Homework 11

Prakash Paudyal

You do not need to include the above statements.

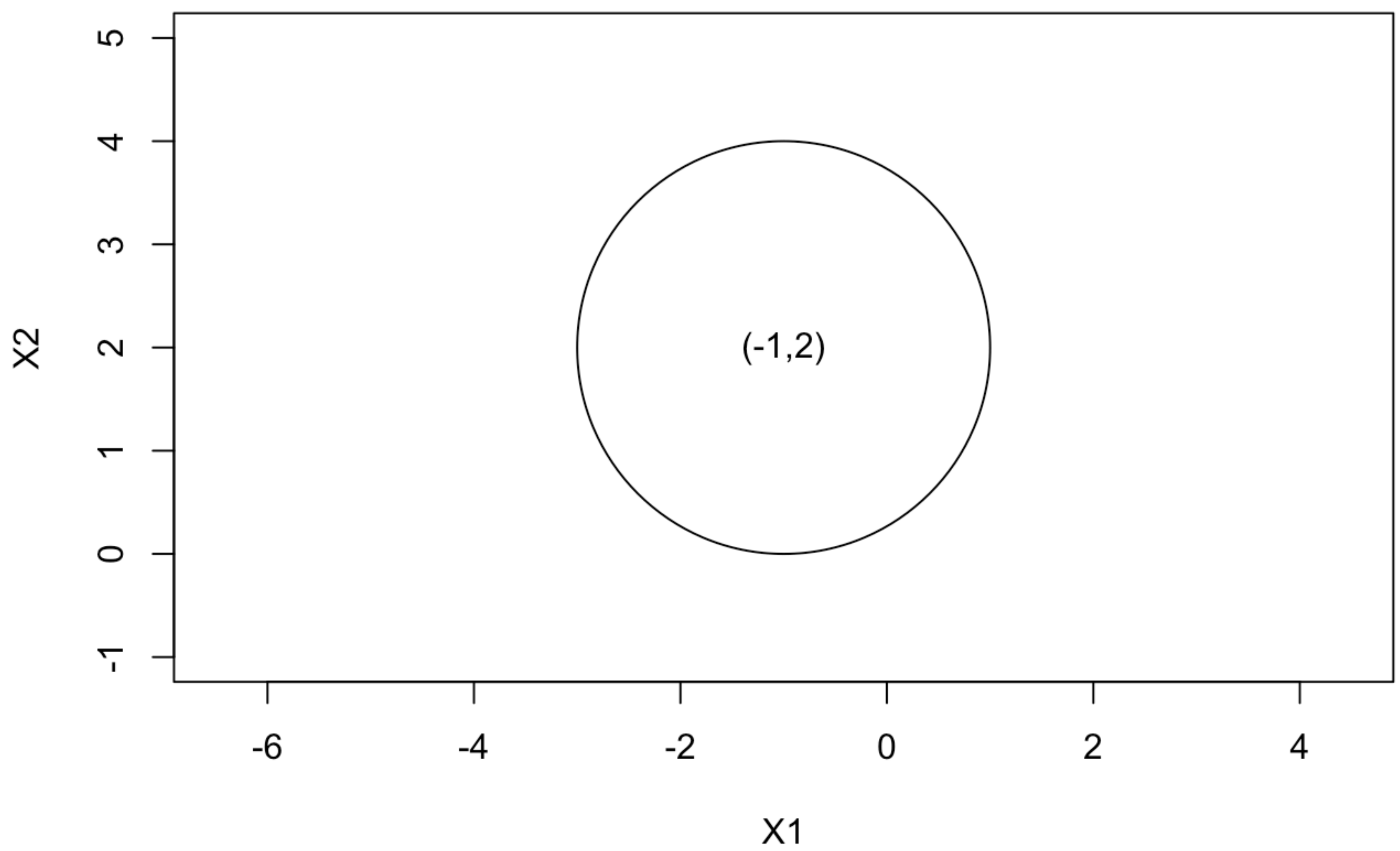
Please do the following problems from the text book ISLR. (use set.seed(702) to replicate your results).

1. Question 9.7.2 pg 368 (2.) We have seen that in $p = 2$ dimensions, a linear decision boundary takes the form $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$.We now investigate a non-linear decision boundary.

(a) Sketch the curve

$$(1 + X_1)^2 + (2 - X_2)^2 = 4$$

Ans Circle with center (h,k) and radius r is $(x - h)^2 + (y - k)^2 = r^2$, Hence given circle has center at (-1,2) and radius 2.



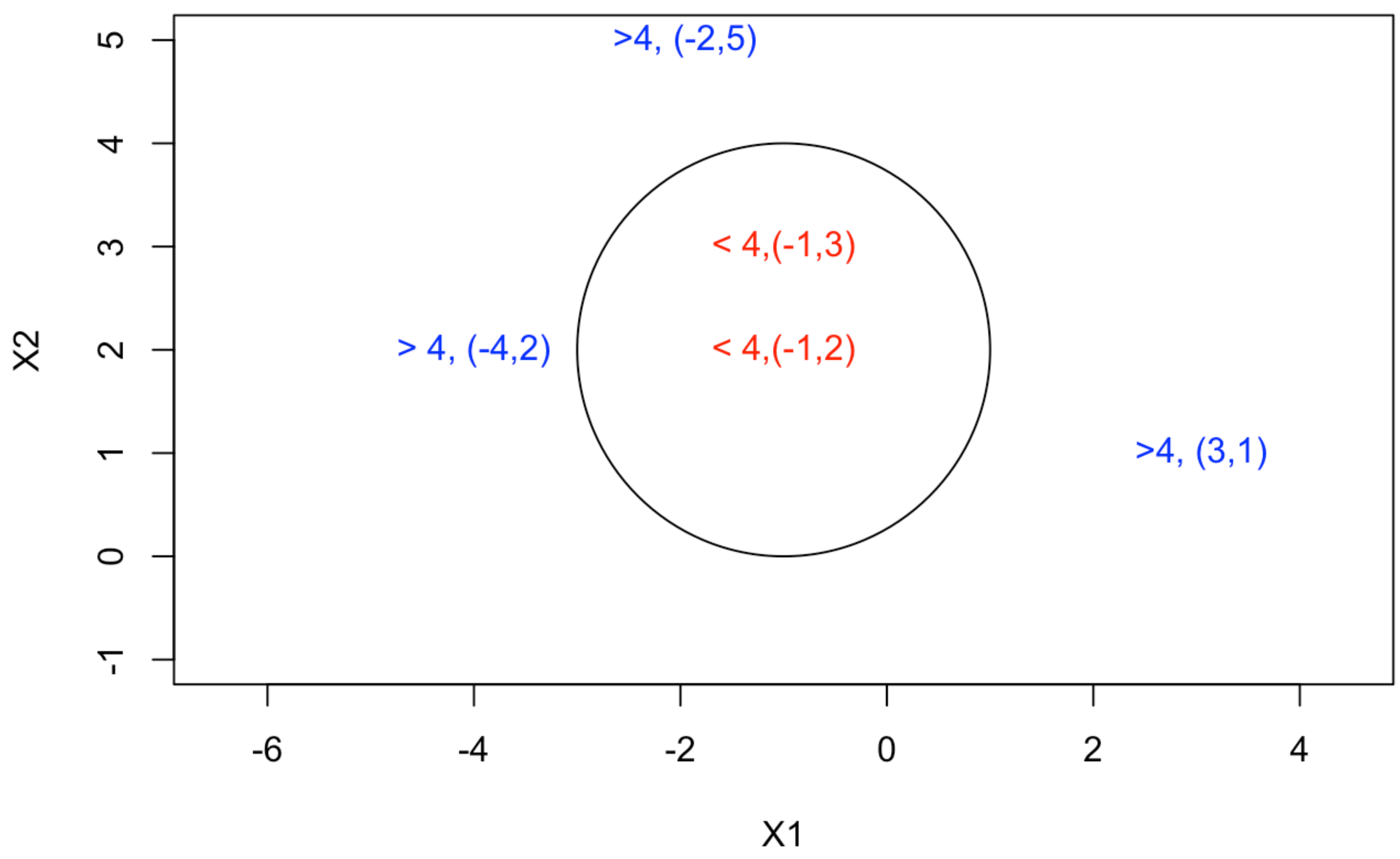
(b) On your sketch, indicate the set of points for which

$$(1 + X_1)^2 + (2 - X_2)^2 > 4,$$

as well as the set of points for which

$$(1 + X_1)^2 + (2 - X_2)^2 \leq 4.$$

Ans Any points of x1 and x2 which yeilds $r^2 < 4$ come uner the circle otherwise outside of circle.

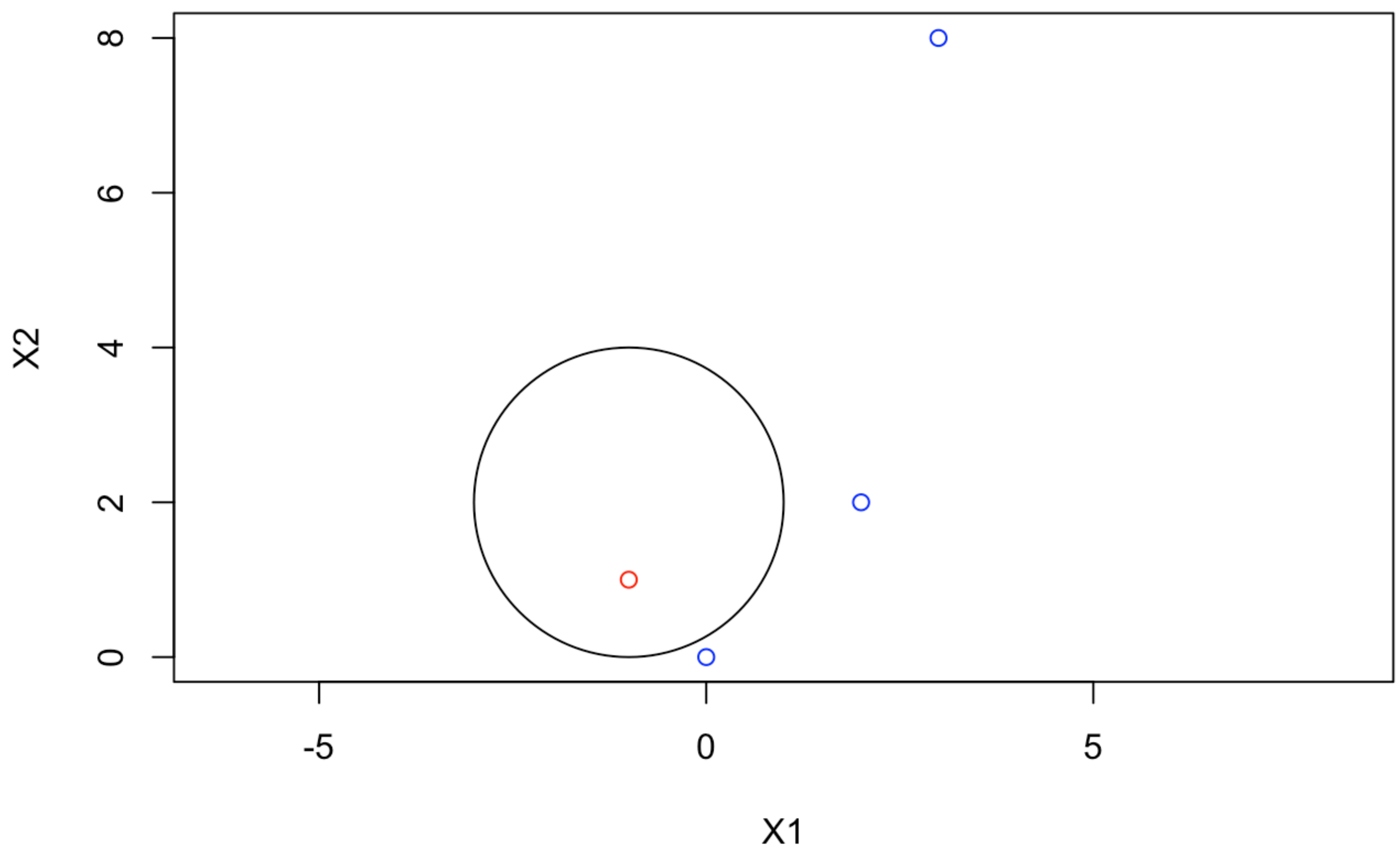


(c) Suppose that a classifier assigns an observation to the blue class if

$$(1 + X_1)^2 + (2 - X_2)^2 > 4,$$

and to the red class otherwise. To what class is the observation $(0, 0)$ classified? $(-1, 1)?(2, 2)?(3, 8)?$

ANS: By substituting given values of X_1 and X_2 in the equation we can check if the result is less or greater than 4. Any result which is less than 4 is red class and the ones that are greater than 4 belong to the blue class. For $(0, 0)$, we have $5 > 4$ (blue class), for $(-1, 1)$, we have $1 < 4$ (red class), for $(2, 2)$, we have $9 > 4$ (blue class), for $(3, 8)$, we have $52 > 4$ (blue class).



(d) Argue that while the decision boundary in (c) is not linear in terms of X_1 and X_2, it is linear in terms of X_1, X_1^2, X_2 and X_2^2. **ANS:** If we expand the equation of the decision boundary

$$(1 + X_1)^2 + (2 - X_2)^2 = 4$$

by

$$X_1^2 + X_2^2 + 2X_1 - 4X_2 + 1 = 0$$

which is sum of quadratic term. By enlarging the feature space by using quadratic function of predictors the result is linear in terms of X_1, X_1^2, X_2 and x_2^2.

2. Question 9.7.7 pg 371 (7.) In this problem, you will use support vector approaches in order to predict whether a given car gets high or low gas mileage based on the Auto data set. **(a)** Create a binary variable that takes on a 1 for cars with gas mileage above the median, and a 0 for cars with gas mileage below the median.

mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name	mpg01
18	8	307	130	3504	12.0	70	1	chevrolet chevelle malibu	0
15	8	350	165	3693	11.5	70	1	buick skylark 320	0
18	8	318	150	3436	11.0	70	1	plymouth satellite	0
16	8	304	150	3433	12.0	70	1	amc rebel sst	0
17	8	302	140	3449	10.5	70	1	ford torino	0
15	8	429	198	4341	10.0	70	1	ford galaxie 500	0

(b)

Fit a support vector classifier to the data with various values of cost, in order to predict whether a car gets high or low gas mileage. Report the cross-validation errors associated with different values of this parameter. Comment on your results.

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##     1
##
## - best performance: 0.01262821
##
## - Detailed performance results:
##    cost      error dispersion
## 1 1e-02 0.07641026 0.05638257
## 2 1e-01 0.05346154 0.04723353
## 3 1e+00 0.01262821 0.01778017
## 4 5e+00 0.02025641 0.01979514
## 5 1e+01 0.02025641 0.01979514
## 6 1e+02 0.03557692 0.01732022
```

Discussion: We see that cost= 1 results in the lowest cross-validation error rate= 0.01262821.

(c)

Now repeat (b), this time using SVMs with radial and polynomial basis kernels, with different values of gamma and degree and cost. Comment on your results.

Radial Kernel

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##    10   0.1
##
## - best performance: 0.02544872
##
## - Detailed performance results:
##    cost gamma      error dispersion
## 1   0.1 1e-02 0.08666667 0.05280917
## 2   1.0 1e-02 0.07391026 0.05451194
## 3   5.0 1e-02 0.05339744 0.04712306
## 4  10.0 1e-02 0.02801282 0.02231663
## 5   0.1 1e-01 0.07641026 0.05638257
## 6   1.0 1e-01 0.05852564 0.04655379
## 7   5.0 1e-01 0.03307692 0.02397013
## 8  10.0 1e-01 0.02544872 0.02076512
## 9   0.1 1e+00 0.57403846 0.03079482
## 10  1.0 1e+00 0.06115385 0.05284582
## 11  5.0 1e+00 0.06121795 0.05009784
## 12 10.0 1e+00 0.06121795 0.05009784
## 13  0.1 5e+00 0.57403846 0.03079482
## 14  1.0 5e+00 0.51032051 0.05208282
## 15  5.0 5e+00 0.51288462 0.05284517
## 16 10.0 5e+00 0.51288462 0.05284517
## 17  0.1 1e+01 0.57403846 0.03079482
## 18  1.0 1e+01 0.53589744 0.05149617
## 19  5.0 1e+01 0.53333333 0.05056569
## 20 10.0 1e+01 0.53333333 0.05056569
## 21  0.1 1e+02 0.57403846 0.03079482
## 22  1.0 1e+02 0.57403846 0.03079482
## 23  5.0 1e+02 0.57403846 0.03079482
## 24 10.0 1e+02 0.57403846 0.03079482
```

Discussion: In radial karnel, the lowest corssvalidation error rate is 0.02544872 at cost=10 and gama =0.1.

Polynomial Kernel

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost degree
##    10      2
##
## - best performance: 0.5663462
##
## - Detailed performance results:
##    cost degree     error dispersion
## 1   0.1      2 0.5740385 0.03079482
## 2   1.0      2 0.5740385 0.03079482
## 3   5.0      2 0.5740385 0.03079482
## 4  10.0      2 0.5663462 0.03129485
## 5   0.1      3 0.5740385 0.03079482
## 6   1.0      3 0.5740385 0.03079482
## 7   5.0      3 0.5740385 0.03079482
## 8  10.0      3 0.5740385 0.03079482
## 9   0.1      4 0.5740385 0.03079482
## 10  1.0      4 0.5740385 0.03079482
## 11  5.0      4 0.5740385 0.03079482
## 12 10.0      4 0.5740385 0.03079482
```

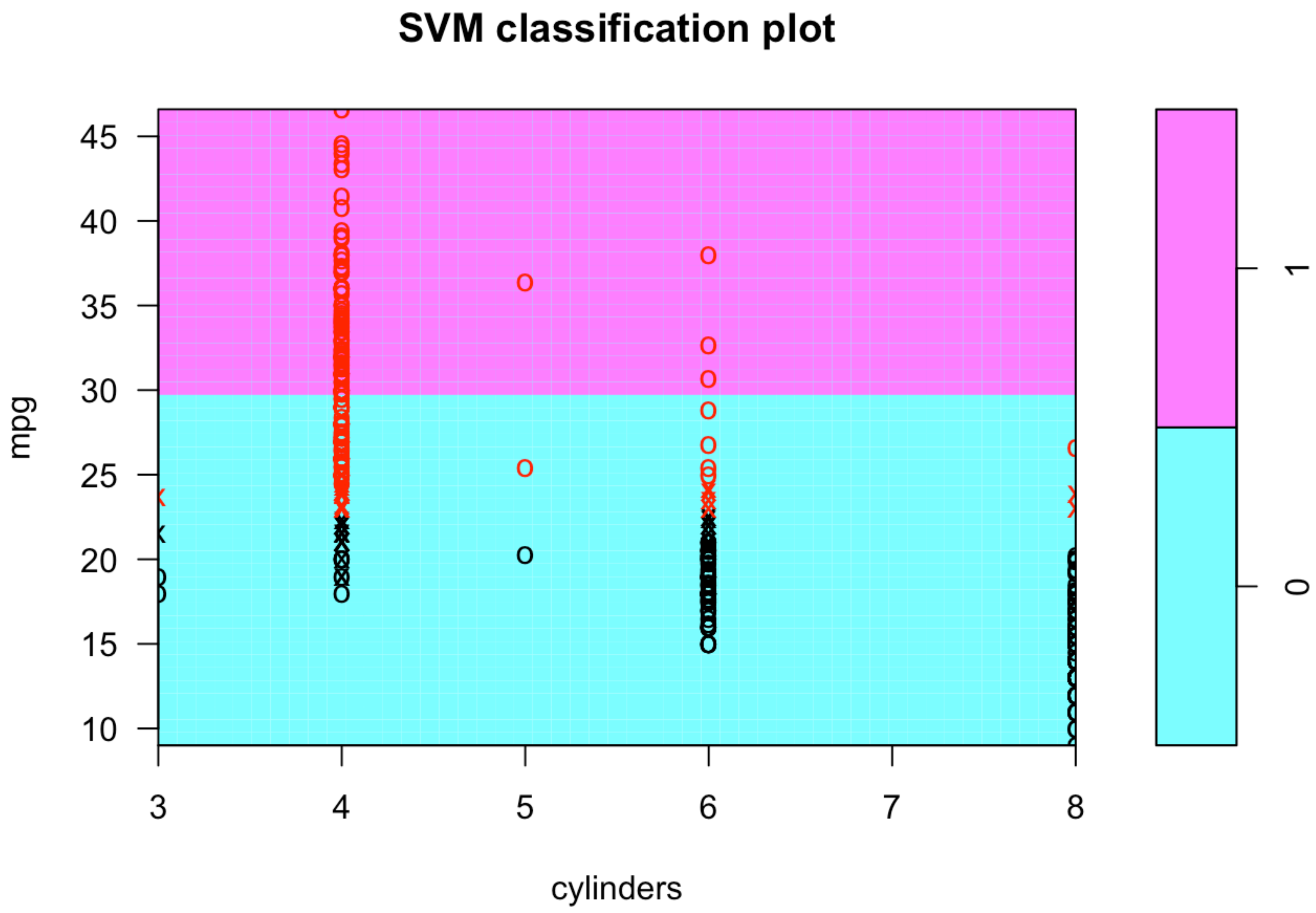
Discussion: In polynomial karnel, the lowest corssvalidation error rate is 0.5663462 at cost=10 and degree =2. Linear kernel produce lowest error rate

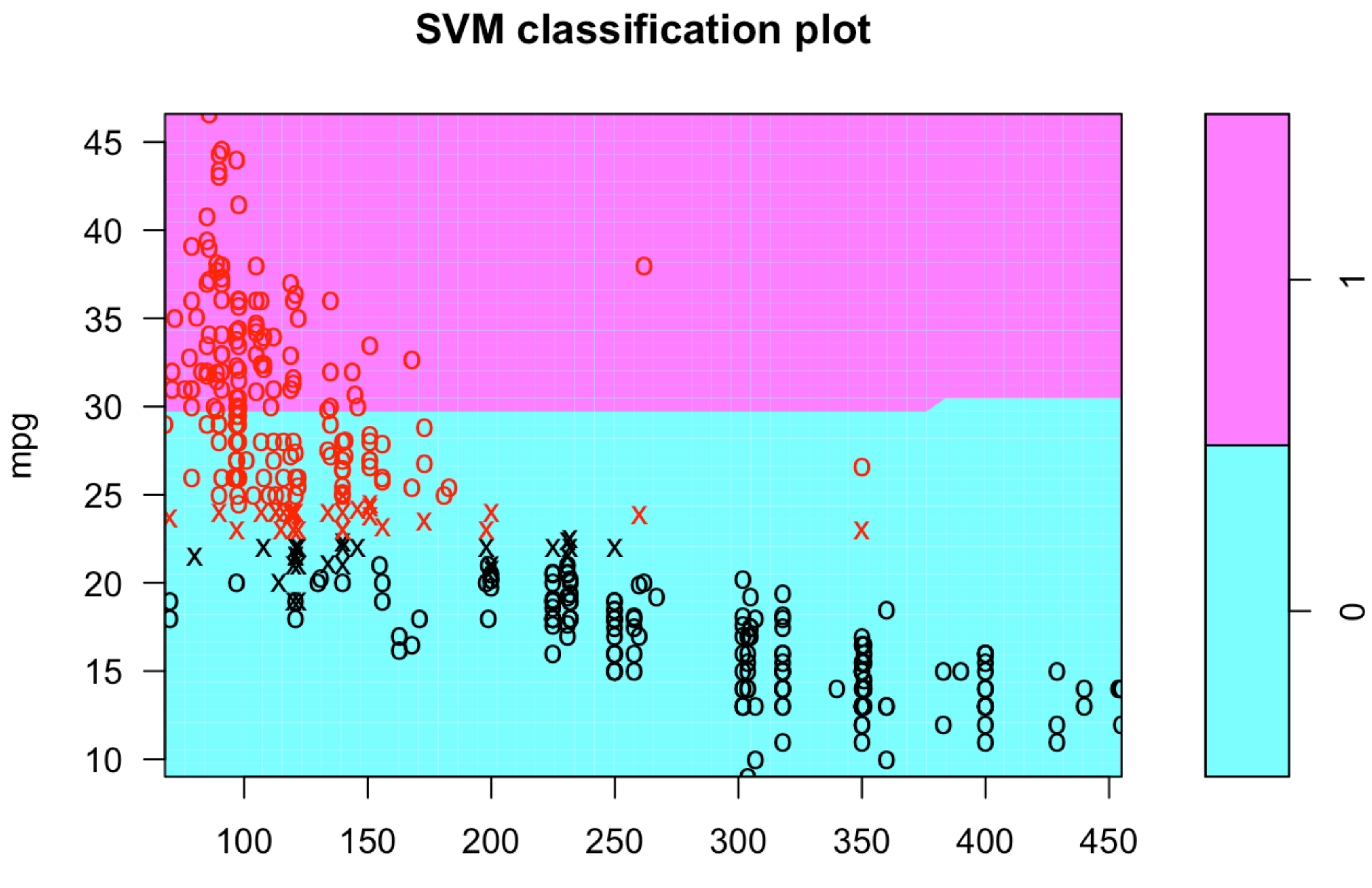
(d) Make some plots to back up your assertions in (b) and (c).

Ans As we can see linear karnel which has lowest cross validation error classified this data well. Polynomial has classification error 56% same suggested by plot, most of the data were missclassified and support vector are around everywhere. Radial one was better than polynomial but still there were larg number of support vector. Hence linear model is best among the other model.

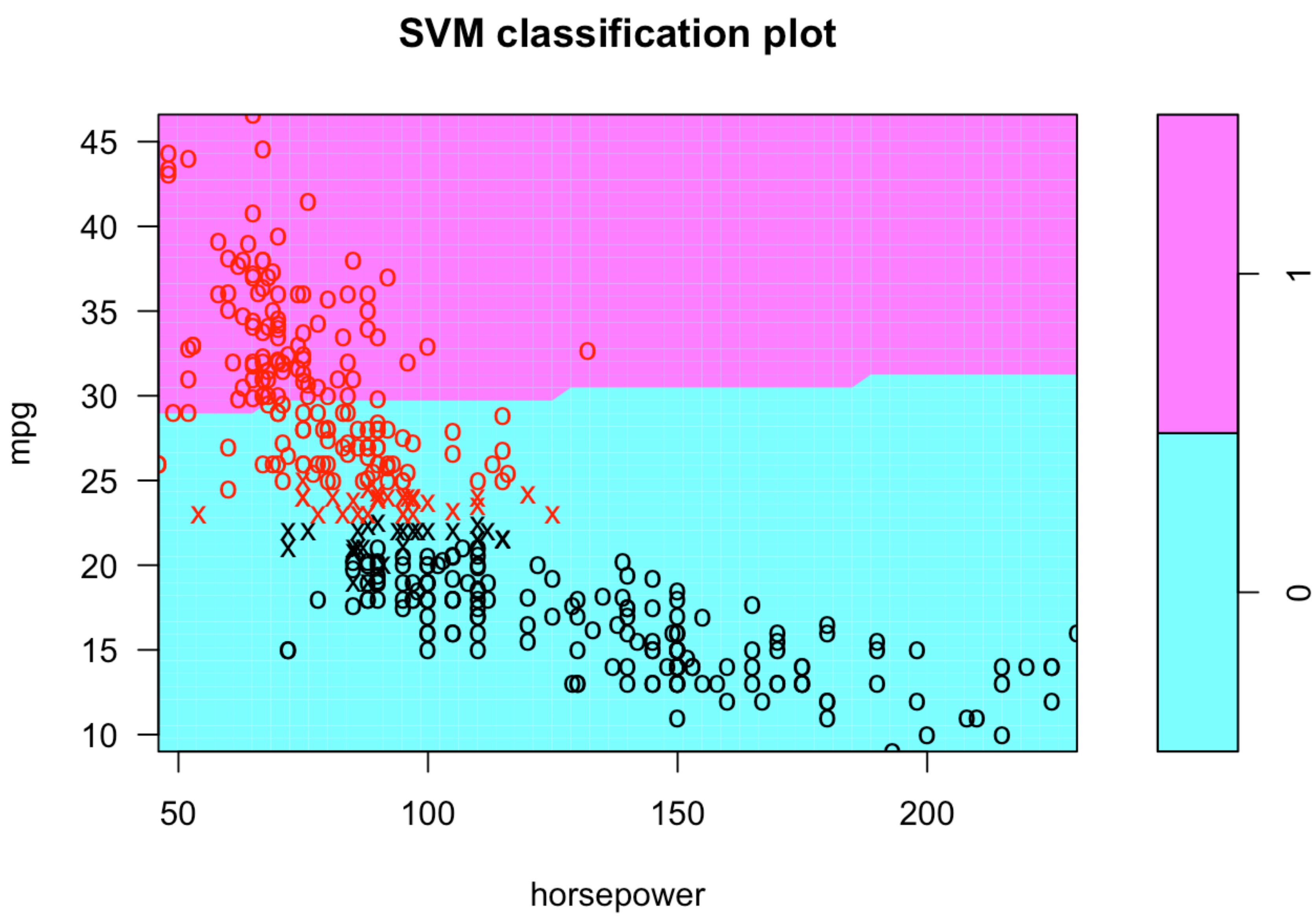
linear kernel

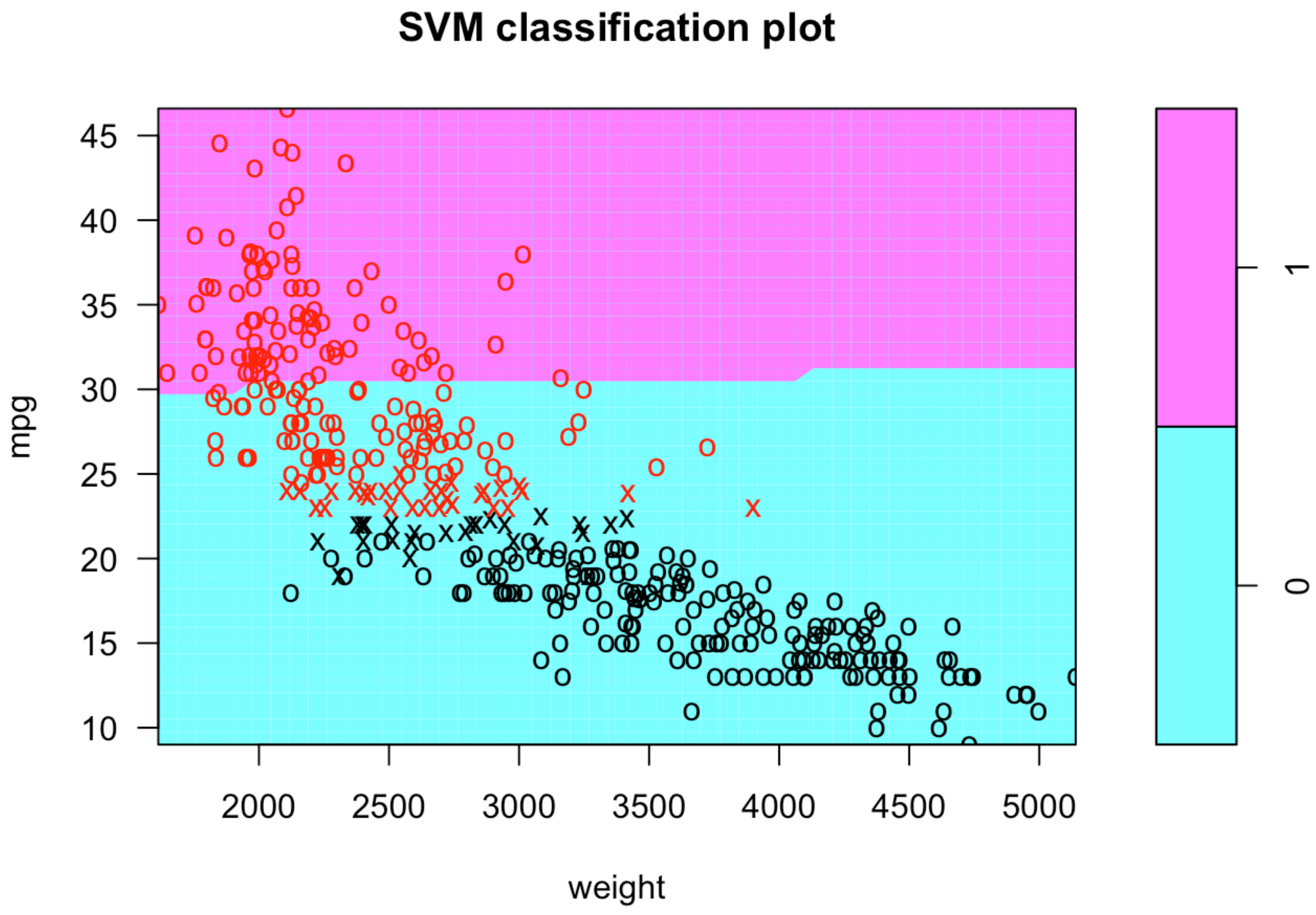
```
##  [1] "mpg"          "cylinders"    "displacement" "horsepower"
##  [5] "weight"       "acceleration" "year"         "origin"
##  [9] "name"         "mpg01"
```

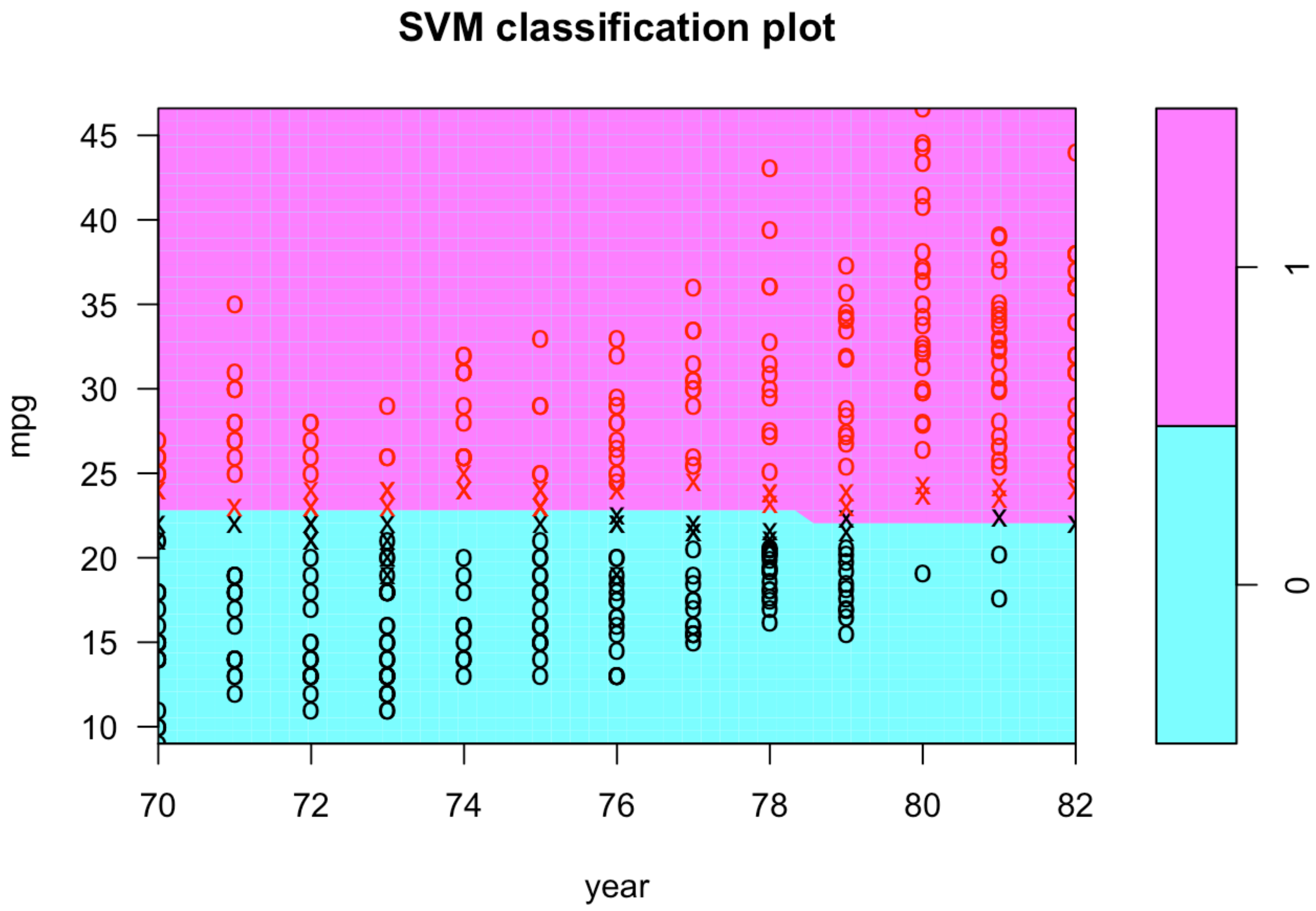

SVM classification plot
mpg
cylinders
45
40
35
30
25
20
15
10
3
4
5
6
7
8
1
0

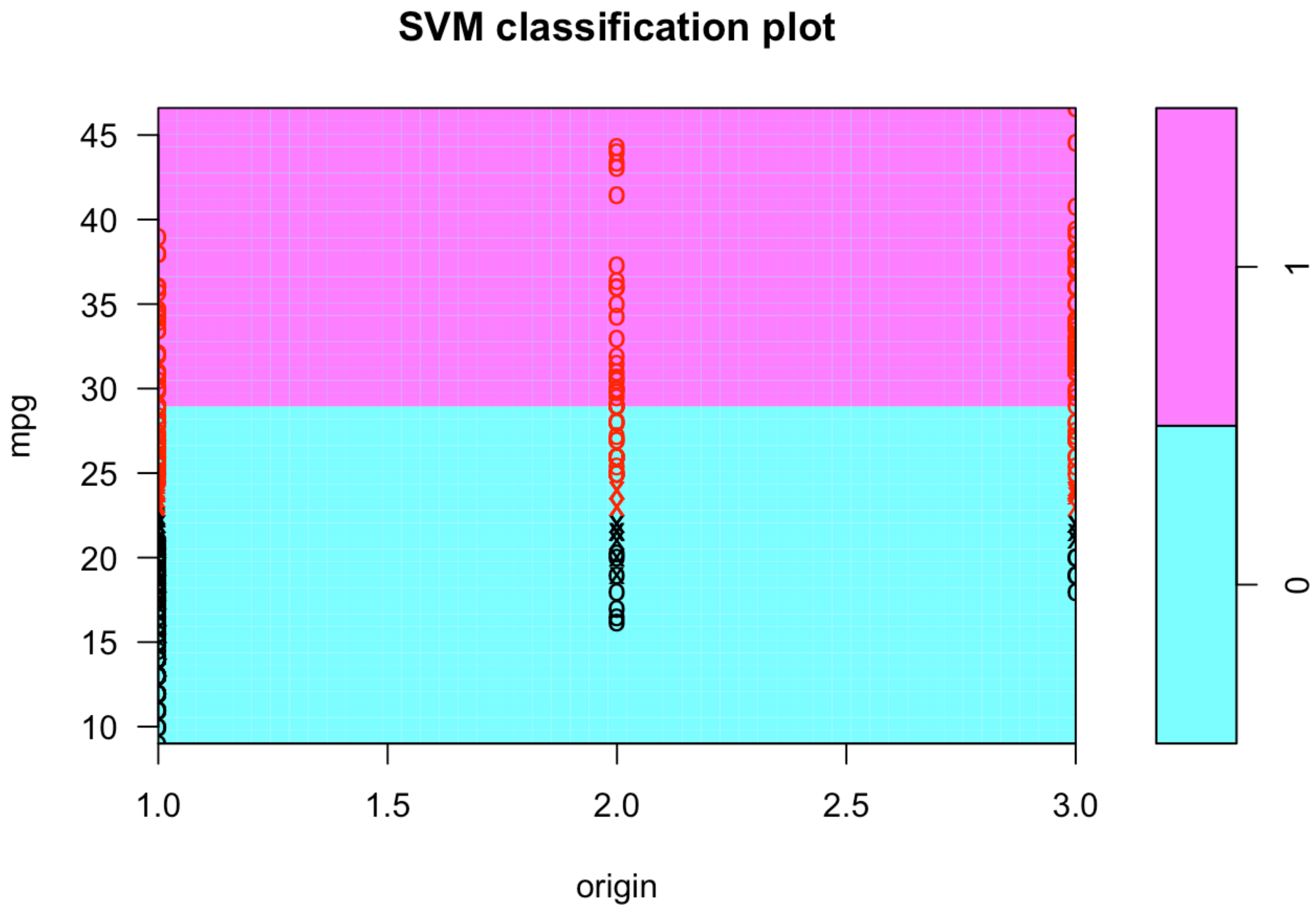

SVM classification plot
mpg
45
40
35
30
25
20
15
10
100
150
200
250
300
350
400
450
1
0

displacement




SVM classification plot
mpg
45
40
35
30
25
20
15
10
2000
2500
3000
3500
4000
4500
5000
weight
1
0


SVM classification plot
mpg
45
40
35
30
25
20
15
10
70
72
74
76
78
80
82
year
1
0

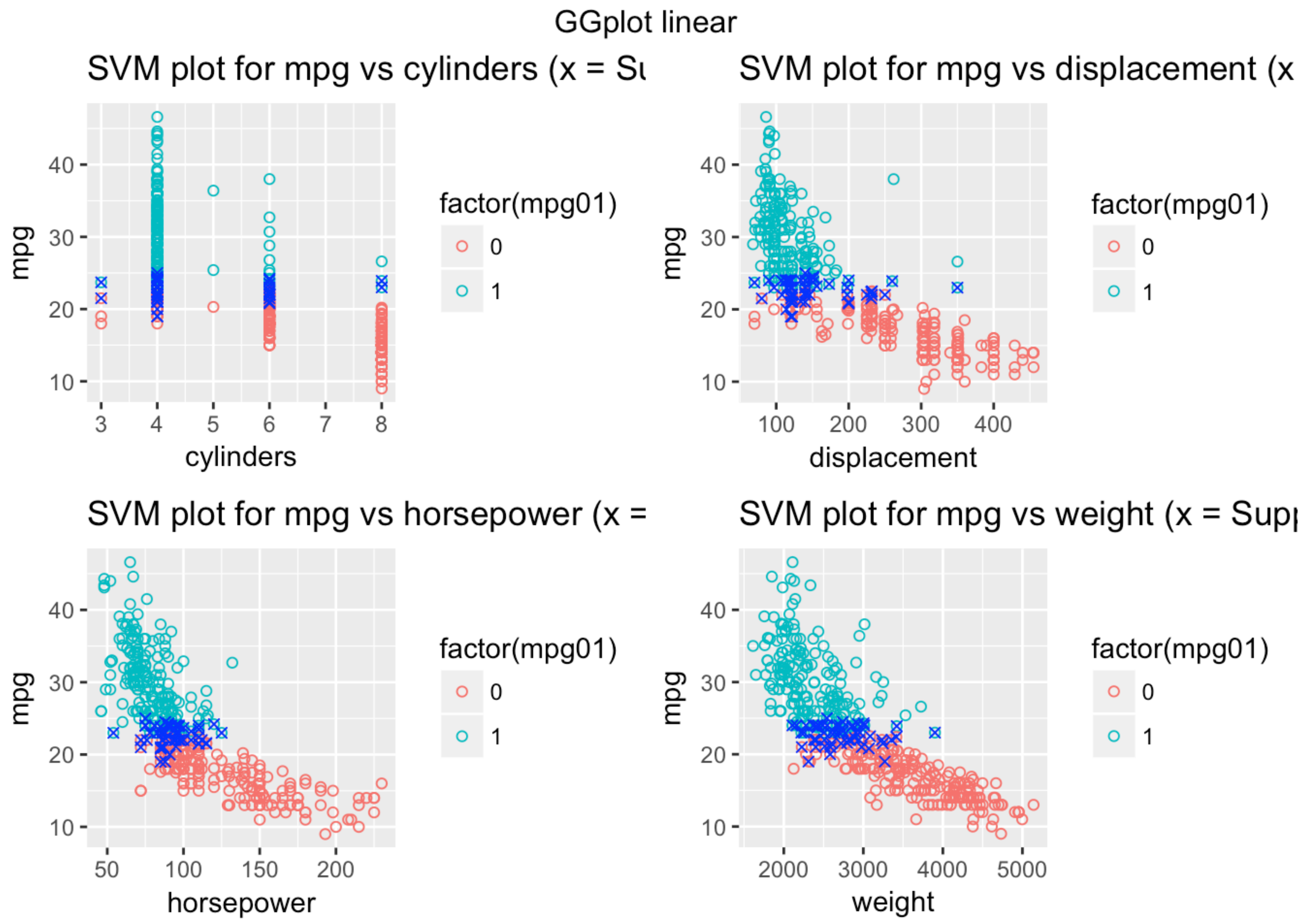

SVM classification plot
mpg
45
40
35
30
25
20
15
10
1.0
1.5
2.0
2.5
3.0
origin
1
0

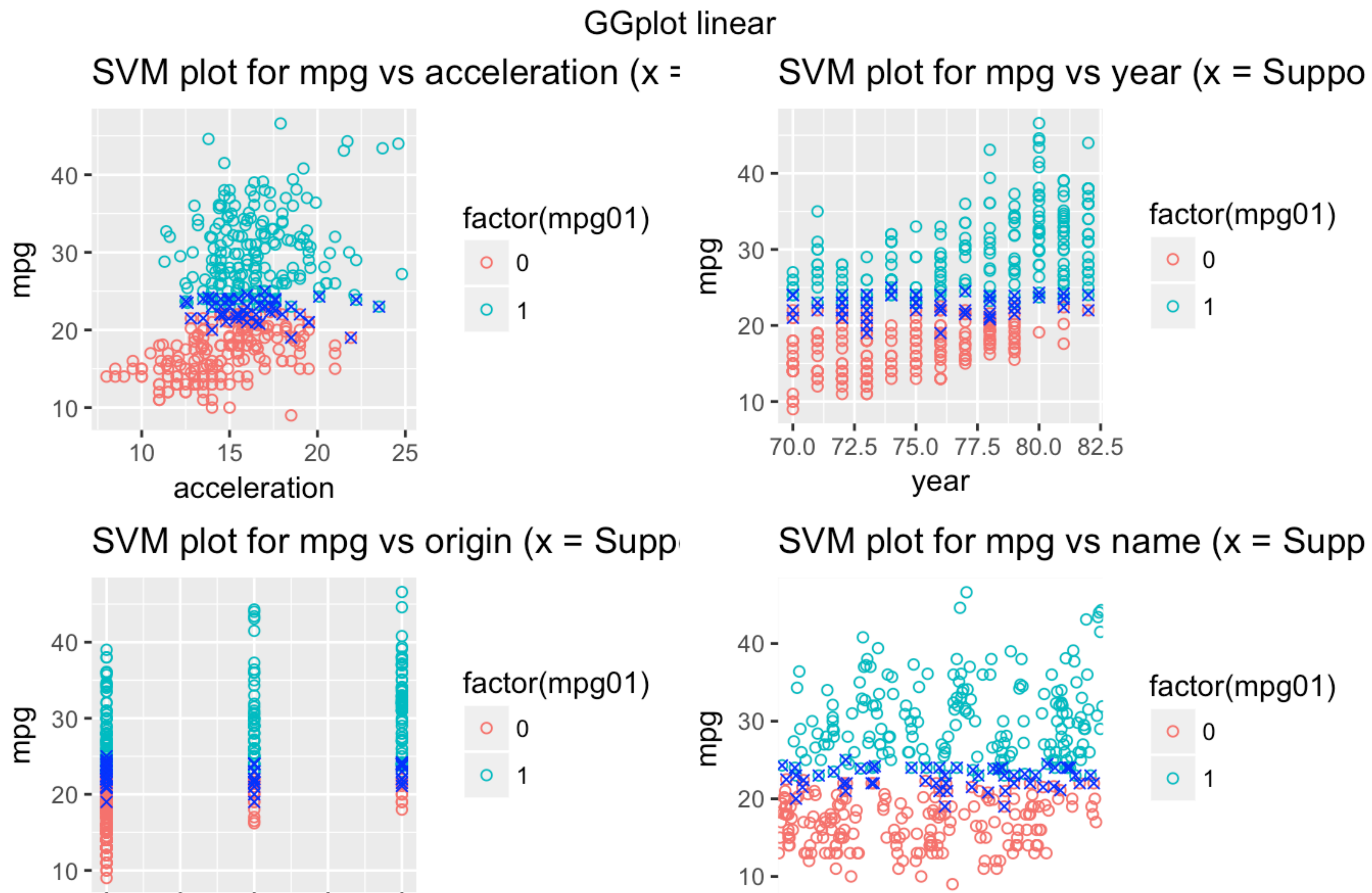
ggplot linear

```
##
## Attaching package: 'ggplot2'
```

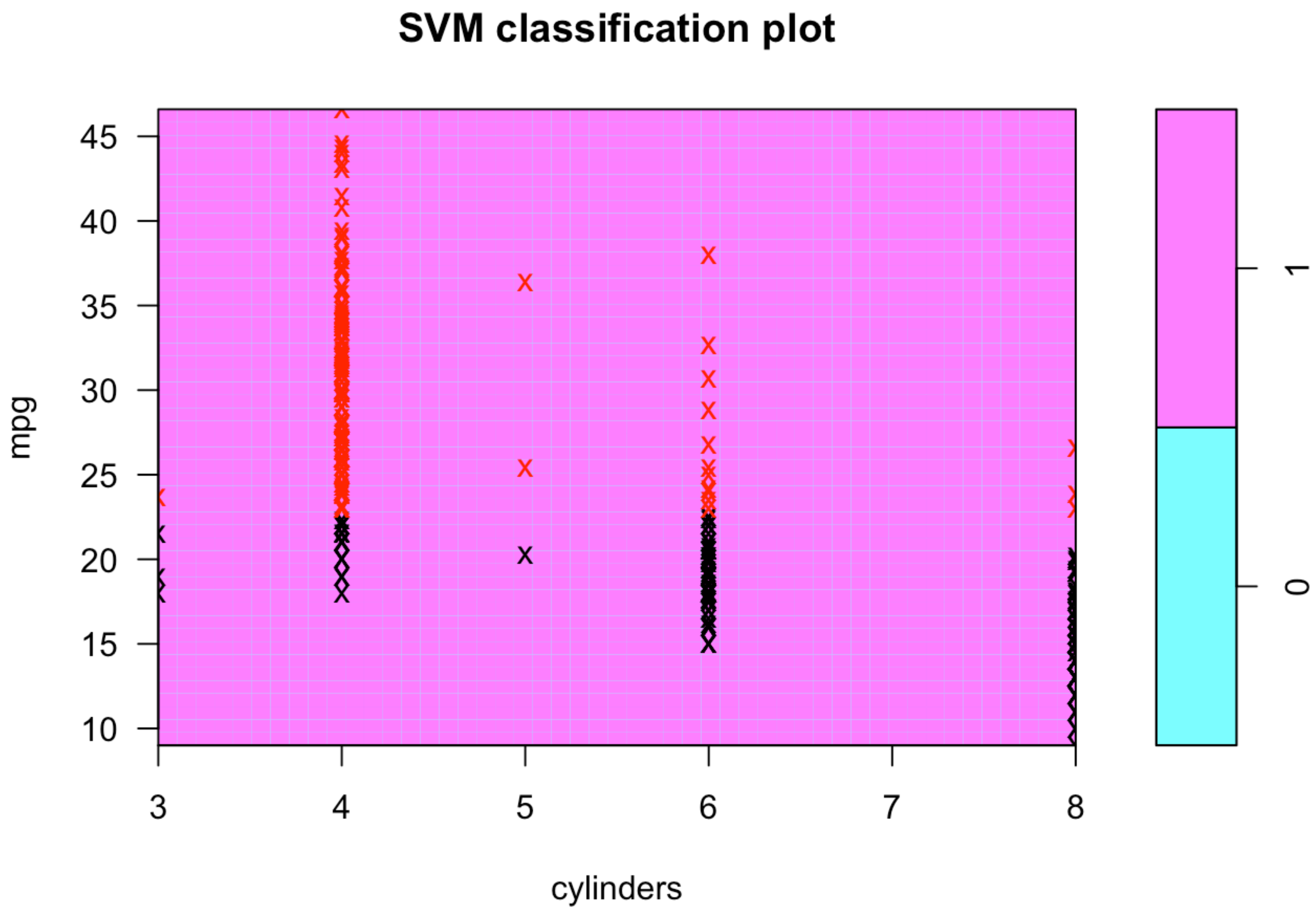
```
## The following object is masked from 'Auto':
##
##     mpg
```

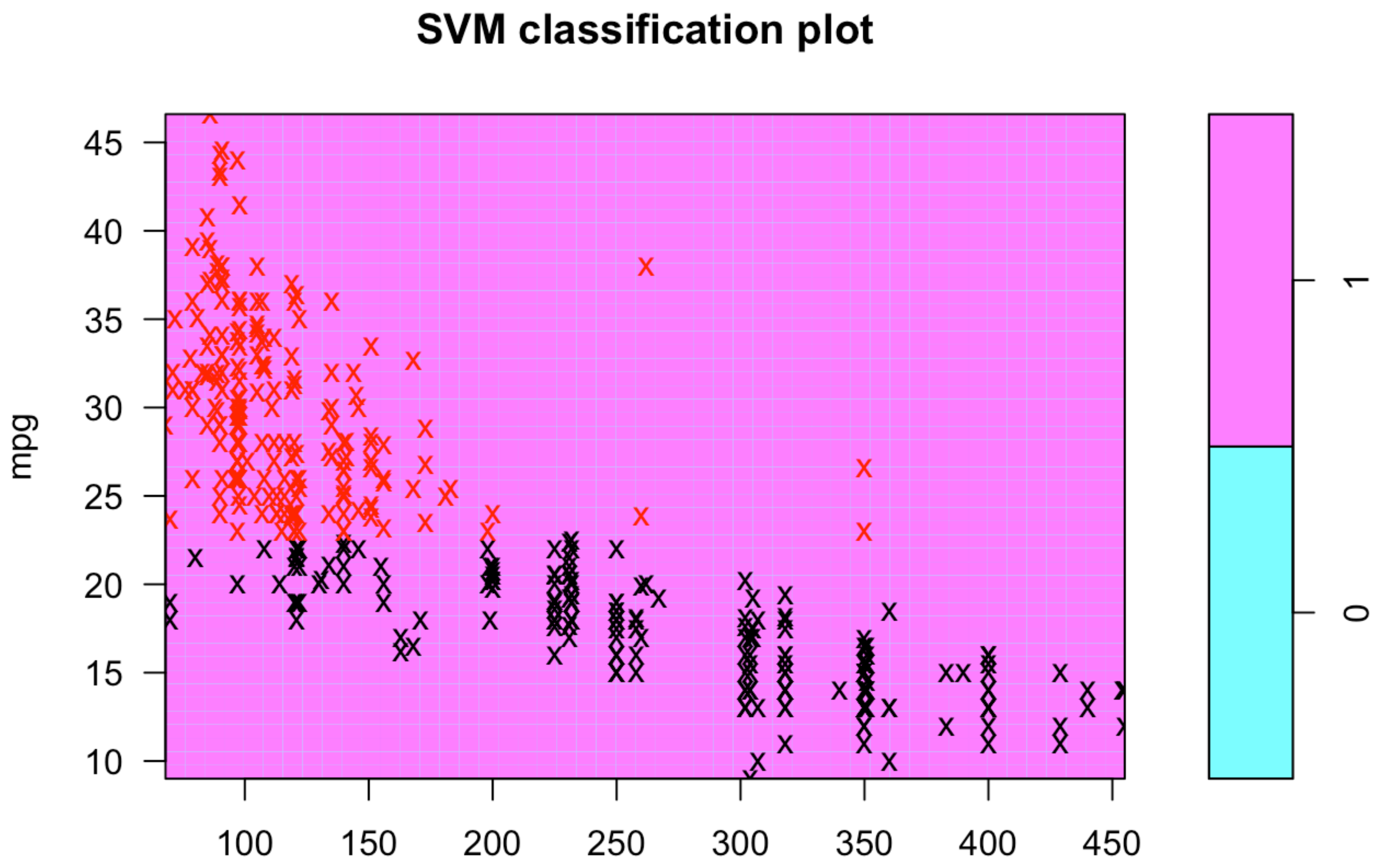
```
##  [1] "mpg"          "cylinders"    "displacement" "horsepower"
##  [5] "weight"       "acceleration" "year"         "origin"
##  [9] "name"         "mpg01"
```



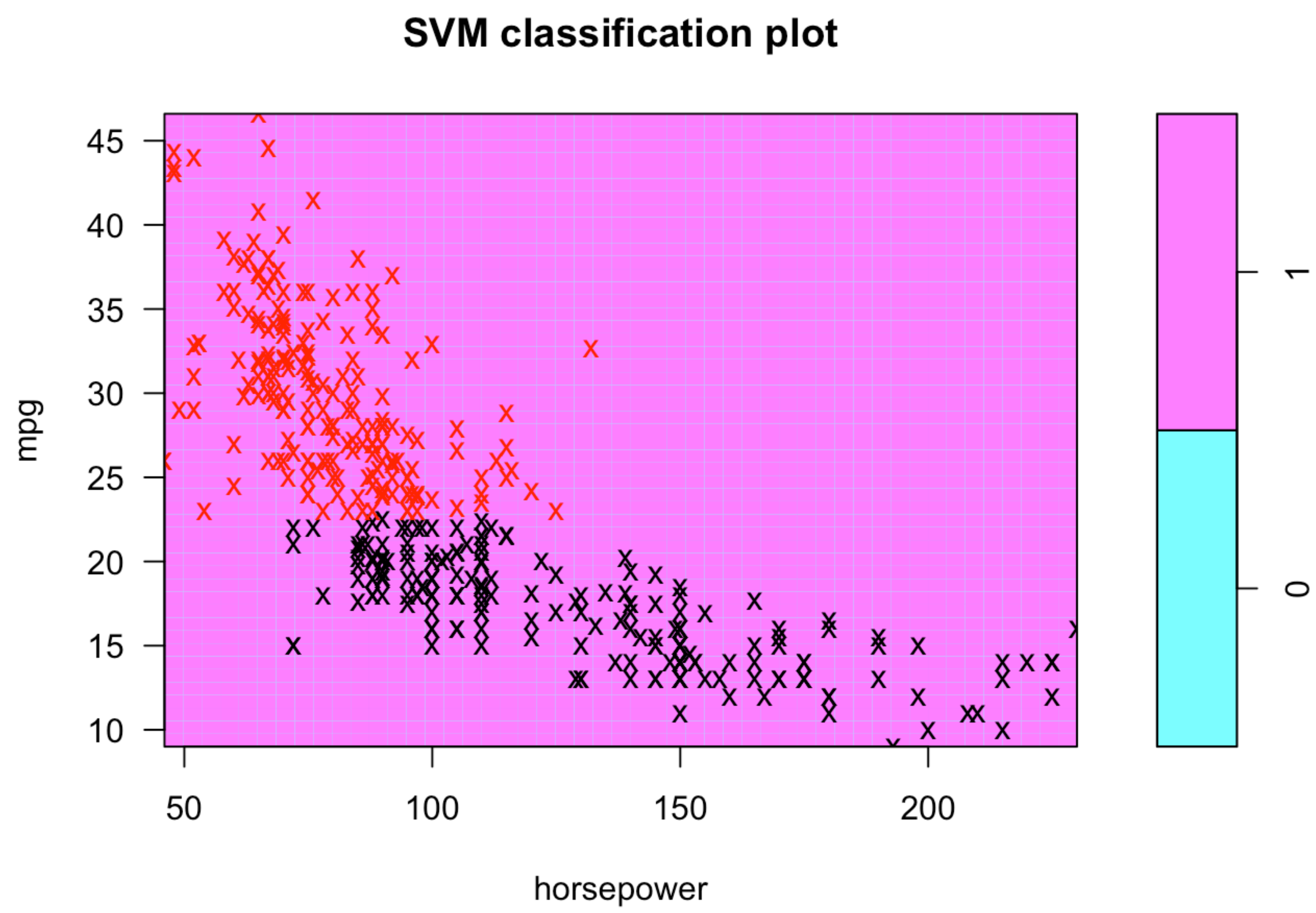


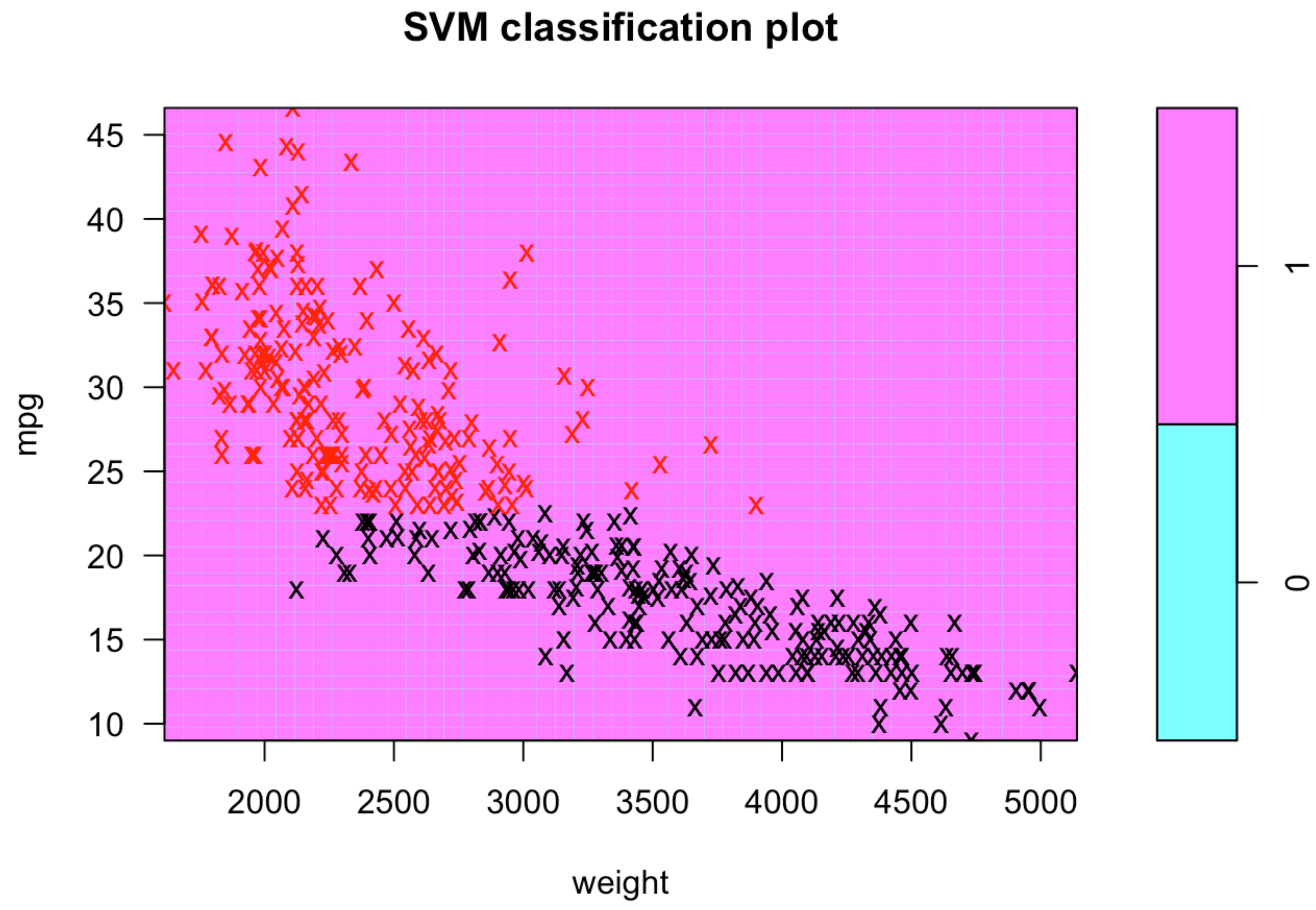
polynomial kernel

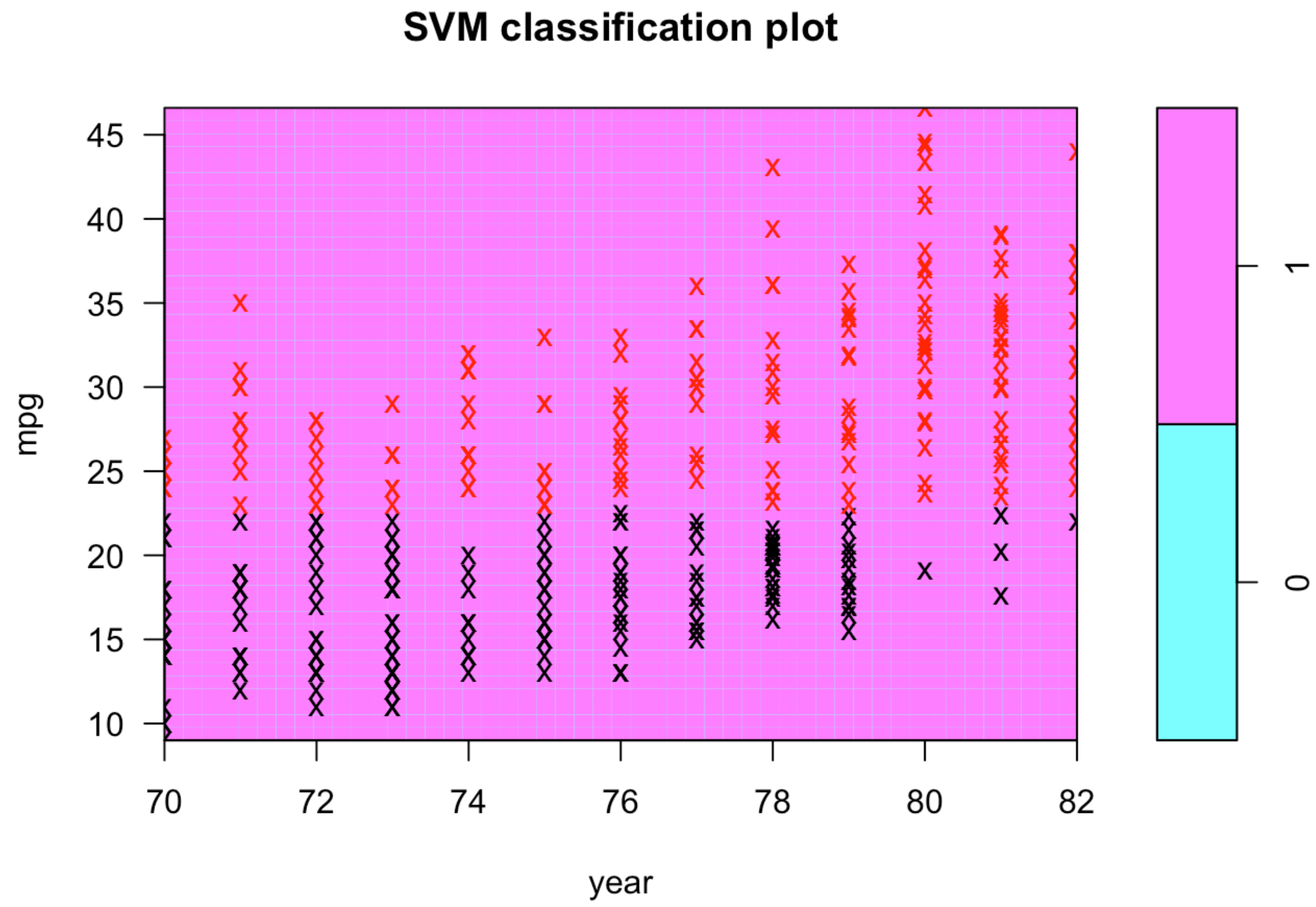


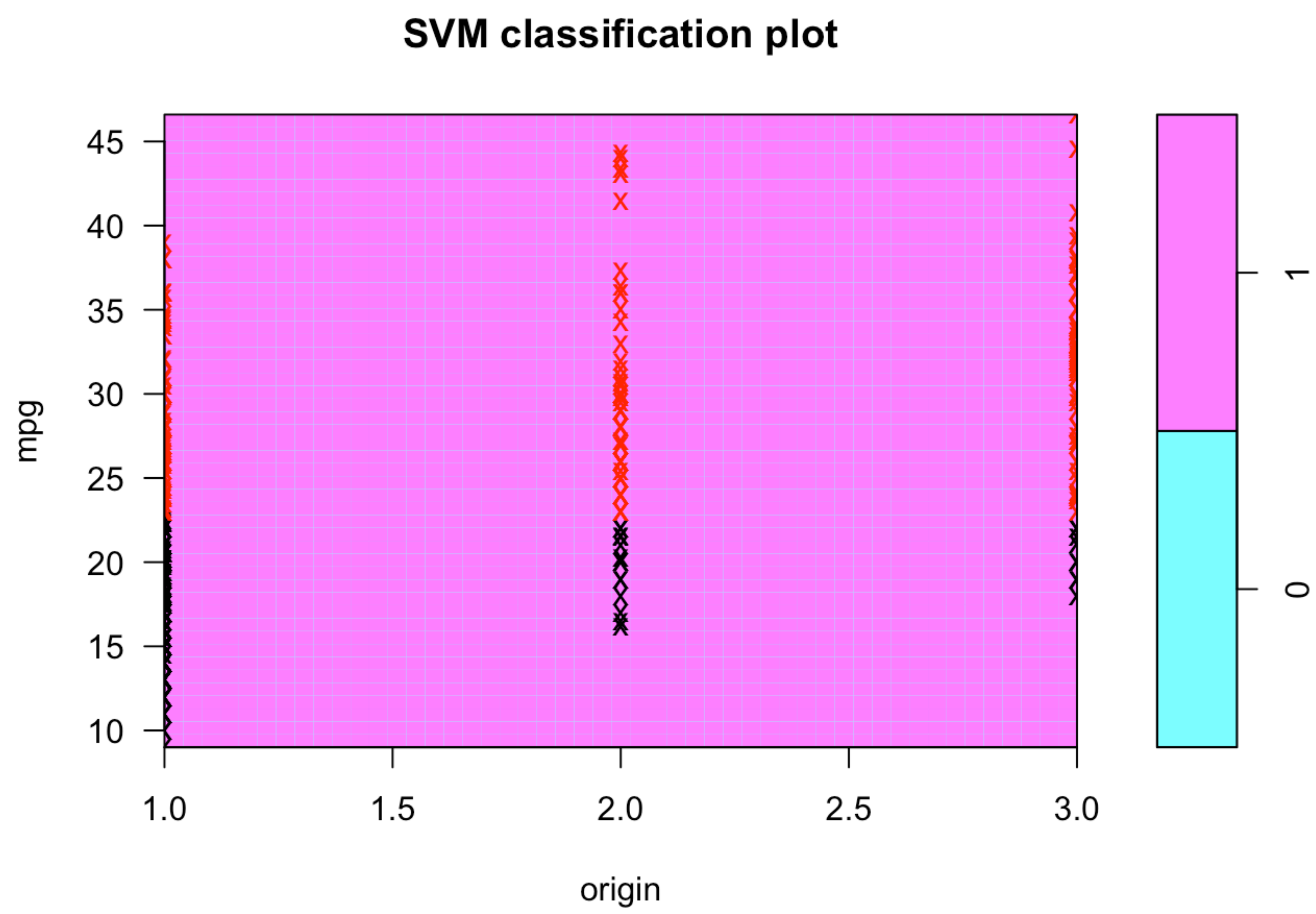


displacement

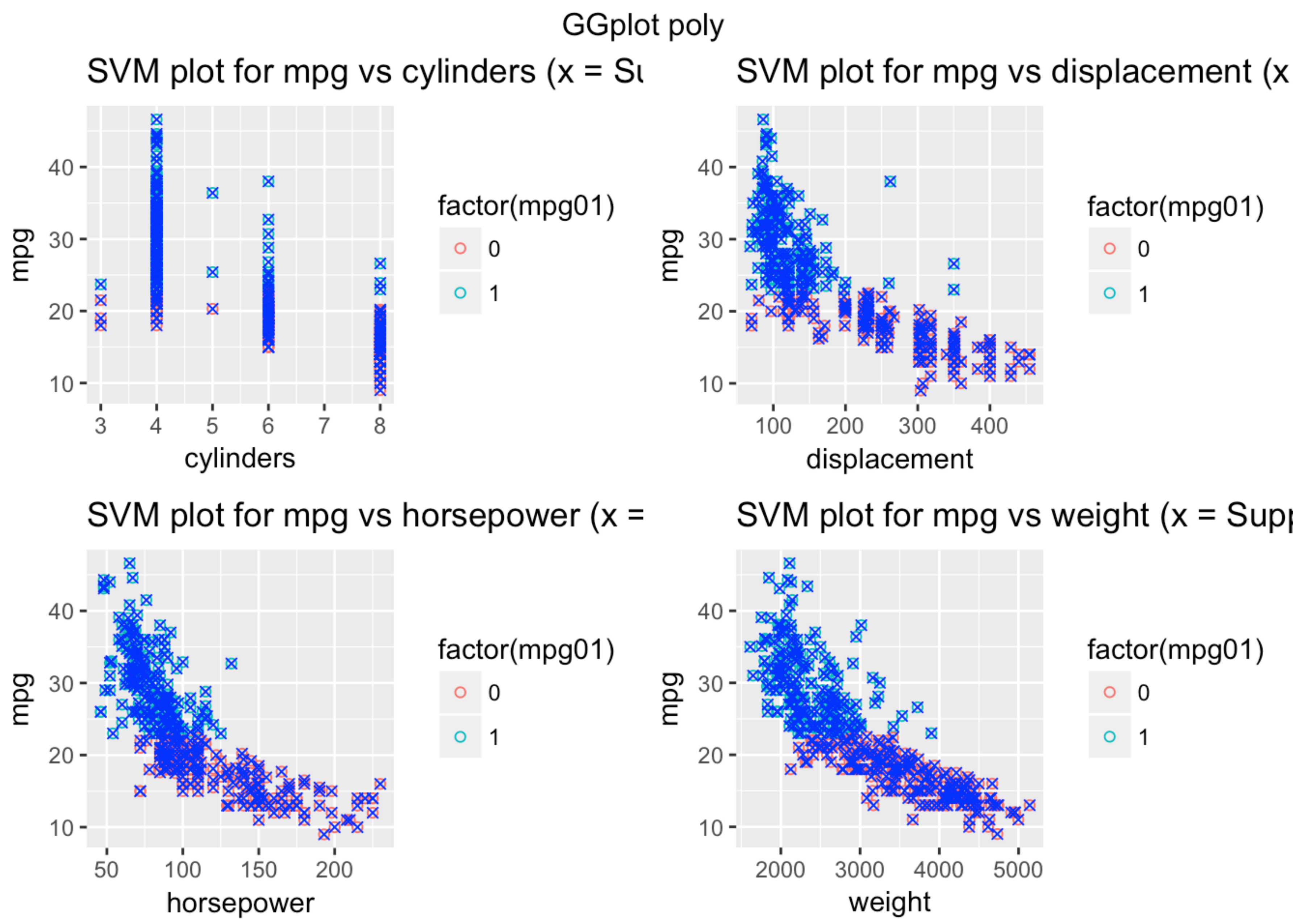


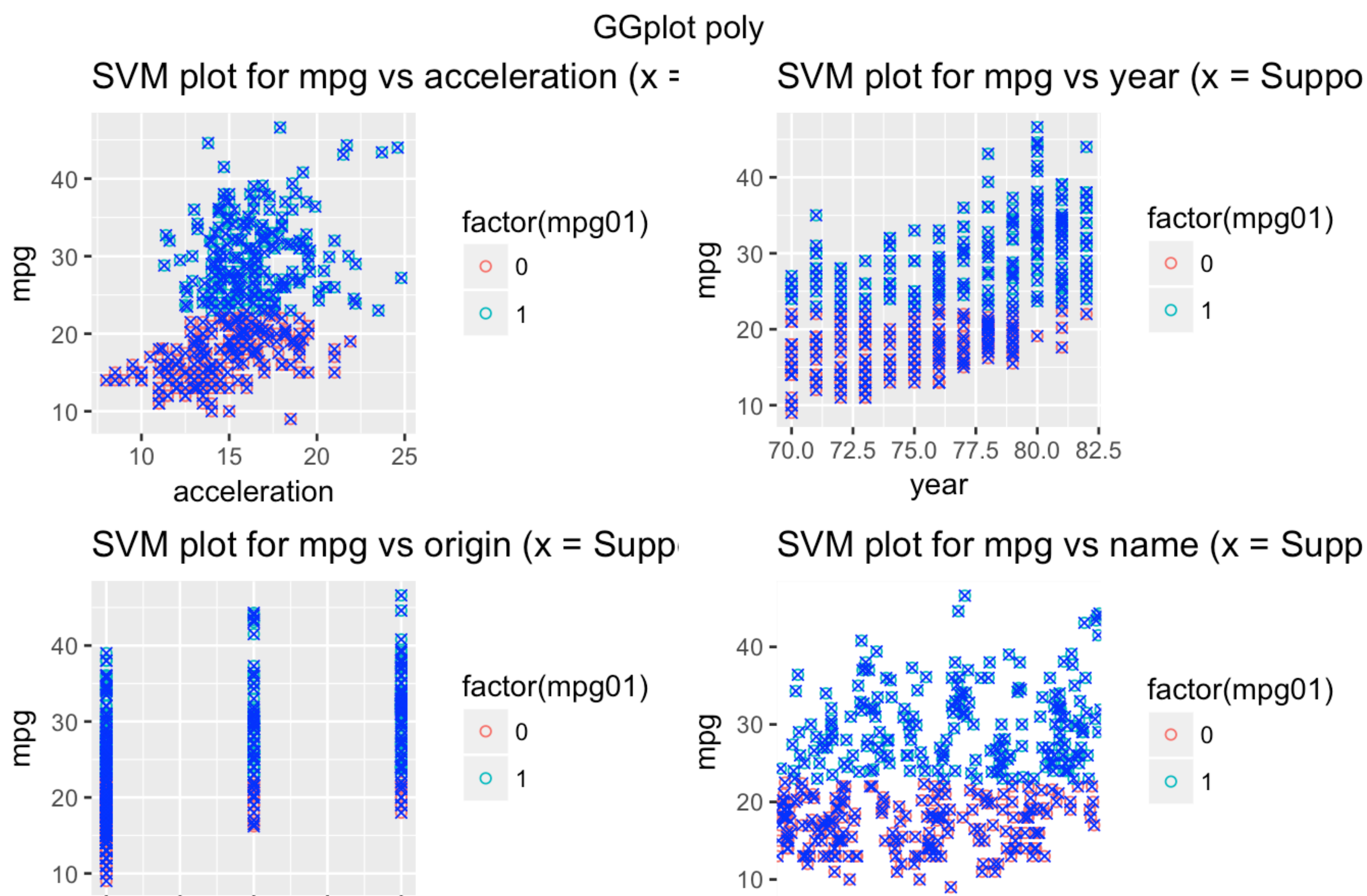

SVM classification plot
mpg
weight
45
40
35
30
25
20
15
10
2000
2500
3000
3500
4000
4500
5000
1
0


SVM classification plot
mpg
year
45
40
35
30
25
20
15
10
70
72
74
76
78
80
82
1
0


SVM classification plot
mpg
origin
45
40
35
30
25
20
15
10
1.0
1.5
2.0
2.5
3.0
1
0

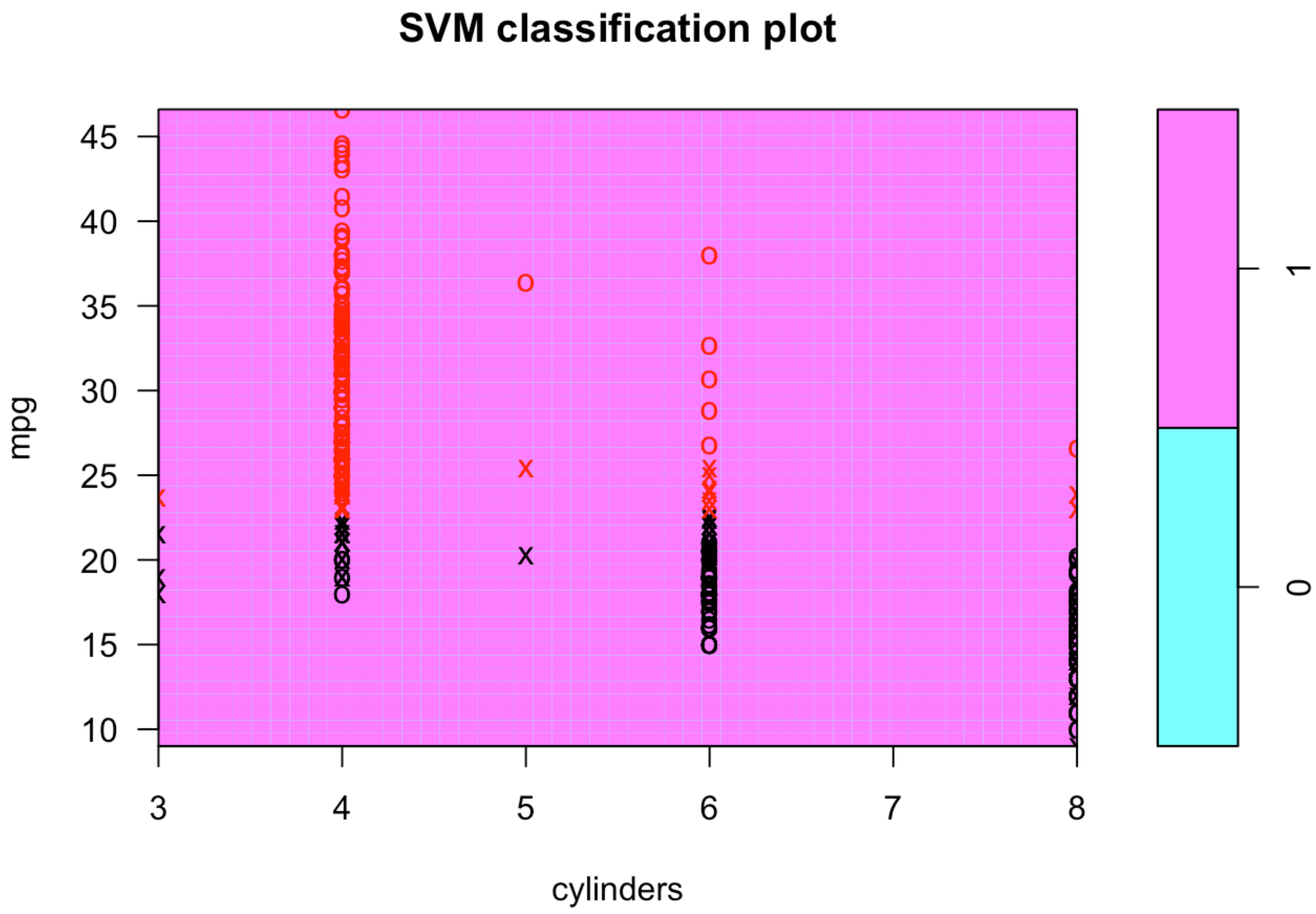
ggplot poly

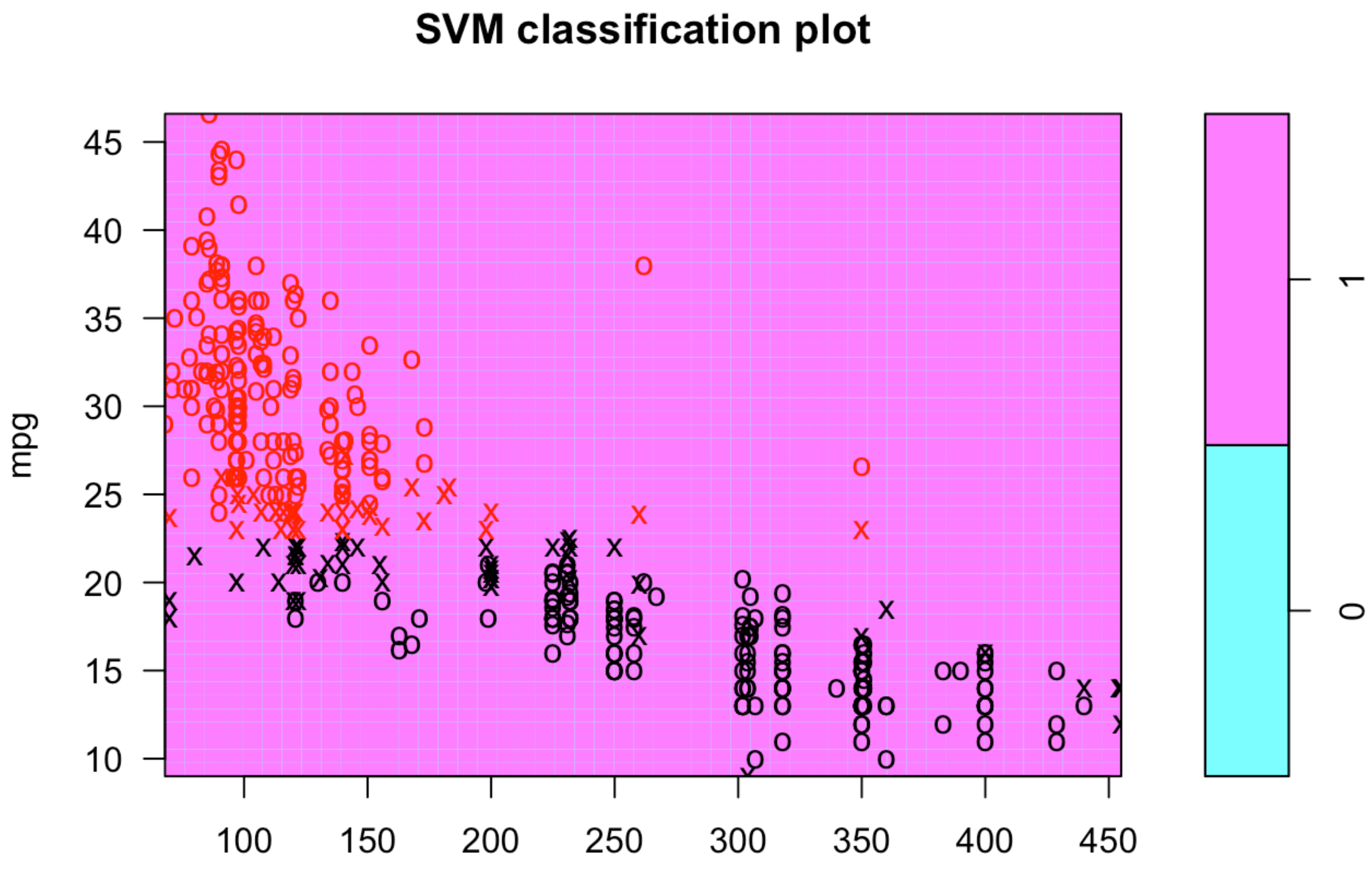




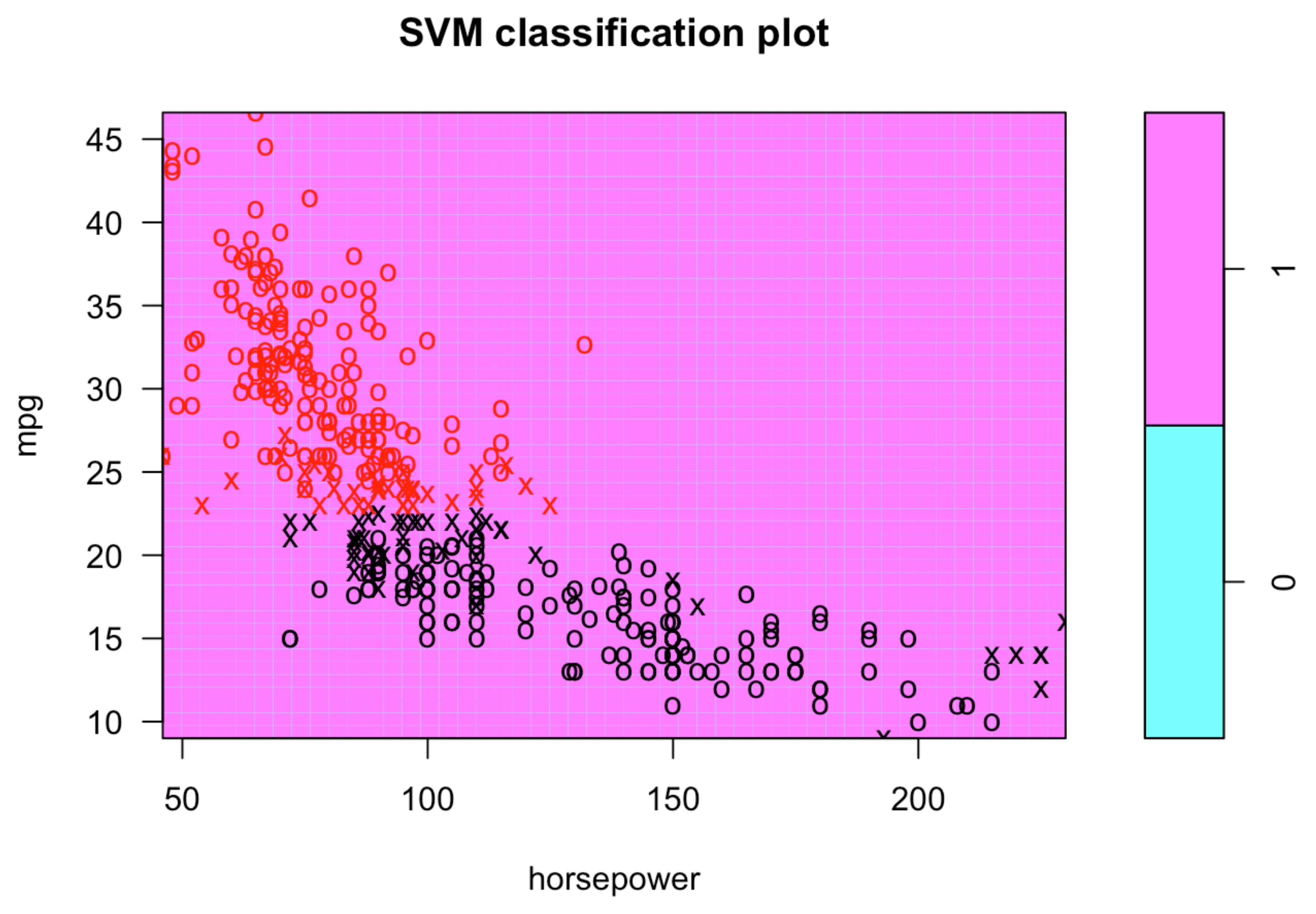
Radial kernel

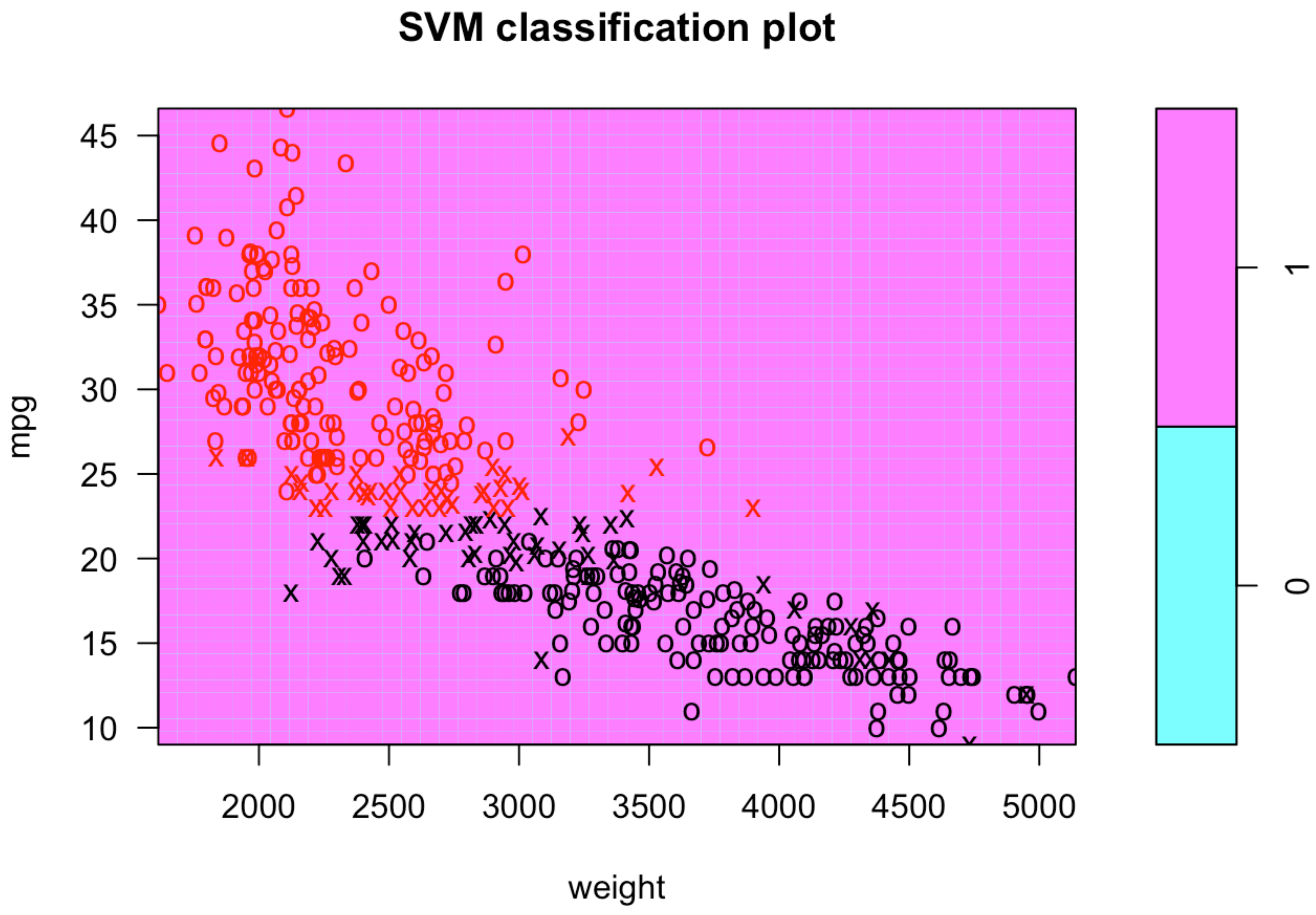

1.0
1.5
2.0
2.5
3.0
origin
name

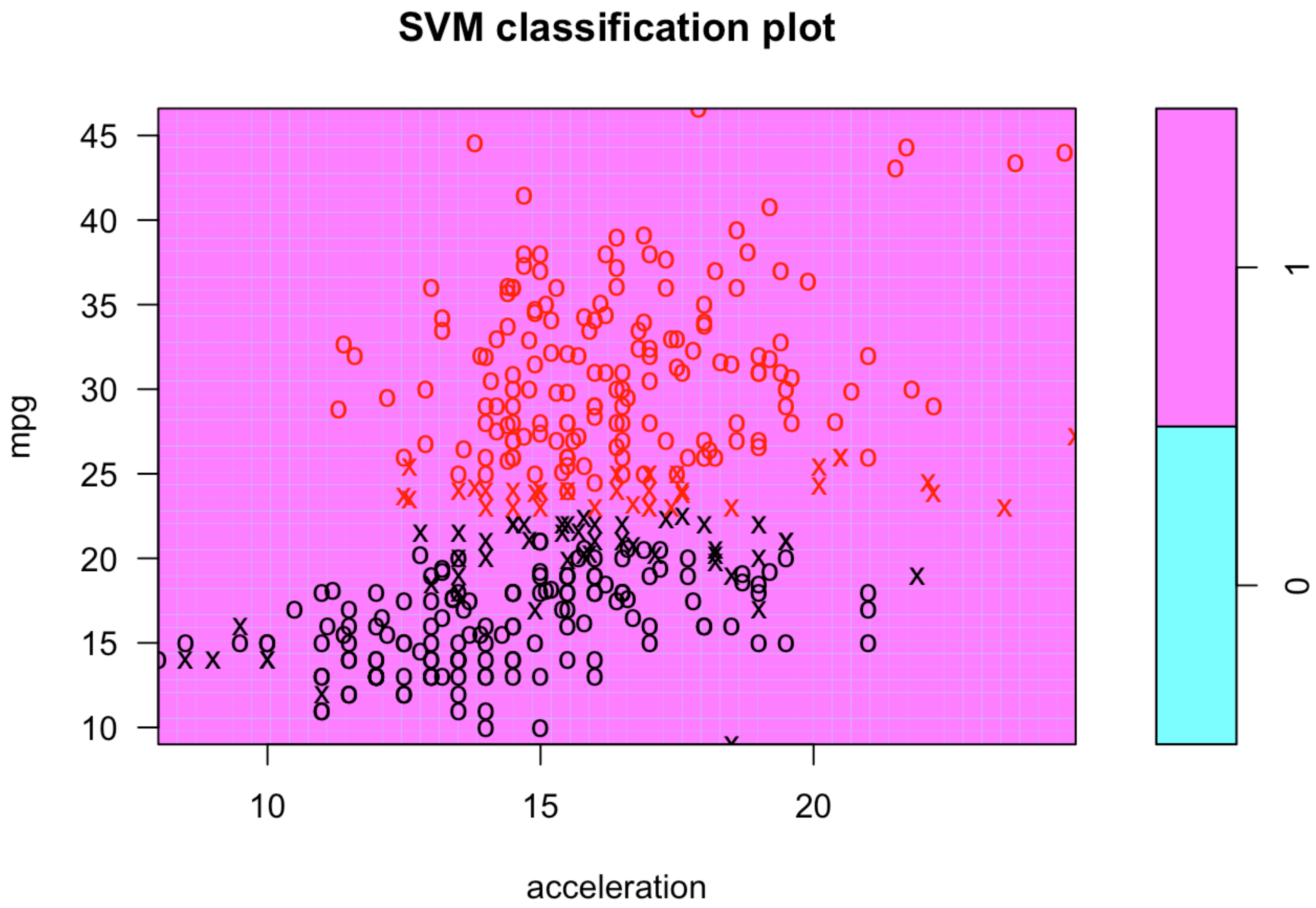

SVM classification plot
mpg
45
40
35
30
25
20
15
10
3
4
5
6
7
8
cylinders
1
0


SVM classification plot
mpg
45
40
35
30
25
20
15
10
100
150
200
250
300
350
400
450
1
0

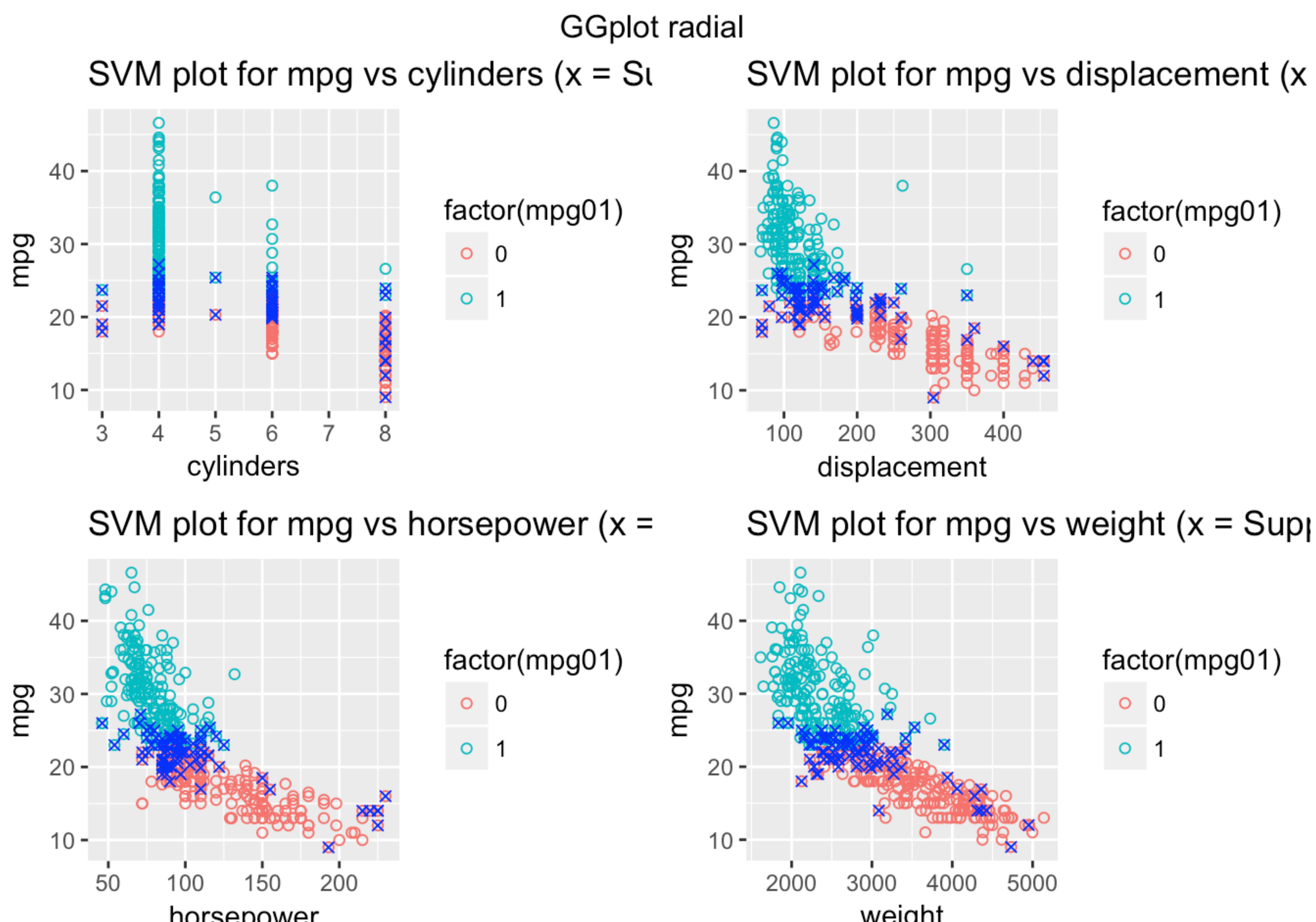
displacement

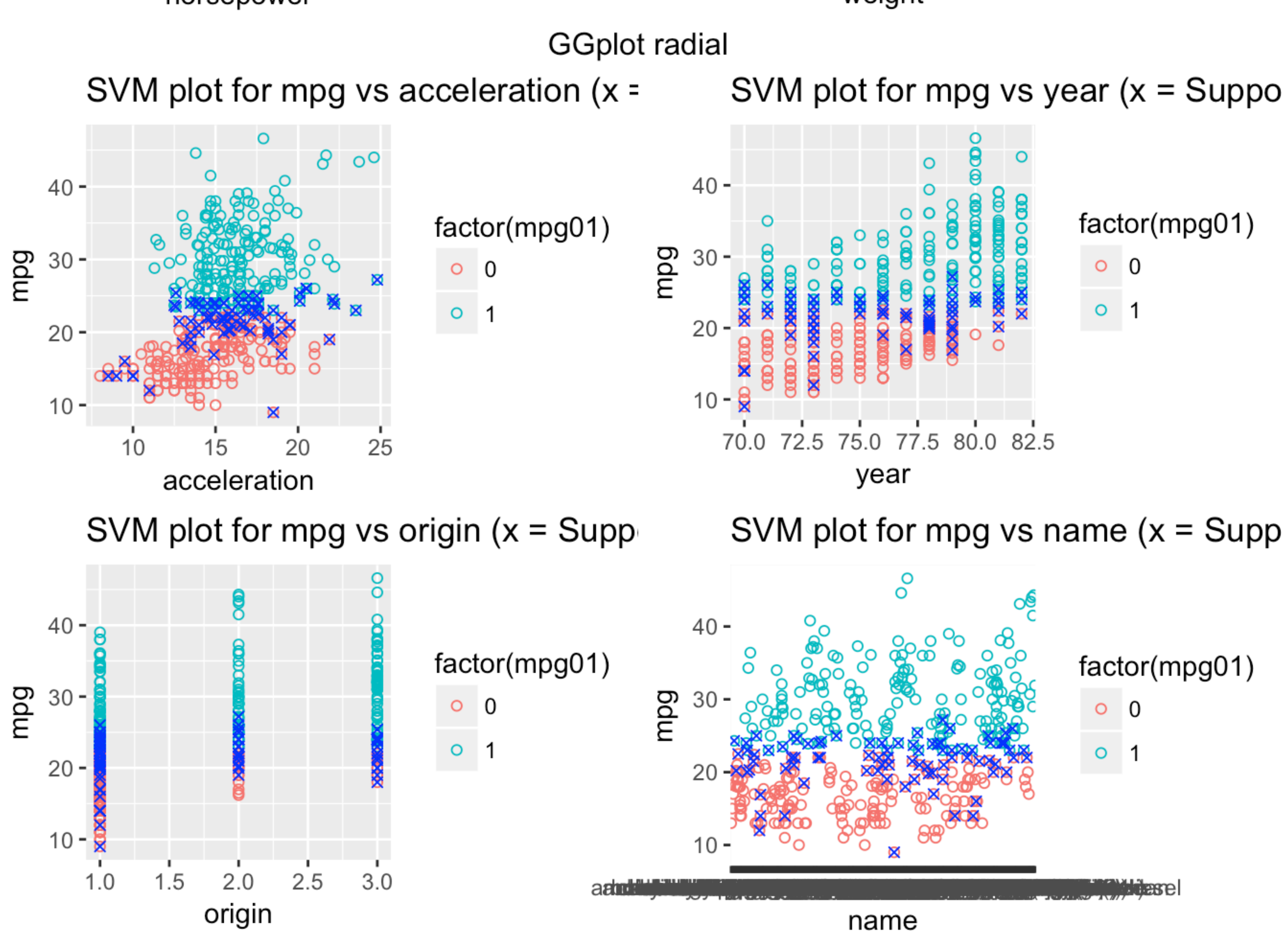






ggplot radial





3. Question 9.7.8 pg 371 (8.) This problem involves the OJ data set which is part of the ISLR package. **(a)** Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

Purchase	WeekofPurchase	StoreID	PriceCH	PriceMM	DiscCH	DiscMM	SpecialCH	SpecialMM	LoyalCH	SalePriceMM	SalePriceCH	PriceDiff	Sto
CH	237	1	1.75	1.99	0.00	0.0	0	0	0.500000	1.99	1.75	0.24	No
CH	239	1	1.75	1.99	0.00	0.3	0	1	0.600000	1.69	1.75	-0.06	No
CH	245	1	1.86	2.09	0.17	0.0	0	0	0.680000	2.09	1.69	0.40	No
MM	227	1	1.69	1.69	0.00	0.0	0	0	0.400000	1.69	1.69	0.00	No
CH	228	7	1.69	1.69	0.00	0.0	0	0	0.956535	1.69	1.69	0.00	Yes
CH	230	7	1.69	1.99	0.00	0.0	0	1	0.965228	1.99	1.69	0.30	Yes

(b)

Fit a support vector classifier to the training data using cost=0.01, with Purchase as the response and the other variables as predictors. Use the summary() function to produce summary statistics, and describe the results obtained.

```
##
## Call:
## svm(formula = Purchase ~ ., data = OJ.train, kernel = "linear",
##     cost = 0.01)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.01
##       gamma:  0.05555556
##
## Number of Support Vectors:  446
##
##  ( 224 222 )
##
##
## Number of Classes:  2
##
## Levels:
##  CH MM
```

Discussion SVM uses the classfiction method since the response variable is factor.Kernel used was linear with cost of constraint violation 0.01 and gama=0.05555556.The numbers of support vectors are 446 out of 800 of which 224 belong to CH and 222 belong to MM.

(c)

What are the training and test error rates?

```
##     train.pred
##       CH  MM
##   CH 428  56
##   MM  79 237
```

Training error rate

```
## [1] 0.16875
```

Test error rate

```
##     test.pred
##       CH  MM
##   CH 146  23
##   MM  19  82
```

```
## [1] 0.1555556
```

(d)

Use the tune() function to select an optimal cost. Consider values in the range 0.01 to 10.

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##       cost
##  0.5623413
##
## - best performance: 0.16625
##
## - Detailed performance results:
##           cost   error dispersion
## 1   0.01000000 0.17375 0.03793727
## 2   0.01778279 0.17750 0.04199868
## 3   0.03162278 0.17750 0.04199868
## 4   0.05623413 0.17125 0.05036326
## 5   0.10000000 0.16750 0.05277047
## 6   0.17782794 0.16875 0.05472469
## 7   0.31622777 0.17000 0.05407043
## 8   0.56234133 0.16625 0.05434266
## 9   1.00000000 0.16750 0.05658082
## 10  1.77827941 0.17375 0.05382908
## 11  3.16227766 0.17250 0.05027701
## 12  5.62341325 0.17500 0.04750731
## 13 10.00000000 0.17375 0.04980866
```

Discussion Optimal cost of constraint violation is 0.5623413 with 10 fold cross validation. Best performance error rate =0.16625

(e)

Compute the training and test error rates using this new value for cost.

Traing error rate

```
##     train.pred
##       CH  MM
##   CH 422  62
##   MM  68 248
```

```
## [1] 0.1625
```

Test error rate

```
##     test.pred
##       CH  MM
##   CH 146  23
##   MM  18  83
```

```
## [1] 0.1518519
```

Discussion: The training error and test error slightly decresesto by using best cost.

(f)

Repeat parts (b) through (e) using a support vector machine with a radial kernel. Use the default value for gamma.

Fit the support vector classifier with cost=0.01

```
##
## Call:
## svm(formula = Purchase ~ ., data = OJ.train, kernel = "radial",
##     cost = 0.01)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  0.01
##       gamma:  0.05555556
##
## Number of Support Vectors:  634
##
##  ( 318 316 )
##
##
## Number of Classes:  2
##
## Levels:
##  CH MM
```

Discussion SVM uses the classfiction method since the response variable is factor.Kernel used was radial with cost of constraint violation 0.01 and gama=0.05555556.The numbers of support vectors are 634 out of 800 of which 318 belong to CH and 316 belong to MM.

Traning error for SVM

```
##     train.pred
##       CH  MM
##   CH 484   0
##   MM 316   0
```

```
## [1] 0.395
```

Test error for SVM

```
##     test.pred
##       CH  MM
##   CH 169   0
##   MM 101   0
```

```
## [1] 0.3740741
```

Discussion With cost=0.01 svm classifier did not peform well the error rate for both training and test data are very high.Actually support vector classifier with radial kernel was able to classify only CH.It did not classified even single MM purchase.

Using tune() to find optimal value of cost

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##      cost
##  3.162278
##
## - best performance: 0.18
##
## - Detailed performance results:
##           cost   error dispersion
## 1   0.01000000 0.39500 0.04721405
## 2   0.01778279 0.39500 0.04721405
## 3   0.03162278 0.34000 0.05827378
## 4   0.05623413 0.20625 0.07101692
## 5   0.10000000 0.19500 0.06379220
## 6   0.17782794 0.19125 0.06292908
## 7   0.31622777 0.18750 0.05833333
## 8   0.56234133 0.18125 0.05870418
## 9   1.00000000 0.18250 0.05809475
## 10  1.77827941 0.18500 0.05096295
## 11  3.16227766 0.18000 0.04216370
## 12  5.62341325 0.18250 0.04571956
## 13 10.00000000 0.19125 0.04528076
```

Discussion optimal value of cost=3.162278

Train and test error with optimal value of cost paramater

```
##     train.pred
##       CH  MM
##   CH 442  42
##   MM  73 243
```

Train Error of model at optimal cost

```
## [1] 0.14375
```

Test Error of model at optimal cost

```
##     test.pred
##       CH  MM
##   CH 150  19
##   MM  21  80
```

```
## [1] 0.1481481
```

Discussion

By tuning the optimal cost and using it to fit model the error rate for both training and test data set decreased. And this model peromed better than previous one.

(g) Repeat parts (b) through (e) using a support vector machine with a polynomial kernel. Set degree=2.

```
##
## Call:
## svm(formula = Purchase ~ ., data = OJ.train, kernel = "poly",
##     degree = 2, cost = 0.01)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  0.01
##      degree:  2
##       gamma:  0.05555556
##      coef.0:  0
##
## Number of Support Vectors:  635
##
##  ( 319 316 )
##
##
## Number of Classes:  2
##
## Levels:
##  CH MM
```

Discussion SVM uses the classfiction method since the response variable is factor.Kernel used was polynomial with cost of constraint violation 0.01 and gama=0.05555556.The numbers of support vectors are 635 out of 800 of which 319 belong to CH and 316 belong to MM

Train Error

```
##     train.pred
##       CH  MM
##   CH 484   0
##   MM 315   1
```

```
## [1] 0.39375
```

Test Error of model

```
##     test.pred
##       CH  MM
##   CH 169   0
##   MM 101   0
```

```
## [1] 0.3740741
```

Discussion With cost=0.01 svm classifier did not peform well the error rate for both training and test data are very high.Actually support vector classifier with poly kernel was only able to classify CH.It did not classified even single MM purchase.

Tuing optimal cost

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##    10
##
## - best performance: 0.1875
##
## - Detailed performance results:
##           cost   error dispersion
## 1   0.01000000 0.39375 0.04832256
## 2   0.01778279 0.37500 0.04930066
## 3   0.03162278 0.36125 0.05756940
## 4   0.05623413 0.34000 0.05767485
## 5   0.10000000 0.32250 0.05797509
## 6   0.17782794 0.25875 0.04041881
## 7   0.31622777 0.22000 0.05658082
## 8   0.56234133 0.22250 0.06395528
## 9   1.00000000 0.21375 0.06050999
## 10  1.77827941 0.19750 0.06032320
## 11  3.16227766 0.19125 0.05104804
## 12  5.62341325 0.19125 0.05138701
## 13 10.00000000 0.18750 0.04249183
```

Discussion optimal cost= 10

```
##     train.pred
##       CH  MM
##   CH 438  46
##   MM  81 235
```

Train Error at optimal cost= 10

```
## [1] 0.15875
```

Test Error at optimal cost= 10

```
##     test.pred
##       CH  MM
##   CH 147  22
##   MM  23  78
```

```
## [1] 0.1666667
```

Discussion: Tuning reduces the training error to 15.87% and test error to 16.66% which is worse than radial kernel but slightly better than linear kernel.

(h) Overall, which approach seems to give the best results on this data?

Ans The linear model peroform beter than other with cost=0.01.But after tuing the best parameter for the cost the model with radial classification produces the best results both for training and test data.Table for errors is given below.

Errors Table

Kernel	Train.error	Test.error	Train.errorCV	Test.errorCV
Linear	0.16875	0.155556	0.16250	0.151852
Radial	0.39500	0.374074	0.14375	0.148148
Polynomial	0.39375	0.374074	0.15875	0.166667

4.

In the past couple of homework assignments you have used different classification methods to analyze the dataset you chose. For this homework, use a support vector machine to model your data. Find the test error using any/all of methods (VSA, K-fold CV). Compare the results you obtained with the result from previous homework. Did the results improve? (Use the table with the previous results to compare)

Ans

SVM produced sligltly small VSA error than random forest but random forest gave less error in 5-fold cross validation.

Error Table

Method	VSA	LOOCV	FOLD.5CV
Logistic Reg	0.1793	0.1873464	0.1219

knn	0.2428	0.3593366	0.2909
LDA	0.1770	0.2168305	0.2195
QDA	0.2155	0.213145	0.2097
MclustDA	0.2118	NA	0.2308
MclustDA (EDDA)	0.2067	0.2067314	0.2028
SVM	0.1715	NA	0.1737
Tree	0.1789	NA	0.1789
Baging	0.1789	NA	0.1854
RandomForest	0.1745	NA	0.1705
Boosting	0.1789	NA	0.1717

SVM for adult data

	... <int>	workclass <fctr>	education <fctr>	educationnum <int>	maritalstatus <fctr>	occupation <fctr>	relationship <fctr>	race <fctr>	sex <fctr>
1	39	gov	graduate	13	notmarried	clerical	outofamily	White	Male
2	50	selfemp	graduate	13	married	highskillabr	husband	White	Male
3	38	private	secndrysch	9	divorce	lowskillabr	outofamily	White	Male
4	53	private	secndrysch	7	married	lowskillabr	husband	Black	Male
5	28	private	graduate	13	married	lowskillabr	wife	Black	Female
6	37	private	master	14	married	highskillabr	wife	White	Female

6 rows | 1-10 of 13 columns

split 75%

VSA

```
##
## Call:
## svm(formula = incomelevel ~ ., data = train, kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.02857143
##
## Number of Support Vectors:  8871
##
##  ( 4462 4409 )
##
##
## Number of Classes:  2
##
## Levels:
##  <=50K >50K
```

```
##        test.pred.a
##         <=50K >50K
##   <=50K  5289  384
##   >50K    909  959
```

```
## [1] 0.1714627
```

kfold

Error estimation of ‘svm’ using 5-fold cross validation: 0.1736882