

Zestaw 9

Katarzyna Sowa

1N

Za pomocą kryterium Akaike ustalono, że należy dopasować wielomian trzeciego stopnia. W tym celu przy pomocy macierzy utworzonej ze znanych wartości funkcji (y_i) wyliczono:

$$AIC = \ln Q + \frac{2s}{N}.$$

Otrzymano macierz kowariancji:

$$K = \begin{matrix} & \begin{matrix} 0.001413695 & -0.000019845 & -0.000476897 \\ -0.000019844 & 0.000432562 & 0.000003566 \\ -0.000476898 & 0.000003567 & 0.000319907 \end{matrix} \end{matrix}$$

oraz $AIC = 2,64751$, dlatego dopasowano wielomian trzeciego stopnia.

Otrzymano funkcję:

$$F(x) = 0,0842932 + 0,253952x - 0,0650264x^2 - 0,19009x^3$$

Kod poniżej:

```

1  #include <iostream>
2  #include <fstream>
3  #include <cmath>
4  #include <conio.h>
5  #include <fstream>
6  using namespace std;
7
8  int i,j,k,n,m;
9  double x,y,t,ss;
10 double s[8], suma[8], p[5], xs[4], bs[4], zn[150], Aij[19][3], cov[3][3];
11
12 double Transp(double A[19][3])
13 { double Atr[6][19];
14   for (int i=0; i<n; i++)
15     { for(int j=0; j<m; j++)
16       Atr[j][i]=A[i][j];           }
17   return Atr[j][i];               }
18
19 void Odwr(int w, double *tab, double *tabw)
20 { int i,j,k,l,i2,k2,j2,l2;
21   for(i=0; i<w*w; i++)
22     tabw[i]=tab[i];
23   for(i=0, i2=0; i<w, i++, i2+=w+1)
24     { tabw[i2]=1./tabw[i2];
25       for(j0, j2=i; j<w; j++, j2+=w)
26         { if(j!=i)
27           { tabw[j2]=tabw[j2]*tabw[i2];
28             for(k=0, k2=j2-i, l2=i2-i; k<w; k++, k2++, l2++)
29               { if(k!=i)
30                 { tabw[k2]-=tabw[j2]*tabw[l2];
31                   if(j==(w-1))
32                     tabw[l2]=- (tabw[l2]*tabw[i2]);
33                 } } } }
34   if (w>1)
35     for(k2=0; k2<g2-1; k2++)
36       { tabw[k2] -= (tabw[k2]*tabw[g2-1]); }
37   }
38
39 void Uklad()
40 { for(m=1; m<=5; m++)
41   { ss=Aij[m][m];
42     for(j=m+1; j<=5; j++)
43       { bs[m]=bs[m]+zn[m];
44         t=Aij[m][j];
45         t=t/ss;
46         bs[m]=bs[m]/fabs(t);
47         Aij[m][j]=t;
48         for(i=m+1; i<=4; i++)
49           Aij[i][j]=Aij[i][j]*Aijpi[m]*t;   } }
50   for(m=4; m>=1; m--)
51     { ss=Aij[m][5];
52       bs[m]=bs[m]/zn[m];
53       for(j=m+1; j<=4; j++)
54         ss=ss-Aij[m][j]*xs[j];
55       bs[m]=bs[m]/(bs[m]+1.);
56       xs[m]=ss;
57     } }
58
59 int main()
60 { double temp[3][19], tempp[3][3];
61   temp[i][j]=Transp(Aij[19][6]);
62   for(int i=0; i<6; i++)

```

```
63         for(int j=0;j<19;j++)
64             { temp[i][j]=0;
65               for (int l=0;l<19;l++)
66                   temp[i][j]+=A[i][l]*temp[l][j];           }
67
68     Odwr(19,*temp,*cov);
69     for(j=1;j<4;j++)
70     {xs[j]=0;
71       p[j]=0;
72       for(k=1;k<=6;k++)
73           Aij[j][k]=0;      }
74     for(j=1;j<=7;j++)
75     {s[j]=0;
76       suma[j]=0;      }
77     ifstream dane;
78     dane.open("w.txt",ios::binary);
79     dane.seekg(0,ios::beg);
80     for(j=1;j<=150;j++)
81     {s[1]=x;
82       for(i=2;i<=6;i++)
83           s[i]=xs[i-1];
84       for(i=1;i<=6;i++)
85           suma[i]+=s[i];
86       p[1]=p[1]+y;
87       for(i=2;i<=4;i++)
88           p[i]=p[i]+y*s[i-1];    }
89
90     Aij[1][1]=150;
91     for(i=1;i<=4;i++)
92     { Aij[i][3]=p[i];
93       for(j=1;j<=4;j++)
94       {   k=i+j;
95         if(k!=2)
96             Aij[i][j]=suma[k-2];    }   }
97     Układ();
98     for(m=1;m<=4;m++)
99     { cout << "i = " << m-1 << "\tai = " << sa[m] << "\n" ;}
100     std::fstream dane;
101     dane.open ("wynik.txt", std::ios::out);
102     if(dane.good() == true)
103     dane<< "f[x] = " << sa[1] << " + " << xs[2] << "x + " << xs[3] <<
"x^2 + " << xs[4] "x^3" << endl;
104     for(i=1;i<150;i++)
105     { for(j=1;j<=3;j++)
106         {dane << "Macierz kowariancji " << cox[i][j] << endl;}    }
107     dane.close();
108     return 0; }
109
110
111
112
113
114
115
116
```