N-мерни променливи и изследване на връзките между тях

Какво представляват "n"-мерните данни?

Едномерните данни, това са масиви/листа от обекти (числа, стрингове, дати, друг тип обекти).

При двумерните данни имаме колекция от едномерни данни. Тоест, представянето е във формата на

матрици, data frame-ове или друга подобна структура, при която най-често по редове са представени

примерите/елементите, а по колони техните признаци (променливите).

Пример за многомерни данни е

data("mtcars")

head(mtcars)

Нека да изследваме обема на двигателя за въпроснтие коли. Първо ще построим хистограма

hist(x = mtcars$disp, col = "red", xlab = "Displacement (u.in.)", main = "Histogram")

summary(mtcars$disp)

sd(mtcars$disp)

abline(v = mean(mtcars$disp), lwd = 2, lty = 4)

abline(v = median(mtcars$disp), lwd = 2, lty = 3, col = "blue")

От хистограмата се вижда, че имаме два пика. Тоест, разпределението на променливата е

бимодално. Черната вертикалана прекъсната линия показва къде се намира средната стойност,

а синята прекъсната - медината. И в двата случая, малко трудно можем да приемем, че тпва е

очакването на разпределението.

Нека сега да проверим, какво би станало, ако групираме данните по броя на цилиндрите

disp_cyl4 <- mtcars$disp[which(mtcars$cyl == 4)]

disp_cyl6 <- mtcars$disp[which(mtcars$cyl == 6)]

```
disp_cyl8 <- mtcars$disp[which(mtcars$cyl == 8)]



par(mfrow = c(2, 2))

hist(x = disp_cyl4, col = "red", xlab = "4 cylinders", main = "Histogram of
displacement (u.in.)")

hist(x = disp_cyl6, col = "lightblue", xlab = "6 cylinders", main = "Histogram
of displacement (u.in.)")

hist(x = disp_cyl8, col = "forestgreen", xlab = "8 cylinders", main = "Histogram
of displacement (u.in.)")

par(mfrow = c(1, 1))




summary(disp_cyl4)

sd(disp_cyl4)



summary(disp_cyl6)

sd(disp_cyl6)



summary(disp_cyl8)

sd(disp_cyl8)



# За групата на двигетелите, които имат 4 цилиндъра, все още не може да получим
добра оценка

# за очакването, по за другите две групи - можем, защото имаме по един връх.

# Анализирайки зависимостите на една променлива от други променливи, ние
успяваме да подобрим

# оценките на параметрите, които са ни неогходи. По този начин правим прогнозите
си по-точни.







# Изследване на двумерни данни



# 1. Категорийни (обясняващи) VS категорийни (зависими)
```

```
# Връзките между тези променливи най-лесно се виждат с помощта на cross таблици
и barplot-ове.



# Пример: Направили сме хипотетично прочуване, което измерва дали студентите,
които пушат,

# учат по-малко.



smokes <- c("Y", "N", "N", "Y", "N", "Y", "Y", "Y", "N", "Y")

amount <- c("0 - 5 hours", "5 - 10 hours", "5 - 10 hours", "more than 10 hours",

"more than 10 hours", "0 - 5 hours", "5 - 10 hours", "0 - 5 hours",

"more than 10 hours", "5 - 10 hours")

table(amount, smokes)

# Данните показват, че пушачите учат по-малко от непушачите. Нека да разгледаме
резулатите

# не като честоти, а като проценти. За целта изпозлвае командата



prop.table(x = table(amount, smokes))

# Показва ни в коя група, колко процента от данните попадат.



prop.table(x = table(amount, smokes), margin = 1)

# Параметърът "margin" задава как желаем да изчисляваме процентите - по редове
или по колони.

# От данните виждаме, че имаме нарастване в процента на непушещите студентите,
спрямо броя на

# часовете, които отделят за учене.



# Сега ще разгледаме графичното представяне на данните.

barplot(table(smokes, amount))

# Малко трудно бихме видяли разликите, освен ако не са фрапиращи.

# В долния код ще се опитаме да нормализираме стойностите като използваме
процентните

# съотношения. При този подход, ясно се вижда превъзходствата на едни признаци в
една група,

# спрямо друга.

barplot(prop.table(x = table(smokes, amount), margin = 2))
```

```
# Друг подход е описаният по-долу

barplot(table(smokes, amount), beside = TRUE, legend.text = T)

# При този подход също лесно се забелязват разликите в отделните групи. В сегашния barplot

# сме задали и легенда



# Освен че можем да изведем легенда на графиката (legend.text = TRUE), то можем и да я

# попълним със стойности, които ни трябват. Попълването е показано в примера по-долу.

barplot(table(amount, smokes), main = "table(amount, smokes)", beside = TRUE,

legend.text = c("less than 5", "5 - 10", "more than 10"))







# 2. Категорийни (обясняващи) VS числови (зависими)

# Когато имаме такава конфигурация при връзките, то най-удачно е да използваме One-way ANOVA

# и t-test или техните непараемтрични еквиваленти. Тези анализи ще ги учим по-нататък в курса

# по статистика. Ако искаме да ги изследваме графично, удачно решение е boxplot графиките.



amount <- c(5, 5, 5, 13, 7, 11, 11, 9, 8, 9, 11, 8, 4, 5, 9, 5, 10, 5, 4, 10)

category <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)

tt <- boxplot(amount ~ category)

# Както се вижда, лесно могат да се сравнят двете категории. Средната дебела линия във всяки

# един boxplot е медианата, страните на правоъгълника са 1 и 3-ти квартил, а дължината на опашките

# са минималната имаксималните стойности, като са изключени потенциалните outlier-и.

# Тоест интерпретацията на тази графика е, че стойнсотите на първата група като цяло са

# по-големи защото и медианата и третия квартил за първата група са по-големи от тези на втората.
```

Отделно, минималната стойност и първия квартил за първата група съвпадат (= 5), докато минималната

стойност на втората група е 4.

3. Числови (обясняващи) VS категорийни (зависими)

Този случай на връзка е сходен с горния. Затова тук също можем да използваме One-way ANOVA или

t-test, непараметричните им еквавалenти и boxplot-ове. Също така можем да използваме и

логистичната регресия.

Този тип връзка ще бъде обяснена по-нататък

4. Числови (обясняващи) VS числови (зависими)

Това е може би групата, за която съществуват най-много похвати за анализи. Променливите от

числов тип могат да бъдат превърнати в категорийни и следователно за тях важат горнтие типове

анализи. Това, разбира се, би довело до загуба на информация, но в определени случаи е по-подходящо

заради по-голямата стабилност на моделите.

Похватите, които са характерни за изследването на този тип връзки са най-често корелационен

анализ и регресионен анализ, както и dotplot (графично представяне на връзката).

Почти винаги изследването на този тип връзка следва последователността dotplot, корелационен

анализ и регресионен анализ.

Да разгледаме пример от данните "mtcars". Интересуват ни променливите disp (обем на двигателя)

и wt (тегло)

```
plot(mtcars$disp, mtcars$wt)

# От графиката се вижда, че съществува положителна линейан връзка. Тоест с нарастване на обема

# на двигателя, нараства и теглото на автомобила. Следователно, можем да използваме линеен модел,

# за да моделираме връзката.







# ИНФОРМАЦИЯ, КОЯТО НЯМА ДА Я ИМА НА ИЗПИТА/КОНТРОЛНИТЕ



# Преди да продължим с корелационния и регресионния анализ, нека да разгледаме друг пример. Този

# път данните ще бъдат симулирани. И връзката няма да бъде линейна, а кубична.



set.seed(4455)

x <- runif(1000, -3, 3)

y <- x^3 - 3 + rnorm(length(x), sd = 2)

plot(x, y)

# Както се вижда от графиката, този тип връзка не прилича на линейна. Но чрез подходяща трансформация,

# връзката може да се представи като линейна. Например, ако създадем нова променлива

x3 <- x^3

# Тогава, новата променлива x3 е в линейна зависимост с променливата y

par(mfrow = c(1, 2))

plot(x, y)

plot(x3, y)

par(mfrow = c(1, 1))
```

```
# Корелационен анализ

# Корелационният анализ измерва силата на линейна връзка между две променливи. Коефициентът на

# корелация (rho) принадлежи на интервала [-1, 1]. Силата на връзката се определя от абсолютната

# стойност на rho. Въпреки, че силата на връзката с субективна, все пак можем да определим някакви

# нива.



N <- 1000

# abs(rho) = 1 - Детерминистична връзка (y = f(x)). За една стойност на x имаме точно една

# единствена стойност на y

set.seed(3654)

x1 <- runif(N)

y1 <- 3*x1 + 4

rho1 <- round(cor(x1, y1), 3)



# 0.9 <= abs(rho) < 1 - Много силна корелация на между x иу

set.seed(3654)

x2 <- runif(N)

y2 <- 3*x2+ 4 + rnorm(N, sd = 0.2)

rho2 <- round(cor(x2, y2), 3)



# 0.75 <= abs(rho) < 0.9 - Силна корелация на между x и y

set.seed(3654)

x3 <- runif(N)

y3 <- -3*x3 + 4 + rnorm(N, sd = 0.5)

rho3 <- round(cor(x3, y3), 3)
```

```
# 0.5 <= abs(rho) < 0.75 - Средна корелация на между x и y
set.seed(3654)
x4 <- runif(N)
y4 <- -3*x4 + 4 + 1*rnorm(N)
rho4 <- round(cor(x4, y4), 3)



# 0 <= abs(rho) < 0.5 - Слаба корелация на между x и y
set.seed(3654)
x5 <- runif(N)
y5 <- 3*x4 + 4 + 3*rnorm(N)
rho5 <- round(cor(x5, y5), 3)

par(mfrow = c(2, 3))
plot(x1, y1, main = paste("rho:", rho1))
abline(a = 4, b = 3, col = "red", lwd = 2)
plot(x2, y2, main = paste("rho:", rho2))
abline(a = 4, b = 3, col = "red", lwd = 2)
plot(x3, y3, main = paste("rho:", rho3))
abline(a = 4, b = -3, col = "red", lwd = 2)
plot(x4, y4, main = paste("rho:", rho4))
abline(a = 4, b = -3, col = "red", lwd = 2)
plot(x5, y5, main = paste("rho:", rho5))
abline(a = 4, b = 3, col = "red", lwd = 2)
par(mfrow = c(1, 1))

# От графиките се вижда, че колко по-разпръснати са наблюденията около правата,
# толкова корелацията намалява
```

```
# Командата за корелация е cor. С командата може да се изследват както връзките
# между две променливи, така и връзките между N-мерни ЧИСЛОВИ данни.

# Формулата за коралация ще я опишем с примера по-долу
X <- x3; Y <- y3


X_mean <- mean(X); Y_mean <- mean(Y)
XY <- (X - X_mean)*(Y - Y_mean)
XX <- (X - X_mean)^2; YY <- (Y - Y_mean)^2
sum(XY)/sqrt(sum(XX)*sum(YY)) # Стойността на корелацията


cor(x3, y3)


cor(mtcars$mpg, y = mtcars$hp)
# Връща ни само едно число - корелацията между двете променливи




cor(mtcars[, c("mpg", "disp", "hp", "drat", "wt", "qsec")])
# Връща СИМЕТРИЧНА матрица (A[i, j] == A[j, i]) с корелациите между отделните
# променливи.
# Интересно е, че, по главния диагонал, всички стойности са единици. Това е
# следствие от формулата



# Съществуват три основни вида корелации - Pearson, Spearman и Kendall. Първата
# корелация е параметрична оценка на връзката между две променливи, докато
останалите
```

```
# две - непараметрични.

# Тоест корелацията на Pearson е по-точна, но е неустойчива при наличието на
outlier-и

# Останалите две корелации са по-стабилни и не толкова точни.



# Най-лесно това ще го демонстрираме с примера по-долу

set.seed(4413)

x <- sort(rnorm(200, mean = 2))

y <- x + sqrt(1 - 0.8^2)*rnorm(length(x))

plot(x, y, main = paste("Pearson's rho:", round(cor(x, y), 2))) # корелацията е
0.85



# Нека обаче да добавим няколко outlier-а

x1 <- c(x, 3.4, 3, 3.8, 3.5, 4, 4.1)

y1 <- c(y, 17, 18.5, 19.2, 19, 20, 22)

plot(x1, y1)

abline(lm(y ~ x), col = "forestgreen", lwd = 2, lty = 4)

abline(lm(y1 ~ x1), col = "darkred", lwd = 2, lty = 3)

text(x = 0.5, y = 18, labels = paste0("Pearson's rho:", round(cor(x1, y1), 2)))

text(x = 0.5, y = 17, labels = paste0("Spearman's rho: ", round(cor(x1, y1,
method = "spearman"), 2)))
```