**CSC570AL Machine Learning Homework 2 (100 points)**

**Applying k-Nearest Neighbors to predict MPG**

1. Download the dataset Auto.csv.
2. Explore the overall structure of the dataset using str(). Describe it one paragraph here. Copy and paste the commands and execute results in R console here. (10 points)

> str(auto)

'data.frame': 397 obs. of 9 variables:

$ mpg : num 18 15 18 16 17 15 14 14 14 15 ...

$ cylinders : int 8 8 8 8 8 8 8 8 8 8 ...

$ displacement: num 307 350 318 304 302 429 454 440 455 390 ...

$ horsepower : Factor w/ 94 levels "?","100","102",..: 17 35 29 29 24 42 47 46 48 40 ...

$ weight : int 3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...

$ acceleration: num 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...

$ year : int 70 70 70 70 70 70 70 70 70 70 ...

$ origin : int 1 1 1 1 1 1 1 1 1 1 ...

$ name : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 2 ...

In the auto dataset there are 397 observations and 9 variables

1. Convert the attribute horsepower from character to integer. Copy and paste the commands and execute results in R console here. (5 points)

> auto$horsepower <- as.integer(auto$horsepower)

> auto$horsepower

[1] 17 35 29 29 24 42 47 46 48 40 37 34 29 48 91 91 93 81 84 50 83 86 91 10 86 46 43 45 41 84 86 91 1 2 5 2 84 2 35 38 31 29 39 37 38 8 69 2 84 82 86 67 72 62 66 57 67 91 76 55 86 82

[63] 35 38 29 31 29 44 32 34 40 93 29 17 24 29 9 72 83 66 82 88 93 76 84 38 29 26 21 29 42 29 33 29 46 48 38 5 2 2 84 91 50 29 36 37 39 2 84 69 90 86 81 6 86 26 49 52 71 87 9 29 8 14

[125] 39 91 1 2 2 64 76 62 71 2 8 5 24 29 29 24 29 79 64 74 53 58 71 71 71 93 89 64 91 5 69 69 37 26 29 27 8 5 8 91 8 8 16 71 79 2 74 92 68 93 93 67 86 91 84 94 11 54 82 77 88 75

[187] 79 24 29 13 30 2 5 77 86 53 57 67 54 2 74 8 91 68 67 71 69 3 29 84 7 13 39 26 17 29 65 76 56 92 67 26 8 26 17 8 5 2 94 39 37 40 28 74 84 71 85 60 79 64 74 93 8 8 51 63 53 67

[249] 57 8 24 23 5 91 81 84 2 86 5 81 8 13 26 35 23 24 65 91 93 71 91 5 81 93 4 15 11 19 68 65 11 81 84 86 8 17 16 22 20 32 25 15 29 68 62 76 76 73 15 68 86 67 67 62 66 86 11 11 86 72

[311] 57 67 62 86 84 86 86 74 86 71 88 71 62 5 62 51 51 64 64 64 1 64 59 18 2 84 1 69 80 80 88 8 80 56 61 57 64 62 59 65 60 62 62 70 1 71 71 2 70 76 72 12 13 8 5 84 81 84 84 84 81 80

[373] 86 88 70 65 65 60 67 84 71 67 64 64 64 8 81 88 9 92 80 86 82 53 80 75 78

1. The horsepower attribute has some missing values. Remove the observations with missing values, i.e., delete the rows with missing values from the data frame. Copy and paste the commands and execute results in R console here. (10 points)

> is.na(auto$horsepower)

[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[32] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[63] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[94] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[125] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[156] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[187] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[218] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[249] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[280] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[311] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[342] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[373] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

> auto <- na.omit(auto)

> str(auto)

'data.frame': 397 obs. of 9 variables:

$ mpg : num 18 15 18 16 17 15 14 14 14 15 ...

$ cylinders : int 8 8 8 8 8 8 8 8 8 8 ...

$ displacement: num 307 350 318 304 302 429 454 440 455 390 ...

$ horsepower : int 17 35 29 29 24 42 47 46 48 40 ...

$ weight : int 3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...

$ acceleration: num 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...

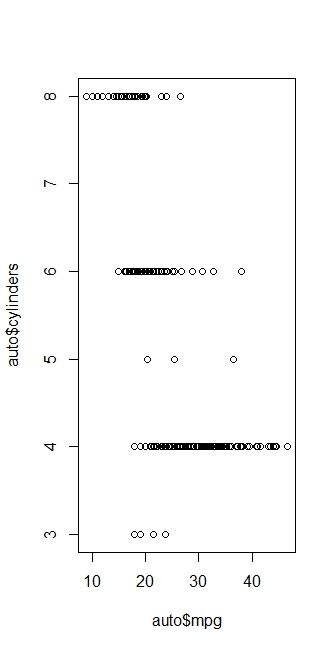
$ year : int 70 70 70 70 70 70 70 70 70 70 ...

$ origin : int 1 1 1 1 1 1 1 1 1 1 ...

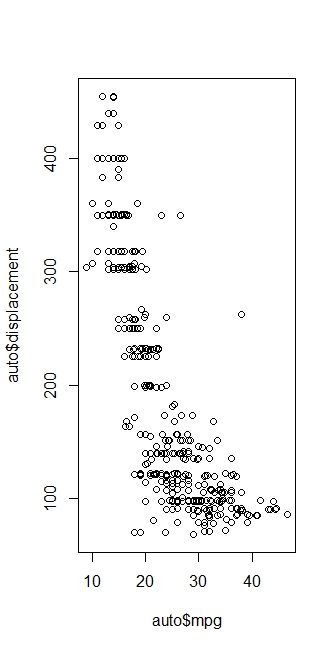
$ name : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 2 ...

1. Explore the data in order to investigate the association between mpg and the other features.  
   Which of the other features seem most likely to be useful in predicting mpg (scatterplots may  
   be useful tools to answer this question). Describe your findings using a paragraph. Copy and paste the commands and execute results in R console here. (15 points)

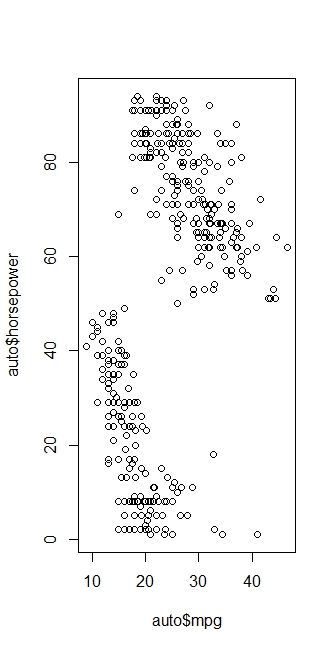
> plot(x = auto$mpg, y = auto$cylinders)



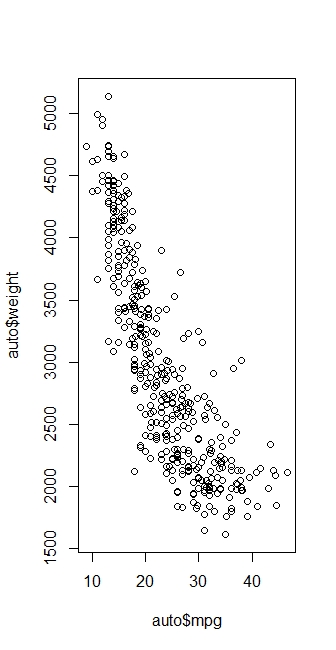
> plot(x = auto$mpg, y = auto$displacement)



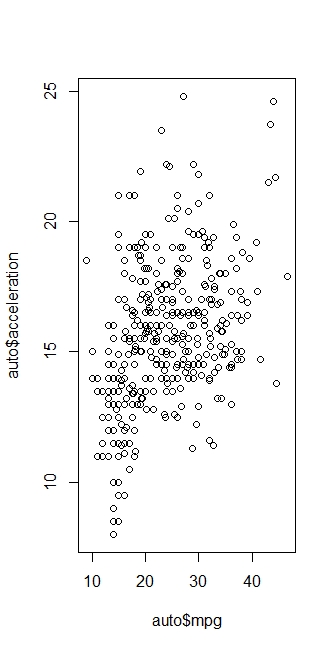
> plot(x = auto$mpg, y = auto$horsepower)



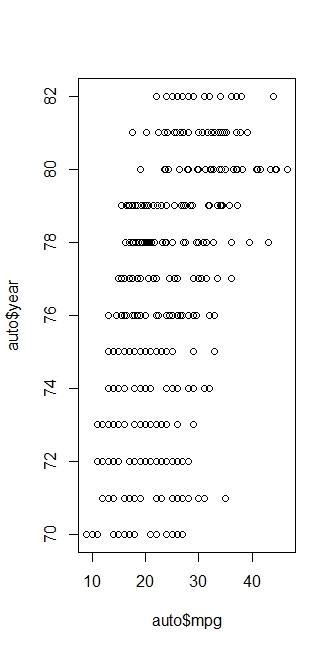
> plot(x = auto$mpg, y = auto$weight)



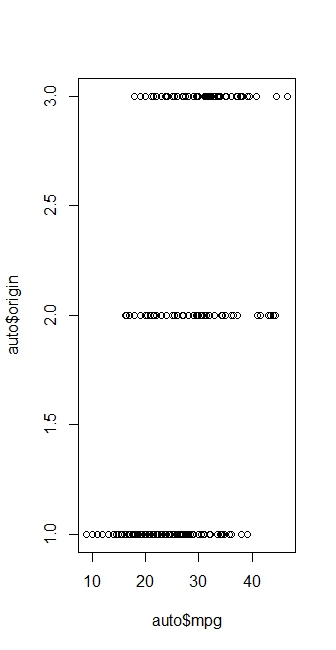
> plot(x = auto$mpg, y = auto$acceleration)

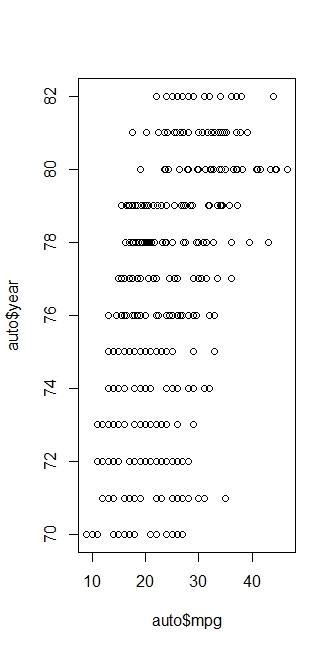


> plot(x = auto$mpg, y = auto$year)



> plot(x = auto$mpg, y = auto$origin)





1. Create a new attribute mpg1 that contains 1 if mpg is strictly greater than its median, and 0 if  
   mpg is equal or less than its median. Copy and paste the commands and execute results in R here. (10 points)

> median <- median(auto$mpg)

> median

[1] 23

> auto$mpg1 <- ifelse(auto$mpg > median,1,0)

> auto$mpg1

[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1

[85] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

[169] 0 0 0 1 1 1 0 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0

[253] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[337] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1

1. Decide which attributes you are going to use to predict mpg1. Remove all remaining  
   attributes, including mpg. Copy and paste the commands and execute results in R console here. (10 points)

> newauto <- auto[c(3,4,5,6,10)]

> str(newauto)

'data.frame': 397 obs. of 5 variables:

$ displacement: num 307 350 318 304 302 429 454 440 455 390 ...

$ horsepower : int 17 35 29 29 24 42 47 46 48 40 ...

$ weight : int 3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...

$ acceleration: num 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...

$ mpg1 : num 0 0 0 0 0 0 0 0 0 0 ...

1. Set the seed of the random number generator to a fixed integer, say 1, so that you can  
   reproduce your work:

> set.seed(1)

> set.seed(1)

1. Normalize the attribute values. Copy and paste the commands and execute results in R console here. (5 points)

> normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x)))}

> autonormdata <- as.data.frame(sapply(newauto[1:4], normalize))

displacement horsepower weight acceleration

1 0.617571059 0.17204301 0.53614970 0.23809524

2 0.728682171 0.36559140 0.58973632 0.20833333

3 0.645994832 0.30107527 0.51686986 0.17857143

4 0.609819121 0.30107527 0.51601928 0.23809524

5 0.604651163 0.24731183 0.52055571 0.14880952

6 0.932816537 0.44086022 0.77346187 0.11904762

7 0.997416021 0.49462366 0.77714772 0.05952381

8 0.961240310 0.48387097 0.76523958 0.02976190

9 1.000000000 0.50537634 0.79727814 0.11904762

10 0.832041344 0.41935484 0.63425007 0.02976190

11 0.813953488 0.38709677 0.55287780 0.11904762

12 0.702842377 0.35483871 0.56592005 0.00000000

13 0.857881137 0.30107527 0.60901616 0.08928571

14 1.000000000 0.50537634 0.41763538 0.11904762

15 0.116279070 0.96774194 0.21519705 0.41666667

16 0.335917313 0.96774194 0.34590303 0.44642857

17 0.338501292 0.98924731 0.32917494 0.44642857

18 0.341085271 0.86021505 0.27615537 0.47619048

19 0.074935401 0.89247312 0.14658350 0.38690476

20 0.074935401 0.52688172 0.06294301 0.74404762

21 0.108527132 0.88172043 0.30025517 0.56547619

22 0.100775194 0.91397849 0.23164162 0.38690476

23 0.093023256 0.96774194 0.21604763 0.56547619

24 0.136950904 0.09677419 0.17607031 0.26785714

25 0.338501292 0.91397849 0.29345052 0.41666667

26 0.754521964 0.48387097 0.85114828 0.35714286

27 0.617571059 0.45161290 0.78338531 0.41666667

28 0.645994832 0.47311828 0.78508648 0.32738095

29 0.609819121 0.43010753 0.88432095 0.62500000

30 0.074935401 0.89247312 0.14658350 0.38690476

31 0.186046512 0.91397849 0.18457613 0.44642857

32 0.116279070 0.96774194 0.17436915 0.35714286

33 0.077519380 0.00000000 0.12276722 0.65476190

34 0.423772610 0.01075269 0.28948115 0.29761905

35 0.405684755 0.04301075 0.51772044 0.44642857

36 0.470284238 0.01075269 0.48653246 0.44642857

37 0.470284238 0.89247312 0.47887723 0.44642857

38 0.423772610 0.01075269 0.47490785 0.44642857

39 0.728682171 0.36559140 0.73603629 0.23809524

40 0.857881137 0.39784946 0.80833570 0.20833333

41 0.731266150 0.32258065 0.72044230 0.32738095

42 0.645994832 0.30107527 0.70399773 0.29761905

43 0.813953488 0.40860215 0.94754749 0.20833333

44 0.857881137 0.38709677 0.88829033 0.23809524

45 0.857881137 0.39784946 1.00000000 0.23809524

46 0.490956072 0.07526882 0.38247803 0.32738095

47 0.186046512 0.73118280 0.22540403 0.65476190

48 0.470284238 0.01075269 0.47320669 0.41666667

49 0.470284238 0.89247312 0.43266232 0.38690476

50 0.139534884 0.87096774 0.17210094 0.35714286

51 0.124031008 0.91397849 0.14459881 0.35714286

52 0.028423773 0.70967742 0.13070598 0.68452381

53 0.051679587 0.76344086 0.12815424 0.38690476

54 0.007751938 0.65591398 0.04536433 0.65476190

55 0.010335917 0.69892473 0.00000000 0.59523810

56 0.074935401 0.60215054 0.06265948 0.65476190

57 0.059431525 0.70967742 0.09696626 0.74404762

58 0.116279070 0.96774194 0.18854551 0.44642857

59 0.076227390 0.80645161 0.14544939 0.53571429

60 0.074935401 0.58064516 0.18174086 0.92261905

61 0.186046512 0.91397849 0.22540403 0.68452381

62 0.139534884 0.87096774 0.17380210 0.50595238

63 0.728682171 0.36559140 0.75446555 0.23809524

64 0.857881137 0.39784946 0.78593706 0.23809524

65 0.645994832 0.30107527 0.71505529 0.32738095

66 0.731266150 0.32258065 0.71335413 0.29761905

67 0.609819121 0.30107527 0.58378225 0.20833333

68 0.932816537 0.46236559 0.85625177 0.17857143

69 0.728682171 0.33333333 0.81910972 0.32738095

70 0.728682171 0.35483871 0.80606748 0.32738095

71 0.857881137 0.41935484 0.79642756 0.26785714

72 0.005167959 0.98924731 0.20328891 0.32738095

73 0.609819121 0.30107527 0.64615821 0.26785714

74 0.617571059 0.17204301 0.70456479 0.35714286

75 0.604651163 0.24731183 0.76013609 0.47619048

76 0.645994832 0.30107527 0.69861072 0.35714286

77 0.136950904 0.08602151 0.37425574 0.38690476

78 0.136950904 0.76344086 0.25460731 0.59523810

79 0.134366925 0.88172043 0.38729799 0.68452381

80 0.072351421 0.69892473 0.16331160 0.59523810

81 0.139534884 0.87096774 0.22171817 0.47619048

82 0.074935401 0.93548387 0.19138078 0.53571429

83 0.134366925 0.98924731 0.25318968 0.38690476

84 0.077519380 0.80645161 0.15622342 0.41666667

85 0.074935401 0.89247312 0.13807769 0.50595238

86 0.728682171 0.39784946 0.70513184 0.29761905

87 0.609819121 0.30107527 0.58378225 0.20833333

88 0.728682171 0.26881720 0.67337681 0.29761905

89 0.604651163 0.21505376 0.68868727 0.38690476

90 0.645994832 0.30107527 0.61355259 0.26785714

91 0.932816537 0.44086022 0.94669691 0.20833333

92 0.857881137 0.30107527 0.80833570 0.23809524

93 0.731266150 0.34408602 0.77969946 0.29761905

94 0.645994832 0.30107527 0.74397505 0.38690476

95 0.961240310 0.48387097 0.88517153 0.17857143

96 1.000000000 0.50537634 0.94641338 0.17857143

97 0.754521964 0.39784946 0.62602779 0.17857143

98 0.405684755 0.04301075 0.42755883 0.50595238

99 0.470284238 0.01075269 0.47207258 0.59523810

100 0.423772610 0.01075269 0.37765807 0.47619048

101 0.470284238 0.89247312 0.39920612 0.50595238

102 0.335917313 0.96774194 0.36603346 0.47619048

103 0.074935401 0.52688172 0.09554862 0.77380952

104 0.857881137 0.30107527 0.95945563 0.35714286

105 0.857881137 0.37634409 0.93365466 0.26785714

106 0.754521964 0.38709677 0.86220584 0.29761905

107 0.728682171 0.40860215 0.81825914 0.26785714

108 0.423772610 0.01075269 0.33342784 0.41666667

109 0.074935401 0.89247312 0.18882903 0.65476190

110 0.186046512 0.73118280 0.22341934 0.68452381

111 0.103359173 0.95698925 0.21718174 0.50595238

112 0.005167959 0.91397849 0.14488234 0.32738095

113 0.139534884 0.86021505 0.19761837 0.62500000

114 0.224806202 0.05376344 0.24354976 0.35714286

115 0.077519380 0.91397849 0.18485965 0.44642857

116 0.728682171 0.26881720 0.70002835 0.29761905

117 0.857881137 0.51612903 0.75559966 0.08928571

118 0.000000000 0.54838710 0.07201588 0.68452381

119 0.124031008 0.75268817 0.15452226 0.44642857

120 0.118863049 0.92473118 0.27473774 0.35714286

121 0.136950904 0.08602151 0.35582648 0.44642857

122 0.645994832 0.30107527 0.50637936 0.17857143

123 0.136950904 0.07526882 0.29685285 0.35714286

124 0.227390181 0.13978495 0.33853133 0.32738095

125 0.728682171 0.40860215 0.58151403 0.17857143

126 0.335917313 0.96774194 0.42217182 0.50595238

127 0.341085271 0.00000000 0.35781117 0.53571429

128 0.423772610 0.01075269 0.36518287 0.47619048

129 0.470284238 0.01075269 0.48851715 0.53571429

130 0.028423773 0.67741935 0.09554862 0.65476190

131 0.139534884 0.80645161 0.23759569 0.50595238

132 0.007751938 0.65591398 0.06322654 0.77380952

133 0.186046512 0.75268817 0.26339665 0.53571429

134 0.470284238 0.01075269 0.61468670 0.53571429

135 0.490956072 0.07526882 0.57244117 0.59523810

136 0.405684755 0.04301075 0.56705415 0.50595238

137 0.604651163 0.24731183 0.71675645 0.35714286

138 0.728682171 0.30107527 0.87496456 0.38690476

139 0.645994832 0.30107527 0.80635101 0.32738095

140 0.604651163 0.24731183 0.85766941 0.47619048

141 0.609819121 0.30107527 0.74964559 0.44642857

142 0.077519380 0.83870968 0.17181741 0.50595238

143 0.028423773 0.67741935 0.09923448 0.44642857

144 0.074935401 0.78494624 0.19478310 0.38690476

145 0.020671835 0.55913978 0.01020697 0.50595238

146 0.038759690 0.61290323 0.11057556 0.65476190

147 0.056847545 0.75268817 0.14516586 0.38690476

148 0.056847545 0.75268817 0.14034590 0.44642857

149 0.124031008 0.75268817 0.17947264 0.35714286

150 0.134366925 0.98924731 0.24836972 0.41666667

151 0.103359173 0.94623656 0.22058407 0.44642857

152 0.028423773 0.67741935 0.10972498 0.47619048

153 0.405684755 0.96774194 0.46810320 0.47619048

154 0.470284238 0.04301075 0.52339098 0.47619048

155 0.470284238 0.73118280 0.51573575 0.77380952

156 0.470284238 0.73118280 0.43804933 0.68452381

157 0.857881137 0.38709677 0.86617522 0.20833333

158 0.728682171 0.26881720 0.80153105 0.35714286

159 0.645994832 0.30107527 0.81797562 0.38690476

160 0.731266150 0.27956989 0.86305642 0.32738095

161 0.421188630 0.07526882 0.65041111 0.77380952

162 0.470284238 0.04301075 0.64757584 0.62500000

163 0.490956072 0.07526882 0.60022682 0.65476190

164 0.405684755 0.96774194 0.61582081 0.65476190

165 0.421188630 0.07526882 0.40430961 0.41666667

166 0.501291990 0.07526882 0.45591154 0.32738095

167 0.604651163 0.16129032 0.44116813 0.23809524

168 0.074935401 0.75268817 0.15820811 0.47619048

169 0.186046512 0.83870968 0.29089878 0.53571429

170 0.423772610 0.01075269 0.36886873 0.47619048

171 0.186046512 0.78494624 0.27757301 0.62500000

172 0.170542636 0.97849462 0.30876099 0.32738095

173 0.056847545 0.72043011 0.17295152 0.50595238

174 0.131782946 0.98924731 0.26424724 0.53571429

175 0.266149871 0.98924731 0.38871562 0.38690476

176 0.056847545 0.70967742 0.09186277 0.35714286

177 0.423772610 0.91397849 0.45307627 0.53571429

178 0.121447028 0.96774194 0.30649277 0.41666667

179 0.134366925 0.89247312 0.38106039 0.53571429

180 0.136950904 1.00000000 0.37765807 0.38690476

181 0.136950904 0.10752688 0.29997165 0.32738095

182 0.059431525 0.56989247 0.05160193 0.56547619

183 0.100775194 0.87096774 0.24128154 0.44642857

184 0.124031008 0.81720430 0.17210094 0.52976190

185 0.186046512 0.93548387 0.27190247 0.41071429

186 0.077519380 0.79569892 0.18202438 0.57738095

187 0.085271318 0.83870968 0.16699745 0.43452381

188 0.612403101 0.24731183 0.73773745 0.29761905

189 0.645994832 0.30107527 0.73064928 0.29761905

190 0.609819121 0.12903226 0.66600510 0.35119048

191 0.731266150 0.31182796 0.73773745 0.28571429

192 0.405684755 0.01075269 0.45931386 0.44047619

193 0.470284238 0.04301075 0.49333711 0.38690476

194 0.341085271 0.81720430 0.39665438 0.57142857

195 0.423772610 0.91397849 0.41735186 0.57142857

196 0.043927649 0.55913978 0.11964843 0.84523810

197 0.077519380 0.60215054 0.15622342 0.83928571

198 0.056847545 0.70967742 0.09186277 0.36904762

199 0.059431525 0.56989247 0.05160193 0.55952381

200 0.405684755 0.01075269 0.57782818 0.57738095

201 0.470284238 0.78494624 0.55599660 0.77380952

202 0.470284238 0.07526882 0.57612702 0.48809524

203 0.490956072 0.96774194 0.44797278 0.58333333

204 0.074935401 0.72043011 0.06010774 0.25000000

205 0.043927649 0.70967742 0.10688971 0.53571429

206 0.074935401 0.75268817 0.15367168 0.50000000

207 0.186046512 0.73118280 0.26991778 0.33333333

208 0.160206718 0.02150538 0.43578112 0.45833333

209 0.645994832 0.30107527 0.65976751 0.30952381

210 0.134366925 0.89247312 0.46980437 0.82738095

211 0.227390181 0.06451613 0.37340516 0.44642857

212 0.258397933 0.12903226 0.62574426 0.51785714

213 0.728682171 0.40860215 0.78451942 0.24404762

214 0.728682171 0.26881720 0.69237312 0.23809524

215 0.604651163 0.17204301 0.63992061 0.41666667

216 0.645994832 0.30107527 0.60731500 0.35714286

217 0.077519380 0.68817204 0.12248370 0.62500000

218 0.111111111 0.80645161 0.15367168 0.40476190

219 0.028423773 0.59139785 0.06010774 0.63095238

220 0.139534884 0.97849462 0.19478310 0.44642857

221 0.043927649 0.70967742 0.09413099 0.52380952

222 0.612403101 0.26881720 0.64275588 0.26785714

223 0.496124031 0.07526882 0.69379076 0.65476190

224 0.645994832 0.26881720 0.71647292 0.33928571

225 0.604651163 0.17204301 0.76041962 0.41071429

226 0.470284238 0.07526882 0.54068614 0.50000000

227 0.421188630 0.04301075 0.51375106 0.52976190

228 0.405684755 0.01075269 0.57187411 0.57738095

229 0.470284238 1.00000000 0.54210377 0.65476190

230 0.857881137 0.40860215 0.73915509 0.18452381

231 0.728682171 0.38709677 0.72356110 0.20238095

232 0.857881137 0.41935484 0.76892543 0.25000000

233 0.731266150 0.29032258 0.77176070 0.38690476

234 0.074935401 0.78494624 0.09271335 0.38690476

235 0.214470284 0.89247312 0.31953502 0.47619048

236 0.074935401 0.75268817 0.18485965 0.60714286

237 0.186046512 0.90322581 0.32378792 0.46428571

238 0.077519380 0.63440860 0.12418486 0.53571429

239 0.077519380 0.83870968 0.13098951 0.47023810

240 0.074935401 0.67741935 0.10547207 0.50000000

241 0.074935401 0.78494624 0.16359512 0.36309524

242 0.201550388 0.98924731 0.34079955 0.38690476

243 0.136950904 0.07526882 0.27984122 0.28571429

244 0.031007752 0.07526882 0.31386447 0.32738095

245 0.056847545 0.53763441 0.10547207 0.80357143

246 0.077519380 0.66666667 0.05301956 0.38095238

247 0.025839793 0.55913978 0.10547207 0.67857143

248 0.043927649 0.70967742 0.12957187 0.63095238

249 0.059431525 0.60215054 0.05301956 0.50000000

250 0.496124031 0.07526882 0.49673944 0.44642857

[ reached 'max' / getOption("max.print") -- omitted 147 rows ]

1. Randomize the order of the rows in the dataset. Copy and paste the commands and execute results in R console here. (5 points)

> autorandomdata = autonormdata[sample(nrow(autonormdata)),]

> autorandomdata

displacement horsepower weight acceleration

324 0.227390181 0.04301075 0.33654664 0.38095238

167 0.604651163 0.16129032 0.44116813 0.23809524

129 0.470284238 0.01075269 0.48851715 0.53571429

299 0.728682171 0.15053763 0.64842642 0.55952381

270 0.095607235 0.75268817 0.17493621 0.38690476

187 0.085271318 0.83870968 0.16699745 0.43452381

307 0.271317829 0.10752688 0.27842359 0.19642857

85 0.074935401 0.89247312 0.13807769 0.50595238

277 0.136950904 0.10752688 0.33512900 0.45833333

362 0.258397933 0.11827957 0.36489935 0.27380952

330 0.059431525 0.67741935 0.06719592 0.34523810

263 0.612403101 0.26881720 0.51375106 0.30952381

329 0.201550388 0.67741935 0.46413382 0.82142857

79 0.134366925 0.88172043 0.38729799 0.68452381

213 0.728682171 0.40860215 0.78451942 0.24404762

37 0.470284238 0.89247312 0.47887723 0.44642857

105 0.857881137 0.37634409 0.93365466 0.26785714

217 0.077519380 0.68817204 0.12248370 0.62500000

366 0.341085271 0.89247312 0.41026368 0.54166667

165 0.421188630 0.07526882 0.40430961 0.41666667

290 0.728682171 0.33333333 0.77884888 0.41071429

388 0.227390181 0.93548387 0.27558832 0.38690476

89 0.604651163 0.21505376 0.68868727 0.38690476

289 0.645994832 0.20430108 0.62857953 0.42857143

340 0.214470284 0.84946237 0.28976467 0.50000000

326 0.056847545 0.53763441 0.13382478 0.81547619

387 0.501291990 0.86021505 0.39750496 0.53571429

42 0.645994832 0.30107527 0.70399773 0.29761905

111 0.103359173 0.95698925 0.21718174 0.50595238

20 0.074935401 0.52688172 0.06294301 0.74404762

44 0.857881137 0.38709677 0.88829033 0.23809524

343 0.173126615 0.84946237 0.21888290 0.29166667

70 0.728682171 0.35483871 0.80606748 0.32738095

121 0.136950904 0.08602151 0.35582648 0.44642857

40 0.857881137 0.39784946 0.80833570 0.20833333

172 0.170542636 0.97849462 0.30876099 0.32738095

25 0.338501292 0.91397849 0.29345052 0.41666667

248 0.043927649 0.70967742 0.12957187 0.63095238

198 0.056847545 0.70967742 0.09186277 0.36904762

39 0.728682171 0.36559140 0.73603629 0.23809524

298 0.297157623 0.77419355 0.54352141 0.72023810

280 0.077519380 0.68817204 0.14800113 0.51190476

160 0.731266150 0.27956989 0.86305642 0.32738095

14 1.000000000 0.50537634 0.41763538 0.11904762

130 0.028423773 0.67741935 0.09554862 0.65476190

45 0.857881137 0.39784946 1.00000000 0.23809524

22 0.100775194 0.91397849 0.23164162 0.38690476

206 0.074935401 0.75268817 0.15367168 0.50000000

230 0.857881137 0.40860215 0.73915509 0.18452381

193 0.470284238 0.04301075 0.49333711 0.38690476

104 0.857881137 0.30107527 0.95945563 0.35714286

372 0.173126615 0.84946237 0.25857669 0.47619048

255 0.341085271 0.86021505 0.38332861 0.46428571

346 0.033591731 0.60215054 0.04167848 0.48214286

347 0.074935401 0.67741935 0.12815424 0.58333333

103 0.074935401 0.52688172 0.09554862 0.77380952

331 0.043927649 0.00000000 0.06294301 0.55357143

13 0.857881137 0.30107527 0.60901616 0.08928571

296 0.077519380 0.80645161 0.08562518 0.38095238

380 0.134366925 0.89247312 0.15508931 0.38690476

176 0.056847545 0.70967742 0.09186277 0.35714286

279 0.054263566 0.72043011 0.10688971 0.41071429

110 0.186046512 0.73118280 0.22341934 0.68452381

84 0.077519380 0.80645161 0.15622342 0.41666667

29 0.609819121 0.43010753 0.88432095 0.62500000

141 0.609819121 0.30107527 0.74964559 0.44642857

252 0.604651163 0.23655914 0.55486249 0.28571429

221 0.043927649 0.70967742 0.09413099 0.52380952

108 0.423772610 0.01075269 0.33342784 0.41666667

304 0.043927649 0.65591398 0.11539552 0.66666667

33 0.077519380 0.00000000 0.12276722 0.65476190

352 0.077519380 0.65591398 0.12248370 0.48809524

149 0.124031008 0.75268817 0.17947264 0.35714286

287 0.604651163 0.16129032 0.59880919 0.32142857

102 0.335917313 0.96774194 0.36603346 0.47619048

145 0.020671835 0.55913978 0.01020697 0.50595238

118 0.000000000 0.54838710 0.07201588 0.68452381

328 0.136950904 0.67741935 0.37907570 0.70833333

107 0.728682171 0.40860215 0.81825914 0.26785714

64 0.857881137 0.39784946 0.78593706 0.23809524

224 0.645994832 0.26881720 0.71647292 0.33928571

342 0.271317829 0.07526882 0.31528211 0.27380952

51 0.124031008 0.91397849 0.14459881 0.35714286

371 0.113695090 0.86021505 0.27275305 0.48809524

377 0.059431525 0.68817204 0.10121917 0.57142857

138 0.728682171 0.30107527 0.87496456 0.38690476

395 0.173126615 0.84946237 0.19336547 0.21428571

394 0.074935401 0.55913978 0.14658350 0.98809524

282 0.341085271 0.86021505 0.39041678 0.60714286

143 0.028423773 0.67741935 0.09923448 0.44642857

285 0.405684755 0.07526882 0.49532180 0.51190476

170 0.423772610 0.01075269 0.36886873 0.47619048

48 0.470284238 0.01075269 0.47320669 0.41666667

204 0.074935401 0.72043011 0.06010774 0.25000000

295 0.046511628 0.65591398 0.10263680 0.42857143

24 0.136950904 0.09677419 0.17607031 0.26785714

181 0.136950904 0.10752688 0.29997165 0.32738095

214 0.728682171 0.26881720 0.69237312 0.23809524

339 0.173126615 0.84946237 0.24865325 0.45833333

225 0.604651163 0.17204301 0.76041962 0.41071429

163 0.490956072 0.07526882 0.60022682 0.65476190

43 0.813953488 0.40860215 0.94754749 0.20833333

1 0.617571059 0.17204301 0.53614970 0.23809524

333 0.054263566 0.62365591 0.06577828 0.43452381

78 0.136950904 0.76344086 0.25460731 0.59523810

284 0.423772610 0.91397849 0.46838673 0.60714286

116 0.728682171 0.26881720 0.70002835 0.29761905

233 0.731266150 0.29032258 0.77176070 0.38690476

61 0.186046512 0.91397849 0.22540403 0.68452381

86 0.728682171 0.39784946 0.70513184 0.29761905

335 0.005167959 0.01075269 0.22880635 0.26785714

49 0.470284238 0.89247312 0.43266232 0.38690476

242 0.201550388 0.98924731 0.34079955 0.38690476

246 0.077519380 0.66666667 0.05301956 0.38095238

247 0.025839793 0.55913978 0.10547207 0.67857143

239 0.077519380 0.83870968 0.13098951 0.47023810

219 0.028423773 0.59139785 0.06010774 0.63095238

135 0.490956072 0.07526882 0.57244117 0.59523810

369 0.113695090 0.89247312 0.29118231 0.63095238

368 0.113695090 0.89247312 0.28125886 0.69047619

315 0.186046512 0.89247312 0.35639354 0.60119048

53 0.051679587 0.76344086 0.12815424 0.38690476

353 0.077519380 0.65591398 0.21746527 0.75595238

65 0.645994832 0.30107527 0.71505529 0.32738095

381 0.100775194 0.75268817 0.16784803 0.38690476

124 0.227390181 0.13978495 0.33853133 0.32738095

77 0.136950904 0.08602151 0.37425574 0.38690476

218 0.111111111 0.80645161 0.15367168 0.40476190

98 0.405684755 0.04301075 0.42755883 0.50595238

194 0.341085271 0.81720430 0.39665438 0.57142857

19 0.074935401 0.89247312 0.14658350 0.38690476

31 0.186046512 0.91397849 0.18457613 0.44642857

174 0.131782946 0.98924731 0.26424724 0.53571429

237 0.186046512 0.90322581 0.32378792 0.46428571

75 0.604651163 0.24731183 0.76013609 0.47619048

16 0.335917313 0.96774194 0.34590303 0.44642857

363 0.201550388 0.12903226 0.37340516 0.34523810

92 0.857881137 0.30107527 0.80833570 0.23809524

122 0.645994832 0.30107527 0.50637936 0.17857143

152 0.028423773 0.67741935 0.10972498 0.47619048

312 0.077519380 0.70967742 0.14374823 0.44642857

207 0.186046512 0.73118280 0.26991778 0.33333333

244 0.031007752 0.07526882 0.31386447 0.32738095

229 0.470284238 1.00000000 0.54210377 0.65476190

249 0.059431525 0.60215054 0.05301956 0.50000000

355 0.082687339 0.00000000 0.20045364 0.46428571

254 0.341085271 0.96774194 0.43719875 0.60714286

396 0.134366925 0.79569892 0.28692940 0.63095238

223 0.496124031 0.07526882 0.69379076 0.65476190

378 0.095607235 0.63440860 0.14516586 0.39880952

314 0.214470284 0.91397849 0.30195634 0.50595238

140 0.604651163 0.24731183 0.85766941 0.47619048

126 0.335917313 0.96774194 0.42217182 0.50595238

354 0.095607235 0.74193548 0.16359512 0.36904762

349 0.054263566 0.62365591 0.12390133 0.55357143

291 0.731266150 0.25806452 0.69208959 0.37500000

262 0.490956072 0.12903226 0.50949816 0.42261905

15 0.116279070 0.96774194 0.21519705 0.41666667

275 0.162790698 0.03225806 0.34505245 0.47023810

274 0.131782946 0.98924731 0.22455344 0.41071429

195 0.423772610 0.91397849 0.41735186 0.57142857

201 0.470284238 0.78494624 0.55599660 0.77380952

323 0.046511628 0.65591398 0.14091296 0.58928571

17 0.338501292 0.98924731 0.32917494 0.44642857

212 0.258397933 0.12903226 0.62574426 0.51785714

127 0.341085271 0.00000000 0.35781117 0.53571429

133 0.186046512 0.75268817 0.26339665 0.53571429

41 0.731266150 0.32258065 0.72044230 0.32738095

281 0.421188630 0.10752688 0.46271619 0.44047619

385 0.059431525 0.67741935 0.10830734 0.48809524

159 0.645994832 0.30107527 0.81797562 0.38690476

117 0.857881137 0.51612903 0.75559966 0.08928571

72 0.005167959 0.98924731 0.20328891 0.32738095

36 0.470284238 0.01075269 0.48653246 0.44642857

286 0.612403101 0.17204301 0.63141480 0.44047619

243 0.136950904 0.07526882 0.27984122 0.28571429

157 0.857881137 0.38709677 0.86617522 0.20833333

383 0.059431525 0.67741935 0.09980153 0.41666667

318 0.074935401 0.78494624 0.16302807 0.46428571

311 0.054263566 0.60215054 0.10065211 0.64285714

276 0.245478036 0.15053763 0.43294585 0.33333333

313 0.046511628 0.65591398 0.11511199 0.50000000

106 0.754521964 0.38709677 0.86220584 0.29761905

185 0.186046512 0.93548387 0.27190247 0.41071429

88 0.728682171 0.26881720 0.67337681 0.29761905

226 0.470284238 0.07526882 0.54068614 0.50000000

338 0.100775194 0.73118280 0.19194783 0.53571429

235 0.214470284 0.89247312 0.31953502 0.47619048

50 0.139534884 0.87096774 0.17210094 0.35714286

80 0.072351421 0.69892473 0.16331160 0.59523810

30 0.074935401 0.89247312 0.14658350 0.38690476

93 0.731266150 0.34408602 0.77969946 0.29761905

309 0.214470284 0.91397849 0.26736603 0.30952381

357 0.103359173 0.75268817 0.20895946 0.52380952

245 0.056847545 0.53763441 0.10547207 0.80357143

293 0.754521964 0.30107527 0.65976751 0.29761905

164 0.405684755 0.96774194 0.61582081 0.65476190

168 0.074935401 0.75268817 0.15820811 0.47619048

360 0.188630491 0.80645161 0.45846328 0.73809524

320 0.134366925 0.75268817 0.26339665 0.56547619

184 0.124031008 0.81720430 0.17210094 0.52976190

391 0.173126615 0.84946237 0.21463000 0.29761905

100 0.423772610 0.01075269 0.37765807 0.47619048

113 0.139534884 0.86021505 0.19761837 0.62500000

364 0.421188630 0.07526882 0.51091579 0.46428571

73 0.609819121 0.30107527 0.64615821 0.26785714

27 0.617571059 0.45161290 0.78338531 0.41666667

222 0.612403101 0.26881720 0.64275588 0.26785714

240 0.074935401 0.67741935 0.10547207 0.50000000

300 0.188630491 0.72043011 0.44712220 1.00000000

38 0.423772610 0.01075269 0.47490785 0.44642857

62 0.139534884 0.87096774 0.17380210 0.50595238

134 0.470284238 0.01075269 0.61468670 0.53571429

132 0.007751938 0.65591398 0.06322654 0.77380952

35 0.405684755 0.04301075 0.51772044 0.44642857

125 0.728682171 0.40860215 0.58151403 0.17857143

99 0.470284238 0.01075269 0.47207258 0.59523810

271 0.170542636 0.96774194 0.25574142 0.40476190

273 0.214470284 0.86021505 0.35214063 0.57142857

71 0.857881137 0.41935484 0.79642756 0.26785714

153 0.405684755 0.96774194 0.46810320 0.47619048

266 0.645994832 0.24731183 0.69946130 0.33928571

382 0.103359173 0.70967742 0.17918911 0.52976190

28 0.645994832 0.47311828 0.78508648 0.32738095

186 0.077519380 0.79569892 0.18202438 0.57738095

148 0.056847545 0.75268817 0.14034590 0.44642857

294 0.054263566 0.72043011 0.08846045 0.35714286

232 0.857881137 0.41935484 0.76892543 0.25000000

370 0.113695090 0.89247312 0.22171817 0.59523810

60 0.074935401 0.58064516 0.18174086 0.92261905

265 0.604651163 0.23655914 0.45137511 0.19047619

202 0.470284238 0.07526882 0.57612702 0.48809524

337 0.186046512 0.00000000 0.36631698 0.37500000

12 0.702842377 0.35483871 0.56592005 0.00000000

350 0.059431525 0.68817204 0.10547207 0.47619048

208 0.160206718 0.02150538 0.43578112 0.45833333

367 0.405684755 0.86021505 0.52509215 0.51190476

269 0.131782946 0.98924731 0.19478310 0.39880952

26 0.754521964 0.48387097 0.85114828 0.35714286

327 0.056847545 0.53763441 0.20470655 0.93452381

197 0.077519380 0.60215054 0.15622342 0.83928571

216 0.645994832 0.30107527 0.60731500 0.35714286

288 0.731266150 0.22580645 0.66402041 0.30952381

302 0.095607235 0.70967742 0.16643039 0.30952381

175 0.266149871 0.98924731 0.38871562 0.38690476

228 0.405684755 0.01075269 0.57187411 0.57738095

189 0.645994832 0.30107527 0.73064928 0.29761905

345 0.046511628 0.64516129 0.07428409 0.50000000

82 0.074935401 0.93548387 0.19138078 0.53571429

97 0.754521964 0.39784946 0.62602779 0.17857143

[ reached 'max' / getOption("max.print") -- omitted 147 rows ]

1. Split the data into a training set and a test set. Use a test set of 100 rows. Copy and paste the commands and execute results in R console here. (5 points)

> autoTrainData = autorandomdata[1:291, ]

> autoTestData = autorandomdata[292:392, ]

> autodata\_trainlabels = newauto[1:291, 5]

> autodata\_testlabels = newauto[292:392, 5]

1. Perform kNN on the training data, with several values of K, in order to predict mpg1. What  
   test errors do you obtain? Which value of K seems to perform the best on this data set? Copy and paste the commands and execute results in R console here. (25 points)

> autoknn <- knn(train = autoTrainData, test = autoTestData,cl = autodata\_trainlabels, k = 1)

> CrossTable(x = autodata\_testlabels, y = autoknn, prop.chisq=FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 101

| autoknn

autodata\_testlabels | 0 | 1 | Row Total |

--------------------|-----------|-----------|-----------|

0 | 6 | 2 | 8 |

| 0.750 | 0.250 | 0.079 |

| 0.086 | 0.065 | |

| 0.059 | 0.020 | |

--------------------|-----------|-----------|-----------|

1 | 64 | 29 | 93 |

| 0.688 | 0.312 | 0.921 |

| 0.914 | 0.935 | |

| 0.634 | 0.287 | |

--------------------|-----------|-----------|-----------|

Column Total | 70 | 31 | 101 |

| 0.693 | 0.307 | |

--------------------|-----------|-----------|-----------|

> autoknn <- knn(train = autoTrainData, test = autoTestData,cl = autodata\_trainlabels, k = 3)

> CrossTable(x = autodata\_testlabels, y = autoknn, prop.chisq=FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 101

| autoknn

autodata\_testlabels | 0 | 1 | Row Total |

--------------------|-----------|-----------|-----------|

0 | 8 | 0 | 8 |

| 1.000 | 0.000 | 0.079 |

| 0.103 | 0.000 | |

| 0.079 | 0.000 | |

--------------------|-----------|-----------|-----------|

1 | 70 | 23 | 93 |

| 0.753 | 0.247 | 0.921 |

| 0.897 | 1.000 | |

| 0.693 | 0.228 | |

--------------------|-----------|-----------|-----------|

Column Total | 78 | 23 | 101 |

| 0.772 | 0.228 | |

--------------------|-----------|-----------|-----------|

> autoknn <- knn(train = autoTrainData, test = autoTestData,cl = autodata\_trainlabels, k = 5)

> CrossTable(x = autodata\_testlabels, y = autoknn, prop.chisq=FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 101

| autoknn

autodata\_testlabels | 0 | 1 | Row Total |

--------------------|-----------|-----------|-----------|

0 | 8 | 0 | 8 |

| 1.000 | 0.000 | 0.079 |

| 0.100 | 0.000 | |

| 0.079 | 0.000 | |

--------------------|-----------|-----------|-----------|

1 | 72 | 21 | 93 |

| 0.774 | 0.226 | 0.921 |

| 0.900 | 1.000 | |

| 0.713 | 0.208 | |

--------------------|-----------|-----------|-----------|

Column Total | 80 | 21 | 101 |

| 0.792 | 0.208 | |

--------------------|-----------|-----------|-----------|

> autoknn <- knn(train = autoTrainData, test = autoTestData,cl = autodata\_trainlabels, k = 8)

> CrossTable(x = autodata\_testlabels, y = autoknn, prop.chisq=FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 101

| autoknn

autodata\_testlabels | 0 | 1 | Row Total |

--------------------|-----------|-----------|-----------|

0 | 7 | 1 | 8 |

| 0.875 | 0.125 | 0.079 |

| 0.079 | 0.083 | |

| 0.069 | 0.010 | |

--------------------|-----------|-----------|-----------|

1 | 82 | 11 | 93 |

| 0.882 | 0.118 | 0.921 |

| 0.921 | 0.917 | |

| 0.812 | 0.109 | |

--------------------|-----------|-----------|-----------|

Column Total | 89 | 12 | 101 |

| 0.881 | 0.119 | |

--------------------|-----------|-----------|-----------|

Commands you might need, which are not on the lecture slides:  
as.intereger(): to coerce a string to integer

sapply(<vector>,<function>): to apply a function to each element of a vector and return

a vector of results

sample(nrow(<data frame)): to create a random sample from 1 to the number of rows in

<data frame>