

Spline-Regression.R

```
# Mengimpor library readxl untuk membaca file Excel
library(readxl)
library(PLRModels)

# Membaca data dari file Excel dan menyimpannya dalam variabel 'data'
data <- read_excel("D:/data_Indonesia.xlsx")

# Mengurutkan data berdasarkan nilai 'CO AQI Value'
data <- data[order(data$'CO AQI Value'),]

# Menampilkan data yang telah diurutkan
data

## # A tibble: 379 × 3
##   `CO AQI Value` `PM2.5 AQI Value` `AQI Value`
##           <dbl>           <dbl>     <dbl>
## 1             0             21         21
## 2             0             55         55
## 3             0             35         35
## 4             0             14         17
## 5             0             30         30
## 6             0             34         34
## 7             0             28         28
## 8             0             17         17
## 9             0             10         10
## 10            0             25         25
## # i 369 more rows

# Mengambil variabel 'CO AQI Value' dan 'AQI Value' dari data
x = data$'CO AQI Value'
y = data$'AQI Value'

# Fungsi untuk menghitung Generalized Cross Validation (GCV)
gcv1<-function(y,x,m,l)
{
  a<-min(x)+1
  b<-max(x)-1
  k<-seq(a,b,l)
  v<-length(k)
  n<-length(y)
  Gcv<-matrix(nrow=v,ncol=1)
  Mse<-matrix(nrow=v,ncol=1)
  for (j in 1:v)
  {
    w<-matrix(0,ncol=m+1,nrow=n)
```

```

    for (i in 1:m)
      w[,i]<-x^(i-1)
    for (i in m+1)
      w[,i]<-trun(x,k[j],m-1)
    wtw<- t(w) %**% w
    z<- MPL(wtw)
    beta<- z %**% (t(w) %**% y)
    h<- w %**% z %**% t(w)
    mu<-w%**beta
    MSE<- t(y-mu) %**% (y-mu)/n
    I<-matrix(0,ncol=n,nrow=n)
    for(i in 1: n)
      I[i,i]<-1
    GCV<-(n^2*MSE)/(sum(diag(I-h)))^2
    Gcv[j]<-GCV
    Mse[j]<-MSE
  }
  R<-matrix(c(k,Gcv,Mse),ncol=3)
  sort.R<-R[order(R[,2]),]
  S<-sort.R[1:10,]
  cat("Untuk spline order",m,"dengan 1 titik knot, diperoleh knot
  optimal=",S[1,1]," dengan GCV minimum=",S[1,2],"dan MSE =",S[1,3])
  cat("\nBerikut 10 nilai GCV terkecil, nilai MSE dan letak titik
  knotnya:\n")
  cat("=====\n")
  cat(" No   Ttk knot   GCV      MSE   \n")
  cat("=====\n")
  S
}

# Fungsi untuk metode Pseudoinverse Least Squares (MPL)
MPL<-function(x,eps=1e-20)
{
  x<-as.matrix(x)
  xsvd<-svd(x)
  diago<-xsvd$d[xsvd$d>eps]
  if(length(diago)==1)
  {
    xplus<-as.matrix(xsvd$v[,1])%**%t(as.matrix(xsvd$u[,1])/diago)
  }
  else
  {
    xplus<-
      xsvd$v[,1:length(diago)]%**%diag(1/diago)%**%t(xsvd$u[,1:length(diago)])
  }
  return(xplus)
}

# Fungsi untuk membentuk model spline
model.spline=function(prediktor,respon,m,knots=c(...))

```

```

{
  y<-respon
  n<-length(y)
  k<-length(knots)
  w<-matrix(0, ncol=m+k, nrow=n)
  for (i in 1:m)
    w[,i]<-prediktor^(i-1)
  for(i in (m+1):(m+k))
    w[,i]<-trun(prediktor,knots[i-m],m-1)
  wtw<-t(w)%*%w
  Z<-MPL(wtw)
  beta<-Z%*%t(w)%*%y
  yfits<-w%*%beta
  res<-y-yfits
  MSE<-t(y-yfits)%*%(y-yfits)/n
  I<-matrix(0,ncol=n,nrow=n)
  for(i in 1:n)
    I[i,i]<-1
  h<-w%*%MPL(wtw)%*%t(w)
  GCV<-(n^2*MSE)/(sum(diag(I-h)))^2
  q<-seq(min(prediktor),max(prediktor),length=1000)
  u<-matrix(0,ncol=m+k,nrow=1000)
  cat("\n Spline orde",m)
  cat("\n Titik Knots   = c( ",format(knots),")")
  cat("\n Nilai GCV      = ",format(GCV),
      "\n Nilai MSE     = ",format(MSE),"\\n")
  cat("\\n
  *****")
  cat("\\n      Koefisien      Estimasi")
  cat("\\n
  *****")
  for(i in 1:(m+k))
    cat("\\n      beta[",i-1,"]      ",format(beta[i]))
  cat("\\n
  *****")
  par(mfrow=c(1,1))
  z0=cbind(prediktor,respon)
  z1=z0[order(z0[,1]),]
  x1=z1[,1]
  y1=z1[,2]
  w1<-matrix(0, ncol=m+k, nrow=n)
  for (i in 1:m)
    w1[,i]<-x1^(i-1)
  for(i in (m+1):(m+k))
    w1[,i]<-trun(x1,knots[i-m],m-1)
  yfits1<-w1%*%beta
  plot(x1,y1, type="p",xlim=c(min(prediktor),max(prediktor)),ylim=c(0,300),
       xlab="COVal",ylab="AQIVal")
  par(new=T)
  plot(x1,yfits1, type="l",col="red",

```

```

    xlim=c(min(prediktor),max(prediktor)),
    ylim=c(0,300),
    xlab=" ",ylab=" ")
}

# Fungsi untuk melakukan transformasi pada data
trun <- function(data,a,power)
{
  data[data<a] <- a
  (data-a)^power
}

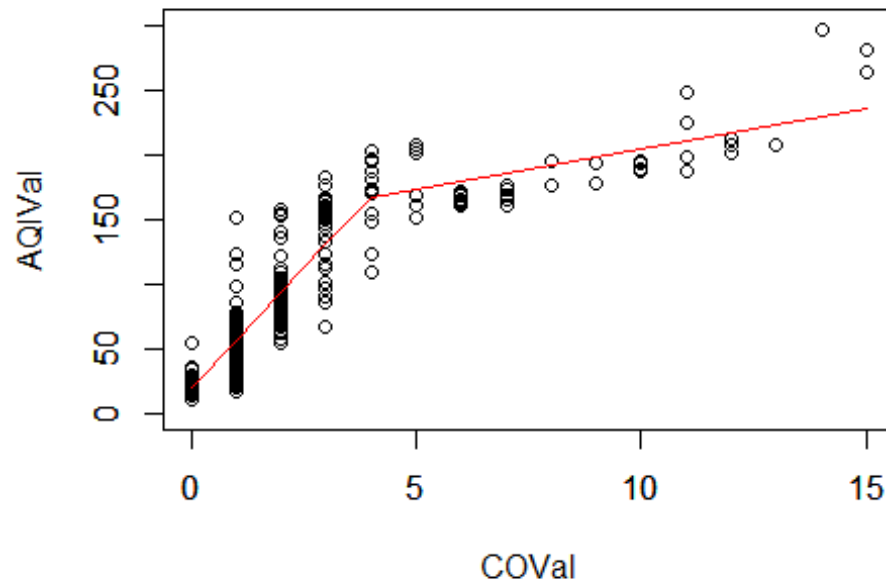
# Memanggil fungsi gcv1 untuk spline order 2 dengan 1 titik knot
gcv1(y, x, 2, 1)

## Untuk spline order 2 dengan 1 titik knot, diperoleh knot optimal= 4
## dengan GCV minimum= 451.6653 dan MSE = 444.5432
## Berikut 10 nilai GCV terkecil, nilai MSE dan letak titik knotnya:
## =====
##   No   Ttk knot   GCV      MSE
## =====
##      [,1]      [,2]      [,3]
## [1,]      4 451.6653 444.5432
## [2,]      3 467.6081 460.2346
## [3,]      5 510.6123 502.5607
## [4,]      6 598.2327 588.7995
## [5,]      2 627.1142 617.2256
## [6,]      7 655.3945 645.0599
## [7,]      8 698.1031 687.0951
## [8,]      9 750.7948 738.9559
## [9,]     10 802.3447 789.6929
## [10,]     11 832.7957 819.6637

# Memanggil fungsi model.spline untuk spline order 2 dengan knot pada nilai 4
model.spline(x, y, 2, c(4))

##
## Spline orde 2
## Titik Knots = c( 4 )
## Nilai GCV   = 451.6653
## Nilai MSE   = 444.5432
##
## *****
##      Koefisen      Estimasi
## *****
##      beta[ 0 ]      19.82167
##      beta[ 1 ]      36.8627
##      beta[ 2 ]     -30.64115
## *****

```



```
# Memanggil fungsi gcv1 untuk spline order 3 dengan 1 titik knot
gcv1(y, x, 3, 1)
```

```
## Untuk spline order 3 dengan 1 titik knot, diperoleh knot optimal= 7
dengan GCV minimum= 468.823 dan MSE = 458.9792
```

```
## Berikut 10 nilai GCV terkecil, nilai MSE dan letak titik knotnya:
```

```
## =====
##   No   Ttk knot   GCV     MSE
## =====
```

```
##      [,1]      [,2]      [,3]
## [1,]    7 468.8230 458.9792
## [2,]    6 470.1112 460.2404
## [3,]    8 475.0669 465.0920
## [4,]    9 484.0824 473.9183
## [5,]    5 486.4972 476.2823
## [6,]   10 494.9016 484.5102
## [7,]   11 506.0165 495.3918
## [8,]   12 516.9226 506.0688
## [9,]    4 517.3795 506.5162
## [10,]  13 531.9625 520.7929
```

```
# Memanggil fungsi model.spline untuk spline order 3 dengan knot pada nilai 7
model.spline(x, y, 3, c(7))
```

```
##
##   Spline orde 3
```

```
## Titik Knots = c( 7 )
## Nilai GCV   = 468.823
## Nilai MSE   = 458.9792
##
## *****
##      Koefisien      Estimasi
## *****
##      beta[ 0 ]      9.991033
##      beta[ 1 ]      51.39532
##      beta[ 2 ]      -3.841005
##      beta[ 3 ]      5.705
## *****
```

