Основи класифікації та кластерний аналіз з R

План

- Регресія з факторними змінними
- Логістична регресія
- Кластерний аналіз

Регресія з факторними змінними

- Регресія від факторних змінних
- Регресія зі штучною змінною
- Логістична регресія
- Інші види регресії

Данні Світового банку

- Для провдення подальших досліджень будемо використовувати відкриті дані Світового банку:

```
require(wbstats)
```

- чисельність населення

```
pop_data <- wb(country = c("UKR"), indicator = "SP.POP.TOTL",
startdate = 1989, enddate = 2019)
```

- ВВП країни на душу населення:

```
gdp_data <- wb(country = c("UKR"), indicator = "NY.GDP.PCAP.KD",
startdate = 1989, enddate = 2019)
```

- Навчальний data.frame:

```
lm_data <- data.frame(date = as.numeric(pop_data$date), POP =
pop_data$value, GDP= gdp_data$value, country = pop_data$iso3c)

lm_data <- lm_data[order(lm_data$date),]
```

Регресія з факторними змінними

```
pop_data <- wb(country = c("RUS", "UKR", "BEL"), indicator = "SP.POP.TOTL", startdate = 1989, enddate = 2019)
gdp_data <- wb(country = c("RUS", "UKR", "BEL"), indicator = "NY.GDP.PCAP.KD", startdate = 1989, enddate = 2019)
lm_data <- data.frame(date = as.numeric(pop_data$date), POP = pop_data$value, GDP= gdp_data$value, country = pop_data$iso3c)
str(lm_data)
fit1_1 <- lm(GDP ~ country, lm_data)
summary(fit1_1)
fit1_2 <- lm(POP ~ country, lm_data)
summary(fit1_2)
```

```
Call:
lm(formula = GDP ~ country, data = lm_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-8117.6 -1098.0   191.7  1933.4  5767.4 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  40311.6      553.9   72.78   <2e-16 ***
countryRUS  -31470.2      783.3  -40.17   <2e-16 ***
countryUKR  -37583.4      783.3  -47.98   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2983 on 84 degrees of freedom
Multiple R-squared:  0.9693,	Adjusted R-squared:  0.9686 
F-statistic:  1326 on 2 and 84 DF,  p-value: < 2.2e-16
```

Штучна (dummy) змінна

- На підприємство працює 15 робітників (6 жінок, 9 чоловіків), кожний з яких відпрацював за остатній місяць деяку кількість годин та отримав за це деяку суму коштів.

- Визначити на основі dummy-регресії, чи існує ризниця в оплаті праці за гендерною ознакою?

№ (j)	Кількість відпрацьованих годин (x)	Стать (Sex)	D	Оплата праці (y) тис. грн.	
N	X	Sex	D	Y	dummy
1	200	ж	0	20	0
2	205	ж	0	21	0
…					
7	190	ч	1	23	190
8	150	ч	1	18	150
9	300	ч	1	35	300

Моделі зі штучними змінними з R

```
> summary (lm(Y ~ X + dummy ,data = dummy))

Call:
lm(formula = Y ~ X + dummy, data = dummy)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2048 -0.7193 -0.1309  0.6046  1.4089

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.829810   1.365994   0.607    0.555
X           0.096684   0.006414  15.074 3.67e-09 ***
dummy       0.018657   0.002382   7.833 4.66e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9712 on 12 degrees of freedom
Multiple R-squared:  0.9644,  Adjusted R-squared:  0.9584
F-statistic: 162.4 on 2 and 12 DF,  p-value: 2.044e-09
```

- Для жінок за кожну відпрацьовану годину оплата праці збільшується на 97 грн., а для чоловіків на 116 (97 +19) грн. Тобто чоловіки заробляють приблизно на 20% більш ніж жінки.

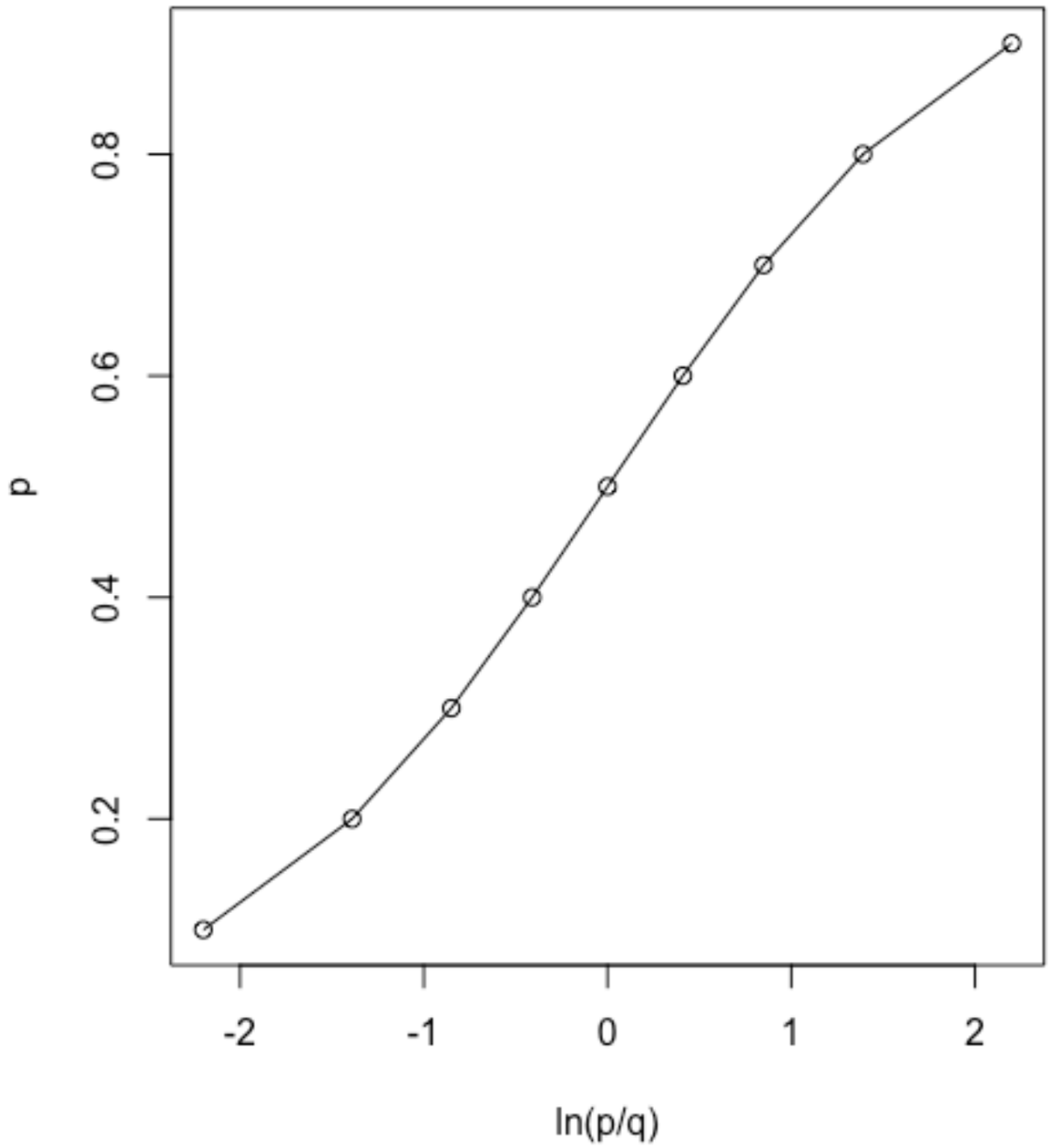
Логістична регресія

- Логістична регресія (англ. logistic regression) — статистичний регресійний метод, що застосовують у випадку, коли залежна змінна є категоріиною (факторною), тобто може набувати тільки двох значень (чи, загальніше, скінченої множини значень).

- Застосовується для розв'язання задачі бінарної класифікації

Функція ймовірності результату

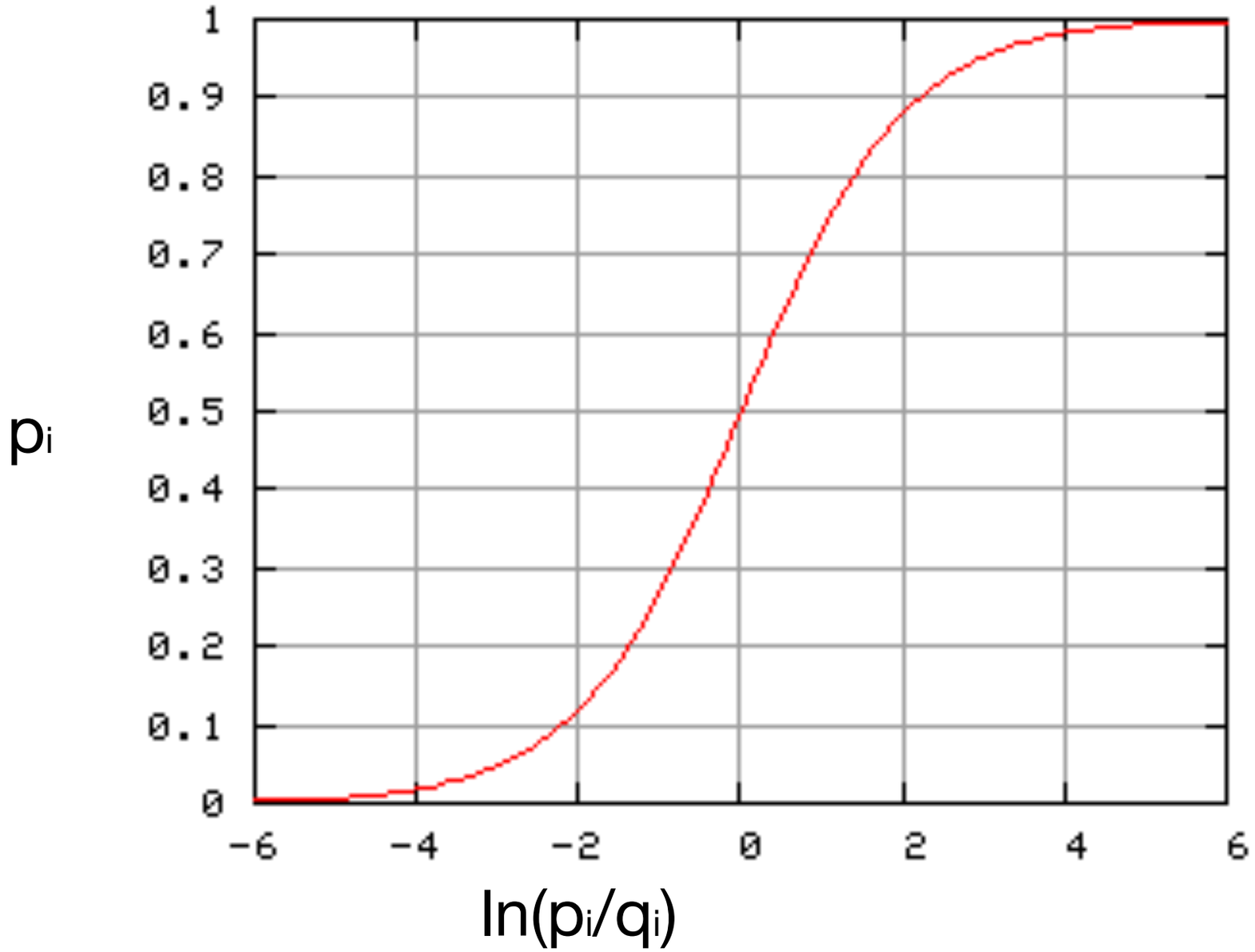
p	q=1-p	p/q	ln(p/q)
0,1	0,9	0,11	-2,20
0,2	0,8	0,25	-1,39
0,3	0,7	0,43	-0,85
0,4	0,6	0,67	-0,41
0,5	0,5	1,00	0,00
0,6	0,4	1,50	0,41
0,7	0,3	2,33	0,85
0,8	0,2	4,00	1,39
0,9	0,1	9,00	2,20



Логістична крива

$$ln(p/q) = ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \cdot x_1$$

$$p = \frac{e^{\beta_0+\beta_1 \cdot x_1}}{1 + e^{\beta_0+\beta_1 \cdot x_1}}$$



Узагальнені лінійні моделі

```
glm(formula, family = gaussian, data, weights, subset,
    na.action, start = NULL, etastart, mustart, offset,
    control = list(...), model = TRUE, method = "glm.fit",
    x = FALSE, y = TRUE, singular.ok = TRUE, contrasts = NULL, ...)
```

- Значення ключових аргументів:
  - formula – задає формулу (особливості) регресійної залежності, що оцінюється зо допомогою функції. Синтаксис аналогічний за функцію lm();
- family – визначає метод, який використовується для оцінки помилок та функції зв’язку, що використовуватиметься в моделі. Доступні наступні основні значення аргументу family:
  - binomial(link = “logit”), binomial(link = “probit”)
  - gaussian(link = “identity”)
  - gamma(link = “inverse”)
  - poisson(link = “log”)
  - quasi(link = “identity”, variance = “constant”)
  - quasibinomial(link = “logit”)
  - quasipoisson(link = “log”).
- data – джерело даних (data.frame) з якого походить інформація для аналізу;
- інші аргументи, значеннях яких співпадає х функцією lm().

Вхідні дані для подальшого аналізу

```
# GC.XPN.TOTL.GD.ZS, Expense (% of GDP)
data1 <- wb(country = c("USA"), indicator = "GC.XPN.TOTL.GD.ZS",
startdate = 1991, enddate = 2016)
length(data1$value)

#FP.CPI.TOTL.ZG, Inflation, consumer prices (annual %)
data2 <- wb(country = c("USA"), indicator = "FP.CPI.TOTL.ZG",
startdate = 1991, enddate = 2016)
length(data2$value)

# GDP growth, %
gdp_data <- wb(country = c("USA"), indicator = "NY.GDP.MKTP.KD.ZG",
startdate = 1991, enddate = 2016)
length(gdp_data$value)

lm_data <- data.frame(date = as.numeric( gdp_data$date ), EXPENCE =
data1$value, CPI = data2$value, GDP=ifelse(gdp_data$value>0,1,0),
country = gdp_data$iso3c)
```

Логістична модель

fit1_2 <- glm(GDP ~ EXPENCE + CPI, lm_data, family = "binomial", x = T)

summary(fit1_2)

```
> fit1_2 <- glm(GDP ~ EXPENCE + CPI, lm_data, family =binomial(link="logit"), x=T)
> summary(fit1_2)

Call:
glm(formula = GDP ~ EXPENCE + CPI, family = binomial(link = "logit"),
    data = lm_data, x = T)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-1.9632766   0.1394443   0.2981369   0.4311430   1.6428616

Coefficients:
             Estimate Std. Error  z value  Pr(>|z|)
(Intercept) 19.7380717  7.9352880  2.48738 0.0128688 *
EXPENCE     -0.7088646  0.3238364 -2.18896 0.0285998 *
CPI         -0.5706799  0.2135650 -2.67216 0.0075365 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 39.560776  on 46  degrees of freedom
Residual deviance: 27.540441  on 44  degrees of freedom
AIC: 33.540441

Number of Fisher Scoring iterations: 6
```

Ефективність

- AIC (Akaike Information Criteria) - це міра відповідності, яка збільшується за використання великої кількості параметрів моделі.
    - AIC -> min
- Null Deviance вказує на відхилення моделі без параметрів, тільки з вільним коефіцієнтом.
    - Null Deviance -> min
- Residual Deviance вказує на відхилення моделі при додаванні незалежних змінних
    - Residual Deviance -> min.

Інтерпретація результатів логістичної регресії та прогнози

$$p = \frac{e^{\beta_0+\beta_1\cdot x_1+\beta_2\cdot x_2}}{1+e^{\beta_0+\beta_1\cdot x_1+\beta_2\cdot x_2}}$$

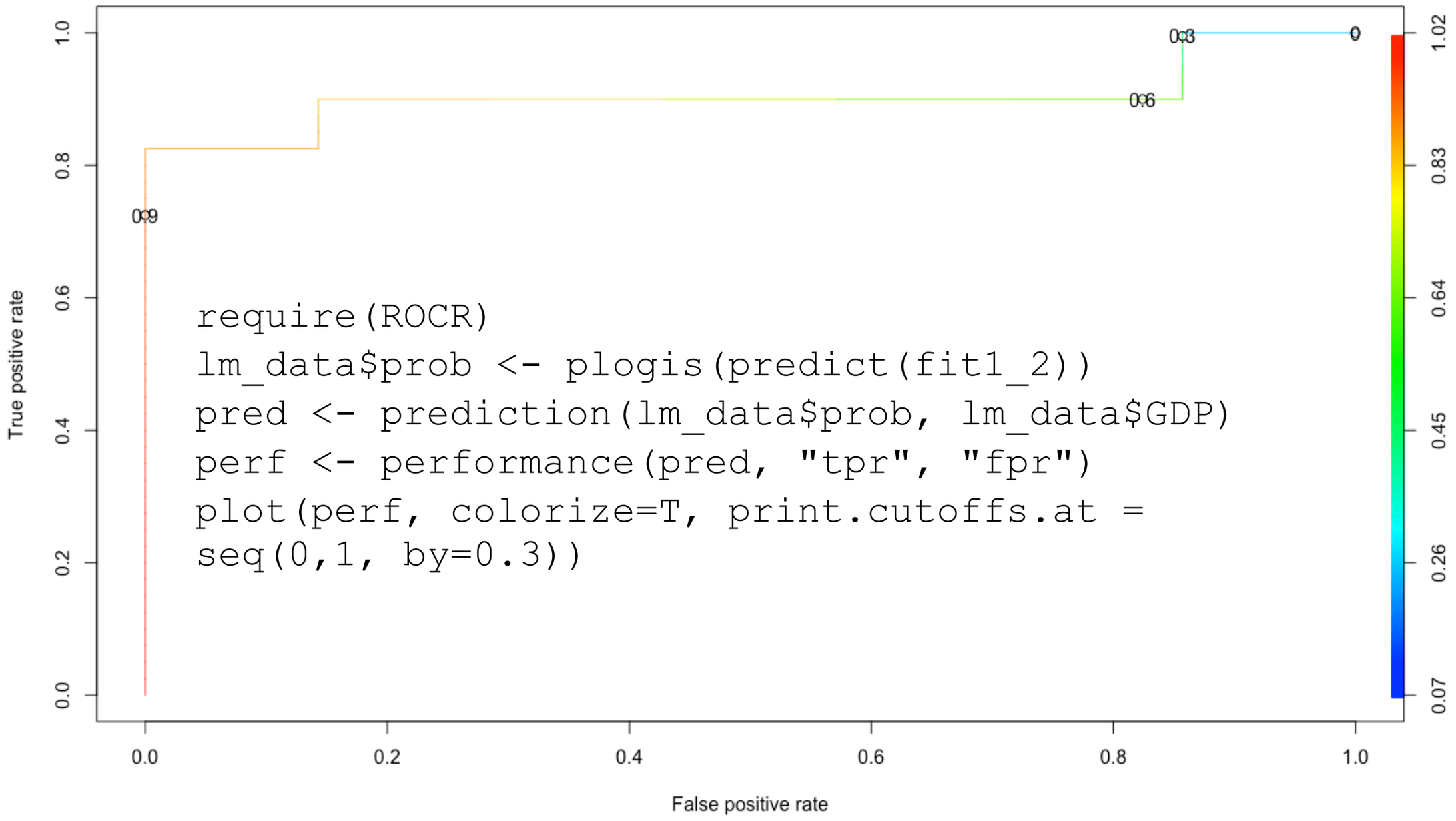
```
> b0 <-fit1_2$coefficients[1]
> b1 <-fit1_2$coefficients[2]
> b2 <-fit1_2$coefficients[3]
>
> new <- data.frame(EXPENCE = c(19, 23, 26), CPI = c(-0.4, 5, 15 ))
> a <- exp(b0 + b1*new[1] + b2*new[2])
> a / (1+a)
          EXPENCE
1 0.9984956900474
2 0.6412480803861
3 0.0007078232385
>
> predict(fit1_2, newdata = new, type = "response")
               1                2                3
0.9984956900474 0.6412480803861 0.0007078232385
```

plogis(predict(fit1_2, newdata = new))

Оцінка граничних ефектів

```
> # install.packages("erer") оцінка граничних ефектів
> library("erer")
> maBina(fit1_2)
            effect error t.value p.value
(Intercept)  1.392 0.571   2.436   0.019
EXPENCE     -0.050 0.023  -2.197   0.033
CPI         -0.040 0.018  -2.289   0.027
Warning message:
In f.xb * coef(w) :
  Recycling array of length 1 in array-vector arithmetic is deprecated.
  Use c() or as.vector() instead.

> maBina(fit1_2, x.mean = F)
            effect error t.value p.value
(Intercept)  1.728 0.709   2.436   0.019
EXPENCE     -0.062 0.028  -2.197   0.033
CPI         -0.050 0.022  -2.289   0.027
```

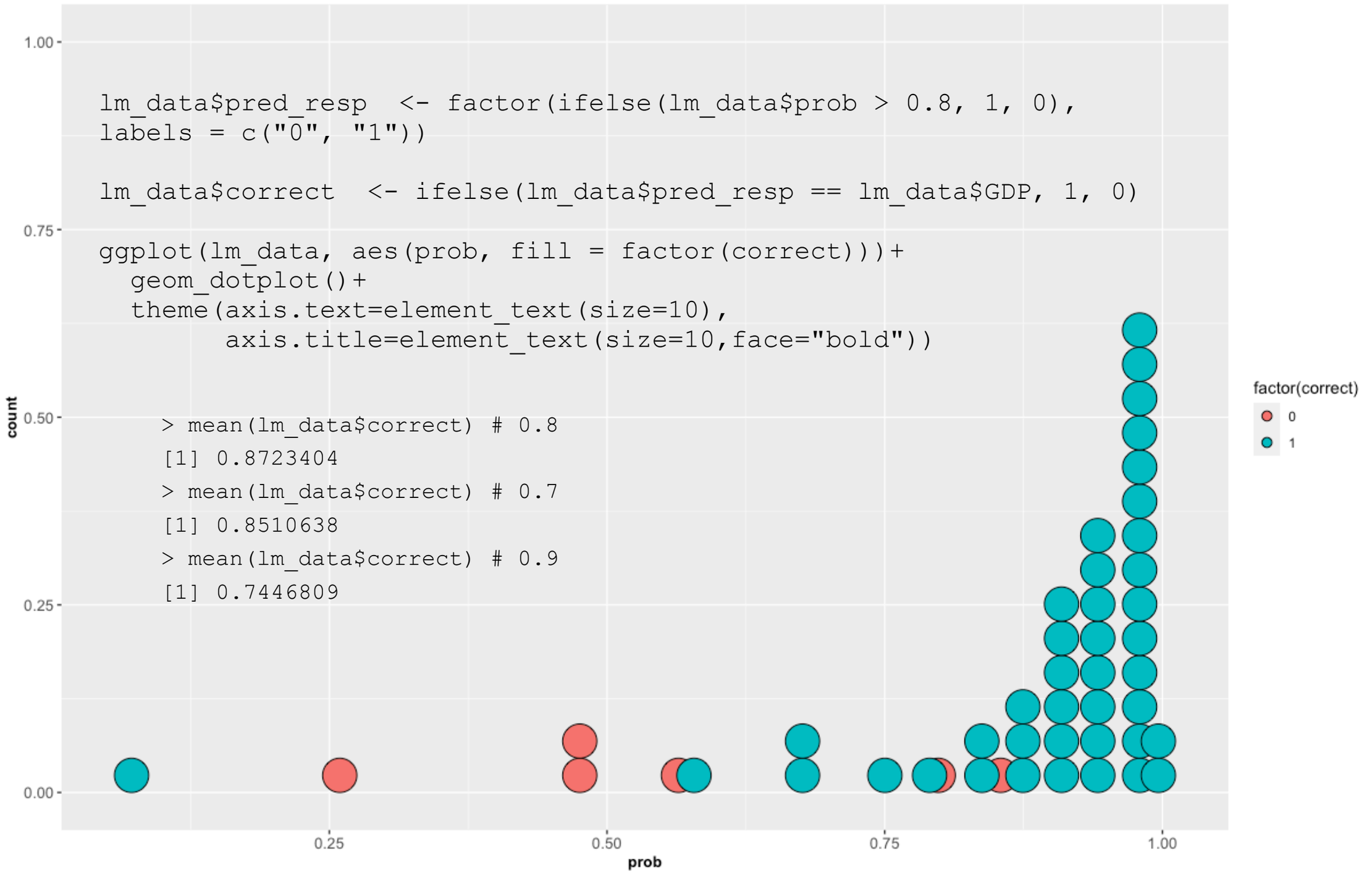
ROC крива


1.0
0.8
0.6
0.4
0.2
0.0
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
1.02
0.83
0.64
0.45
0.26
0.07
0.9
0.6
0.3
0
require(ROCR)
lm_data$prob <- plogis(predict(fit1_2))
pred <- prediction(lm_data$prob, lm_data$GDP)
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize=T, print.cutoffs.at =
seq(0,1, by=0.3))

Оцінка ефективності прогнозів

```
lm_data$pred_resp  <- factor(ifelse(lm_data$prob > 0.8, 1, 0),
labels = c("0", "1"))

lm_data$correct  <- ifelse(lm_data$pred_resp == lm_data$GDP, 1, 0)

ggplot(lm_data, aes(prob, fill = factor(correct)))+
  geom_dotplot()+
  theme(axis.text=element_text(size=10),
        axis.title=element_text(size=10,face="bold"))
```

```
> mean(lm_data$correct) # 0.8
[1] 0.8723404
> mean(lm_data$correct) # 0.7
[1] 0.8510638
> mean(lm_data$correct) # 0.9
[1] 0.7446809
```



Кластерний аналіз

- Кластерний аналіз (англ. Data clustering) — задача розбиття заданої вибірки об'єктів (ситуацій) на підмножини, які називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися.

- Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу завдань навчання без вчителя.

Базові алгоритми

- Група алгоритмів 1. k-mean алгоритм або підхід K-середніх. Для цієї групи алгоритмів характерним є визначення векторів середніх значень показників, що інтерпретується як «центр ваги» групи.

- Група алгоритмів 2. Ієрархічні алгоритми мають відношення до теоретико-графові підходу. Результати кластеризації представляються у вигляді дерева – дендрограмми, як наочного засобу аналізу.

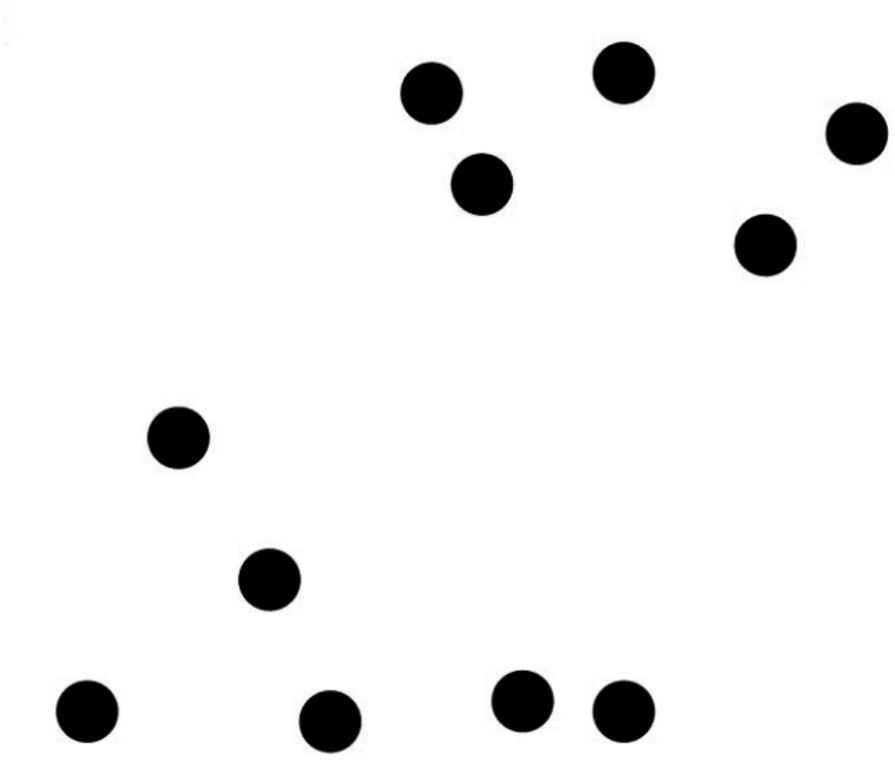
    - агломеративні (об’єднувальні, поетапно об'єднують найближчі групи або об'єкти)

    - розділювальні (поетапний поділ вихідної групи (одного великого кластреру) на найбільш віддалені підгрупи).

- Група алгоритмів 4, що заснована на теорії графів. До цієї групи алгоритмів належить поширений алгоритм найкоротшого незамкнутого шляху. При реалізації алгоритмів цієї групи будується мінімальна структура графів. У цьому графі вершини відповідають об'єктам (елементам, що кластеризуються), а ребра мають довжину, що дорівнює оцінкам відстані між відповідними елементами.

- Група алгоритмів 5, що засновані на методах нечіткої логіки під час проведення кластерного аналізу. При використанні даного підходу передбачається, що кожен кластер є нечіткою множиною вхідних об’єктів.

- Група алгоритмів 6, які засновані на використанні штучних нейронних мереж.



Кластеризація з R: *Метод k-means*.

Метод k-means.
Базова функція для виконання кластеризації зазначений методом є функція

```
kmeans(x, centers, iter.max = 10, nstart = 1,
       algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
"MacQueen"), trace=FALSE)
```

Значення ключових аргументів функції *kmeans()* наступні:

x - числова матриця даних, data.frame або інший об'єкт, який може бути визначений як вхідні данні;

centers - визначає кількість кластерів або набір початкових центрів кластерів. Якщо представлено числом кластерів, то в якості початкових центрів вибирається випадкова множина рядків з множини *x*;

iter.max – максимальна кількість дозволених ітерацій;

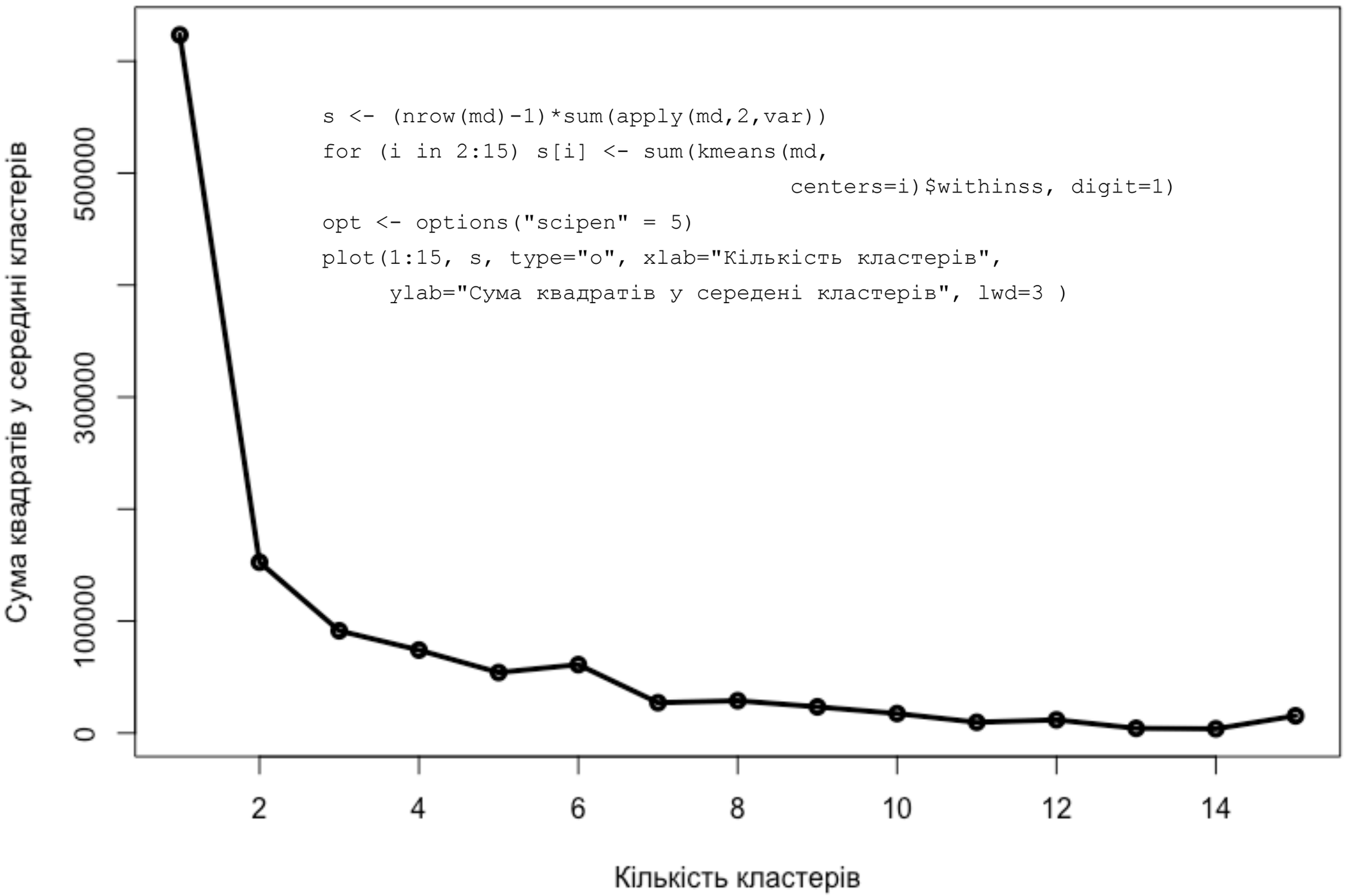
nstart – визначає скільки випадкових множин вибирається, якщо кількість кластерів задано певним числом;

algorithm – алгоритм за яким здійснюється кластеризація;

trace – логічне або ціле число, що використовується тільки в методі "Hartigan-Wong". Якщо позитивне (або TRUE), виводиться додаткова інформація про хід роботи алгоритму. Більш високі значення позитивного числа визначає більшу деталізації інформації щодо виконання алгоритму.

Графік залежності суми квадратів у середині кластерів від кількості кластерів

```
s <- (nrow(md)-1)*sum(apply(md,2,var))
for (i in 2:15) s[i] <- sum(kmeans(md,
                                   centers=i)$withinss, digit=1)
opt <- options("scipen" = 5)
plot(1:15, s, type="o", xlab="Кількість кластерів",
     ylab="Сума квадратів у середені кластерів", lwd=3 )
```



Приклад *k-means* аналізу з R

```
fit <- kmeans(md, 5) # 5 cluster solution
> # get cluster means
> aggregate(md,by=list(fit$cluster),FUN=mean)
  Group.1         mpg       disp       drat           wt       qsec
1       1 -0.83698600  0.9887119 -0.8294431  0.86437076 -0.5046758
2       2  0.44954345 -0.7255351  0.6049193 -0.06873063  2.8267546
3       3  1.65523937 -1.1624447  1.2252295 -1.37384616  0.3075550
4       4 -0.04199887 -0.4695618  0.6236221 -0.29676814 -0.3427641
5       5  0.20895733 -0.3491393 -0.5406284 -0.13771677  1.1577033

> # append cluster assignment

> md <- data.frame(md, cluster = fit$cluster)
> head(md)

                     mpg       disp       drat           wt       qsec cluster
Mazda RX4          0.1508848 -0.57061982  0.5675137 -0.610399567 -0.7771651       4
Mazda RX4 Wag      0.1508848 -0.57061982  0.5675137 -0.349785269 -0.4637808       4
Datsun 710         0.4495434 -0.99018209  0.4739996 -0.917004624  0.4260068       4
Hornet 4 Drive     0.2172534  0.22009369 -0.9661175 -0.002299538  0.8904872       5
Hornet Sportabout -0.2307345  1.04308123 -0.8351978  0.227654255 -0.4637808       1
Valiant           -0.3302874 -0.04616698 -1.5646078  0.248094592  1.3269868       5
```

Ієрархічна кластеризація з R

```
hclust(d, method = "complete", members = NULL)
```

d – матриця відмінності, що створюється за допомогою функції *dist()*;

method – метод ієрархічної кластерної агломерації, що застосовується у аналізі даних. Доступні наступні скорочені назви методів: "ward.D", "ward.D2", "single", "complete", "average" (UPGMA), "mcquitty" (WPGMA), "median" (WPGMC) або "centroid" (UPGMC);

members – NULL або вектор з розміром довжини d. Якщо значення аргументу відмінного від NULL, аргумент *d* вважається матрицею несхожості між кластерами, а не різницею між однаковими і членами, дає кількість спостережень на кластер.

Для проведення ієрархічної кластеризації використовується функція dist(). Ця функція обчислює значення і повертає матрицю розрахованих відстаней між даними, що обчислюються на базі міри відстаней між рядками матриці даних.

```
dist(x, method = "euclidean", diag = FALSE,
     upper = FALSE, p = 2)
```

x – числова матриця, data.frame або обєкт функції *dist()*;

method – визначає методи розрахунку відстаней. Доступні значення аргументи: "euclidean", "maximum", "manhattan", "canberra", "binary"або "minkowski";

p - сила відстані Мінькoвського (Minkowski);

логічні змінні (TRUE або FALSE):

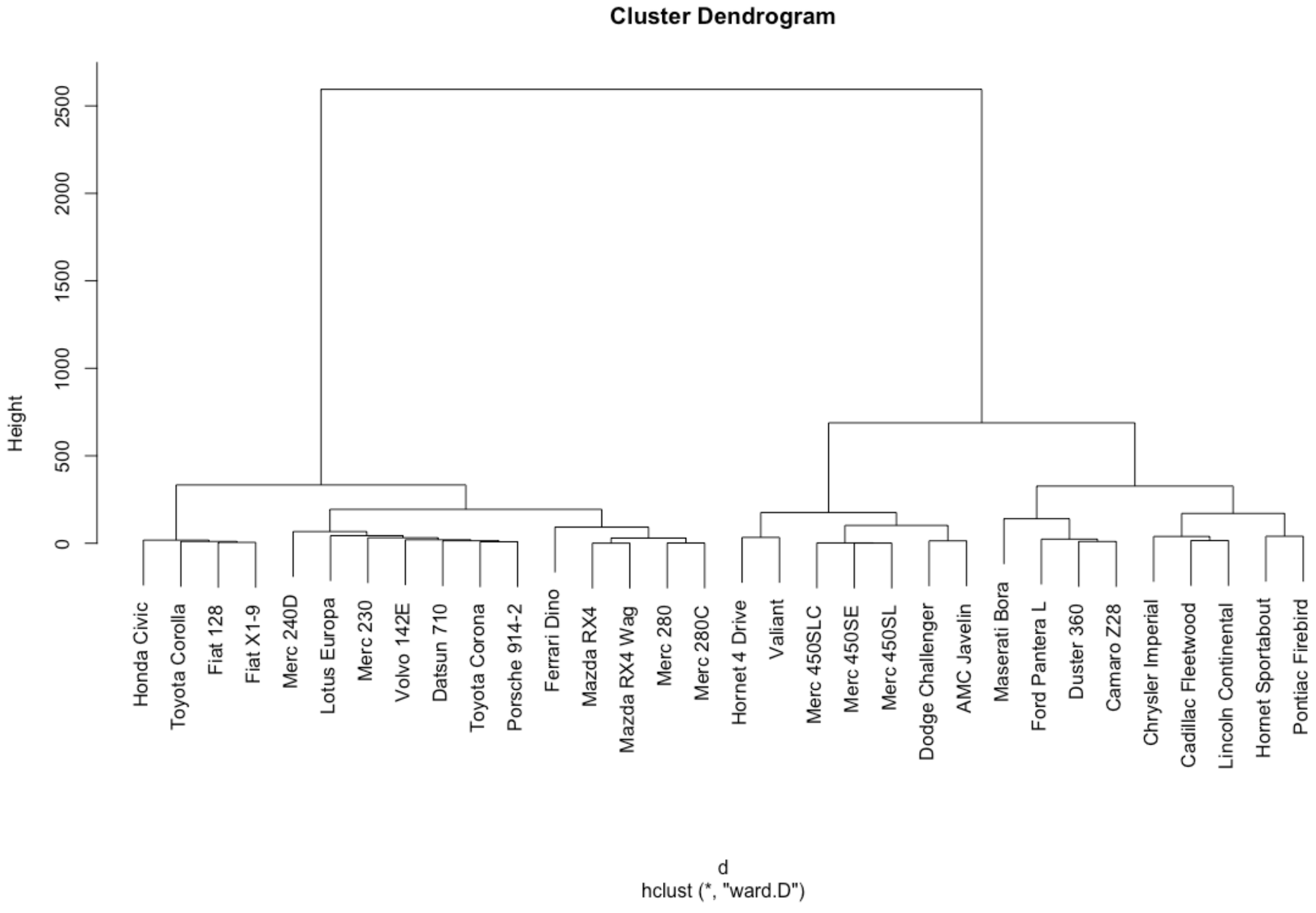
diag - вказує, чи повинна виводитись діагональ матриці відстаней функцією *print.dist()*. Значення діагоналі дорівнює нулю;

upper - вказує, чи повинен виводитись верхній трикутник матриці відстані *print.dist*().

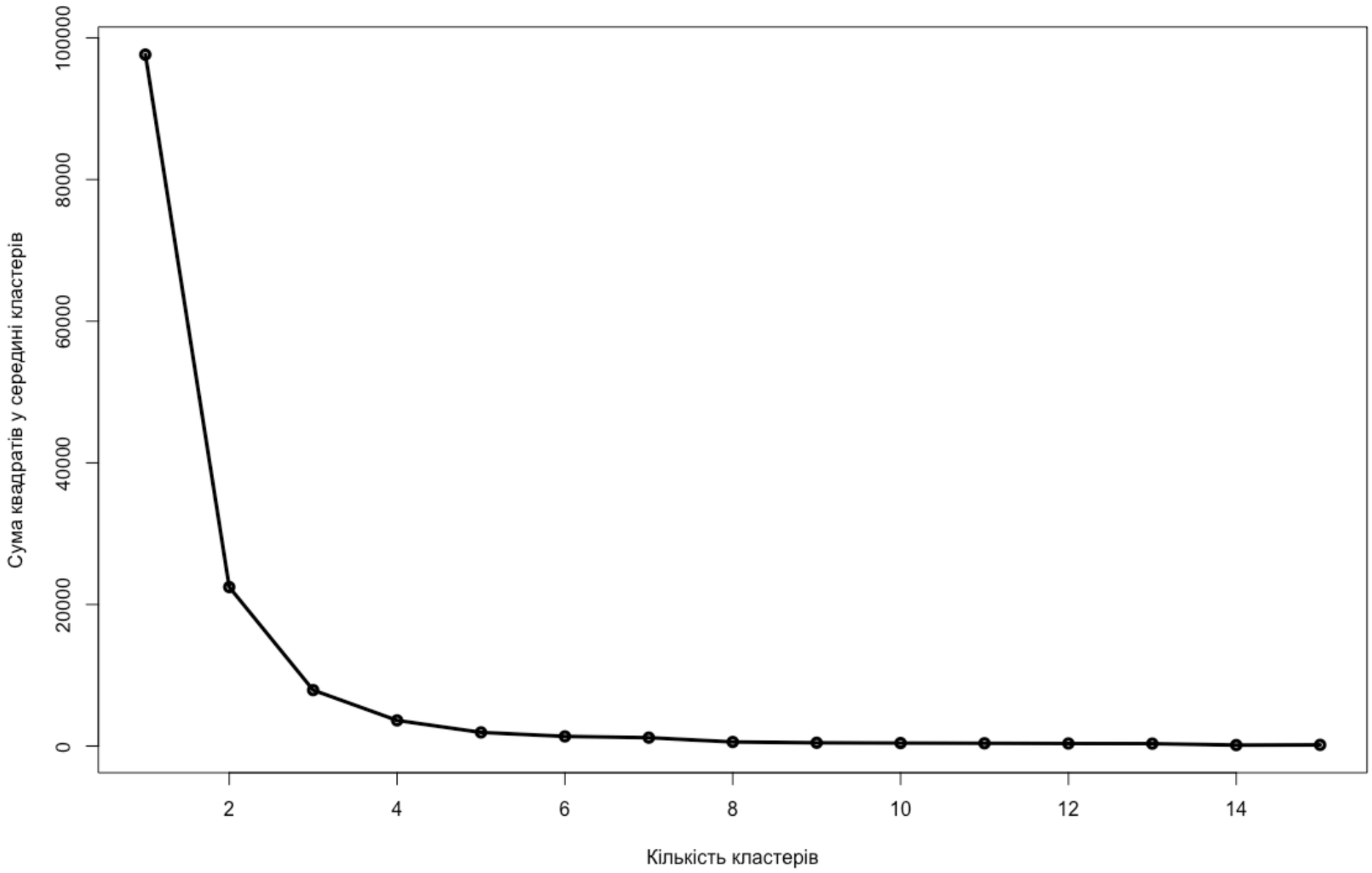
Приклад ієрархічної кластеризації з R

```
# Ward Hierarchical Clustering

d <- dist(md, method = "euclidean") #
distance matrix

fit <- hclust(d, method="ward.D")

plot(fit) # display dendogram

groups <- cutree(fit, k=5) # cut tree into 5
clusters

# draw dendogram with red borders around the
5 clusters

rect.hclust(fit, k=5, border="red")
```

Дендрограма



Графік залежності суми квадратів дисперсній у середині кластерів від кількості кластерів



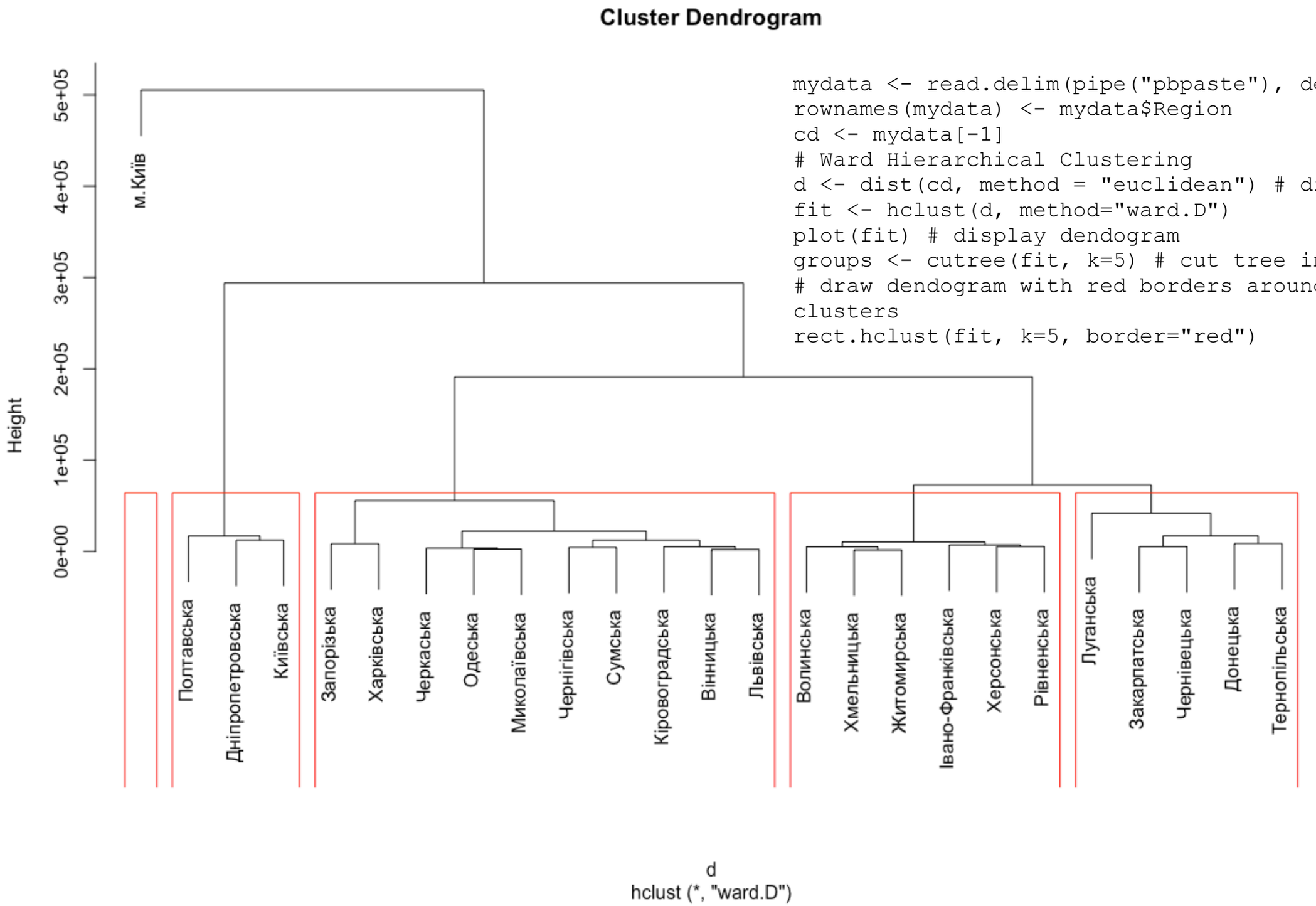
k-means «Валовий регіональний продукт у розрахунку на одну особу 2014-2017 рр.»

```
# Determine number of clusters for Regional
Clustering
s <- (nrow(cd)-1)*sum(apply(cd,2,var))
for (i in 2:15) s[i] <- sum(kmeans(cd,
                                   centers=i)
$withinss, digit=1)
plot(1:15, s, type="o", xlab="Кількість
кластерів",
     ylab="Сума квадратів у середині кластерів",
lwd=3 )

fit <- kmeans(cd, 5) # 5 cluster solution
# get cluster means
aggregate(cd,by=list(fit$cluster),FUN=mean)
summary(fit)
# append cluster assignment
cd <- data.frame(cd, cluster = fit$cluster)

fit <- kmeans(cd, 4) # 5 cluster solution
# get cluster means
aggregate(cd,by=list(fit$cluster),FUN=mean)
summary(fit)
# append cluster assignment
cd <- data.frame(cd, cluster = fit$cluster)
```

Таблиця 1

	X2014	X2015	X2016	X2017	cluster	cluster.1
м.Київ	124163	155904	191736	238622	3	5
Полтавська	48040	66390	81145	106248	1	4
Дніпропетровська	53749	65897	75396	97137	1	4
Київська	46058	60109	74216	90027	1	4
Запорізька	37251	50609	59729	75306	1	1
Харківська	35328	45816	57150	69489	2	1
Одеська	31268	41682	50159	62701	2	1
Миколаївська	30357	41501	50091	60549	2	1
Черкаська	30628	40759	48025	59697	2	1
Вінницька	27249	37270	46615	58384	2	1
Львівська	28731	37338	45319	58221	2	1
Чернігівська	26530	35196	41726	55198	2	2
Кіровоградська	29223	39356	47469	55183	2	1
Сумська	26943	37170	41741	51419	2	2
Волинська	23218	30387	34310	49987	2	2
Хмельницька	24662	31660	37881	49916	2	2
Житомирська	23678	30698	38520	49737	2	2
Івано-Франківська	27232	33170	37220	46312	2	2
Херсонська	21725	30246	36585	45532	2	2
Рівненська	24762	30350	33958	42038	4	2
Донецька	27771	26864	32318	39411	4	2
Тернопільська	20228	24963	29247	38593	4	2
Закарпатська	19170	22989	25727	34202	4	3
Чернівецька	16552	20338	23365	31509	4	3
Луганська	14079	10778	14251	13883	4	3

Дендрограма «Валовий регіональний продукт у розрахунку на одну особу 2014-2017 рр.»




```
mydata <- read.delim(pipe("pbpaste"), dec=',')
rownames(mydata) <- mydata$Region
cd <- mydata[-1]
# Ward Hierarchical Clustering
d <- dist(cd, method = "euclidean") # distance matrix
fit <- hclust(d, method="ward.D")
plot(fit) # display dendogram
groups <- cutree(fit, k=5) # cut tree into 5 clusters
# draw dendogram with red borders around the 5
clusters
rect.hclust(fit, k=5, border="red")
```