finalproject

phantomOfLaMancha

3/26/2021

```r
get.reduced.model = function(model, i){
  # convenient helper to return the new model with ith feature removed
  # i can be vector or number

  # first column of data will be response variable, other columns are features of original
  # model, intercept wouldn't appear here as a feature
  data = model$model
  r = nrow(data)
  c = ncol(data)

  # special case if there is only 1 feature left
  if(c==2){
    return(lm(data[1:r,1]~1))
  }

  # we shouldn't receive a model with only intercept
  if(c==1){
    stop("get.reduced.model() recieved a model with intercept only")
  }

  # explanatory variable
  names = colnames(data)[2:c]
  # response variable
  yname = colnames(data)[1]
  formu = as.formula( paste(yname, "~", paste( names[-i], collapse = "+")))
  # new model
  m =  lm(formu , data=data)
  return(m)
}

## note: right now this function could only do 10 fold

get_col <- function(mat,i,j, breaks, cols=NULL, palette="Blues") {
    if (is.null(cols)) {
        cols <- brewer.pal(length(breaks)+1, palette)}
    val <- 1
    for (b in breaks) {
      if (is.na(mat [i,j])){
        val <- 0
      }
      else if (mat[i,j] > b) {
            val <- val + 1}
            }
    cols[val]
    }

require(RColorBrewer)
```

```
## Loading required package: RColorBrewer
```

```
col_areas <- function(matrix,
                                                 breaks=NULL,
                                                 cols=NULL,
                                                 palette="Blues",
                                                 xlab="West    <----------->    East",
                                                 ylab="South   <----------->   North",
                                                 ...){
    if (is.null(breaks)) {
            breaks <- unique(fivenum(matrix))}

  plot(c(0, 100*ncol(matrix)),
            c(0, 100*nrow(matrix)), frame.plot=TRUE,
            type="n",
            xlab=xlab,
            ylab=ylab, axes=FALSE, ...)

  nr <- nrow(matrix)
  nc <- ncol(matrix)
    for (i in 1:nr) {
        for (j in 1:nc) {
            rect((j-1)*100,
                 (nr-i+1)*100,
                 j*100,
                 (nr-i)*100,
                 border=NA,
                 col=get_col(matrix,i,j,breaks,cols,palette))
                 }
               }
}
```

understanding our polulation:

```
library("eikosograms")
```

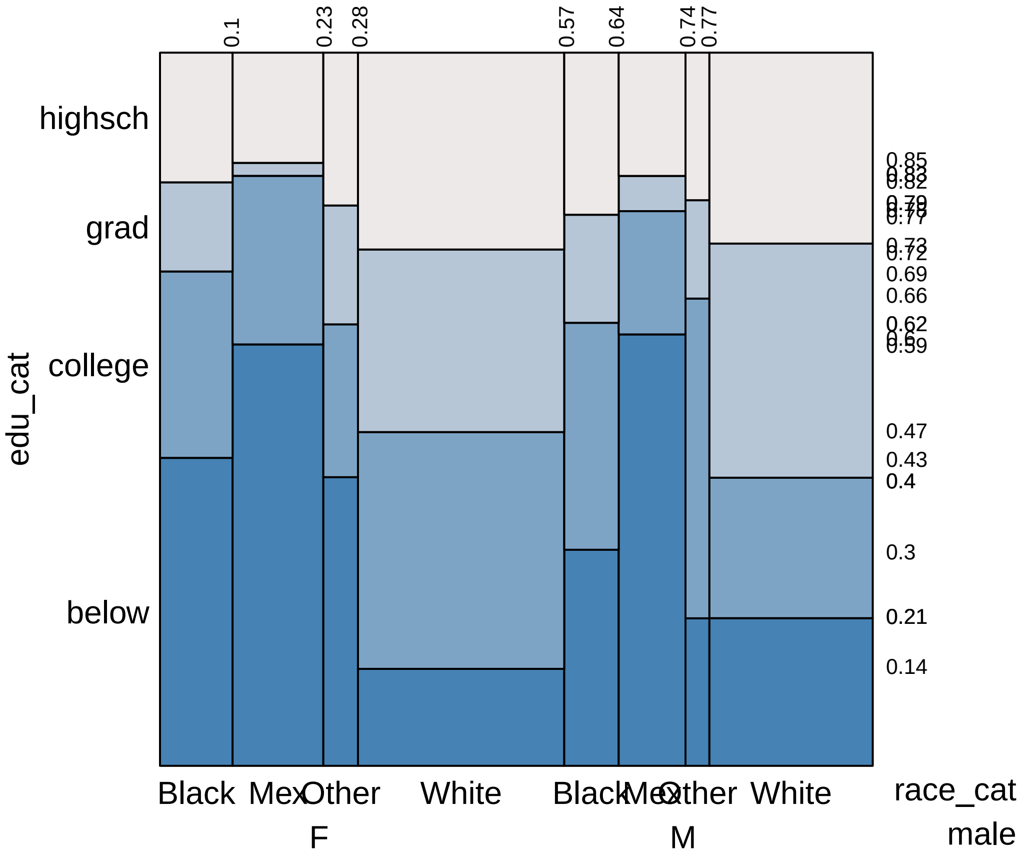
```
## Warning: package 'eikosograms' was built under R version 4.0.4
```

```
library("venneuler")
```

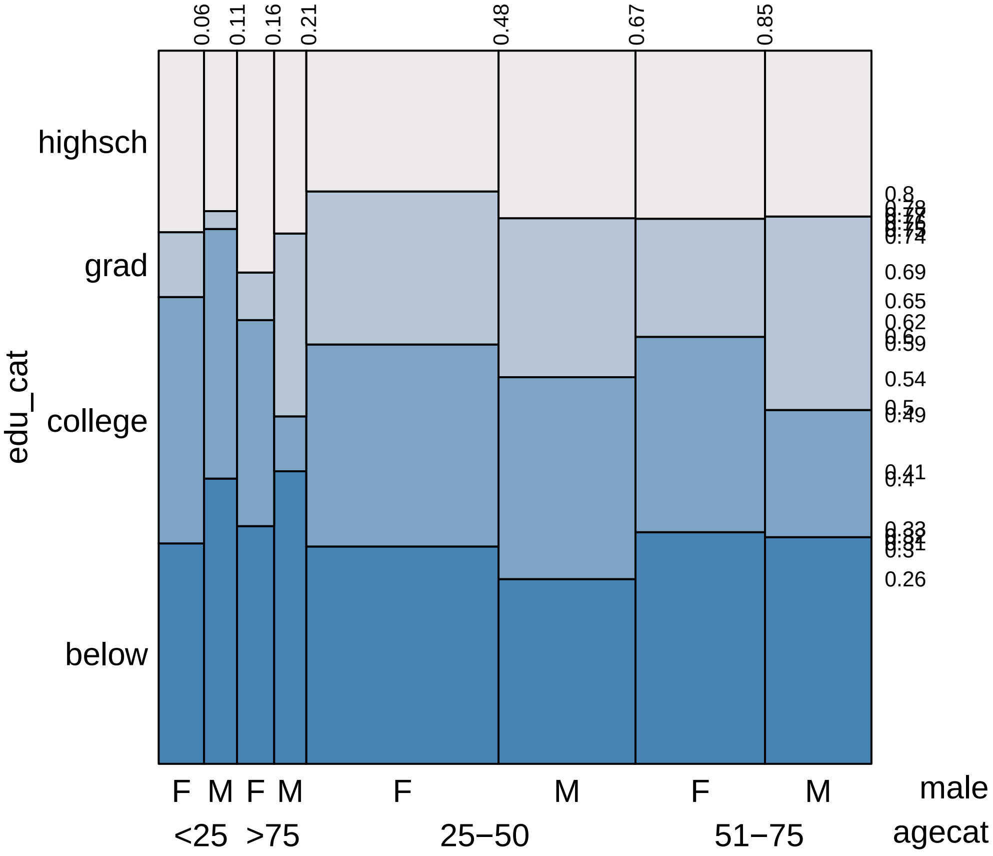
```
## Warning: package 'venneuler' was built under R version 4.0.3

## Loading required package: rJava

## Warning: package 'rJava' was built under R version 4.0.3
```

```
data = read.csv("pollutants.csv")

# change factor features to reasonable names

ind = data$male == 1
```

```r
data$male[ind] = "M"
data$male[!ind] = "F"
data$agecat = ceiling(data$ageyrs/25 )
agecat = c("<25","25-50","51-75",">75")

for (i in 1:4){
  ind = data$agecat == i
  data$agecat[ind] = agecat[i]
}


edu=c("below", "highsch", "college","grad")
for (i in 1:4){
  ind = data$edu_cat == i
  data$edu_cat[ind] = edu[i]
}

race=c("Other", "Mex", "Black","White")
for (i in 1:4){
  ind = data$race_cat == i
  data$race_cat[ind] = race[i]
}




eikos(edu_cat~ race_cat + male ,data=data)
```



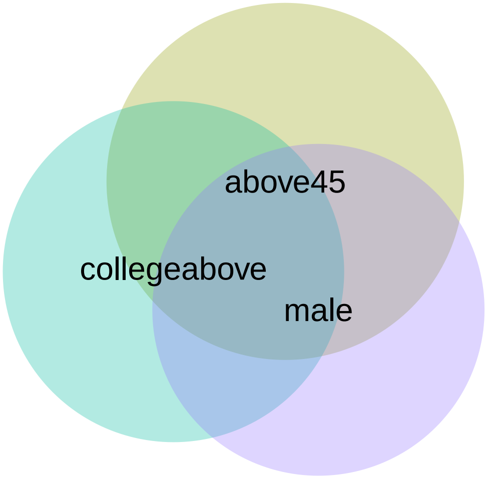
```
eikos(edu_cat~ male+agecat ,data=data)
```



```
# look at intersection

# note surface of above 45 should be approximately half of surface of total population

collegeabove = which( (data$edu_cat == "college") + (data$edu_cat == "grad") ==1 )
collegeabove.names = rep("collegeabove", length(collegeabove ))

white= which( data$race_cat == "White" )
white.names = rep("White", length(white))

median(data$ageyrs)
```

```
## [1] 46
```

```
above45 = which(data$ageyrs>45)
above45.names= rep("above45", length(above45))

male = which(data$male == "M")
male.names = rep("male", length(male))

female = which(data$male == "F")
female.names = rep("female", length(female))

subjectinfo = c(above45, collegeabove, male)
names = c(above45.names , collegeabove.names, male.names)
ven = venneuler(data.frame(elements = subjectinfo, sets=names))
plot(ven)
```



```
# get rid of the agecat data we added
if (colnames(data)[ ncol(data)] == "agecat"){
  data = data[,-ncol(data)]
}
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.0.4
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-1
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.0.4
```

```
## Loading required package: carData
```

```
data = read.csv("pollutants.csv")

# the index does not really mean anything
data = data[,-1]

nTotal = nrow(data)

#change some feature to factor type
data$race_cat = factor(data$race_cat)
data$edu_cat = factor(data$edu_cat)
```

```
data$male = factor(data$male)
data$smokenow= factor(data$smokenow)

data.train = data[1:700,]
data.test = data[701:nTotal,]
runif(1)
```

```
## [1] 0.1793597
```

correlation between features high correlation -> coeffients have large variance

```
model = lm(length~. , data=data)
#original vif

vif(model)
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## POP_PCB1           33.044120  1        5.748401
## POP_PCB2           34.281125  1        5.855009
## POP_PCB3            9.351143  1        3.057964
## POP_PCB4           31.742239  1        5.634025
## POP_PCB5           59.896895  1        7.739308
## POP_PCB6           11.386658  1        3.374412
## POP_PCB7            4.870075  1        2.206825
## POP_PCB8           12.982575  1        3.603134
## POP_PCB9           12.441595  1        3.527264
## POP_PCB10           6.020678  1        2.453707
## POP_PCB11           4.725769  1        2.173883
## POP_dioxin1         5.276251  1        2.297009
## POP_dioxin2         5.413132  1        2.326614
## POP_dioxin3         4.398509  1        2.097262
## POP_furan1          6.154213  1        2.480769
## POP_furan2          6.195336  1        2.489043
## POP_furan3          4.464346  1        2.112900
## POP_furan4          1.821809  1        1.349744
## whitecell_count     1.548380  1        1.244339
## lymphocyte_pct  12250.336528  1      110.681238
## monocyte_pct      726.843372  1       26.960033
## eosinophils_pct 15071.561945  1      122.766290
## basophils_pct     867.412798  1       29.451873
## neutrophils_pct    37.984114  1        6.163125
## BMI                 1.263662  1        1.124127
## edu_cat             1.543109  3        1.074978
## race_cat            2.052848  3        1.127352
## male                1.350324  1        1.162034
## ageyrs              3.238631  1        1.799620
## yrssmoke            2.204139  1        1.484634
## smokenow            4.006708  1        2.001676
## ln_lbxcot           3.963407  1        1.990831
```

```
t1=colnames( model$model)


while (TRUE) {
  score = vif(model)
  if (max(score) <10){
    break
  }
  ind = which.max(score)
  # this is safe with factor data type
  model = get.reduced.model(model, ind)
}
# reduced model vif
vif(model)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## POP_PCB3        5.310340  1        2.304417
## POP_PCB6        9.083828  1        3.013939
## POP_PCB7        4.686485  1        2.164829
## POP_PCB8        5.894052  1        2.427767
## POP_PCB9        7.640480  1        2.764142
## POP_PCB10       5.149483  1        2.269247
## POP_PCB11       4.210120  1        2.051858
## POP_dioxin1     5.184345  1        2.276916
## POP_dioxin2     5.275271  1        2.296796
## POP_dioxin3     4.311410  1        2.076394
## POP_furan1      6.000097  1        2.449509
## POP_furan2      6.154621  1        2.480851
## POP_furan3      4.412739  1        2.100652
## POP_furan4      1.812793  1        1.346400
## whitecell_count 1.533642  1        1.238403
## lymphocyte_pct  1.370966  1        1.170882
## monocyte_pct    1.255543  1        1.120510
## basophils_pct   1.097132  1        1.047441
## neutrophils_pct 1.083675  1        1.040997
## BMI             1.257562  1        1.121411
## edu_cat         1.498239  3        1.069704
## race_cat        2.012804  3        1.123657
## male            1.345703  1        1.160045
## ageyrs          3.224432  1        1.795670
## yrssmoke        2.147610  1        1.465473
## smokenow        3.967106  1        1.991759
## ln_lbxcot       3.946223  1        1.986510
```

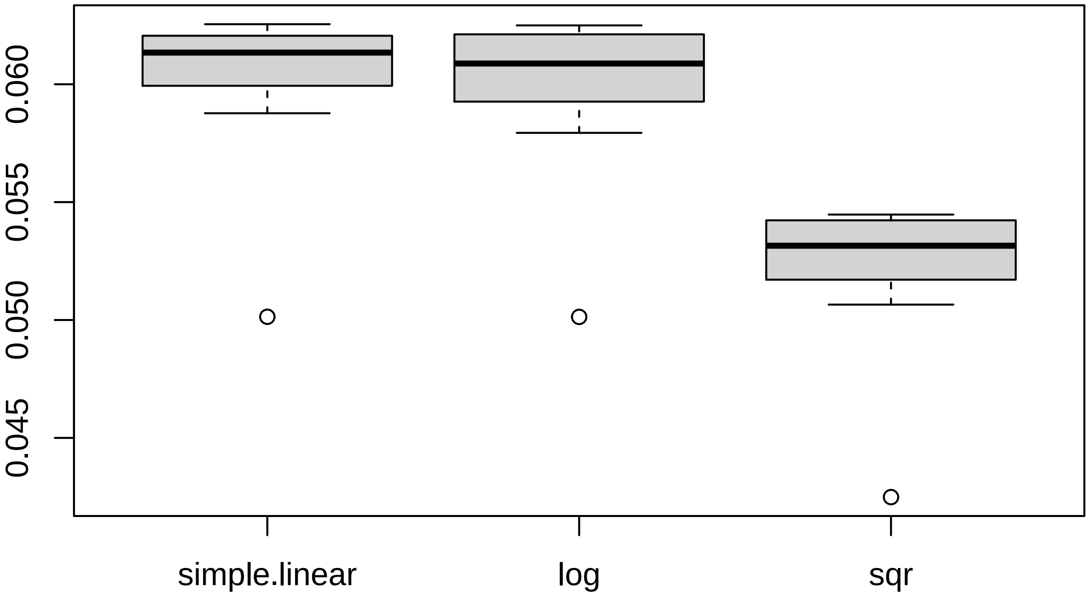
```
t2=colnames( model$model)

setdiff(t1,t2)
```

```
## [1] "POP_PCB1"        "POP_PCB2"        "POP_PCB4"        "POP_PCB5"
## [5] "eosinophils_pct"
```

does one feature alone explain the model?

we fit length to each corvariate in a linear/log/square model

```
Xfull = lm(length~., data=data)$model

res = matrix(0, nrow = (ncol(Xfull)), ncol = 3)


for(c in 2:ncol(Xfull)){
  model = lm(data$length~Xfull[,c])
  #res[,1] is simple linear models
  #res[,2] is log linear models
  #res[,3] is square models
  res[c,1] = mean(model$residuals^2)
  # we won't fit log or sqaure model for catogrical variable because it's bad
  if(! is.factor(Xfull[,c])){
    modelpower2 = lm(data$length~poly( Xfull[,c], 2))
    modellog = lm(log(data$length)~ Xfull[,c])
    res[c,2] = mean(modelpower2$residuals^2)
    res[c,3] = mean(modellog$residuals^2)
  }
}

removezero = function(v){
  v[v==0] = NA
  v
}

# how do these models perform in terms of mse
box = list(simple.linear=removezero(res[,1]), log=removezero(res[,2]), sqr=removezero(res[,3]) )
boxplot(box)
```



```
which.min(removezero(res[,1]))
```

```
## [1] 30
```

```
which.min(removezero(res[,2]))
```

```
## [1] 30
```

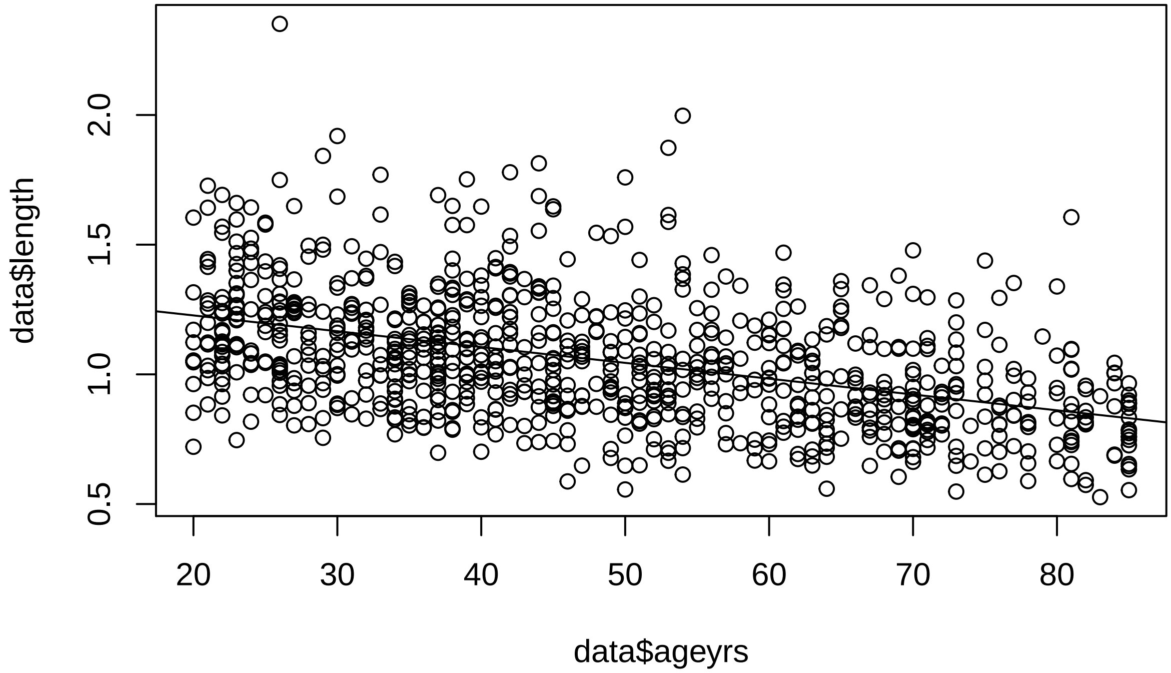
```
which.min(removezero(res[,3]))
```

```
## [1] 30
```

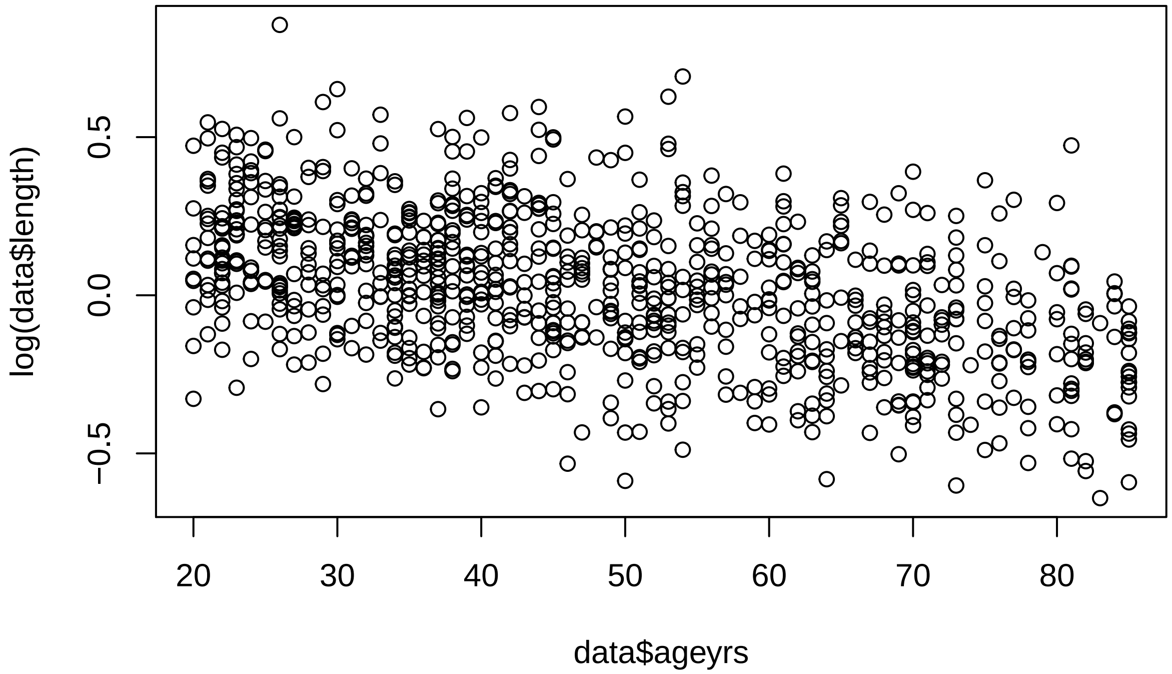
```
# which is the best single feature
colnames(Xfull)[30]
```

```
## [1] "ageyrs"
```

```
# what does the best model look like
simplelinear = lm(length~ageyrs, data=data)
plot(data$ageyrs, data$length)
abline(simplelinear$coefficients)
```



```
simplelog= lm(log(length)~ageyrs, data=data )
plot(data$ageyrs, log(data$length))
```



```
summary(simplelog)
```

```
## 
## Call:
## lm(formula = log(length) ~ ageyrs, data = data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.6025 -0.1358 -0.0042  0.1359  0.6999 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.3155139  0.0198704   15.88   <2e-16 ***
## ageyrs      -0.0059956  0.0003844  -15.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2064 on 862 degrees of freedom
## Multiple R-squared:  0.2201, Adjusted R-squared:  0.2192 
## F-statistic: 243.2 on 1 and 862 DF,  p-value: < 2.2e-16
```

```
summary(simplelinear)
```

```
## 
## Call:
## lm(formula = length ~ ageyrs, data = data)
## 
```

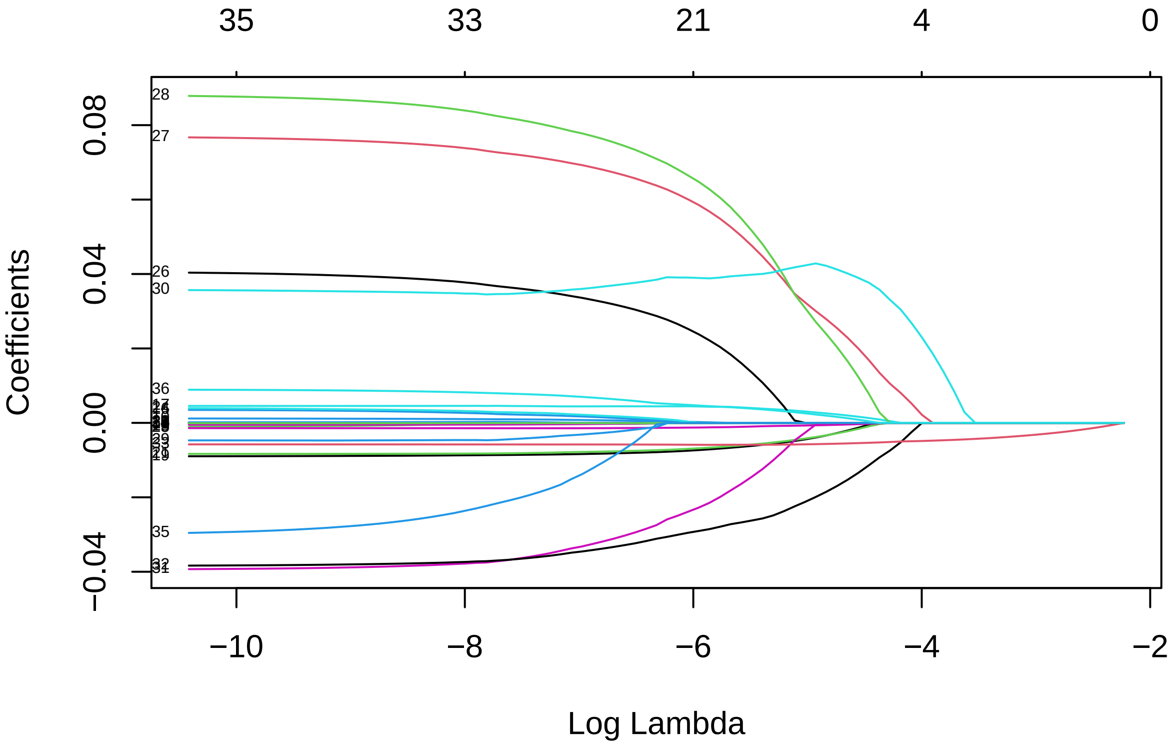
```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.50649 -0.15488 -0.02453  0.12404  1.16057 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.3492575  0.0215848   62.51   <2e-16 ***
## ageyrs      -0.0060995  0.0004176  -14.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2242 on 862 degrees of freedom
## Multiple R-squared:  0.1984, Adjusted R-squared:  0.1975 
## F-statistic: 213.4 on 1 and 862 DF,  p-value: < 2.2e-16
```

seems there is a linear relationship but looks insufficient.

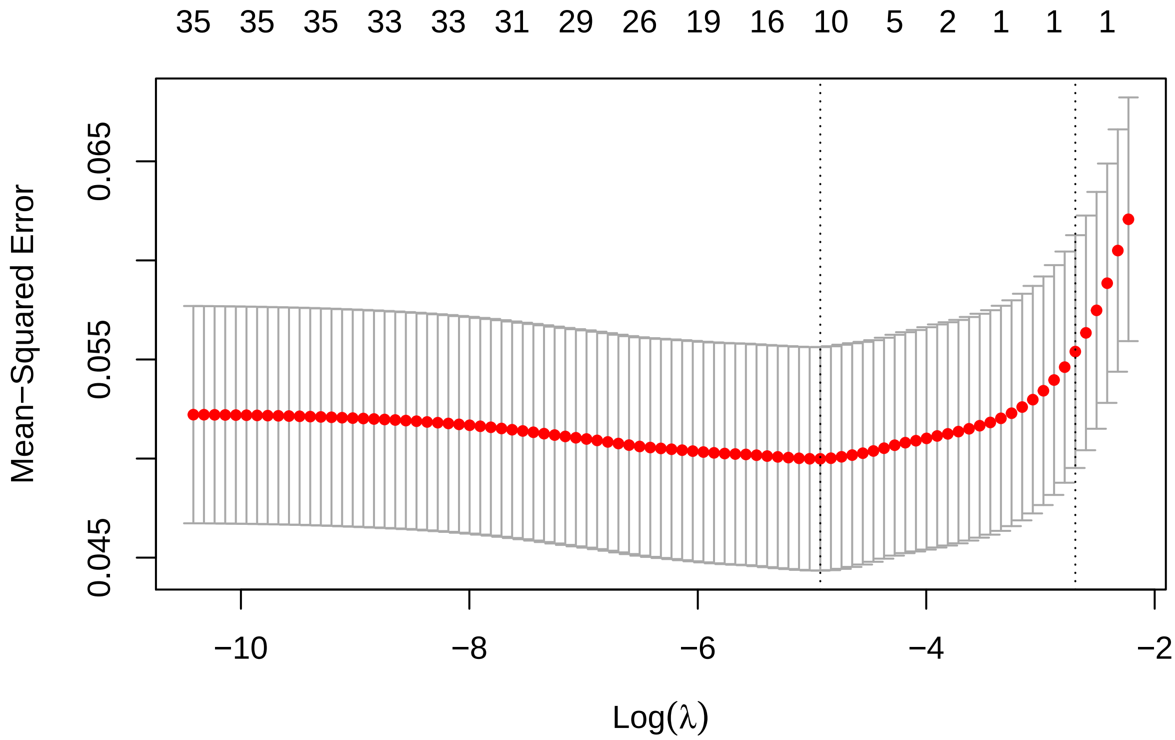
Also seems sqr or log does not do exponentially better here

how is model choosen by automated soluation

```r
### LASSO
## fit models
M = model.matrix(lm(length~., data=data))
y_train = data$length[1:700]
X_train = M[1:700,-1]
y_test= data$length[701:nTotal]
X_test= M[701:nTotal,-1]

M_lasso <- glmnet(x=X_train,y=y_train,alpha = 1)
####

####
## plot paths
plot(M_lasso,xvar = "lambda",label=TRUE)
```



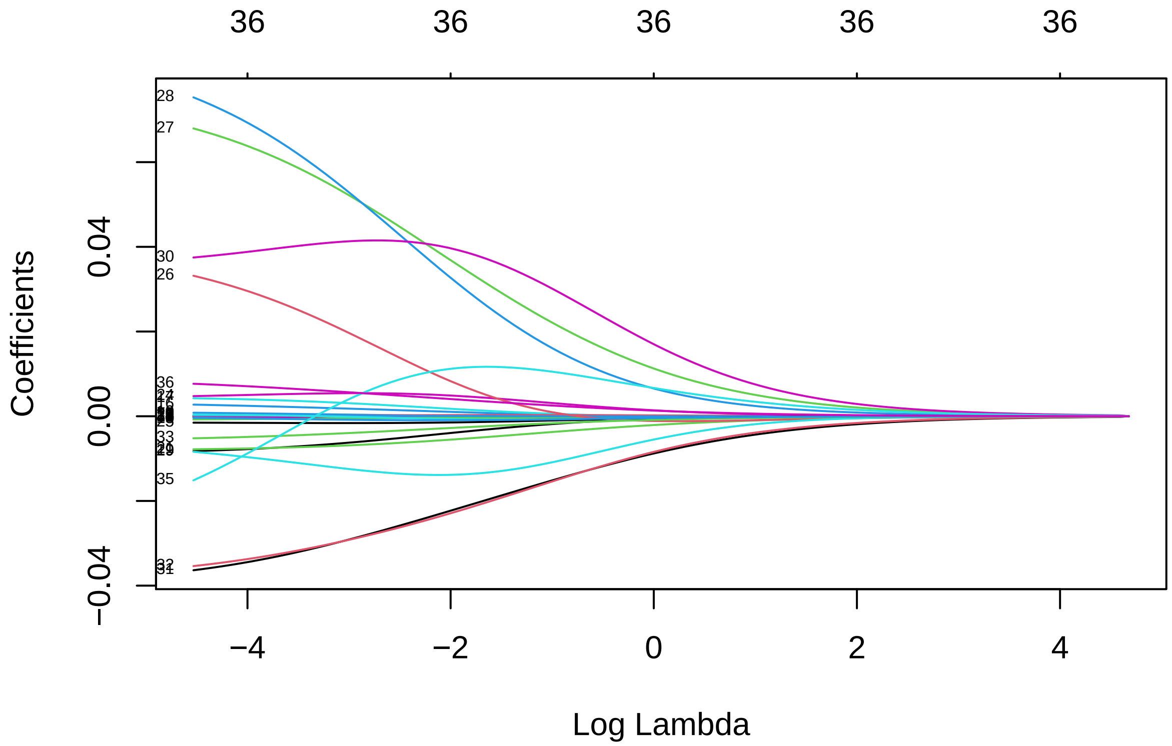
```
## fit with crossval
cvfit_lasso <-  cv.glmnet(x=X_train,y=y_train,alpha = 1)

## plot MSPEs by lambda
plot(cvfit_lasso)
```



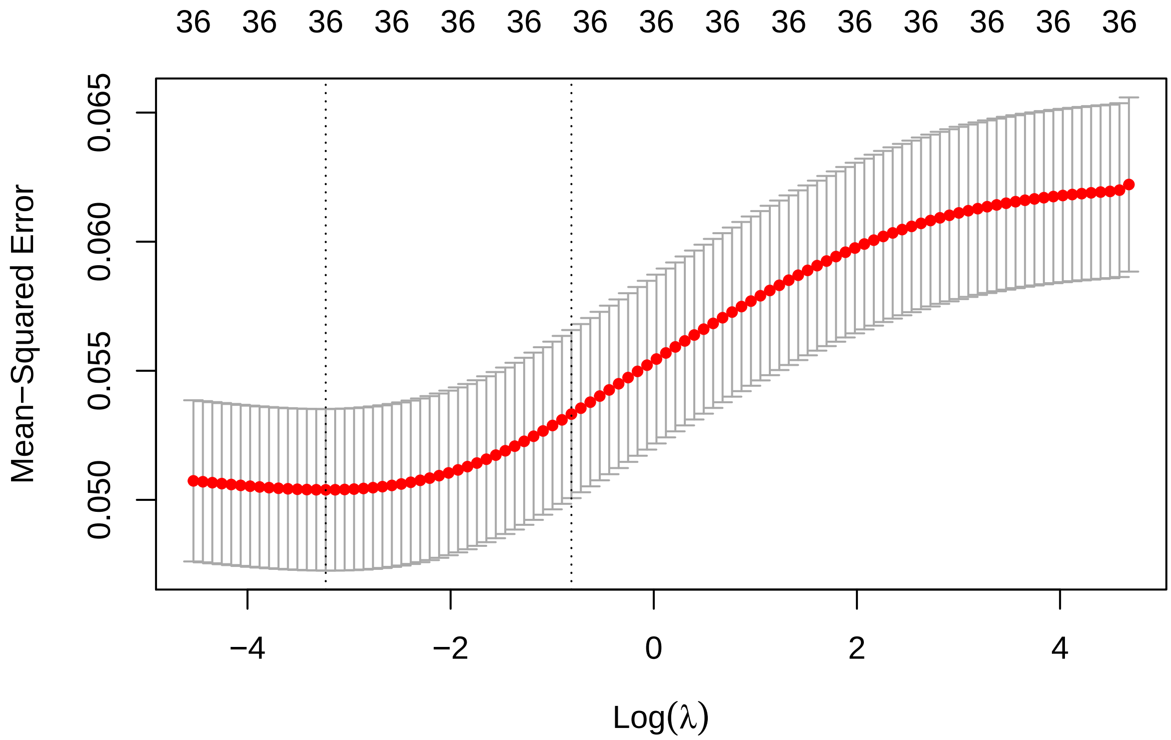
```
## estimated betas for minimum lambda
coef(cvfit_lasso, s = "lambda.min")
```

```
## 37 x 1 sparse Matrix of class "dgCMatrix"
##                             1
## (Intercept)      1.3842232601
## POP_PCB1         .
## POP_PCB2         .
## POP_PCB3         .
## POP_PCB4         .
## POP_PCB5         .
## POP_PCB6         .
## POP_PCB7         .
## POP_PCB8         .
## POP_PCB9         .
## POP_PCB10        .
## POP_PCB11        .
## POP_dioxin1      .
## POP_dioxin2      .
## POP_dioxin3      .
## POP_furan1       .
## POP_furan2       .
## POP_furan3       0.0028070052
## POP_furan4       .
## whitecell_count -0.0039335511
## lymphocyte_pct   .
## monocyte_pct    -0.0038009579
## eosinophils_pct  .
## basophils_pct    .
## neutrophils_pct  .
## BMI             -0.0006529997
## edu_cat2         .
## edu_cat3         0.0300337480
## edu_cat4         0.0271284749
## race_cat2        .
## race_cat3        0.0428406270
## race_cat4       -0.0004400906
## male1           -0.0198997627
## ageyrs          -0.0057148348
## yrssmoke         .
## smokenow1        .
## ln_lbxcot        0.0022435675
```

```r
## predictions
pred_lasso <- predict(cvfit_lasso,newx=X_test,  s="lambda.min")

## MSPE in test set
MSPE_lasso <- mean((pred_lasso-y_test)^2)






## RIDGE
## fit models
M_ridge <- glmnet(x=X_train,y=y_train,alpha = 0)

## plot paths
plot(M_ridge,xvar = "lambda",label=TRUE)
```



```r
## fit with crossval
cvfit_ridge <-  cv.glmnet(x=X_train,y=y_train,alpha = 0)

## plot MSPEs by lambda
plot(cvfit_ridge)
```



```
## estimated betas for minimum lambda
coef(cvfit_ridge, s = "lambda.min")## alternatively could use "lambda.1se"
```

```
## 37 x 1 sparse Matrix of class "dgCMatrix"
##                             1
## (Intercept)      1.398419e+00
## POP_PCB1        -3.167862e-07
## POP_PCB2        -1.881041e-07
## POP_PCB3         1.175129e-06
## POP_PCB4        -3.170587e-08
## POP_PCB5        -3.325161e-08
## POP_PCB6         9.761732e-08
## POP_PCB7        -5.783142e-07
## POP_PCB8        -4.333730e-07
## POP_PCB9         9.992878e-08
## POP_PCB10        4.588160e-04
## POP_PCB11        6.565258e-05
## POP_dioxin1     -9.777443e-05
## POP_dioxin2     -3.212980e-04
## POP_dioxin3     -1.000259e-05
## POP_furan1      -5.382392e-04
## POP_furan2       1.958387e-03
## POP_furan3       3.337740e-03
## POP_furan4      -6.422250e-05
## whitecell_count -6.669859e-03
## lymphocyte_pct   1.822139e-04
## monocyte_pct    -7.091461e-03
## eosinophils_pct  1.796237e-04
## basophils_pct   -1.583838e-05
## neutrophils_pct  5.394501e-03
## BMI             -1.612589e-03
## edu_cat2         2.224745e-02
## edu_cat3         5.528497e-02
## edu_cat4         5.718504e-02
## race_cat2       -1.182497e-02
## race_cat3        4.090029e-02
## race_cat4       -3.051308e-02
## male1           -3.038560e-02
## ageyrs          -4.210199e-03
## yrssmoke        -7.673946e-04
## smokenow1        1.321777e-03
## ln_lbxcot        6.010704e-03
```

```r
## predictions
pred_ridge <- predict(cvfit_ridge,newx=X_test,  s="lambda.min")

## MSPE in test set
MSPE_ridge <- mean((pred_ridge-y_test)^2)


## stepwise

M0 = lm(length~1, data=data.train)
Mfull = lm(length~., data=data.train)
Mstep <- step(object = M0,
              scope = list(lower = M0, upper = Mfull),
              direction = "both", trace = 1, k = 2)
```

```
## Start:  AIC=-1943.58
## length ~ 1
##
##                    Df Sum of Sq    RSS     AIC
## + ageyrs           1    8.1006 35.352 -2086.0
## + POP_dioxin2      1    2.5259 40.927 -1983.5
## + POP_PCB2         1    2.3184 41.135 -1980.0
## + POP_PCB1         1    2.2646 41.188 -1979.0
## + POP_PCB8         1    2.0272 41.426 -1975.0
## + POP_PCB7         1    1.9125 41.540 -1973.1
## + POP_PCB10        1    1.8958 41.557 -1972.8
## + POP_PCB5         1    1.7698 41.683 -1970.7
## + POP_PCB4         1    1.5900 41.863 -1967.7
## + POP_PCB9         1    1.5790 41.874 -1967.5
## + yrssmoke         1    1.2307 42.222 -1961.7
## + POP_dioxin1      1    1.1190 42.334 -1959.8
## + POP_dioxin3      1    0.9838 42.469 -1957.6
## + POP_furan1       1    0.9474 42.506 -1957.0
## + race_cat         3    1.1467 42.306 -1956.3
## + POP_furan3       1    0.8617 42.591 -1955.6
## + POP_PCB3         1    0.8509 42.602 -1955.4
## + POP_PCB6         1    0.8195 42.633 -1954.9
## + edu_cat          3    0.9666 42.486 -1953.3
## + ln_lbxcot        1    0.7157 42.737 -1953.2
## + monocyte_pct     1    0.6965 42.757 -1952.9
## + POP_furan2       1    0.6520 42.801 -1952.2
## + male             1    0.4558 42.997 -1949.0
## + smokenow         1    0.3435 43.109 -1947.1
## + POP_PCB11        1    0.3355 43.117 -1947.0
## + basophils_pct    1    0.1275 43.326 -1943.6
## <none>                         43.453 -1943.6
## + lymphocyte_pct   1    0.1189 43.334 -1943.5
## + BMI              1    0.1073 43.346 -1943.3
## + POP_furan4       1    0.0082 43.445 -1941.7
## + whitecell_count  1    0.0047 43.448 -1941.7
## + eosinophils_pct  1    0.0022 43.451 -1941.6
## + neutrophils_pct  1    0.0014 43.452 -1941.6
##
## Step:  AIC=-2086
## length ~ ageyrs
##
##                   Df Sum of Sq    RSS     AIC
## + POP_furan3       1    0.6348 34.718 -2096.7
## + race_cat         3    0.5707 34.782 -2091.4
## + POP_PCB10        1    0.3651 34.987 -2091.3
## + edu_cat          3    0.5171 34.835 -2090.3
## + POP_furan2       1    0.2625 35.090 -2089.2
## + POP_PCB3         1    0.2184 35.134 -2088.3
## + whitecell_count  1    0.1940 35.158 -2087.8
## + male             1    0.1935 35.159 -2087.8
## + POP_PCB5         1    0.1800 35.172 -2087.6
## + POP_PCB4         1    0.1769 35.176 -2087.5
## + POP_PCB11        1    0.1652 35.187 -2087.3
## + POP_PCB6         1    0.1534 35.199 -2087.0
## + POP_furan1       1    0.1528 35.200 -2087.0
## + POP_dioxin2      1    0.1495 35.203 -2087.0
## + POP_PCB9         1    0.1363 35.216 -2086.7
## + POP_PCB7         1    0.1181 35.234 -2086.3
## + BMI              1    0.1179 35.235 -2086.3
## <none>                         35.352 -2086.0
## + POP_PCB2         1    0.0989 35.254 -2086.0
## + monocyte_pct     1    0.0844 35.268 -2085.7
## + ln_lbxcot        1    0.0829 35.270 -2085.6
## + lymphocyte_pct   1    0.0645 35.288 -2085.3
## + POP_PCB1         1    0.0518 35.301 -2085.0
## + eosinophils_pct  1    0.0267 35.326 -2084.5
## + POP_PCB8         1    0.0166 35.336 -2084.3
## + neutrophils_pct  1    0.0142 35.338 -2084.3
## + POP_furan4       1    0.0111 35.341 -2084.2
## + yrssmoke         1    0.0110 35.341 -2084.2
## + smokenow         1    0.0062 35.346 -2084.1
## + POP_dioxin3      1    0.0028 35.350 -2084.1
## + basophils_pct    1    0.0011 35.351 -2084.0
## + POP_dioxin1      1    0.0003 35.352 -2084.0
## - ageyrs           1    8.1006 43.453 -1943.6
##
## Step:  AIC=-2096.68
## length ~ ageyrs + POP_furan3
##
##                  Df Sum of Sq    RSS     AIC
## + edu_cat         3    0.4625 34.255 -2100.1
## + race_cat        3    0.4447 34.273 -2099.7
## + whitecell_count  1    0.1585 34.559 -2097.9
## + male            1    0.1552 34.562 -2097.8
## + monocyte_pct    1    0.1038 34.614 -2096.8
## <none>                       34.718 -2096.7
## + ln_lbxcot       1    0.0916 34.626 -2096.5
## + BMI             1    0.0716 34.646 -2096.1
## + lymphocyte_pct  1    0.0579 34.660 -2095.8
## + POP_PCB3        1    0.0383 34.679 -2095.5
## + POP_dioxin1     1    0.0324 34.685 -2095.3
## + POP_PCB6        1    0.0211 34.697 -2095.1
## + eosinophils_pct  1    0.0204 34.697 -2095.1
## + POP_PCB10       1    0.0192 34.698 -2095.1
## + smokenow        1    0.0153 34.702 -2095.0
## + POP_PCB11       1    0.0140 34.704 -2095.0
## + POP_dioxin3     1    0.0133 34.704 -2094.9
## + POP_PCB4        1    0.0109 34.707 -2094.9
## + POP_dioxin2     1    0.0101 34.708 -2094.9
## + POP_furan4      1    0.0099 34.708 -2094.9
## + neutrophils_pct  1    0.0063 34.711 -2094.8
## + POP_PCB5        1    0.0059 34.712 -2094.8
## + POP_furan1      1    0.0057 34.712 -2094.8
## + POP_PCB1        1    0.0038 34.714 -2094.8
## + POP_PCB9        1    0.0021 34.715 -2094.7
## + POP_PCB8        1    0.0018 34.716 -2094.7
## + basophils_pct   1    0.0010 34.717 -2094.7
## + POP_PCB2        1    0.0007 34.717 -2094.7
## + POP_PCB7        1    0.0000 34.718 -2094.7
## + yrssmoke        1    0.0000 34.718 -2094.7
## + POP_furan2      1    0.0000 34.718 -2094.7
## - POP_furan3      1    0.6348 35.352 -2086.0
## - ageyrs          1    7.8737 42.591 -1955.6
##
## Step:  AIC=-2100.07
## length ~ ageyrs + POP_furan3 + edu_cat
##
##                  Df Sum of Sq    RSS     AIC
## + race_cat        3    0.5443 33.711 -2105.3
## + male            1    0.1706 34.084 -2101.6
## + ln_lbxcot       1    0.1657 34.089 -2101.5
## + whitecell_count  1    0.1331 34.122 -2100.8
## + monocyte_pct    1    0.1242 34.131 -2100.6
## <none>                       34.255 -2100.1
## + lymphocyte_pct  1    0.0941 34.161 -2100.0
## + POP_PCB3        1    0.0557 34.199 -2099.2
## + BMI             1    0.0556 34.199 -2099.2
## + smokenow        1    0.0408 34.214 -2098.9
## + eosinophils_pct  1    0.0384 34.217 -2098.9
## + POP_PCB6        1    0.0250 34.230 -2098.6
## + POP_PCB4        1    0.0197 34.235 -2098.5
## + POP_PCB11        1    0.0167 34.238 -2098.4
## + POP_PCB5         1    0.0097 34.245 -2098.3
## + POP_PCB9         1    0.0093 34.246 -2098.3
## + POP_dioxin1      1    0.0082 34.247 -2098.2
## + POP_PCB10        1    0.0059 34.249 -2098.2
## + POP_PCB1         1    0.0058 34.249 -2098.2
## + yrssmoke         1    0.0043 34.251 -2098.2
## + POP_furan2       1    0.0039 34.251 -2098.2
## + POP_dioxin2      1    0.0037 34.251 -2098.2
## + POP_PCB8         1    0.0025 34.253 -2098.1
## + POP_furan4       1    0.0018 34.253 -2098.1
## + neutrophils_pct  1    0.0017 34.253 -2098.1
## + POP_dioxin3      1    0.0005 34.255 -2098.1
## + basophils_pct    1    0.0004 34.255 -2098.1
## + POP_furan1       1    0.0002 34.255 -2098.1
## + POP_PCB2         1    0.0002 34.255 -2098.1
## + POP_PCB7         1    0.0001 34.255 -2098.1
## - edu_cat          3    0.4625 34.718 -2096.7
## - POP_furan3       1    0.5803 34.835 -2090.3
## - ageyrs           1    7.4000 41.655 -1965.2
##
## Step:  AIC=-2105.28
## length ~ ageyrs + POP_furan3 + edu_cat + race_cat
##
##                   Df Sum of Sq    RSS     AIC
## + male             1    0.1809 33.530 -2107.1
## + ln_lbxcot        1    0.1519 33.559 -2106.4
## + monocyte_pct     1    0.1507 33.560 -2106.4
## <none>                         33.711 -2105.3
## + smokenow         1    0.0677 33.643 -2104.7
## + BMI              1    0.0651 33.646 -2104.6
## + whitecell_count  1    0.0515 33.659 -2104.4
## + POP_PCB3         1    0.0316 33.679 -2103.9
## + POP_PCB1         1    0.0315 33.679 -2103.9
## + POP_dioxin2      1    0.0282 33.683 -2103.9
## + POP_furan4       1    0.0282 33.683 -2103.9
## + POP_dioxin1      1    0.0261 33.685 -2103.8
## + POP_furan1       1    0.0187 33.692 -2103.7
## + lymphocyte_pct   1    0.0161 33.695 -2103.6
## + POP_PCB8         1    0.0142 33.697 -2103.6
## + POP_PCB2         1    0.0138 33.697 -2103.6
## + POP_PCB6         1    0.0104 33.700 -2103.5
## + POP_dioxin3      1    0.0096 33.701 -2103.5
## + yrssmoke         1    0.0072 33.704 -2103.4
## + POP_PCB9         1    0.0052 33.706 -2103.4
## + POP_PCB11        1    0.0045 33.706 -2103.4
## + neutrophils_pct  1    0.0037 33.707 -2103.4
## + POP_furan2       1    0.0022 33.709 -2103.3
## + basophils_pct    1    0.0010 33.710 -2103.3
## + POP_PCB5         1    0.0009 33.710 -2103.3
## + POP_PCB4         1    0.0008 33.710 -2103.3
## + POP_PCB10        1    0.0006 33.710 -2103.3
## + eosinophils_pct  1    0.0006 33.710 -2103.3
## + POP_PCB7         1    0.0002 33.711 -2103.3
## - race_cat         3    0.5443 34.255 -2100.1
## - edu_cat          3    0.5621 34.273 -2099.7
## - POP_furan3       1    0.5014 34.212 -2096.9
## - ageyrs           1    6.5742 40.285 -1982.6
##
## Step:  AIC=-2107.05
## length ~ ageyrs + POP_furan3 + edu_cat + race_cat + male
##
##                   Df Sum of Sq    RSS     AIC
## + ln_lbxcot        1    0.2160 33.314 -2109.6
## <none>                          33.530 -2107.1
## + monocyte_pct     1    0.0947 33.435 -2107.0
## + smokenow         1    0.0809 33.449 -2106.7
## + BMI              1    0.0687 33.461 -2106.5
## + whitecell_count  1    0.0683 33.461 -2106.5
## + POP_dioxin1      1    0.0379 33.492 -2105.8
## + POP_dioxin3      1    0.0271 33.503 -2105.6
## + yrssmoke         1    0.0227 33.507 -2105.5
## + POP_PCB3         1    0.0223 33.508 -2105.5
## + POP_dioxin2      1    0.0212 33.509 -2105.5
## + POP_furan4       1    0.0152 33.515 -2105.4
## + POP_PCB1         1    0.0148 33.515 -2105.4
## + lymphocyte_pct   1    0.0144 33.515 -2105.3
## + POP_PCB10        1    0.0143 33.516 -2105.3
## - male             1    0.1809 33.711 -2105.3
## + POP_furan1       1    0.0110 33.519 -2105.3
## + POP_PCB7         1    0.0073 33.523 -2105.2
## + neutrophils_pct  1    0.0048 33.525 -2105.2
## + POP_PCB2         1    0.0039 33.526 -2105.1
## + POP_PCB6         1    0.0028 33.527 -2105.1
## + POP_PCB8         1    0.0025 33.527 -2105.1
## + eosinophils_pct  1    0.0024 33.527 -2105.1
## + POP_PCB9         1    0.0014 33.528 -2105.1
## + POP_PCB11        1    0.0012 33.529 -2105.1
## + POP_PCB4         1    0.0009 33.529 -2105.1
## + basophils_pct    1    0.0004 33.529 -2105.1
## + POP_furan2       1    0.0000 33.530 -2105.1
## + POP_PCB5         1    0.0000 33.530 -2105.1
## - race_cat         3    0.5546 34.084 -2101.6
## - edu_cat          3    0.5850 34.115 -2100.9
## - POP_furan3       1    0.4627 33.993 -2099.5
## - ageyrs           1    6.2900 39.820 -1988.7
##
## Step:  AIC=-2109.57
## length ~ ageyrs + POP_furan3 + edu_cat + race_cat + male + ln_lbxcot
##
##                   Df Sum of Sq    RSS     AIC
## + whitecell_count  1    0.1260 33.188 -2110.2
## <none>                          33.314 -2109.6
## + monocyte_pct     1    0.0908 33.223 -2109.5
## + BMI              1    0.0459 33.268 -2108.5
## + POP_dioxin2      1    0.0306 33.283 -2108.2
## + smokenow         1    0.0302 33.284 -2108.2
## + POP_PCB3         1    0.0262 33.288 -2108.1
## + POP_dioxin3      1    0.0244 33.289 -2108.1
## + POP_furan4       1    0.0224 33.291 -2108.0
## + POP_PCB1         1    0.0182 33.296 -2108.0
## + POP_dioxin1      1    0.0136 33.300 -2107.9
## + POP_furan1       1    0.0123 33.302 -2107.8
## + lymphocyte_pct   1    0.0112 33.303 -2107.8
## + POP_PCB10        1    0.0102 33.304 -2107.8
## + yrssmoke         1    0.0098 33.304 -2107.8
## + POP_PCB6         1    0.0069 33.307 -2107.7
## + POP_PCB2         1    0.0058 33.308 -2107.7
## + POP_PCB11        1    0.0052 33.309 -2107.7
## + POP_PCB7         1    0.0051 33.309 -2107.7
## + neutrophils_pct  1    0.0046 33.309 -2107.7
## + POP_PCB8         1    0.0046 33.309 -2107.7
## + POP_PCB9         1    0.0030 33.311 -2107.6
## + eosinophils_pct  1    0.0014 33.312 -2107.6
## + POP_PCB4         1    0.0010 33.313 -2107.6
## + basophils_pct    1    0.0004 33.313 -2107.6
## + POP_PCB5         1    0.0000 33.314 -2107.6
## + POP_furan2       1    0.0000 33.314 -2107.6
## - ln_lbxcot        1    0.2160 33.530 -2107.1
## - male             1    0.2450 33.559 -2106.4
## - race_cat         3    0.5435 33.857 -2104.2
## - POP_furan3       1    0.4918 33.806 -2101.3
## - edu_cat          3    0.7275 34.041 -2100.4
## - ageyrs           1    5.5940 38.908 -2002.9
##
## Step:  AIC=-2110.23
## length ~ ageyrs + POP_furan3 + edu_cat + race_cat + male + ln_lbxcot +
##     whitecell_count
##
##                    Df Sum of Sq    RSS     AIC
## + monocyte_pct     1    0.1843 33.004 -2112.1
## <none>                          33.188 -2110.2
## - whitecell_count  1    0.1260 33.314 -2109.6
## + POP_dioxin2      1    0.0339 33.154 -2108.9
## + BMI              1    0.0285 33.159 -2108.8
## + POP_PCB3         1    0.0279 33.160 -2108.8
## + POP_dioxin3      1    0.0240 33.164 -2108.7
## + POP_furan4       1    0.0232 33.165 -2108.7
## + smokenow         1    0.0227 33.165 -2108.7
## + POP_PCB1         1    0.0222 33.166 -2108.7
## + POP_dioxin1      1    0.0169 33.171 -2108.6
## + eosinophils_pct  1    0.0145 33.173 -2108.5
## + POP_furan1       1    0.0132 33.175 -2108.5
## + POP_PCB10        1    0.0097 33.178 -2108.4
## + POP_PCB6         1    0.0085 33.179 -2108.4
## + POP_PCB11        1    0.0080 33.180 -2108.4
## + POP_PCB8         1    0.0078 33.180 -2108.4
## + POP_PCB2         1    0.0077 33.180 -2108.4
## + neutrophils_pct  1    0.0057 33.182 -2108.3
## + POP_PCB7         1    0.0047 33.183 -2108.3
## + yrssmoke         1    0.0046 33.183 -2108.3
## + POP_PCB9         1    0.0043 33.184 -2108.3
## + POP_PCB4         1    0.0016 33.186 -2108.3
## + lymphocyte_pct   1    0.0007 33.187 -2108.2
## + POP_furan2       1    0.0004 33.187 -2108.2
## + POP_PCB5         1    0.0002 33.188 -2108.2
## + basophils_pct    1    0.0002 33.188 -2108.2
## - race_cat         3    0.4227 33.611 -2107.4
## - ln_lbxcot        1    0.2736 33.461 -2106.5
## - male             1    0.2819 33.470 -2106.3
## - POP_furan3       1    0.4723 33.660 -2102.3
## - edu_cat          3    0.6907 33.879 -2101.8
## - ageyrs           1    5.7106 38.898 -2001.1
##
## Step:  AIC=-2112.13
## length ~ ageyrs + POP_furan3 + edu_cat + race_cat + male + ln_lbxcot +
##     whitecell_count + monocyte_pct
##
##                    Df Sum of Sq    RSS     AIC
## <none>                          33.004 -2112.1
## + POP_dioxin2      1    0.0312 32.972 -2110.8
## + BMI              1    0.0311 32.972 -2110.8
## + POP_dioxin3      1    0.0266 32.977 -2110.7
## + POP_PCB3         1    0.0264 32.977 -2110.7
## + POP_PCB1         1    0.0195 32.984 -2110.5
## + POP_dioxin1      1    0.0186 32.985 -2110.5
## + POP_furan4       1    0.0184 32.985 -2110.5
## + smokenow         1    0.0184 32.985 -2110.5
## + POP_PCB10        1    0.0137 32.990 -2110.4
## + POP_furan1       1    0.0086 32.995 -2110.3
## + POP_PCB6         1    0.0084 32.995 -2110.3
## + POP_PCB11        1    0.0074 32.996 -2110.3
## + neutrophils_pct  1    0.0065 32.997 -2110.3
## + POP_PCB2         1    0.0061 32.997 -2110.3
## - monocyte_pct     1    0.1843 33.188 -2110.2
## + POP_PCB8         1    0.0048 32.999 -2110.2
## + POP_PCB9         1    0.0043 32.999 -2110.2
## + yrssmoke         1    0.0036 33.000 -2110.2
## + POP_PCB7         1    0.0033 33.000 -2110.2
## + POP_PCB4         1    0.0020 33.002 -2110.2
## + basophils_pct    1    0.0012 33.002 -2110.2
## + lymphocyte_pct   1    0.0009 33.003 -2110.1
## + eosinophils_pct  1    0.0002 33.003 -2110.1
## + POP_PCB5         1    0.0001 33.003 -2110.1
## + POP_furan2       1    0.0000 33.004 -2110.1
## - male             1    0.1983 33.202 -2109.9
## - race_cat         3    0.4099 33.413 -2109.5
## - whitecell_count  1    0.2195 33.223 -2109.5
## - ln_lbxcot        1    0.2938 33.297 -2107.9
## - POP_furan3       1    0.4891 33.493 -2103.8
## - edu_cat          3    0.7085 33.712 -2103.3
## - ageyrs           1    5.4747 38.478 -2006.7
```

```r
MSPE_step = mean((  predict(Mstep, newdata=data.test) - y_test)^2)

p = predict(Mstep, newdata=data.test)
```

```
cvfit_lasso$del
```

```
## NULL
```

```
MSPE_lasso
```

```
## [1] 0.05028345
```

```
MSPE_ridge
```

```
## [1] 0.05283856
```

```
MSPE_step
```

```
## [1] 0.05387623
```

models by automated selection makes little sense for interpretation

pollutants and bioinfo makes little sense and there are too many covariate

lets see if there is a smaller good model

say we try to fit with only 2 features

```
# lasso choose the same single variable
min(which((M_lasso$lambda)<=exp( -2.5)))
```

```
## [1] 4
```

```
coefs = M_lasso$beta[,4]
which(coefs!=0)
```

```
## ageyrs
##     33
```

```
library("plot3D")
```

```
## Warning: package 'plot3D' was built under R version 4.0.4
```

```
# 2 feature lasso choose
i = min(which((M_lasso$lambda)<=exp( -3.96)))
coefs = M_lasso$beta[,i]
choosen=which(coefs!=0)
coefs[choosen]
```

```
##     edu_cat3    race_cat3       ageyrs
##  0.002132031  0.022925033 -0.004863833
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.4
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
## Warning: package 'tibble' was built under R version 4.0.4
```

```
## Warning: package 'tidyr' was built under R version 4.0.4
## Warning: package 'readr' was built under R version 4.0.4
```

```
## Warning: package 'dplyr' was built under R version 4.0.4
```

```
## Warning: package 'forcats' was built under R version 4.0.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x dplyr::recode() masks car::recode()
## x purrr::some()   masks car::some()
## x tidyr::unpack() masks Matrix::unpack()
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.4
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

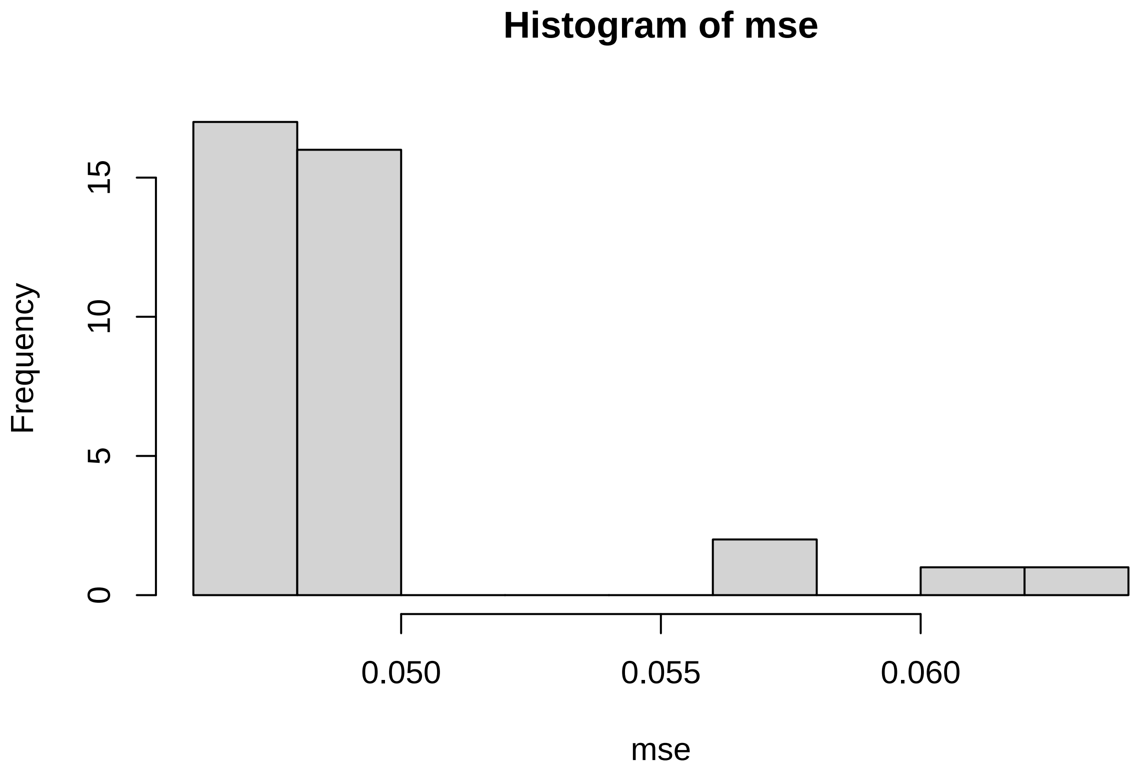
```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.0.4
```

```
models= regsubsets(length~., data=data, nvmax=2)
summary(models)
```

```
## Subset selection object
## Call: regsubsets.formula(length ~ ., data = data, nvmax = 2)
## 36 Variables  (and intercept)
##                  Forced in Forced out
## POP_PCB1             FALSE      FALSE
## POP_PCB2             FALSE      FALSE
## POP_PCB3             FALSE      FALSE
## POP_PCB4             FALSE      FALSE
## POP_PCB5             FALSE      FALSE
## POP_PCB6             FALSE      FALSE
## POP_PCB7             FALSE      FALSE
## POP_PCB8             FALSE      FALSE
## POP_PCB9             FALSE      FALSE
## POP_PCB10            FALSE      FALSE
## POP_PCB11            FALSE      FALSE
## POP_dioxin1          FALSE      FALSE
## POP_dioxin2          FALSE      FALSE
## POP_dioxin3          FALSE      FALSE
## POP_furan1           FALSE      FALSE
## POP_furan2           FALSE      FALSE
## POP_furan3           FALSE      FALSE
## POP_furan4           FALSE      FALSE
## whitecell_count      FALSE      FALSE
## lymphocyte_pct       FALSE      FALSE
## monocyte_pct        FALSE      FALSE
## eosinophils_pct     FALSE      FALSE
## basophils_pct       FALSE      FALSE
## neutrophils_pct     FALSE      FALSE
## BMI                 FALSE      FALSE
## edu_cat2            FALSE      FALSE
## edu_cat3            FALSE      FALSE
## edu_cat4            FALSE      FALSE
## race_cat2           FALSE      FALSE
## race_cat3           FALSE      FALSE
## race_cat4           FALSE      FALSE
## male1               FALSE      FALSE
## ageyrs              FALSE      FALSE
## yrssmoke            FALSE      FALSE
## smokenow1           FALSE      FALSE
## ln_lbxcot           FALSE      FALSE
## 1 subsets of each size up to 2
## Selection Algorithm: exhaustive
##          POP_PCB1 POP_PCB2 POP_PCB3 POP_PCB4 POP_PCB5 POP_PCB6 POP_PCB7
## 1  ( 1 ) " "      " "      " "      " "      " "      " "      " "
## 2  ( 1 ) " "      " "      " "      " "      " "      " "      " "
##          POP_PCB8 POP_PCB9 POP_PCB10 POP_PCB11 POP_dioxin1 POP_dioxin2
## 1  ( 1 ) " "      " "      " "       " "       " "         " "
## 2  ( 1 ) " "      " "      " "       " "       " "         " "
##          POP_dioxin3 POP_furan1 POP_furan2 POP_furan3 POP_furan4
## 1  ( 1 ) " "         " "        " "        " "        " "
## 2  ( 1 ) " "         " "        " "        "*"        " "
##          whitecell_count lymphocyte_pct monocyte_pct eosinophils_pct
## 1  ( 1 ) " "             " "            " "          " "
## 2  ( 1 ) " "             " "            " "          " "
##          basophils_pct neutrophils_pct BMI edu_cat2 edu_cat3 edu_cat4 race_cat2
## 1  ( 1 ) " "           " "             " " " "      " "      " "      " "
## 2  ( 1 ) " "           " "             " " " "      " "      " "      " "
##          race_cat3 race_cat4 male1 ageyrs yrssmoke smokenow1 ln_lbxcot
## 1  ( 1 ) " "       " "       " "   "*"    " "      " "       " "
## 2  ( 1 ) " "       " "       " "   "*"    " "      " "       " "
```

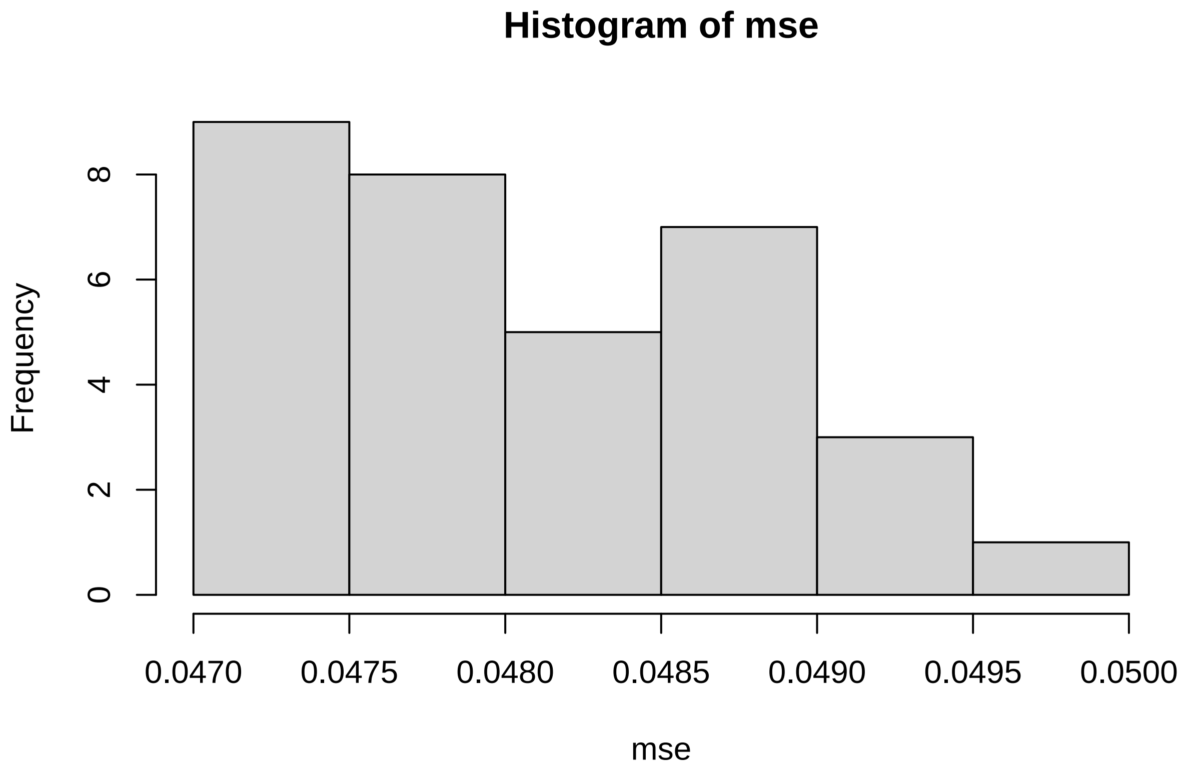
```
# rss of all 2 feature model, we see no magical model
mse = models$rss/nrow(data)
hist(mse)
```



```r
str(models)
```

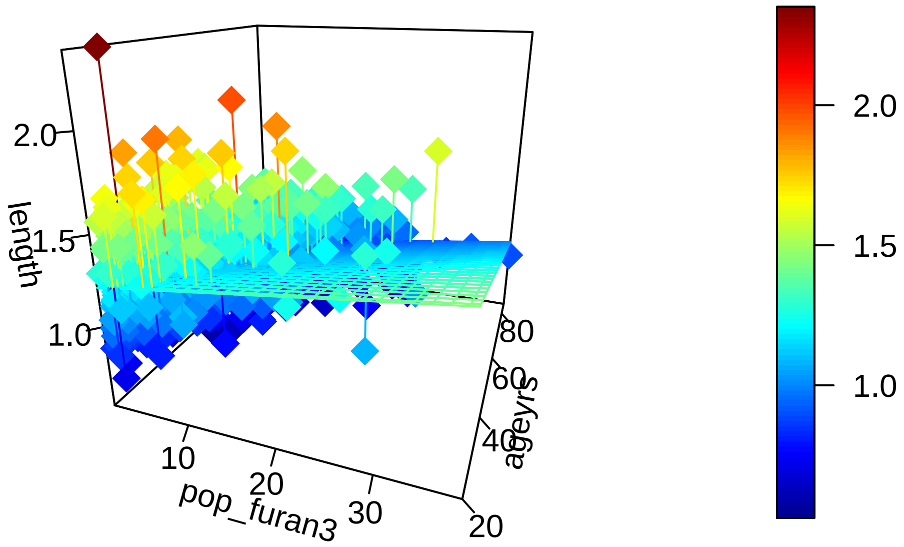
```
## List of 28
##  $ np       : int 37
##  $ nrbar    : int 666
##  $ d        : num [1:37] 864 205481 118 3859 201242 ...
##  $ rbar     : num [1:666] 10.604 0.231 -0.98 48.355 0.433 ...
##  $ thetab   : num [1:37] 1.05431 -0.00291 0.14662 0.01175 -0.00554 ...
##  $ first    : int 2
##  $ last     : int 37
##  $ vorder   : int [1:37] 1 35 36 37 34 33 32 31 30 29 ...
##  $ tol      : num [1:37] 1.47e-08 9.80e-07 2.43e-08 2.38e-07 1.55e-06 ...
##  $ rss      : num [1:37] 54 52.3 49.8 49.2 43.1 ...
##  $ bound    : num [1:37] 54 43.3 42.5 0 0 ...
##  $ nvmax    : int 3
##  $ ress     : num [1:3, 1] 54 43.3 42.5
##  $ ir       : int 3
##  $ nbest    : int 1
##  $ lopt     : int [1:6, 1] 1 1 34 1 18 34
##  $ il       : int 6
##  $ ier      : int 0
##  $ xnames   : chr [1:37] "(Intercept)" "POP_PCB1" "POP_PCB2" "POP_PCB3" ...
##  $ method   : chr "exhaustive"
##  $ force.in : Named logi [1:37] TRUE FALSE FALSE FALSE FALSE FALSE ...
##   ..- attr(*, "names")= chr [1:37] "" "POP_PCB1" "POP_PCB2" "POP_PCB3" ...
##  $ force.out: Named logi [1:37] FALSE FALSE FALSE FALSE FALSE FALSE ...
##   ..- attr(*, "names")= chr [1:37] "" "POP_PCB1" "POP_PCB2" "POP_PCB3" ...
##  $ sserr    : num 40.8
##  $ intercept: logi TRUE
##  $ lindep   : logi [1:37] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ nullrss  : num 54
##  $ nn       : int 864
##  $ call     : language regsubsets.formula(length ~ ., data = data, nvmax = 2)
##  - attr(*, "class")= chr "regsubsets"
```

```
# what about small mse in more detail, at least in terms of mse
mse=mse[mse<0.05]
hist(mse)
```

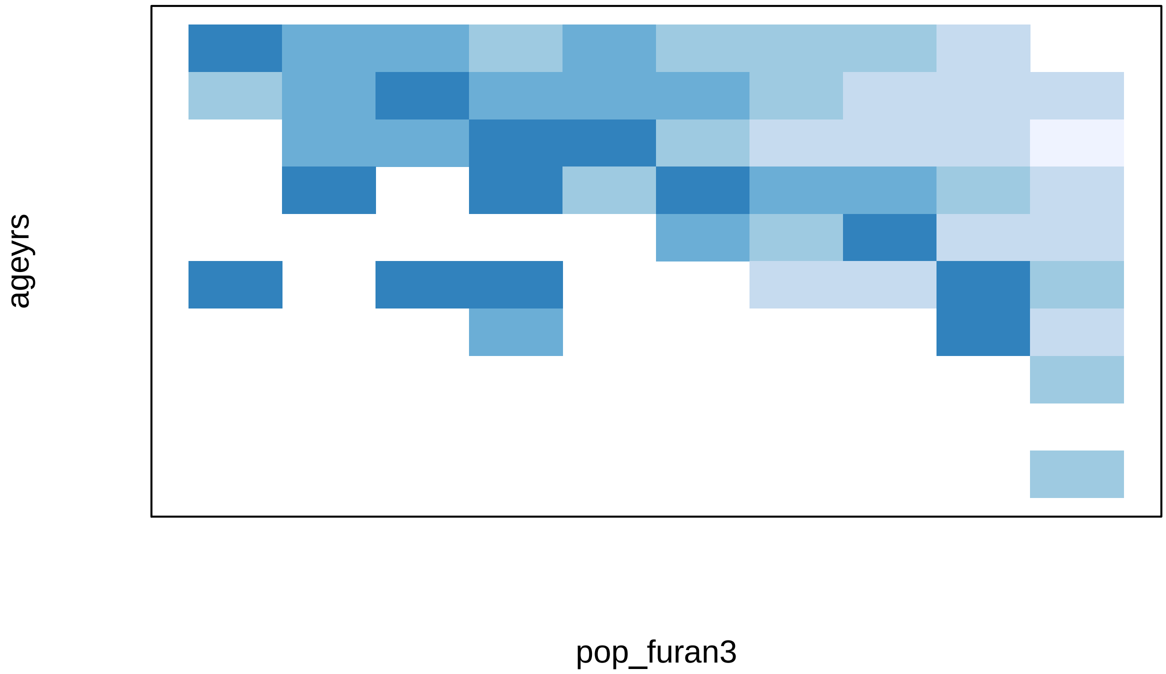


```
# what does the best 2 feature model look like?
z=data$length
y=data$ageyrs
x=data$POP_furan3

fit <- lm(z ~ x + y)
# predict values on regular xy grid
grid.lines = 26
x.pred <- seq(min(x), max(x), length.out = grid.lines)
y.pred <- seq(min(y), max(y), length.out = grid.lines)
xy <- expand.grid( x = x.pred, y = y.pred)
z.pred <- matrix(predict(fit, newdata = xy),
                 nrow = grid.lines, ncol = grid.lines)
# fitted points for droplines to surface
```

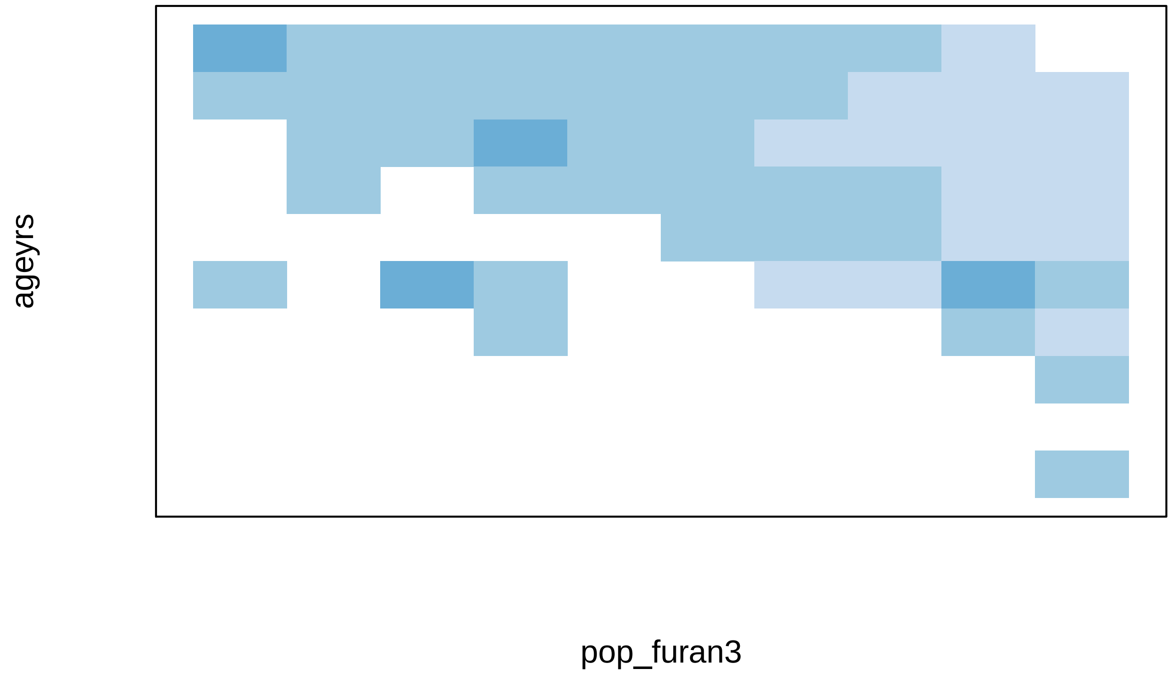
```
fitpoints = predict(fit)
# scatter plot with regression plane
scatter3D(x, y, z, pch = 18, cex = 2,
    theta = 20, phi = 20, ticktype = "detailed",
    surf = list(x = x.pred, y = y.pred, z = z.pred,
    facets = NA, fit = fitpoints), xlab="pop_furan3", ylab="ageyrs",zlab="length")
```



```
#turn ageyrs and pop_furan into grids

miny=min(y)
intervaly = (max(y)-miny)/10
minx=min(x)
intervalx = (max(x)-minx)/10

xy = matrix(0, nrow = 10, ncol = 10)
count = matrix(0, nrow = 10, ncol = 10)
for (i in 1:nrow(data)){
  xgrid = (x[i]-minx)/intervalx
  ygrid = (y[i]-miny)/intervaly
  count[xgrid,ygrid] = 1 + count[xgrid,ygrid]
  xy[xgrid,ygrid] = xy[xgrid, ygrid] + z[i]
}
xygrid = xy/count
col_areas(xygrid,xlab="pop_furan3", ylab="ageyrs")
```



```
maxz=max(z)
minz=min(z)
breaks = seq( minz, maxz, by=(maxz-minz)/5 )
col_areas(xygrid,xlab="pop_furan3", ylab="ageyrs", breaks = breaks)
```



```
# anyway how does this compare to the best fit?
```

4/4

perhaps pollutants is related to length we try it

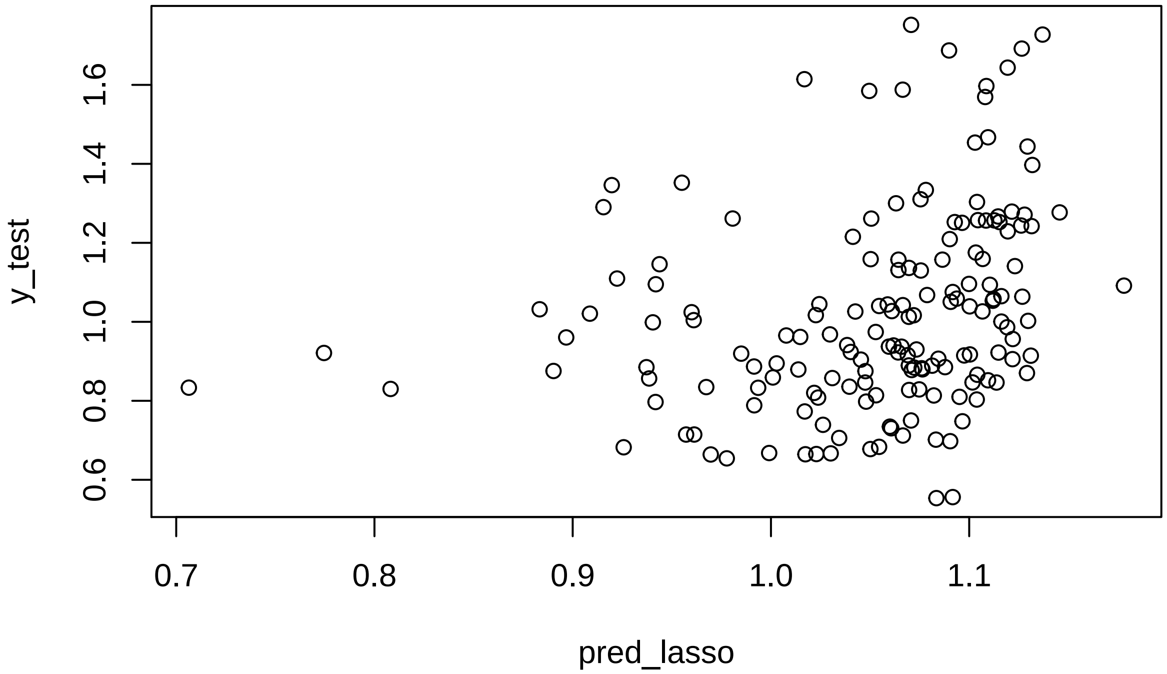
pollutants values are very large, we log transform it. and erroranalysis looks better

```
cols = colnames(data)
po.ind = str_detect(cols, "POP")

# this is to test transformation of data's result on lasso result
# limitation: transformation of other feature, some we cannot transform because they have value 0 or ne
lasso.on.pollutants =function(data){
  M = model.matrix(lm(length~., data=data))
  cols = colnames(M)
  po.ind = str_detect(cols, "POP")
  y_train = data$length[1:700]
  X_train = M[1:700,po.ind]
  y_test= data$length[701:nTotal]
  X_test= M[701:nTotal,(1:ncol(M))[po.ind]]



  M_lasso <- glmnet(x=X_train,y=y_train,alpha = 1)
  ## plot paths

  ## fit with crossval
  cvfit_lasso <-  cv.glmnet(x=X_train,y=y_train,alpha = 1)

  ## plot MSPEs by lambda

  ## estimated betas for minimum lambda

  ## predictions
  pred_lasso <- predict(cvfit_lasso,newx=X_test,  s="lambda.min")

  ## MSPE in test set
  MSPE_lasso <- mean((pred_lasso-y_test)^2)
  print(paste("mspe",MSPE_lasso) )

  plot(pred_lasso, y_test)

  return( coef(cvfit_lasso, s = "lambda.min"))

}



model = lasso.on.pollutants(data)
```

```
## [1] "mspe 0.0593195170954763"
```

```
#########################
newdata=data
model = lasso.on.pollutants(newdata)
```

```
## [1] "mspe 0.0593195170954763"
```



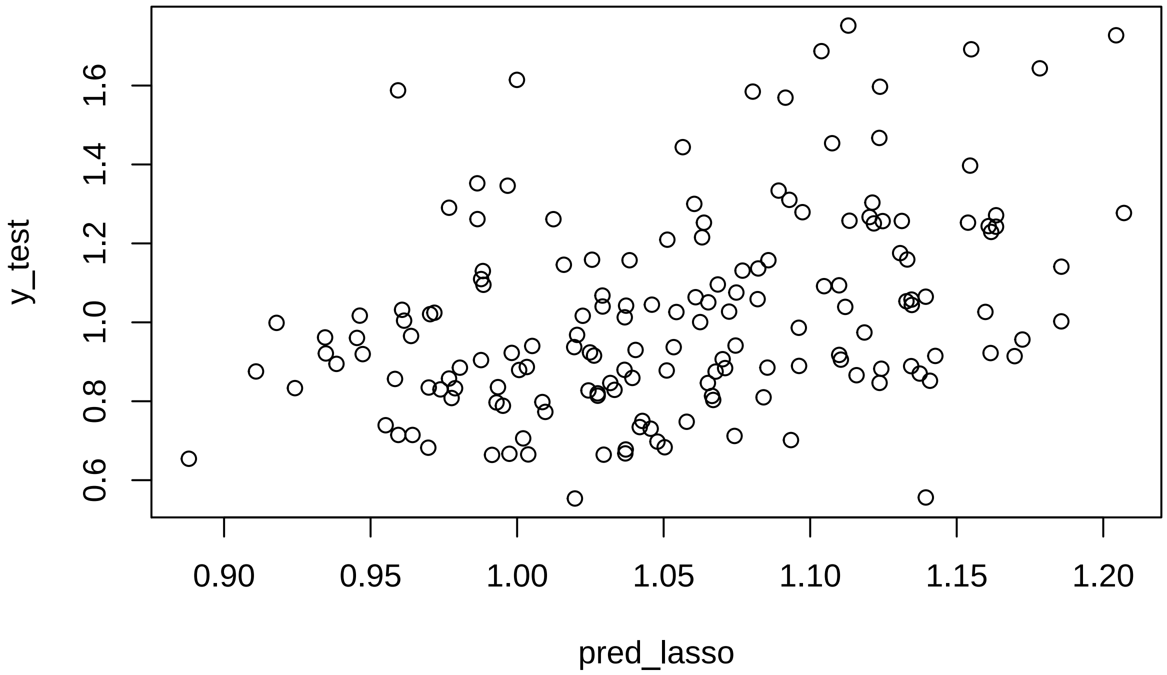
```
model
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept)  1.132304e+00
## POP_PCB1    -4.932827e-07
## POP_PCB2     .
## POP_PCB3     1.121184e-06
## POP_PCB4     .
## POP_PCB5     .
## POP_PCB6     .
## POP_PCB7    -1.276039e-06
## POP_PCB8    -1.218873e-06
## POP_PCB9     .
## POP_PCB10    .
## POP_PCB11    .
## POP_dioxin1 -1.983167e-04
## POP_dioxin2 -1.152378e-03
## POP_dioxin3 -4.213315e-06
## POP_furan1   .
## POP_furan2   8.488115e-04
## POP_furan3   2.835885e-03
## POP_furan4   8.332073e-04
```

```
# log transform
newdata = data
newdata[,po.ind] = log(data[,po.ind])
```

```
chosen.po.ind= which(lasso.on.pollutants(newdata)!=0)
```

```
## [1] "mspe 0.0558513952944322"
```



```
chosen.po.ind= chosen.po.ind[2:length(chosen.po.ind)]
```

we see effect of pollutants vary based on new features

new features are selected based on forward stepwise with 10 fold cv

we are not using built in because cv is usually better than aic bic

```
kfolds.cv <- function(dat, expr){
  kfolds=10
  mspe = rep(0, kfolds)
  ind = rep(1:kfolds, length=nrow(dat))
  for(ii in 1:kfolds) {
    train<- which(ind!=ii) # training observations
    M.cv <- lm(expr, data=data[train,])
    # cross-validation residuals
    M.res <- dat$length[-train] - # test observations
      predict(M.cv, newdat = dat[-train,]) # prediction with training dat
    # mspe
    mspe[ii] <- mean(M.res^2)
  }
  mean(mspe)
}
```

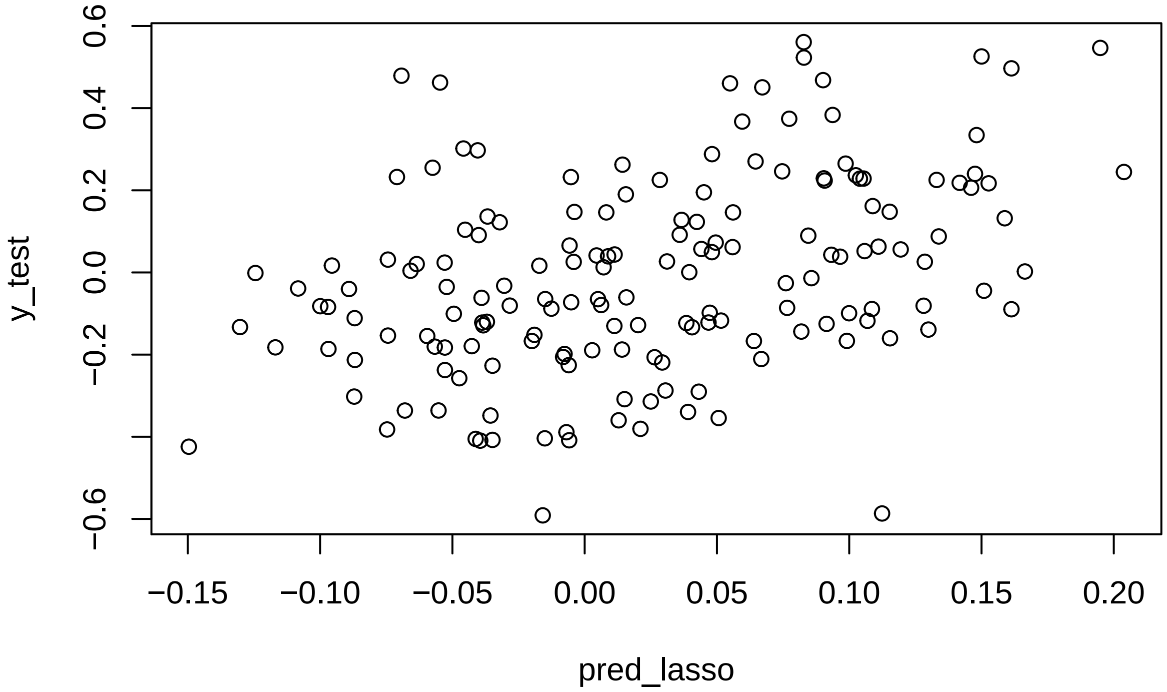
```r
# limits:
# only contains history of beta values of initial features
# forward selection won't remove already-added features

forward.change = function(data, expr, show=FALSE){
  model = lm(expr, data=newdata)
  initial.colname = names( model$coefficients)[-1]
  tempnames = colnames(data)
  cv.hist=c()
  aic.hist = c()
  coef.hist = list()
  model.hist=list()
  j=0
  models = list()
  while (TRUE) {
    j=j+1
    print(paste("step", j))
    cov.in.m = colnames(model$model)
    cov.all = colnames(newdata)
    names.to.try = cov.all[! cov.all %in% cov.in.m]
    nn = length(names.to.try)
    #update tracks
    cv.hist[j]=kfolds.cv(newdata, expr)
    aic.hist[j] = extractAIC(model)[2]
    coef.hist[[j]] = coef(model)
    model.hist[[j]] = model

    cv.score = rep(0, nn)
    if(length(names.to.try) == 0){
      print("chose all ")
      break
    }
    for (i in 1:nn) {
      name = names.to.try[i]
      newexpr =   paste(expr,  "+", name )
      newmodel = lm(newexpr, data=newdata)
      cv.score[i] = kfolds.cv(newdata, newexpr)
    }
    ind = which.min(cv.score)
    if(cv.score[ind]>cv.hist[j]){
      print ("done choosing model")
      break
    }else{
      # update our model
      print(paste("added", names.to.try[ind]))
      expr = paste(expr,"+", names.to.try[ind])
      model =  lm(expr, data=newdata)
      models[[j]] = model
    }
  }
  plot(cv.hist, main = "cv")
  plot(aic.hist, main = "aic")
  i = length(initial.colname)
  j = length(coef.hist)
  M = matrix(0, nrow = i, ncol = j)
  for (ii in 1:i){
    for (jj in 1:j) {
      M[ii,jj] =  coef.hist[[jj]][initial.colname[ii]]
    }
  }
  if(show==TRUE){
    par(cex=0.7)
    plot(M[1,], main=initial.colname[[1]], type = 'l', col=1, ylim = range(M))
    if(i!=1){
      for (a in 2:i){
        lines(1:j, M[a,] ,col=a)
      }
      legend("topright",legend = initial.colname, col = 1:i, pch=1)
    }
  }
  return(list(cv=cv.hist, coef=coef.hist, aic=aic.hist, model=model.hist))
}
```

we will analysis these

```
set.seed("24601")

## log transform
#newdata=data
#newdata[,po.ind] = log(newdata[,po.ind])
#
#
## forward start from length over pollutants
## limitation: we don't know about pollutants' meaning
#expr = paste("length~", paste(colnames(data)[po.ind] , collapse = "+"))
#
#forward.change(newdata, expr,TRUE)
#
## start from over ageyrs
#expr = "length~ageyrs"
#forward.change(newdata, expr, TRUE)
#
#
## start from chosen pollutens
#chosen.pos = colnames(data) [chosen.po.ind]
#expr = paste("length~", paste(chosen.pos, collapse = "+"))
#t=forward.change(newdata, expr, TRUE)
#
## the last step vary by a lot becasue large aif -> large variance on beta, we shouldn't consider last
#
#
## log transform on y

newdata = data
newdata[,po.ind] = log(data[,po.ind])
newdata[,1] = log(data[,1])
```

```
chosen.po.ind= which(lasso.on.pollutants(newdata)!=0)
```

```
## [1] "mspe 0.0493484151548836"
```

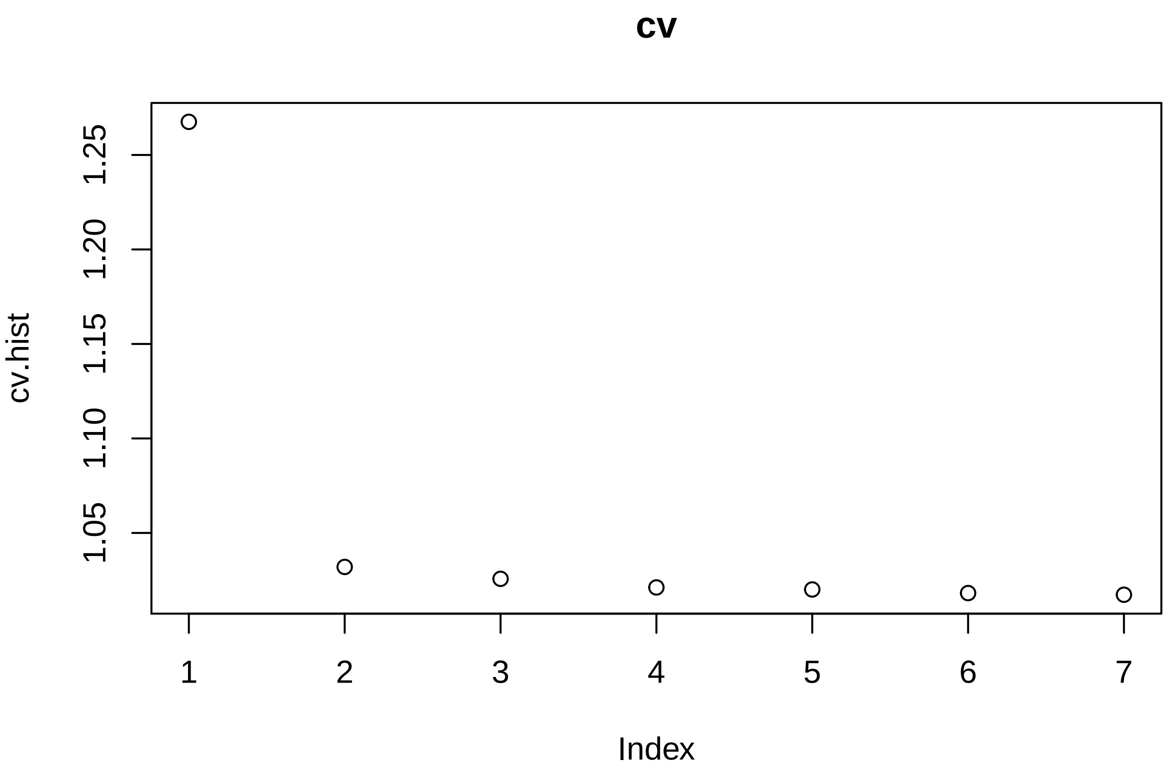


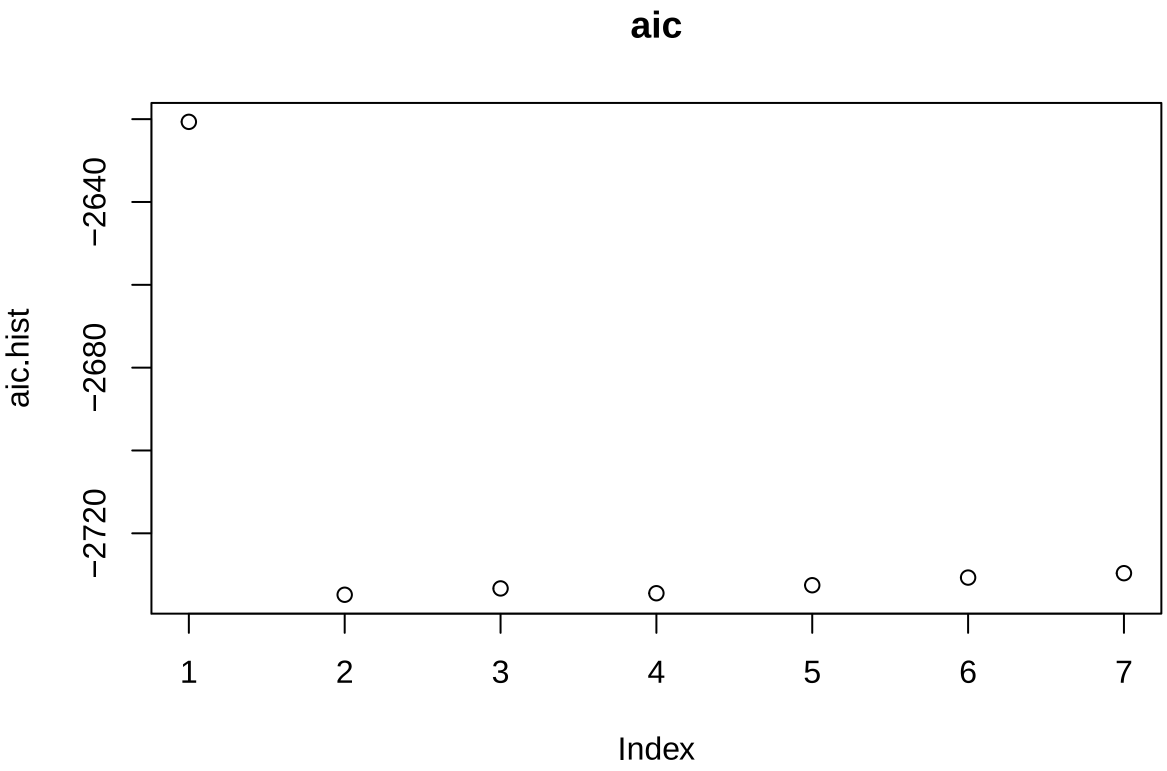
```
chosen.po.ind= chosen.po.ind[2:length(chosen.po.ind)]
print(chosen.po.ind)
```

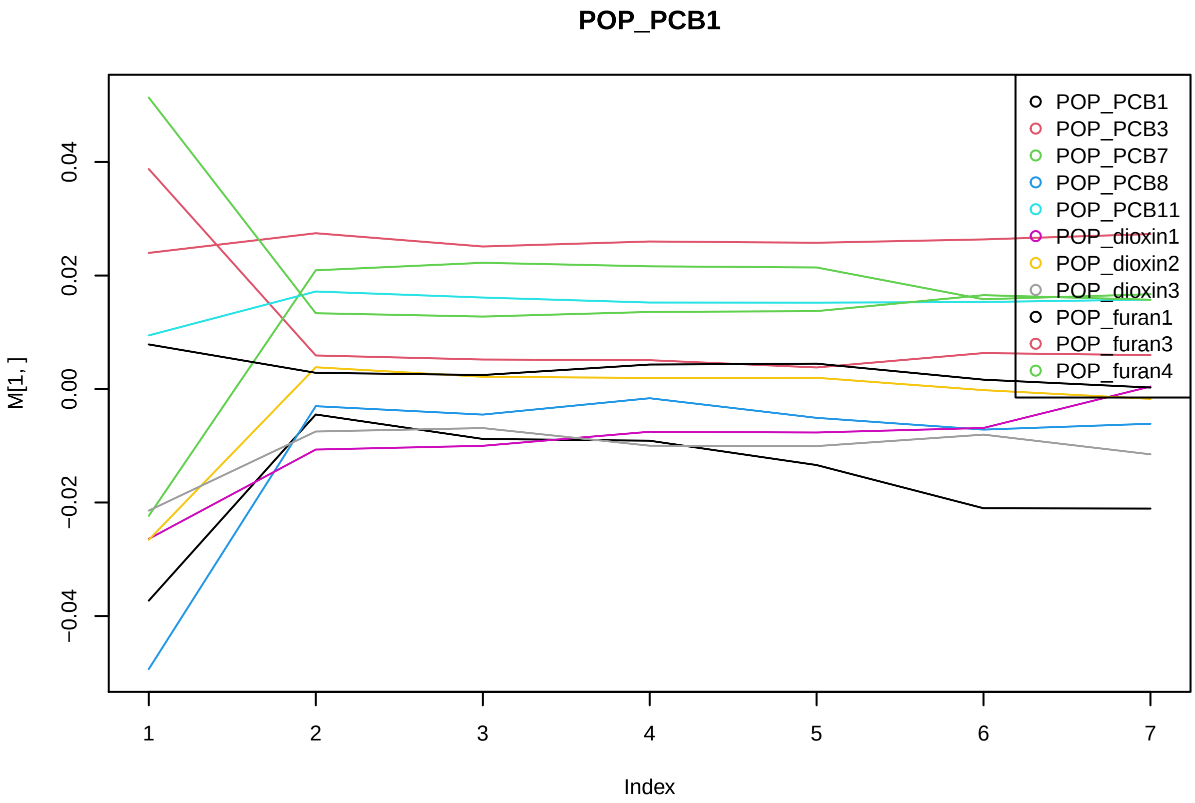
```
##  [1]  2  4  8  9 12 13 14 15 16 18 19
```

```
expr = paste("length~", paste(colnames(data)[chosen.po.ind] , collapse = "+"))
t=forward.change(newdata, expr, TRUE)
```

```
## [1] "step 1"
## [1] "added ageyrs"
## [1] "step 2"
## [1] "added POP_PCB10"
## [1] "step 3"
## [1] "added monocyte_pct"
## [1] "step 4"
## [1] "added POP_PCB2"
## [1] "step 5"
## [1] "added edu_cat"
## [1] "step 6"
## [1] "added smokenow"
## [1] "step 7"
## [1] "done choosing model"
```


cv
cv.hist
1.05
1.10
1.15
1.20
1.25
1
2
3
4
5
6
7
Index


aic
aic.hist
−2640
−2680
−2720
1
2
3
4
5
6
7
Index



```
# forward start from lm(length~1) done
# chosen pollute + other by forward done
# error analysis
# visualize the smoke stuff
# how the coefficients vary

newdata$length =log(newdata$length)

errorAnalysis(t$model[[2]])
t$model[[2]]

m = lm(length~1, data=newdata)
Mstep <- step(object = t$model[[2]],
              scope = list(lower = m, upper = t$model[[2]]),
              direction = "both", trace = 1, k = 2)


# path of the "error analysis stuff/ cook's distance dffits"
m = lm(length ~ POP_PCB7 + POP_PCB11 + POP_furan3 + ageyrs, data=newdata)

mean(m$residuals^2)
errorAnalysis(m)

mean(data$length)
mean( log(data$length))
plot(( m$fitted.values),log( data$length))
```

```r
abline(0,1)

y = rnorm(100)+10

sd(y)
y=y-4
sd(log(y))
m= t$model[[2]]

mean((data$length - exp(m$fitted.values))^2)
m=t$model[[4]]


m
mean((log(data$length) - (m$fitted.values))^2)
mean((data$length - exp(m$fitted.values))^2)
kfolds.cv(newdata,"length ~ POP_PCB7 + POP_PCB11 + POP_furan3 + ageyrs")

outliers()
plot.outliers(m)
jackknife.res(m, lev)
length(outliers(m))

plot.jackknife.res <- function(M){
  res <- resid(M) # raw residuals

  Xmat <- model.matrix(M) ## design matrix
  H <- Xmat%*%solve(t(Xmat)%*%Xmat)%*%t(Xmat) ## Hat matrix
  diag(H)
  lev <- hatvalues(M) ## leverage (h_i)
  hbar <- mean(lev) ## \bar{h}
  ids <- which(lev>2*hbar) ## x values for labelling points >2hbar
  n <- nobs(M)
  p <- length(attr(terms(M),"term.labels"))
  stud <- res/(sigma(M)*sqrt(1-lev)) # studentized residuals
  jack <- stud*sqrt((n-p-2)/(n-p-1-stud^2))
  plot(jack,ylab="Studentized Jackknife Residuals")
  points(jack[ids]~ids,col="red",pch=19) ## add high leverage points
  text(ids,jack[ids], labels=ids, cex= 0.6, pos=2) ## label points >2hbar
}



jackknife.res <- function(M){
  res <- resid(M) # raw residuals

  Xmat <- model.matrix(M) ## design matrix
  H <- Xmat%*%solve(t(Xmat)%*%Xmat)%*%t(Xmat) ## Hat matrix
  diag(H)
  lev <- hatvalues(M) ## leverage (h_i)
  hbar <- mean(lev) ## \bar{h}
  ids <- which(lev>2*hbar)
  return(ids)
}

plot.jackknife.res(m)

length(jackknife.res(m))
```