaprio  30 Colin Trevorrow Jurassic World Bryce Dallas Howard  18 Joss Whedon The Avengers Chris Hemsworth  795 Joss Whedon The Avengers Chris Hemsworth  67 Christopher Nolan The Dark Knight Christian Bale  241 George Lucas Star Wars: Episode I - The Phantom Menace Natalie Portman  3025 George Lucas Star Wars: Episode IV - A New Hope Harrison Ford  9 Joss Whedon Avengers: Age of Ultron Chris Hemsworth  4 Christopher Nolan The Dark Knight Rises Tom Hardy    gross imdb\_score  1 760505847 7.9  27 658672302 7.7  30 652177271 7.0  18 623279547 8.1  795 623279547 8.1  67 533316061 9.0  241 474544677 6.5  3025 460935665 8.7  9 458991599 7.5  4 448130642 8.5 |
| 367 | # install package 'sqldf'  > install.packages("sqldf")  > library(sqldf) |
|  | # maximum gross\_edit  > df\_grp <- sqldf("SELECT director\_name, movie\_title, actor\_1\_name, gross, imdb\_score  + FROM df  + GROUP BY director\_name, movie\_title, actor\_1\_name, gross, imdb\_score  + ORDER BY gross DESC") |
| 368 | # maximum gross\_edit  > head(df\_grp,10)  director\_name movie\_title actor\_1\_name  1 James Cameron Avatar CCH Pounder  2 James Cameron Titanic Leonardo DiCaprio  3 Colin Trevorrow Jurassic World Bryce Dallas Howard  4 Joss Whedon The Avengers Chris Hemsworth  5 Christopher Nolan The Dark Knight Christian Bale  6 George Lucas Star Wars: Episode I - The Phantom Menace Natalie Portman  7 George Lucas Star Wars: Episode IV - A New Hope Harrison Ford  8 Joss Whedon Avengers: Age of Ultron Chris Hemsworth  9 Christopher Nolan The Dark Knight Rises Tom Hardy  10 Andrew Adamson Shrek 2 Rupert Everett    gross imdb\_score  1 760505847 7.9  2 658672302 7.7  3 652177271 7.0  4 623279547 8.1  5 533316061 9.0  6 474544677 6.5  7 460935665 8.7  8 458991599 7.5  9 448130642 8.5  10 436471036 7.2 |
| 369 | # maximum imdb\_score  > max\_score <- df[order(-df$imdb\_score),]  > head(max\_score[ , c("director\_name", "movie\_title", "imdb\_score")], 10)  director\_name movie\_title imdb\_score  2766 John Blanchard Towering Inferno 9.5  1938 Frank Darabont The Shawshank Redemption 9.3  3467 Francis Ford Coppola The Godfather 9.2  2825 Dekalog 9.1  3208 Dekalog 9.1  4410 John Stockwell Kickboxer: Vengeance 9.1  67 Christopher Nolan The Dark Knight 9.0  2838 Francis Ford Coppola The Godfather: Part II 9.0  3482 Fargo 9.0  340 Peter Jackson The Lord of the Rings: The Return of the King 8.9 |
|  | > summary(df$imdb\_score)  Min. 1st Qu. Median Mean 3rd Qu. Max.  1.600 5.800 6.600 6.442 7.200 9.500 |
| 370 | # imdb score\_hist\_basic  > hist(df$imdb\_score, main = "score hist", xlab = "score") |
|  | # install 'ggplot2'  > install.packages("ggplot2")  > library(ggplot) |
| 371 | # imdb score\_ggplot\_hist  > g1 <- ggplot(df, aes(x=df$imdb\_score))  > g1 + geom\_histogram(breaks=seq(0, 10, by = 0.5), fill = "white", colour = "black") |
| 372 | > g1 + geom\_histogram(breaks=seq(0, 10, by = 0.5), aes(fill = ..count..), colour = "black") |
| 373 | > ggplot(df, aes(y=df$gross, x=df$imdb\_score)) +  + geom\_point(aes(color = df$title\_year), size = 2, alpha = 0.7) |
| 374 | > g1 <- ggplot(df, aes(y=df$gross, x=df$imdb\_score))  > p1 <- geom\_point(aes(color = df$title\_year), size = 2, alpha = 0.7) |
|  | > g1 + p1 +  + annotate("rect", xmin = 5.9, xmax = 8.0, ymin = 0, ymax = 8e+08,  + alpha = 0.3, fill = "orange") + stat\_smooth(method = lm, colour = "green") |
| 376 | > install.packages("corrplot")  > library(corrplot) |
|  | > df\_cor <- df[, c(3:6, 8, 9, 13, 14,1 9, 23, 24, 25, 26, 28)] |
| 377 | > df\_cor <- na.omit(df\_cor) |
|  | > str(df\_cor)  'data.frame': 3879 obs. of 14 variables:  (하략) |
|  | > df\_cor <- cor(df\_cor)  > df\_cor <- round(df\_cor, digits = 2) |
|  | > df\_cor  num\_critic\_for\_reviews duration director\_facebook\_likes  num\_critic\_for\_reviews 1.00 0.24 0.18  duration 0.24 1.00 0.18  director\_facebook\_likes 0.18 0.18 1.00  actor\_3\_facebook\_likes 0.26 0.13 0.12  actor\_1\_facebook\_likes 0.17 0.09 0.09  gross 0.48 0.25 0.14  num\_voted\_users 0.60 0.34 0.30  cast\_total\_facebook\_likes 0.24 0.12 0.12  num\_user\_for\_reviews 0.57 0.36 0.22  budget 0.11 0.07 0.02  title\_year 0.39 -0.13 -0.05  actor\_2\_facebook\_likes 0.26 0.13 0.12  imdb\_score 0.34 0.36 0.19  movie\_facebook\_likes 0.70 0.22 0.16  (하략) |
| 378 | > corrplot(df\_cor) |
| 379 | > corrplot(df\_cor, # 데이터셋  + type = {"lower"}, # 플롯 아랫쪽 출력  + order="AOE", # cor 높은 변수끼리 배열  + tl.col ="black", tl.cex = 0.7, tl.srt = 45, # 변수 text label 설정  + addCoef.col = "black", number.cex = 0.6) # 상관계수 text 설정 |
| 382 | # regression analysis(linear model)  > df\_lm <- df[, c(3:6, 8, 9, 13, 14, 19, 23, 24, 25, 26, 27, 28)] |
|  | > df\_lm\_model <- lm(imdb\_score ~., data = df\_lm)  > summary(df\_lm\_model) |
| 383 | Call:  lm(formula = imdb\_score ~ ., data = df\_lm)  Residuals:  Min 1Q Median 3Q Max  -4.4048 -0.4644 0.0767 0.5784 2.4642  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 4.695e+01 3.347e+00 14.029 < 2e-16 \*\*\*  num\_critic\_for\_reviews 2.572e-03 1.960e-04 13.123 < 2e-16 \*\*\*  duration 1.058e-02 6.786e-04 15.590 < 2e-16 \*\*\*  director\_facebook\_likes 4.163e-06 4.790e-06 0.869 0.38489  actor\_3\_facebook\_likes 6.833e-05 2.142e-05 3.190 0.00143 \*\*  actor\_1\_facebook\_likes 7.144e-05 1.304e-05 5.477 4.60e-08 \*\*\*  gross -1.957e-09 2.690e-10 -7.275 4.18e-13 \*\*\*  num\_voted\_users 3.713e-06 1.740e-07 21.343 < 2e-16 \*\*\*  cast\_total\_facebook\_likes -7.050e-05 1.300e-05 -5.423 6.21e-08 \*\*\*  num\_user\_for\_reviews -6.094e-04 5.750e-05 -10.598 < 2e-16 \*\*\*  budget -4.232e-11 6.199e-11 -0.683 0.49477  title\_year -2.093e-02 1.675e-03 -12.495 < 2e-16 \*\*\*  actor\_2\_facebook\_likes 7.413e-05 1.372e-05 5.403 6.97e-08 \*\*\*  aspect\_ratio -7.269e-02 4.096e-02 -1.775 0.07604 .  movie\_facebook\_likes -2.149e-06 9.543e-07 -2.252 0.02439 \*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.8507 on 3792 degrees of freedom  (1236 observations deleted due to missingness)  Multiple R-squared: 0.3549, Adjusted R-squared: 0.3525  F-statistic: 149 on 14 and 3792 DF, p-value: < 2.2e-16 |
| 384 | # 다중공선성 유무 판단  > install.packages("car")  > library(car)  > vif(df\_lm\_model)  num\_critic\_for\_reviews duration director\_facebook\_likes  3.095085 1.248428 1.121677  actor\_3\_facebook\_likes actor\_1\_facebook\_likes gross  8.555097 213.194559 1.874748  num\_voted\_users cast\_total\_facebook\_likes num\_user\_for\_reviews  3.659903 321.922317 2.924166  budget title\_year actor\_2\_facebook\_likes  1.018927 1.462761 20.237434  aspect\_ratio movie\_facebook\_likes  1.094579 2.213150 |
|  | # regression\_tuning  > df\_lm <- df\_lm[, c(1:3, 6:7, 9:11, 13:14)]  > df\_lm <- na.omit(df\_lm) |
| 385 | > install.packages("caret")  > library(caret)  > set.seed(11111)  > temp\_train <- createDataPartition(y = df\_lm$imdb\_score, p = 0.7, list = FALSE)  > df\_train <- df\_lm[temp\_train, ]  > df\_test <- df\_lm[-temp\_train, ] |
|  | > count(df\_train)  # A tibble: 1 x 1  n  <int>  1 2672  > count(df\_test)  # A tibble: 1 x 1  n  <int>  1 1143 |
|  | # regression analysis\_train  > lm\_df <- lm(imdb\_score ~., data = df\_train)  > summary(lm\_df)  Call:  lm(formula = imdb\_score ~ ., data = df\_train)  Residuals:  Min 1Q Median 3Q Max  -4.4476 -0.4779 0.0844 0.5765 2.4807  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 4.958e+01 3.939e+00 12.589 < 2e-16 \*\*\*  num\_critic\_for\_reviews 2.354e-03 1.987e-04 11.845 < 2e-16 \*\*\*  duration 9.638e-03 8.107e-04 11.888 < 2e-16 \*\*\*  director\_facebook\_likes 8.670e-06 5.738e-06 1.511 0.131  gross -2.079e-09 3.041e-10 -6.837 1.00e-11 \*\*\*  num\_voted\_users 3.479e-06 1.997e-07 17.416 < 2e-16 \*\*\*  num\_user\_for\_reviews -5.065e-04 6.690e-05 -7.571 5.09e-14 \*\*\*  budget -4.467e-11 6.765e-11 -0.660 0.509  title\_year -2.225e-02 1.972e-03 -11.281 < 2e-16 \*\*\*  aspect\_ratio -5.272e-02 4.547e-02 -1.159 0.246  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.8632 on 2662 degrees of freedom  Multiple R-squared: 0.3433, Adjusted R-squared: 0.3411  F-statistic: 154.6 on 9 and 2662 DF, p-value: < 2.2e-16 |
| 386 | # regression\_backward  > lm\_df\_bw <- step(lm\_df, direction = “backward") |
| 387 | Start: AIC=-776.06  imdb\_score ~ num\_critic\_for\_reviews + duration + director\_facebook\_likes +  gross + num\_voted\_users + num\_user\_for\_reviews + budget +  title\_year + aspect\_ratio  Df Sum of Sq RSS AIC  - budget 1 0.325 1983.9 -777.62  - aspect\_ratio 1 1.002 1984.6 -776.71  <none> 1983.6 -776.06  - director\_facebook\_likes 1 1.701 1985.3 -775.77  - gross 1 34.827 2018.4 -731.55  - num\_user\_for\_reviews 1 42.708 2026.3 -721.14  - title\_year 1 94.831 2078.4 -653.28  - num\_critic\_for\_reviews 1 104.543 2088.1 -640.82  - duration 1 105.308 2088.9 -639.84  - num\_voted\_users 1 226.027 2209.6 -489.72  (중략)  Step: AIC=-778.29  imdb\_score ~ num\_critic\_for\_reviews + duration + director\_facebook\_likes +  gross + num\_voted\_users + num\_user\_for\_reviews + title\_year  Df Sum of Sq RSS AIC  <none> 1984.9 -778.29  - director\_facebook\_likes 1 1.711 1986.6 -777.99  - gross 1 35.064 2020.0 -733.51  - num\_user\_for\_reviews 1 42.936 2027.8 -723.11   * title\_year 1 102.627 2087.5 -645.59 * num\_critic\_for\_reviews 1 103.631 2088.5 -644.31   - duration 1 104.313 2089.2 -643.44  - num\_voted\_users 1 226.980 2211.9 -490.98 |
|  | > summary(lm\_df\_bw)  Call:  lm(formula = imdb\_score ~ num\_critic\_for\_reviews + duration +  director\_facebook\_likes + gross + num\_voted\_users + num\_user\_for\_reviews +  title\_year, data = df\_train)  Residuals:  Min 1Q Median 3Q Max  -4.4290 -0.4795 0.0843 0.5754 2.4716  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 5.041e+01 3.878e+00 12.997 < 2e-16 \*\*\*  num\_critic\_for\_reviews 2.338e-03 1.983e-04 11.794 < 2e-16 \*\*\*  duration 9.477e-03 8.010e-04 11.832 < 2e-16 \*\*\*  director\_facebook\_likes 8.694e-06 5.738e-06 1.515 0.13  gross -2.083e-09 3.037e-10 -6.860 8.53e-12 \*\*\*  num\_voted\_users 3.485e-06 1.996e-07 17.454 < 2e-16 \*\*\*  num\_user\_for\_reviews -5.078e-04 6.689e-05 -7.591 4.35e-14 \*\*\*  title\_year -2.270e-02 1.934e-03 -11.736 < 2e-16 \*\*\*---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.8632 on 2664 degrees of freedom  Multiple R-squared: 0.3429, Adjusted R-squared: 0.3411  F-statistic: 198.6 on 7 and 2664 DF, p-value: < 2.2e-16 |
| 388 | # anova analysis  > anova(lm\_df\_bw, lm\_df)  Analysis of Variance Table  Model 1: imdb\_score ~ num\_critic\_for\_reviews + duration + gross + num\_voted\_users +  num\_user\_for\_reviews + title\_year  Model 2: imdb\_score ~ num\_critic\_for\_reviews + duration + director\_facebook\_likes +  gross + num\_voted\_users + num\_user\_for\_reviews + budget + title\_year + aspect\_ratio  Res.Df RSS Df Sum of Sq F Pr(>F)  1 2665 1935.8  2 2662 1934.4 3 1.3275 0.6089 0.6092 |
| 389 | > pred <- predict(lm\_df\_bw, df\_test)  > pred  1 9 11 13 14 15 16 18 19 23 29  8.128007 7.571242 7.021112 6.941294 7.041050 7.271850 7.793424 9.308548 7.634567 7.296784 7.045200  34 38 40 42 44 45 51 54 55 57 58  6.963453 6.998954 7.346210 6.281834 7.473301 7.147062 7.796968 7.158480 6.657081 6.993961 6.249346  (중략) |
| 390 | > cor(pred, df\_test$imdb\_score)  [1] 0.5971406 |