Лабораторна 2: Лінійна регресія.

Бібліотеки

Функція library() використовується для підключення бібліотек та даних, які не включені в базовий набір. Ми підключаємо пакети MASS та ISLR package.

```
> library (MASS)
> library (ISLR)
```

Якщо ви отримали повідомлення про помилку під час завантаження будь-якої з цих бібліотек, це ймовірно вказує на те, що відповідна бібліотека ще не встановлена. Деякі бібліотеки, такі як MASS, поставляються з R і не потребують окремого встановлення. Однак інші пакети, такі як ISLR, потрібно завантажувати при першому використанні. Наприклад, у системі Windows виберіть Install package на вкладці Packages. Після вибору будь-якого дзеркала, з'явиться список доступних пакетів. Просто виберіть пакунок, який ви хочете встановити і R автоматично завантажить пакет. Як варіант, це можна зробити в командному рядку R через install.packages ("ISLR"). Це потрібно зробити лише під час першого використання пакета. Однак, функцію library () потрібно викликати кожного разу, коли ви хочете використовувати даний пакет.

Проста лінійна регресія
Бібліотека MASS містить дані Boston, що включає medv (медіану вартості будинків) для 506 районів навколо Бостона. Ми хочемо передбачити medv використовуючи 13 предикторів серед яких rm (середня кількість кімнат в будинку), age (середній вік будинків), lstat (відсоток домогосподарств з низьким рівнем соціально-економічного статусу).

```
> fix(Boston )
> names(Boston )
```

```
[1] "crim" "zn" "indus" "chas" "nox" "rm" "age"
[8] "dis" "rad" "tax" "ptratio " "black" "lstat" "medv"
```

Можемо використати команду ?Boston.

Ми використаємо функцію lm(), щоб оцінити просту лінійну регресію
з medv як залежна змінна і lstat – незалежна. Базовий синтаксис: lm(y ~x,data), де y залежна змінна, x - предиктор, data множина, де зберігаються значення обох змінних.

```
> lm.fit =lm(medv ~lstat)
Error in eval(expr , envir , enclos ) : Object "medv" not found
```

Помилка виникла, оскільки невідомо місцезнаходження medv and lstat. Чуть модифікувавши попередню команду отримаємо:

```
> lm.fit =lm(medv ~lstat ,data=Boston )
> attach (Boston )
> lm.fit =lm(medv ~lstat)
```

Ввівши lm.fit, отримаємо базову інформацію про модель. Для більш детального опису потрібно використати summary(lm.fit).

```
> lm.fit
Call:
lm(formula = medv ~ lstat)
Coefficients:
(Intercept)          lstat
34.55                -0.95
```

```
> summary(lm.fit)

Call:
lm(formula = medv ~ lstat)

Residuals:
    Min      1Q  Median      3Q     Max
-15.17   -3.99   -1.32    2.03   24.50

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  34.5538     0.5626    61.4   <2e-16 ***
lstat        -0.9500     0.0387   -24.5   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 6.22 on 504 degrees of freedom
Multiple R-squared: 0.544,      Adjusted R-squared: 0.543
F-statistic:  602 on 1 and 504 DF,  p-value: <2e-16
```

Функція names() дозволяє отримати додаткову інформацію про частини lm.fit. Їх ми можемо отримати з допомогою їхніх імен, наприклад, lm.fit$coefficients. Інший спосіб це зробити, використати функцію coef().

```
> names(lm.fit)
 [1] "coefficients"  "residuals"     "effects"
 [4] "rank"          "fitted.values" "assign"
 [7] "qr"            "df.residual"   "xlevels"
[10] "call"          "terms"         "model"
> coef(lm.fit)
(Intercept)       lstat
      34.55       -0.95
```

Для отримання даних про інтервали довіри, використовуємо confint().

```
> confint(lm.fit)
              2.5 % 97.5 %
(Intercept) 33.45 35.659
lstat       -1.03 -0.874
```

Функція predict() повертає інтервали довіри та передбачення для medv за заданого рівня lstat.

```
> predict(lm.fit,data.frame(lstat=(c(5,10,15))),
        interval="confidence")
    fit   lwr   upr
1 29.80 29.01 30.60
2 25.05 24.47 25.63
3 20.30 19.73 20.87
> predict(lm.fit,data.frame(lstat=(c(5,10,15))),
        interval="prediction")
    fit    lwr   upr
1 29.80 17.566 42.04
2 25.05 12.828 37.28
3 20.30  8.078 32.53
```

Зобразимо наші дані на графікуразом з прямою з допомогою функцій plot() та abline().

```
> plot(lstat ,medv)
> abline (lm.fit)
```

Бачимо наявність нелінійного зв'язку між розглянутими змінними lstat та medv. Дослідимо це пізніше.
Функція abline() дозволяє намалювати довільну пряму, не лише нашу оцінку. Задавши відповідні параметри a та b, ми вводимо abline(a,b). Команда lwd=3 збільшує товщину прямої на 3 (працює також для plot() та lines()). З допомогою опції pch можемо змінювати графічні символи.

```
> abline (lm.fit ,lwd =3)
> abline (lm.fit ,lwd =3, col ="red ")
> plot(lstat ,medv ,col ="red ")
```

```
> plot(lstat ,medv ,pch =20)
> plot(lstat ,medv ,pch ="+")
> plot (1:20 ,1:20, pch =1:20)
```

Далі ми розглянемо деякі діагностичні графіки. Застосування автоматично виробляє чотири діагностичні графіки можна отримати застосувавши plot () безпосередньо до lm (). Загалом, ця команда буде видавати по одному графіку за раз, а натискання клавіші Enter генеруватиме наступний графік. Щоб зобразити всі графіки одночасно можна використати par(). Наприклад, par (mfrow = c (2,2)) ділить область графіків на сітку 2 × 2.

```
> par(mfrow =c(2,2))
> plot(lm.fit)
```

З іншого боку, можемо використати функцію residuals(, щоб отримати список залишків моделі. Функція rstudent() перетворить залишки до стандартизованого вигляду і ми матимемо можливість самі побудувати відповідний графік.

```
> plot(predict (lm.fit), residuals (lm.fit))
> plot(predict (lm.fit), rstudent (lm.fit))
```

Для обчислення левередж статистики використовуємо функцію hatvalues().

```
> plot(hatvalues (lm.fit ))
> which.max (hatvalues (lm.fit))
375
```

Функція which.max() повертає індекс максимального елемента вектора, тобто спостереження з максимальним значенням левередж статистики.

Множинна лінійна регресія (Багатовимірна лінійна регресія)

Для того, щоб оцінити модель множинної лінійної регресії з використанням найменших квадратів, ми знову використовуємо функцію lm (). Синтаксис lm (y~x1 + x2 + x3) використовується для оцінки моделі із трьома предикторами, x1, x2 та x3. Функція summary() виводить коефіцієнти регресії для всіх предикторів.

```
> lm.fit=lm(medv~lstat+age,data=Boston)
> summary(lm.fit)

Call:
lm(formula = medv ~ lstat + age, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-15.98   -3.98   -1.28    1.97   23.16

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.2228     0.7308   45.46   <2e-16 ***
lstat        -1.0321     0.0482  -21.42   <2e-16 ***
age           0.0345     0.0122    2.83   0.0049 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1  1

Residual standard error: 6.17 on 503 degrees of freedom
Multiple R-squared: 0.551,      Adjusted R-squared: 0.549
F-statistic:  309 on 2 and 503 DF,  p-value: <2e-16
```

Дані Boston містять 13 змінних і трохи не зручно вводити їх всі, тому є можливість використати скорочену форму:

```
> lm.fit =lm(medv ~.,data=Boston )
> summary (lm.fit)
Call:
lm(formula = medv ~ ., data = Boston )
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-15.594  -2.730  -0.518   1.777  26.199

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288
chas         2.687e+00  8.616e-01   3.118 0.001925 **
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.761e-03  -3.280 0.001112 **
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-Squared: 0.7406,     Adjusted R-squared: 0.7338
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Ми можемо отримати доступ до окремих компонентів зведеного об’єкта за назвою (введіть ?summary.lm, щоб побачити, що доступно). Звідси summary(lm.fit)$r.sq дає нам R2, а summary(lm.fit)$sigma - RSE. Vif () функція, яка є частиною пакету car, може використовуватися для обчислення vif. Більшість значень VIFів є низькими та помірними для цих даних. Пакет car не є частиною базової інсталяції R, тому його потрібно завантажити вперше окремо.

```
> library(car)
> vif(lm.fit)
    crim       zn    indus     chas      nox       rm      age
    1.79     2.30     3.99     1.07     4.39     1.93     3.10
     dis      rad      tax  ptratio    black    lstat
    3.96     7.48     9.01     1.80     1.35     2.94
```

Якщо хочемо побудувати регресію відносно всіх змінних крім якоїсь однієї, то можемо ввести

```
> lm.fit1=lm(medv~.-age ,data=Boston )
> summary (lm.fit1)
...
```

Також можна використати функцію update()

> lm.fit1=update (lm.fit , ~.-age)

Змінна взаємодії

З допомогою функції lm() легко включити до нашої моделі змінну взаємодії. Синтаксис lstat:black включає змінну взаємодії між lstat і black. Синтаксис lstat*age одночасно включає lstat, age та змінну lstat×age, по суті це короткий запис для lstat+age+lstat:age.

```
> summary(lm(medv~lstat*age,data=Boston))

Call:
lm(formula = medv ~ lstat * age, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-15.81   -4.04   -1.33    2.08   27.55

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 36.088536   1.469835   24.55  < 2e-16 ***
lstat       -1.392117   0.167456   -8.31  8.8e-16 ***
age         -0.000721   0.019879   -0.04    0.971
lstat:age    0.004156   0.001852    2.24    0.025 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.15 on 502 degrees of freedom
Multiple R-squared: 0.556,  Adjusted R-squared: 0.553
F-statistic:  209 on 3 and 502 DF,  p-value: <2e-16
```

Нелінійне перетворення предикторів

Функція lm() дозволяє використовувати нелінійні перетворення незалежних змінних. Наприклад, нехай маючи змінну X, ми хочемо використаи X^2 як предиктор, для цього використовуємо I(X^2).

```
> lm.fit2=lm(medv~lstat+I(lstat^2))
> summary(lm.fit2)

Call:
lm(formula = medv ~ lstat + I(lstat^2))

Residuals:
    Min      1Q  Median      3Q     Max
-15.28   -3.83   -0.53    2.31   25.41
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 42.86201    0.87208    49.1   <2e-16 ***
lstat       -2.33282    0.12380   -18.8   <2e-16 ***
I(lstat^2)   0.04355    0.00375    11.6   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.52 on 503 degrees of freedom
Multiple R-squared: 0.641,  Adjusted R-squared: 0.639
F-statistic:  449 on 2 and 503 DF,  p-value: <2e-16
```

Використовуючи функцію anova() порівняємо обидві моделі.

```
> lm.fit=lm(medv~lstat)
> anova(lm.fit,lm.fit2)
Analysis of Variance Table

Model 1: medv ~ lstat
Model 2: medv ~ lstat + I(lstat^2)
  Res.Df   RSS Df Sum of Sq   F Pr(>F)
1    504 19472
2    503 15347  1      4125 135 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Функція anova() порівнює дві моделі на основі статистичного тесту. Нульовою гіпотезою є те, що обидві моделі однаково добре описують дані, а альтернативною – що повна модель краща. В нашому випадку модель з нелінійним перетворенням є очевидно краща. Повернемося до графічного аналізу

```
> par(mfrow=c(2,2))
> plot(lm.fit2)
```

який підтвердить наші спостереження.

Для включення в модель вищих порядків можна використати функцію poly()

```
> lm.fit5=lm(medv~poly(lstat,5))
> summary(lm.fit5)

Call:
lm(formula = medv ~ poly(lstat, 5))

Residuals:
    Min      1Q  Median      3Q     Max
-13.543  -3.104  -0.705   2.084  27.115

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        22.533      0.232   97.20  < 2e-16 ***
poly(lstat, 5)1 -152.460      5.215  -29.24  < 2e-16 ***
poly(lstat, 5)2   64.227      5.215   12.32  < 2e-16 ***
poly(lstat, 5)3  -27.051      5.215   -5.19  3.1e-07 ***
poly(lstat, 5)4   25.452      5.215    4.88  1.4e-06 ***
poly(lstat, 5)5  -19.252      5.215   -3.69  0.00025 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.21 on 500 degrees of freedom
Multiple R-squared: 0.682,  Adjusted R-squared: 0.679
F-statistic:  214 on 5 and 500 DF,  p-value: <2e-16
```

Отже, всі предиктори включно з 5-ю степенню мають вплив на залежну змінну. Для використання логарифмічного перетворення маємо функцію log().
> summary (lm(medv ~log(rm),data=Boston))
...

Якісні предиктори
Розглянемо дані Carseats з бібліотеки ISLR. Спробуємо передбачити Sales (дитячі автокрісла) в 400 локаціях на основі певних предикторів.

```
> fix(Carseats)
> names(Carseats)
 [1] "Sales"       "CompPrice"   "Income"      "Advertising"
 [5] "Population"  "Price"       "ShelveLoc"   "Age"
 [9] "Education"   "Urban"       "US"
```

Дані Carseats містять якісну змінну Shelveloc, індикатор якості розташування стелажів – тобто простору всередині магазину, в якому виставляються автокрісла – у кожній локації. Показник Shelveloc приймає три можливі значення: Bad, Medium і Good. Розпізнавши змінну Shelveloc як якісну, R генерує фіктивні змінні автоматично. Ми також додаємо до моделі змінні взаємодії.

```
> lm.fit=lm(Sales~.+Income:Advertising+Price:Age,data=Carseats)
> summary(lm.fit)

Call:
lm(formula = Sales ~ . + Income:Advertising + Price:Age, data =
    Carseats)

Residuals:
    Min      1Q  Median      3Q     Max
-2.921  -0.750   0.018   0.675   3.341

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          6.575565   1.008747    6.52  2.2e-10 ***
CompPrice            0.092937   0.004118   22.57  < 2e-16 ***
Income               0.010894   0.002604    4.18  3.6e-05 ***
Advertising          0.070246   0.022609    3.11  0.00203 **
Population           0.000159   0.000368    0.43  0.66533
Price               -0.100806   0.007440  -13.55  < 2e-16 ***
ShelveLocGood        4.848676   0.152838   31.72  < 2e-16 ***
ShelveLocMedium      1.953262   0.125768   15.53  < 2e-16 ***
Age                 -0.057947   0.015951   -3.63  0.00032 ***
Education           -0.020852   0.019613   -1.06  0.28836
UrbanYes             0.140160   0.112402    1.25  0.21317
USYes               -0.157557   0.148923   -1.06  0.29073
Income:Advertising   0.000751   0.000278    2.70  0.00729 **
Price:Age            0.000107   0.000133    0.80  0.42381
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.01 on 386 degrees of freedom
Multiple R-squared: 0.876,      Adjusted R-squared: 0.872
F-statistic:  210 on 13 and 386 DF,  p-value: <2e-16
```

Функція contrasts() повертає кодування використане для фіктивних мінних.

> attach (Carseats)
> contrasts (ShelveLoc)
Good Medium
Bad 0 0
Good 1 0
Medium 0 1

За додатковою інформацією можна звернутися через ?contrasts.

3.6.7 Написання функцій

Нехай ми хочемо задати просту функцію, яка читає в бібліотеках ISLR і MASS і називається LoadLibraries ().

```
> LoadLibraries=function(){
+ library(ISLR)
+ library(MASS)
+ print("The libraries have been loaded.")
+ }
```

Тепер ввівши LoadLibraries ми отримаємо довідку

```
> LoadLibraries
function(){
library(ISLR)
library(MASS)
print("The libraries have been loaded.")
}
```

А ввівши LoadLibraries() запустимо цю функцію на виконання.

> LoadLibraries()
[1] "The libraries have been loaded ."