

# Machine Learning I Group Work

*Adriana Ricklin, Yvonne Schaerli, Christina Sudermann, Carole Mattmann*

*11 June 2020*

## Contents

<b>1 Packages</b>	<b>1</b>
<b>2 Import and data cleaning</b>	<b>1</b>
<b>3 Linear models -&gt; Christina</b>	<b>2</b>
<b>4 Linear models (GAM &amp; Polynomial) -&gt; Yvonne</b>	<b>2</b>
<b>5 GLM and cross validation -&gt; Carole</b>	<b>2</b>
5.1 Generalised Linear Models for count data . . . . .	2
5.2 Generalised Linear Models for binomial data . . . . .	4
5.3 Cross Validation . . . . .	6
<b>6 Decision Trees -&gt; Adriana</b>	<b>8</b>
<b>7 Support Vector Machines -&gt; Carole</b>	<b>8</b>
7.1 two classes linear . . . . .	8
7.2 two classes polynomial . . . . .	12

## 1 Packages

```
library(gridExtra)
library(tidyverse) # included ggplot2
library(e1071)     # for SVM
```

## 2 Import and data cleaning

```
insurance <- read.csv("../01_data/insurance.csv", header=TRUE)
str(insurance)

## 'data.frame': 1338 obs. of 7 variables:
## $ age : int 19 18 28 33 32 31 46 37 37 60 ...
## $ sex : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
## $ bmi : num 27.9 33.8 33 22.7 28.9 ...
## $ children: int 0 1 3 0 0 0 1 3 2 0 ...
## $ smoker : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
## $ region : Factor w/ 4 levels "northeast","northwest",...: 4 3 3 2 2 3 3 2 1 2 ...
## $ charges : num 16885 1726 4449 21984 3867 ...

# smoker = 1 / nonsmoker = 0
insurance$smoker <- as.character(insurance$smoker)
insurance$smoker[insurance$smoker == "yes"] <- "1"
insurance$smoker[insurance$smoker == "no"] <- "0"
insurance$smoker <- as.factor(insurance$smoker)
```

```
head(insurance)
```

```
##   age    sex    bmi children smoker    region    charges
## 1  19 female 27.900         0      1 southwest 16884.924
## 2  18   male 33.770         1      0 southeast  1725.552
## 3  28   male 33.000         3      0 southeast  4449.462
## 4  33   male 22.705         0      0 northwest 21984.471
## 5  32   male 28.880         0      0 northwest  3866.855
## 6  31 female 25.740         0      0 southeast  3756.622
```

### 3 Linear models -> Christina

### 4 Linear models (GAM & Polynomial) -> Yvonne

### 5 GLM and cross validation -> Carole

#### 5.1 Generalised Linear Models for count data

##### 5.1.1 Original data

The number of children an insured person has is analysed. We have the following data on children per person. The number of children ranges from 0 to 5 with a median of 1.

```
summary(insurance$children)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.000   0.000   1.000   1.095   2.000   5.000
```

##### 5.1.2 Poisson model

To model count data (number of children) the poisson model is used. An analysis performed beforehand showed that only the variables “charges” and “smoker” have a significant impact on the number of children.

```
glm.children <- glm(children ~ smoker+charges,
                    data=insurance,
                    family = "poisson")
```

```
summary(glm.children)
```

```
##
## Call:
## glm(formula = children ~ smoker + charges, family = "poisson",
##      data = insurance)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8561  -1.4318  -0.1057   0.7768   2.9717
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.706e-02  4.213e-02  -0.880   0.3790
## smoker1     -3.239e-01  1.058e-01  -3.061   0.0022 **
## charges      1.419e-05  3.365e-06   4.217 2.48e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 2001.6  on 1337  degrees of freedom
## Residual deviance: 1984.1  on 1335  degrees of freedom
## AIC: 3879.4
##
## Number of Fisher Scoring iterations: 5
```

To get the coefficients, the log transformation needs to be reversed:

```
exp(coef(glm.children))
```

```
## (Intercept)      smoker1      charges
##    0.9636169    0.7233085    1.0000142
```

Smoker (factor): The model shows that for the factor smoker (yes/no), a smoker has on average 72% of the number of children a non-smoker has. The more common-sense interpretation might be the other way around, that people who have 1 or more children smoke less, but for the moment we have no proof of that.

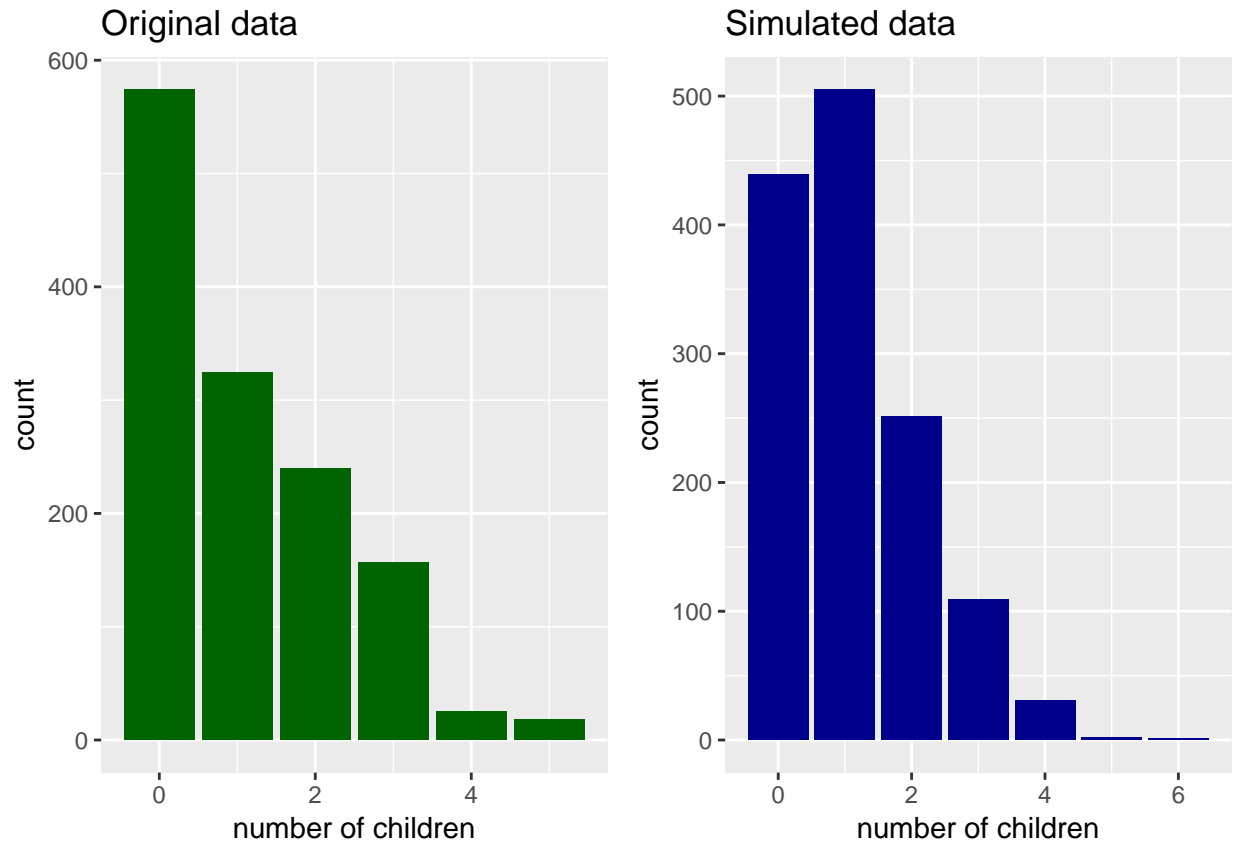
Charges: A person with higher charges will on average have more children. If charges are increased by 1000 dollars, the calculated number of children increases by 1.4%.

### 5.1.3 Simulation of data and comparison

With the calculated model, data is simulated:

```
##      sim_1
## Min.    :0.000
## 1st Qu.:0.000
## Median :1.000
## Mean   :1.102
## 3rd Qu.:2.000
## Max.   :6.000
```

The original and the simulated data are compared visually. The number of children from the simulated data (0-6) seem to be plausible. The distribution has a strong downwards trend starting at 1 like the original data. However the model does not seem to generate enough data with 0 children.



## 5.2 Generalised Linear Models for binomial data

A model is fitted that predicts if a person is a smoker or not. Only the significant values age, bmi and charges are used.

```
glm.smoker.2 <- glm(smoker ~ age+bmi+charges,
  data=insurance,
  family = "binomial")

summary(glm.smoker.2)
```

```
##
## Call:
## glm(formula = smoker ~ age + bmi + charges, family = "binomial",
##      data = insurance)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.09442  -0.10998  -0.04475  -0.00970   1.53727
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.311e+00  1.029e+00   5.163 2.43e-07 ***
## age         -9.875e-02  1.300e-02  -7.597 3.02e-14 ***
## bmi         -3.481e-01  4.309e-02  -8.078 6.60e-16 ***
## charges      3.822e-04  2.917e-05  13.104 < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 1356.63  on 1337  degrees of freedom
## Residual deviance:  311.98  on 1334  degrees of freedom
## AIC: 319.98
##
## Number of Fisher Scoring iterations: 8
```

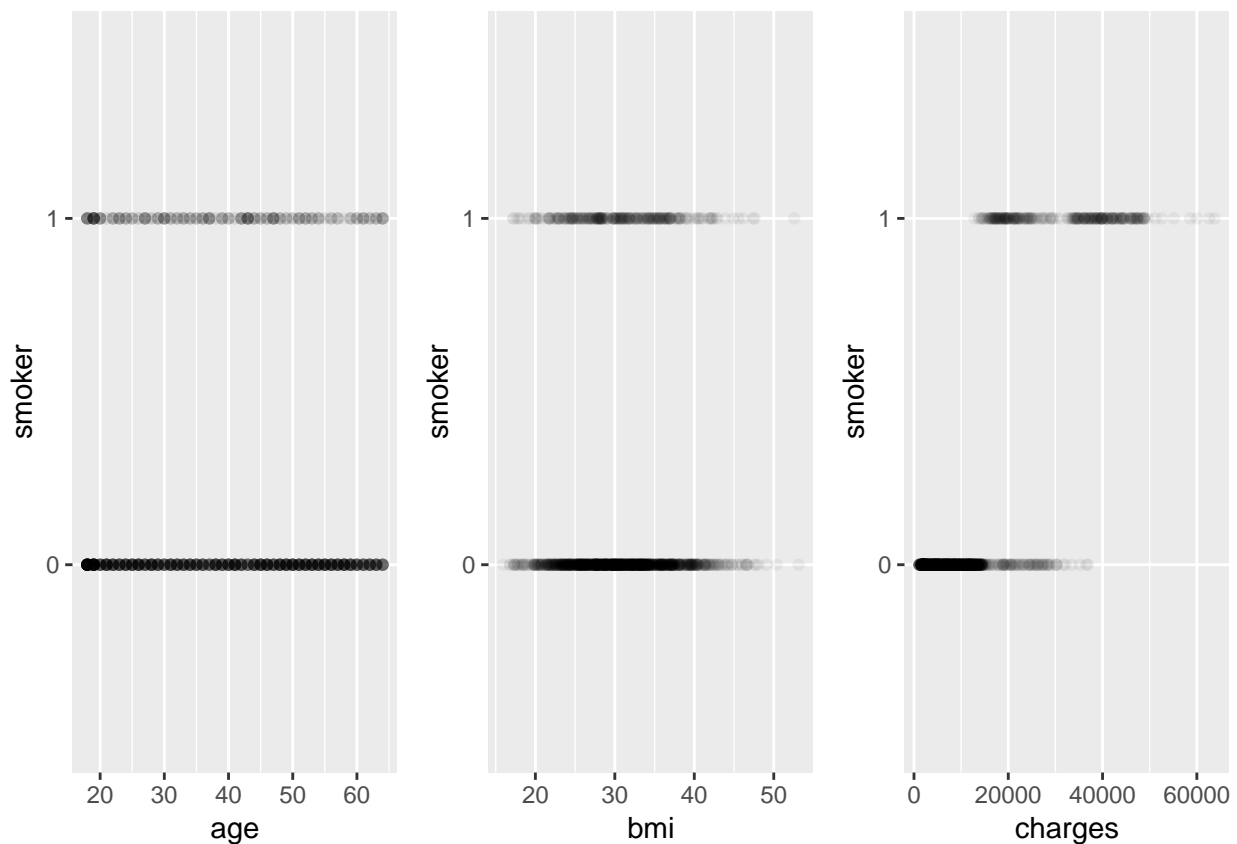
```
exp(coef(glm.smoker.2))
```

```
## (Intercept)      age      bmi    charges
## 202.5686249   0.9059677  0.7060520  1.0003823
```

Age and BMI has a negative effect on smoker. This means the higher a persons BMI and age, the lower the probability that the person is a smoker. Charges has a positive effect. This means the higher a persons charges, the higher is the possibility that the person smokes.

### 5.2.1 Graphical analysis

This can also be explored graphically, at least for charges it is clearly visible that smokers have higher charges.



### 5.2.2 Estimating the model performance

The predicted values are transformed into binary (beforehand they indicated the probability) and compared with the actual data.

```
fitted.smoker.disc <- ifelse(fitted(glm.smoker.2) < 0.5,
                             yes = 0, no = 1)
head(fitted.smoker.disc)

## 1 2 3 4 5 6
## 1 0 0 1 0 0

d.obs.fit.smoker <- data.frame(obs = insurance$smoker,
                               fitted = fitted.smoker.disc)
head(d.obs.fit.smoker)

##   obs fitted
## 1    1      1
## 2    0      0
## 3    0      0
## 4    0      1
## 5    0      0
## 6    0      0
```

We observe the following fit:

```
##      fit
## obs    0    1
##   0 1028   36
##   1   23  251
```

### 5.3 Cross Validation

Three linear models are cross validated:

```
lm.1 <- lm(data=insurance, charges~children+smoker+bmi+age+region+sex)

lm.2 <- lm(data=insurance, charges~children+smoker)

lm.3 <- lm(data=insurance, charges~(poly(bmi, degree=3))
           +(poly(age, degree=2))+children+smoker)
```

The in sample performance, using R Squared as measure is the following:

```
summary(lm.1)$r.squared
```

```
## [1] 0.750913
```

```
summary(lm.2)$r.squared
```

```
## [1] 0.6236038
```

```
summary(lm.3)$r.squared
```

```
## [1] 0.754424
```

The out of sample performance is computed by using 50:50 training and test data and repeating the process 100 times.

```
set.seed(5)

r.squared.lm.1 <- c()
r.squared.lm.2 <- c()
r.squared.lm.3 <- c()
```

```

for(i in 1:100){

  # prepare data

  train.YES <- sample(x=c(TRUE,FALSE),
                     size=nrow(insurance),
                     replace = TRUE)

  table(train.YES)

  insurance.train <- insurance[train.YES, ]
  insurance.test <- insurance[!train.YES, ]

  # fit model with train data

  lm.1.train <- lm(formula = formula(lm.1),
                  data = insurance.train)

  lm.2.train <- lm(formula = formula(lm.2),
                  data = insurance.train)

  lm.3.train <- lm(formula = formula(lm.3),
                  data = insurance.train)

  # make prediction on test data
  lm.1.predict <- predict(lm.1.train,
                        newdata = insurance.test)

  lm.2.predict <- predict(lm.2.train,
                        newdata = insurance.test)

  lm.3.predict <- predict(lm.3.train,
                        newdata = insurance.test)

  # compute r.squared and save in list

  r.squared.lm.1[i] <- cor(lm.1.predict,
                        insurance.test$charges)^2

  r.squared.lm.2[i] <- cor(lm.2.predict,
                        insurance.test$charges)^2

  r.squared.lm.3[i] <- cor(lm.3.predict,
                        insurance.test$charges)^2
}

```

The out of sample performance, using R Squared as measure is the following:

```

#lm.1
mean(r.squared.lm.1)

```

```
## [1] 0.7474233
```

```

#lm.2
mean(r.squared.lm.2)

```

```
## [1] 0.6235398
```

```
#lm.3
```

```
mean(r.squared.lm.3)
```

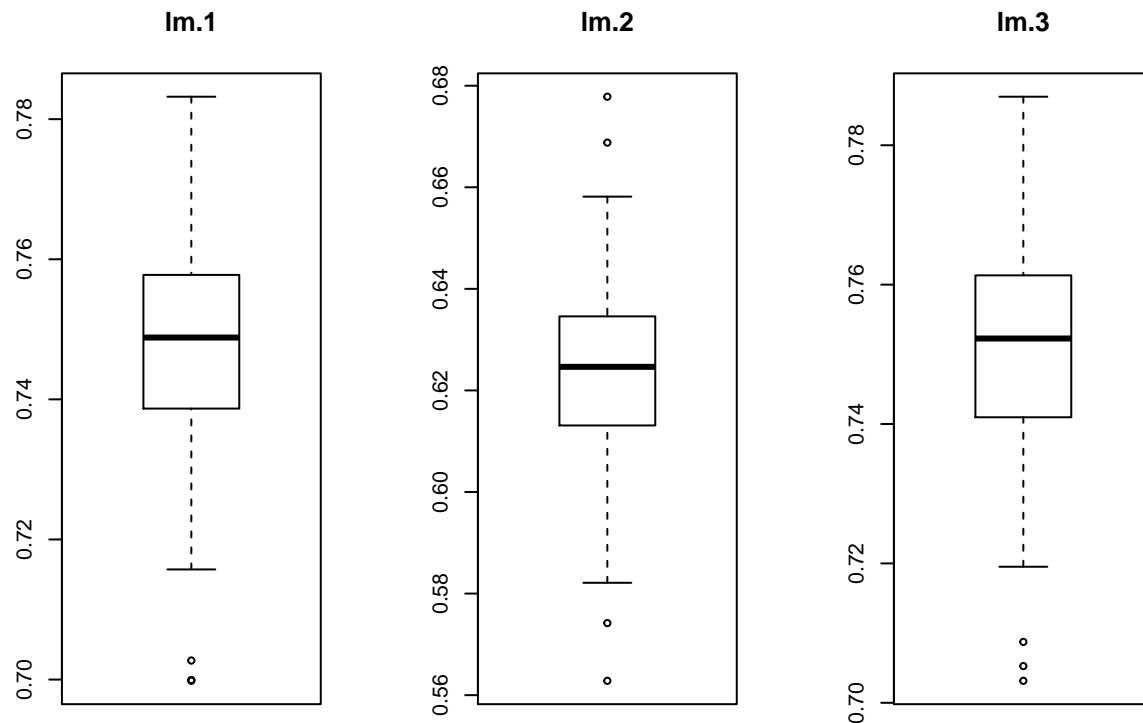
```
## [1] 0.7505379
```

```
par(mfrow=c(1,3))
```

```
boxplot(r.squared.lm.1, main="lm.1")
```

```
boxplot(r.squared.lm.2, main="lm.2")
```

```
boxplot(r.squared.lm.3, main="lm.3")
```



It can be observed that lm.3 performs slightly better but it is also the most complicated. lm.1 might be the better model, the performance is just slightly lower and it is simpler.

## 6 Decision Trees -> Adriana

## 7 Support Vector Machines -> Carole

In this chapter, Support Vector Machines are applied to the insurance dataset.

### 7.1 two classes linear

By visual exploration, we discover that smokers and non-smokers form two groups. We fit an SVM for those two groups. The lines in the graph are the first guess of how the data could be divided.

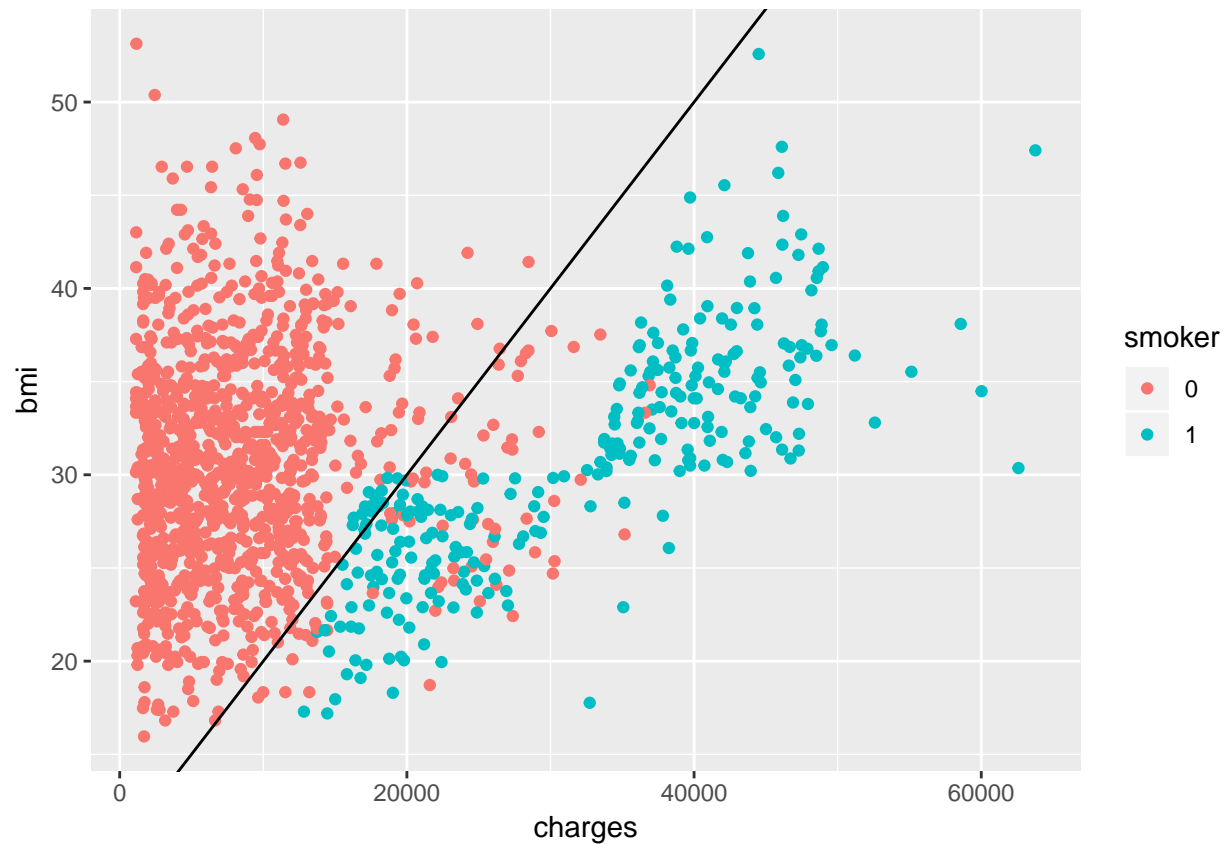
```
# guess of hyperplane
```

```
slope <- 0.001
```



```
intercept <- 10
```

```
ggplot(data = insurance) +  
  geom_point(aes(x = charges, y = bmi, color=smoker))+  
  geom_abline(slope = slope, intercept = intercept)
```



A linear SVM is tuned using different cost ranges.

```
set.seed(5)  
#get optimal cost  
cost_range <-  
  c(1e-10,  
    1e-7,  
    1e-5,  
    0.001,  
    0.0025,  
    0.005,  
    0.0075,  
    0.01,  
    0.1,  
    1,  
    5,  
    10,  
    100,  
    200)
```

```
tune.out.1 <- tune(
  svm,
  smoker ~ charges+bmi,
  data = insurance.train,
  kernel = "linear",
  ranges = list(cost = cost_range)
)
```

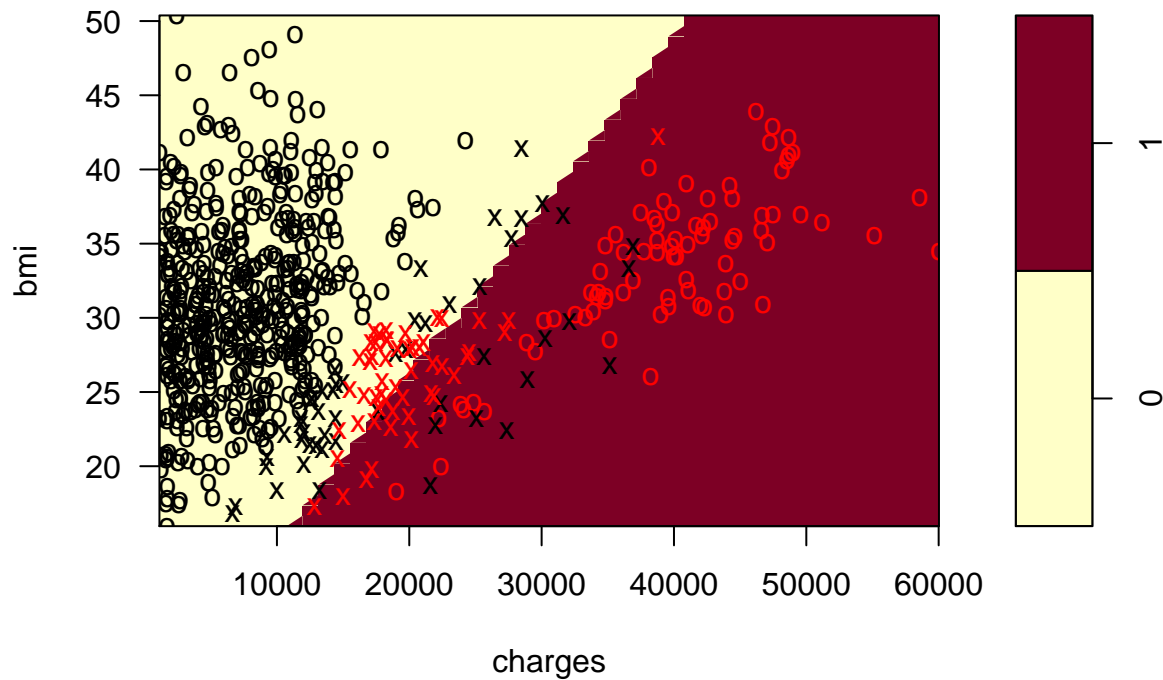
The following is the best model. A cost of 5 is used. There are 104 support vectors.

```
#best model
bestmod <- tune.out.1$best.model
summary(bestmod)

##
## Call:
## best.tune(method = svm, train.x = smoker ~ charges + bmi, data = insurance.train,
##   ranges = list(cost = cost_range), kernel = "linear")
##
##
## Parameters:
##   SVM-Type:  C-classification
##   SVM-Kernel: linear
##         cost: 5
##
## Number of Support Vectors: 104
##
## ( 52 52 )
##
##
## Number of Classes: 2
##
## Levels:
## 0 1

plot(bestmod, insurance.train, bmi~charges)
```

## SVM classification plot



The confusion matrices and the classification error rate for train and test data are:

```
# confusion matrix train
confusion.lin.train <- table(predict = predict(bestmod, insurance.train),
                             truth = insurance.train$smoker)
confusion.lin.train

##           truth
## predict    0    1
##           0 514  26
##           1  18 106

#classification error rate train
(confusion.lin.train[1,2]+confusion.lin.train[2,1])/sum(confusion.lin.train[1:2,1:2])

## [1] 0.06626506

# confusion matrix linear test
confusion.lin.test <- table(predict = predict(bestmod, insurance.test),
                             truth = insurance.test$smoker)
confusion.lin.test

##           truth
## predict    0    1
##           0 509  18
##           1  23 124

#classification error rate test
(confusion.lin.test[1,2]+confusion.lin.test[2,1])/sum(confusion.lin.test[1:2,1:2])
```

```
## [1] 0.06083086
```

## 7.2 two classes polynomial

A polynomial model was also tested but the classification error rate was slightly higher:

```
set.seed(5)
#get optimal cost and degree for polynomial

cost_range <-
  c(1e-10,
    1e-7,
    1e-5,
    0.001,
    0.0025,
    0.005,
    0.0075,
    0.01,
    0.1,
    1,
    5,
    10,
    100)

degree_range <- 2:4

tune.out.poly <- tune(
  svm,
  smoker ~ charges+bmi,
  data = insurance.train,
  kernel = "polynomial",
  ranges = list(cost = cost_range, degree = degree_range))

#best model poly
tune.out.poly$best.parameters

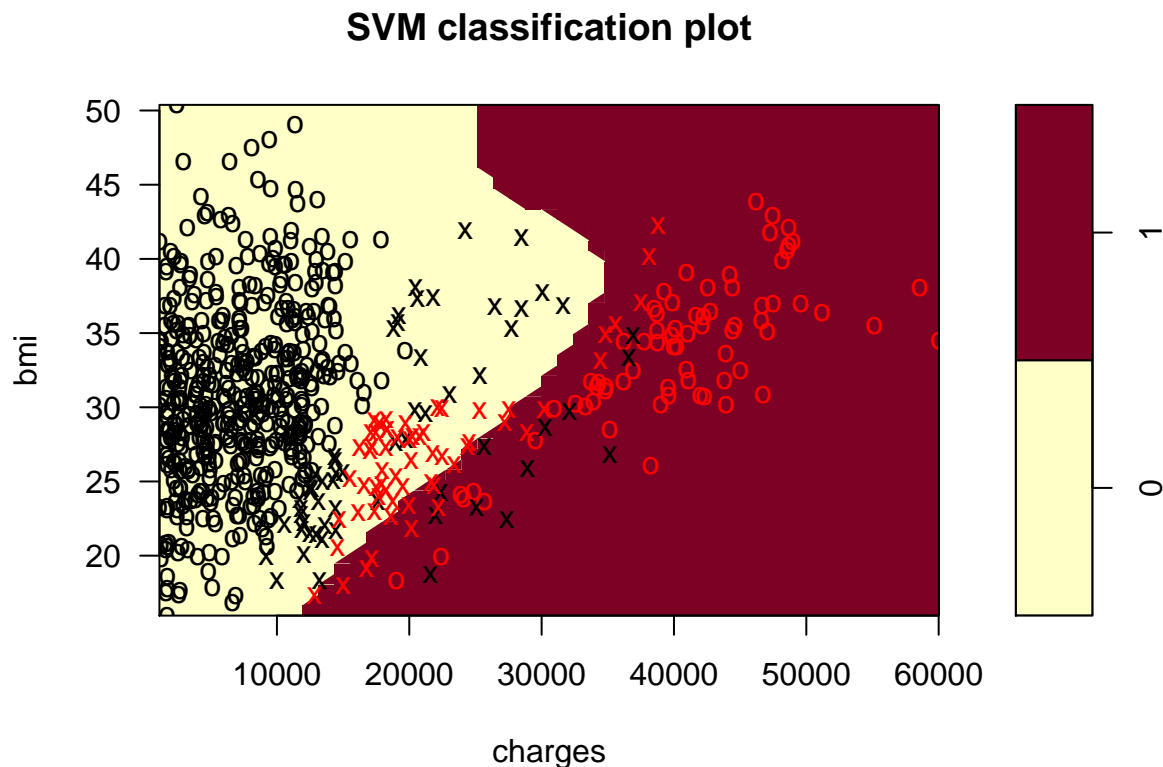
##      cost degree
## 26   100      3

bestmod.poly <- tune.out.poly$best.model
summary(bestmod.poly)

##
## Call:
## best.tune(method = svm, train.x = smoker ~ charges + bmi, data = insurance.train,
##      ranges = list(cost = cost_range, degree = degree_range),
##      kernel = "polynomial")
##
##
## Parameters:
##   SVM-Type:  C-classification
##   SVM-Kernel: polynomial
##      cost:  100
##     degree:  3
##    coef.0:  0
##
```

```
## Number of Support Vectors: 120
##
## ( 60 60 )
##
##
## Number of Classes: 2
##
## Levels:
## 0 1
```

```
plot(bestmod.poly, insurance.train, bmi~charges)
```



```
# confusion matrix
confusion.poly.train <- table(predict = predict(bestmod.poly, insurance.train),
                              truth = insurance.train$smoker)

confusion.poly.test <- table(predict = predict(bestmod.poly, insurance.test),
                              truth = insurance.test$smoker)

#classification error rate

(confusion.poly.train[1,2]+confusion.poly.train[2,1])/sum(confusion.poly.train[1:2,1:2])

## [1] 0.0753012

(confusion.poly.test[1,2]+confusion.poly.test[2,1])/sum(confusion.poly.test[1:2,1:2])

## [1] 0.06379822
```