R Notebook

Code

Hide

```
library(e1071)
library(MASS)
df <- read.csv("smoking.csv")
str(df)
```

```
'data.frame':	55692 obs. of  27 variables:
 $ ID                 : int  0 1 2 3 4 5 6 7 9 10 ...
 $ gender             : chr  "F" "F" "M" "M" ...
 $ age                : int  40 40 55 40 40 30 40 45 50 45 ...
 $ height.cm.         : int  155 160 170 165 155 180 160 165 150 175 ...
 $ weight.kg.         : int  60 60 60 70 60 75 60 90 60 75 ...
 $ waist.cm.          : num  81.3 81 80 88 86 85 85.5 96 85 89 ...
 $ eyesight.left.     : num  1.2 0.8 0.8 1.5 1 1.2 1 1.2 0.7 1 ...
 $ eyesight.right.    : num  1 0.6 0.8 1.5 1 1.2 1 1 0.8 1 ...
 $ hearing.left.      : num  1 1 1 1 1 1 1 1 1 1 ...
 $ hearing.right.     : num  1 1 1 1 1 1 1 1 1 1 ...
 $ systolic           : num  114 119 138 100 120 128 116 153 115 113 ...
 $ relaxation         : num  73 70 86 60 74 76 82 96 74 64 ...
 $ fasting.blood.sugar: num  94 130 89 96 80 95 94 158 86 94 ...
 $ Cholesterol        : num  215 192 242 322 184 217 226 222 210 198 ...
 $ triglyceride       : num  82 115 182 254 74 199 68 269 66 147 ...
 $ HDL                : num  73 42 55 45 62 48 55 34 48 43 ...
 $ LDL                : num  126 127 151 226 107 129 157 134 149 126 ...
 $ hemoglobin         : num  12.9 12.7 15.8 14.7 12.5 16.2 17 15 13.7 16 ...
 $ Urine.protein      : num  1 1 1 1 1 1 1 1 1 1 ...
 $ serum.creatinine   : num  0.7 0.6 1 1 0.6 1.2 0.7 1.3 0.8 0.8 ...
 $ AST                : num  18 22 21 19 16 18 21 38 31 26 ...
 $ ALT                : num  19 19 16 26 14 27 27 71 31 24 ...
 $ Gtp                : num  27 18 22 18 22 33 39 111 14 63 ...
 $ oral               : chr  "Y" "Y" "Y" "Y" ...
 $ dental.caries      : int  0 0 0 0 0 0 1 0 0 0 ...
 $ tartar             : chr  "Y" "Y" "N" "Y" ...
 $ smoking            : int  0 0 1 0 0 0 1 0 0 0 ...
```

Hide

```
df <- subset(df, select=-c(ID,oral))
```

Format columns.

Hide

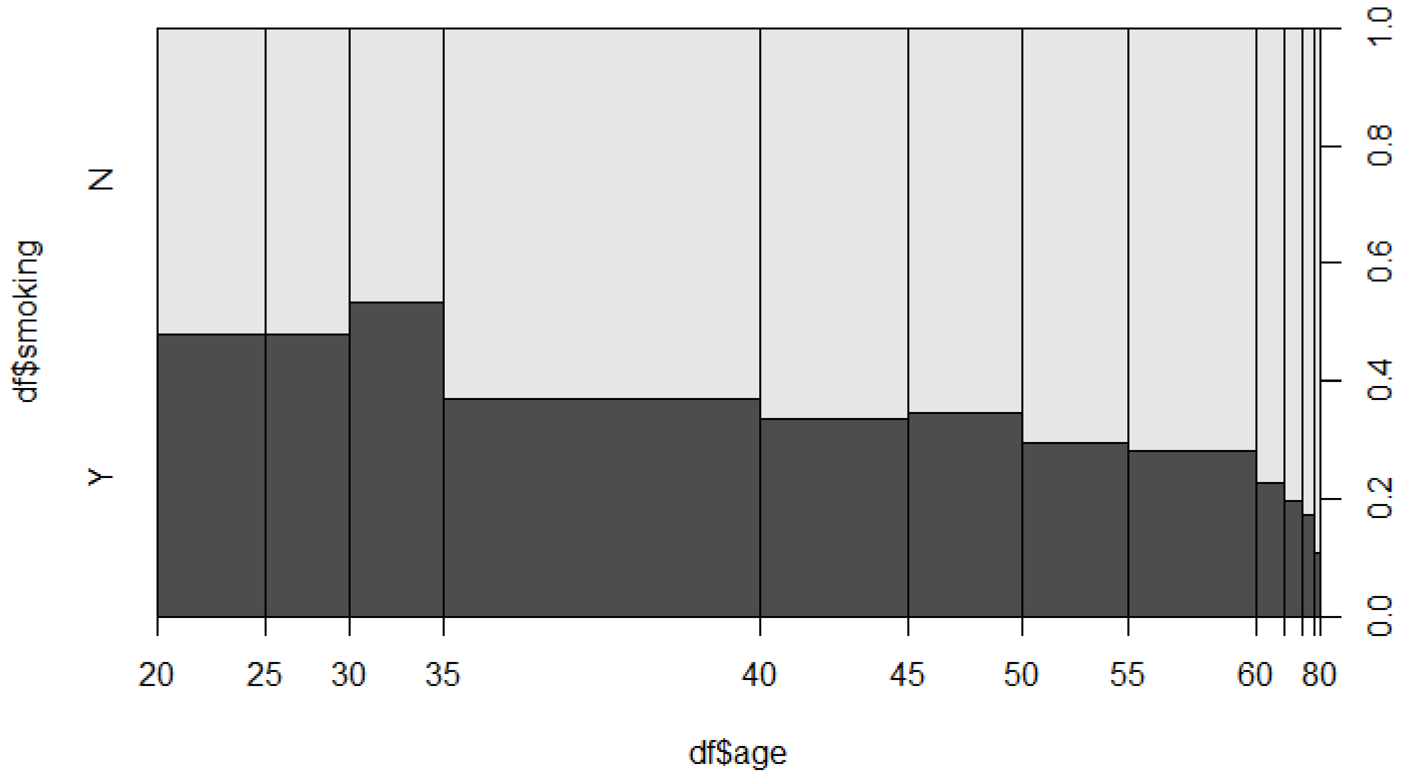
```
df$gender <- factor(df$gender)
# df$oral <- factor(df$oral) #single factor
df$dental.caries <- factor(df$dental.caries)
df$tartar <- factor(df$tartar)
df$smoking <- factor(df$smoking)
df$hearing.left. <- factor(df$hearing.left.)
df$hearing.right. <- factor(df$hearing.right.)

levels(df$dental.caries) <- c("N","Y")
levels(df$smoking) <- c("N","Y")
```

Plot smoking as a function of age.

Hide

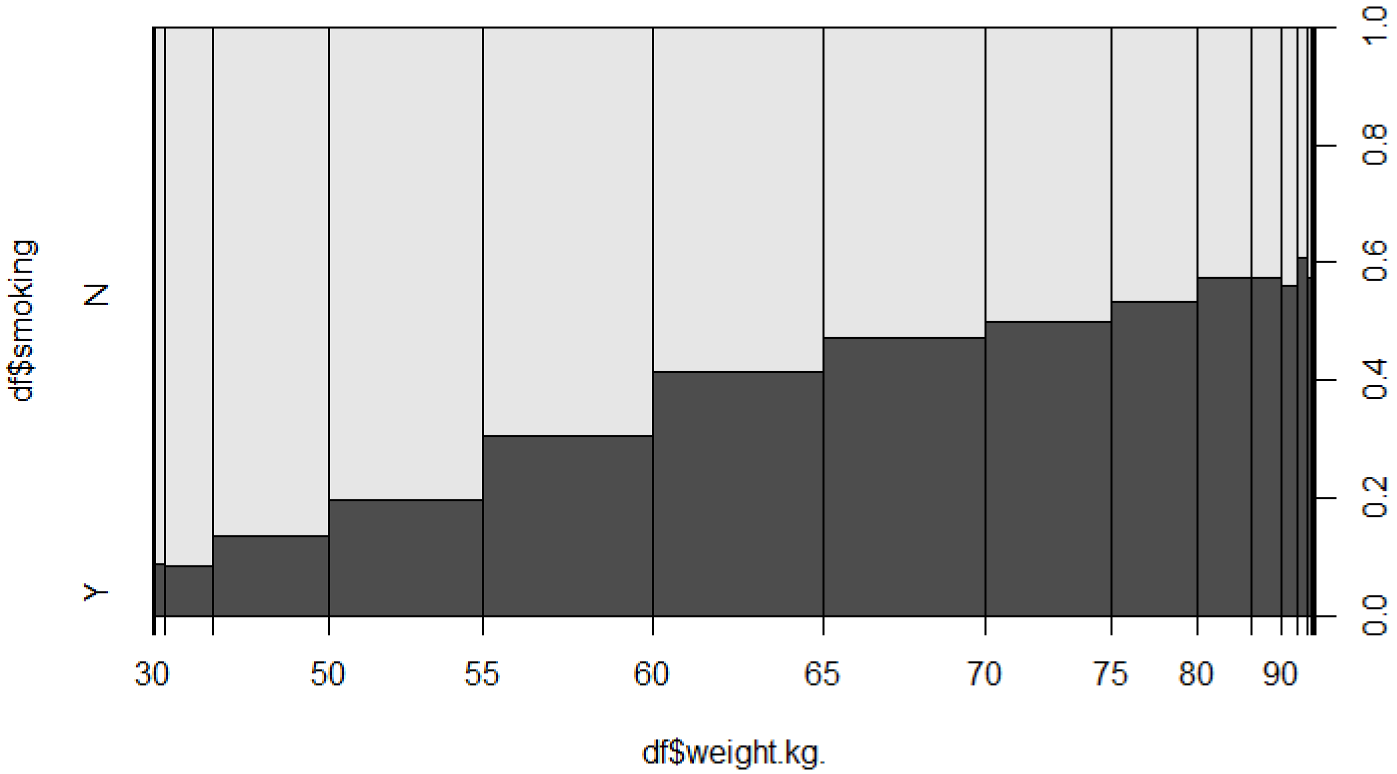
```
plot(df$smoking~df$age)
```



Plot emission as a function of population.

Hide

```
plot(df$smoking~df$weight.kg.)
```



Training and testing data.

Hide

```
set.seed(1234)
spec <- c(train=.6, test=.2, validate=.2)
df_ <- df
i <- sample(1:nrow(df),10000,replace=FALSE)
df <- df[i,]
i <- sample(cut(1:nrow(df),
                nrow(df)*cumsum(c(0,spec)), labels=names(spec)))
train <- df[i=="train",]
test <- df[i=="test",]
vald <- df[i=="validate",]
```

Run svm.

Hide

```
svm1 <- svm(smoking~., data=train, kernel="linear", cost=10, scale=TRUE)
summary(svm1)
```

```

Call:
svm(formula = smoking ~ ., data = train, kernel = "linear", cost = 10, scale = TRUE)


Parameters:
   SVM-Type:  C-classification 
 SVM-Kernel:  linear 
       cost:  10 

Number of Support Vectors:  3468

 ( 1729 1739 )


Number of Classes:  2 

Levels: 
 N Y
```

Try different costs.

Hide

```
tune_svm1 <- tune(svm, smoking~., data=vald, kernel="linear", ranges=list(cost=c(0.001,0.01,0.1,
1,5,10,100)))
```

```

WARNING: reaching max number of iterations

WARNING: reaching max number of iterations
```

Hide

```
summary(tune_svm1)
```

```

Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation 

- best parameters:
```

cost <dbl>
1

1 row

```
- best performance: 0.257

- Detailed performance results:
```

cost	error	dispersion
<dbl>	<dbl>	<dbl>
1e-03	0.2720	0.02463060
1e-02	0.2600	0.01509231
1e-01	0.2580	0.02347576
1e+00	0.2570	0.02584140
5e+00	0.2605	0.02178812
1e+01	0.2600	0.02223611
1e+02	0.2600	0.02211083

7 rows

```
NA
```

Try with polynomial.

Hide

```
tune_svm2 <- tune(svm, smoking~., data=vald, kernel="polynomial", ranges=list(cost=c(0.001,0.01,
0.1,1,5,10,100)))
summary(tune_svm2)
```

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
```

cost
<dbl>
10

1 row

```
- best performance: 0.2735

- Detailed performance results:
```

cost	error	dispersion
<dbl>	<dbl>	<dbl>
1e-03	0.3655	0.01300641
1e-02	0.3600	0.01414214
1e-01	0.3420	0.03172801
1e+00	0.2985	0.02858224
5e+00	0.2760	0.03777124
1e+01	0.2735	0.04048662
1e+02	0.3160	0.02144761

7 rows

```
NA
```

Try with radial.

Hide

```
tune_svm3 <- tune(svm, smoking~., data=vald, kernel="radial", ranges=list(cost=c(0.001,0.01,0.1,
1,5,10,100)))
summary(tune_svm3)
```

```


Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
```

cost
<dbl>
1

1 row

```


- best performance: 0.252

- Detailed performance results:
```

cost	error	dispersion
<dbl>	<dbl>	<dbl>
1e-03	0.3655	0.01553669

cost	error	dispersion
<dbl>	<dbl>	<dbl>
1e-02	0.3655	0.01553669
1e-01	0.2760	0.02736583
1e+00	0.2520	0.02097618
5e+00	0.2700	0.02905933
1e+01	0.2770	0.02760837
1e+02	0.3065	0.03574990

7 rows

```
NA
```

These algorithms are fairly slow. Each model takes about 5 minutes to calculate. This makes it difficult to make minor adjustments to the parameters for testing. It seems that radial with cost of 1 is the best. Due to the size of the dataset, I had to sample it in order to run the algorithms in a timely manner.