

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

ПРАКТИЧНЕ МАШИННЕ НАВЧАННЯ.

ЗАДАЧА ВІДНОВЛЕННЯ РЕГРЕСІЇ

Звіт

до лабораторної роботи №3

з курсу «Методи і засоби комп'ютерного навчання»

Виконав:

студент гр. СПКм-12

Сергієнко В.Р.

Прийняв

Пукач А.І.

Львів 2014

Мета: оволодіти навиками з побудови та відновленню регресії в системі R.

Короткі теоретичні відомості

Як побудувати регресію, використовуючи функцію *lm* в системі R?

```
lm(formula, data, subset, weights, na.action,  
method = "qr", model = TRUE, x = FALSE, y = FALSE,  
qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)
```

formula – символічний опис моделі. Деталі див. нижче.

data – фрейм даних, звідки беруться дані. Якщо не вказано (або вказано *NULL*), тоді всі дані, що зустрічаються в моделі – це вектори робочого простору. Інакше, дані – це поля зазначеного фрейму даних.

subset – вектор, що визначає підмножину даних, які беруть участь у моделі. Необов'язковий параметр.

weight – вектор ваг. Може бути або *NULL*, або числовим. В останньому випадку ваги використовуються в методі найменших квадратів. Необов'язковий параметр.

na.action – функція, яка показує, що треба зробити з моделлю, якщо в ній є *NA*.

method – метод, що використовується. Доступно тільки "qr".

model, *x*, *y*, *qr* – логічні значення. Якщо *TRUE*, тоді функція повертає відповідні компоненти моделі: знайдені коефіцієнти β_j , матрицю даних *X*, стовпець відповідей *y*, QR-розкладання.

singular.ok – якщо *FALSE*, тоді виродженість (ранг матриці *X* менше числа стовпців) призводить до помилки.

Регуляризація в системі R.

```
lm.ridge(formula, data, lambda = 0 , ...)
```

formula – символічний опис моделі пошуку;

data – база, з якої беруться дані для завантаження в формулу. Необхідно вказувати, якщо у формулі є дані з цієї бази;

lambda – параметр регуляризації (скаляр або вектор)

Індивідуальне завдання

1. Завантажити дані з файлу *reglab2.txt*. Використовуючи функцію *lm*, побудувати регресію (використати різні моделі). Вибрати найкращу модель, пояснити свій вибір.

```
data=read.table("C:\Program  
Files\LAB_YEAR_5\MZKN\Lab_02\reglab2.txt",header=TRUE)  
data
```

	y	x1	x2	x3	x4
1	3.4697200	0.233627529	8.355490e-01	0.102965349	0.457427677
2	0.7684485	0.117919942	9.054379e-02	0.258778422	0.283950966
3	2.8803737	0.091520175	7.975925e-01	0.198527723	0.699286808
...					
198	3.4712281	0.661586764	2.315702e-01	0.800237676	0.609558141
199	3.6339970	0.828216618	7.627500e-02	0.129718604	0.965083616
200	5.1173370	0.830951396	5.460322e-01	0.348425596	0.629569939

```

y=c(data[1,1]);
x1=c(data[1,2]);
x2=c(data[1,3]);
x3=c(data[1,4]);
x4=c(data[1,5]);
for (i in 2:200)
{
  y=c(y,data[i,1]);
  x1=c(x1,data[i,2]);
  x2=c(x2,data[i,3]);
  x3=c(x3,data[i,4]);
  x4=c(x4,data[i,5]);
}
fit1 <- lm(y ~ poly((x1+x2+x3+x4), 2))
fit1

```

Call:
lm(formula = y ~ poly((x1 + x2 + x3 + x4), 2))

Coefficients:

	(Intercept)	poly((x1 + x2 + x3 + x4), 2)1	poly((x1 + x2 + x3 + x4), 2)2
		3.586	14.010
1.132			

```

fit2 <- lm(y ~ (x1+x2+x3+x4) + I((x1+x2+x3+x4)^2))
fit2

```

Call:
lm(formula = y ~ (x1 + x2 + x3 + x4) + I((x1 + x2 + x3 + x4)^2))

Coefficients:

	(Intercept)	x1	
x2			
	-0.023255	4.012178	
3.026485			
	x3	x4	I((x1 + x2 + x3 + x4)^2)
	0.117034	0.108769	-
0.003676			

```

fit3 <- lm(y ~ (x1+x2+x3+x4))
fit3
Call:
lm(formula = y ~ (x1 + x2 + x3 + x4))

```

Coefficients:

(Intercept)	x1	x2	x3	x4
-0.009641	3.996991	3.011691	0.102261	0.094398

```

fit4 <- lm(y ~ (x1+x2+x3+x4)^2)
fit4

```

```
Call:
lm(formula = y ~ (x1 + x2 + x3 + x4)^2)
```

Coefficients:

(Intercept)	x1	x2	x3	x4
x1:x2	x1:x3			
-0.004213	3.978034	3.012931	0.092796	0.104854
0.036871	0.008958			
x1:x4	x2:x3	x2:x4	x3:x4	
-0.007653	-0.011085	-0.034027	0.018583	

3. Завантажити дані з файлу *alligators.txt*. Вибрати регресійну модель (можливо нелінійну), що відображає залежність ваги алігатора від його довжини.

```
data=read.table("/C:\Program
Files\LAB_YEAR_5\MZKN\Lab_02\alligators.txt",header=TRUE)
data
```

	Length	Weight	Len.3	Ln.Length.	Ln.Weight.
1	58	28	195112	4.060443	3.332205
2	61	44	226981	4.110874	3.784190
3	63	33	250047	4.143135	3.496508
4	68	39	314432	4.219508	3.663562
5	69	36	328509	4.234107	3.583519
6	72	38	373248	4.276666	3.637586
7	72	61	373248	4.276666	4.110874
8	74	54	405224	4.304065	3.988984
9	74	51	405224	4.304065	3.931826
10	76	42	438976	4.330733	3.737670
11	78	57	474552	4.356709	4.043051
12	82	80	551368	4.406719	4.382027
13	85	84	614125	4.442651	4.430817
14	86	83	636056	4.454347	4.418841
15	86	80	636056	4.454347	4.382027
16	86	90	636056	4.454347	4.499810
17	88	70	681472	4.477337	4.248495
18	89	84	704969	4.488636	4.430817
19	90	106	729000	4.499810	4.663439
20	90	102	729000	4.499810	4.624973
21	94	110	830584	4.543295	4.700480
22	94	130	830584	4.543295	4.867534
23	114	197	1481544	4.736198	5.283204
24	128	366	2097152	4.852030	5.902633
25	147	640	3176523	4.990433	6.461468

```
x=c(data[1,1]);
y=c(data[1,2]);
for (i in 2:25)
{
```

```

    x=c(x,data[i,1]);
    y=c(y,data[i,2]);
  }
fit1<-lm(y~x+I(x^2))
fit1
Call:
lm(formula = y ~ x + I(x^2))

Coefficients:
(Intercept)          x          I(x^2)
   410.48412   -11.31755    0.08662

> r=c(fit1[[1]])
> y1=r[1]+r[2]*x+r[3]*x^2
> xy=data.frame(x=x,y=y)
> fit2<-nls(y~b*x^3,start=list(b=1))
> fit2
Nonlinear regression model
  model: y ~ b * x^3
 data: parent.frame()
      b
0.0001703
residual sum-of-squares: 24310

Number of iterations to convergence: 1
Achieved convergence tolerance: 2.611e-09

> y2=0.0001703*(x^3)
> fit3=nls(y~a/x^-2,start=list(a=1))
> fit3
Nonlinear regression model
  model: y ~ a/x^-2
 data: parent.frame()
      a
0.0178
residual sum-of-squares: 117801

Number of iterations to convergence: 1
Achieved convergence tolerance: 1.565e-09
y3=0.0178/(x^-2)
plot(xy, type = "n", main = "Line's Regression")
lines(x, y1, "o")
lines(x, y2, "b")
lines(x, y, "p")
lines(x, y3, "l")

```

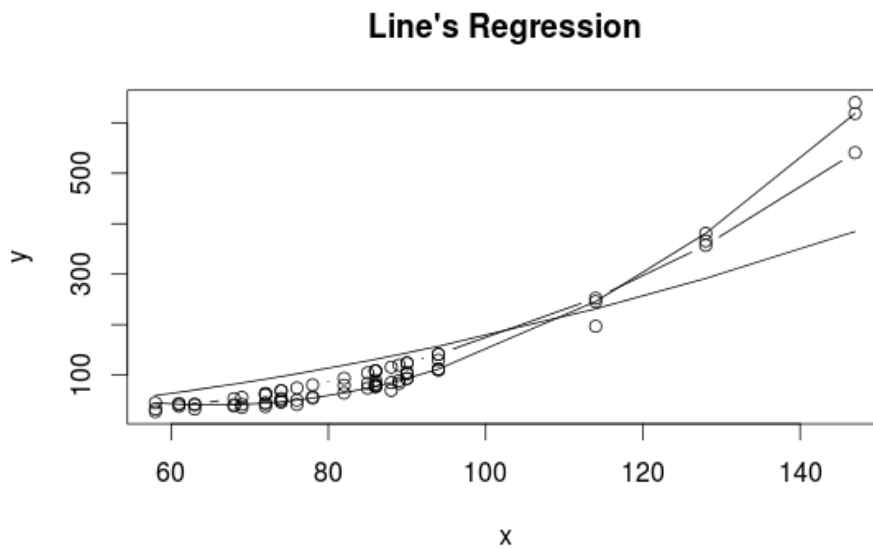


Рис1. Регресійна залежність між вагою і довжиною алігатора

Висновок

Виконуючи лабораторну роботу, визначено, що найкращу регресійну залежність між вагою і довжиною алігатора показує модель $lm(y \sim x + I(x^2))$.