Lab Assignment 3

Kejin Qian

11/30/2018

Problem 1 Short answers: Please answer the following questions in a few sentences. Based on our labs,

a. What is the purpose of doing Markov Chain Monte Carlo (MCMC)?

We know that one big problem in Monte Carlo methods is in obtaining sample points from complex distributions p(x). MCMC can solve this problem and accomplish the goal of sampling points from distribution p(.) that is difficult to sample from directly by constructing a Markov Chain whose stationary distribution is exactly equal to our targetdistribution p(.).

---

b. What is the difference between the Metropolis Algorithm and the Metropolis Hastings Algorithm?

Metropolis Algorithm is a special case of Metropolis Hastings Algorithm. Metropolis Algorithm requires the proposal distribution q(.|.) to be symmetric where q($\theta_1|\theta_2$) = q($\theta_2|\theta_1$). Metropolis Hastings Algorithm is a generalization of the Metropolis Algorithm, and for MHA, q(.|.) does not have to be symmetric.

---

c. What is the purpose of Ridge regression? What is the purpose of LASSO regression?

The general OLS sometimes will have problems like too many predictors (fitting full model without penalization will result in large prediction intervals, and LS regression estimator may not uniquely exist.), multicollinearity(LS estimates are unbiased, but their variances are large so they may be far from the true value.) and ill-conditioned $(X'X)^{-1}$(LS estimates depend upon $(X'X)^{-1}$, we would have problems in computing beta_LS if X'X were singular or nearly singular).

We can use lasso and ridge regressions to solve the above problems.

1. Lasso:

Minimizes SSE subject to λ* $\sum|\beta_j|$ <= s

Lasso regression performs both variable selection and regularization. It is able to solve the above problems by forcing the sum of the absolute value of the regression coefficients to be less than a fixed value, which forces certain coefficients to be exactly set to zero, effectively choosing a simpler model that does not include those coefficients.

2. Ridge

Minimizes SSE subject to λ * $\sum\beta_j^2$ <= s

Ridge regression performs regularization. As λ increases, the sum of the squares of the coefficients is forced to be less than a fixed value, but this only shrinks the size of the coefficients towards zero. Ridge regression does not set any of the coefficients to exactly zero, so different from Lasso, Ridge regression can not do variable selection.

---

d. State the IIA assumption for Multinomial Logit discrete choice model.

Independence of irrelevant alternatives (IIA): the ratio of the probabilities of choosing two alternatives is independent of the presence or attributes of any other alternative.

Problem 2

The gas_mileage.csv data set contains the response Mpg (Miles per gallon) and 11 predictors such as Displacement, Horsepower, Torque, etc. for 32 cars.

Load Data

```
gas_mileage <- read.csv("gas_mileage.csv")
```

a. Fit quantile regression models for the 0.05, 0.10, 0.15, . . . , 0.90, 0.95th conditional quantiles for Mpg regressed on all the predictors using the quantreg package in R.

```
library(quantreg)
```

```
## Loading required package: SparseM

##
## Attaching package: 'SparseM'

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

```
seq = seq(0.05, 0.95, 0.05)
rfit <- rq(Mpg ~ ., tau = seq, data = gas_mileage)
```

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

```
summary(rfit)
```

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

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

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

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

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

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

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

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

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

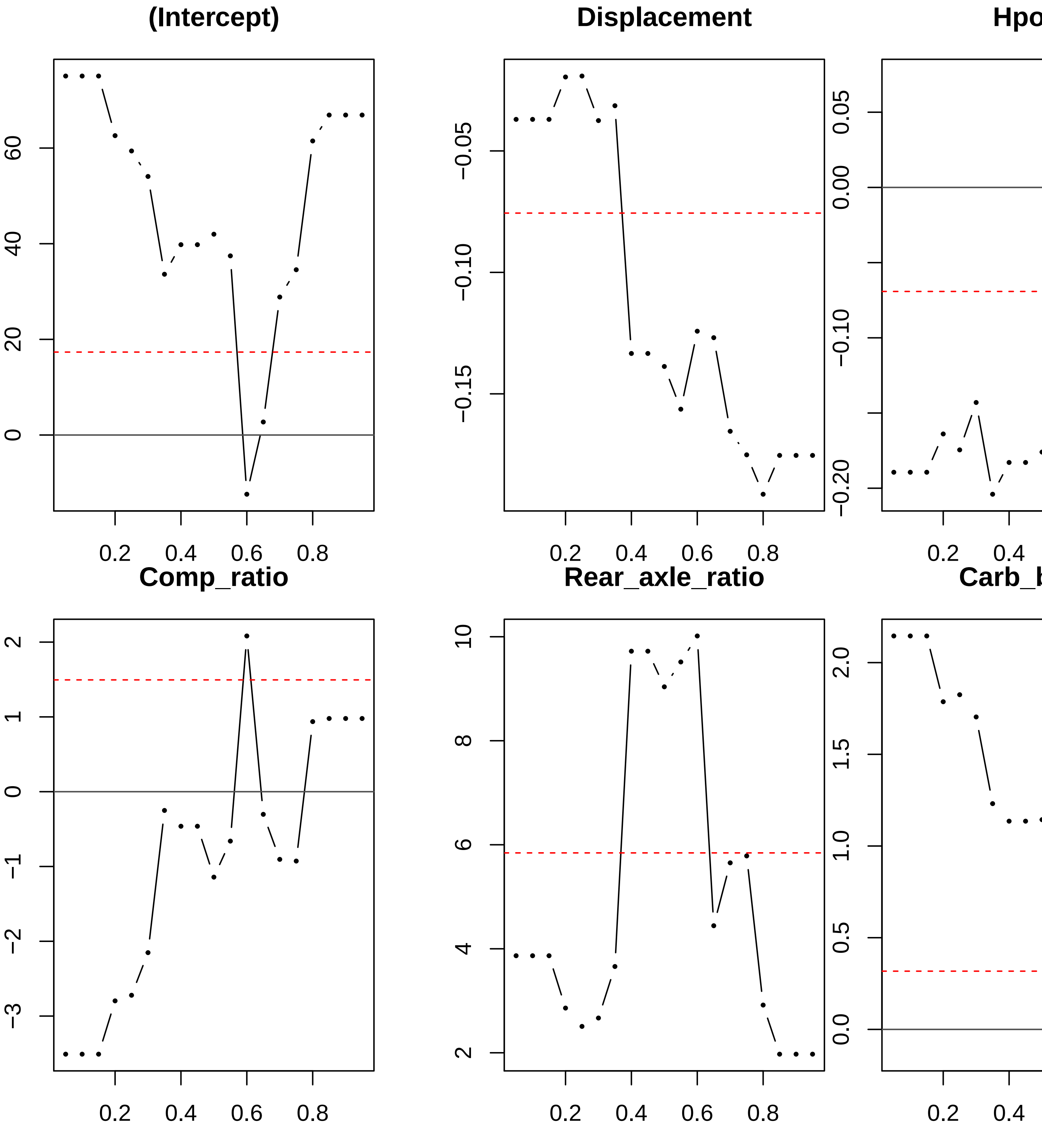
```
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.05
##
## Coefficients:
##                 coefficients   lower bd       upper bd
## (Intercept)       7.505845e+01 -1.797693e+308  1.797693e+308
## Displacement     -3.701000e-02 -1.797693e+308  1.797693e+308
## Hpower           -1.893800e-01 -1.797693e+308  1.797693e+308
## Torque            1.094900e-01 -1.797693e+308  1.797693e+308
## Comp_ratio       -3.509360e+00 -1.797693e+308  1.797693e+308
## Rear_axle_ratio   3.866260e+00 -1.797693e+308  1.797693e+308
## Carb_barrels      2.145330e+00 -1.797693e+308  1.797693e+308
## No._speeds       -2.299040e+00 -1.797693e+308  1.797693e+308
## Length            1.753600e-01 -1.797693e+308  1.797693e+308
## Width            -6.623400e-01 -1.797693e+308  1.797693e+308
## Weight           -3.030000e-03 -1.797693e+308  1.797693e+308
## Trans._type      -9.004500e-01  -1.792682e+01  1.797693e+308
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.1
##
## Coefficients:
##                 coefficients   lower bd       upper bd
## (Intercept)       7.505845e+01  -2.640074e+02   1.965771e+02
## Displacement     -3.701000e-02  -3.574400e-01   6.540000e-02
## Hpower           -1.893800e-01  -7.592400e-01   1.053380e+00
## Torque            1.094900e-01  -3.856000e-01   8.116000e-01
## Comp_ratio       -3.509360e+00  -1.141334e+01   7.802265e+01
## Rear_axle_ratio   3.866260e+00  -1.949856e+01   3.144942e+01
## Carb_barrels      2.145330e+00  -1.083878e+01   1.214711e+01
## No._speeds       -2.299040e+00  -9.998130e+00   1.812914e+01
## Length            1.753600e-01  -2.232600e-01  1.797693e+308
## Width            -6.623400e-01 -1.797693e+308   1.918620e+00
## Weight           -3.030000e-03  -1.060100e-01   1.284000e-02
## Trans._type      -9.004500e-01  -1.561480e+00  1.797693e+308
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.15
##
## Coefficients:
##                 coefficients   lower bd       upper bd
## (Intercept)       7.505845e+01  -9.002075e+01   1.453873e+02
## Displacement     -3.701000e-02  -2.327100e-01   2.910000e-02
## Hpower           -1.893800e-01  -6.259600e-01   6.757800e-01
## Torque            1.094900e-01  -2.939300e-01   5.021700e-01
## Comp_ratio       -3.509360e+00  -6.623030e+00   2.989379e+01
## Rear_axle_ratio   3.866260e+00  -1.374687e+01   1.842395e+01
## Carb_barrels      2.145330e+00  -3.081880e+00   6.189830e+00
## No._speeds       -2.299040e+00  -9.698530e+00   1.010556e+01
## Length            1.753600e-01  -8.571000e-02   2.162340e+00
## Width            -6.623400e-01  -3.833210e+00   4.010500e-01
## Weight           -3.030000e-03  -1.328000e-02   1.131000e-02
## Trans._type      -9.004500e-01  -1.446450e+00  1.797693e+308
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.2
##
## Coefficients:
##                 coefficients   lower bd       upper bd
## (Intercept)       6.259344e+01  -8.228754e+01   1.409044e+02
## Displacement     -1.956000e-02  -2.040000e-01   3.166000e-02
## Hpower           -1.639200e-01  -6.078400e-01   4.992700e-01
## Torque            8.250000e-02  -3.315400e-01   4.444400e-01
## Comp_ratio       -2.796880e+00  -6.437820e+00   1.030132e+01
## Rear_axle_ratio   2.859870e+00  -4.345210e+00   1.796188e+01
## Carb_barrels      1.786780e+00  -1.398360e+00   3.303940e+00
## No._speeds       -1.428330e+00  -9.994610e+00   1.355025e+01
## Length            1.922900e-01  -1.138700e-01   1.237590e+00
## Width            -5.698600e-01  -3.078290e+00   5.256000e-02
## Weight           -4.420000e-03  -1.309000e-02   1.036000e-02
## Trans._type      -4.470000e-01  -7.606060e+00  1.797693e+308
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.25
##
## Coefficients:
##                 coefficients   lower bd       upper bd
## (Intercept)       5.939339e+01  -8.167520e+01   1.244924e+02
## Displacement     -1.917000e-02  -2.322600e-01   2.464000e-02
## Hpower           -1.745200e-01  -5.456900e-01   3.766700e-01
## Torque            8.982000e-02  -3.224100e-01   4.848900e-01
## Comp_ratio       -2.721790e+00  -6.584030e+00   1.024147e+01
## Rear_axle_ratio   2.507430e+00  -6.154160e+00   1.816992e+01
## Carb_barrels      1.825000e+00  -1.590480e+00   3.191410e+00
## No._speeds       -9.305200e-01  -1.021943e+01   1.580215e+01
## Length            1.858100e-01  -1.563300e-01   4.075000e-01
## Width            -5.308900e-01  -2.755050e+00   2.577000e-02
## Weight           -4.380000e-03  -1.345000e-02   9.000000e-03
## Trans._type      -4.767800e-01  -7.956070e+00  1.797693e+308
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.3
##
## Coefficients:
##                 coefficients lower bd  upper bd
## (Intercept)      54.06294    -68.83438 103.95882
## Displacement     -0.03751     -0.22369   0.02329
## Hpower           -0.14300     -0.49277   0.31943
## Torque            0.09195     -0.33155   0.43812
## Comp_ratio       -2.15210     -6.28234   9.89148
## Rear_axle_ratio   2.66851     -6.44198  18.14440
## Carb_barrels      1.70373     -3.17755   3.36442
## No._speeds       -1.60050    -10.35158  14.36612
## Length            0.19950     -0.16919   0.42062
## Width            -0.52344     -1.20202   0.04226
## Weight           -0.00444     -0.00998   0.00998
## Trans._type       0.00138     -9.84964  18.44084
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.35
##
## Coefficients:
##                  coefficients lower bd  upper bd
## (Intercept)      33.61471    -64.66366 114.81804
## Displacement     -0.03139     -0.21008   0.03422
## Hpower           -0.20400     -0.44658   0.30928
## Torque            0.13156     -0.27674   0.31270
## Comp_ratio       -0.25080     -5.45183   9.81983
## Rear_axle_ratio   3.65908     -7.03406  14.90364
## Carb_barrels      1.23102     -3.39051   3.63315
## No._speeds        1.41816    -10.18349  11.84650
## Length            0.23047     -0.16893   0.42550
## Width            -0.72708     -1.12616   0.06438
## Weight           -0.00460     -0.00969   0.01709
## Trans._type       1.21189    -13.55527  19.91186
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.4
##
## Coefficients:
##                  coefficients lower bd  upper bd
## (Intercept)      39.79782    -38.02844 113.89174
## Displacement     -0.13338     -0.20434   0.03074
## Hpower           -0.18288     -0.42267   0.26439
## Torque            0.24622     -0.04369   0.30530
## Comp_ratio       -0.46214     -5.25613   8.45928
## Rear_axle_ratio   9.72169     -7.02632  13.60216
## Carb_barrels      1.13543     -2.96256   3.81884
## No._speeds       -4.67178    -10.06583  11.59511
## Length            0.22521     -0.17691   0.45815
## Width            -0.71592     -0.96215   0.04934
## Weight           -0.00493     -0.00970   0.01547
## Trans._type       2.03764    -13.21112  13.78413
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.45
##
## Coefficients:
##                  coefficients lower bd  upper bd
## (Intercept)      39.79782    -56.56228 106.18042
## Displacement     -0.13338     -0.20343   0.02052
## Hpower           -0.18288     -0.41773   0.25501
## Torque            0.24622     -0.01230   0.30261
## Comp_ratio       -0.46214     -6.14907   8.28425
## Rear_axle_ratio   9.72169     -6.94519  13.35862
## Carb_barrels      1.13543     -2.98675   4.21629
## No._speeds       -4.67178    -10.00668  11.72722
## Length            0.22521     -0.18485   0.43406
## Width            -0.71592     -1.16886   0.17787
## Weight           -0.00493     -0.00847   0.01610
## Trans._type       2.03764    -15.49451   7.66150
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.5
##
## Coefficients:
##                  coefficients lower bd  upper bd
## (Intercept)      41.98707    -50.15249  99.41846
## Displacement     -0.13873     -0.19219   0.01530
## Hpower           -0.17596     -0.39591   0.25625
## Torque            0.24692     -0.02048   0.29231
## Comp_ratio       -1.14223     -6.05074   8.13403
## Rear_axle_ratio   9.03682     -6.58867  12.87569
## Carb_barrels      1.14349     -2.74990   4.52378
## No._speeds       -3.91968     -9.28143   7.94056
## Length            0.17526     -0.17574   0.40710
## Width            -0.54095     -1.21406   0.19273
## Weight           -0.00472     -0.01453   0.01580
## Trans._type       1.99845    -16.08817  12.71580
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.55
##
## Coefficients:
##                  coefficients lower bd  upper bd
## (Intercept)      37.45543    -44.82510  83.71515
## Displacement     -0.15632     -0.18890   0.00376
## Hpower           -0.16826     -0.39300   0.25379
## Torque            0.26247     -0.01384   0.30666
## Comp_ratio       -0.66081     -6.06884   6.68266
## Rear_axle_ratio   9.51487     -6.24103  12.86802
## Carb_barrels      1.04178     -3.13414   4.18934
## No._speeds       -4.62124     -9.61926   8.96272
## Length            0.13267     -0.10225   0.52539
## Width            -0.40408     -1.49854   0.22254
## Weight           -0.00460     -0.01807   0.01441
## Trans._type       2.58728    -17.09597  11.63718
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.6
##
## Coefficients:
##                  coefficients lower bd  upper bd
## (Intercept)     -12.38280    -43.03643  95.08684
## Displacement     -0.12421     -0.41794  -0.00553
## Hpower           -0.03070     -0.35527   0.24415
## Torque            0.16519     -0.02707   0.42386
## Comp_ratio        2.08188     -5.70257   6.47639
## Rear_axle_ratio  10.01460     -6.14963  12.04353
## Carb_barrels      1.43890     -2.71410   4.09294
## No._speeds       -7.01770     -9.16567   8.71186
## Length            0.37290     -0.10354   0.51369
## Width            -0.29559     -1.54439   0.35325
## Weight           -0.01231     -0.02441   0.00933
## Trans._type       3.20547    -17.37450  10.84163
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.65
##
## Coefficients:
##                 coefficients lower bd  upper bd
## (Intercept)       2.72420    -62.53270  90.21213
## Displacement     -0.12688     -0.45468   0.03413
## Hpower            0.01245     -0.33805   0.20142
## Torque            0.13632     -0.01474   0.71181
## Comp_ratio       -0.30299     -6.43194   7.23641
## Rear_axle_ratio   4.44313     -6.87306  12.41785
## Carb_barrels      0.97970     -3.14994   4.08618
## No._speeds       -1.92379     -9.72640  11.20294
## Length            0.24256     -0.02695   0.54294
## Width             0.07790     -1.54193   0.34287
## Weight           -0.01072     -0.02450   0.00551
## Trans._type       3.86325    -17.61289   6.83024
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.7
##
## Coefficients:
##                 coefficients lower bd  upper bd
## (Intercept)      28.85096    -75.12977 102.50991
## Displacement     -0.16541     -0.47664   0.05931
## Hpower            0.07405     -0.33272   0.20573
## Torque            0.18091      0.03334   0.66419
## Comp_ratio       -0.90495     -6.34058   7.71359
## Rear_axle_ratio   5.65233     -7.01015  14.03433
## Carb_barrels     -0.13504     -2.96208   4.04653
## No._speeds       -2.93528    -10.54811  11.40447
## Length            0.16370     -0.07872   0.53613
## Width            -0.19469     -1.21537   0.36292
## Weight           -0.00779     -0.02598   0.00638
## Trans._type       2.07428    -23.65402   5.03042
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.75
```

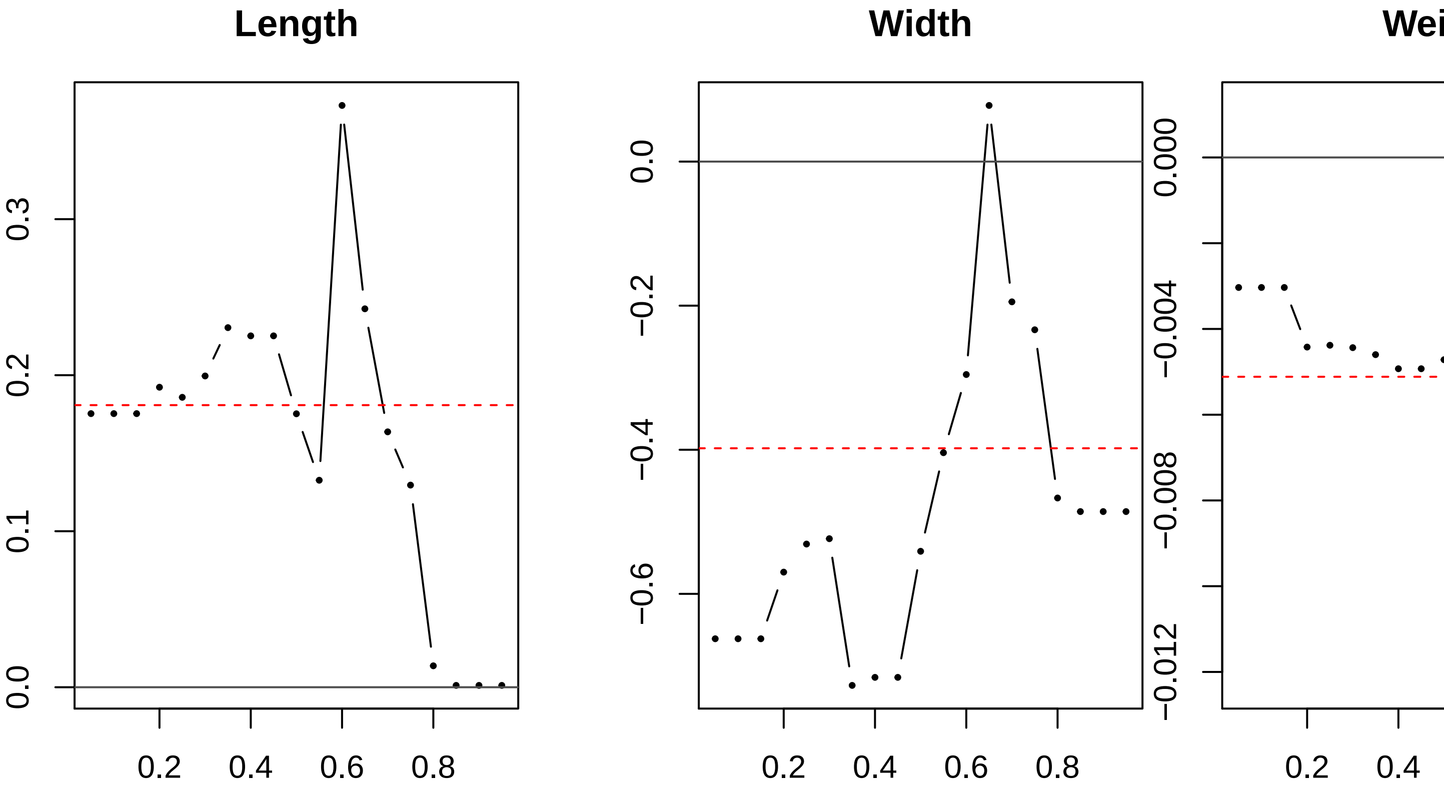
```
##
## Coefficients:
##                  coefficients   lower bd       upper bd
## (Intercept)       3.455691e+01  -8.684394e+01   1.032997e+02
## Displacement     -1.751100e-01  -4.660100e-01   6.019000e-02
## Hpower            5.674000e-02  -3.025600e-01   8.576000e-02
## Torque            2.073900e-01  -1.951000e-01   5.179700e-01
## Comp_ratio       -9.275300e-01  -7.579510e+00   9.662210e+00
## Rear_axle_ratio   5.785450e+00  -6.660930e+00   1.305027e+01
## Carb_barrels     -7.231000e-02  -3.181530e+00   4.833050e+00
## No._speeds       -3.165050e+00  -1.308105e+01   1.568430e+01
## Length            1.295500e-01  -1.320200e-01   6.347100e-01
## Width            -2.334800e-01  -1.300490e+00   3.444300e-01
## Weight           -6.460000e-03  -2.710000e-02   9.380000e-03
## Trans._type       3.597200e-01 -1.797693e+308   5.314290e+00
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.8
##
## Coefficients:
##                  coefficients   lower bd       upper bd
## (Intercept)       6.148552e+01  -1.049836e+02   8.566354e+01
## Displacement     -1.913300e-01  -4.137200e-01   6.737000e-02
## Hpower           -8.712000e-02  -2.164400e-01   7.954000e-02
## Torque            2.833300e-01  -2.153400e-01   4.907800e-01
## Comp_ratio        9.368600e-01  -7.735370e+00   9.631920e+00
## Rear_axle_ratio   2.917710e+00  -4.611710e+00   1.369960e+01
## Carb_barrels      1.512300e-01  -4.358200e+00   4.657640e+00
## No._speeds       -4.994060e+00  -1.314589e+01   1.682156e+01
## Length            1.373000e-02  -1.543800e-01   7.594600e-01
## Width            -4.669700e-01  -1.331300e+00   1.108440e+00
## Weight            9.900000e-04  -3.790000e-02   3.420000e-03
## Trans._type      -9.478690e+00 -1.797693e+308   7.201720e+00
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.85
##
## Coefficients:
##                  coefficients   lower bd       upper bd
## (Intercept)       6.690518e+01  -1.017219e+02   8.340677e+01
## Displacement     -1.753400e-01  -4.133800e-01   8.903000e-02
## Hpower           -7.653000e-02  -2.252300e-01   2.891000e-02
## Torque            2.567900e-01  -2.193400e-01   5.192900e-01
## Comp_ratio        9.785700e-01  -1.052048e+01   1.013836e+01
## Rear_axle_ratio   1.973560e+00  -4.461560e+00   1.404317e+01
## Carb_barrels      1.741000e-02  -5.369720e+00   4.663750e+00
## No._speeds       -4.769530e+00  -1.477001e+01   1.962953e+01
## Length            1.180000e-03  -2.910870e+00   7.777500e-01
## Width            -4.858100e-01  -1.369200e+00   4.014110e+00
## Weight            1.210000e-03  -4.319000e-02   3.710000e-03
## Trans._type      -1.012671e+01 -1.797693e+308   7.245470e+00
##
```

```
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.9
##
## Coefficients:
##                  coefficients   lower bd       upper bd      
## (Intercept)        6.690518e+01  -9.810508e+01   8.661455e+01
## Displacement      -1.753400e-01  -4.236900e-01   1.289300e-01
## Hpower            -7.653000e-02  -2.656700e-01   4.391000e-02
## Torque             2.567900e-01  -3.484200e-01   5.416000e-01
## Comp_ratio         9.785700e-01  -3.524620e+01   2.352705e+01
## Rear_axle_ratio    1.973560e+00  -6.904900e+00   1.521520e+01
## Carb_barrels       1.741000e-02  -9.354370e+00   4.553580e+00
## No._speeds        -4.769530e+00  -2.477762e+01   2.793282e+01
## Length             1.180000e-03 -1.797693e+308   9.343800e-01
## Width             -4.858100e-01  -5.684390e+00  1.797693e+308
## Weight             1.210000e-03  -4.721000e-02   5.040000e-03
## Trans._type       -1.012671e+01 -1.797693e+308   7.331570e+00
##
## Call: rq(formula = Mpg ~ ., tau = seq, data = gas_mileage)
##
## tau: [1] 0.95
##
## Coefficients:
##                  coefficients   lower bd       upper bd      
## (Intercept)        6.690518e+01 -1.797693e+308  1.797693e+308
## Displacement      -1.753400e-01 -1.797693e+308  1.797693e+308
## Hpower            -7.653000e-02 -1.797693e+308  1.797693e+308
## Torque             2.567900e-01 -1.797693e+308  1.797693e+308
## Comp_ratio         9.785700e-01 -1.797693e+308  1.797693e+308
## Rear_axle_ratio    1.973560e+00 -1.797693e+308  1.797693e+308
## Carb_barrels       1.741000e-02 -1.797693e+308  1.797693e+308
## No._speeds        -4.769530e+00 -1.797693e+308  1.797693e+308
## Length             1.180000e-03 -1.797693e+308  1.797693e+308
## Width             -4.858100e-01 -1.797693e+308  1.797693e+308
## Weight             1.210000e-03 -1.797693e+308  1.797693e+308
## Trans._type       -1.012671e+01 -1.797693e+308   7.544440e+00
```

b. Plot the results using the plot function.

```r
plot(rfit, mfrow=c(1,2))
```


(Intercept)
0
20
40
60
0.2
0.4
0.6
0.8
Displacement
−0.05
−0.10
−0.15
0.2
0.4
0.6
0.8
Hpo
0.05
0.00
−0.10
−0.20
0.2
0.4
Comp_ratio
−3
−2
−1
0
1
2
0.2
0.4
0.6
0.8
Rear_axle_ratio
2
4
6
8
10
0.2
0.4
0.6
0.8
Carb_b
0.0
0.5
1.0
1.5
2.0
0.2
0.4



c. Interpret the results for 3 predictors of your choice.

1. Carb_barrels: From the plot of Carb_barrels, we can see that the magnitudes of effects of Carb_barrels had a general decreasing trend over quantiles. From the plot, we can see that the OLS coefficient for Carb_barrels is around 0.3, and between 0 quantile and about 0.675 quantile of Carb_barrels, the magnitude of the positive effect of Carb_barrels at each quantile are considerably higher than the OLS coefficient. And after about 0.675 quantile, the coefficients dropped below the OLS conefficients and were fluctuating around 0. Also at 0.7 and 0.75 quantiles, the magnitudes of effects of Carb_barrels to Mpg became negative. So we can conclude that generally, the lower the Carb_barrels, the larger the positive effects to Mpg.

2. Displacement: From the plot of Displacement, we can see that at all the quantiles, the magnitudes of effects of Displacement on Mpg are negative. Also, similar to Car_barrels, we can see a general decreasing trend over quantiles. Between 0 quantile and 0.35 quantile, the magnitudes of effects of Displacement at each quantile are above the OLS coefficient(around -0.075). And start from around 0.4 quantile of Displacement, the magnitudes of the effects of Displacement started to drop below the OLS coefficients with small fluctuations. And it dropped to around -0.175 at 100% quantile.

3. Rear_axle_ratio: From the plot of Rear_axle_ratio, we can see that at all the quantiles, the magnitudes of effects of Rear_axle_ratio are positive. And the OLS coefficient of predictor Real_axle_ratio is about 5.8. Between 0 quantile and 0.35 quantile of Rear_axle_ratio, the magnitudes of effects of Rear_axle_ratio at each quantile are below the OLS coefficient. Specifically, for 0, 0.5, 0.1 quantiles, the magnitudes of effects are actually constant at 3.866260. Then it dropped a little bit at 0.2 and 0.25 quantile but increased again at 0.3 quantile. Between 0.4 and 0.6 quantile, we can see that the magnitudes of effect of Rear_axle_ratio are considerably higher compared to the OLS coefficient. But it dropped below OLS coefficient again starting from 0.65 quantile. Although the effects of Rear_axle_ratio increased a little bit in the next two quantile intervals, but they were still below the OLS coefficents. And starting from 0.8 quantile, the magnitudes of effects sharply decreased to around 2.

---

d. Report the summary for the conditional median (0.50th conditional quantile) using the bootstrap method for computing standard errors of regression coefficients.

```
rfit1 <- rq(Mpg ~ ., tau = .5, data = gas_mileage)
summary(rfit1, se = "boot")
```

```
##
## Call: rq(formula = Mpg ~ ., tau = 0.5, data = gas_mileage)
##
## tau: [1] 0.5
##
## Coefficients:
##                  Value    Std. Error t value  Pr(>|t|)
## (Intercept)     41.98707 53.31533    0.78752  0.44122
## Displacement    -0.13873  0.10845   -1.27915  0.21709
## Hpower          -0.17596  0.21830   -0.80604  0.43074
## Torque           0.24692  0.18498    1.33486  0.19856
## Comp_ratio      -1.14223  5.30295   -0.21539  0.83188
## Rear_axle_ratio  9.03682  6.87239    1.31495  0.20504
## Carb_barrels     1.14349  2.54006    0.45018  0.65795
## No._speeds      -3.91968  8.05995   -0.48631  0.63261
## Length           0.17526  0.29051    0.60330  0.55384
## Width           -0.54095  0.67582   -0.80044  0.43390
## Weight          -0.00472  0.01035   -0.45596  0.65387
## Trans._type      1.99845  9.06978    0.22034  0.82808
```

Problem 3

The car.csv data contains the response y = 0,1 (whether a family purchases a new car, yes = 1, no = 0) and 2 predictors: family income and age of car.

Load Data

```
car <- read.csv("car.csv")
```

a. Fit a support vector machine to predict the response using default setting for kernels and hyper-parameters in the svm function in e1071 package.

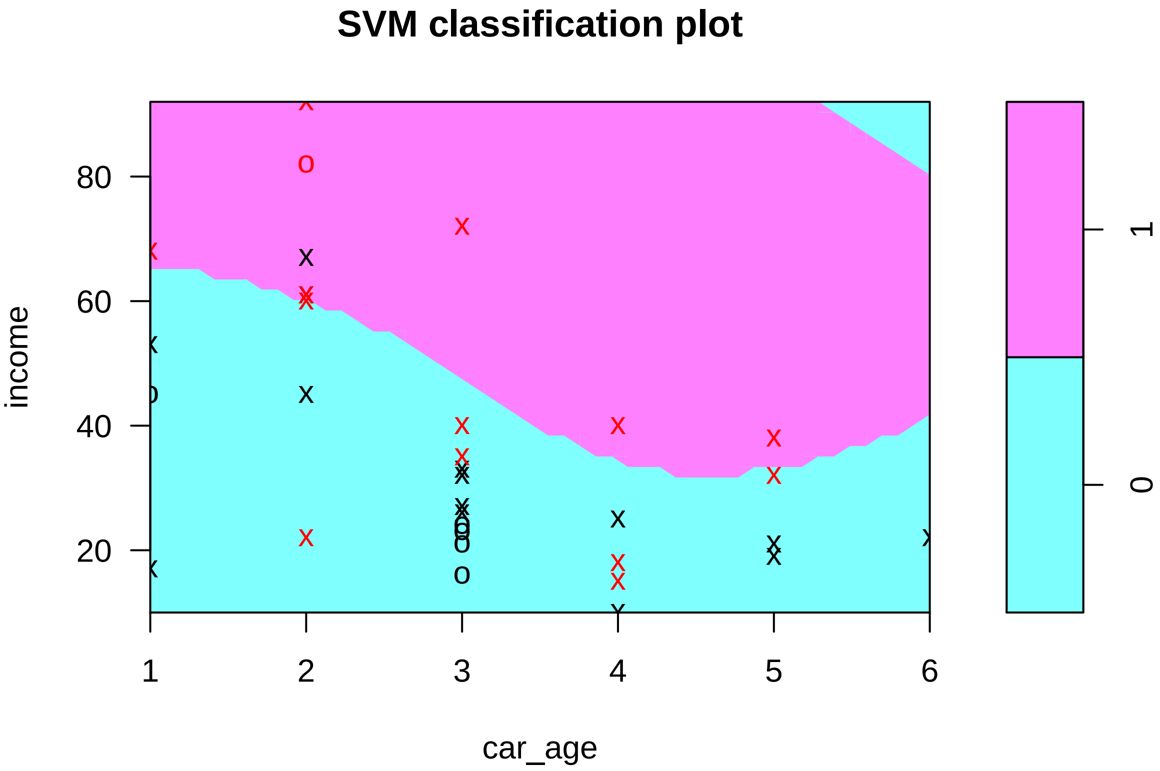
```
library(e1071)
svm = svm(factor(y) ~ ., data = car)
summary(svm)
```

```
##
## Call:
## svm(formula = factor(y) ~ ., data = car)
##
##
## 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
```

b. Plot the result using the plot function.

```
plot(svm, car, income~car_age)
```



c. Predict the response for a family with income = 50, car age = 5.

```
newdata <- data.frame(income = 50, car_age = 5)
newdata
```

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

```
newdata$pred <- predict(svm, newdata = newdata, type = "response")
newdata
```

```
##   income car_age pred
## 1     50       5    1
```

The predicted response for a family with income = 50, car age = 5 is 1(purchases a new car).