MSiA 400 Lab 3 Harish Chockalingam

```
#Problem 1
#a
#Markov Chain Monte Carlo is a method for sampling probablity
#distribution using Markov chain. By repeatedly applying a Markov chain,
#with the random sampling of Monte Carlo method, to the sample
#the sample distribution will be more closer to the stationary or actual
#population distrubtion

#b
#The main difference is that the Metropolis algorithm requires the
#probablity density distribution to select the next sample value to
#be symmetric while the Metropolis-Hasting algorithm the probablity
#density distribution can be asymmetric

#c
#Ridge Regresion is a method to optimize the predictors' coefficients so that
#multicolinearity can be prevented and avoids for over fitting of the model by
#penalizing some of the coefficients so they shirnk to zero

#Lasso Regression is similar to Ridge Regression however it is more conservative
#in that some of the coefficients shrink exactly zero

#d
#IIA or indenpendce of irrelevant alternatives is the probablity of choosing
#one of two alternatives is independent when another alternative is introduced
#i.e if X is prefered to the alternative Y in the set {X,Y} then introducing
#the alternative Z to the set {X,Y,Z} will not change preference to X
```

```
#Problem 2
#a
library(quantreg)
```

```
## Loading required package: SparseM
```

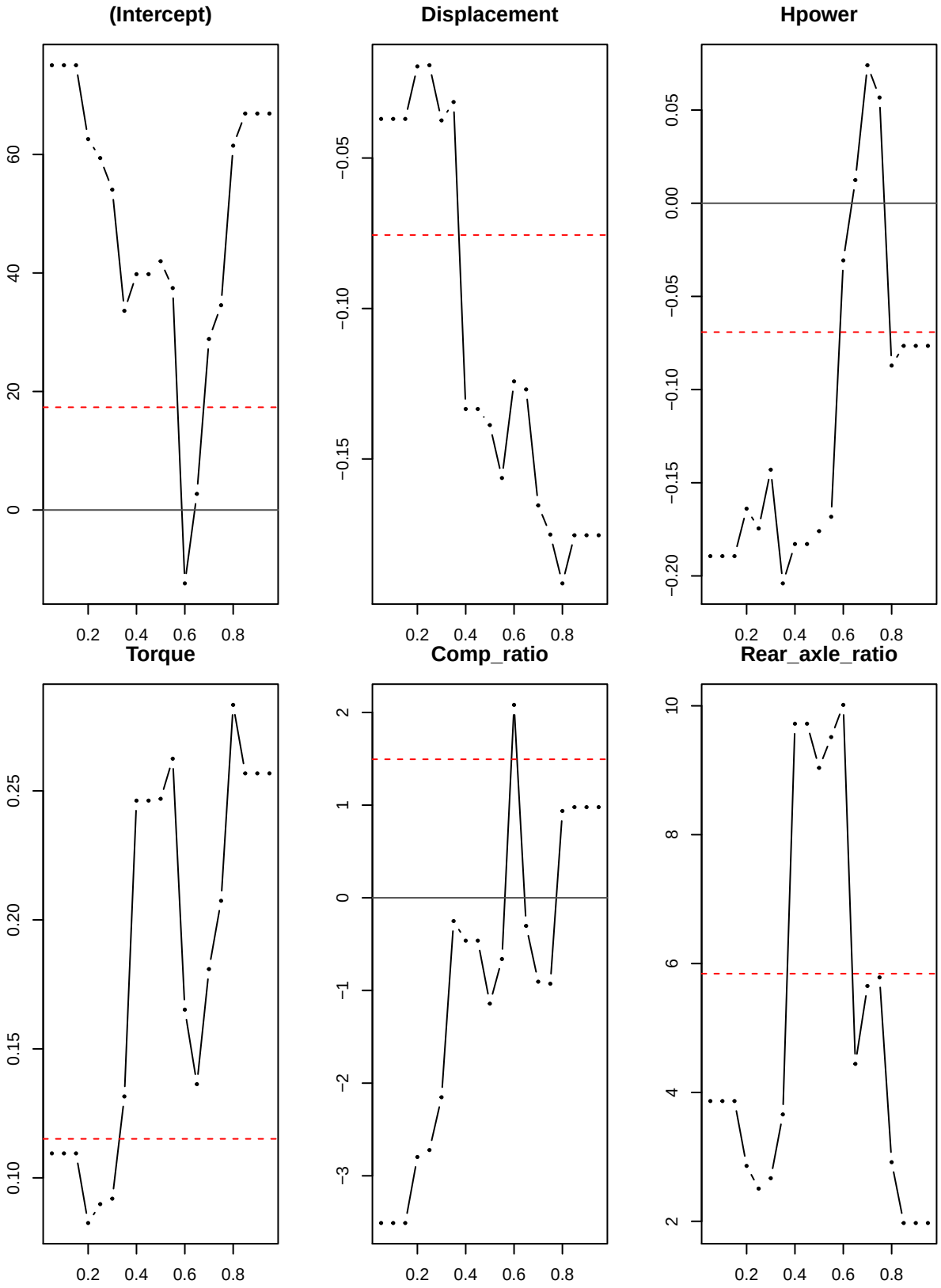
```
##
## Attaching package: 'SparseM'
```

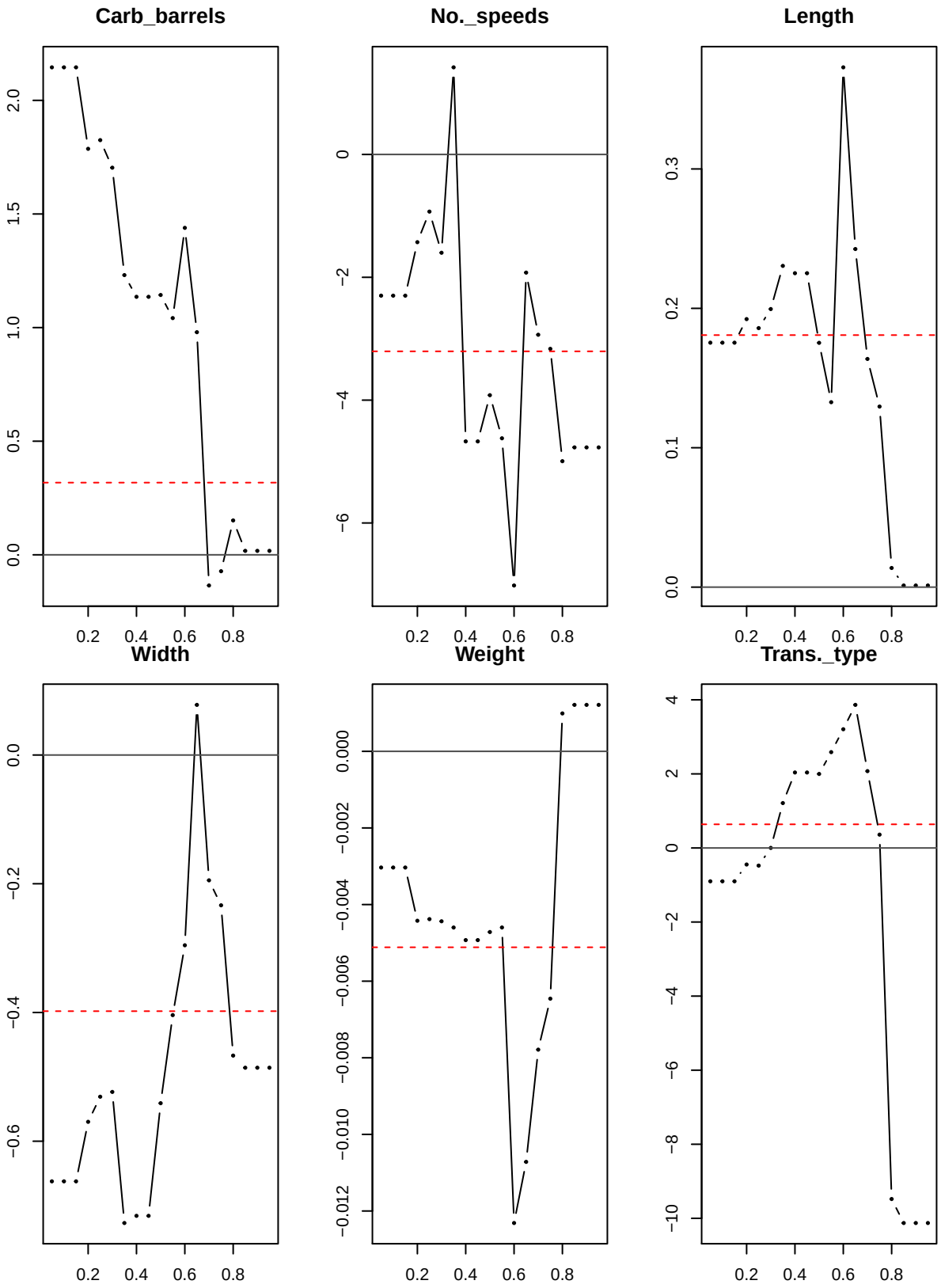
```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
data=read.csv('gas_mileage.csv')
quants<-seq(0.05,0.95,0.05)
fit1<-rq(Mpg~.,data=data,tau=quants) #Regressing all predictors on MPG
```

```
## Warning in rq.fit.br(x, y, tau = tau, ...): Solution may be nonunique
```

```
#b
plot(fit1,mfrow = c(1,3)) #showing plots of fitted model 2 at a time
```


(Intercept)
Displacement
Hpower
Torque
Comp_ratio
Rear_axle_ratio




```
#c
#From the plots the, looking at predictors Displacement, Horsepower,
#and torque. The increasing quantiles of displacement the coeffficient for
```

```
#deteremining MPG shows a declining trend while for increasing quantiles
#of horsepower the coefficient for determining MPG shows an increasing
#trend but MPG would decline after the 0.7 quantile of horsepower.
#Similary the increasing quantiles of torque the coefficient for determining
#MPG increases but makes a dip at the 0.6 quantile of torque and then MPG
#continues to increase.
```

```
#d
fit3<-rq(Mpg~.,data=data, tau=0.5)
summary(fit3, se='boot')
```

```
##
## Call: rq(formula = Mpg ~ ., tau = 0.5, data = data)
##
## tau: [1] 0.5
##
## Coefficients:
##                 Value    Std. Error t value  Pr(>|t|)
## (Intercept)     41.98707 54.37187    0.77222  0.45000
## Displacement    -0.13873  0.10214   -1.35818  0.19119
## Hpower          -0.17596  0.21685   -0.81140  0.42773
## Torque           0.24692  0.16416    1.50419  0.14988
## Comp_ratio      -1.14223  5.35643   -0.21324  0.83353
## Rear_axle_ratio  9.03682  7.04579    1.28259  0.21591
## Carb_barrels     1.14349  2.40574    0.47532  0.64028
## No._speeds      -3.91968  8.27544   -0.47365  0.64144
## Length           0.17526  0.29525    0.59361  0.56016
## Width           -0.54095  0.72532   -0.74580  0.46542
## Weight          -0.00472  0.00990   -0.47670  0.63931
## Trans._type      1.99845  7.78781    0.25661  0.80039
```

```
#By bootstraping we can determine the estimators of a given population
#but using a signifacance level of 0.1, the summary shows
#a high P-value seemingly at the 0.5 quantile the predictors are insignificant
```

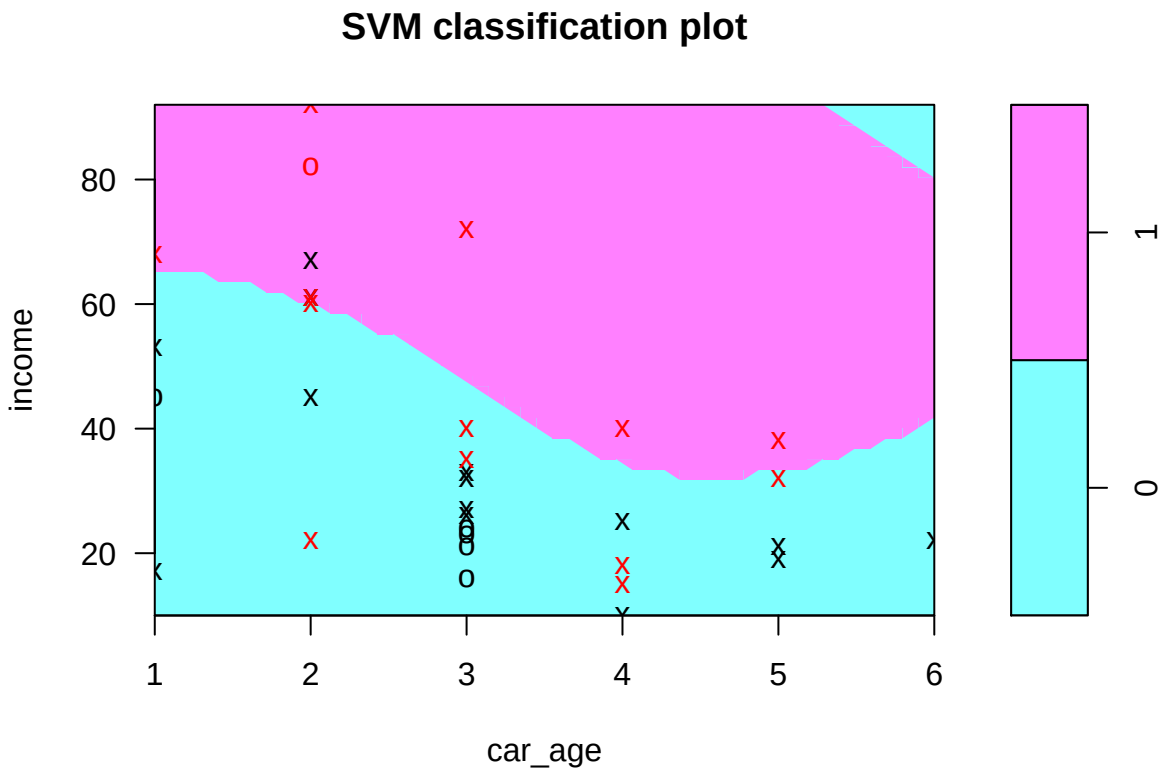
```
#3
#a
library(e1071)
data2=read.csv('car.csv')
svm=svm(factor(y)~.,data=data2)
summary(svm)
```

```
##
## Call:
## svm(formula = factor(y) ~ ., data = data2)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.5
##
## Number of Support Vectors:  27
##
```

```
##  ( 14 13 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
#The this model has a radial plane for classification
#I think the classification would perform better
#if the kernal was tuned to be linear
```

```
#b
plot(svm, data=data2)
```




```
#This SVM model doesn't seem to classify the observations well.
#There are some observations that are misclassified like at car_age=2
#and income =20 actually bought a new car. Only one observation
#at car_age=2 and income=65 is classified correctly as 1
```

```
#c
a=data.frame(income=50,car_age=5)
a$predict=predict(svm,a,type='response')
a
```

```
##   income car_age predict
## 1     50       5       1
```

```
#For income=50 and car_age=5 the prediction
#is 1 or to purchase new car
```