Mathematical Programing

Additional Homework 2

2014110374 정상만

HW4_14_3)

스크립트

```
D=[29.65 28.55 28.65 30.15 29.35 29.75
29.25 30.65 28.15 29.85 29.05 30.25 30.85
28.75 29.65 30.45 29.15 30.45 33.65 29.35
29.75 31.25 29.45 30.15 29.65 30.55 29.65
29.25];
% a) mean (평균)
m=mean(D);
% b) median (중앙값)
me=median(D);
% c) mode (최빈값)
frq=mode(D);
% d) range (범위)
ran=range(D);
% e) standard deviation (표준편차)
s=std(D);
% f) variance (분산)
var=s^2;
% g) coefficient of variation
cv=s/m;
% h) histogram, range : 28 to 34 and
increments : 0.4
inc=28:0.4:34;
hist(D);
% i) 68% of the readings
a=m-s; b=m+s;
I=D(D>a & D<b);
p=length(I)/length(D);
% the list of the results of a)~i)
A=[m me frq ran s var cv p]';
```

실행값

i) 를 제외한 나머지 계산 결과는 생략한다.

p =0.7857

(이는 68% 와 약 0.10% 정도의 차이를 갖는다.)

설명

Matlab 내장함수를 이용하는 것이 공식을 만드는 것 보다 더 효율적이므로, 이를 이용하여 계산하는 것이 정신건강에 유익하다. 따라서 설명란은 해당하는 공식을 나열하는 것으로 만족한다.

a) 표본평균

$$\overline{x}=\frac{\text{모든 관측값의 합계}}{\text{총 자료의 개수}}=\frac{x_1+x_2+\cdots+x_n}{n}$$

b) 중앙값 (Median) : 관측값을 크기 순서로 배열하면

(i) 자료의 개수 n 이 홀수 : $\frac{(n+1)}{2}$ 번째 관측값

(ii) 자료의 개수 n 이 짝수 : $\frac{n}{2}$ 번째 관측값과 $\frac{n}{2}+1$ 번째 관측값의 평균

c) 최빈값 (Mode)

관측값 중에서 가장 자주 나오는 값.

d) 범위 (Range)

(관측값 중에서 최댓값) - (관측값 중에서 최소값)

e) 표준편차 (Standard Deviation) : $s=+\sqrt{s^2}$

s^2 는 분산이다.

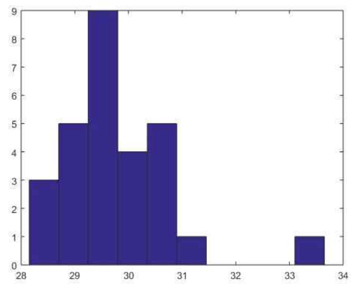
f) 분산 (Variance)

$$s^2=\frac{\text{편차의 제곱합}}{n-1}=\frac{1}{n-1}\sum(x_i-\overline{x})^2$$

g) 변동계수 (Coefficient of Variation)

$$CV=\frac{\text{표준편차}}{\text{표본평균}}\times 100$$

h)



HW4_14_14)

스크립트

```
clear all
clc
% bacteria K and oxygen concentration C
c=[0.5 0.8 1.5 2.5 4];
k=[1.1 2.5 5.3 7.6 8.9];
% slope, intercept
x=1./c.^2; y=1./k; n=5;
bar_x=mean(x); bar_y=mean(y);
a1=(n*sum(x.*y)-sum(x)*sum(y))/(n*sum(x.^2
)-(sum(x))^2);
a0=bar_y-a1*bar_x;
k_max=1/a0;
Cs=a1*k_max;
% Linear regression
y_hat=@(t) k_max*((t.^2)./(Cs+t.^2));
ct=0:0.1:5;
scatter(c,k,'MarkerFaceColor','r','MarkerE
dgeColor','r');
hold on
plot(ct,y_hat(ct),'k');
title('Linear regression of use the
linearized equation')
xlabel('c'); ylabel('k');
```

실행값

```
>> y_hat(2)

ans = 6.7950
```

설명

주어진 방정식은 다음과 같다.

$$k=\frac{k_{\max}c^2}{c_s+c^2}\ ,\ c_s,k_{\max}\ :\ \text{parameter.}$$

이 방정식을 선형화 하면 다음 방정식을 얻는다.

$$\frac{1}{k}=\frac{1}{k_{\max}}+\frac{c_s}{k_{\max}}\left(\frac{1}{c^2}\right)$$

이 때, $x=\frac{1}{k}, y=\frac{1}{c^2}$ 로 두고, 절편과 기울기는 각각 $a_0=\frac{1}{k_{\max}}, a_1=\frac{c_s}{k_{\max}}$ 로 두자. 최소제곱법을 이용한 각각의 절편과 기울기는 다음의 공식으로 구한다.

$$a_0=\bar{y}-a_1\bar{x},\ \ \bar{x},\bar{y}\ \text{는 관측값의 평균}$$

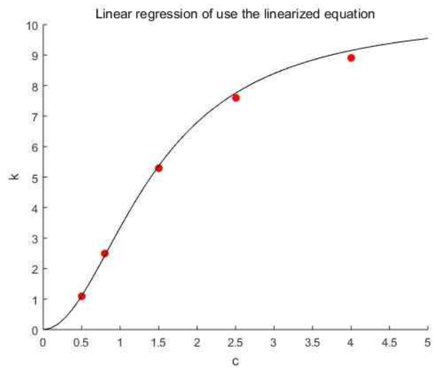
$$a_1=\frac{n\Sigma x_iy_i-\Sigma x_i\Sigma y_i}{n\Sigma x_i^2-(\Sigma x_i)^2},\ n\text{은 관측값의 개수}$$

이를 이용하여 최종적으로 각 parameter를 구하면

$$k_{\max}=10.3449,\ \ c_s=2.0897$$

따라서 최종적으로 선형회귀를 이용한 방정식은

$$\therefore k=(10.3449)\frac{c^2}{(2.0897)+c^2}$$



HW4_14_35)

스크립트

```
clear all
clc
% initial value
v0=25;
y0=1;
theta0=(50*pi)/180;
g=9.81;
% Monte Carlo equation
fn=@(t)
tan(theta0)*t-((g*t.^2)/(2*(v0^2)*cos(thet
a0)^2))+y0;
% a) the maximum height (analytically)
max_x1=((v0^2)*(tan(theta0)*cos(theta0)^2)
)/g;
% b) the maximum height (numerically)
x=linspace(0,60,10000);
y=fn(x);
max(y);
plot(x,y,'k');
for i=1:length(x);
 max_x2=x(i);
end
max_x2=x(find(fn(x)==max(y)));
hold on
plot(max_x2,fn(max_x2),'o','MarkerFaceColo
r','r','MarkerEdgeColor','r');
title('The maximum value of Monte Carlo
equation from 0m to 60m');
xlabel('x-axis'); ylabel('y-axis');
```

실행값

계산 결과는 생략한다. (설명란에 기입)

설명

a) 몬테 카를로 방정식을 함수로 생각하자. 즉

$$y=(\tan\theta_0)x-\frac{g}{2v_0^2\cos^2\theta_0}x^2+y_0$$

이 때, 초깃값은 $g=9.81$, $\theta_0=50$, $v_0=25$, $y_0=1$.
높이 y가 최대가 되는 x를 구하기 위해, y를 미분하면

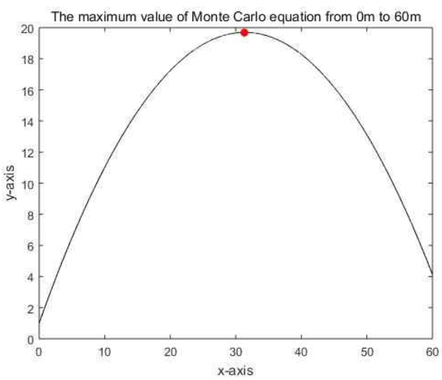
$$y'=\tan\theta_0-\frac{g}{v_0^2\cos^2\theta_0}x=0 \Rightarrow \therefore x=\frac{1}{g}(v_0^2\cos^2\theta_0)$$

$$y''=\frac{g}{v_0^2\cos^2\theta_0}>0,$$ 이계도함수 판정에 의해 그래프는 볼록한 형태를 가지므로,

$$\therefore y \text{ has the maximum at } x=\frac{1}{g}(v_0^2\cos^2\theta_0)$$

Matlab을 이용하여 x를 계산하면 $x=31.3713$,
따라서 최대 높이는 $y(31.3173)=19.6934$ 이다.

b) 수치적으로 주어진 높이에 대한 방정식이 최대가 되게 하는 x를 문제에서 제시한 조건에 따라 구한다.
거리 x의 길이를 스크립트와 같이 구간 $(0,60)$을 10000등분으로 하는 벡터를 생성하고, 이를 이용하여 주어진 높이에 대한 벡터를 구한다.
이제 스크립트를 참조하면, 구하고자 하는 값이 $x=31.3711$ 임을 얻는다. 이 때 $y=19.6934$ 이다.



HW4_15_4)

<table>
<tr><th>스크립트</th><th>설명</th></tr>
<tr><td>

```
function p=HW4_15_4(x,y,m)
n = length(x);
if length(y)~=n, error('x and y must be
same length'); end
for i = 1:m+1
for j = 1:i
k = i+j-2;
s = 0;
for l = 1:n
s = s + x(l)^k;
end
A(i,j) = s;
A(j,i) = s;
end
s = 0;
for l = 1:n
s = s + y(l)*x(l)^(i-1);
end
b(i) = s;
end
p = A\b';
```

실행값

```
x=[3 4 5 7 8 9 11 12];
y=[1.6 3.6 4.4 3.4 2.2 2.8 3.8 4.6];

>> HW4_15_4(x,y,3)

ans =

  -11.4887
    7.1438
   -1.0412
    0.0467
```

</td><td>

다항식 회귀를 하기 위하여, 산포도를 가장 적절히 나타낼 수 있는 선을 찾는 문제는 결국, 잔차의 제곱합의 최소값을 구하는 문제로 귀결되고, 이는 편미분을 통하여 구하고자 하는 다항함수의 각 계수를 얻고자 하는 것과 같다. 여기서 편미분을 통하여 얻어지는 식들을 연립하면, 이는 다항식의 차수에 따른 행렬방정식의 문제로 생각할 수 있다. 즉, $A\vec{x}=\vec{b}$를 만족하는 $\vec{x}=\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$을 구하는 문제가 된다. 여기서 이 해집합의 성분은 다항식의 각 계수가 된다. 그러면 이제, 우리는 일반화된 다항식을 가지고 산포도에 가장 적절하게 맞는 회귀선을 만들기 위한 각 계수값을 알고 있으므로, 구한 계수값을 다항식에 대입하여 직선 혹은 곡선의 형태를 갖는 새로운 최소제곱회귀선을 얻을 수 있다.

스크립트에서는 그러한 회귀선의 계수를 손쉽게 구할 수 있도록 하는 함수를 만드는 것에 대한 내용이다.
다항식의 차수, 주어진 data인 x,y 만 갖고 있으면
그에 따른 최소제곱회귀선이 되도록 하는 계수를 곧바로 구할 수 있다.

</td></tr>
</table>

HW4_15_14)

스크립트	설명

```
% a) Linearize
S=[0.01 0.05 0.1 0.5 1 5 10 50 100];
v0=[6.078e-11 7.595e-9 6.063e-8 5.788e-6
1.737e-5 2.423e-5 2.430e-5 2.431e-5
2.431e-5];
ap=1./S.^3;
bt=1./v0;
a=linregr(ap,bt);
Km=1/a(2);
K=Km*a(1);
format long
A=[Km K]'
v0p=Km*S.^3./(K+S.^3);
figure(1)
loglog(S,v0,'o','MarkerFaceColor','r','MarkerEdgeColor','r')
hold on
loglog(S,v0p,'k')
title('Linear regression');
hold off
% b) Nonlinear regression
format long
a1=fminsearch(@fSSR, [2e-5, 1], [], S, v0)
v0p1 = a1(1)*S.^3./(a1(2)+S.^3);
figure(2)
loglog(S,v0,'o','MarkerFaceColor','k','MarkerEdgeColor','k')
hold on
loglog(S,v0p1,'b')
title('Nonlinear regression');
hold off
```

실행값

a)

A = 0.000024154744195　0.397411705178929

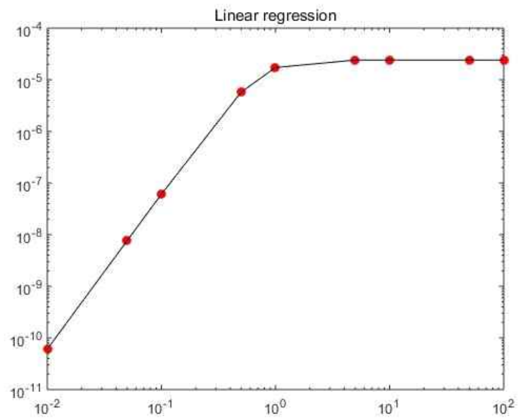
b)

a1 =0.000024309983030　0.399763145338804

a)

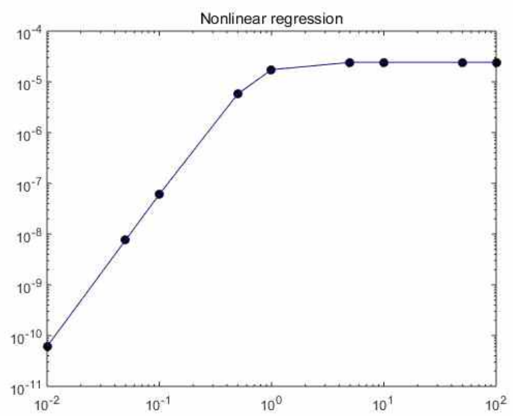
14장에서와 마찬가지로, 선형화한 후 계산한다.
계산 방법은 14장 14번 문제와 유사하므로 생략한다.
기울기와 절편을 구하는 과정은“linregr“ M-file 함수를 이용하였다. 이를 이용하여 Km과 K를 구하고 최종적으로 만들어지는 회귀방정식은 다음과 같다.

$$v_0 = \frac{2.415474 \times 10^{-5}[S]^3}{0.39741 + [S]^3}$$



b)

M-file 함수인 fSSR 과 내장함수 fminsearch를 이용하면 손쉽게 값을 구할 수 있다.
(a)와 b)에서 사용한 M-file 함수는 과제에 같이 첨부)



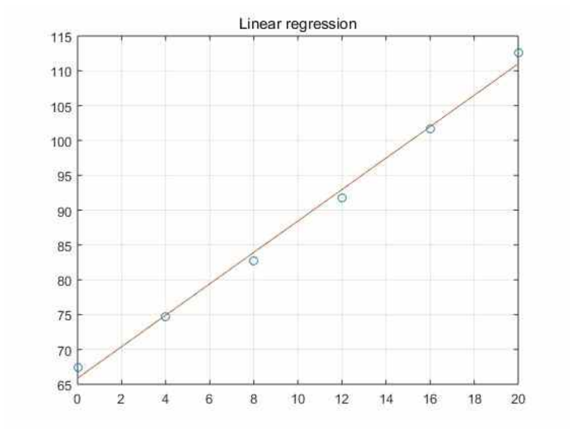
HW4_15_16)

스크립트

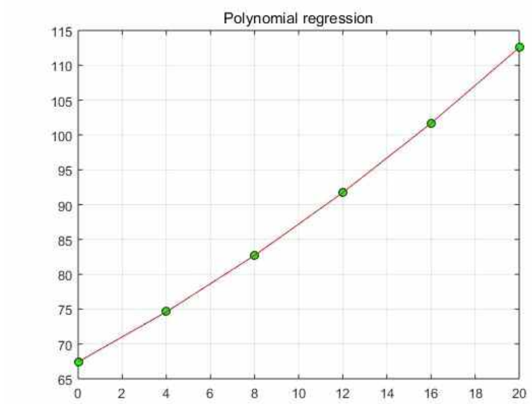
```
clear all
clc
x=[0 4 8 12 16 20];
y=[67.38 74.67 82.74 91.69 101.60 112.58];
% Linear regression
figure(1)
[a1,r2_1]=linregr(x,y);
title('Linear regression');
r2_1
% Polynomial regression
a2=HW4_15_4(x,y,2);
p_2=@(t) a2(3)*t.^2+a2(2)*t+a2(1);
figure(2)
plot(x,y,'o','MarkerFaceColor','g','MarkerEdgeColor','k')
hold on
plot(x,p_2(x),'r')
title('Polynomial regression');
hold off
grid on
sd=sum((y-mean(y)).^2);
resd=sum((y-(p_2(x))).^2);
r2_2=(sd-resd)/sd
% Exponential model
a3=fminsearch(@fSSR2,[1,1],[],x,y);
exn=@(s) a3(1)*exp(a3(2)*s);
figure(3)
plot(x,y,'o','MarkerFaceColor','r','MarkerEdgeColor','k')
hold on
plot(x,exn(x),'k--')
title('Exponential regression');
hold off
grid on
resd_e=sum((y-(exn(x))).^2);
r2_3=(sd-resd_e)/sd
```

설명

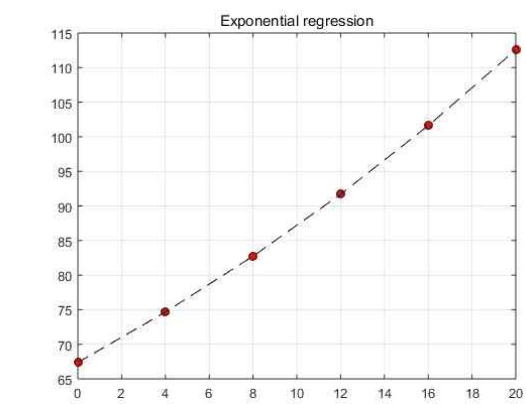
1. Linear regression



2. Polynomial regression



3. Exponential model



실행값 r2_1 = 0.994451342744813 r2_2 = 0.999988910999538 r2_3 = 0.999999973224518	① Linear 의 경우: 결정계수 $r^2 = 0.994451342744813$ ② Polynomial 의 경우: 결정계수 $r^2 = 0.999988910999538$ ③ Exponential 의 경우: 결정계수 $r^2 = 0.999999973224518$ ∴ <u>The order of best fit : ③ > ② > ①</u>

※ 서효원 박사님 특강 中-π 의 수치적 근삿값 구하기

스크립트	설명

```
clear all
clc
% Numerical solution of pi
N=5000;
N_inner=0;
for i=1:N;
    x=rand(N,1);
    y=rand(N,1);
if x(i)^2+y(i)^2 < 1, N_inner=N_inner+1;
end
end
N_total=N;
Nunerical_pi=4*(N_inner/N_total);
disp('Numerical Solution of Pi is')
disp(Numerical_pi)
```

실행값

```
Numerical Solution of Pi is
   3.1452
```

직교좌표계에서, 1×1크기의 정사각형으로 유계된 1사분면 영역을 생각하자, 그리고 원점을 중심으로 하는 반지름이 1인 원의 1사분면 영역을 생각하자.

사각형 전체 넓이에서 사분원의 넓이를 나누면

$$P(X^2 + Y^2 < 1) = \frac{A_{\in \neq r}}{A_{square}} = \frac{\pi}{4}$$

인 확률적인 문제로 생각할 수 있다.

따라서 π의 수치적인 근사값을 구하기 위한 모델은

$$\pi = 4\left(\frac{N_{\in \neq r}}{N_{total}}\right)$$

$N_{\in \neq r}$: 사분원 내의 난수의 개수

N_{total} : 사각형 전체 난수의 개수