STAT 331 Final Project

Henry Xu, 20779975

09/04/2021

```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
```

```r
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
```

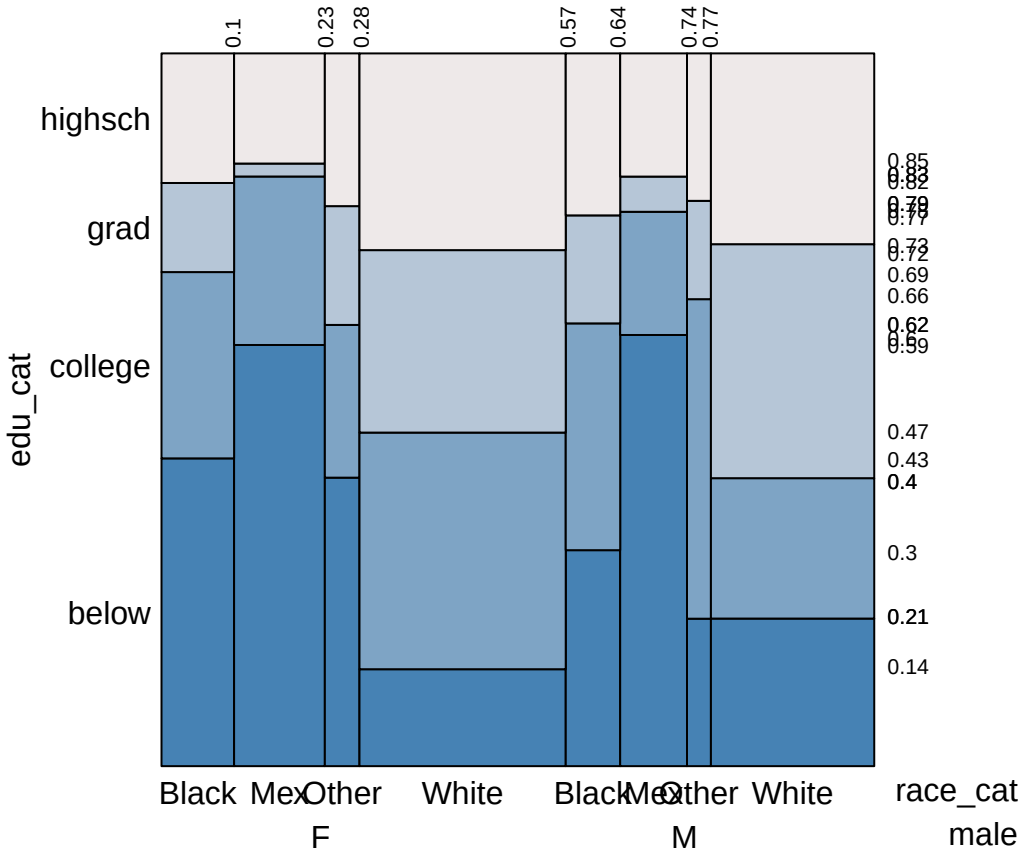
```
## Loading required package: rJava
```

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

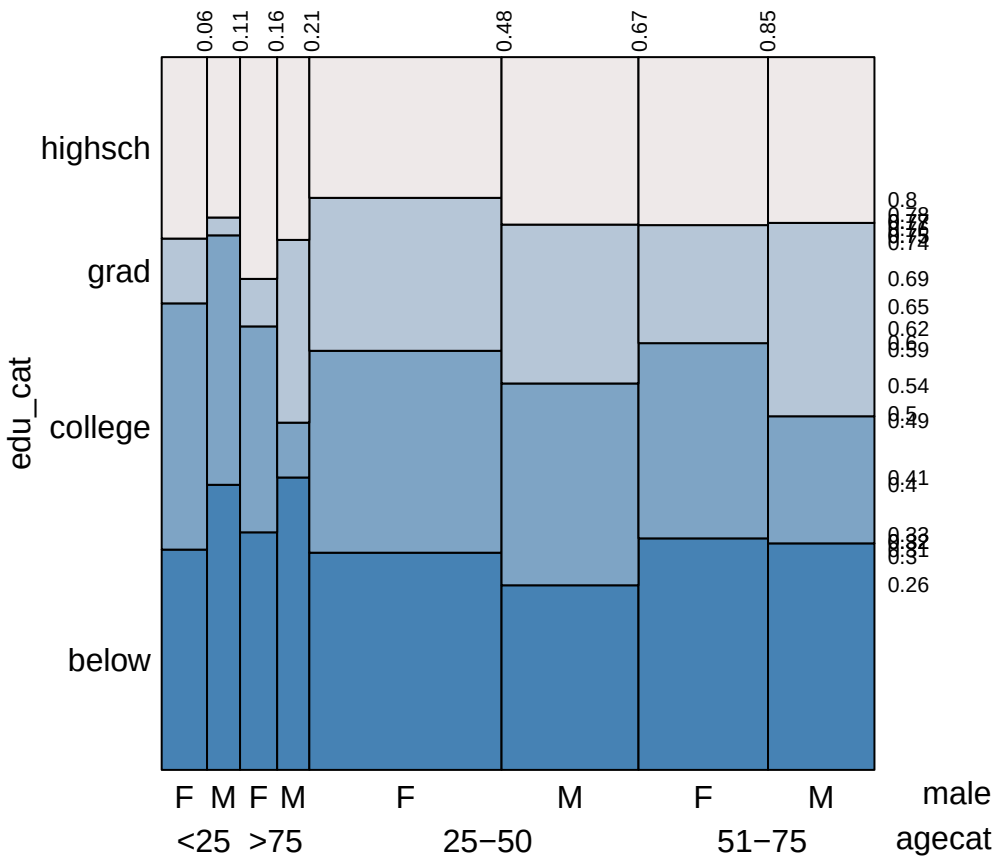
```
data = read.csv("pollutants.csv")


# change factor features to reasonable names

ind = data$male == 1
data$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)
```



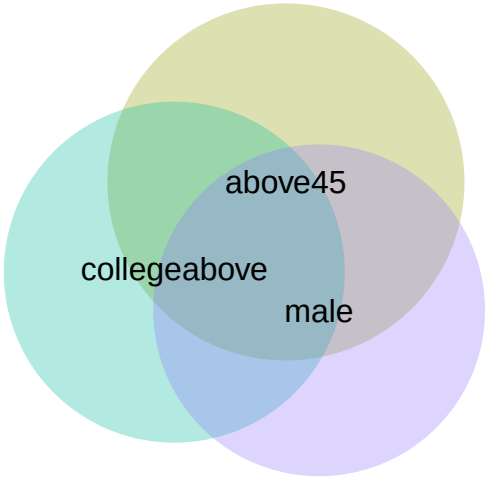

```
eikos(edu_cat~ male+agecat ,data=data)
```




```
# 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
```

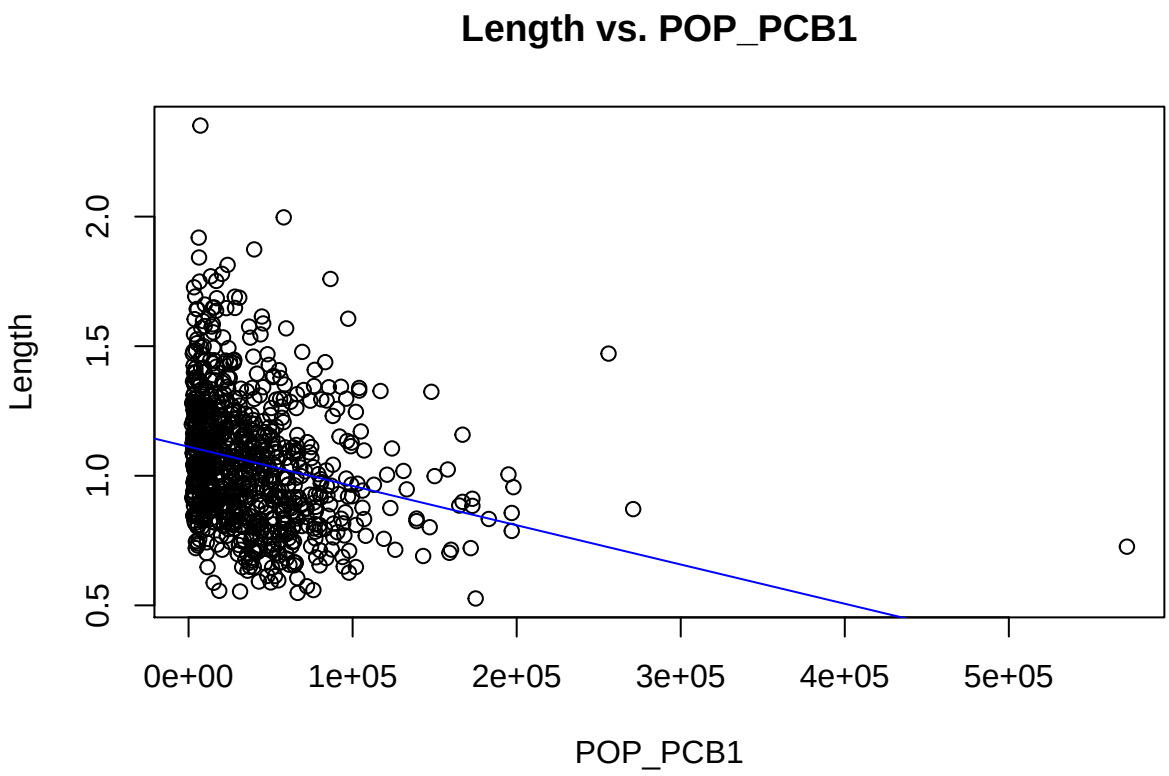
```
data = read.csv("pollutants.csv")


# 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)
```

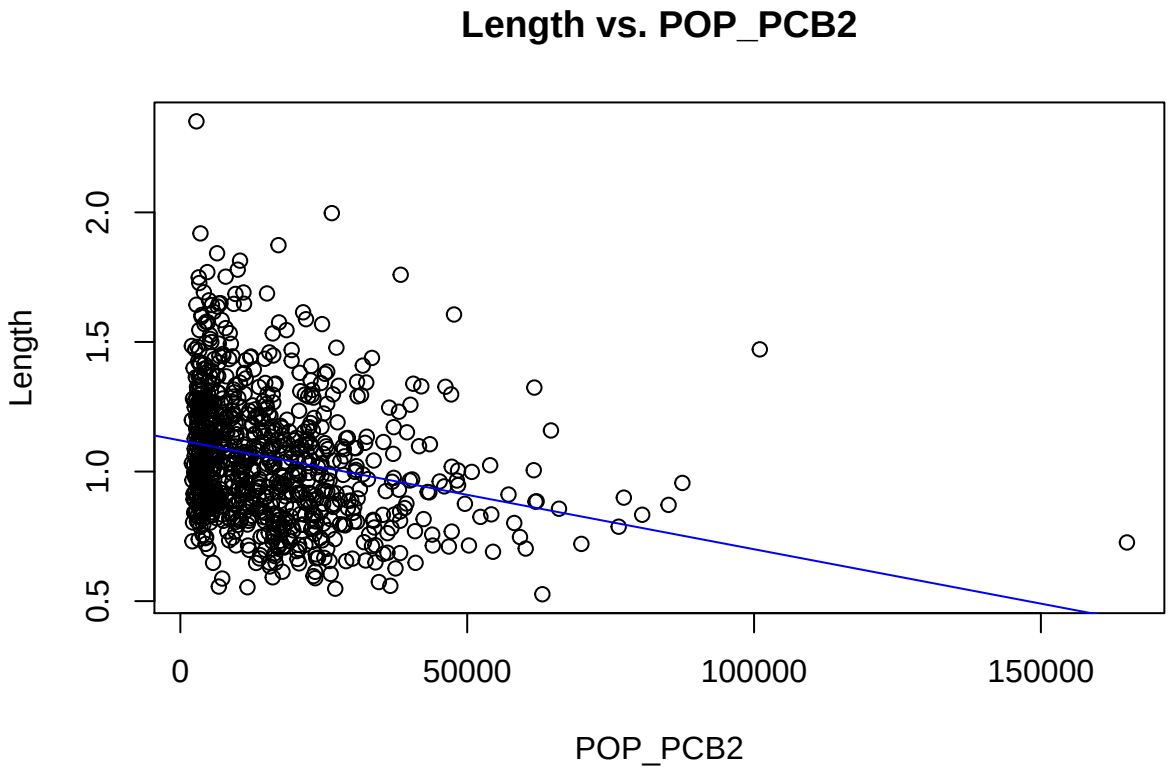
```r
data$edu_cat = factor(data$edu_cat)
data$male = factor(data$male)
data$smokenow= factor(data$smokenow)

summary.stats <- matrix(NA,nrow = ncol(data),ncol = 7)
cov.names <- colnames(data)
for(i in 1:ncol(data)){
  summary.stats[i,1] <- cov.names[i]
  summary.stats[i,2:(1+length(summary(data[,i])))] <- round(summary(data[,i]),2)
}

knitr::kable(summary.stats,caption = "Summary Statistics",
             col.names = c("Name", "Min.", "1st Qu.",
                           "Median", "Mean", "3rd Qu.","Max."))
```

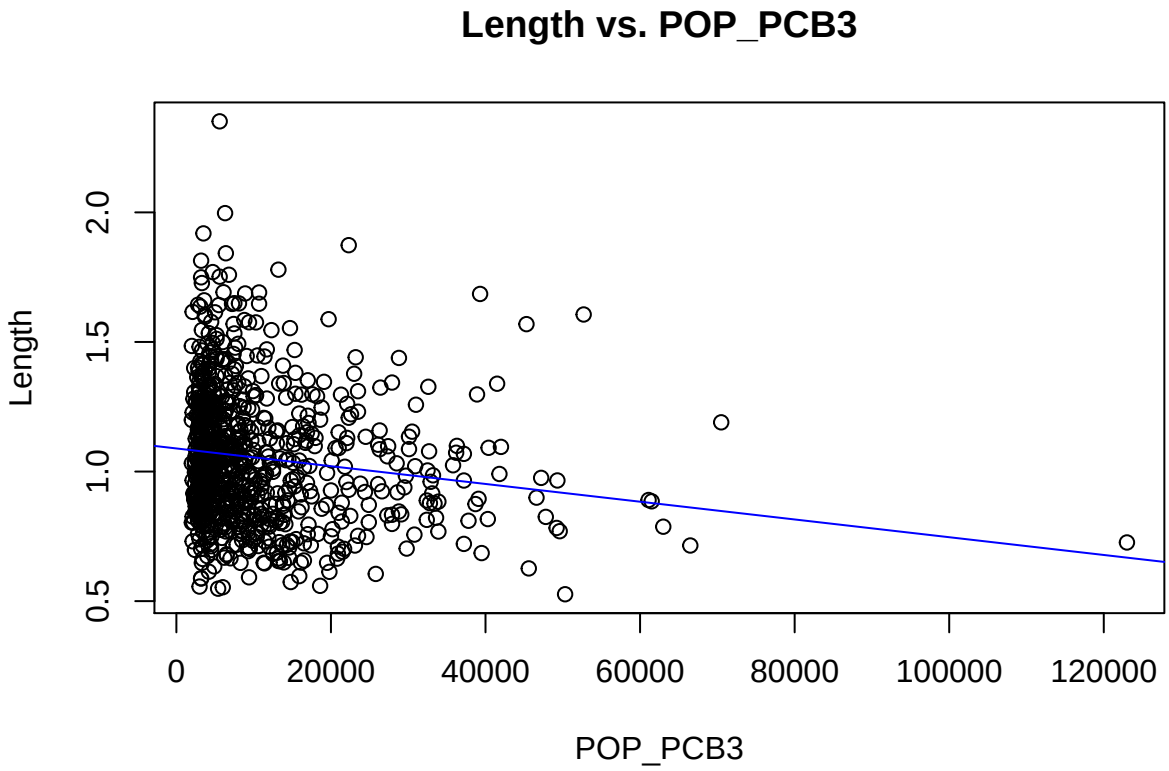
Table 1: Summary Statistics

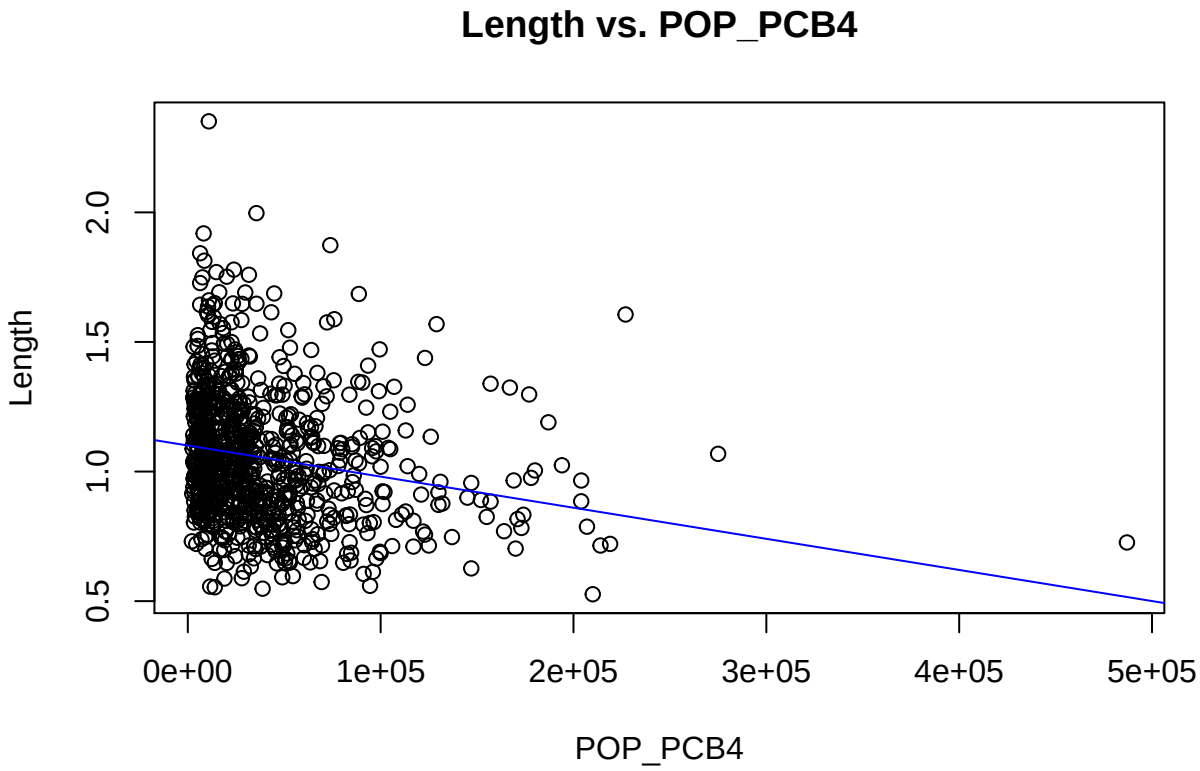
Name	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
length	0.53	0.88	1.03	1.05	1.21	2.35
POP_PCB1	2000	9975	27600	38082.18	53325	572000
POP_PCB2	2000	4800	11500	15636.81	21825	165000
POP_PCB3	2000	3700	6200	10157.75	12000	123000
POP_PCB4	2100	11475	25550	38455.79	50650	487000
POP_PCB5	2100	15600	36300	52650.23	68625	708000
POP_PCB6	2000	4400	9400	16820.02	19500	319000
POP_PCB7	1100	4000	7450	12681.94	15625	144000
POP_PCB8	1100	3800	6950	10529.75	14425	187000
POP_PCB9	1100	3900	8050	12220.25	16025	144000
POP_PCB10	1.7	9.1	18.35	24.49	34.9	172
POP_PCB11	1.3	14.8	24.5	38.15	42.95	845
POP_dioxin1	1.9	23.9	41.35	57.65	71.62	760
POP_dioxin2	1.4	21.28	37.8	47.81	62.42	281
POP_dioxin3	36.8	196.98	342.5	494.42	603	8190
POP_furan1	1	3.2	5.2	6.37	7.7	44.4
POP_furan2	0.8	2.6	4.2	5.39	6.82	33.5
POP_furan3	0.7	2.2	5.05	6.67	9.3	38.3
POP_furan4	0.9	6.4	9.65	11.54	14	234
whitecell_count	2.3	5.6	6.9	7.19	8.3	20.1
lymphocyte_pct	5.8	24	28.95	29.92	35.42	73.4
monocyte_pct	1.6	6.6	7.7	7.94	9.1	23.8
eosinophils_pct	21.6	52.35	59.3	58.62	65.23	88.1
basophils_pct	0	1.5	2.3	2.9	3.7	28.2
neutrophils_pct	0	0.4	0.6	0.67	0.8	5.5
BMI	16.16	23.88	27.38	28.09	31.17	62.99
edu_cat	270	199	228	167	NA	NA
race_cat	71	191	154	448	NA	NA
male	490	374	NA	NA	NA	NA
ageyrs	20	34	46	48.36	63	85
yrssmoke	0	0	0	10.6	20	69
smokenow	664	200	NA	NA	NA	NA
ln_lbxcot	-4.51	-4.07	-2.73	-0.98	2.8	6.58

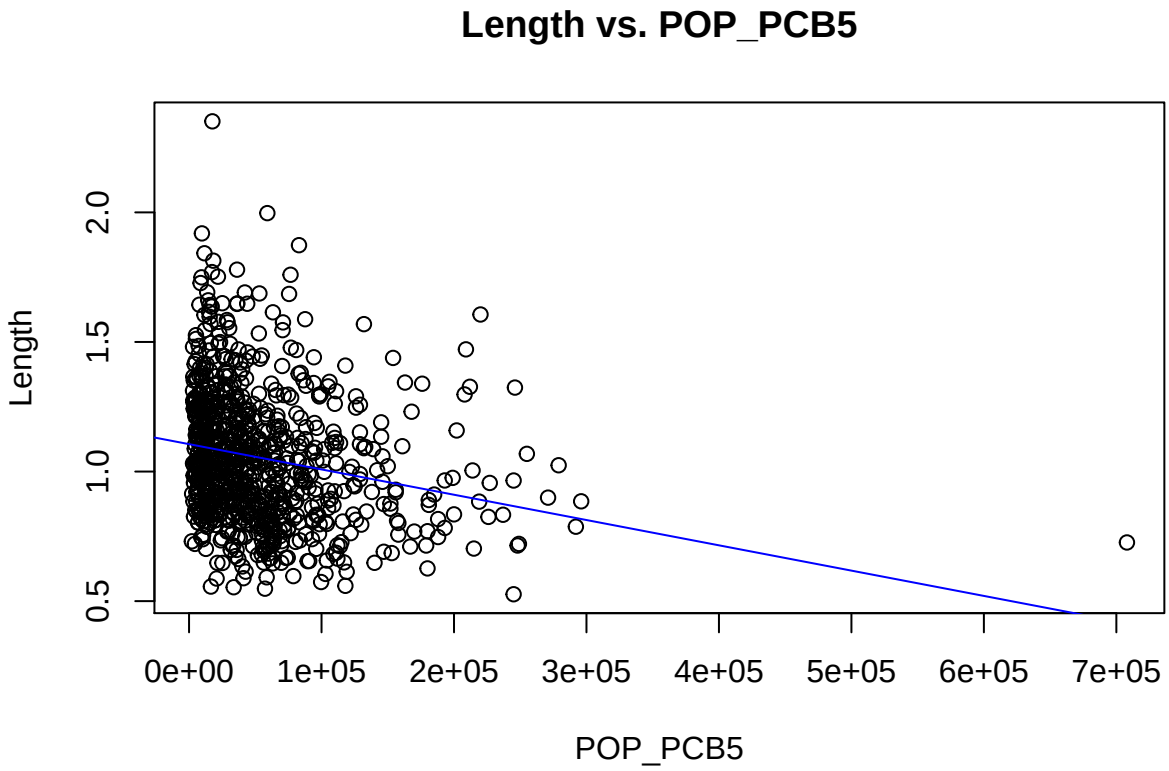
```r
for(i in 1:length(cov.names[-1])){
  temp.model <- lm(paste0("length ~ ",cov.names[i+1]),data = data)
  plot(data[,cov.names[i+1]],data$length, main = paste0("Length vs. ",cov.names[i+1]),
       ylab = "Length", xlab = cov.names[i+1])
  abline(temp.model,col = "blue")
}
```

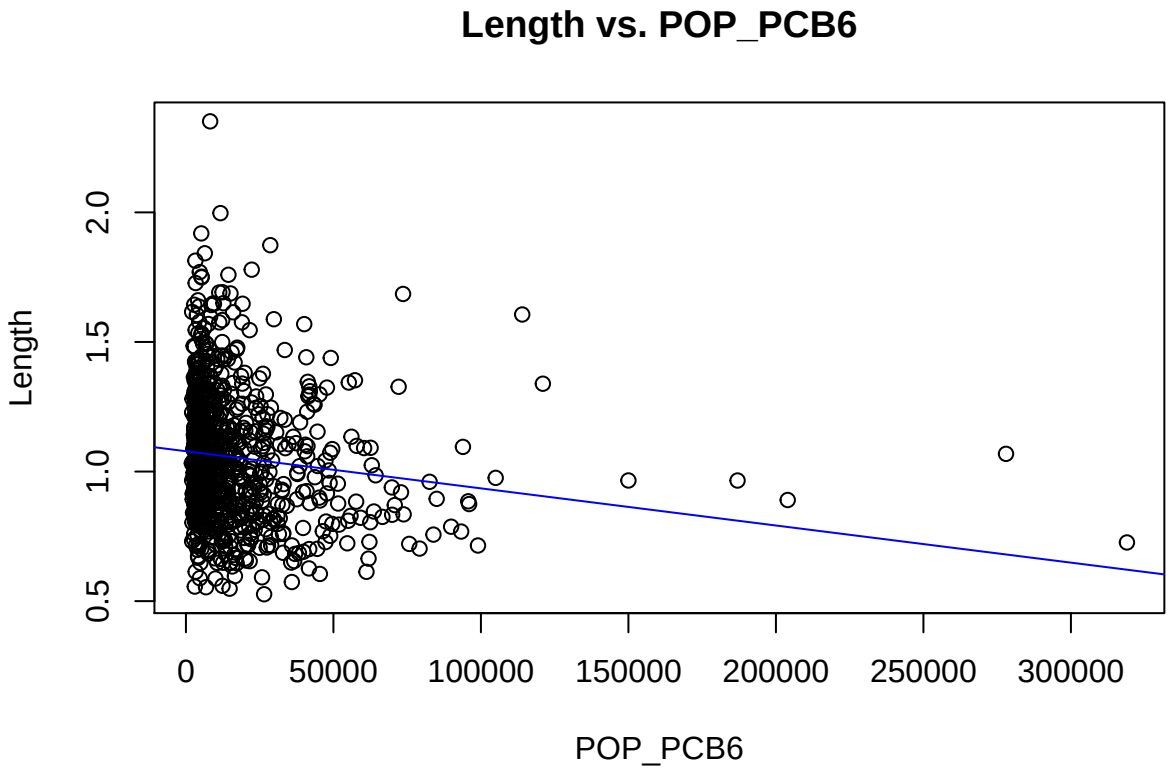


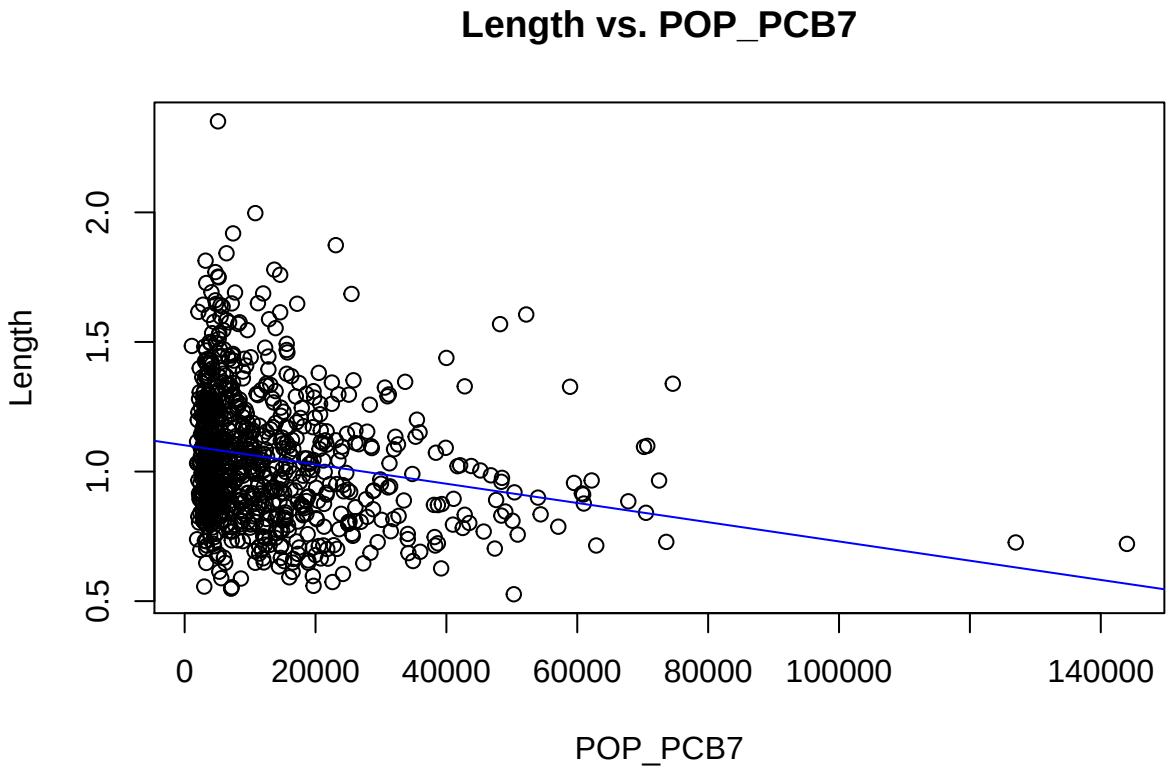

Length vs. POP_PCB2
Length
0.5
1.0
1.5
2.0
0
50000
100000
150000
POP_PCB2

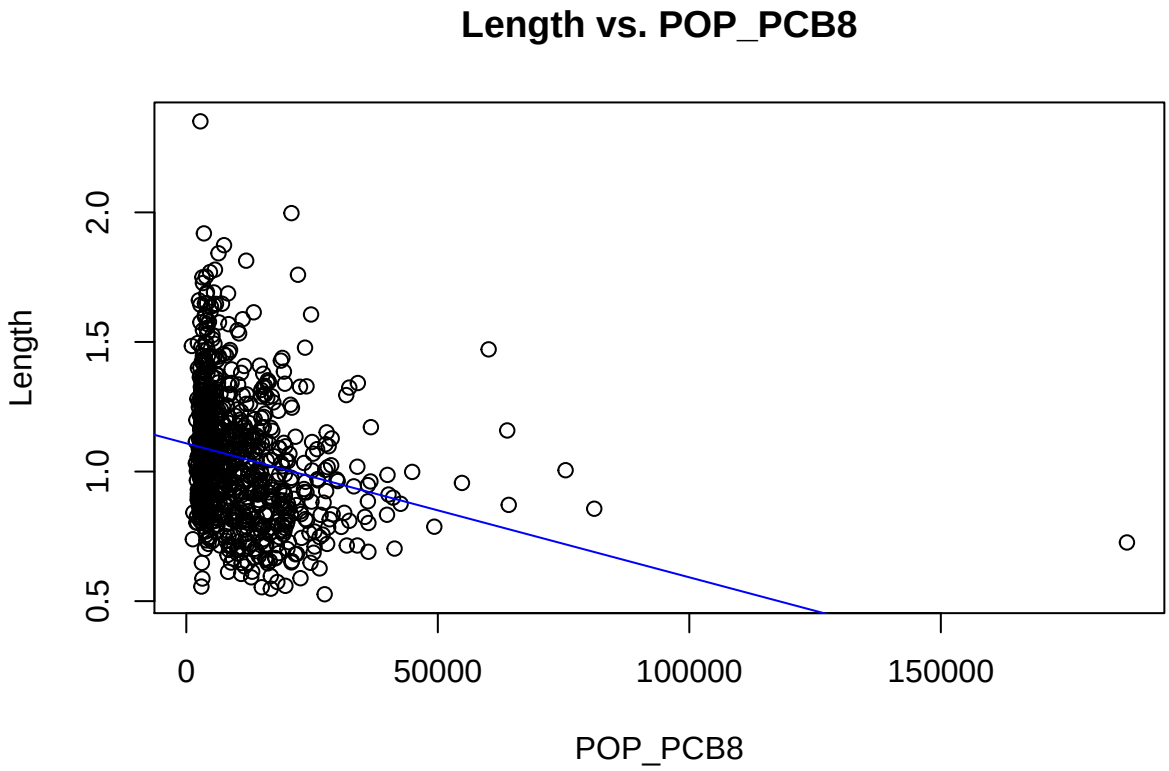

Length vs. POP_PCB3
Length
0.5
1.0
1.5
2.0
0
20000
40000
60000
80000
100000
120000
POP_PCB3

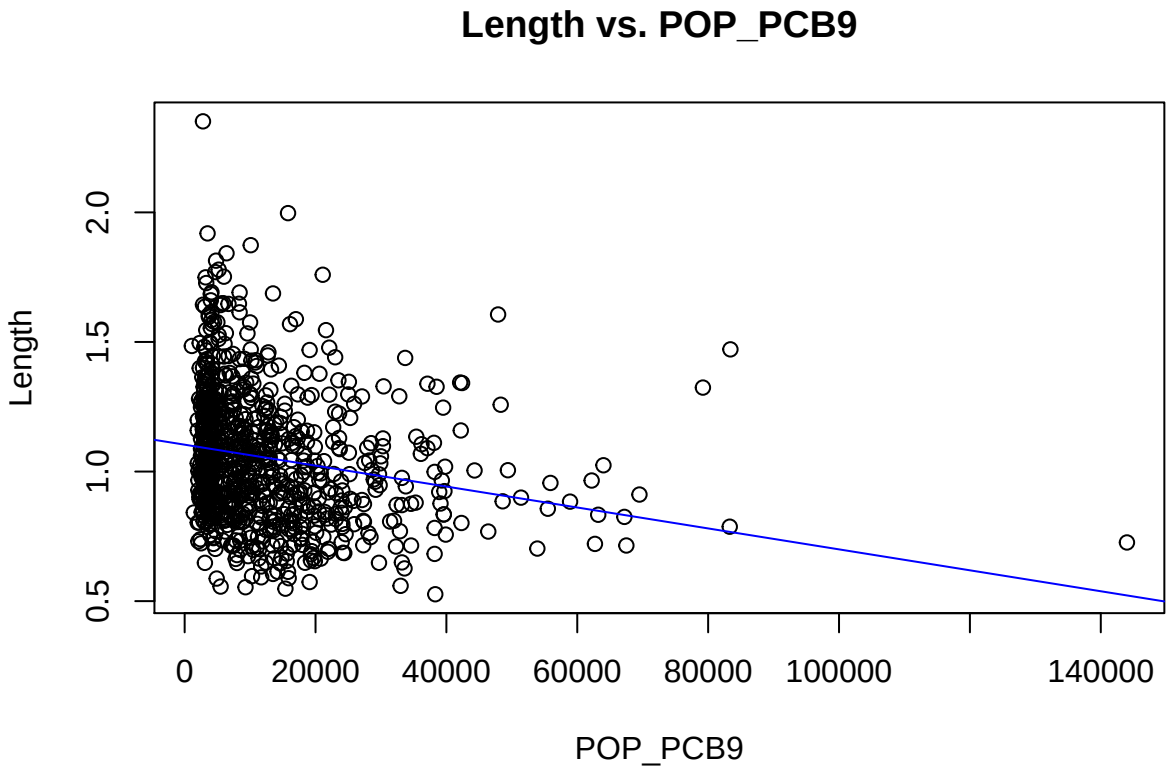

Length vs. POP_PCB4
Length
0.5
1.0
1.5
2.0
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
POP_PCB4

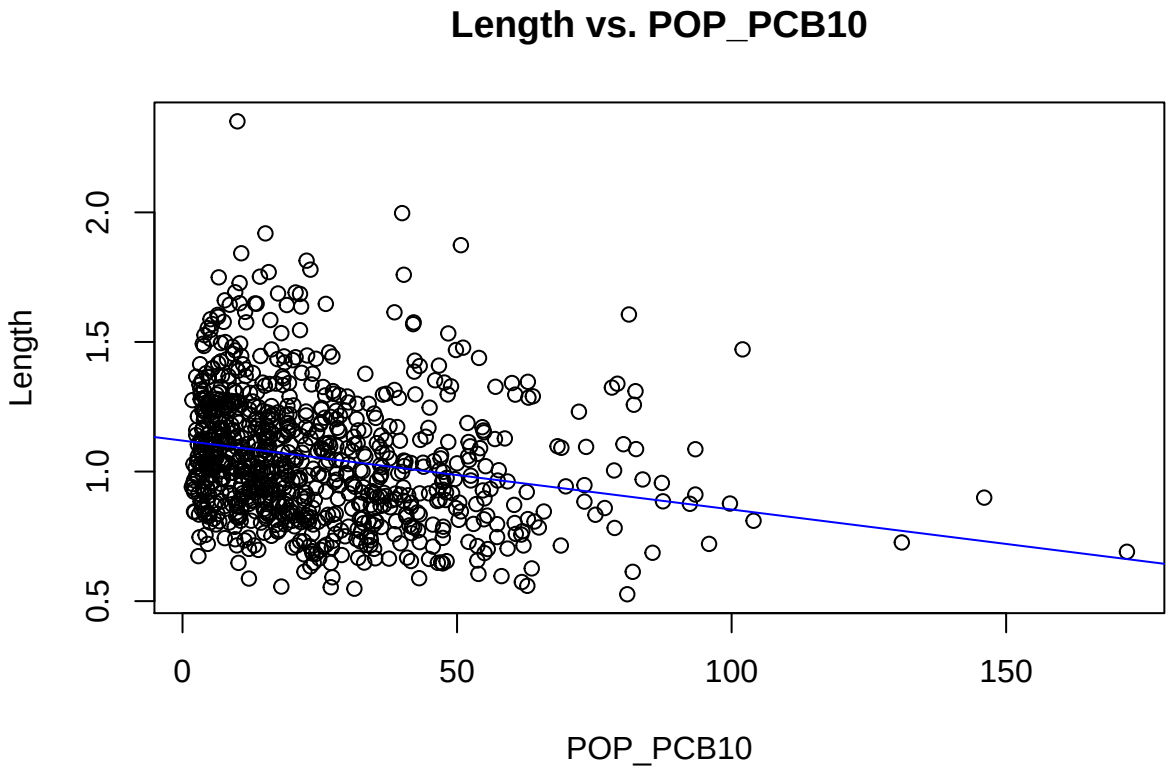

Length vs. POP_PCB5
Length
0.5
1.0
1.5
2.0
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
POP_PCB5

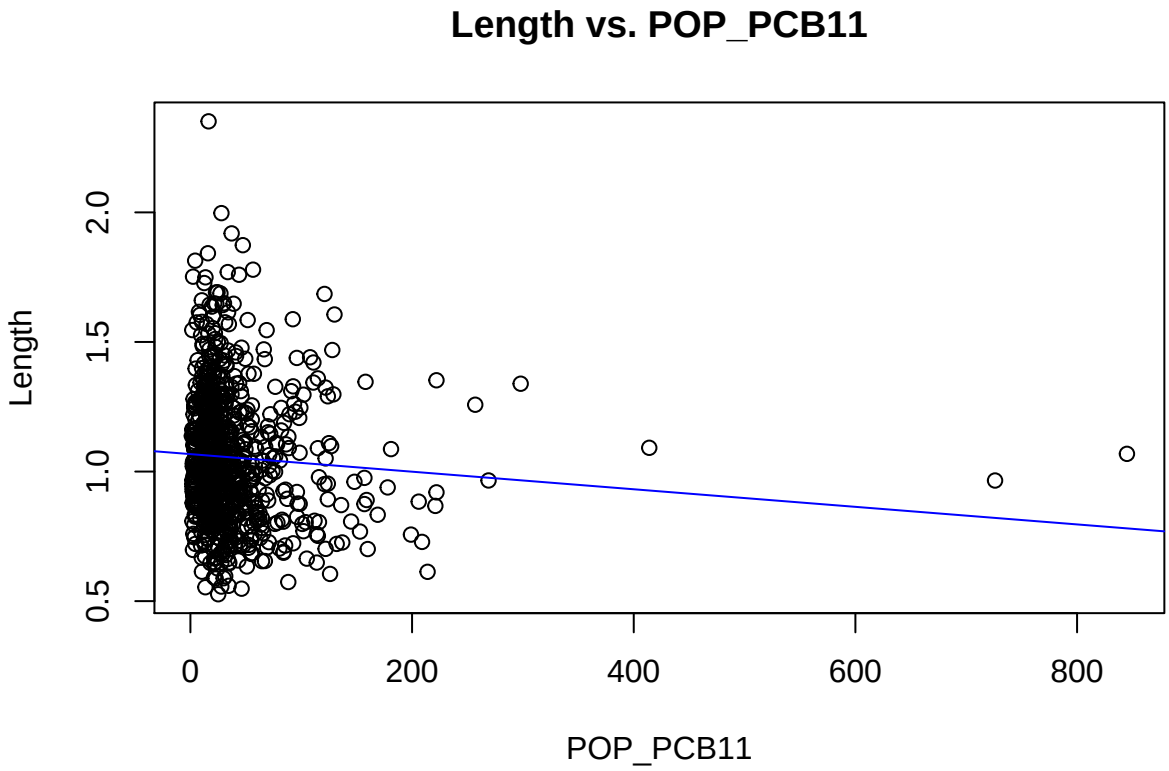

Length vs. POP_PCB6
Length
0.5
1.0
1.5
2.0
0
50000
100000
150000
200000
250000
300000
POP_PCB6

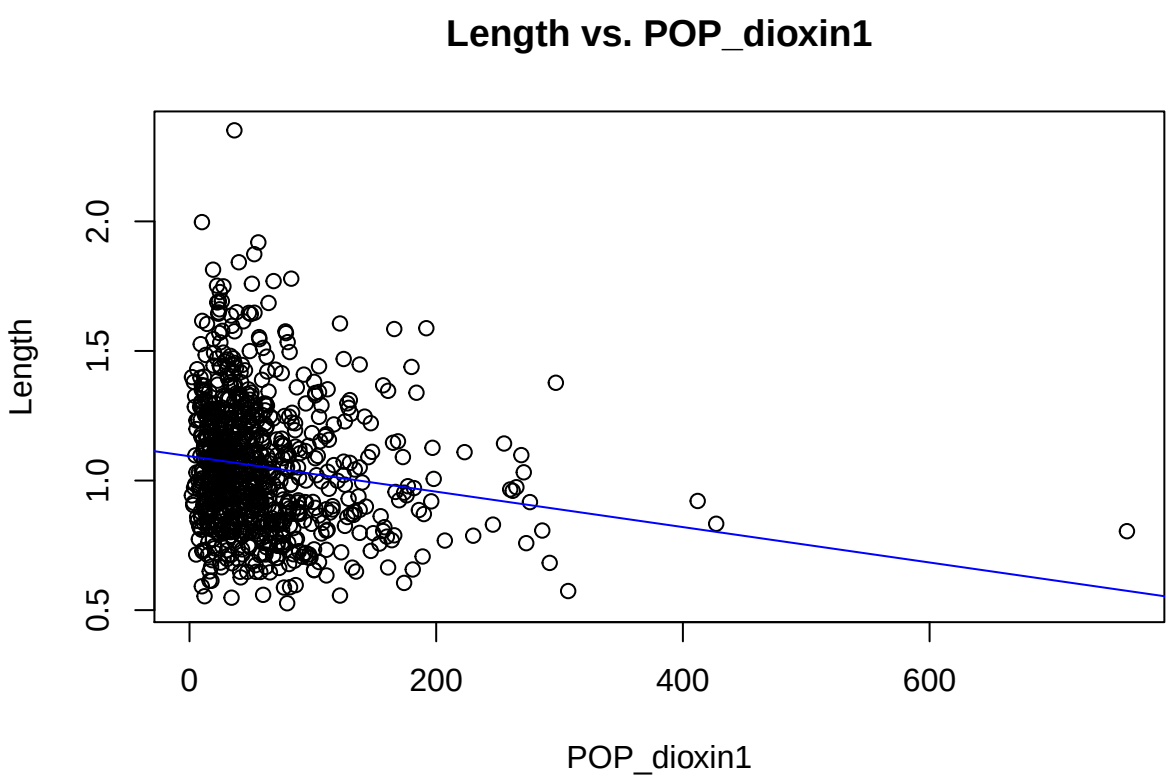

Length vs. POP_PCB7
Length
0.5
1.0
1.5
2.0
0
20000
40000
60000
80000
100000
140000
POP_PCB7

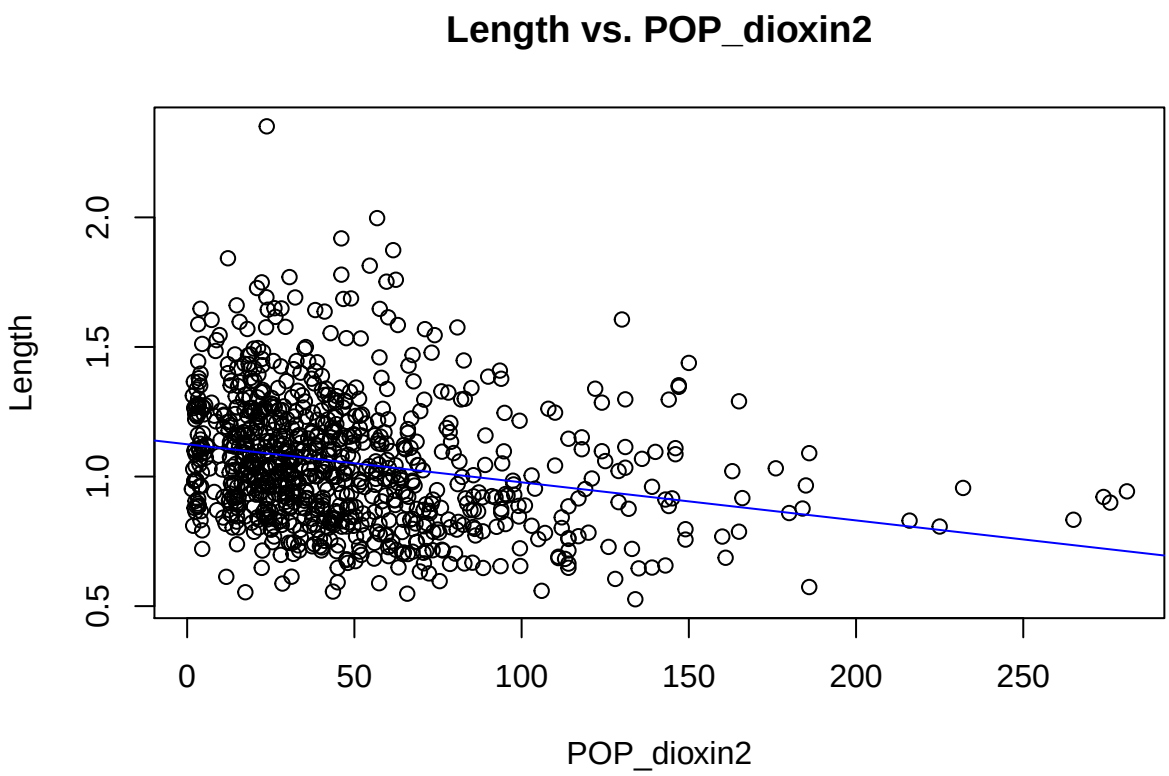

Length vs. POP_PCB8
Length
0.5
1.0
1.5
2.0
0
50000
100000
150000
POP_PCB8

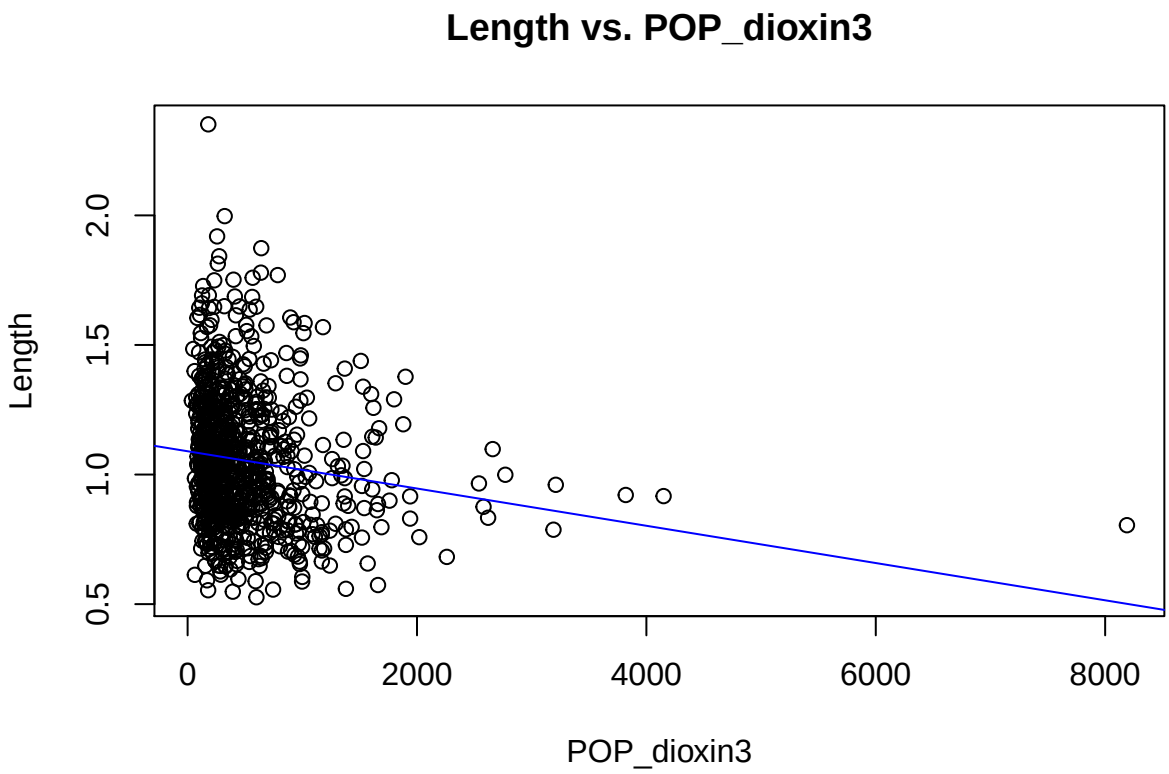

Length vs. POP_PCB9
Length
0.5
1.0
1.5
2.0
0
20000
40000
60000
80000
100000
140000
POP_PCB9

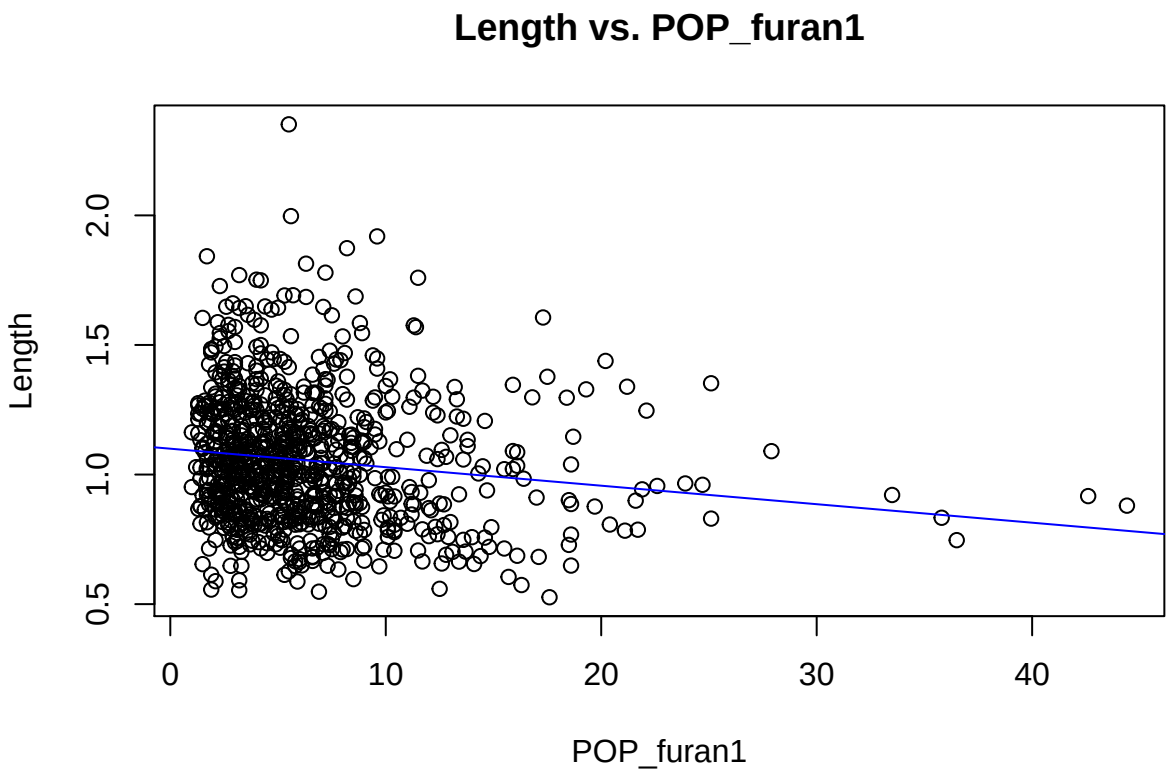

Length vs. POP_PCB10
Length
0.5
1.0
1.5
2.0
0
50
100
150
POP_PCB10

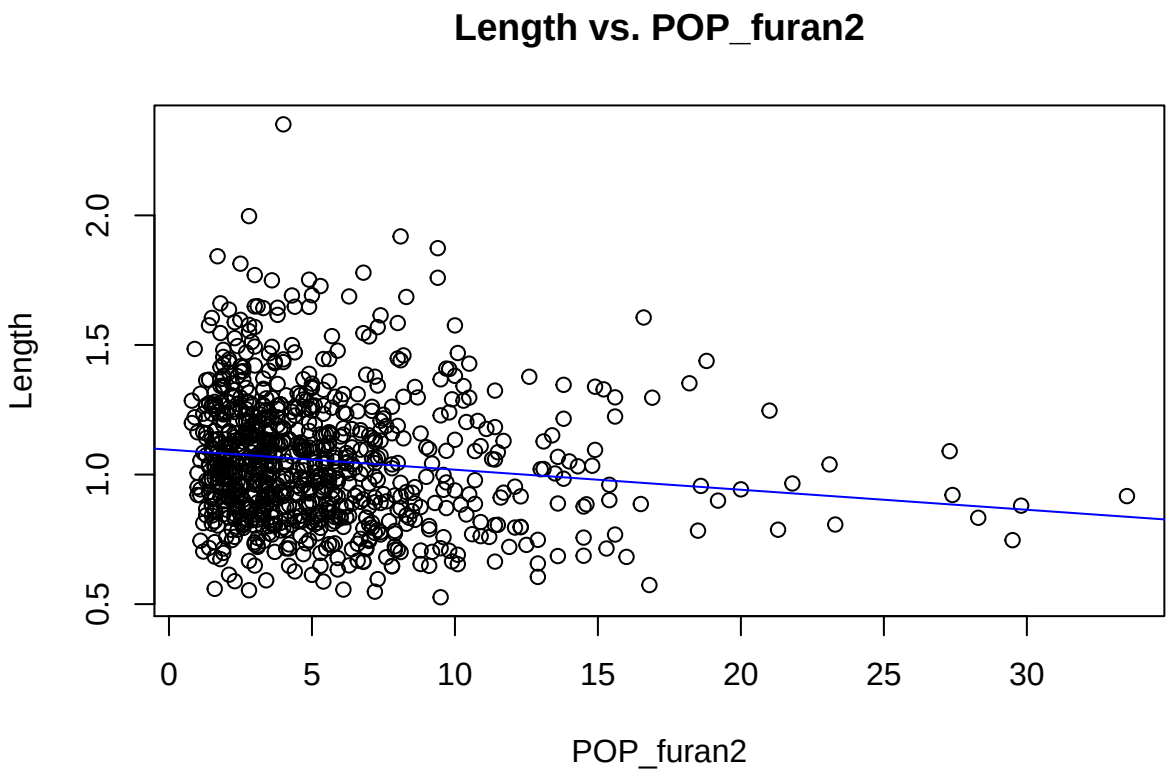

Length vs. POP_PCB11
Length
0.5
1.0
1.5
2.0
0
200
400
600
800
POP_PCB11

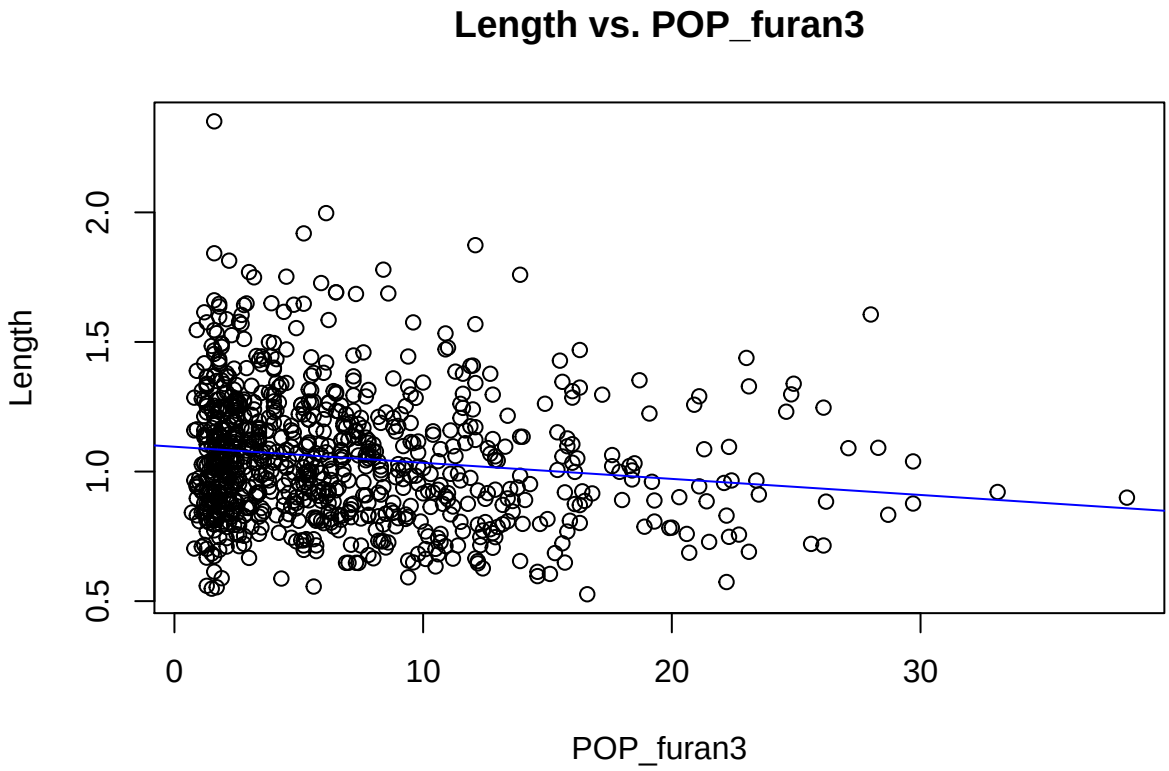

Length vs. POP_dioxin1
Length
0.5
1.0
1.5
2.0
0
200
400
600
POP_dioxin1

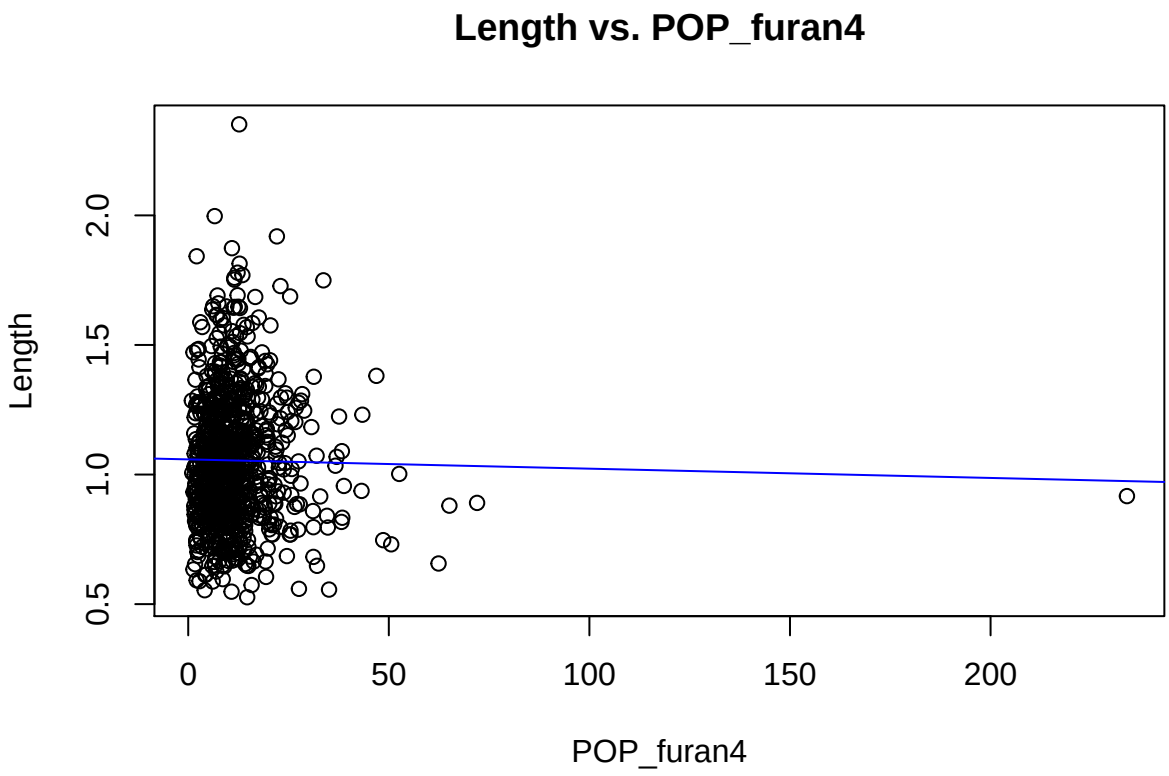

Length vs. POP_dioxin2
Length
0.5
1.0
1.5
2.0
0
50
100
150
200
250
POP_dioxin2

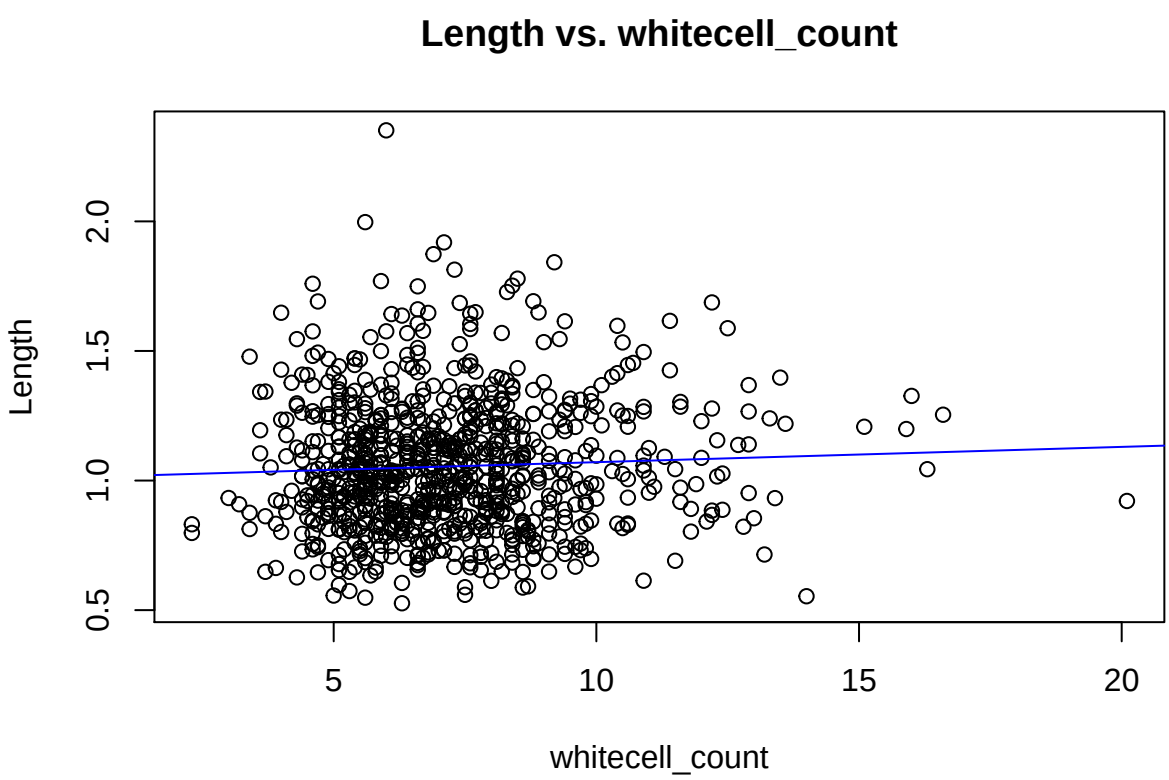

Length vs. POP_dioxin3
Length
0.5
1.0
1.5
2.0
0
2000
4000
6000
8000
POP_dioxin3

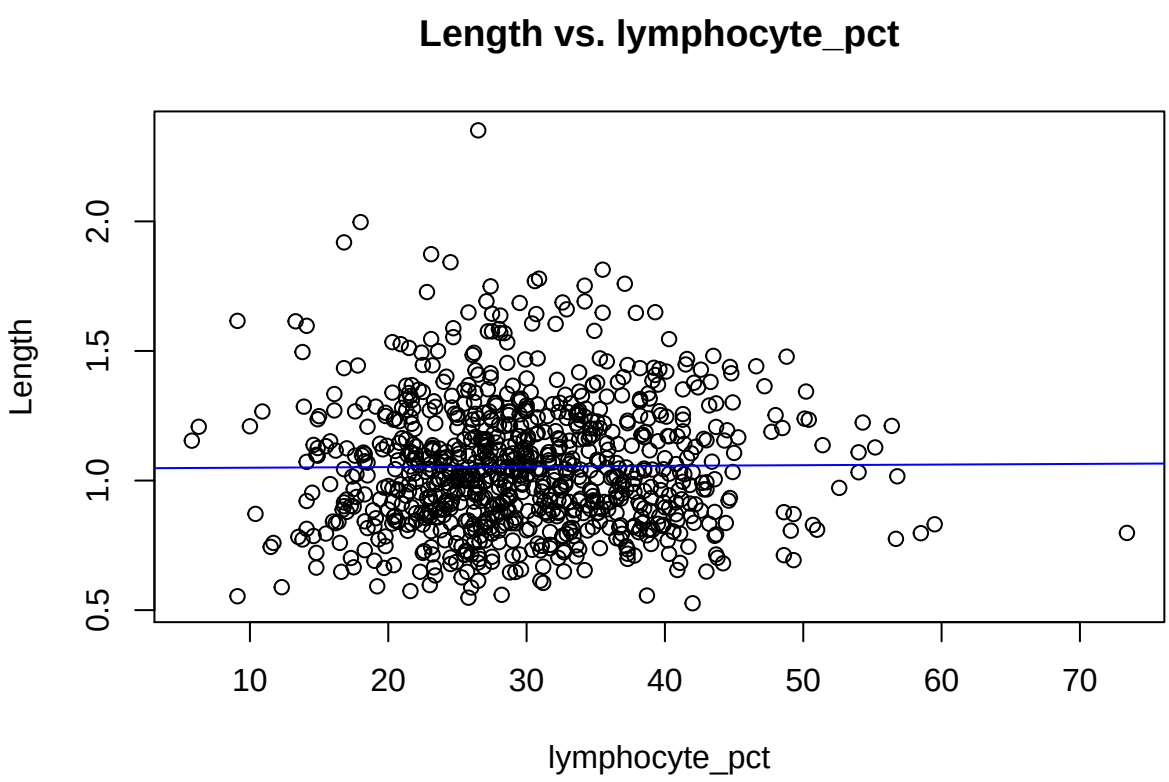

Length vs. POP_furan1
Length
0.5
1.0
1.5
2.0
0
10
20
30
40
POP_furan1

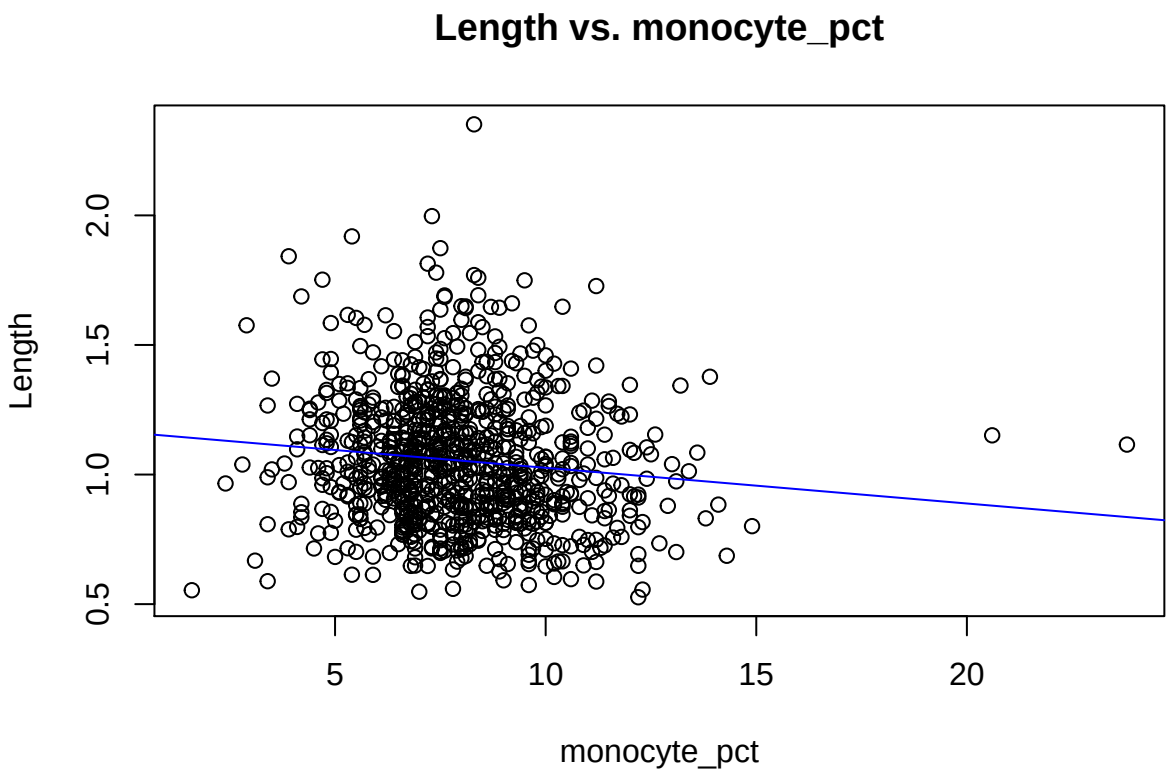

Length vs. POP_furan2
Length
0.5
1.0
1.5
2.0
0
5
10
15
20
25
30
POP_furan2

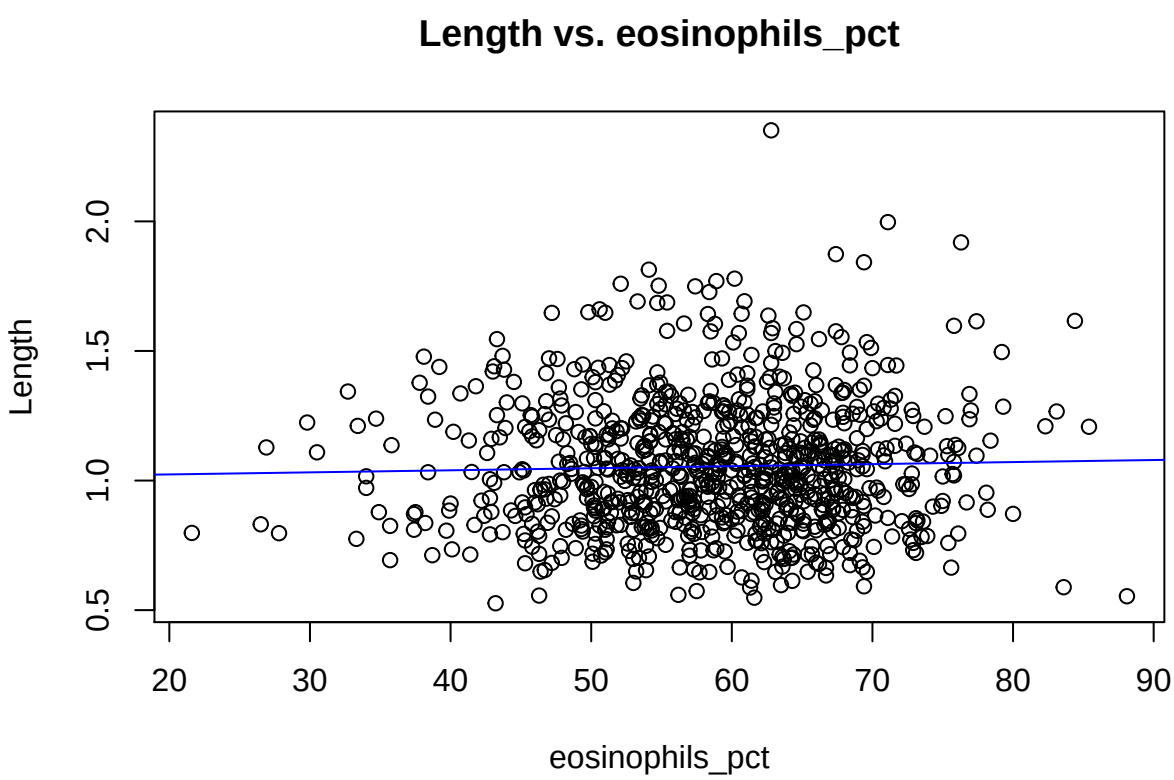

Length vs. POP_furan3
Length
0.5
1.0
1.5
2.0
0
10
20
30
POP_furan3

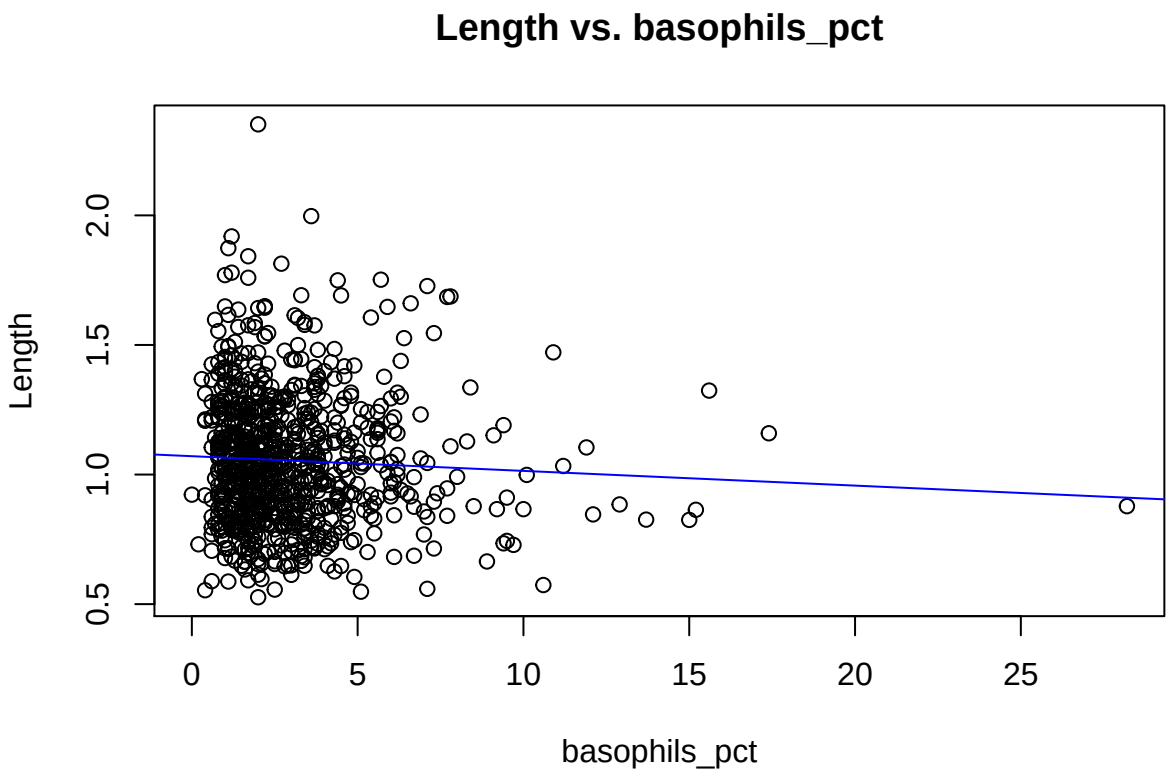

Length vs. POP_furan4
Length
0.5
1.0
1.5
2.0
0
50
100
150
200
POP_furan4

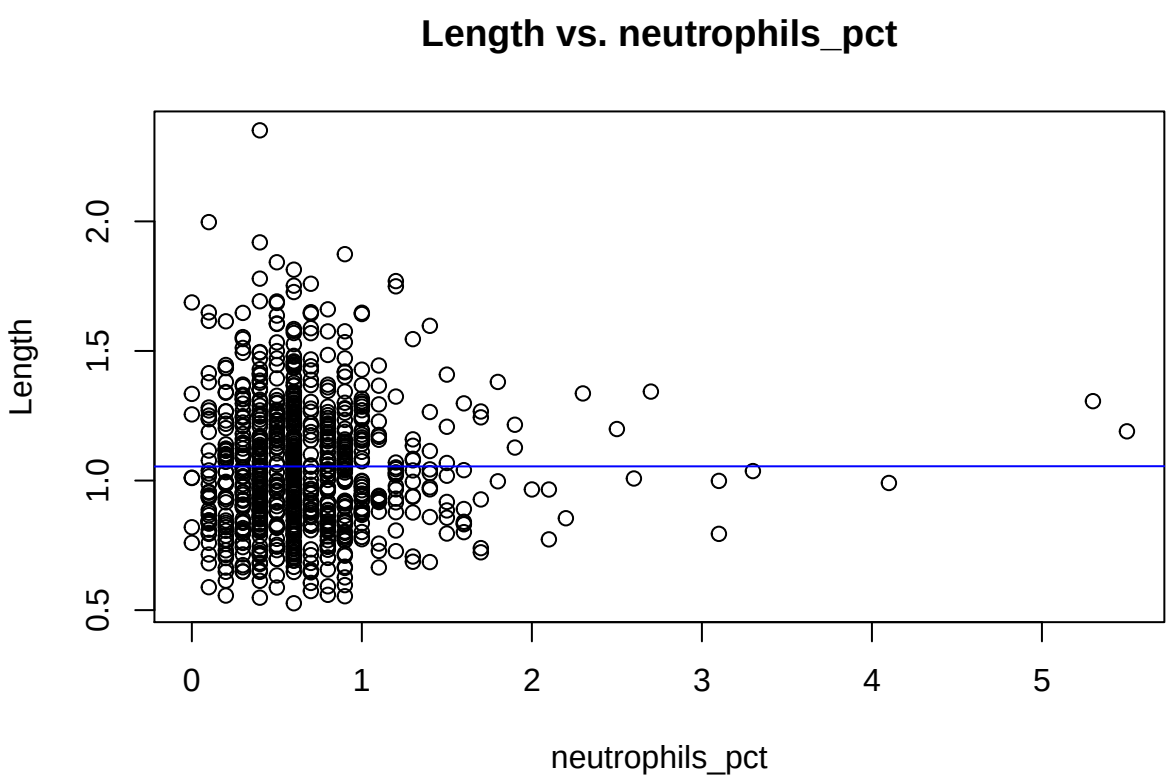

Length vs. whitecell_count
Length
0.5
1.0
1.5
2.0
5
10
15
20
whitecell_count

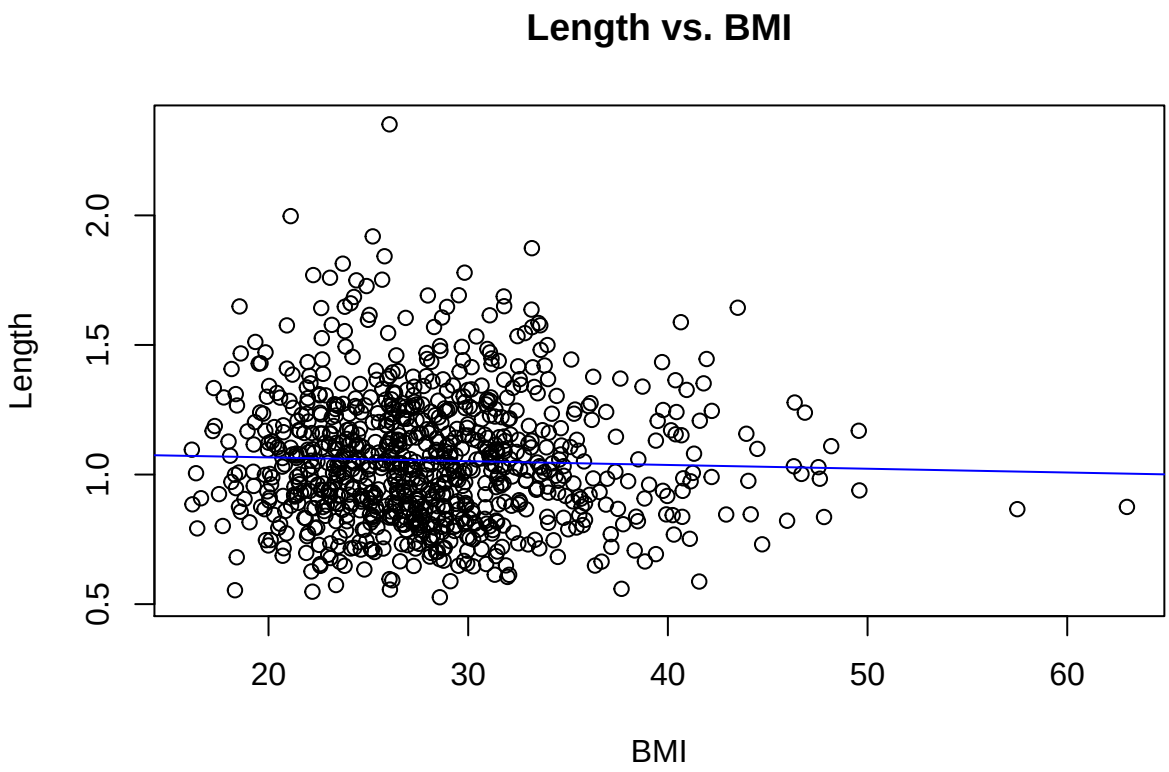

Length vs. lymphocyte_pct
Length
0.5
1.0
1.5
2.0
10
20
30
40
50
60
70
lymphocyte_pct


Length vs. monocyte_pct
Length
0.5
1.0
1.5
2.0
5
10
15
20
monocyte_pct

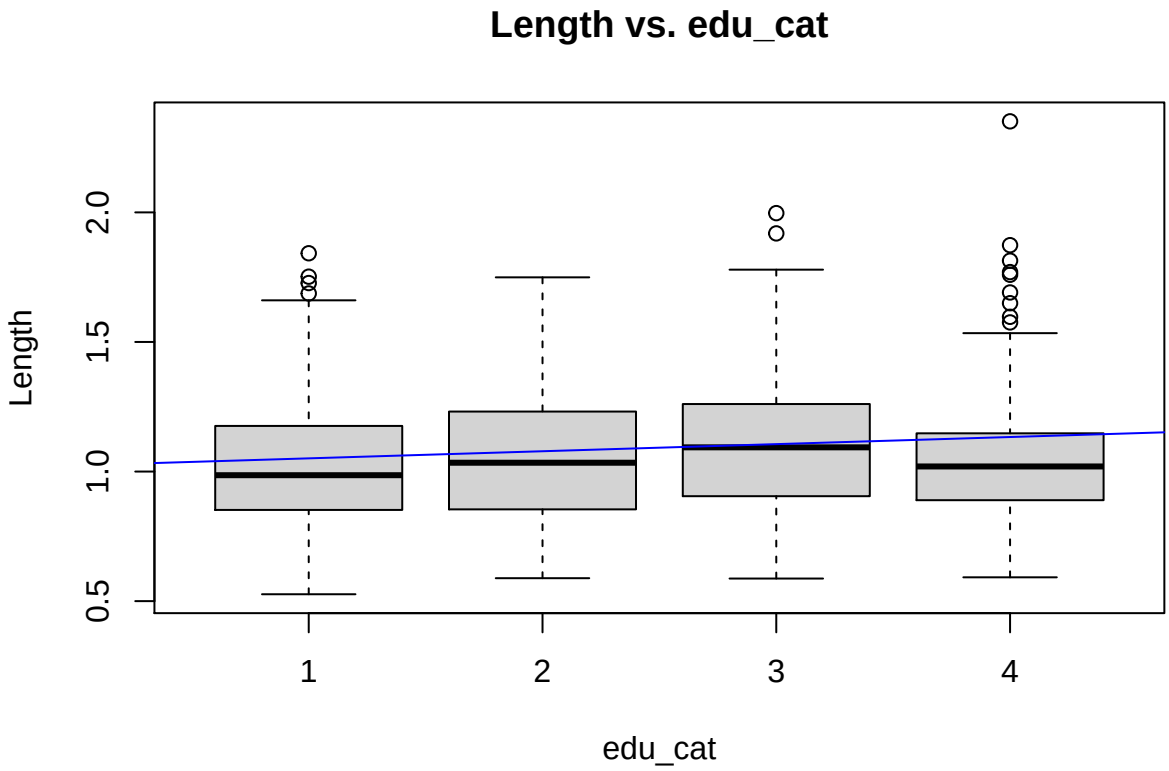

Length vs. eosinophils_pct
Length
0.5
1.0
1.5
2.0
20
30
40
50
60
70
80
90
eosinophils_pct


Length vs. basophils_pct
Length
0.5
1.0
1.5
2.0
0
5
10
15
20
25
basophils_pct

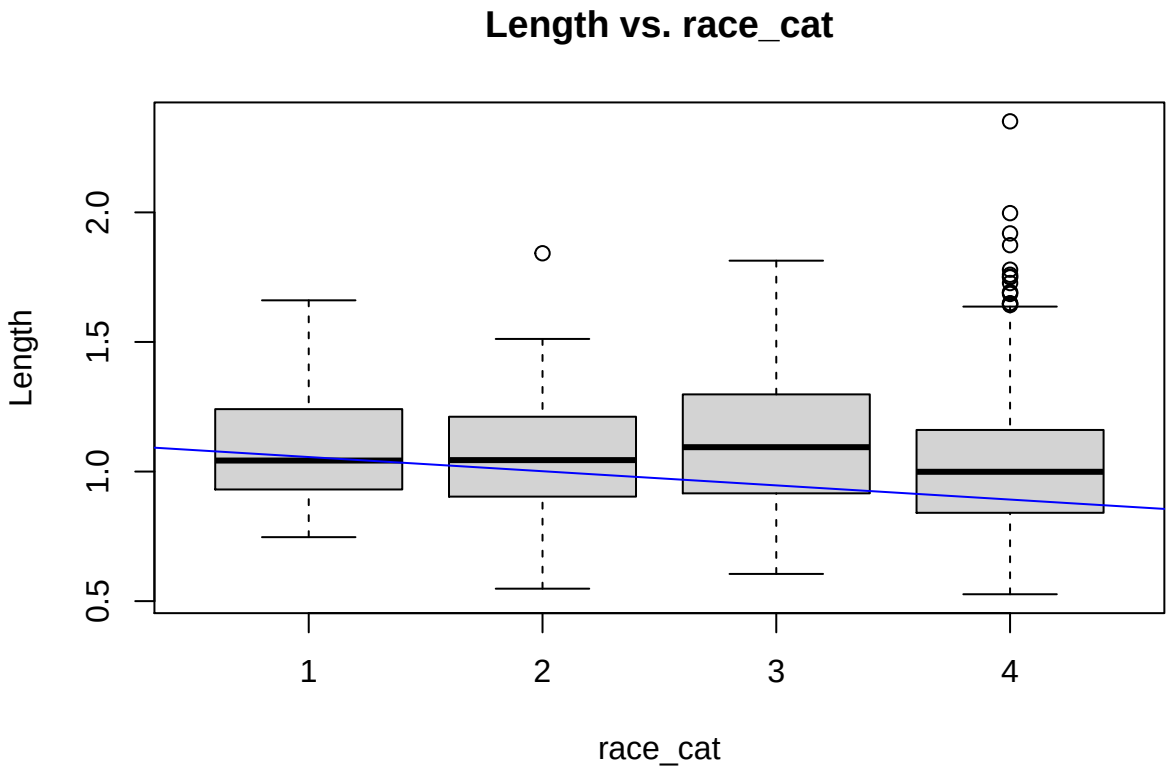

Length vs. neutrophils_pct
Length
0.5
1.0
1.5
2.0
0
1
2
3
4
5
neutrophils_pct

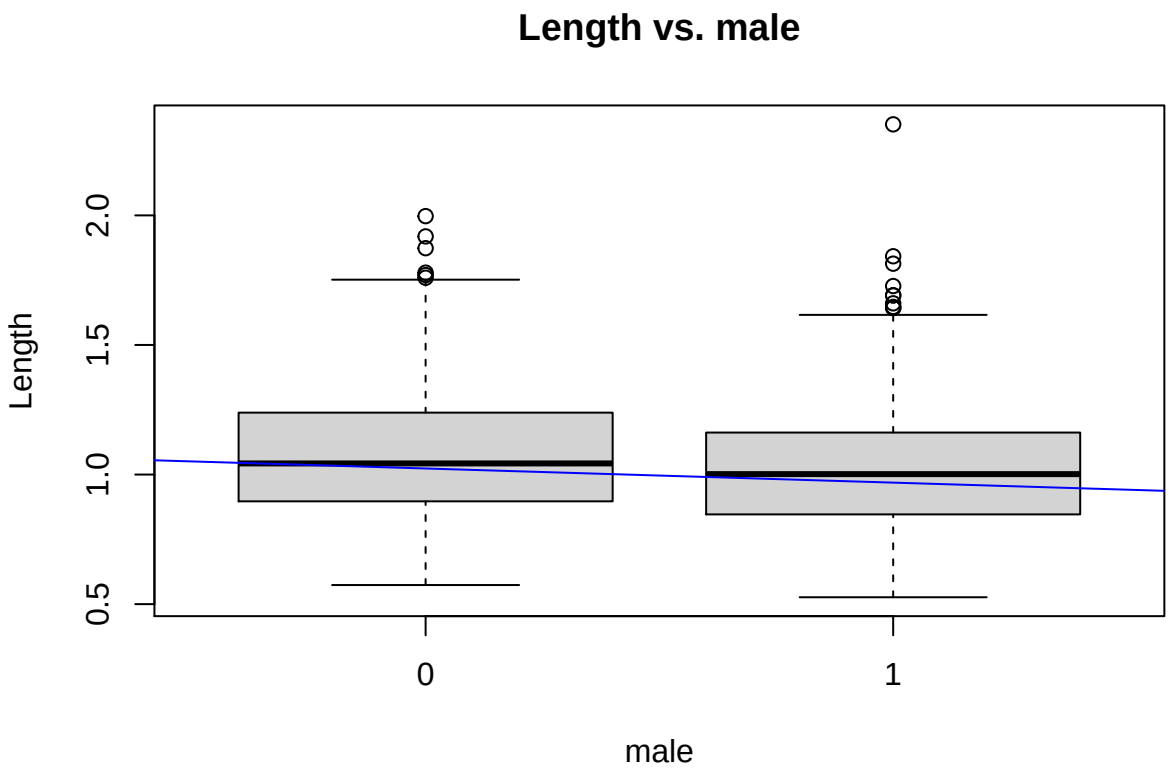


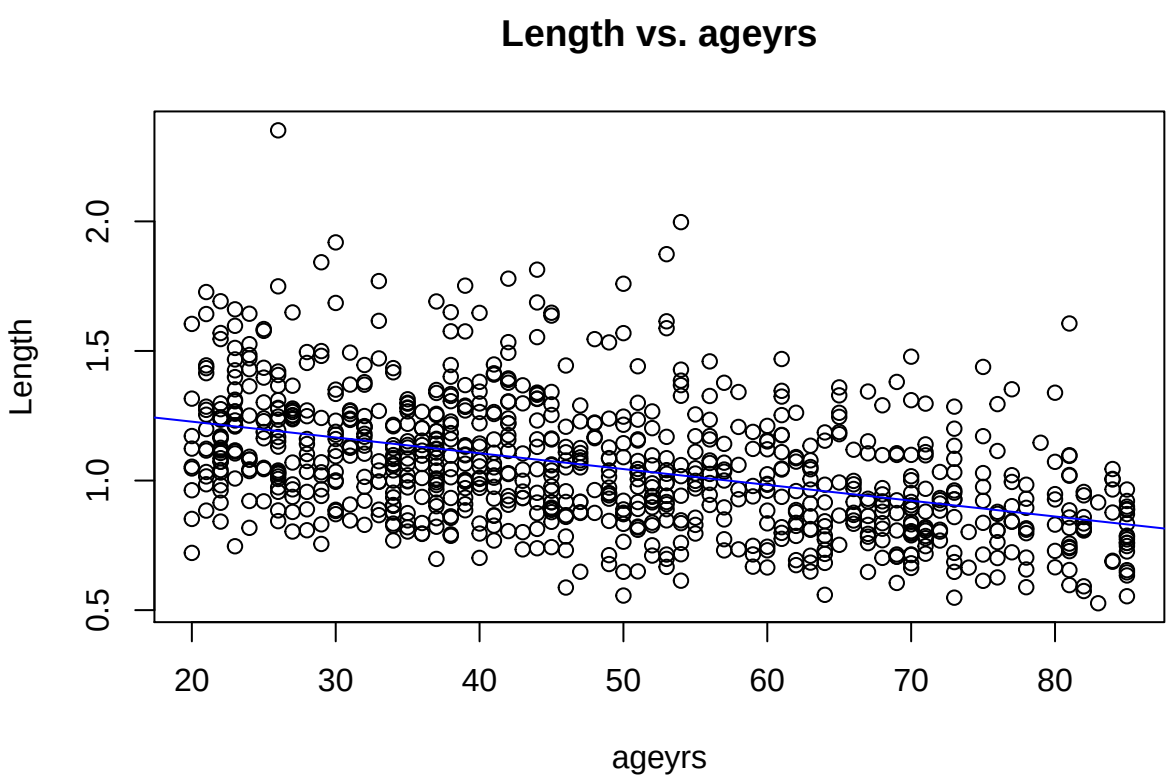

```
## Warning in abline(temp.model, col = "blue"): only using the first two of 4
## regression coefficients
```

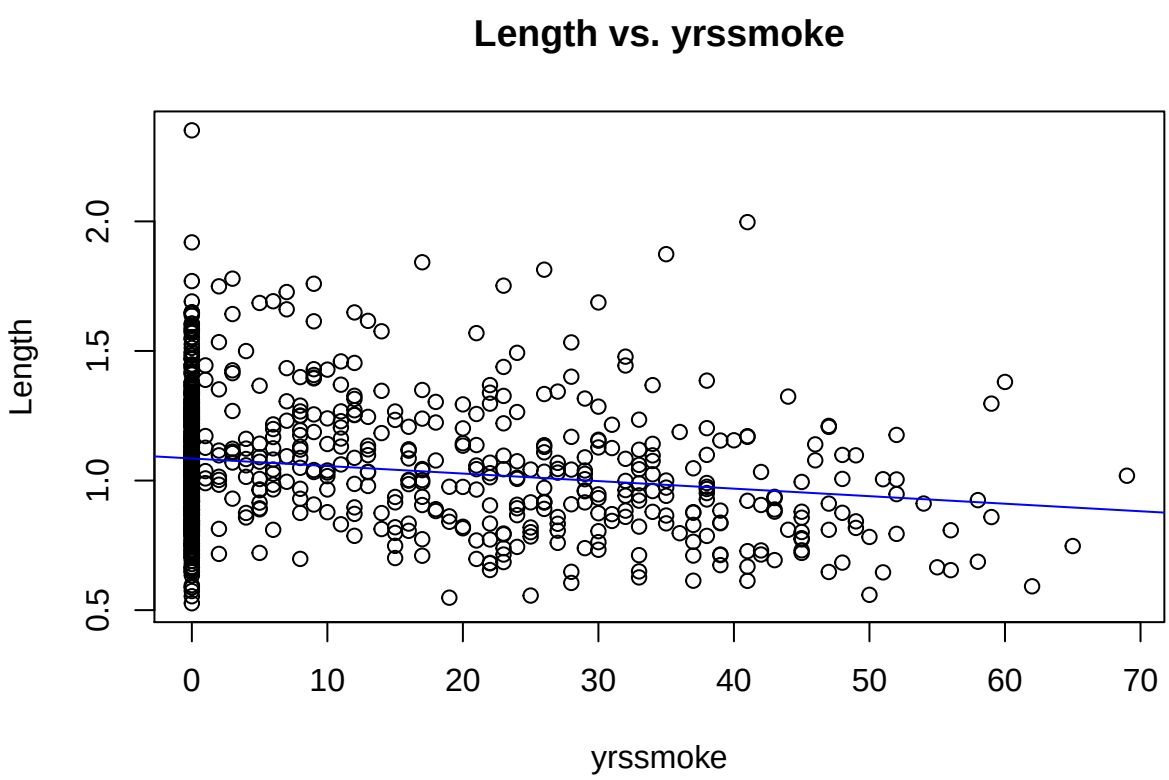


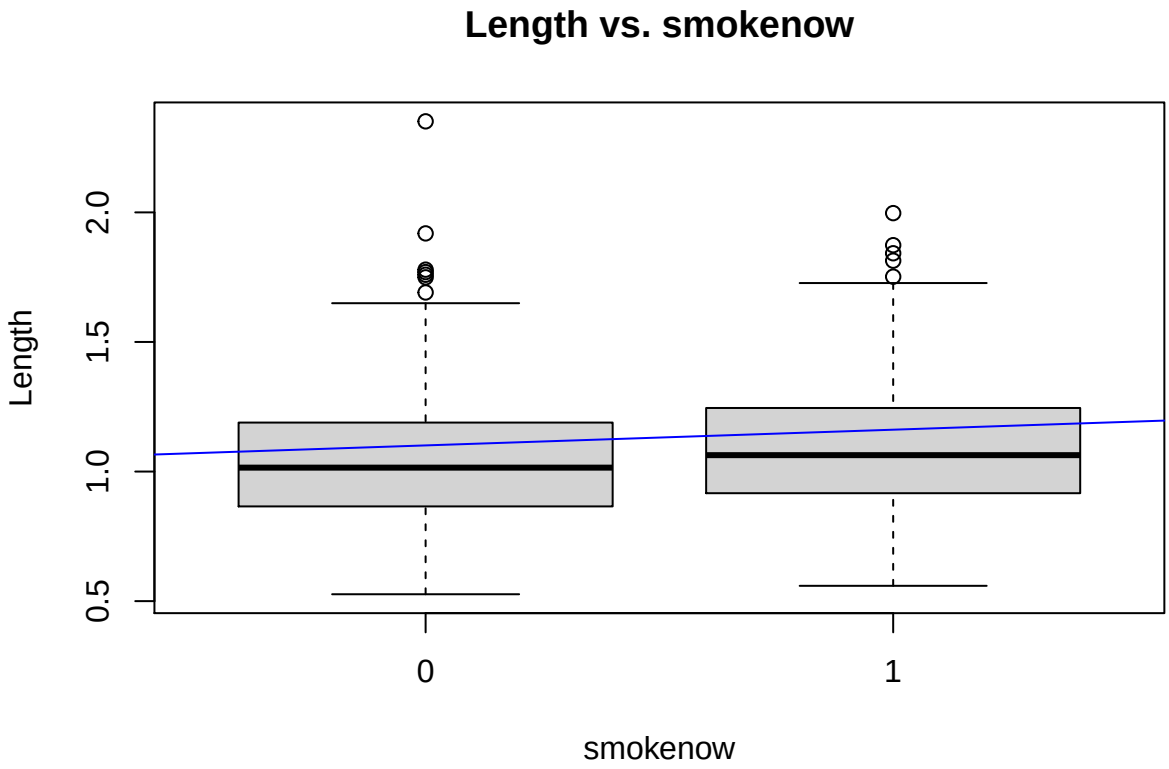

```
## Warning in abline(temp.model, col = "blue"): only using the first two of 4
## regression coefficients
```

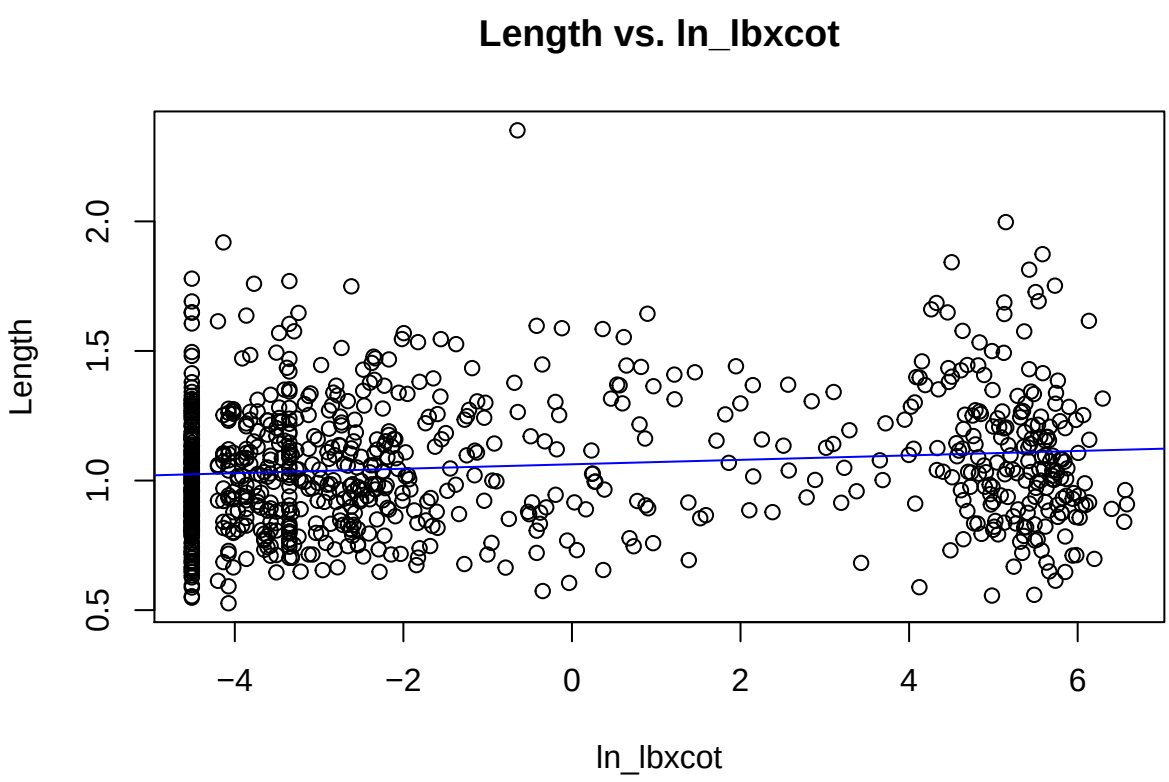

Length vs. race_cat
Length
0.5
1.0
1.5
2.0
1
2
3
4
race_cat


Length vs. male
Length
0.5
1.0
1.5
2.0
0
1
male


Length vs. ageyrs
Length
0.5
1.0
1.5
2.0
20
30
40
50
60
70
80
ageyrs


Length vs. yrssmoke
Length
0.5
1.0
1.5
2.0
0
10
20
30
40
50
60
70
yrssmoke


Length vs. smokenow
Length
0.5
1.0
1.5
2.0
0
1
smokenow




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

```
## [1] 0.3315467
```

```
# 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)
```

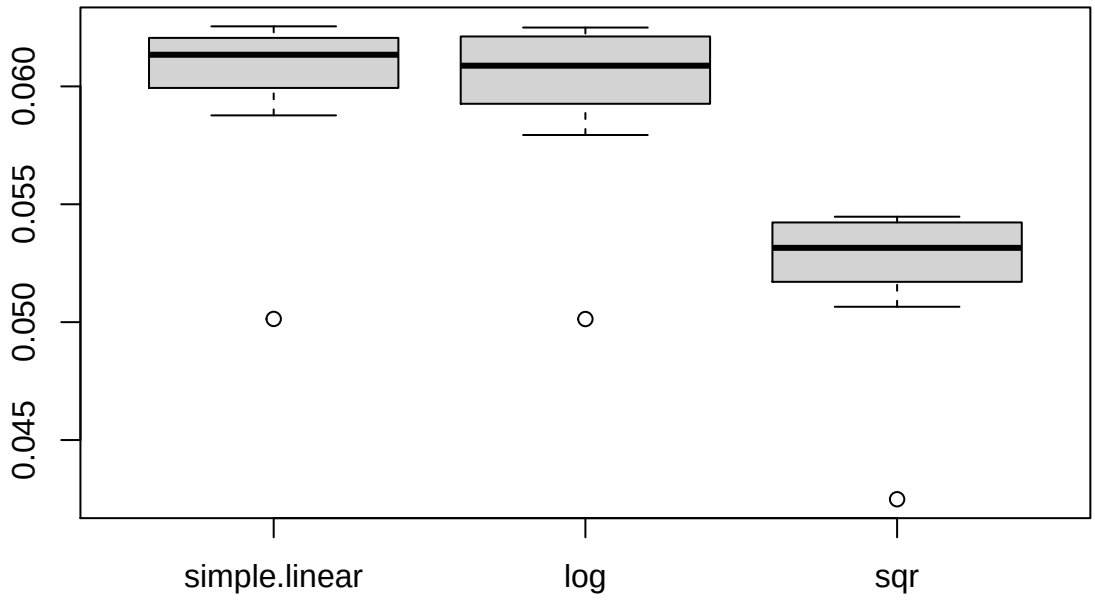
```
##                     GVIF Df GVIF^(1/(2*Df))
## POP_PCB3        5.310340  1        2.304417
## POP_PCB6        9.083828  1        3.013939
## POP_PCB7        4.686485  1        2.164829
## POP_PCB8        5.894052  1        2.427767
## POP_PCB9        7.640480  1        2.764142
## POP_PCB10       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
```

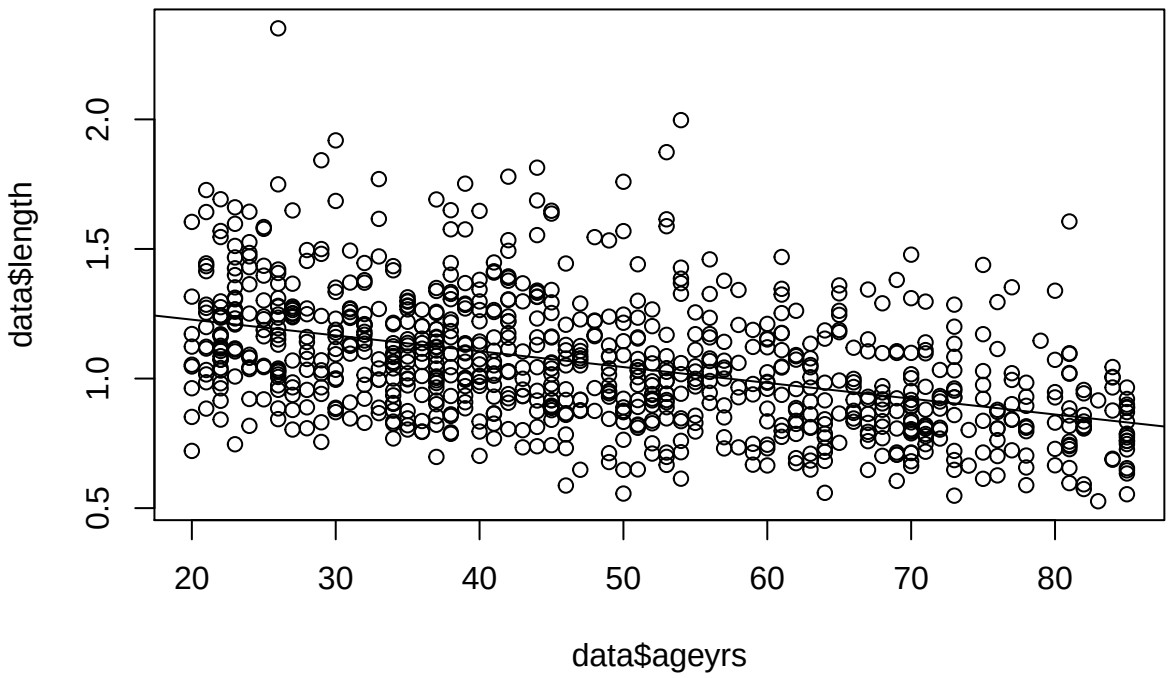
```
which.min(removezero(res[,3]))
```

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

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

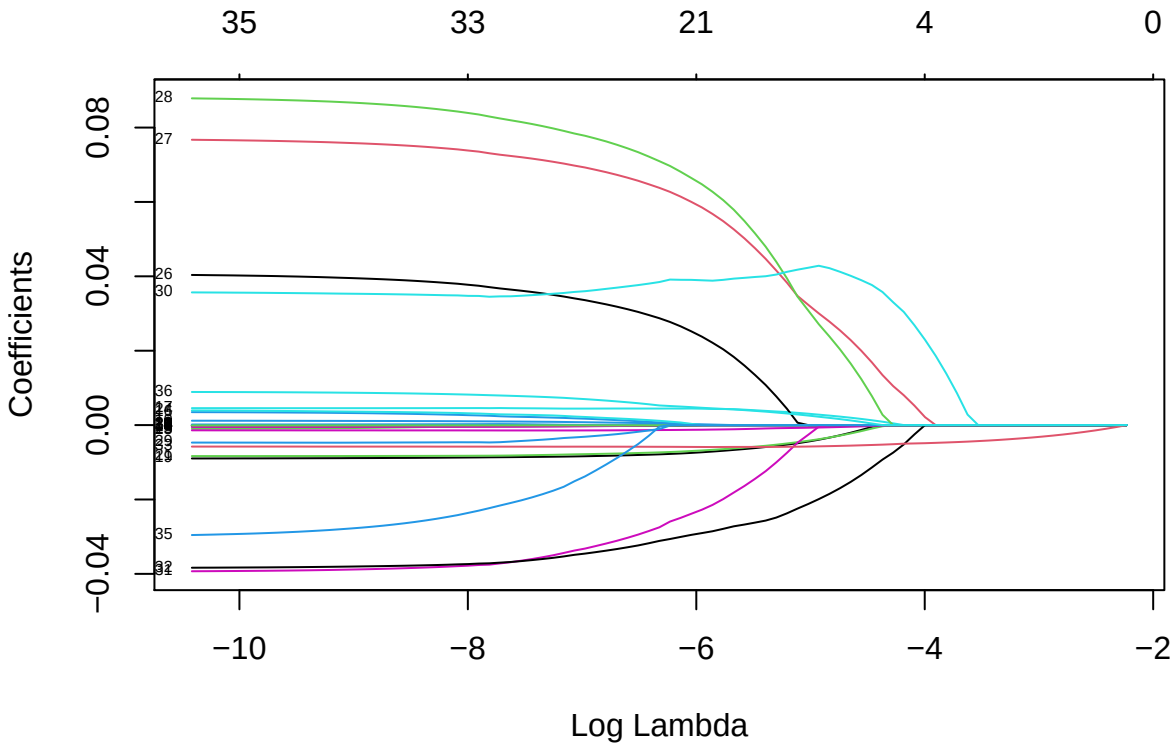
```
## [1] "ageyrs"
```

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

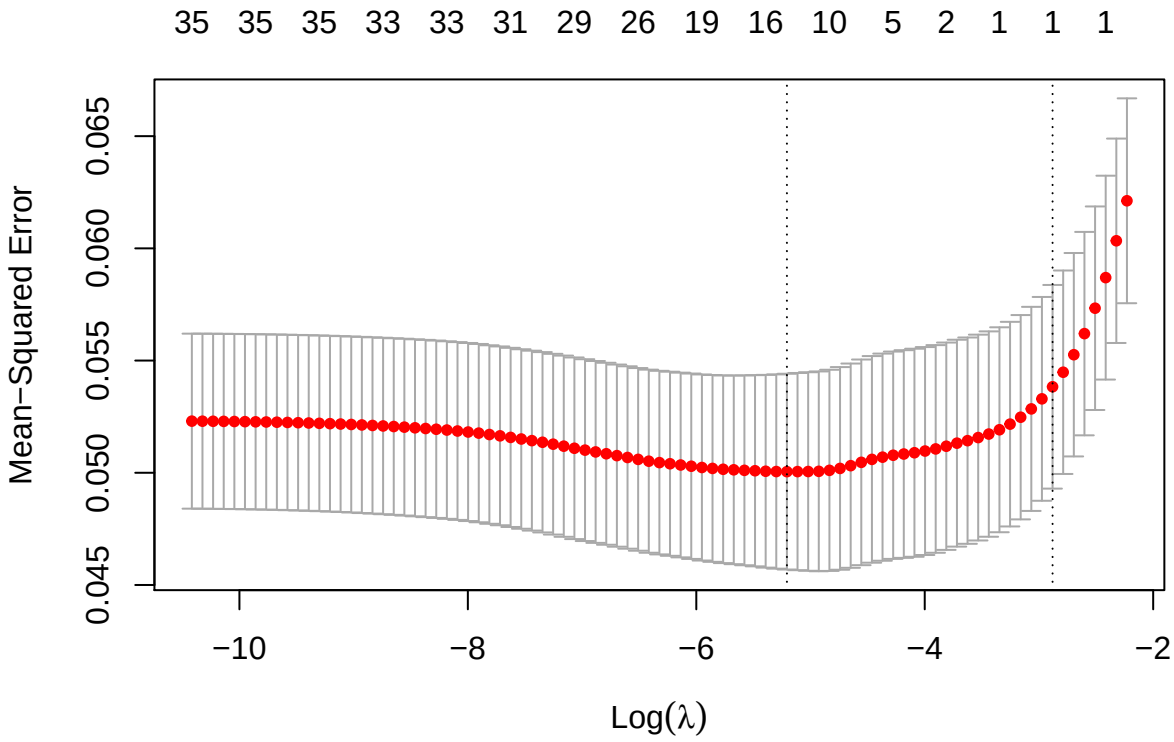



```
#seems there is a linear relationship but looks insufficient.

#Also seems sqr or log does not do exponentially better here




#how is model choosen by automated soluation
```

```
### 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)
```



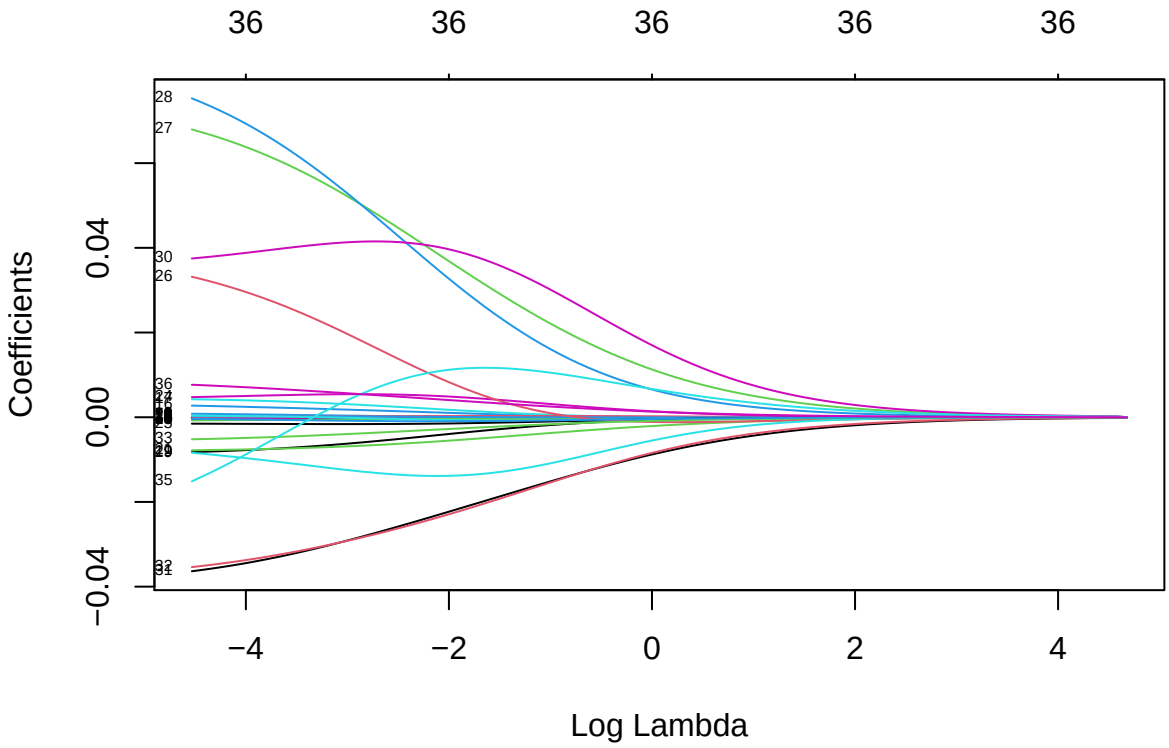

```
## estimated betas for minimum lambda
coef(cvfit_lasso, s = "lambda.min")
```

```
## 37 x 1 sparse Matrix of class "dgCMatrix"
##                                1
## (Intercept)      1.408740e+00
## POP_PCB1         .
## POP_PCB2         .
## POP_PCB3         .
## POP_PCB4         .
## POP_PCB5         .
## POP_PCB6         .
## POP_PCB7         .
## POP_PCB8         .
## POP_PCB9         .
## POP_PCB10        .
## POP_PCB11        5.259764e-06
## POP_dioxin1      .
## POP_dioxin2      .
## POP_dioxin3     -1.028874e-06
## POP_furan1       .
## POP_furan2       .
## POP_furan3       3.508576e-03
## POP_furan4       .
## whitecell_count -5.246881e-03
## lymphocyte_pct   .
```

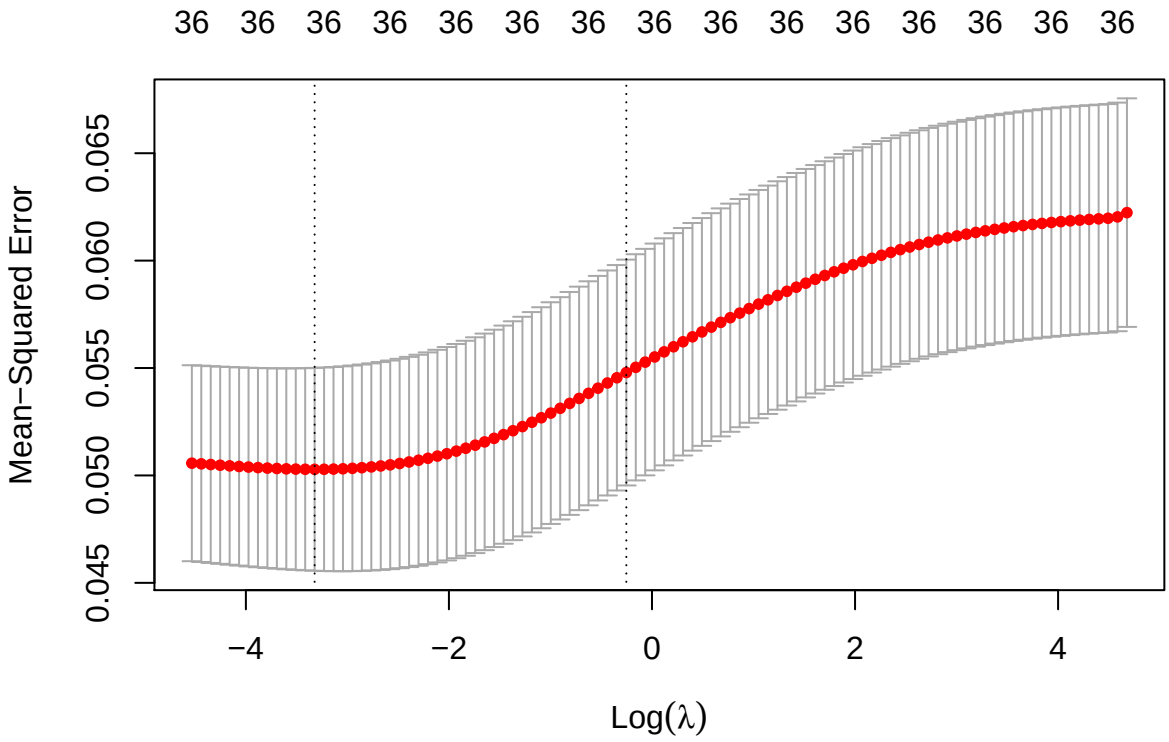
```
## monocyte_pct    -4.946454e-03
## eosinophils_pct  .
## basophils_pct    .
## neutrophils_pct  .
## BMI             -7.954861e-04
## edu_cat2         4.402327e-03
## edu_cat3         3.830631e-02
## edu_cat4         3.947583e-02
## race_cat2        .
## race_cat3        4.119864e-02
## race_cat4       -7.459623e-03
## male1           -2.373316e-02
## ageyrs          -5.830219e-03
## yrssmoke         .
## smokenow1        .
## ln_lbxcot        3.130894e-03
```

```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)
```

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

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




```
## 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)      1.402393e+00
## POP_PCB1        -3.315955e-07
## POP_PCB2        -1.709945e-07
## POP_PCB3         1.223631e-06
## POP_PCB4        -3.529380e-08
## POP_PCB5        -3.725731e-08
## POP_PCB6         1.020459e-07
## POP_PCB7        -5.868158e-07
## POP_PCB8        -4.043131e-07
## POP_PCB9         1.089738e-07
## POP_PCB10        4.879923e-04
## POP_PCB11        6.407079e-05
## POP_dioxin1     -9.500593e-05
## POP_dioxin2     -3.235386e-04
## POP_dioxin3     -9.818505e-06
## POP_furan1      -5.531125e-04
## POP_furan2       2.023438e-03
## POP_furan3       3.433249e-03
## POP_furan4      -8.598438e-05
## whitecell_count -6.831758e-03
## lymphocyte_pct   1.772330e-04
```

```
## monocyte_pct    -7.170060e-03
## eosinophils_pct  1.851318e-04
## basophils_pct    2.686656e-05
## neutrophils_pct  5.362656e-03
## BMI             -1.608879e-03
## edu_cat2         2.325345e-02
## edu_cat3         5.647763e-02
## edu_cat4         5.885813e-02
## race_cat2       -1.155603e-02
## race_cat3        4.068633e-02
## race_cat4       -3.105414e-02
## male1           -3.084793e-02
## ageyrs          -4.304469e-03
## yrssmoke        -7.389594e-04
## smokenow1        1.525498e-04
## ln_lbxcot        6.150215e-03
```

```
## 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
```

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

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

```
cvfit_lasso$del
```

```
## NULL
```

```
MSPE_lasso
```

```
## [1] 0.05089224
```

```
MSPE_ridge
```

```
## [1] 0.05287106
```

```
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
```

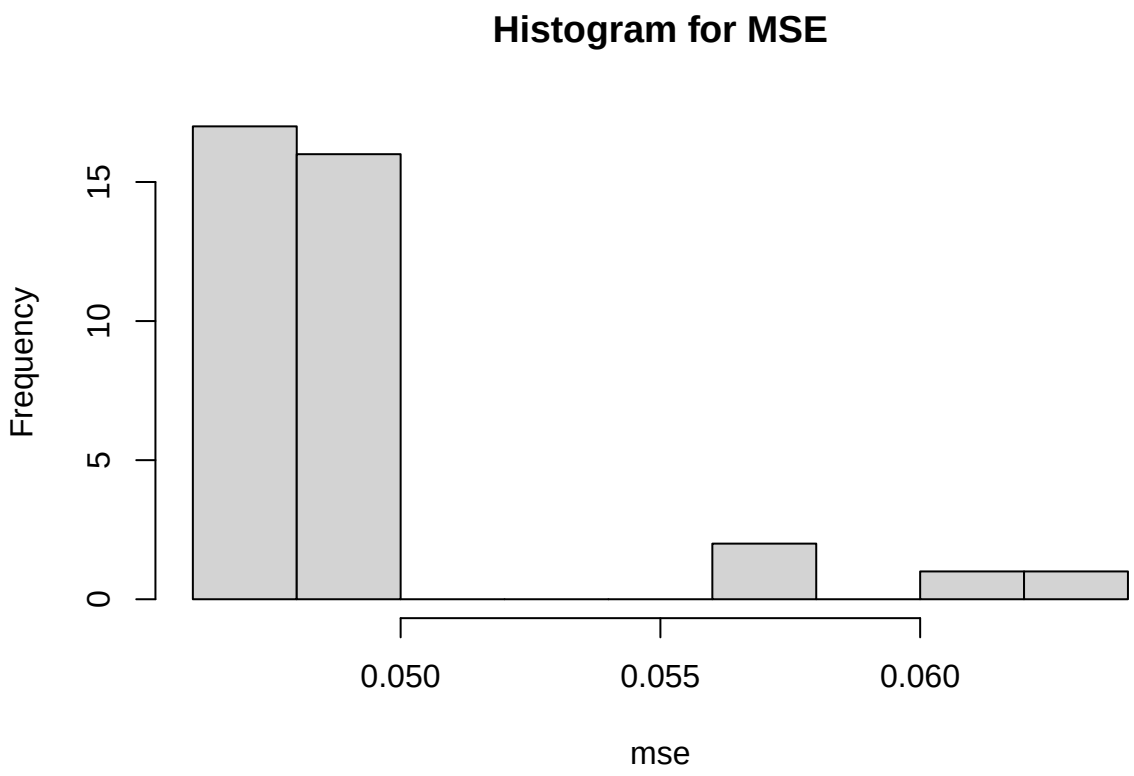
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## 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 ) " "       " "       " "   "*"    " "      " "       " "
```

```r
# rss of all 2 feature model, we see no magical model
mse = models$rss/nrow(data)
hist(mse, main = "Histogram for MSE")
```

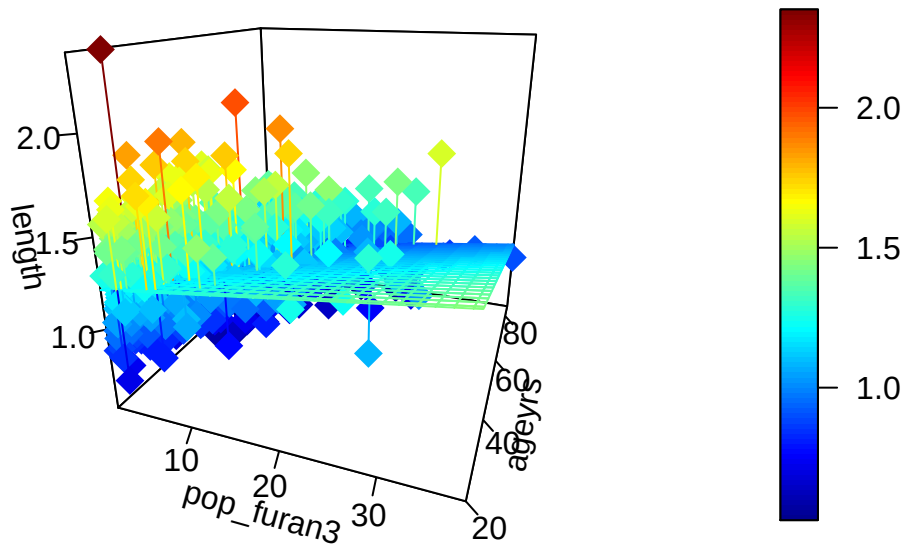



```
str(models)
```

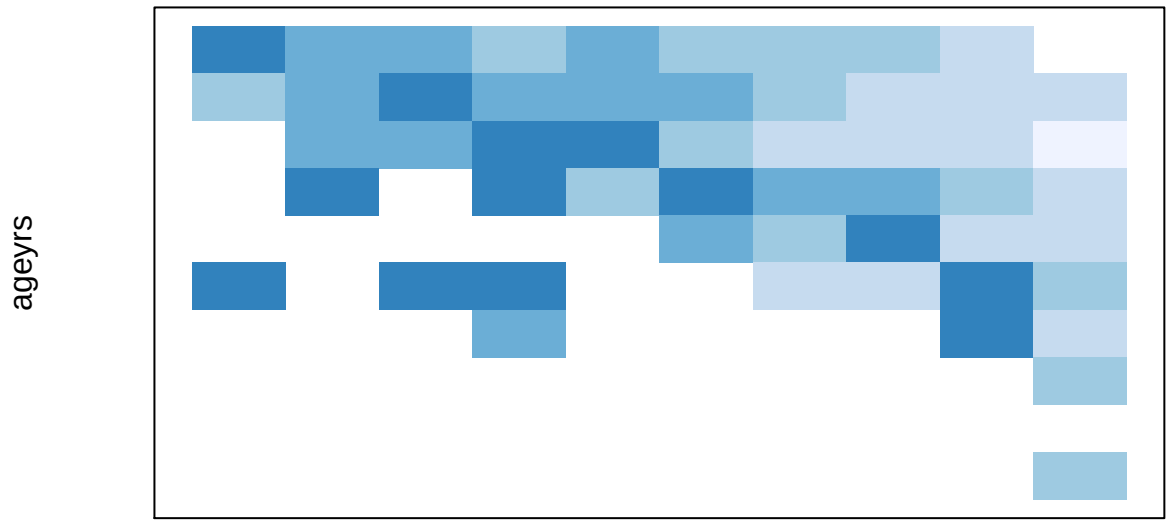
```
## List of 28
##  $ np       : int 37
##  $ nrbar    : int 666
##  $ d        : num [1:37] 864 205481 118 3859 201242 ...
##  $ rbar     : num [1:666] 10.604 0.231 -0.98 48.355 0.433 ...
##  $ thetab   : num [1:37] 1.05431 -0.00291 0.14662 0.01175 -0.00554 ...
##  $ first    : int 2
##  $ last     : 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"
```

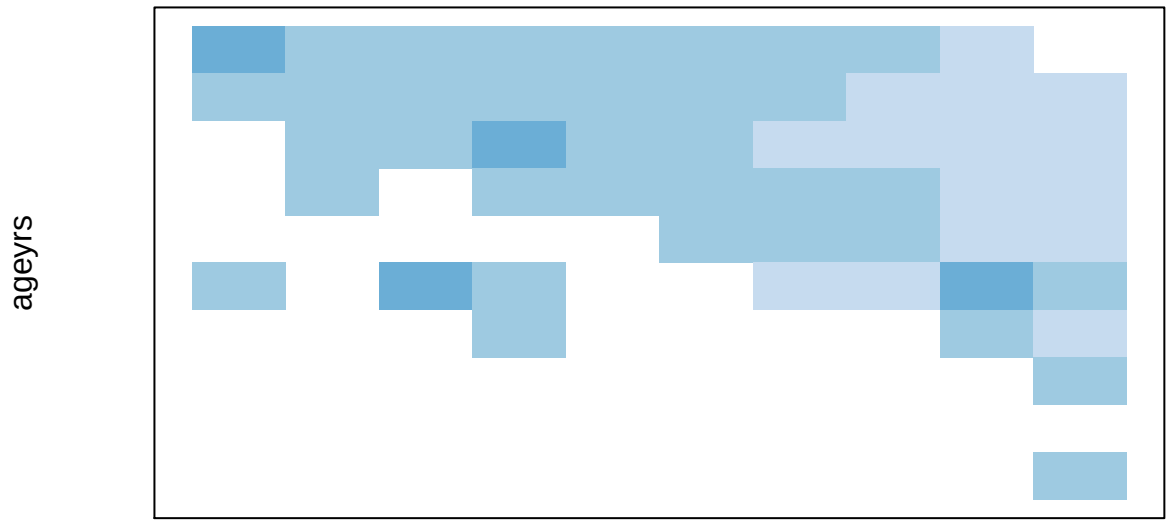
```r
# what does the best 2 feature model look like?
z=data$length
y=data$ageyrs
x=data$POP_furan3

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)
```

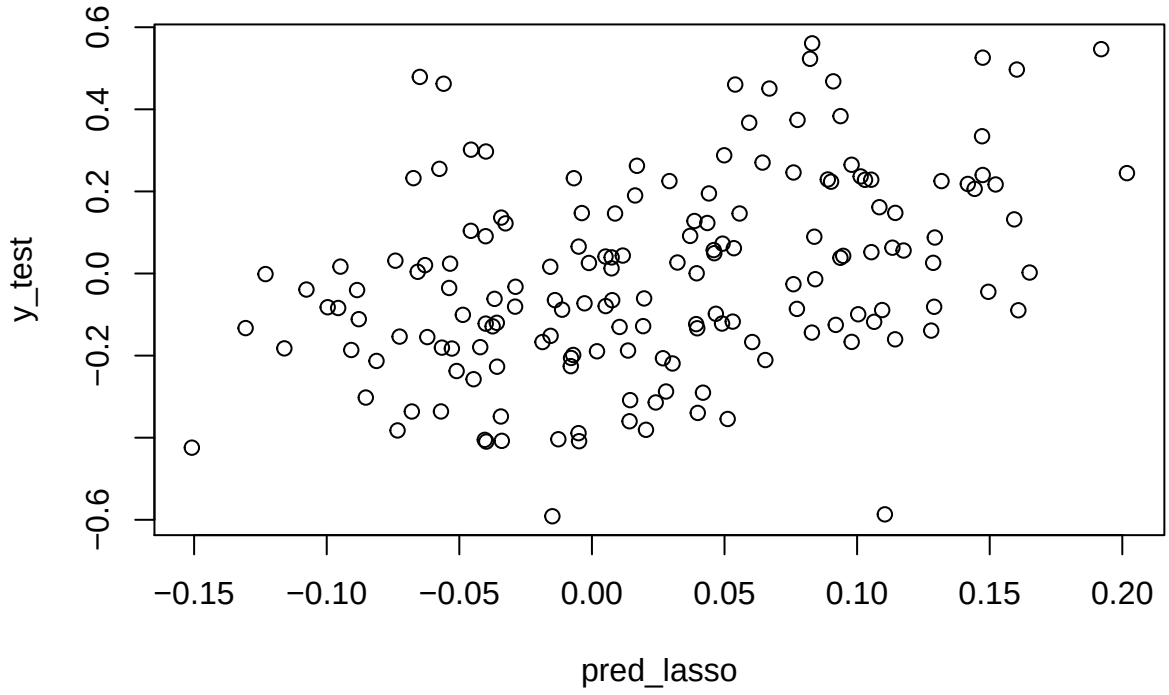



```
# anyway how does this compare to the best fit?
```

```
# 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")
seed <- "20779975"
# 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_1){
  set.seed(seed)
  M = model.matrix(lm(length~., data=data_1))
  cols = colnames(M)
  po.ind = str_detect(cols, "POP")
  y_train = data_1$length[1:700]
  X_train = M[1:700,po.ind]
  y_test= data_1$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"))

}



##########################


# log transform
newdata = data
newdata$length = log(newdata$length)
newdata[,po.ind] = log(data[,po.ind])
newdata2 = data
newdata2[,po.ind] = log(data[,po.ind])
chosen.po.ind= which(lasso.on.pollutants(newdata)!=0)
```

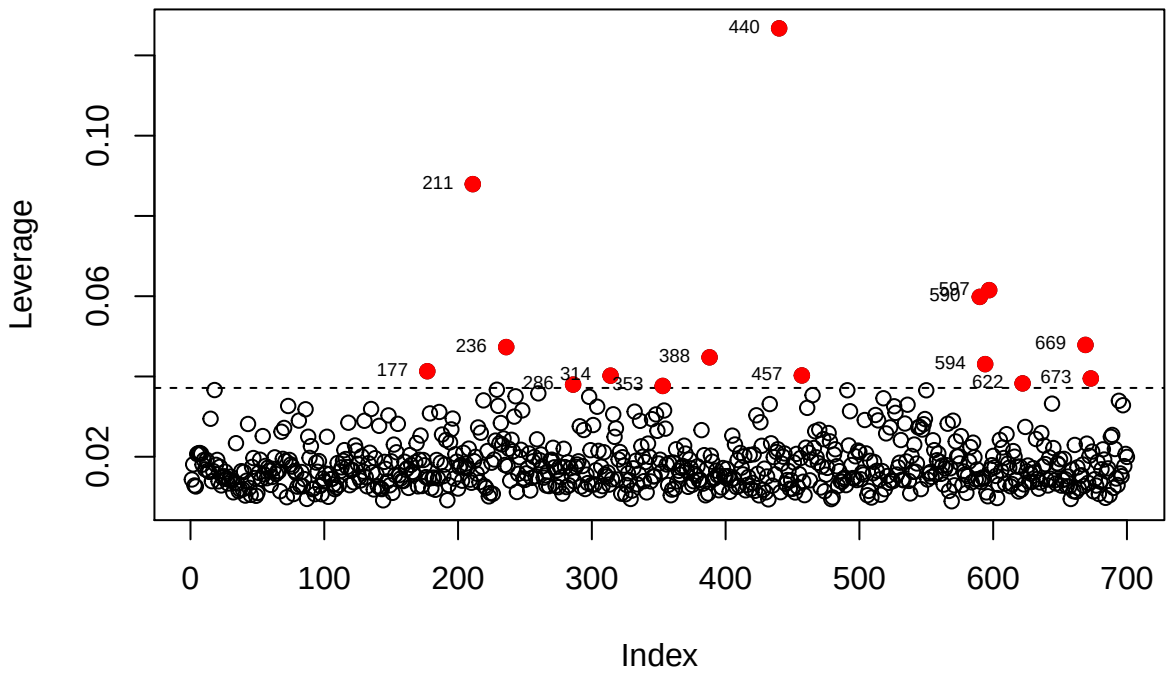
```
## [1] "mspe 0.0493594027827774"
```



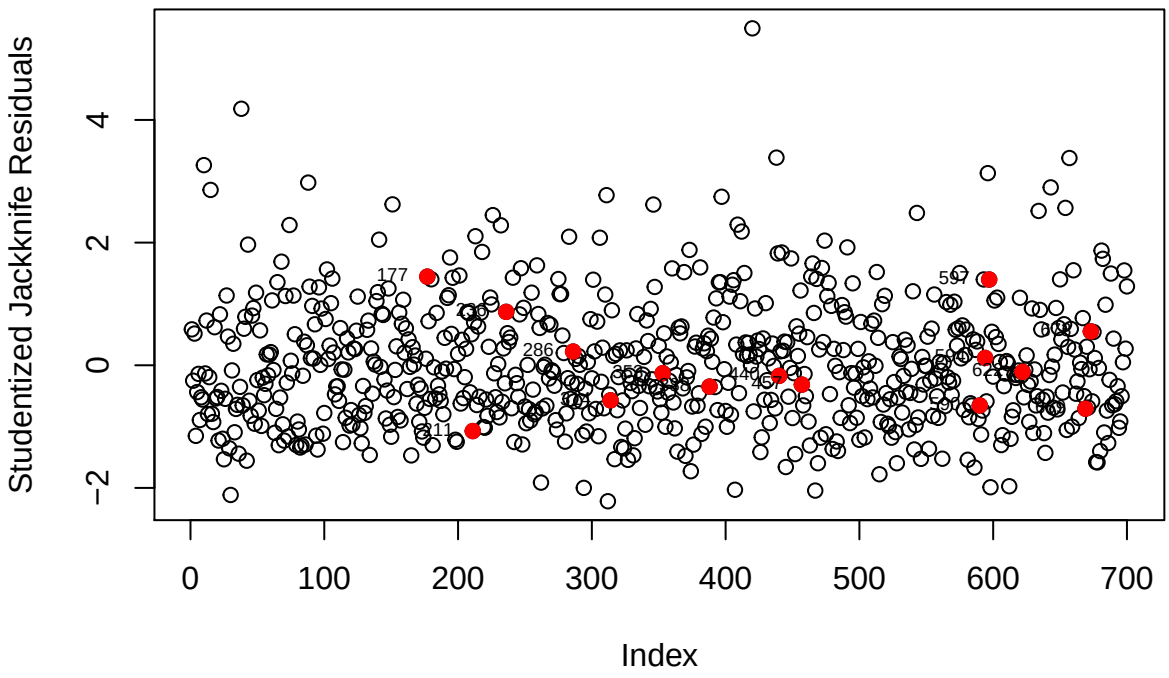

```
chosen.po.ind= chosen.po.ind[2:length(chosen.po.ind)]
```

```
## Influence
## determining outliers
plot.outliers <- function(M){
  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}
  c(sum(lev),ncol(model.matrix(M)))## check trace is same as rank of

  ## plot leverage
  plot(lev,ylab="Leverage")
  abline(h=2*hbar,lty=2) ## add line at 2hbar
  ids <- which(lev>2*hbar) ## x values for labelling points >2hbar
  points(lev[ids]~ids,col="red",pch=19) ## add red points >2hbar
  text(x=ids,y=lev[ids], labels=ids, cex= 0.6, pos=2) ## label points >2hbar
}
outliers <- function(M){
  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}
  c(sum(lev),ncol(model.matrix(M)))## check trace is same as rank of
```

```
  which(lev > 2*hbar)
}

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.outliers(Mstep)
```




```
plot.jackknife.res(Mstep)
```




```
## helpful functions for plotting influence
##----------------DFFITS-----------------------------------------------------------
# Calculates influential points based on DFFITS.
DFFITS <- function(M,method = 1, cutoff = 0.05){
  data <- M$model
  p <- length(attr(terms(M),"term.labels"))
  n <- nobs(M)
  ## check leverage
  h <- hatvalues(M)
  ##----------------DFFITS-----------------
  dffits_m <- dffits(M)
  if(method == 1){
  cutoff <- 2*sqrt((p+1)/n)
  }
  which(abs(dffits_m)>cutoff)
}

plot.DFFITS <- function(M, method = 1, cutoff = 0.05){
  data <- M$model
  p <- length(attr(terms(M),"term.labels"))
  n <- nobs(M)
  ## check leverage
  h <- hatvalues(M)


  dffits_m <- dffits(M)
```

```
  if(method == 1){
    cutoff <- 2*sqrt((p+1)/n)
  }

  ## plot DFFITS
  plot(dffits_m,ylab="DFFITS")
  abline(h=cutoff,lty=2, col = "red")  ## add thresholds
  abline(h=-cutoff,lty=2, col = "red")
  ## highlight influential points
  dff_ind <- which(abs(dffits_m)>cutoff)
  points(dffits_m[dff_ind]~dff_ind,col="red",pch=19) ## add red points
  text(y=dffits_m[dff_ind],x=dff_ind, labels=dff_ind, pos=2) ## label high influence points
  abline(h = cutoff, col = "red", lty = 2)
  abline(h = -cutoff, col = "red", lty = 2)
}


##----------------Cook's Distance----------------------------------------------------------------
# Calculates influential points based on Cook's Distance
CD <- function(M, cutoff = 0.5){
  p <- length(attr(terms(M),"term.labels"))
  n <- nobs(M)
  D <- cooks.distance(M) # Cook's distance
  ## influential points
  which(pf(D,p+1,n-p-1,lower.tail=TRUE)>cutoff)
}

plot.CD <- function(M,method = 1, cutoff = 0.5){
  # method = 1 is default (may not print any influential points if cutoff is not low enough)
  # method = else <- calculate using simple R method
  if(method == 1){
    p <- length(attr(terms(M),"term.labels"))
    n <- nobs(M)
    D <- cooks.distance(M) # Cook's distance
    ## influential points
    inf_ind <- which(pf(D,p+1,n-p-1,lower.tail=TRUE)>cutoff)

    ## plot cook's Distance
    plot(D,ylab="Cook's Distance")
    points(D[inf_ind]~inf_ind,col="red",pch=19) ## add red points
    text(y=D[inf_ind],x=inf_ind, labels=inf_ind, pos=4) ## label high influence points
  }else{
    plot(M,which = 4)
  }
}


##----------------DFBETAS----------------------------------------------------------------
# Calculates influential points based on DFBETAS.
DFBETAS <- function(M, method = 1, cutoff = 0.05){
  DFBETAS <- dfbetas(M)
  dim(DFBETAS)
  n <- nobs(M)
```

```
  # method = 1 <- default cutoff 2/sqrt(n)

  if(method == 1){
    cutoff <- 2/sqrt(n)
  }

  vals <- list()
  for(i in 2:dim(DFBETAS)[2]){
    vals[[i]] <- which(abs(DFBETAS[,i])>cutoff)
  }
  vals
}

plot.DFBETAS <- function(M, method = 1, cutoff = 0.05){

  n <- nobs(M)

  # method = 1 <- default cutoff 2/sqrt(n)
  if(method == 1){
    cutoff <- 2/sqrt(n)
  }


  DFBETAS <- dfbetas(M)
  dim(DFBETAS)
  ## beta1
  for(i in 2:dim(DFBETAS)[2]){
    plot(DFBETAS[,i], type="h",xlab="Obs. Number",
         ylab=bquote(beta[.(i)]), main = "DFBETAS")
    show_points <- which(abs(DFBETAS[,i])>cutoff)
    points(x=show_points,y=DFBETAS[show_points,i],pch=19,col="red")
    abline(h = cutoff, col = "red", lty = 2)
    abline(h = -cutoff, col = "red", lty = 2)
    text(x=show_points,y=DFBETAS[show_points,i],labels=show_points,pos=2)
  }
}
```

```
# Error Analysis
errorAnalysis <- function(M){
  ## residuals
  newdata <- M$model
  res1 <- resid(M) # raw residuals
  stud1 <- res1/(sigma(M)*sqrt(1-hatvalues(M))) # studentized residuals

  ## plot distribution of studentized residuals
  hist(stud1,breaks="FD",
     probability=TRUE,xlim=c(-4,4),
     xlab="Studentized Residuals",
     main="Distribution of Residuals")
  grid <- seq(-3.5,3.5,by=0.05)
  lines(x=grid,y=dnorm(grid),col="blue") # add N(0,1) pdf

  ## qqplot of studentized residuals
```

```
  qqnorm(stud1)
  abline(0,1) # add 45 degree line

  ## plot of residuals vs X
  factors <- attr(terms(M),"term.labels")
  for(i in 1:length(factors)){
    ind <- which(colnames(newdata)==factors[i])
    plot(res1 ~ newdata[,ind],ylab = "residuals",
         xlab = factors[i], main = paste0("Residuals vs ",factors[i]), ylim = c(-1,2))
  }

  ## plot of studentized residuals vs fitted values
  plot(stud1~fitted(M),
     xlab="Fitted Vals",
     ylab="Studentized Residuals",
     main="Residuals vs Fitted")

}
```

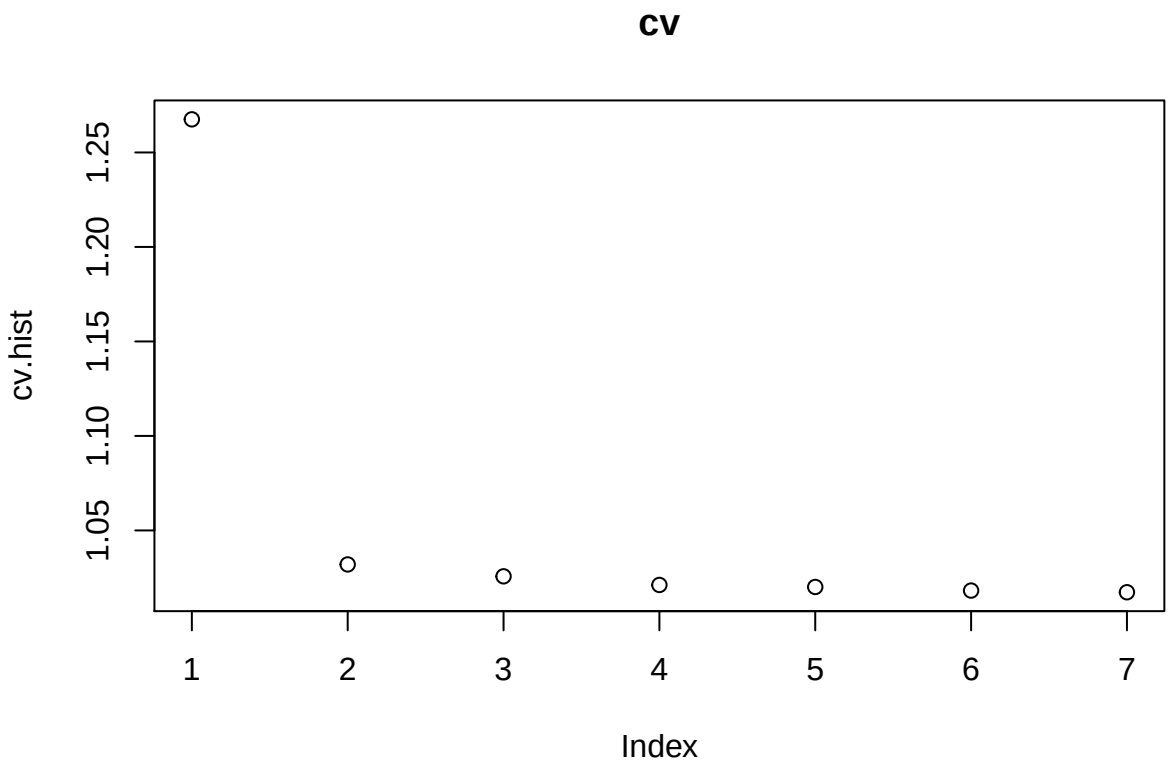
```
kfolds.cv <- function(dat, expr){
  kfolds=10
  mspe = rep(0, kfolds)
  ind = rep(1:kfolds, length=nrow(dat))
  for(ii in 1:kfolds) {
    train<- which(ind!=ii) # training observations
    M.cv <- lm(expr, data=data[train,])
    # cross-validation residuals
    M.res <- dat$length[-train] - # test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


# 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()
  DFFITS.hist = c()
  outliers.hist = c()
  j=0
  models = list()
  while (TRUE) {
    j=j+1
    # FOR METHODS EXPLANATIONprint(paste("step", j))
```

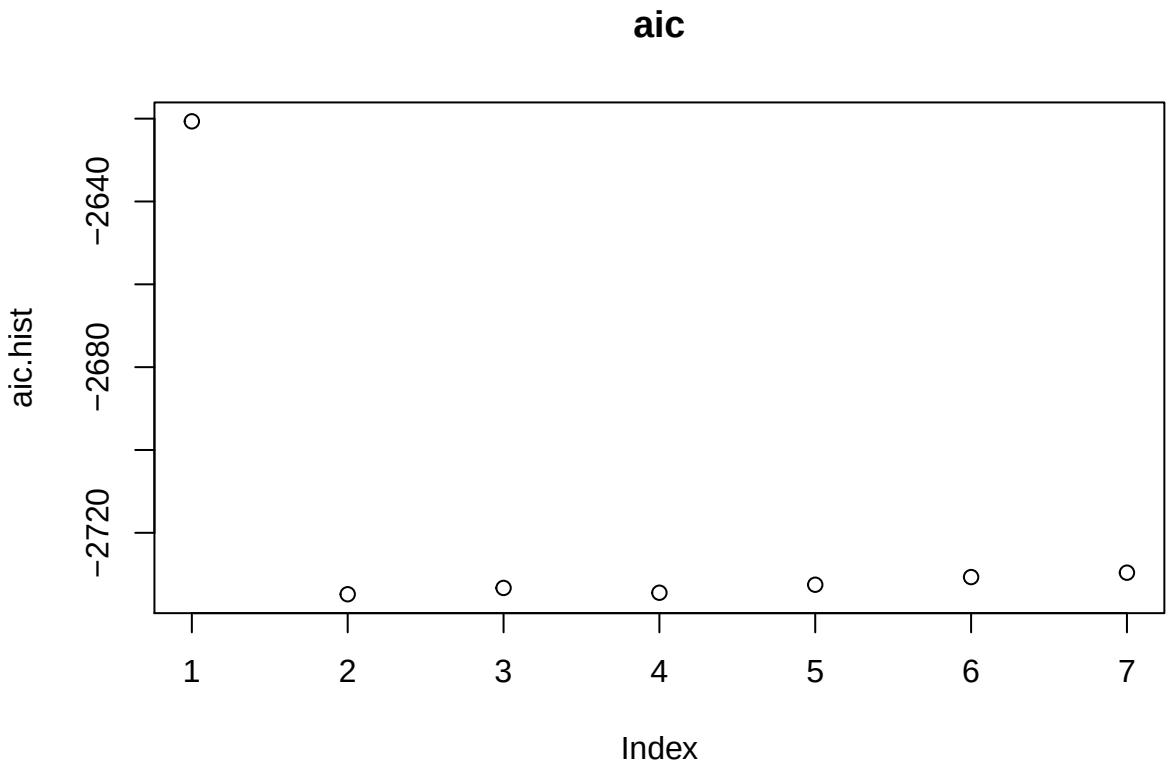
```
  cov.in.m = colnames(model$model)
  cov.all = colnames(newdata)
  names.to.try = cov.all[! cov.all %in% cov.in.m]
  nn = length(names.to.try)
  #update tracks
  cv.hist[j]=kfolds.cv(newdata, expr)
  aic.hist[j] = extractAIC(model)[2]
  coef.hist[[j]] = coef(model)
  DFFITS.hist[j] = length(DFFITS(model))
  outliers.hist[j] = length(outliers(model))
  models[[j]] = model
  cv.score = rep(0, nn)
  if(length(names.to.try) == 0){
    # FOR METHODS EXPLANATIONprint("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]){
    # FOR METHODS EXPLANATIONprint ("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)
  }
}
plot(cv.hist, main = "cv")
plot(aic.hist, main = "aic")
plot(DFFITS.hist, main = "# of Influential Points - DFFITS")
plot(outliers.hist, main = "# of Outliers")
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="coefficent of pollutants", 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)
```

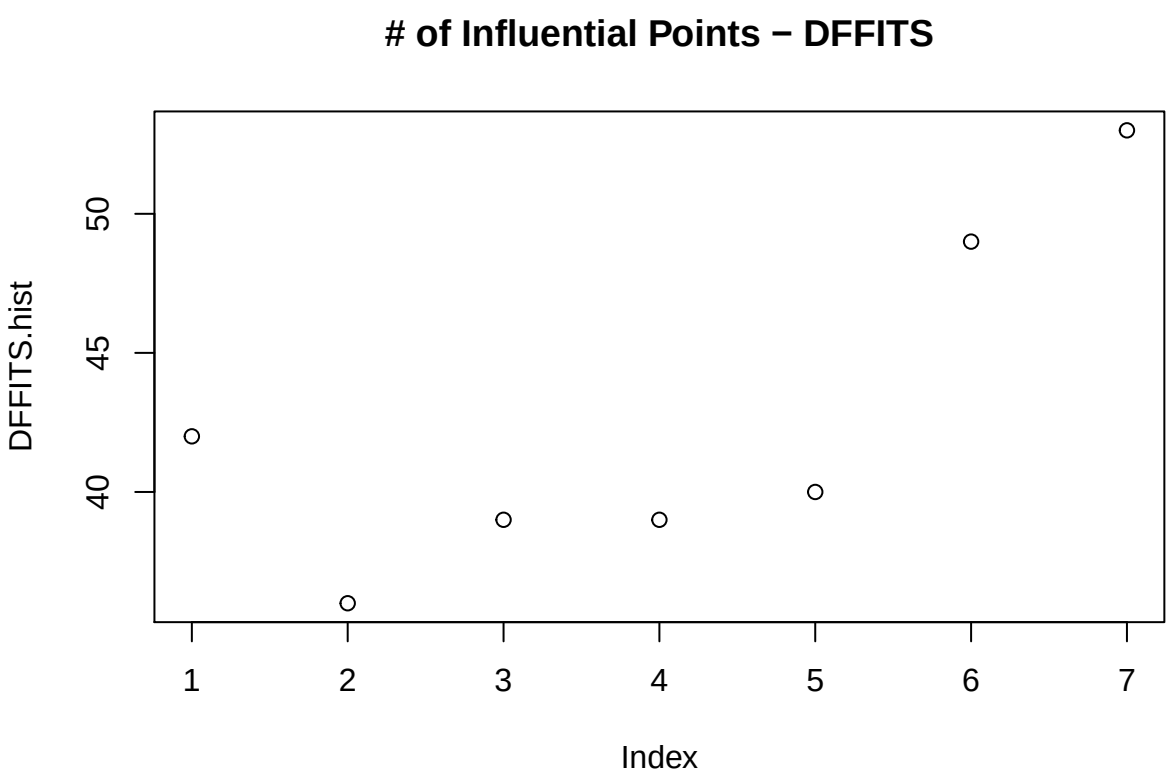
```
    }
  }
  return(list(cv=cv.hist, coef=coef.hist, aic=aic.hist,
              outliers=outliers.hist, DFFITS = DFFITS.hist,
              models = models))
}
```

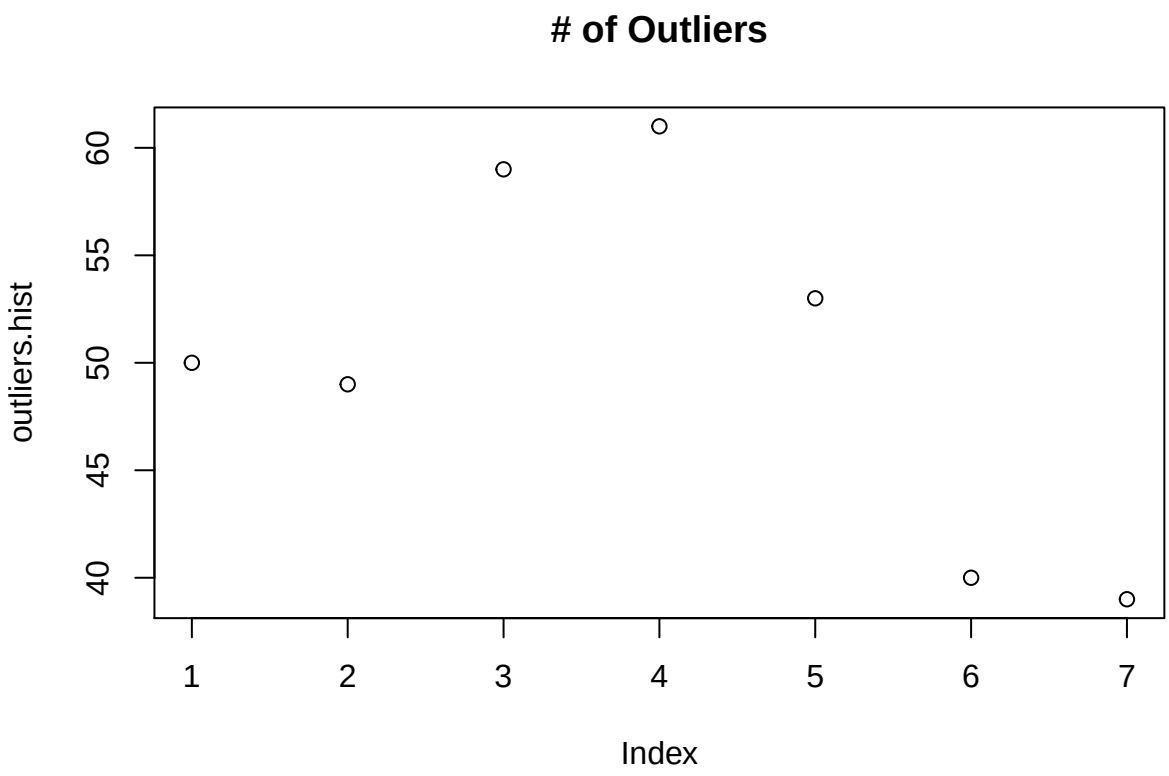
```
# we will analysis these
set.seed(seed)

# log transform
newdata=data
newdata$length <- log(newdata$length)
newdata[,po.ind] = log(newdata[,po.ind])

# start from chosen pollutens
chosen.pos = colnames(newdata)[chosen.po.ind]
expr = paste("length~", paste(chosen.pos, collapse = "+"))
lasso.pollu.model = (lm(expr,data=newdata))


t=forward.change(newdata, expr, TRUE)
```

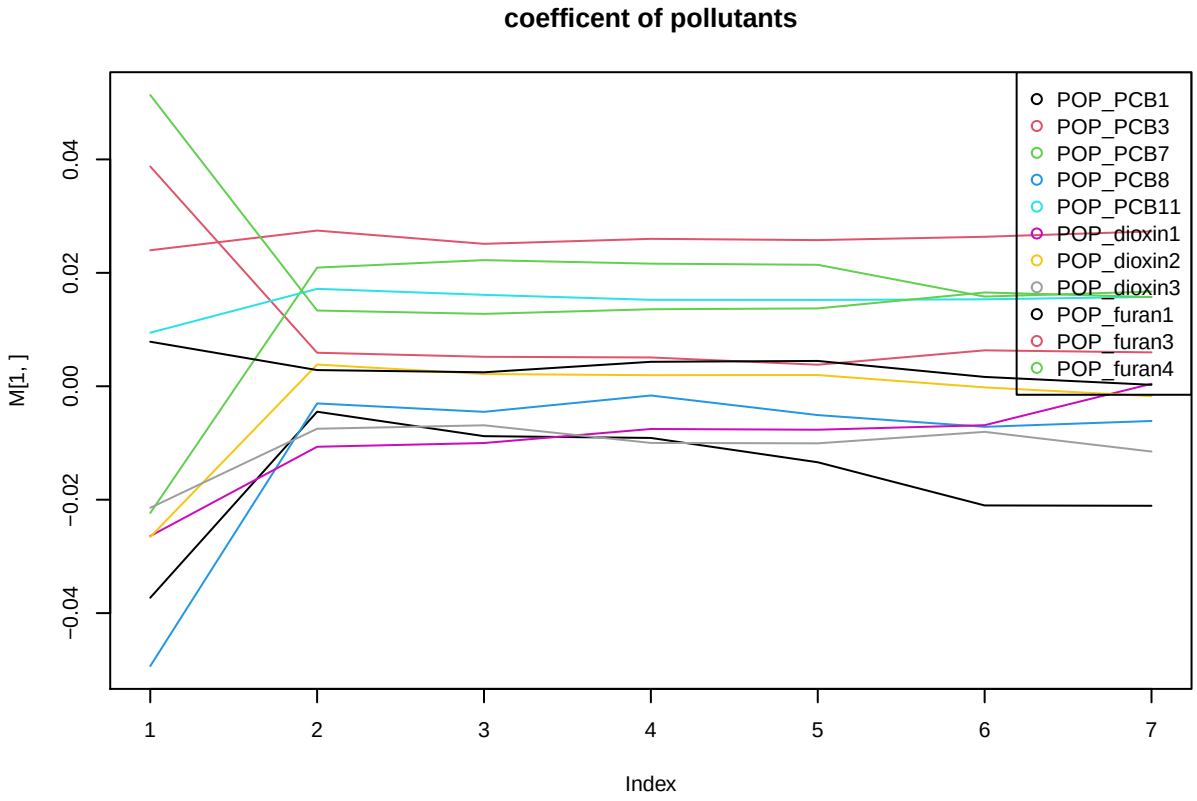
```
## [1] "added ageyrs"
## [1] "added POP_PCB10"
## [1] "added monocyte_pct"
## [1] "added POP_PCB2"
## [1] "added edu_cat"
## [1] "added smokenow"
```


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


# of Influential Points − DFFITS
DFFITS.hist
40
45
50
1
2
3
4
5
6
7
Index


# of Outliers
outliers.hist
40
45
50
55
60
1
2
3
4
5
6
7
Index




```
# the last step vary by a lot becasue large aif -> large variance on beta, we shouldn't consider last s


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

# path of the "error analysis stuff/ cook's distance dffits"

finalModel <- t$models[[2]]

summary(finalModel)
```

```
##
## Call:
## lm(formula = expr, data = newdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58353 -0.13296 -0.00239  0.13048  0.69185
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.1575373  0.1315420   1.198   0.2314
```

```
## POP_PCB1    -0.0044886  0.0163771  -0.274   0.7841
## POP_PCB3     0.0059126  0.0176141   0.336   0.7372
## POP_PCB7     0.0209225  0.0181920   1.150   0.2504
## POP_PCB8    -0.0030293  0.0195994  -0.155   0.8772
## POP_PCB11    0.0171819  0.0097167   1.768   0.0774 .
## POP_dioxin1 -0.0106703  0.0143878  -0.742   0.4585
## POP_dioxin2  0.0038354  0.0128612   0.298   0.7656
## POP_dioxin3 -0.0074880  0.0161115  -0.465   0.6422
## POP_furan1   0.0028556  0.0184173   0.155   0.8768
## POP_furan3   0.0274499  0.0118849   2.310   0.0211 *
## POP_furan4   0.0133555  0.0125493   1.064   0.2875
## ageyrs      -0.0074950  0.0006772 -11.067   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2039 on 851 degrees of freedom
## Multiple R-squared:  0.2483, Adjusted R-squared:  0.2377
## F-statistic: 23.43 on 12 and 851 DF,  p-value: < 2.2e-16
```

```
summary(lm(data$length~data$POP_furan4))
```

```
##
## Call:
## lm(formula = data$length ~ data$POP_furan4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52661 -0.17867 -0.02668  0.15557  1.29734
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.0584473  0.0122963  86.079   <2e-16 ***
## data$POP_furan4 -0.0003581  0.0007682  -0.466    0.641
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2504 on 862 degrees of freedom
## Multiple R-squared:  0.0002521,  Adjusted R-squared:  -0.0009077
## F-statistic: 0.2173 on 1 and 862 DF,  p-value: 0.6412
```

```
finalModel$coefficients[ finalModel$coefficients<0.01]
```

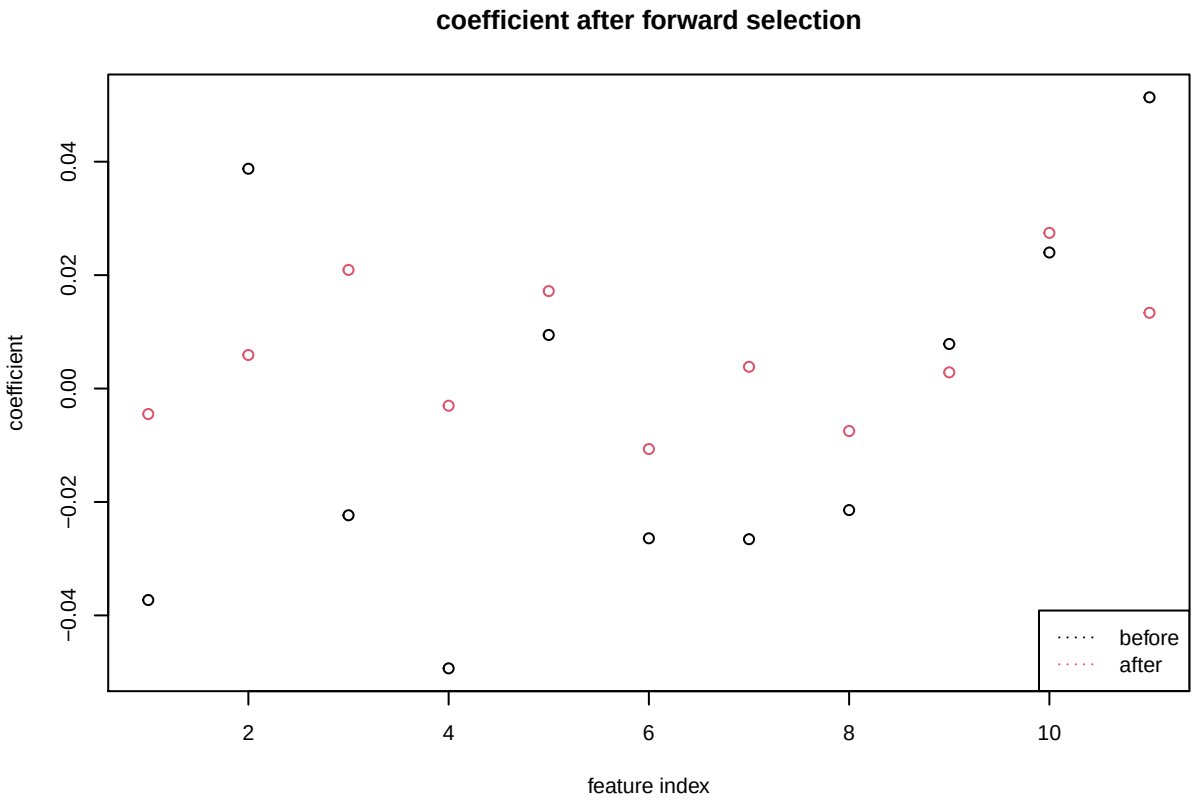
```
##     POP_PCB1     POP_PCB3     POP_PCB8  POP_dioxin1  POP_dioxin2  POP_dioxin3
## -0.004488595  0.005912602 -0.003029345 -0.010670329  0.003835385 -0.007487969
##   POP_furan1       ageyrs
##  0.002855554 -0.007495015
```

```
finalModel$coefficients[ finalModel$coefficients>0.01]
```

```
## (Intercept)    POP_PCB7   POP_PCB11  POP_furan3  POP_furan4
##  0.15753730  0.02092253  0.01718193  0.02744992  0.01335549
```

```
#vif(finalModel)


numpara= length(lasso.pollu.model$coefficients)-1
# excluding length in full model
```

```
plot(rep(1:numpara,2),  c(lasso.pollu.model$coefficients[-1],finalModel$coefficients[-c(1,numpara+2)] )
     col=rep(c(1,2), each=numpara), xlab = "feature index", ylab="coefficient", main = "coefficient aft
legend("bottomright", legend=c("before", "after"), col=c(1,2), lty=3)
```




```
coef(lasso.pollu.model)
```

```
##  (Intercept)     POP_PCB1     POP_PCB3     POP_PCB7     POP_PCB8    POP_PCB11
##  0.823419273 -0.037291353  0.038756534 -0.022342147 -0.049337110  0.009449720
##  POP_dioxin1  POP_dioxin2  POP_dioxin3   POP_furan1   POP_furan3   POP_furan4
## -0.026408081 -0.026566899 -0.021431406  0.007854683  0.023985959  0.051351220
```

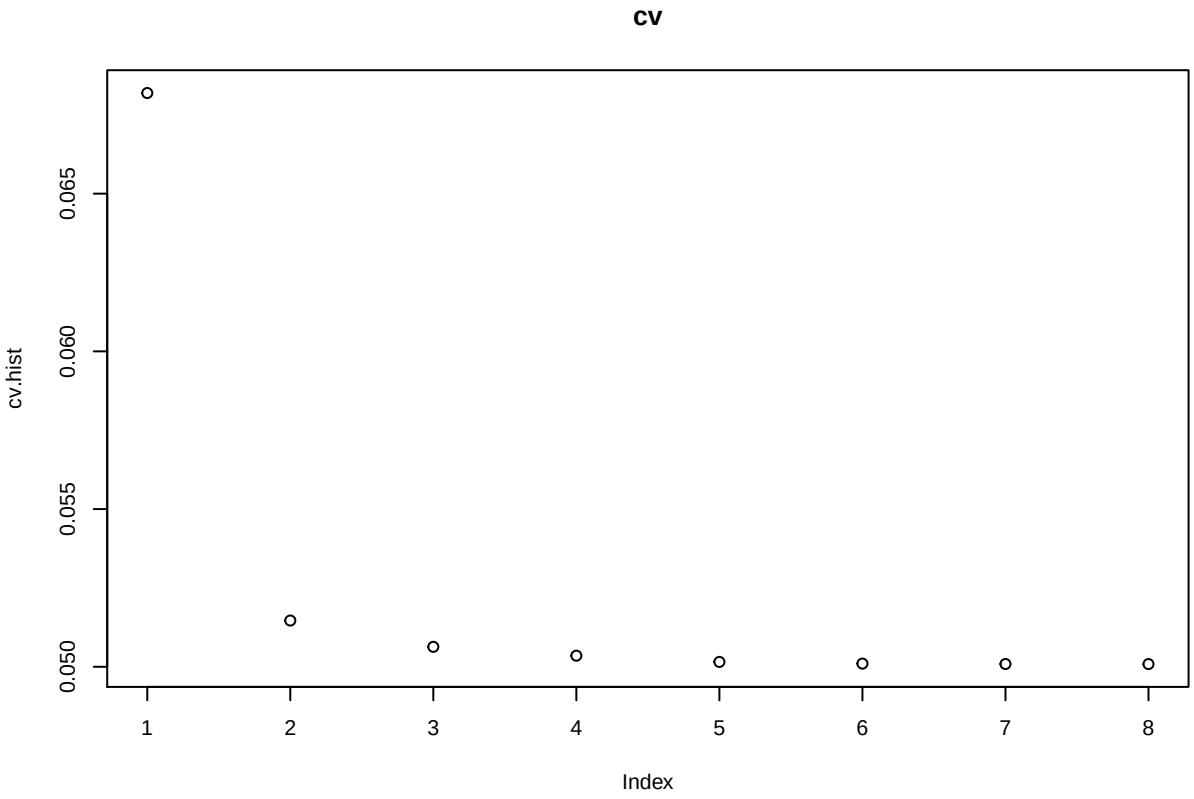
```
coef(finalModel)
```

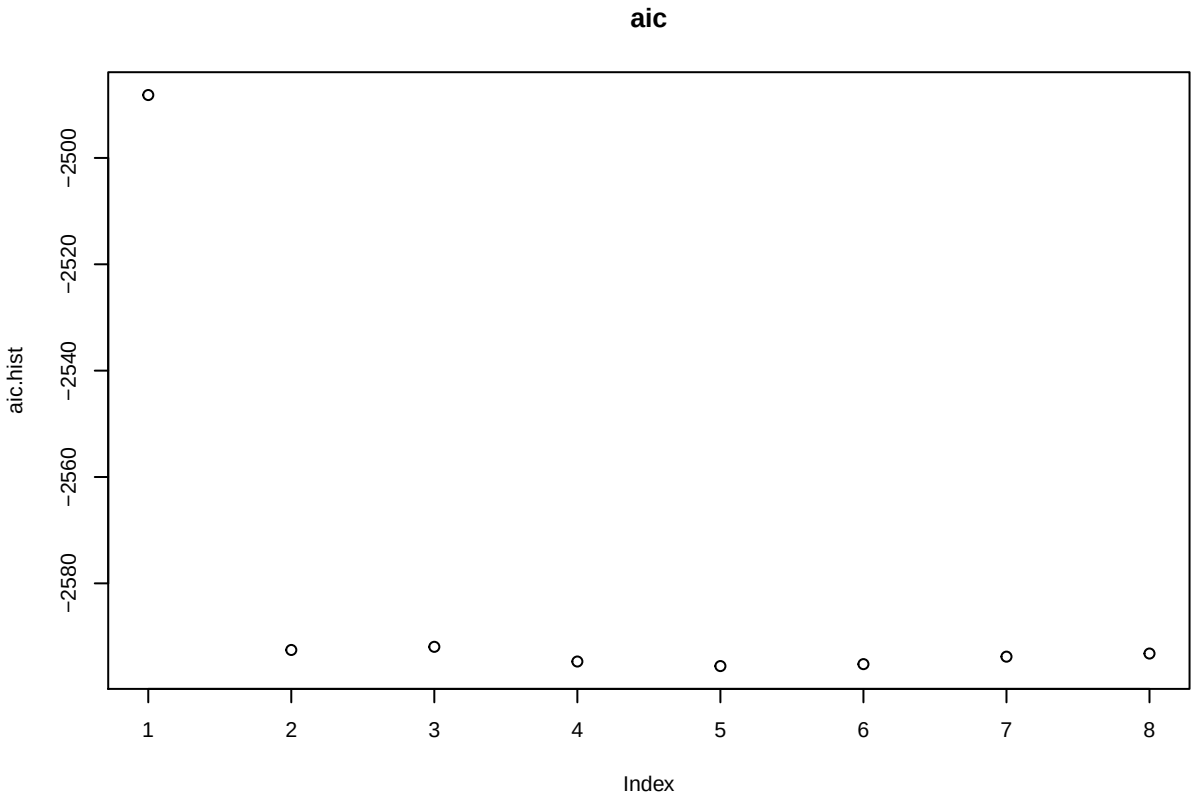
```
##  (Intercept)     POP_PCB1     POP_PCB3     POP_PCB7     POP_PCB8    POP_PCB11
##  0.157537300 -0.004488595  0.005912602  0.020922531 -0.003029345  0.017181935
##  POP_dioxin1  POP_dioxin2  POP_dioxin3   POP_furan1   POP_furan3   POP_furan4
## -0.010670329  0.003835385 -0.007487969  0.002855554  0.027449917  0.013355489
##       ageyrs
## -0.007495015
```

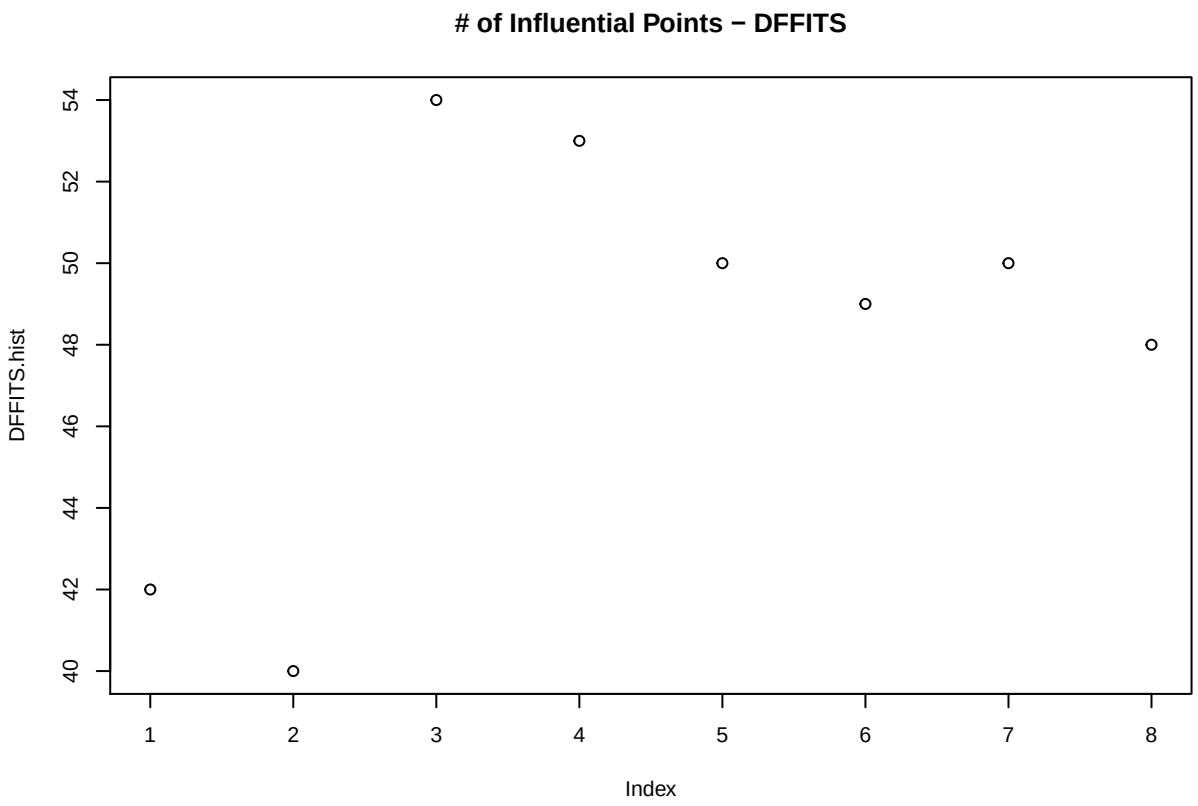
```
newdata=data
newdata[,po.ind] = log(newdata[,po.ind])
chosen.pos = colnames(newdata)[chosen.po.ind]
expr = paste("length~", paste(chosen.pos, collapse = "+"))

t=forward.change(newdata, expr, TRUE)
```

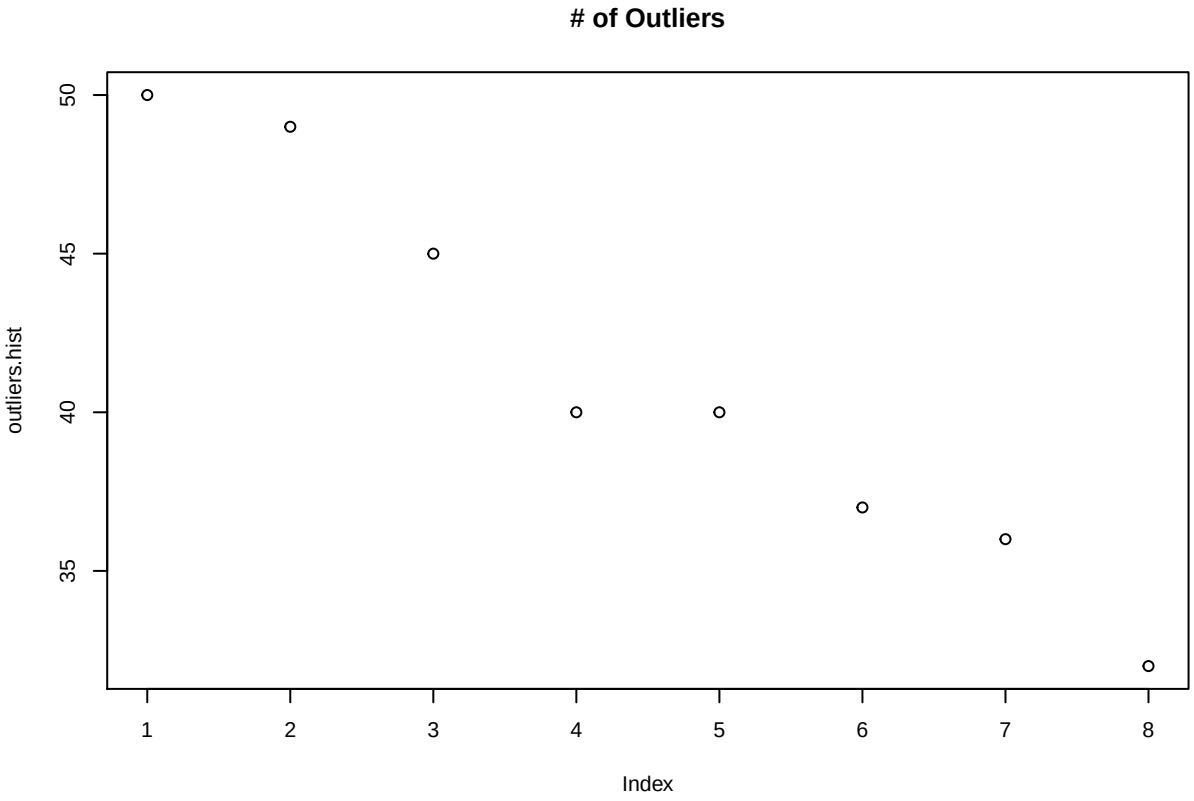
```
## [1] "added ageyrs"
```

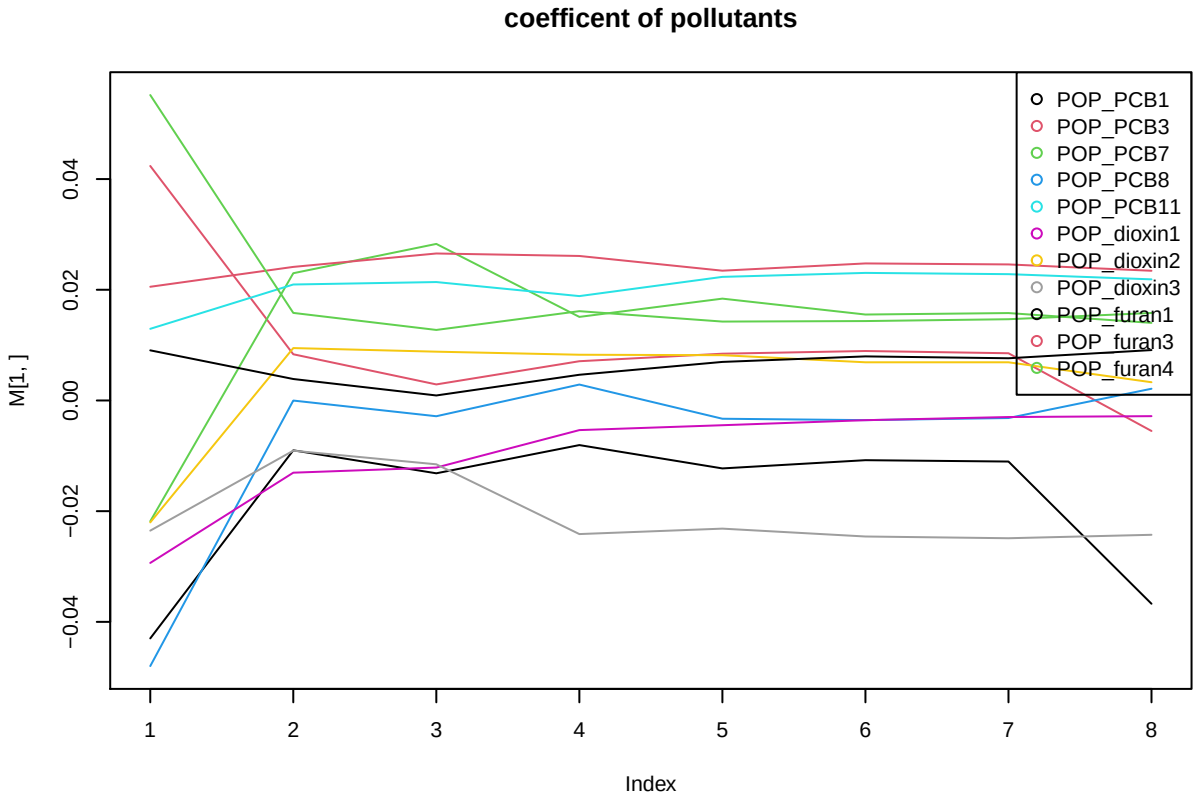
```
## [1] "added race_cat"
## [1] "added male"
## [1] "added BMI"
## [1] "added eosinophils_pct"
## [1] "added neutrophils_pct"
## [1] "added POP_PCB5"
```



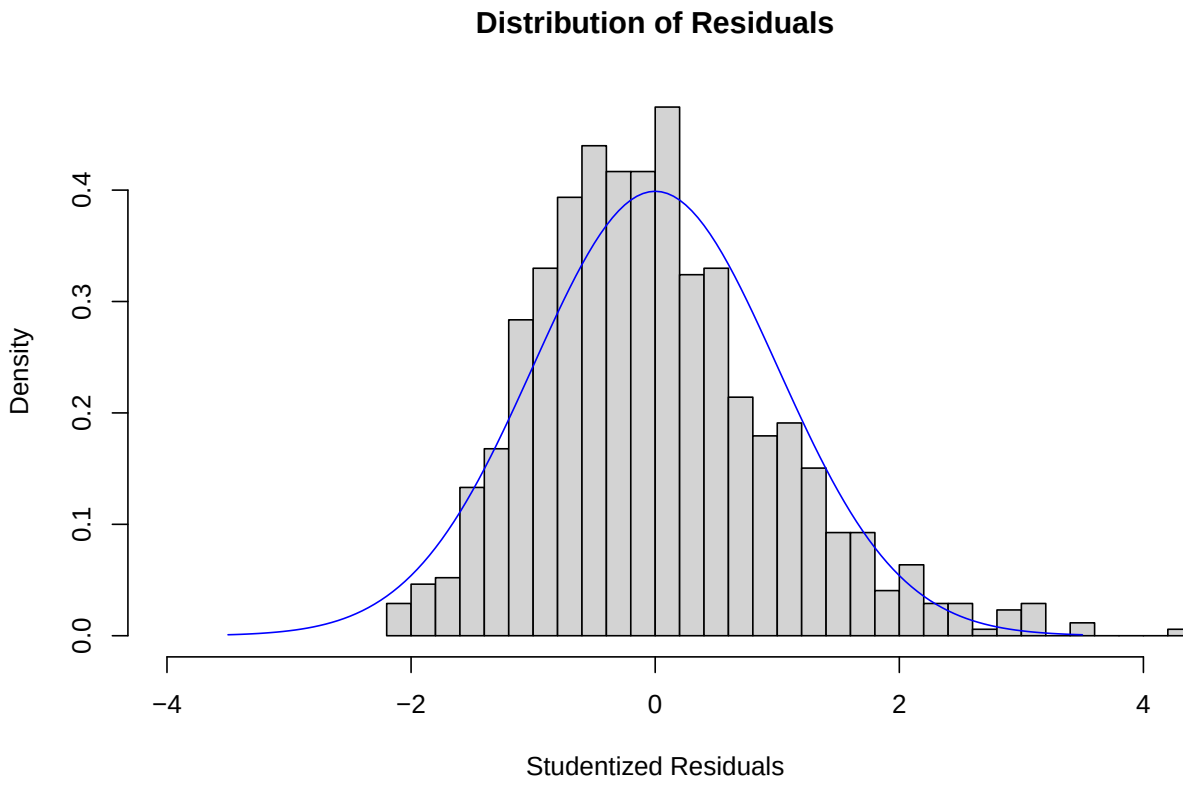

aic
aic.hist
−2500
−2520
−2540
−2560
−2580
1
2
3
4
5
6
7
8
Index

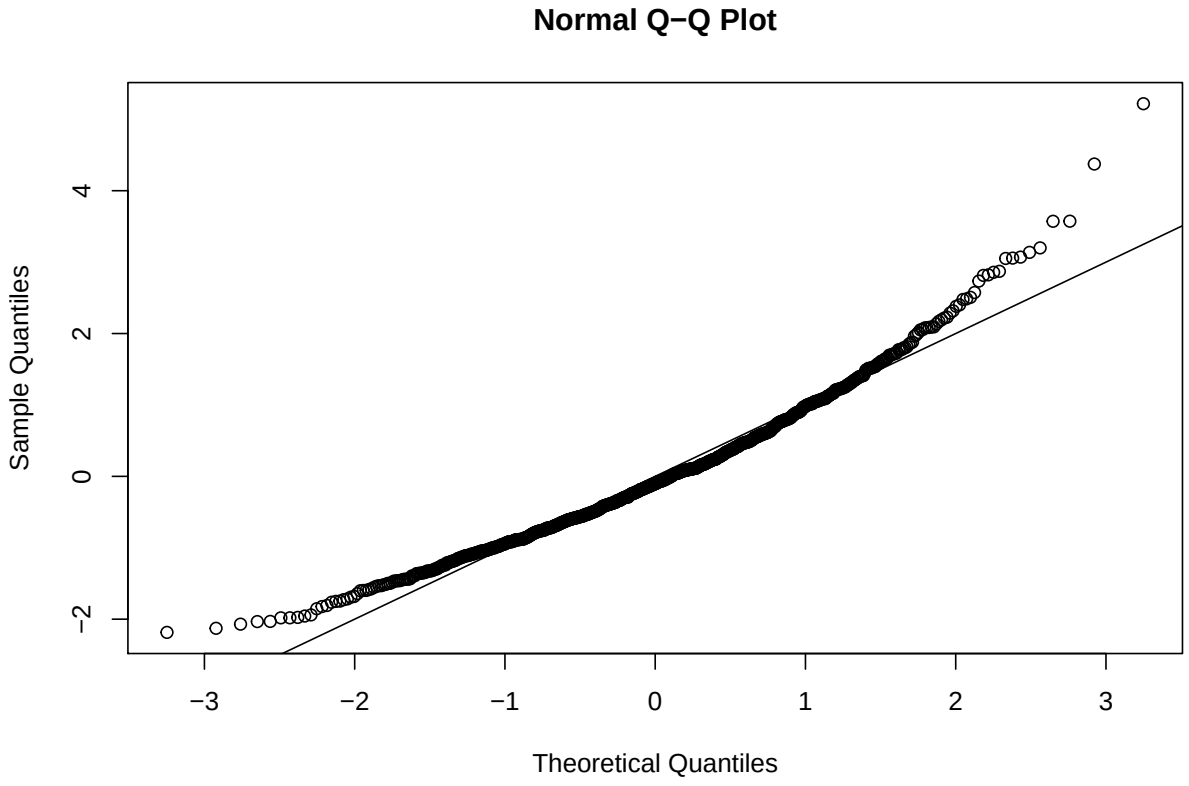

# of Influential Points − DFFITS
DFFITS.hist
40
42
44
46
48
50
52
54
1
2
3
4
5
6
7
8
Index

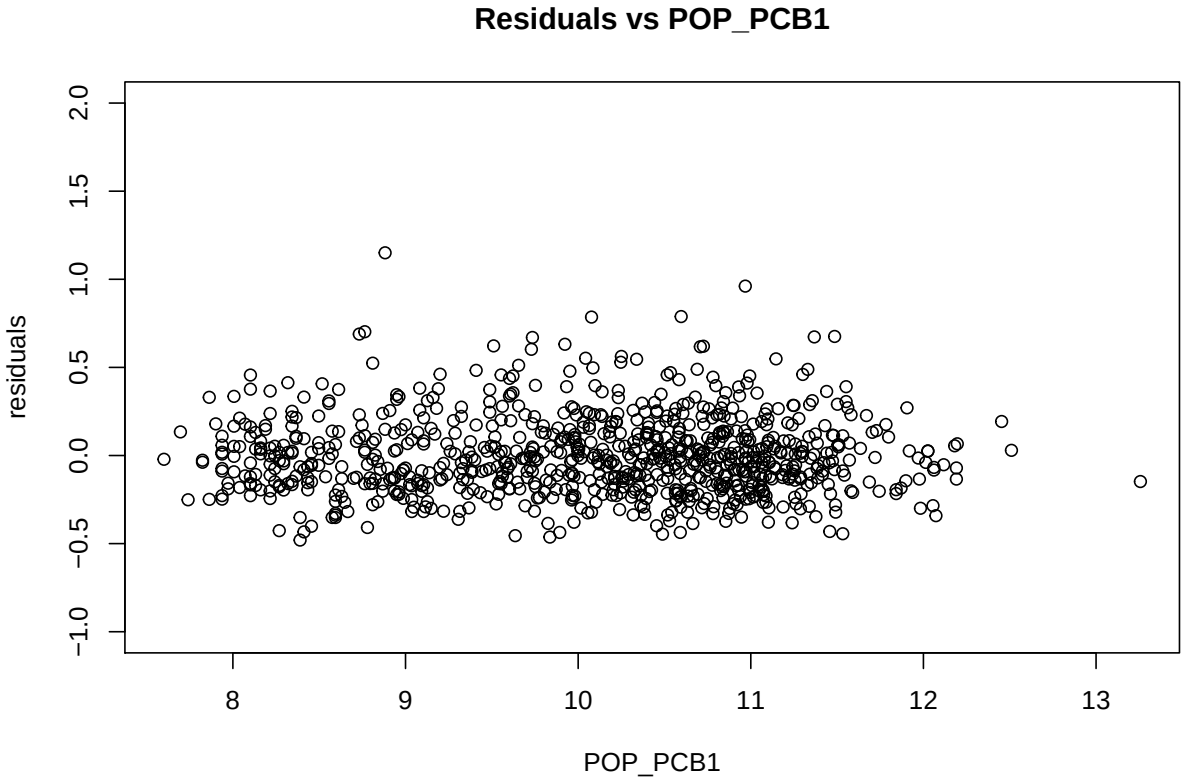

# of Outliers
outliers.hist
50
45
40
35
1
2
3
4
5
6
7
8
Index

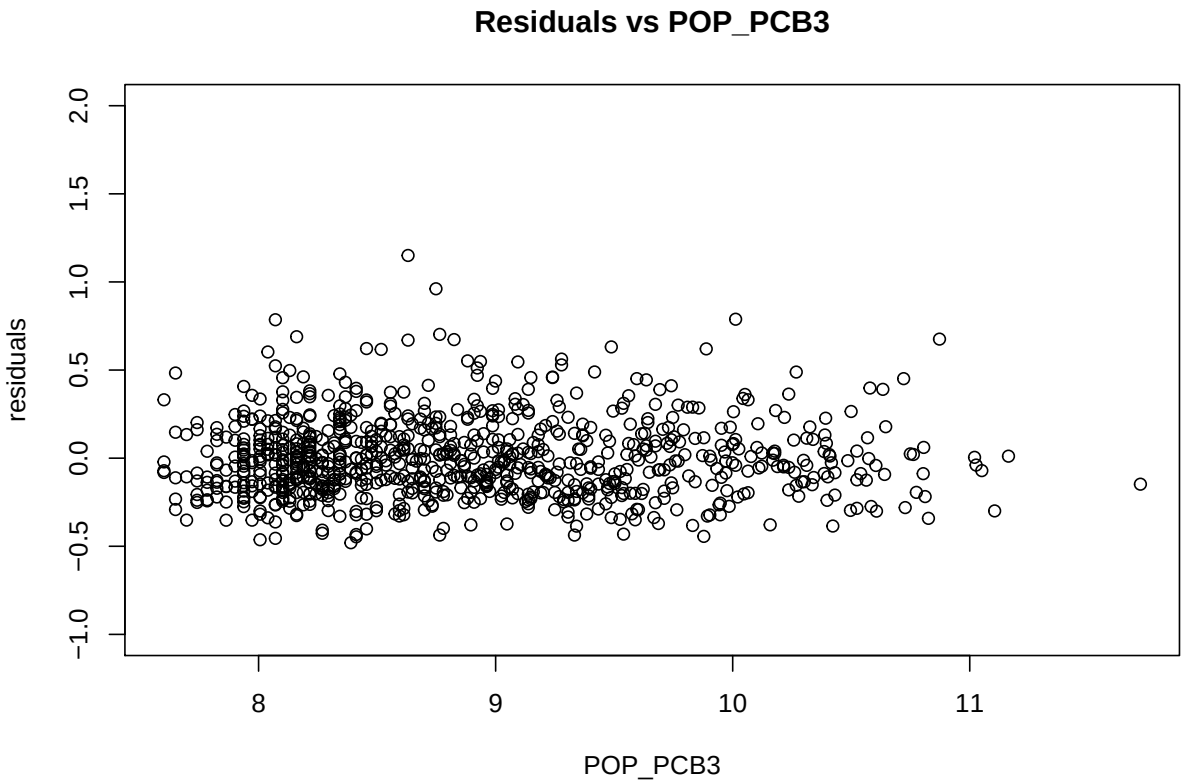


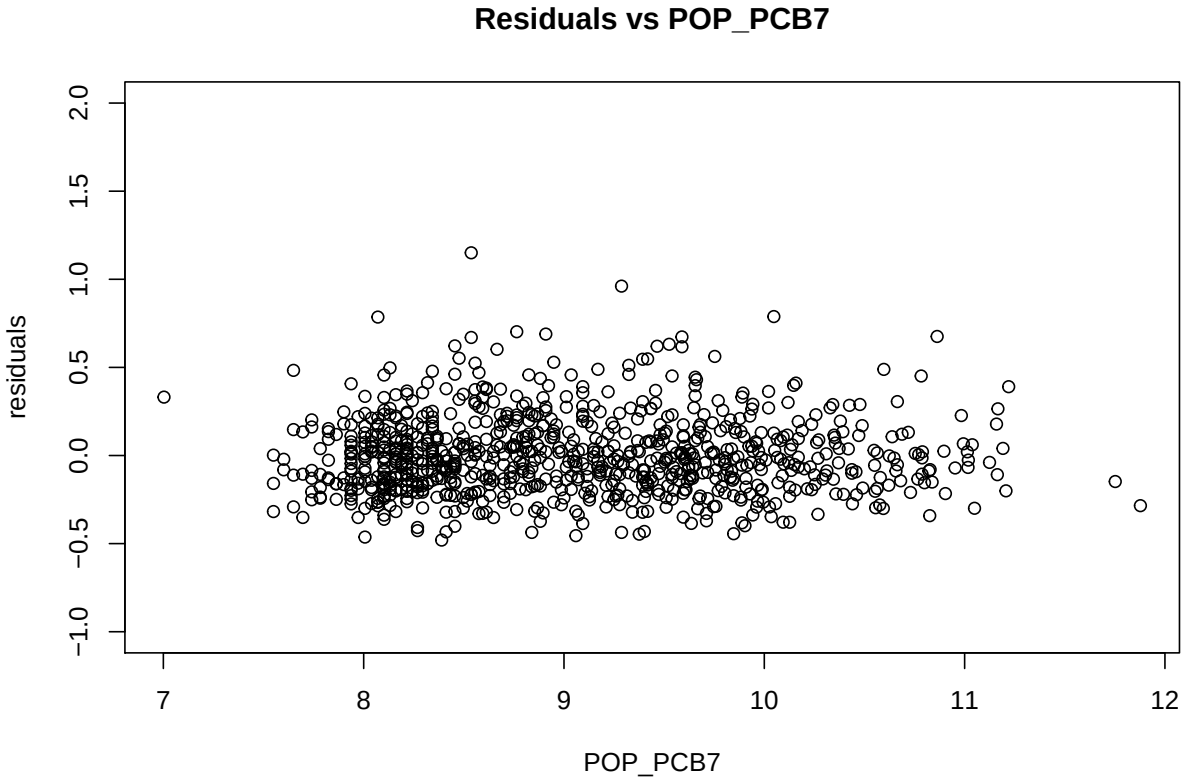

```
errorAnalysis(t$models[[2]])
```

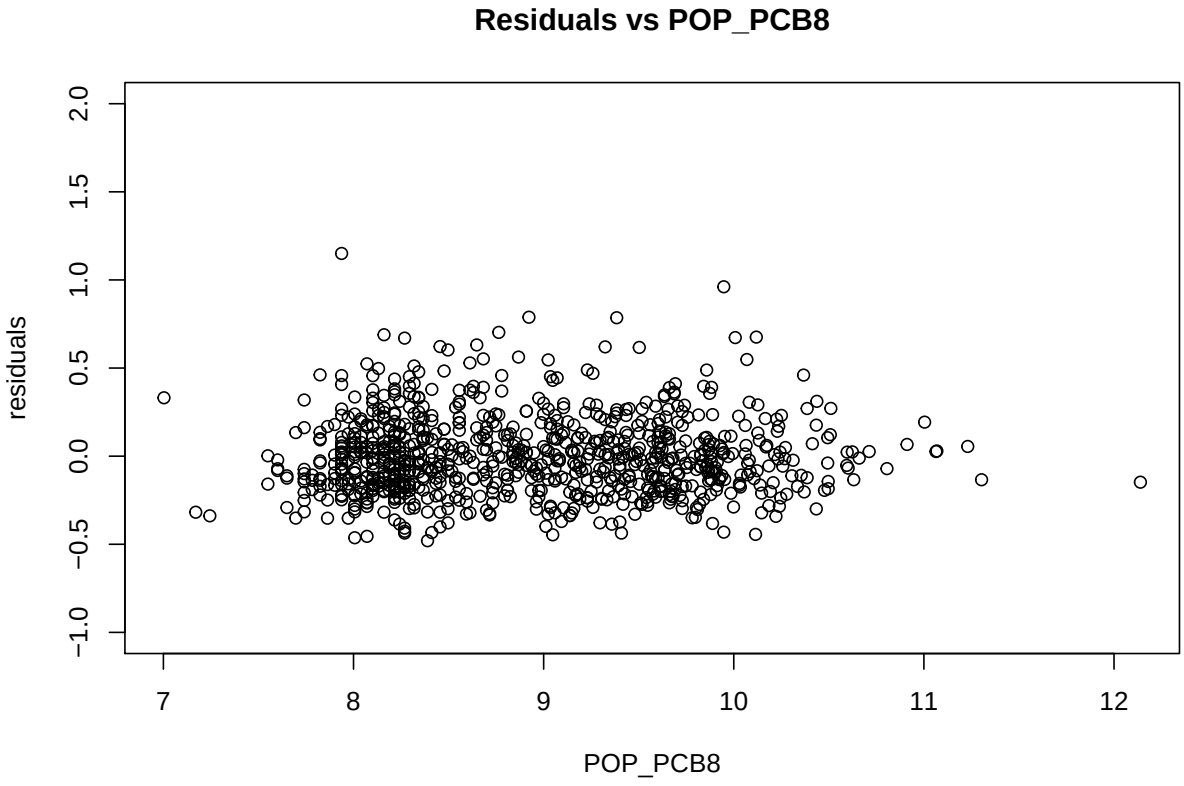

Distribution of Residuals
Density
0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
4
Studentized Residuals

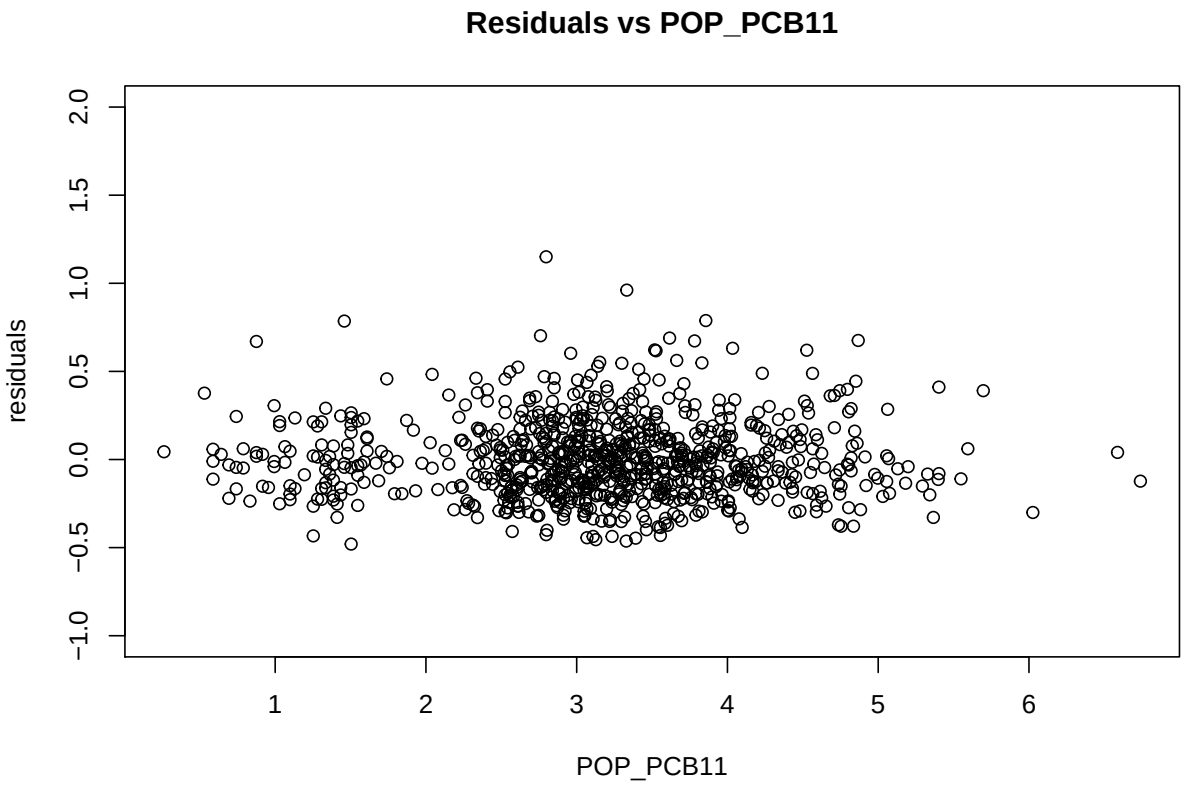

Normal Q−Q Plot
Sample Quantiles
−2
0
2
4
−3
−2
−1
0
1
2
3
Theoretical Quantiles

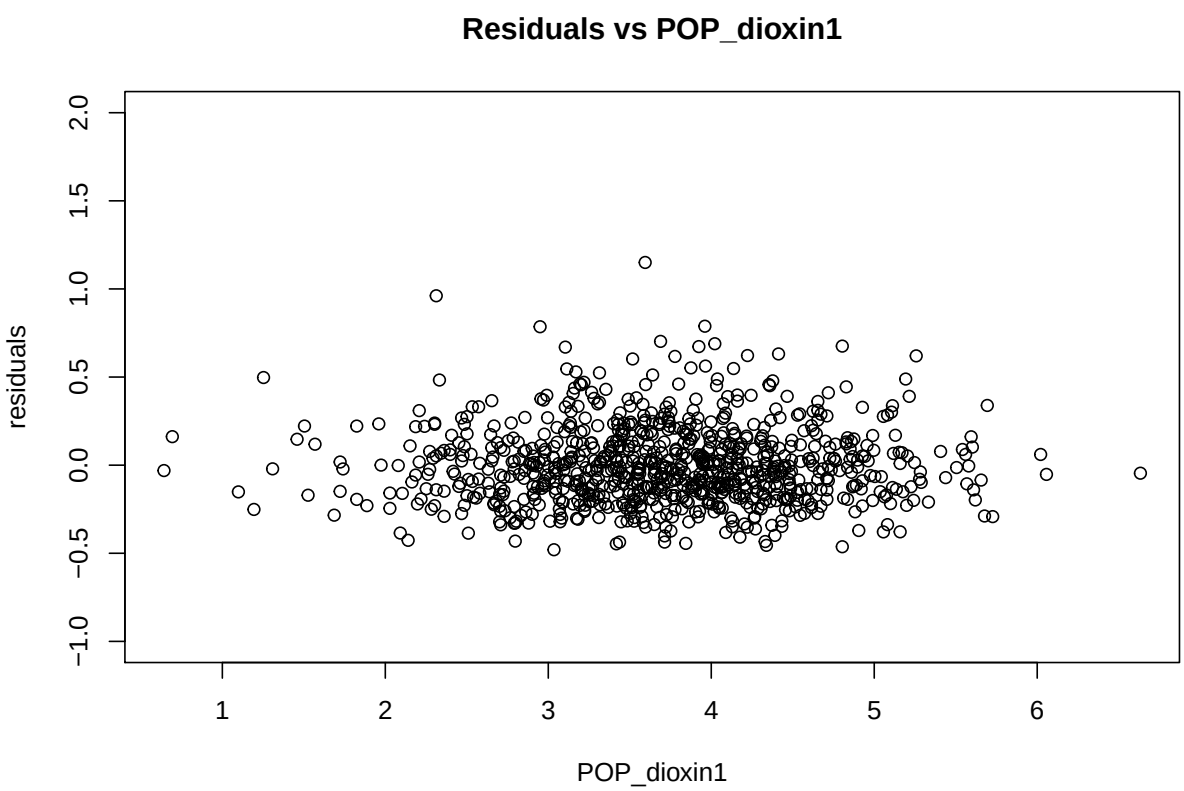

Residuals vs POP_PCB1
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
8
9
10
11
12
13
POP_PCB1

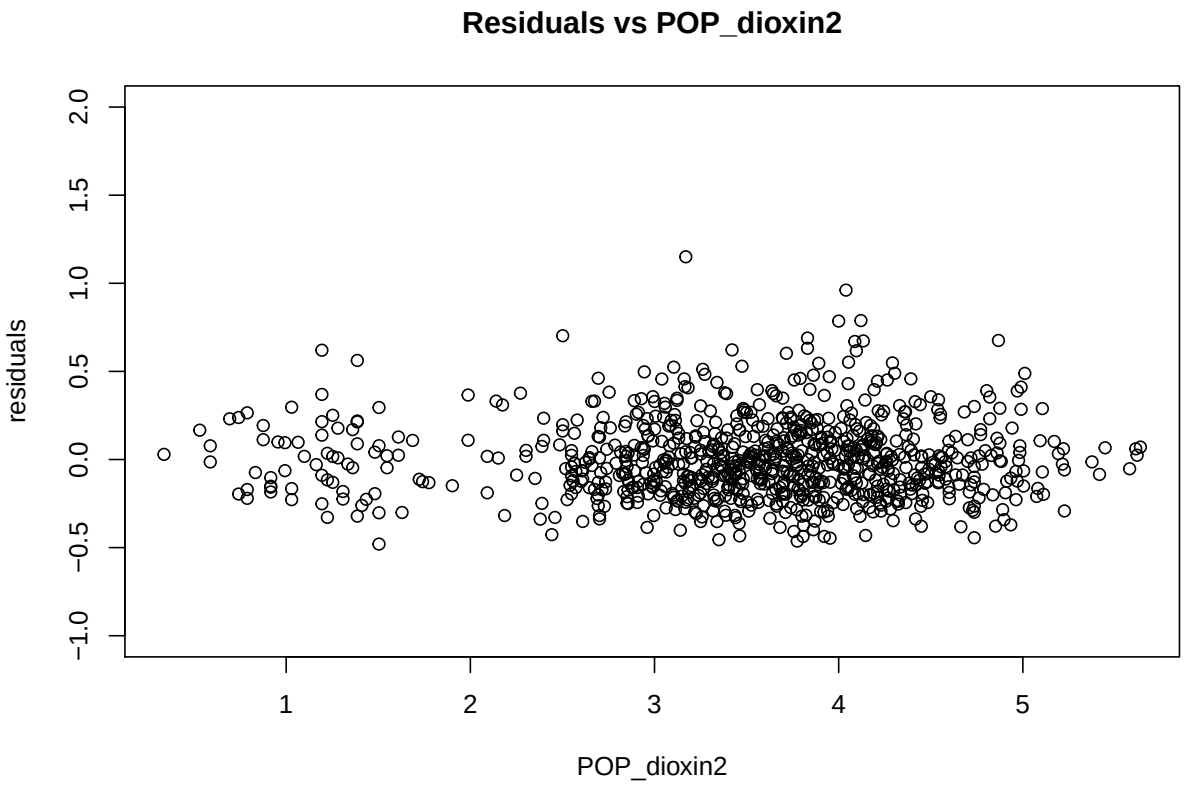

Residuals vs POP_PCB3
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
8
9
10
11
POP_PCB3

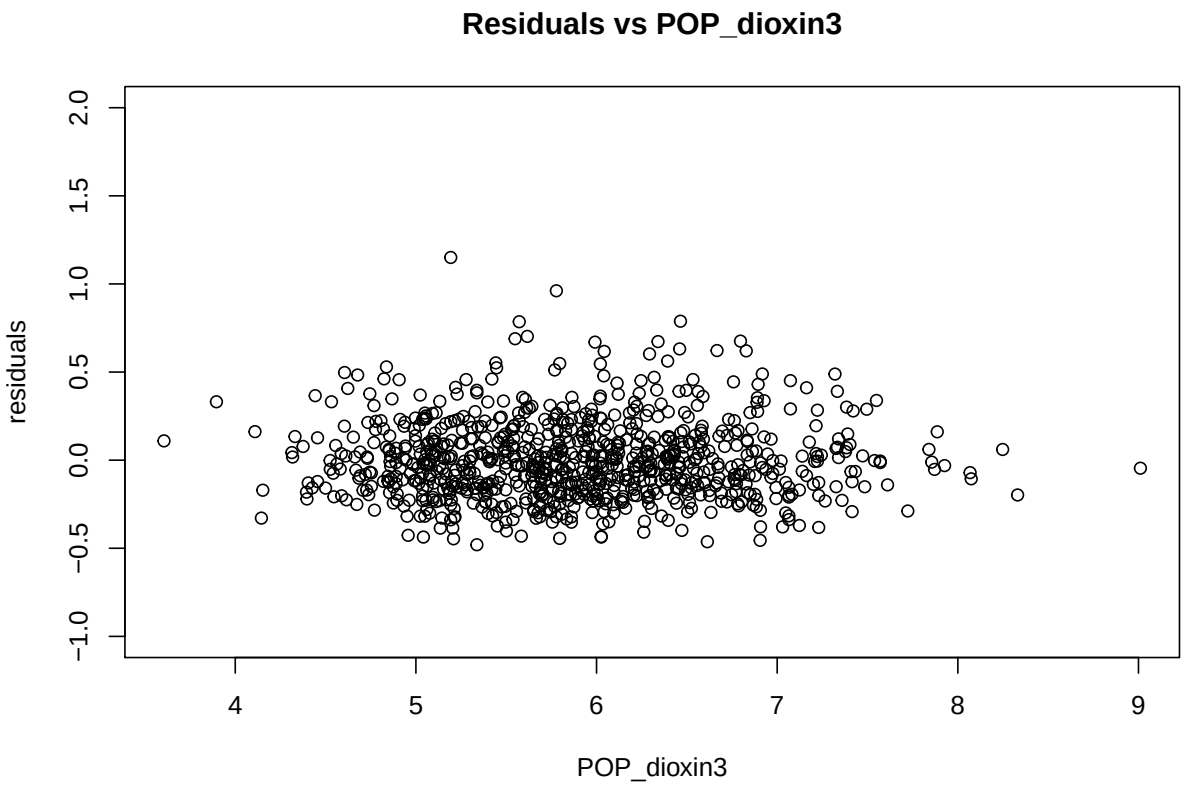

Residuals vs POP_PCB7
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
7
8
9
10
11
12
POP_PCB7

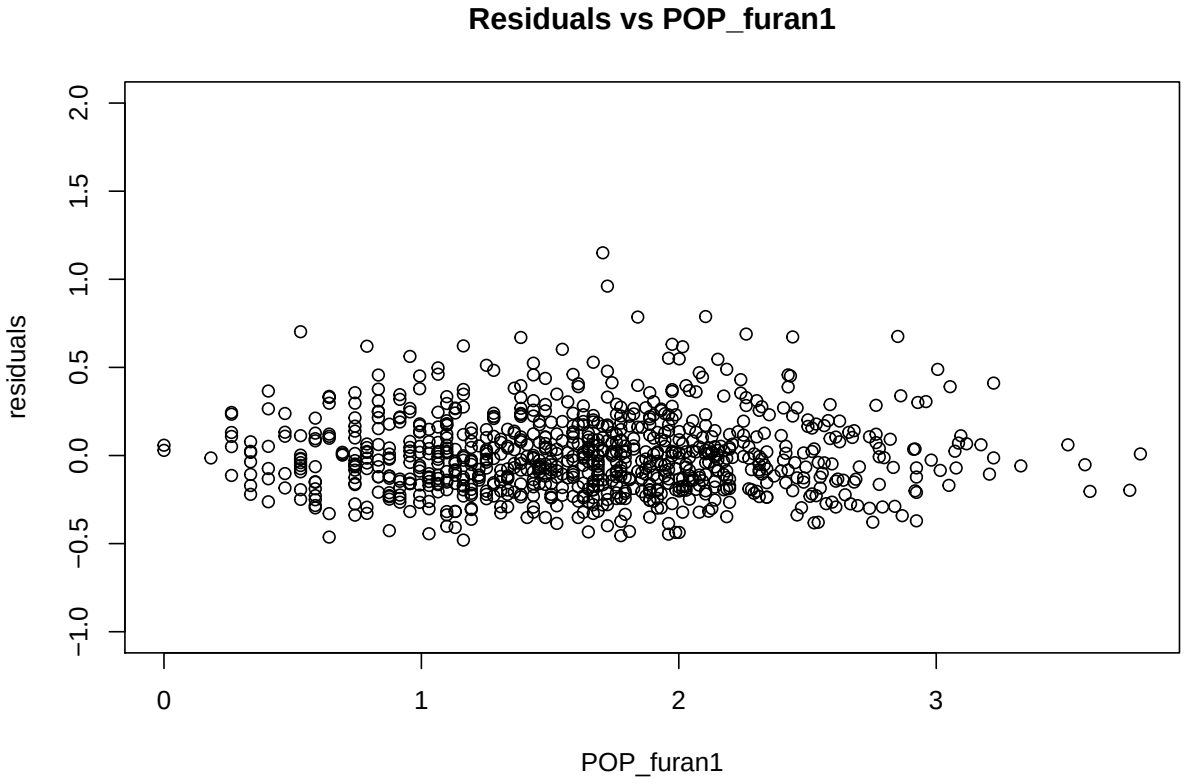

Residuals vs POP_PCB8
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
7
8
9
10
11
12
POP_PCB8

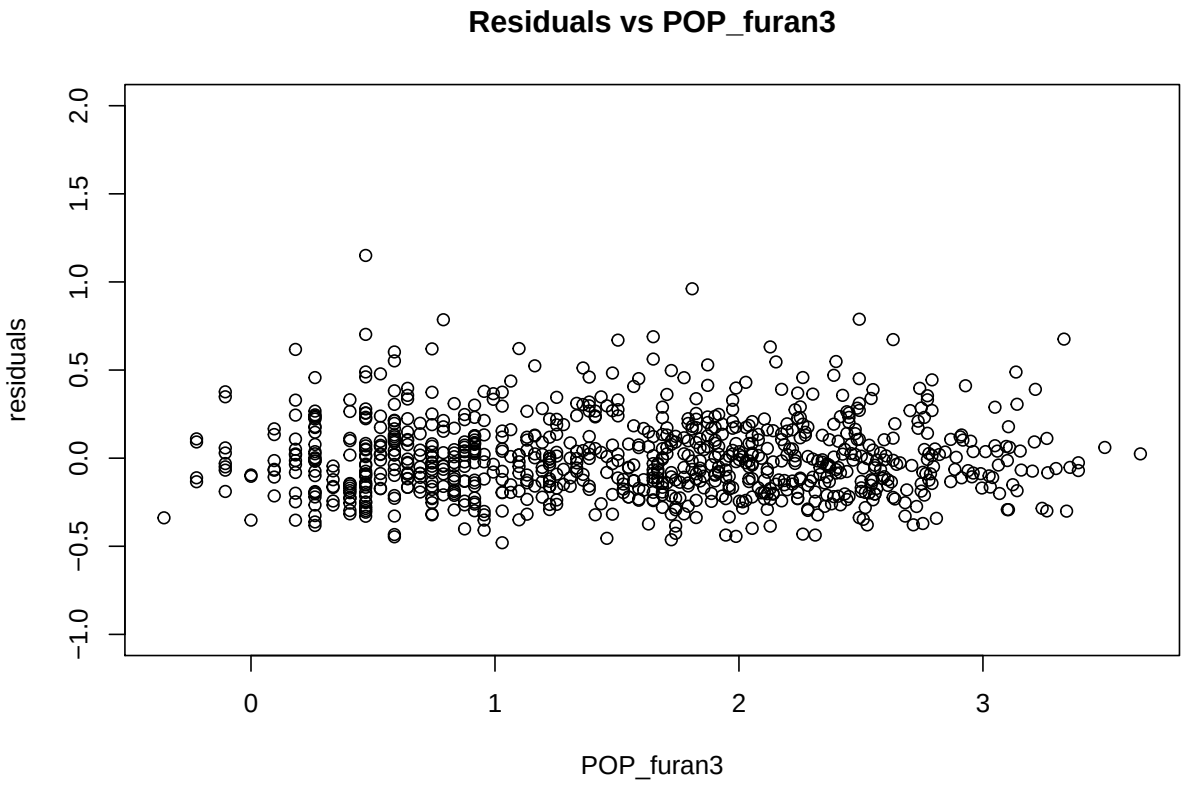

Residuals vs POP_PCB11
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
1
2
3
4
5
6
POP_PCB11

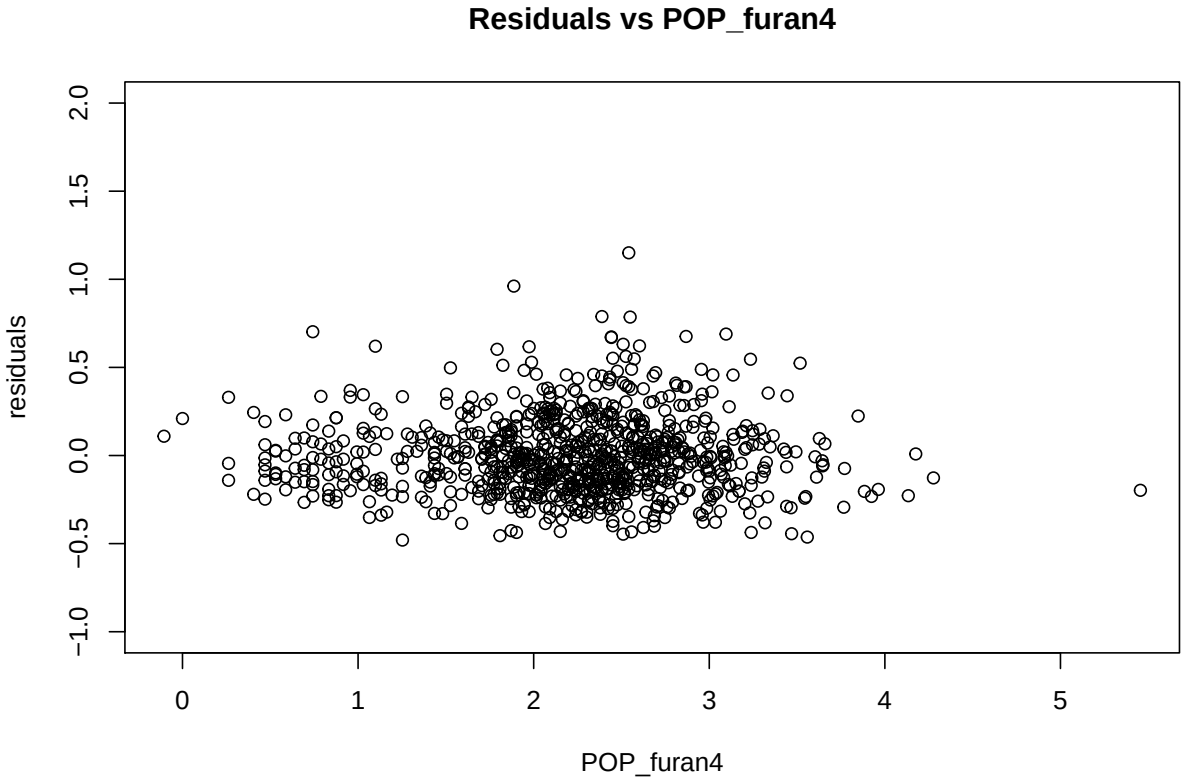

Residuals vs POP_dioxin1
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
1
2
3
4
5
6
POP_dioxin1

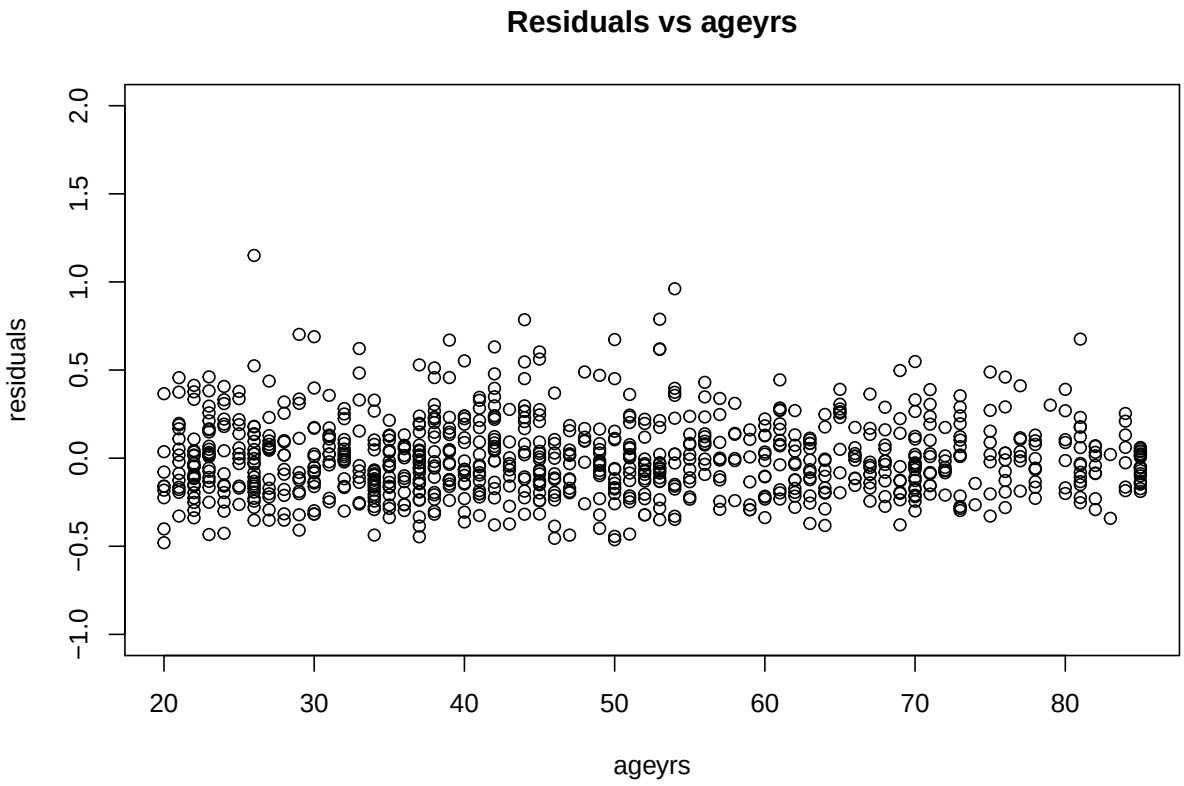

Residuals vs POP_dioxin2
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
1
2
3
4
5
POP_dioxin2

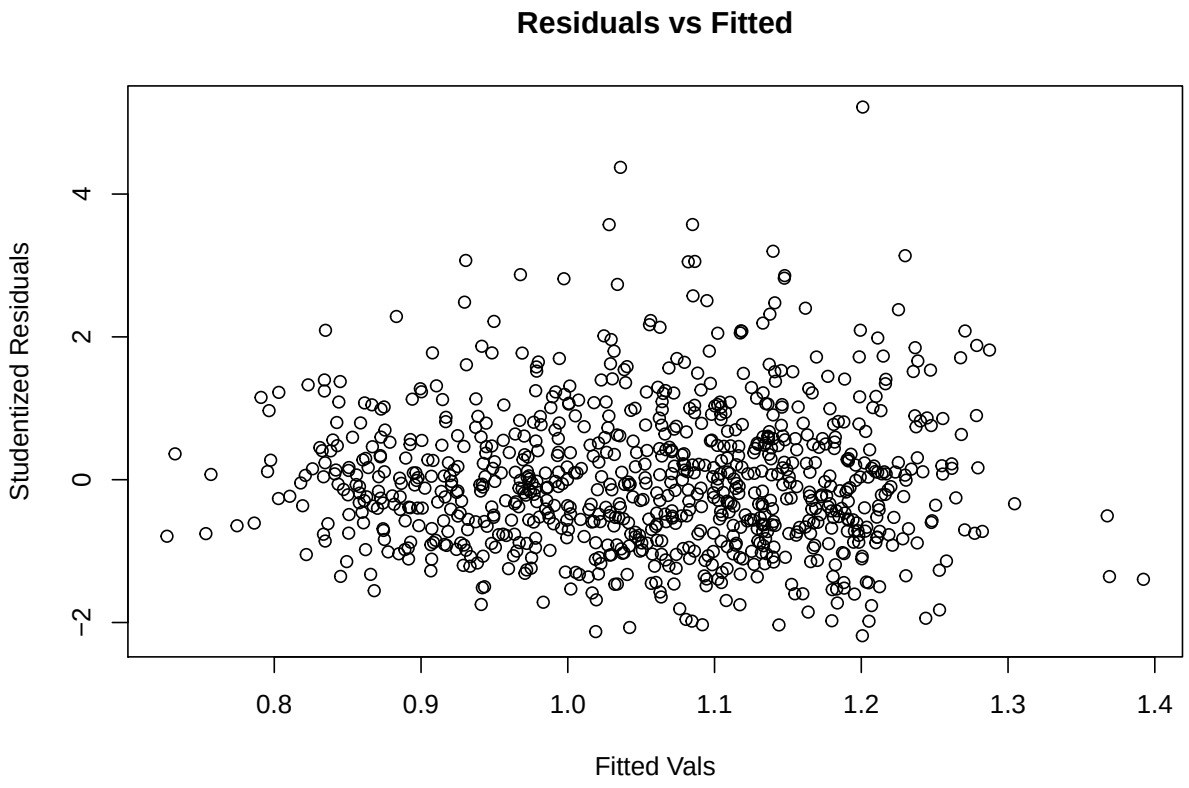

Residuals vs POP_dioxin3
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
4
5
6
7
8
9
POP_dioxin3


Residuals vs POP_furan1
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
POP_furan1

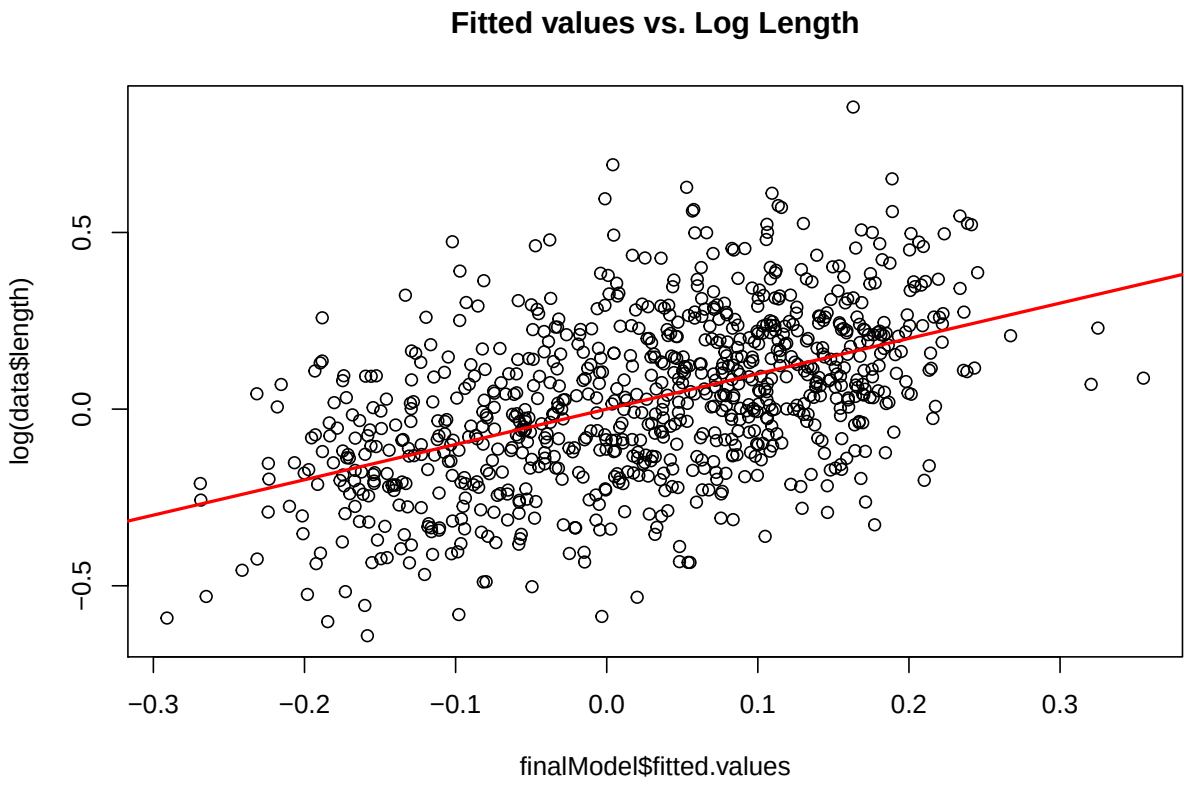

Residuals vs POP_furan3
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
POP_furan3


Residuals vs POP_furan4
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
POP_furan4


Residuals vs ageyrs
residuals
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
20
30
40
50
60
70
80
ageyrs



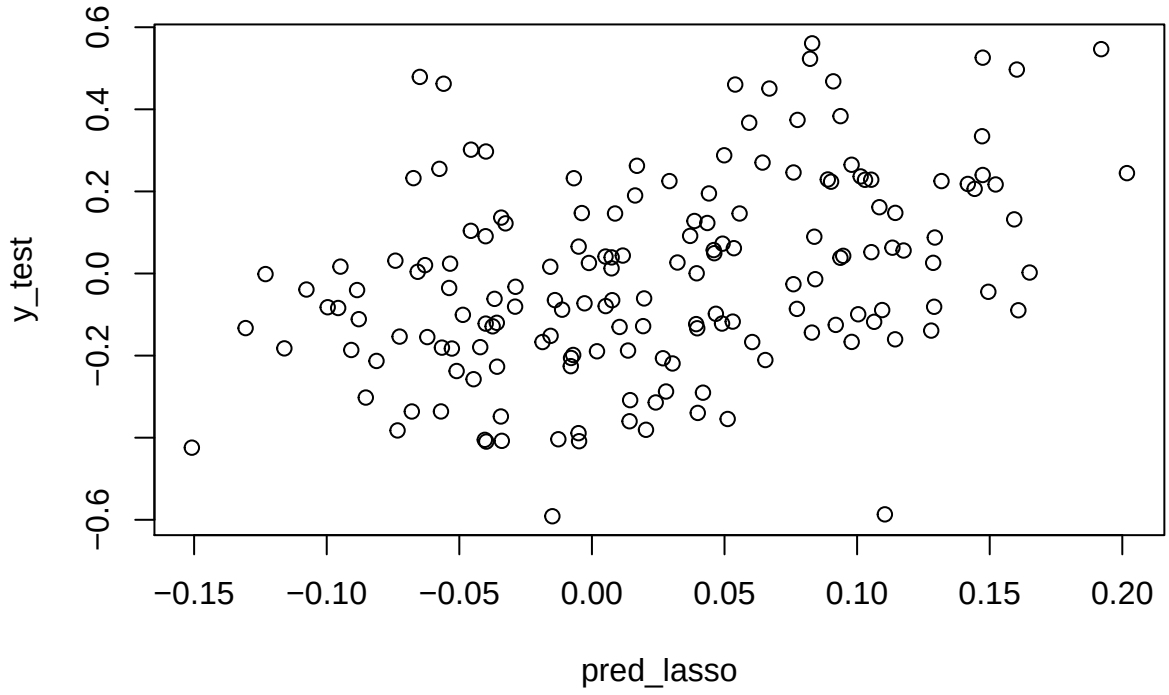

```
plot(x = finalModel$fitted.values, y = log( data$length),
     main = "Fitted values vs. Log Length")
abline(a = 0, b = 1,col = "red", lwd= 2)
```




```
newdata2 <- newdata
newdata2$length <- log(newdata2$length)
```

```
# our final model
newdata=data
newdata[,po.ind]=log(data[,po.ind])
newdata[,1]=log(data[,1])
chosen.po.ind=lasso.on.pollutants(newdata)
```

```
## [1] "mspe 0.0493594027827774"
```

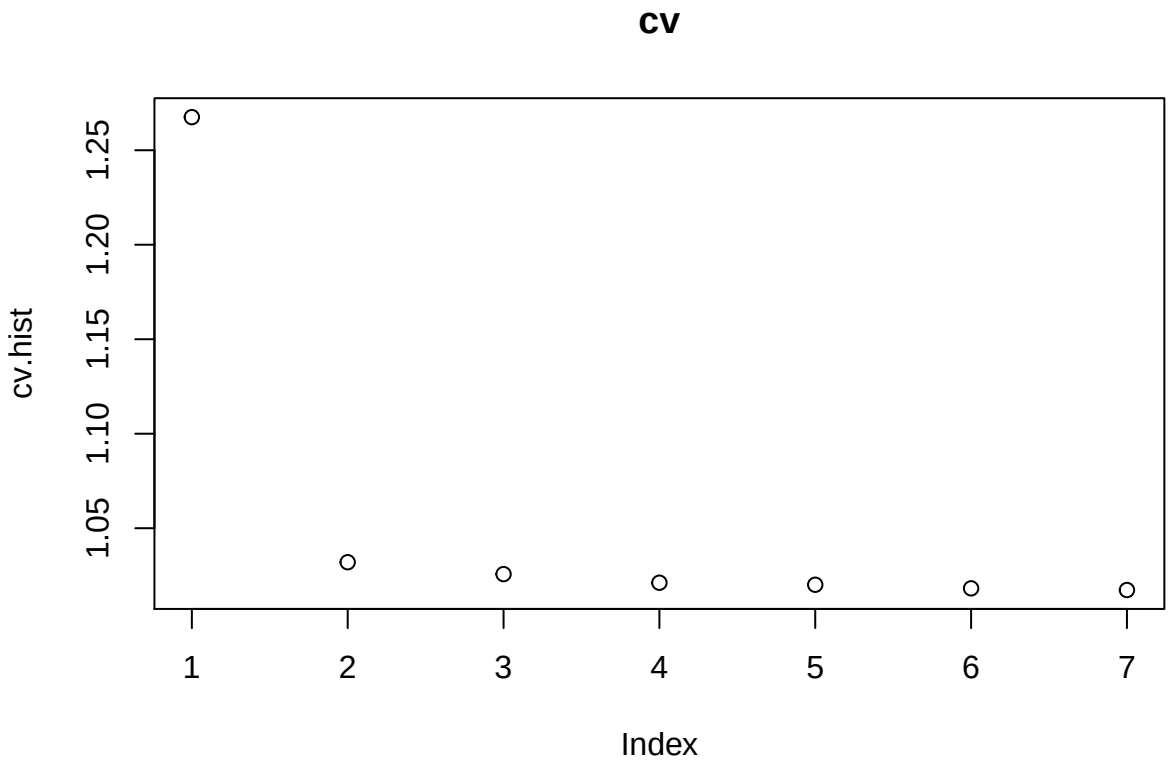


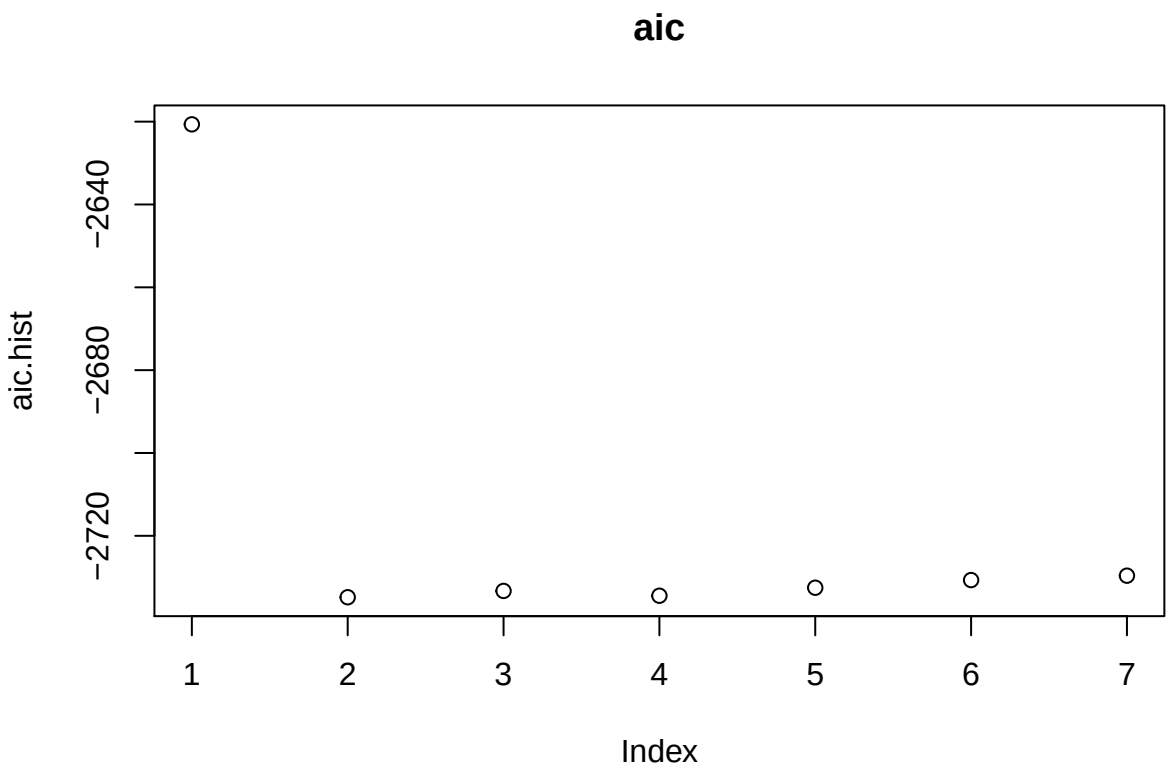

```
chosen.po.ind
```

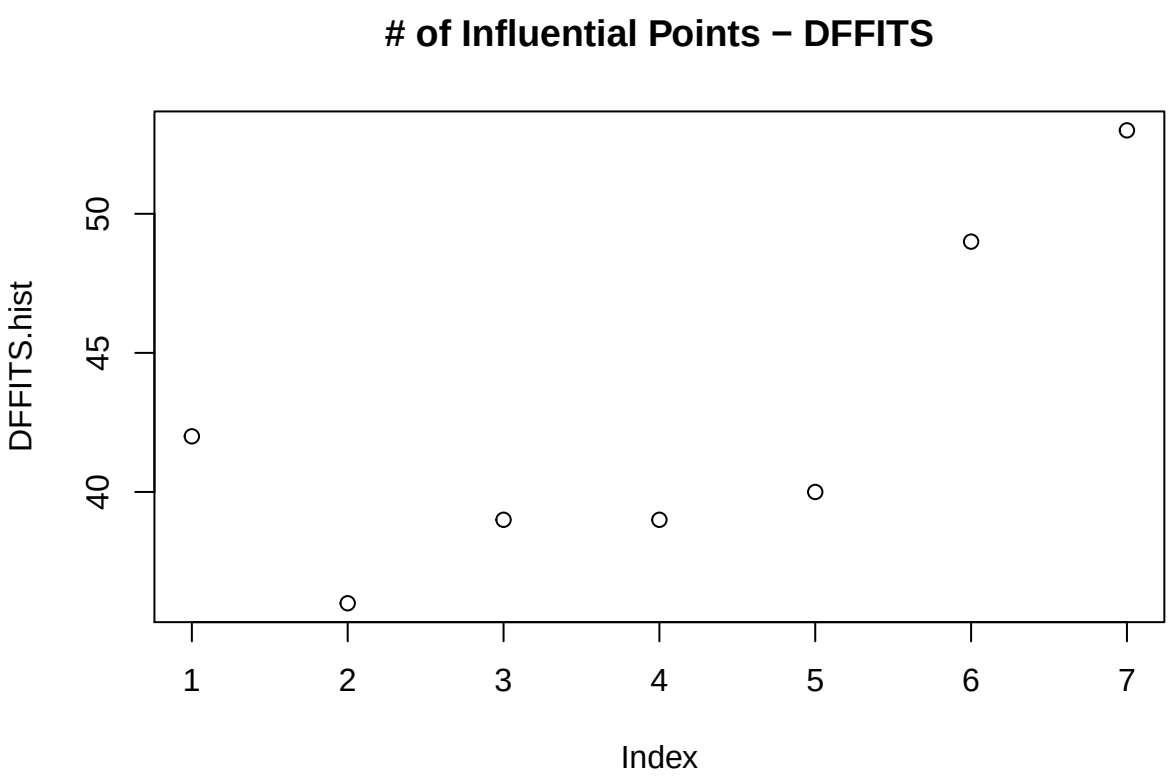
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  7.482254e-01
## POP_PCB1    -1.655992e-02
## POP_PCB2     .
## POP_PCB3     2.082158e-02
## POP_PCB4     .
## POP_PCB5     .
## POP_PCB6     .
## POP_PCB7    -1.195072e-02
## POP_PCB8    -5.626413e-02
## POP_PCB9     .
## POP_PCB10    .
## POP_PCB11    1.449799e-03
## POP_dioxin1 -1.945094e-02
## POP_dioxin2 -1.738420e-02
## POP_dioxin3 -2.059236e-02
## POP_furan1  -3.078953e-05
## POP_furan2   .
## POP_furan3   1.034533e-02
## POP_furan4   4.979942e-02
```

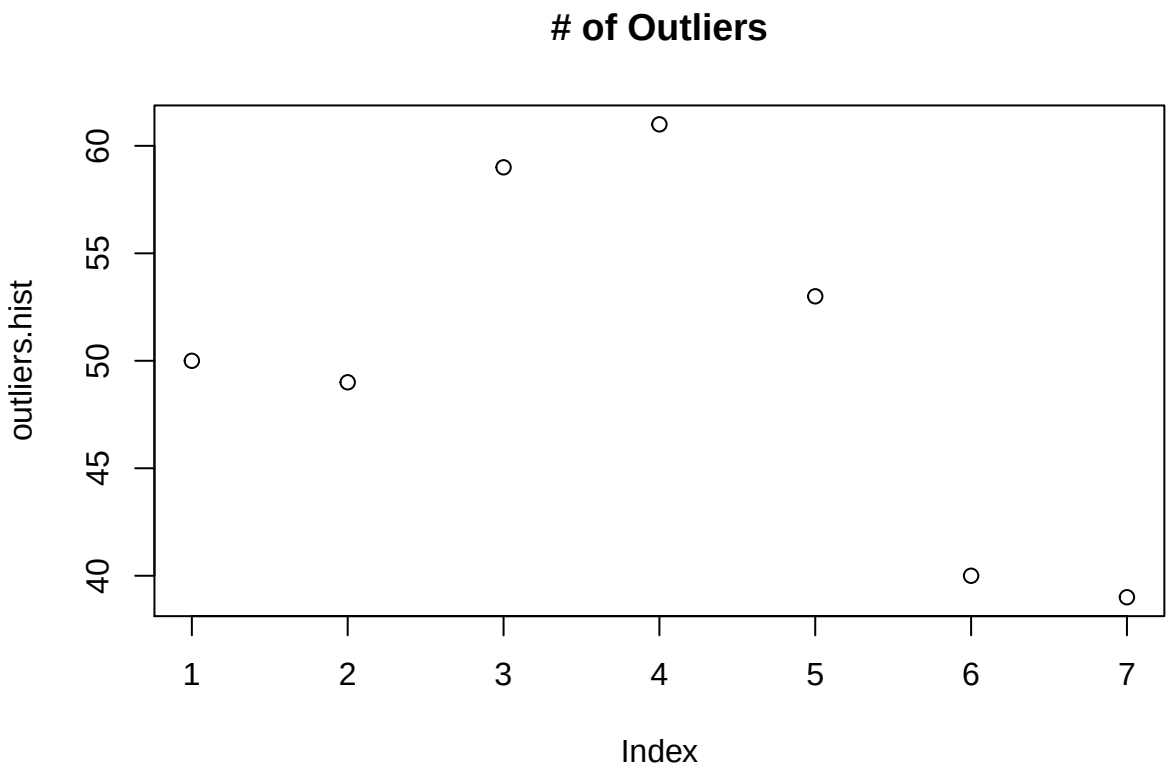
```
expr = paste("length~", paste(chosen.pos, collapse = "+"))
t =forward.change(newdata, expr,TRUE)
```

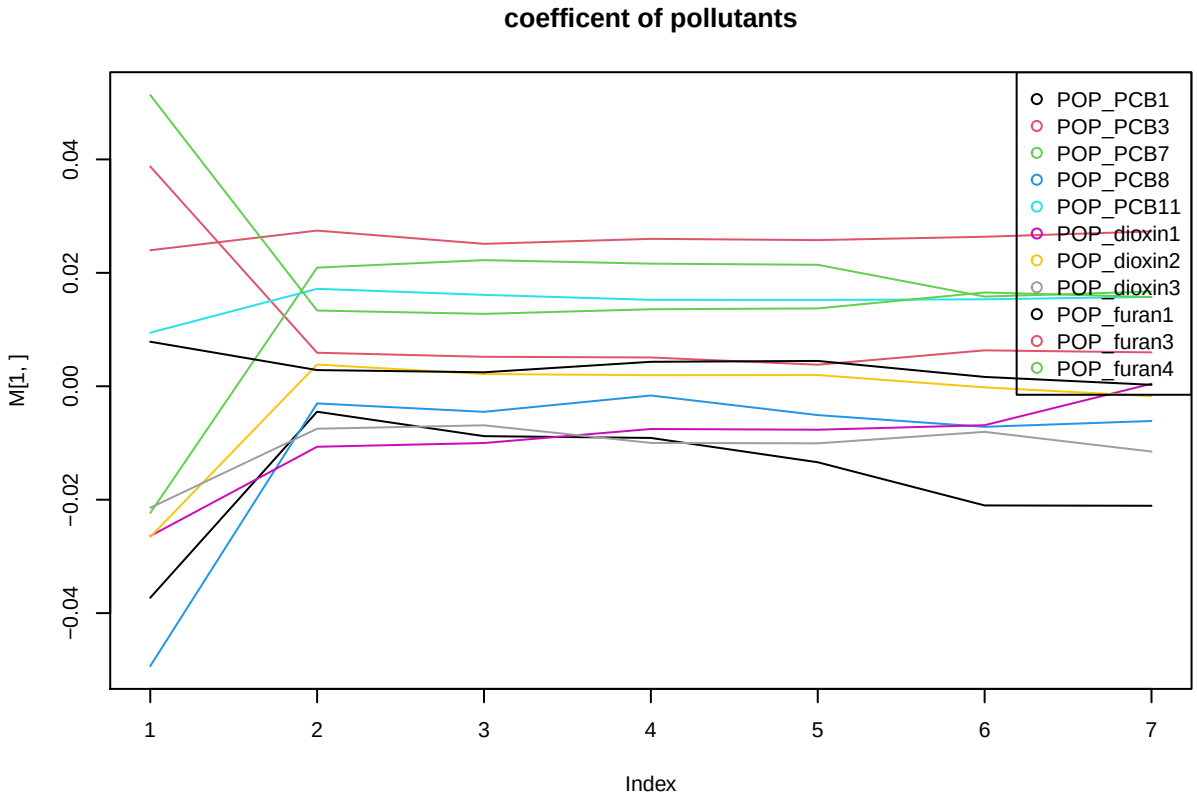
```
## [1] "added ageyrs"
## [1] "added POP_PCB10"
## [1] "added monocyte_pct"
## [1] "added POP_PCB2"
## [1] "added edu_cat"
## [1] "added smokenow"
```




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

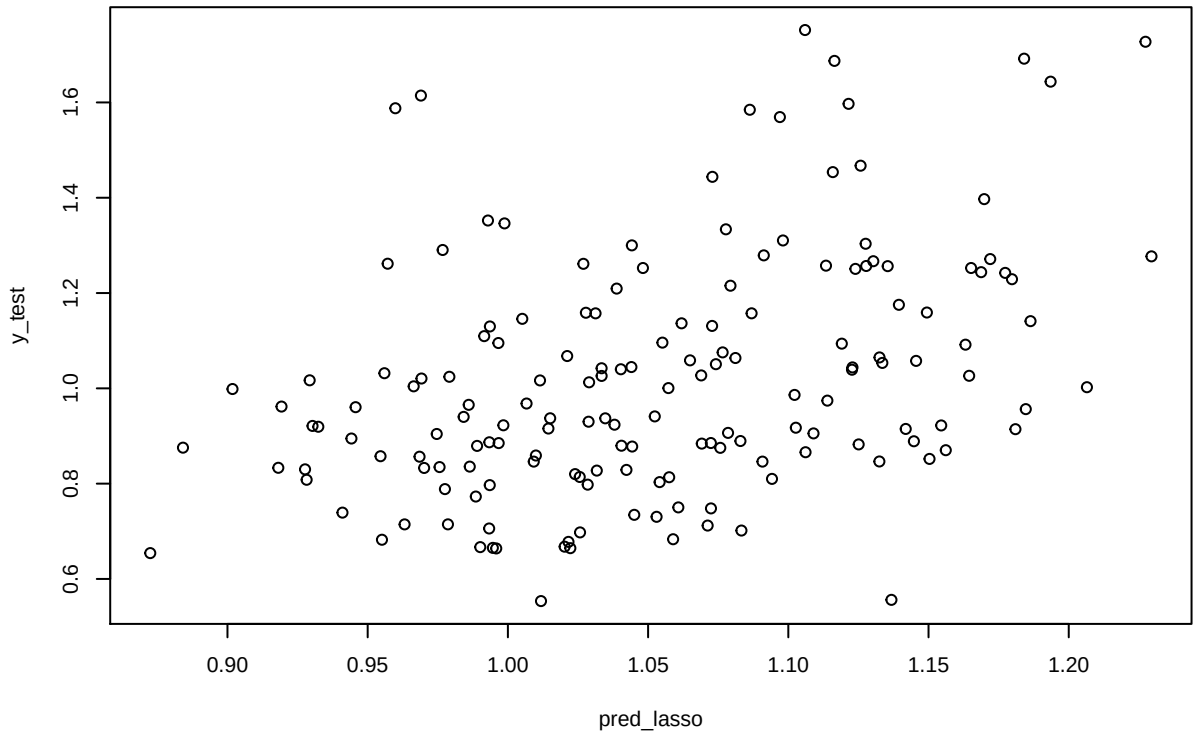

# of Influential Points − DFFITS
DFFITS.hist
40
45
50
1
2
3
4
5
6
7
Index


# of Outliers
outliers.hist
40
45
50
55
60
1
2
3
4
5
6
7
Index




```
newdata=data
newdata[,po.ind]=log(data[,po.ind])
chosen.po.ind=lasso.on.pollutants(newdata)
```

```
## [1] "mspe 0.0549633507362908"
```

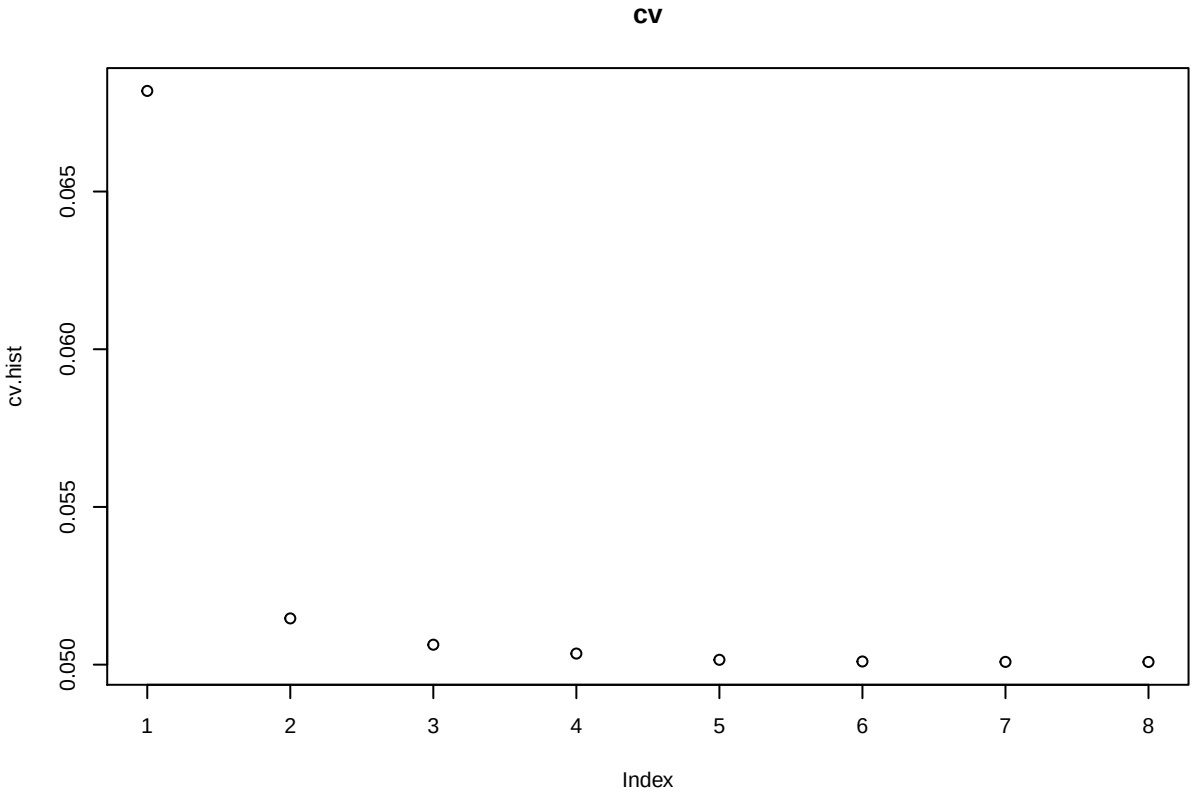


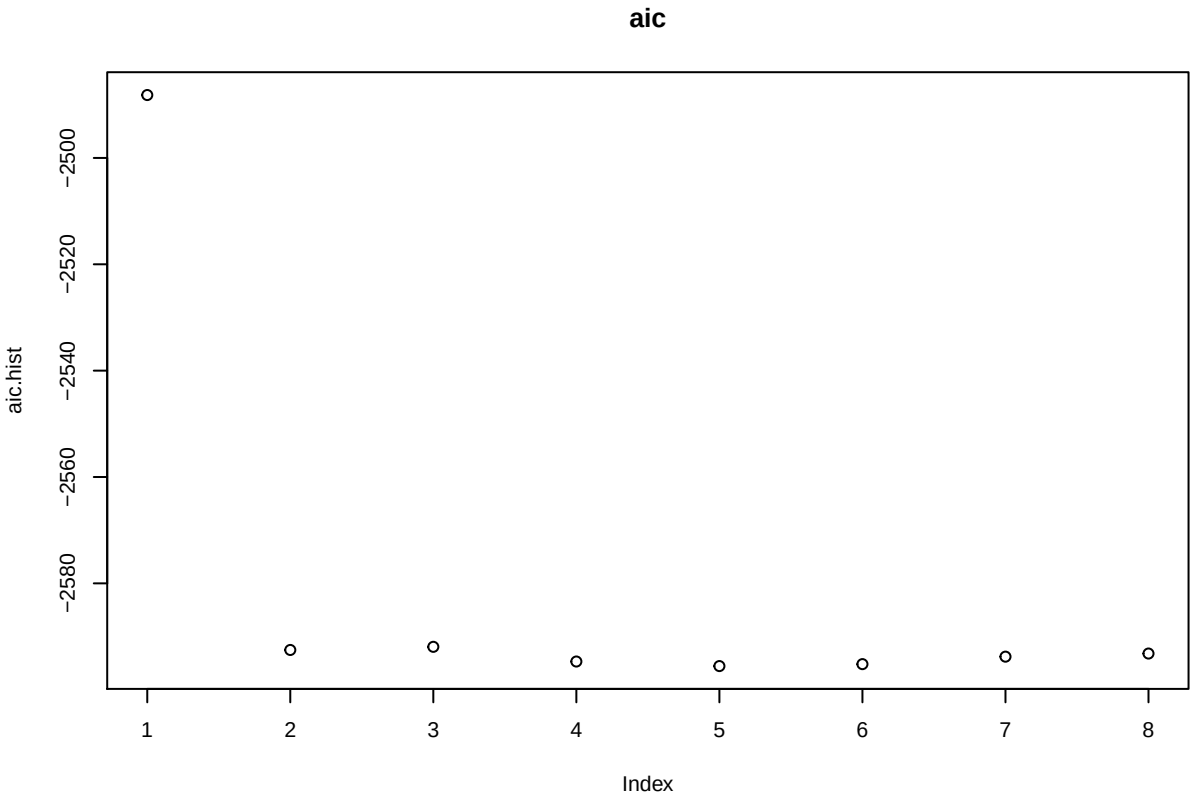

```
chosen.po.ind
```

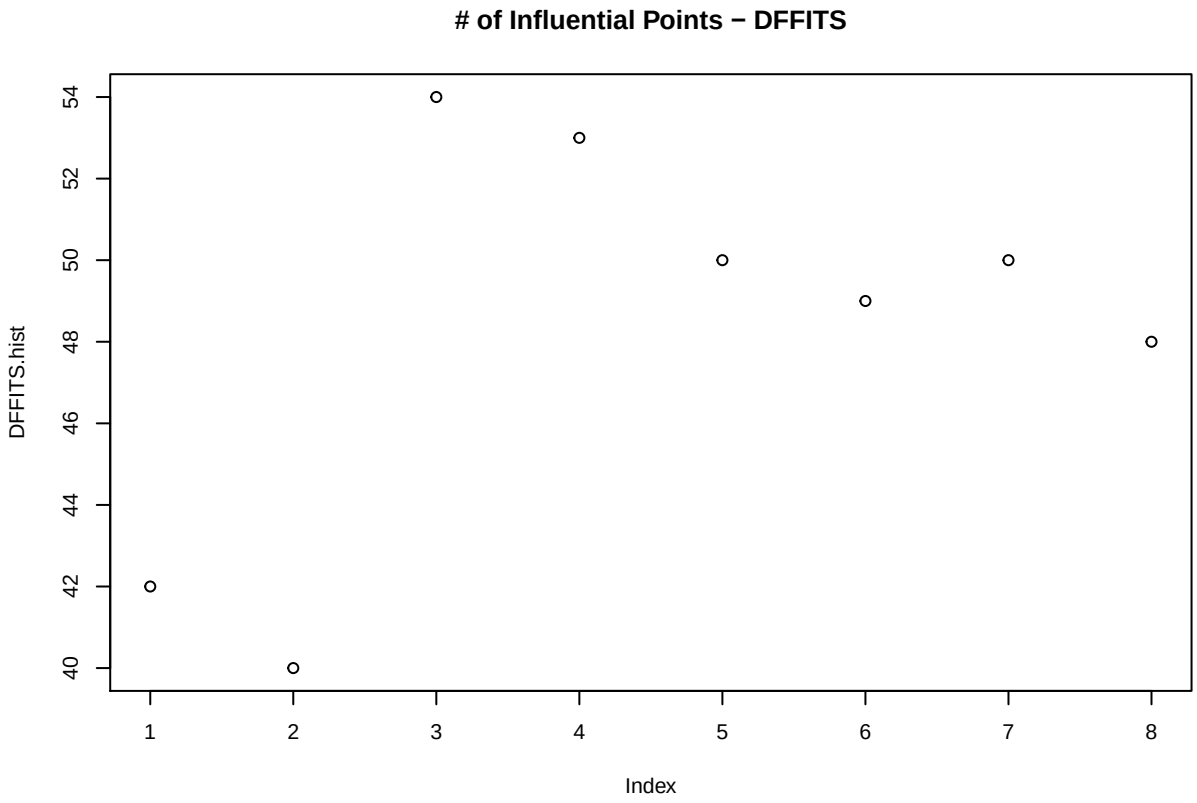
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  1.766034565
## POP_PCB1    -0.021767380
## POP_PCB2     .
## POP_PCB3     0.024091714
## POP_PCB4     .
## POP_PCB5     .
## POP_PCB6     .
## POP_PCB7    -0.010379955
## POP_PCB8    -0.055165978
## POP_PCB9     .
## POP_PCB10    .
## POP_PCB11    0.006035454
## POP_dioxin1 -0.022049982
## POP_dioxin2 -0.012170612
## POP_dioxin3 -0.023306362
## POP_furan1   .
## POP_furan2   .
## POP_furan3   0.006628585
## POP_furan4   0.053746150
```

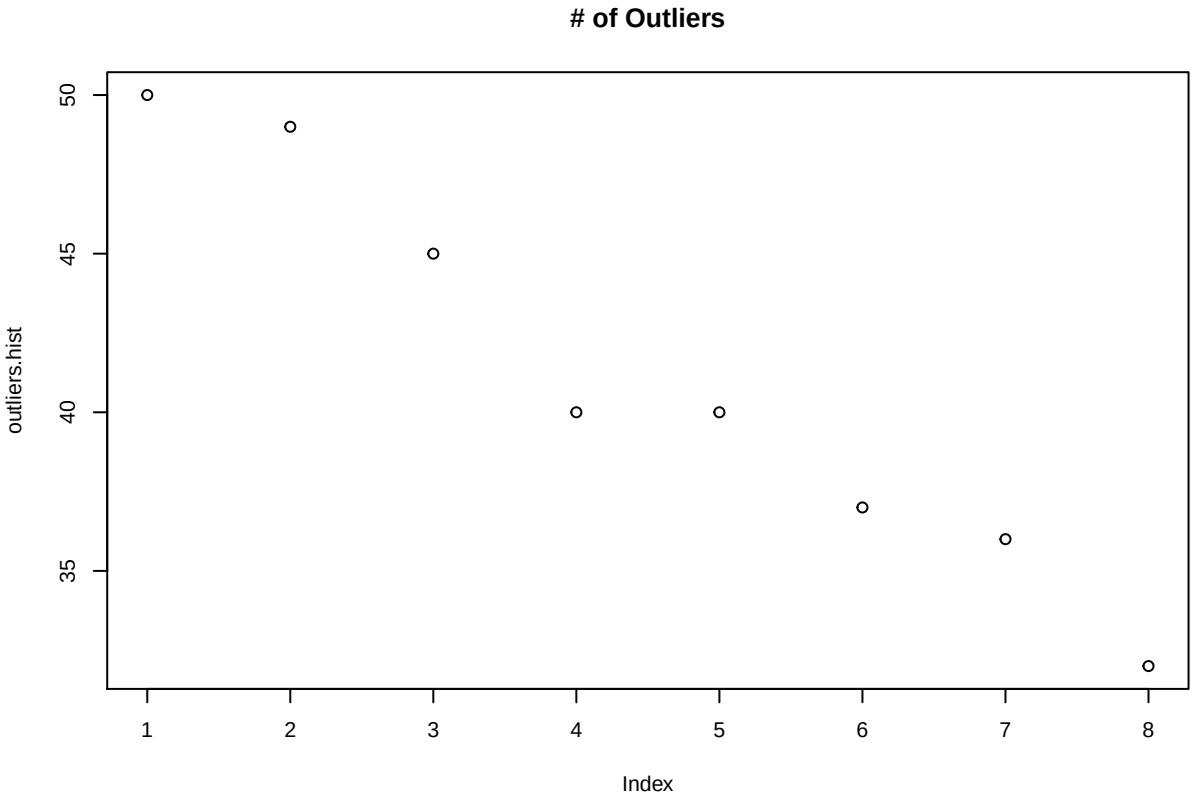
```
expr = paste("length~", paste(chosen.pos, collapse = "+"))
t =forward.change(newdata, expr,TRUE)
```

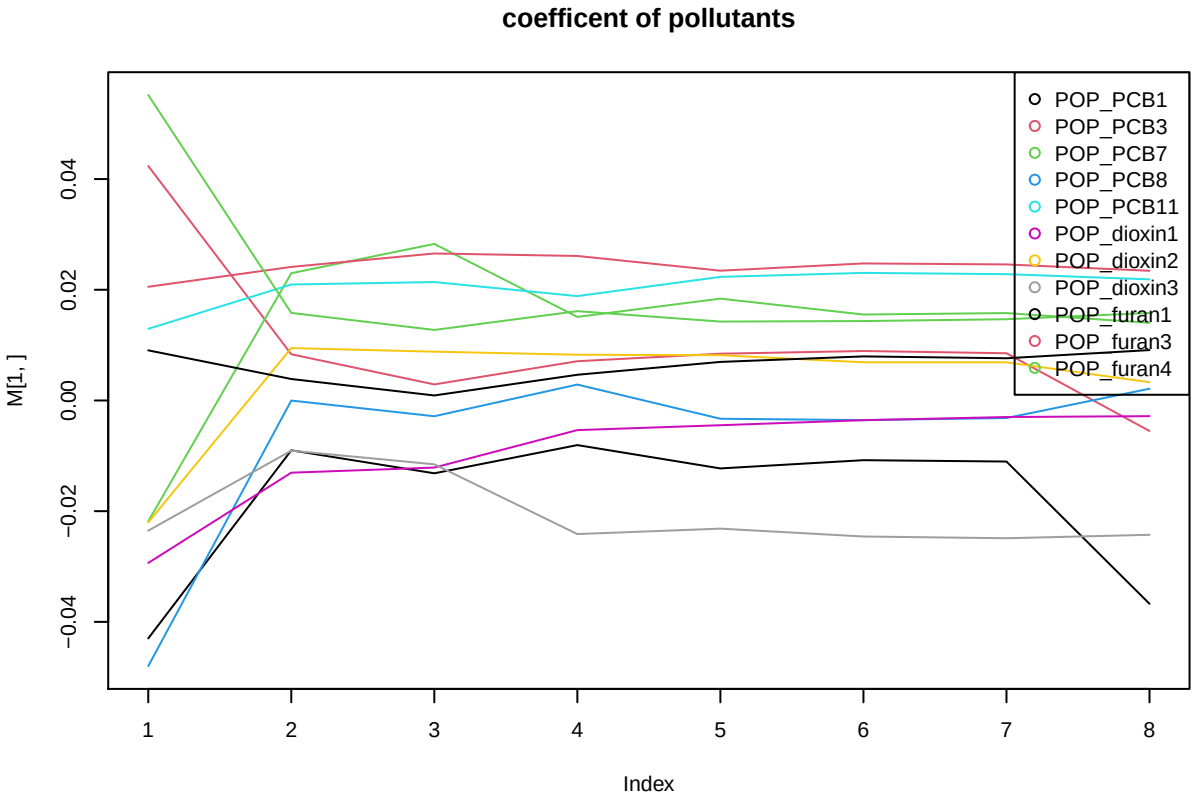
```
## [1] "added ageyrs"
## [1] "added race_cat"
## [1] "added male"
## [1] "added BMI"
## [1] "added eosinophils_pct"
## [1] "added neutrophils_pct"
## [1] "added POP_PCB5"
```




aic
aic.hist
−2500
−2520
−2540
−2560
−2580
1
2
3
4
5
6
7
8
Index


# of Influential Points − DFFITS
DFFITS.hist
40
42
44
46
48
50
52
54
1
2
3
4
5
6
7
8
Index


# of Outliers
outliers.hist
50
45
40
35
1
2
3
4
5
6
7
8
Index


coefficent of pollutants
POP_PCB1
POP_PCB3
POP_PCB7
POP_PCB8
POP_PCB11
POP_dioxin1
POP_dioxin2
POP_dioxin3
POP_furan1
POP_furan3
POP_furan4
M[1,]
0.04
0.02
0.00
−0.02
−0.04
1
2
3
4
5
6
7
8
Index