```
library(corrplot)
library(glue)
library(ISLR2)
library(psych)

options(repr.plot.width = 20, repr.plot.height = 20)
my_corrplot = function(x) {
    corrplot(x, method = "circle", type = "lower",
         tl.cex = 2,  # text label size
         cl.cex = 2,  # color legend text size
         number.cex = 2  # size of correlation coefficients
    )
}
```

add Code add Markdown

Chapter 2 Conceptual

1. For each of parts (a) through (d), indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answer.
(a) The sample size n is extremely large, and the number of predictors p is small.
When the sample size is very large and the number of predictors is small, flexible methods are generally more suitable. The large dataset offers protection against overfitting, and a flexible model can more effectively find valuable relationships between the predictors and the dependent variable.

(b) The number of predictors p is extremely large, and the number of observations n is small.
When there's high dimensionality and a small sample size, the risk of overfitting becomes significant. This scenario clearly calls for a more rigid model; otherwise, the algorithm may establish arbitrary relationships that are not predictive in another dataset.

(c) The relationship between the predictors and response is highly non-linear.
Inflexible models struggle to capture the relationships between independent and dependent variables in a non-linear system. It's challenging to fit a linear system to data that isn't linearly dependent. A more flexible model will better approximate the relationship.

(d) The variance of the error terms, i.e. $\sigma 2 = \text{Var}(\epsilon)$, is extremely high.
Linear models excel in high-variance datasets. A more flexible model is likely to overfit to the noise in the data, whereas a linear model might not capture all the structure but will avoid overfitting.

2. Explain whether each scenario is a classification or regression problem, and indicate whether we are most interested in inference or prediction. Finally, provide n and p.
(a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary. This is a regression problem since the dependent variable, the CEO's salary, is continuous, not discrete. The question states, "We are interested in understanding which factors affect CEO salary," hence, it's an inference problem. The goal is not to predict the CEO's salary, but to understand which features influence it. The sample size is 500 (the number of firms), and there are three features: profit, number of employees, and industry.

(b) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether

it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.
The success or failure is binary, so this is a classification problem. The desired outcome is forward-looking, aiming to predict what will happen, so it is predictive in nature. The sample size is 20 (the number of products), and there are 13 features: product price, marketing budget, competition price, and ten others. Whether the product was a success or failure is a dependent variable.

(c) We are interested in predicting the % change in the USD/Euro exchange rate in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the USD/Euro, the % change in the US market, the % change in the British market, and the % change in the German market.
The % change in the exchange rate will yield a continuous output, so this is a regression problem. The question states we are interested in predicting the percent change, hence, it's a prediction problem. There are 52 samples, one for each week of the year, and three features: the percent change in the US, British, and German markets.

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

[70]:
```
Table2_4_7 = data.frame(
    "Obs" = c(1:6),
    "X_1" = c(0,2,0,0,-1,1),
    "X_2" = c(3,0,1,1,0,1),
    "X_3" = c(0,0,3,2,1,1),
    "Y" = c('Red', 'Red', 'Red', 'Green', 'Green', 'Red')
)
View(Table2_4_7)
```

A data.frame: 6 × 5

Obs	X_1	X_2	X_3	Y
<int>	<dbl>	<dbl>	<dbl>	<chr>
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	-1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when X1 = X2 = X3 = 0 using K-nearest neighbors.

(a) Compute the Euclidean distance between each observation and the test point, X1 = X2 = X3 = 0.
The Euclidean Distance formula is sqrt((p1-q1)^2 + (p2-q2)^2+...+(pn-qn)^2). We need to find the distance between each observation and the origin (0,0,0).

Observation 1: sqrt((0-0)^2 + (3-0)^2 + (0-0)^2) = 3 Observation 2: sqrt((2-0)^2 + (0-0)^2 + (0-0)^2) = 2 Observation 3: sqrt((0-0)^2 + (1-0)^2 + (3-0)^2) = sqrt(10) ~3.16 Observation 4: sqrt((0-0)^2 + (1-0)^2 + (2-0)^2) = sqrt(5) ~2.24 Observation 5: sqrt((-1-0)^2 + (0-0)^2 + (1-0)^2) = sqrt(2) ~1.41 Observation 6: sqrt((1-0)^2 + (1-0)^2 + (1-0)^2) = sqrt(3) ~1.73

(b) What is our prediction with K = 1? Why?
The closest neighbor to (0, 0, 0) in our dataset is Observation 5, as 1.41 is the nearest value and we are only selecting one neighbor. Thus, we would predict 'Green'.

(c) What is our prediction with K = 3? Why?
With K=3, we select the three closest neighbors to (0,0,0). These are observations 5, 6, and 2. The Y values for 5, 6, and 2 are Green, Red, and Red, respectively. Since two out of the three neighbors are red, we would predict 'Red'.

(d) If the Bayes decision boundary in this problem is highly nonlinear, then would we expect the best value for K to be large or small? Why?
A large K value would smooth or flatten the decision boundary towards linearity. Therefore, in this case, it would be better to have a small K value. However, if we select too small a K, it may result in overfitting.

Chapter 2 Applied:

Exercise 8: The College Dataset.

[71]:
```
college = read.csv("/kaggle/input/cs5565-datasets/ALL CSV FILES - 2nd Edition/Col
```

[72]:
```
rownames(college) = college[,1]
View(college)
```

A data.frame: 777 × 19

	X	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outsta
	<chr>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int
Abilene Christian University	Abilene Christian University	Yes	1660	1232	721	23	52	2885	537	74
Adelphi University	Adelphi University	Yes	2186	1924	512	16	29	2683	1227	122
Adrian College	Adrian College	Yes	1428	1097	336	22	50	1036	99	112
Agnes Scott College	Agnes Scott College	Yes	417	349	137	60	89	510	63	129
Alaska Pacific University	Alaska Pacific University	Yes	193	146	55	16	44	249	869	75
Albertson College	Albertson College	Yes	587	479	158	38	62	678	41	135

	X	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outsta
	<chr>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int
Albertus Magnus College	Albertus Magnus College	Yes	353	340	103	17	45	416	230	132
Albion College	Albion College	Yes	1899	1720	489	37	68	1594	32	138
Albright College	Albright College	Yes	1038	839	227	30	63	973	306	155
Alderson-Broaddus College	Alderson-Broaddus College	Yes	582	498	172	21	44	799	78	104
Alfred University	Alfred University	Yes	1732	1425	472	37	75	1830	110	165
Allegheny College	Allegheny College	Yes	2652	1900	484	44	77	1707	44	170
Allentown Coll. of St. Francis de Sales	Allentown Coll. of St. Francis de Sales	Yes	1179	780	290	38	64	1130	638	96
Alma College	Alma College	Yes	1267	1080	385	44	73	1306	28	125
Alverno College	Alverno College	Yes	494	313	157	23	46	1317	1235	83
American International College	American International College	Yes	1420	1093	220	9	22	1018	287	87
Amherst College	Amherst College	Yes	4302	992	418	83	96	1593	5	197
Anderson University	Anderson University	Yes	1216	908	423	19	40	1819	281	101
Andrews University	Andrews University	Yes	1130	704	322	14	23	1586	326	99
Angelo State University	Angelo State University	No	3540	2001	1016	24	54	4190	1512	51
Antioch University	Antioch University	Yes	713	661	252	25	44	712	23	154
Appalachian State University	Appalachian State University	No	7313	4664	1910	20	63	9940	1035	68
Aquinas College	Aquinas College	Yes	619	516	219	20	51	1251	767	112
Arizona State University Main campus	Arizona State University Main campus	No	12809	10308	3761	24	49	22593	7585	74
Arkansas College	Arkansas College	Yes	708	334	166	46	74	530	182	86

X	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outsta
<chr>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int
(Lyon College) (Lyon College)									
Arkansas Tech University Arkansas Tech University	No	1734	1729	951	12	52	3602	939	34
Assumption College Assumption College	Yes	2135	1700	491	23	59	1708	689	120
Auburn University-Main Campus Auburn University-Main Campus	No	7548	6791	3070	25	57	16262	1716	63
Augsburg College Augsburg College	Yes	662	513	257	12	30	2074	726	119
Augustana College IL Augustana College IL	Yes	1879	1658	497	36	69	1950	38	133
⋮ ⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Westfield State College Westfield State College	No	3100	2150	825	3	20	3234	941	55
Westminster College MO Westminster College MO	Yes	662	553	184	20	43	665	37	107
Westminster College Westminster College	Yes	996	866	377	29	58	1411	72	120
Westminster College of Salt Lake City Westminster College of Salt Lake City	Yes	917	720	213	21	60	979	743	88
Westmont College Westmont College	No	950	713	351	42	72	1276	9	143
Wheaton College IL Wheaton College IL	Yes	1432	920	548	56	84	2200	56	114
Westminster College PA Westminster College PA	Yes	1738	1373	417	21	55	1335	30	184
Wheeling Jesuit College Wheeling Jesuit College	Yes	903	755	213	15	49	971	305	105
Whitman College Whitman College	Yes	1861	998	359	45	77	1220	46	166
Whittier College Whittier College	Yes	1681	1069	344	35	63	1235	30	162
Whitworth College Whitworth College	Yes	1121	926	372	43	70	1270	160	126
Widener University Widener University	Yes	2139	1492	502	24	64	2186	2171	123
Wilkes University Wilkes University	Yes	1631	1431	434	15	36	1803	603	111

	X	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outsta
	<chr>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int
Willamette University	Willamette University	Yes	1658	1327	395	49	80	1595	159	148
William Jewell College	William Jewell College	Yes	663	547	315	32	67	1279	75	100
William Woods University	William Woods University	Yes	469	435	227	17	39	851	120	105
Williams College	Williams College	Yes	4186	1245	526	81	96	1988	29	196
Wilson College	Wilson College	Yes	167	130	46	16	50	199	676	114
Wingate College	Wingate College	Yes	1239	1017	383	10	34	1207	157	78
Winona State University	Winona State University	No	3325	2047	1301	20	45	5800	872	42
Winthrop University	Winthrop University	No	2320	1805	769	24	61	3395	670	64
Wisconsin Lutheran College	Wisconsin Lutheran College	Yes	152	128	75	17	41	282	22	91
Wittenberg University	Wittenberg University	Yes	1979	1739	575	42	68	1980	144	159
Wofford College	Wofford College	Yes	1501	935	273	51	83	1059	34	126
Worcester Polytechnic Institute	Worcester Polytechnic Institute	Yes	2768	2314	682	49	86	2802	86	158
Worcester State College	Worcester State College	No	2197	1515	543	4	26	3089	2029	67
Xavier University	Xavier University	Yes	1959	1805	695	24	47	2849	1107	115
Xavier University of Louisiana	Xavier University of Louisiana	Yes	2097	1915	695	34	61	2793	166	69
Yale University	Yale University	Yes	10705	2453	1317	95	99	5217	83	198
York College of Pennsylvania	York College of Pennsylvania	Yes	2989	1855	691	28	63	2988	1726	49

[73]:

```
college = college[,-1]
View(college)
```

A data.frame: 777 × 18

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Boa
	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537	7440	33
Adelphi University	Yes	2186	1924	512	16	29	2683	1227	12280	64
Adrian College	Yes	1428	1097	336	22	50	1036	99	11250	37
Agnes Scott College	Yes	417	349	137	60	89	510	63	12960	54
Alaska Pacific University	Yes	193	146	55	16	44	249	869	7560	41
Albertson College	Yes	587	479	158	38	62	678	41	13500	33
Albertus Magnus College	Yes	353	340	103	17	45	416	230	13290	57
Albion College	Yes	1899	1720	489	37	68	1594	32	13868	48
Albright College	Yes	1038	839	227	30	63	973	306	15595	44
Alderson-Broaddus College	Yes	582	498	172	21	44	799	78	10468	33
Alfred University	Yes	1732	1425	472	37	75	1830	110	16548	54
Allegheny College	Yes	2652	1900	484	44	77	1707	44	17080	44
Allentown Coll. of St. Francis de Sales	Yes	1179	780	290	38	64	1130	638	9690	47
Alma College	Yes	1267	1080	385	44	73	1306	28	12572	45
Alverno College	Yes	494	313	157	23	46	1317	1235	8352	36
American International College	Yes	1420	1093	220	9	22	1018	287	8700	47
Amherst College	Yes	4302	992	418	83	96	1593	5	19760	53
Anderson University	Yes	1216	908	423	19	40	1819	281	10100	35

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Boa
	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int
Andrews University	Yes	1130	704	322	14	23	1586	326	9996	30
Angelo State University	No	3540	2001	1016	24	54	4190	1512	5130	35
Antioch University	Yes	713	661	252	25	44	712	23	15476	33
Appalachian State University	No	7313	4664	1910	20	63	9940	1035	6806	25
Aquinas College	Yes	619	516	219	20	51	1251	767	11208	41
Arizona State University Main campus	No	12809	10308	3761	24	49	22593	7585	7434	48
Arkansas College (Lyon College)	Yes	708	334	166	46	74	530	182	8644	39
Arkansas Tech University	No	1734	1729	951	12	52	3602	939	3460	26
Assumption College	Yes	2135	1700	491	23	59	1708	689	12000	59
Auburn University-Main Campus	No	7548	6791	3070	25	57	16262	1716	6300	39
Augsburg College	Yes	662	513	257	12	30	2074	726	11902	43
Augustana College IL	Yes	1879	1658	497	36	69	1950	38	13353	41
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Westfield State College	No	3100	2150	825	3	20	3234	941	5542	37
Westminster College MO	Yes	662	553	184	20	43	665	37	10720	40
Westminster College	Yes	996	866	377	29	58	1411	72	12065	36
Westminster College of Salt Lake City	Yes	917	720	213	21	60	979	743	8820	40
Westmont College	No	950	713	351	42	72	1276	9	14320	53

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Boa
	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int
Wheaton College IL	Yes	1432	920	548	56	84	2200	56	11480	42
Westminster College PA	Yes	1738	1373	417	21	55	1335	30	18460	59
Wheeling Jesuit College	Yes	903	755	213	15	49	971	305	10500	45
Whitman College	Yes	1861	998	359	45	77	1220	46	16670	49
Whittier College	Yes	1681	1069	344	35	63	1235	30	16249	56
Whitworth College	Yes	1121	926	372	43	70	1270	160	12660	45
Widener University	Yes	2139	1492	502	24	64	2186	2171	12350	53
Wilkes University	Yes	1631	1431	434	15	36	1803	603	11150	51
Willamette University	Yes	1658	1327	395	49	80	1595	159	14800	46
William Jewell College	Yes	663	547	315	32	67	1279	75	10060	29
William Woods University	Yes	469	435	227	17	39	851	120	10535	43
Williams College	Yes	4186	1245	526	81	96	1988	29	19629	57
Wilson College	Yes	167	130	46	16	50	199	676	11428	50
Wingate College	Yes	1239	1017	383	10	34	1207	157	7820	34
Winona State University	No	3325	2047	1301	20	45	5800	872	4200	27
Winthrop University	No	2320	1805	769	24	61	3395	670	6400	33
Wisconsin Lutheran College	Yes	152	128	75	17	41	282	22	9100	37
Wittenberg University	Yes	1979	1739	575	42	68	1980	144	15948	44
Wofford College	Yes	1501	935	273	51	83	1059	34	12680	41
Worcester Polytechnic	Yes	2768	2314	682	49	86	2802	86	15884	53

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Boa
	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int
Institute										
Worcester State College	No	2197	1515	543	4	26	3089	2029	6797	39
Xavier University	Yes	1959	1805	695	24	47	2849	1107	11520	49
Xavier University of Louisiana	Yes	2097	1915	695	34	61	2793	166	6900	42
Yale University	Yes	10705	2453	1317	95	99	5217	83	19840	65
York College of Pennsylvania	Yes	2989	1855	691	28	63	2988	1726	4990	35

[74]:
```
summary(college)
```

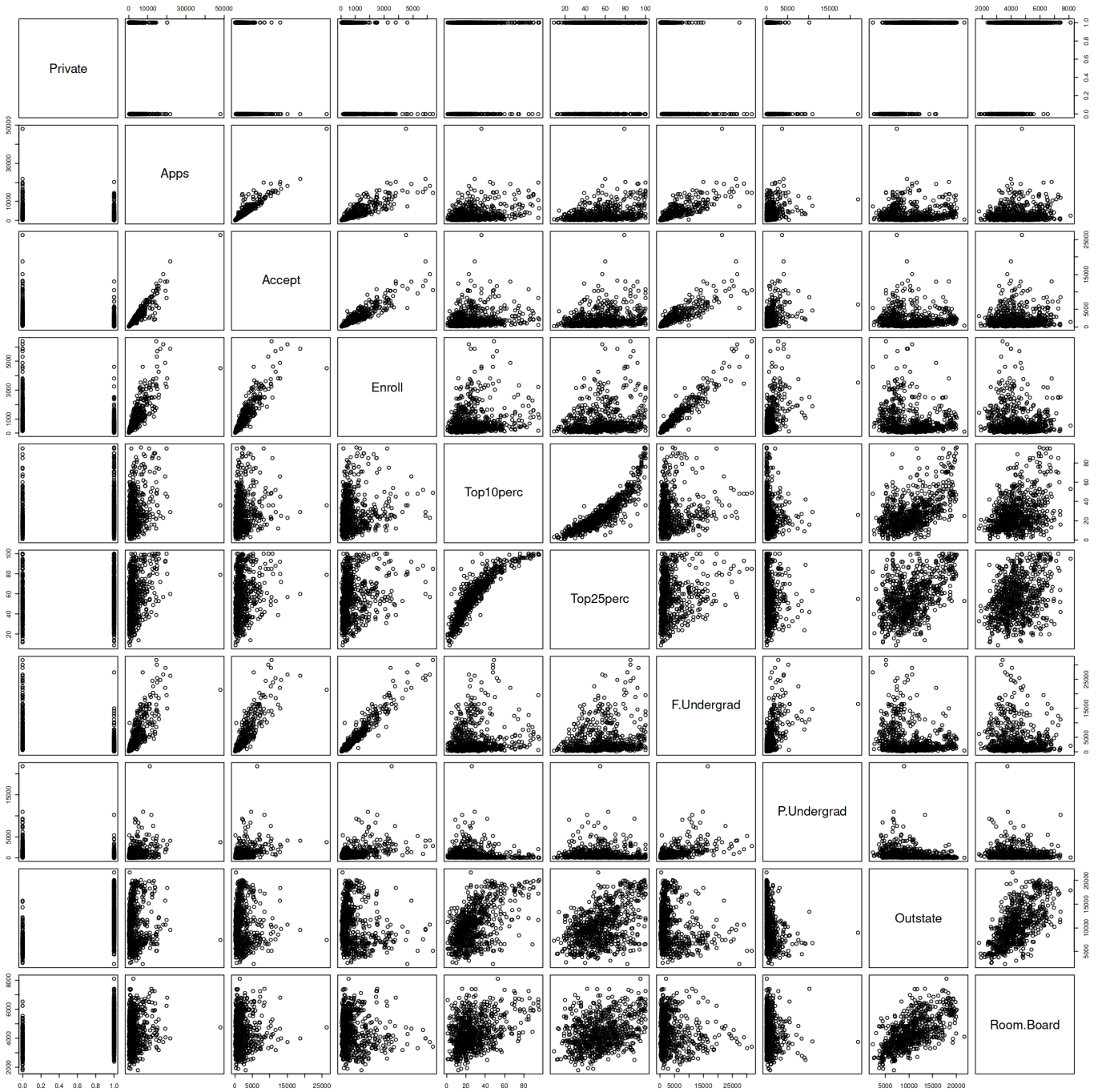
```
  Private             Apps           Accept          Enroll       
 Length:777         Min.   :   81   Min.   :   72   Min.   :  35  
 Class :character   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242  
 Mode  :character   Median : 1558   Median : 1110   Median : 434  
                    Mean   : 3002   Mean   : 2019   Mean   : 780  
                    3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902  
                    Max.   :48094   Max.   :26330   Max.   :6392  
   Top10perc       Top25perc      F.Undergrad     P.Undergrad     
 Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0  
 1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0  
 Median :23.00   Median : 54.0   Median : 1707   Median :  353.0  
 Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3  
 3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0  
 Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0  
    Outstate       Room.Board       Books           Personal     
 Min.   : 2340   Min.   :1780   Min.   :  96.0   Min.   :  250  
 1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0   1st Qu.:  850  
 Median : 9990   Median :4200   Median : 500.0   Median :1200  
 Mean   :10441   Mean   :4358   Mean   : 549.4   Mean   :1341  
 3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700  
 Max.   :21700   Max.   :8124   Max.   :2340.0   Max.   :6800  
      PhD            Terminal       S.F.Ratio      perc.alumni     
 Min.   :  8.00   Min.   : 24.0   Min.   : 2.50   Min.   : 0.00  
 1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00  
 Median : 75.00   Median : 82.0   Median :13.60   Median :21.00  
 Mean   : 72.66   Mean   : 79.7   Mean   :14.09   Mean   :22.74  
 3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00  
 Max.   :103.00   Max.   :100.0   Max.   :39.80   Max.   :64.00  
```

```
     Expend         Grad.Rate
 Min.   : 3186    Min.   : 10.00
 1st Qu.: 6751    1st Qu.: 53.00
 Median : 8377    Median : 65.00
 Mean   : 9660    Mean   : 65.46
 3rd Qu.:10830    3rd Qu.: 78.00
 Max.   :56233    Max.   :118.00
```

pairs(college[,1:10] threw an error due to private being a string. That's why I quantized it below.

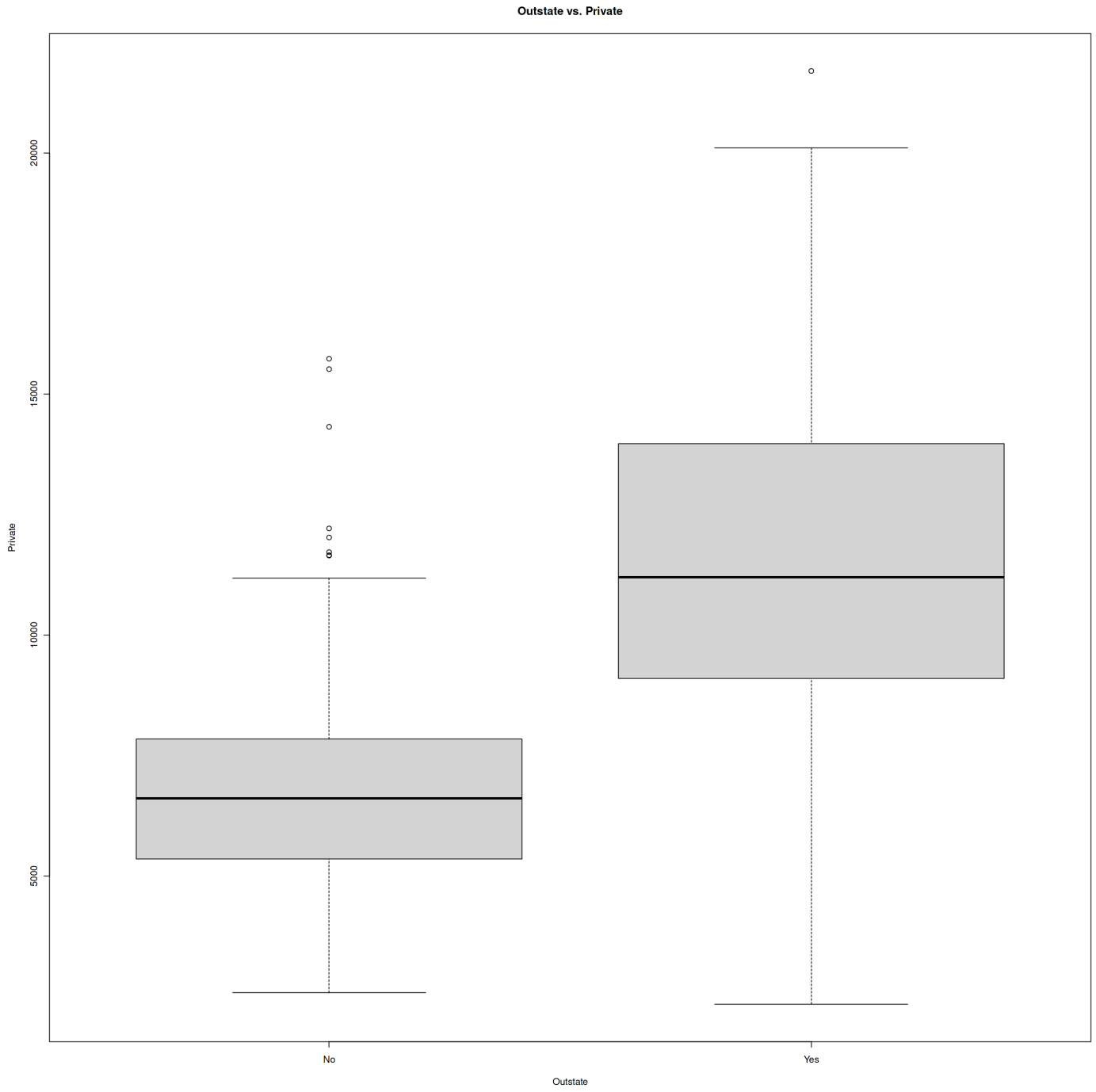
[75]:

```
college$Private = ifelse(college$Private == 'Yes', 1, 0)
pairs(college[,1:10], cex.labels = 2.0)
```



[76]:

```
college$Private = ifelse(college$Private == 1, "Yes", "No")
boxplot(college$Outstate ~ college$Private,
        main = 'Outstate vs. Private',
        xlab = "Outstate",
        ylab = 'Private',
       )
```



[77]:
```
Elite = rep('No', nrow(college))
Elite[college$Top10perc > 50] = 'Yes'
Elite = as.factor(Elite)
college <- data.frame(college, Elite)
```

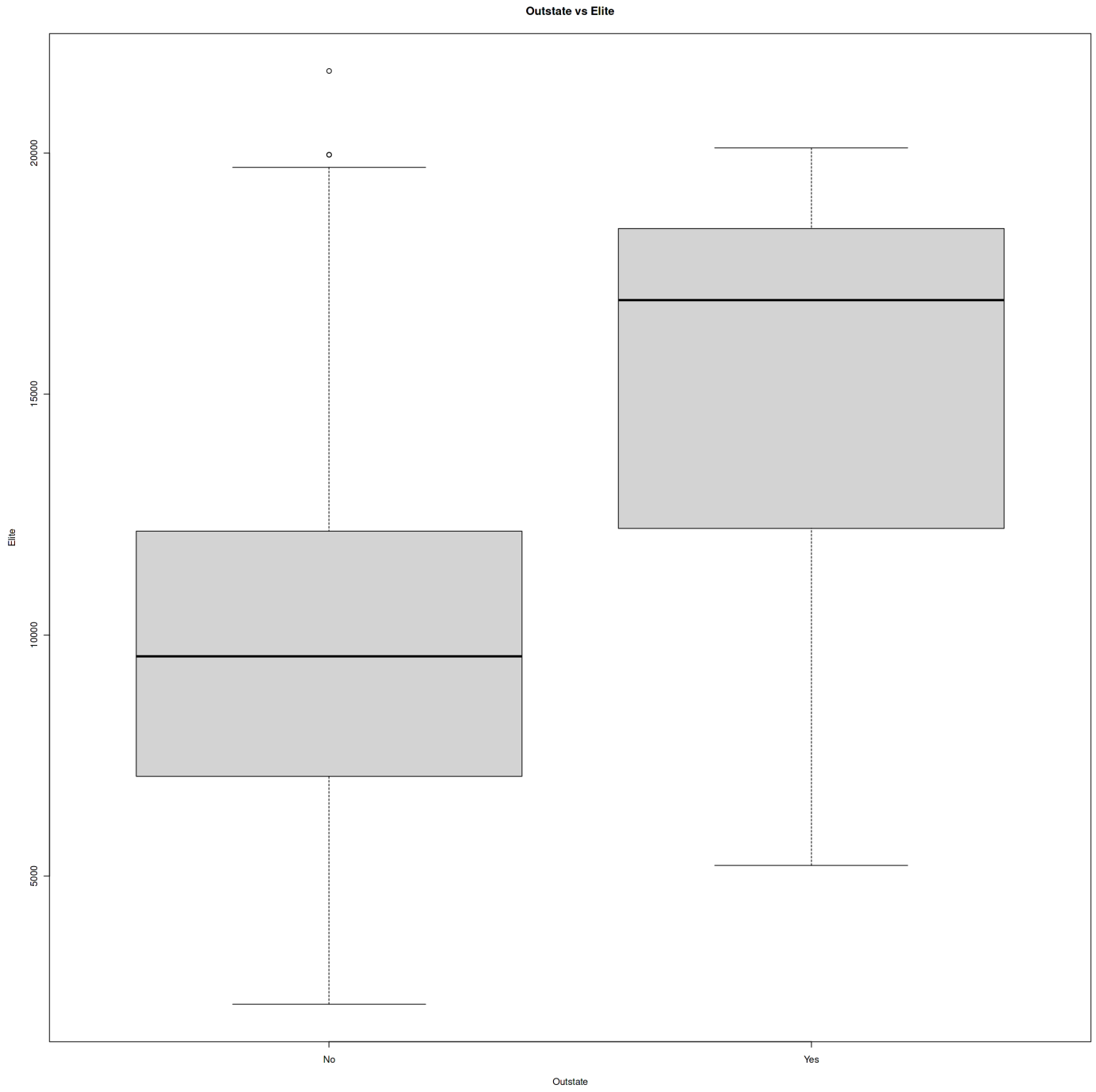
[78]:
```
summary(college)
```

```
  Private        Apps           Accept          Enroll
 Length:777     Min.   :   81   Min.   :   72   Min.   :  35
```

```
Class :character   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242
Mode  :character   Median : 1558   Median : 1110   Median : 434
                   Mean   : 3002   Mean   : 2019   Mean   : 780
                   3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902
                   Max.   :48094   Max.   :26330   Max.   :6392
  Top10perc       Top25perc      F.Undergrad     P.Undergrad
Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :   1.0
1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:  95.0
Median :23.00   Median : 54.0   Median : 1707   Median :  353.0
Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3
3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0
Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0
   Outstate       Room.Board       Books           Personal
Min.   : 2340   Min.   :1780   Min.   :  96.0   Min.   : 250
1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850
Median : 9990   Median :4200   Median : 500.0   Median :1200
Mean   :10441   Mean   :4358   Mean   : 549.4   Mean   :1341
3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700
Max.   :21700   Max.   :8124   Max.   :2340.0   Max.   :6800
     PhD            Terminal       S.F.Ratio       perc.alumni
Min.   :  8.00   Min.   : 24.0   Min.   : 2.50   Min.   : 0.00
1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00
Median : 75.00   Median : 82.0   Median :13.60   Median :21.00
Mean   : 72.66   Mean   : 79.7   Mean   :14.09   Mean   :22.74
3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00
Max.   :103.00   Max.   :100.0   Max.   :39.80   Max.   :64.00
    Expend        Grad.Rate      Elite
Min.   : 3186   Min.   : 10.00   No :699
1st Qu.: 6751   1st Qu.: 53.00   Yes: 78
Median : 8377   Median : 65.00
Mean   : 9660   Mean   : 65.46
3rd Qu.:10830   3rd Qu.: 78.00
Max.   :56233   Max.   :118.00
```

[79]:

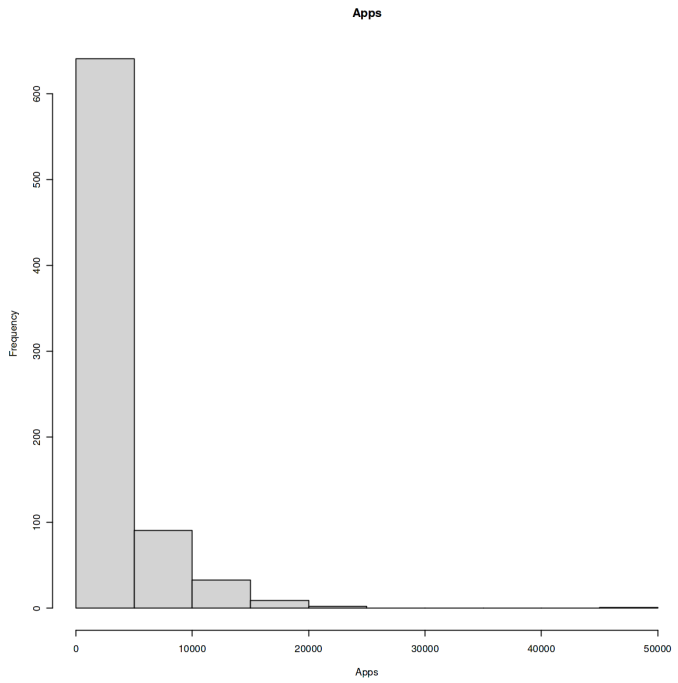
```
boxplot(college$Outstate ~ college$Elite,
       main = "Outstate vs Elite",
       xlab = "Outstate",
       ylab = "Elite",
       )
```

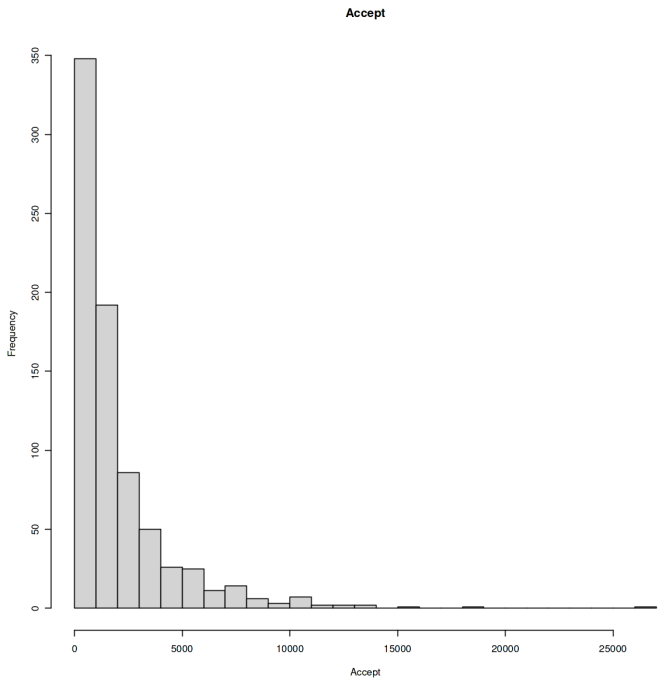


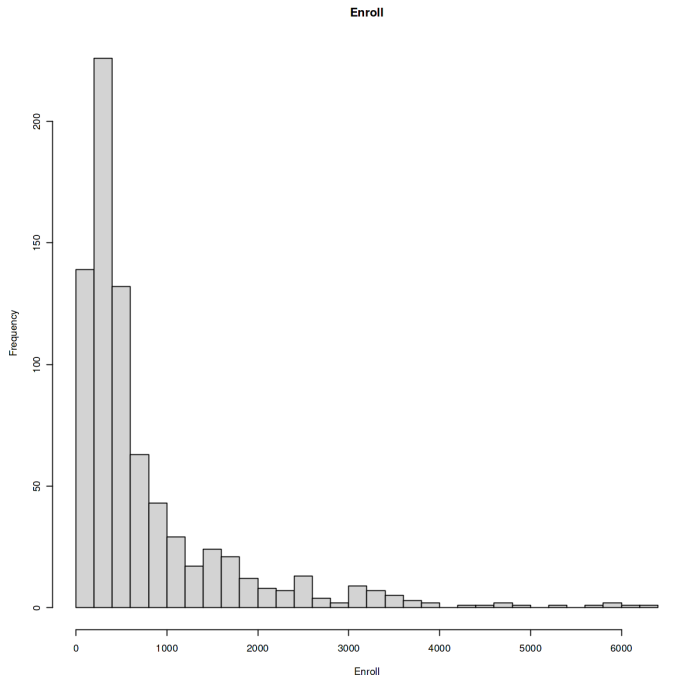
[80]:

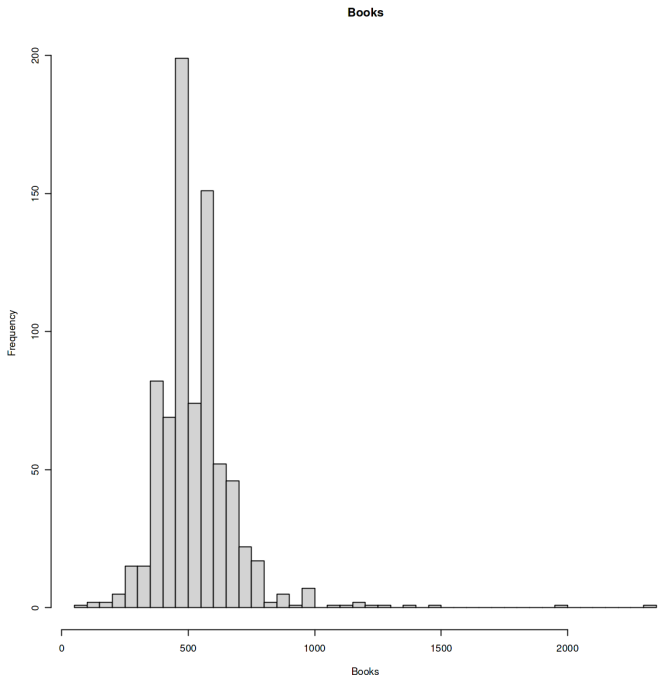
```
par(mfrow = c(2, 2))
hist(
    college$Apps,
    main = "Apps",
    xlab = "Apps",
    breaks = 10,
)
hist(
    college$Accept,
    main = "Accept",
    xlab = "Accept",
```

```
    breaks = 20,
)
hist(
    college$Enroll,
    main = "Enroll",
    xlab = "Enroll",
    breaks = 30,
)
hist(
    college$Books,
    main = "Books",
    xlab = "Books",
    breaks = 40,
)
```

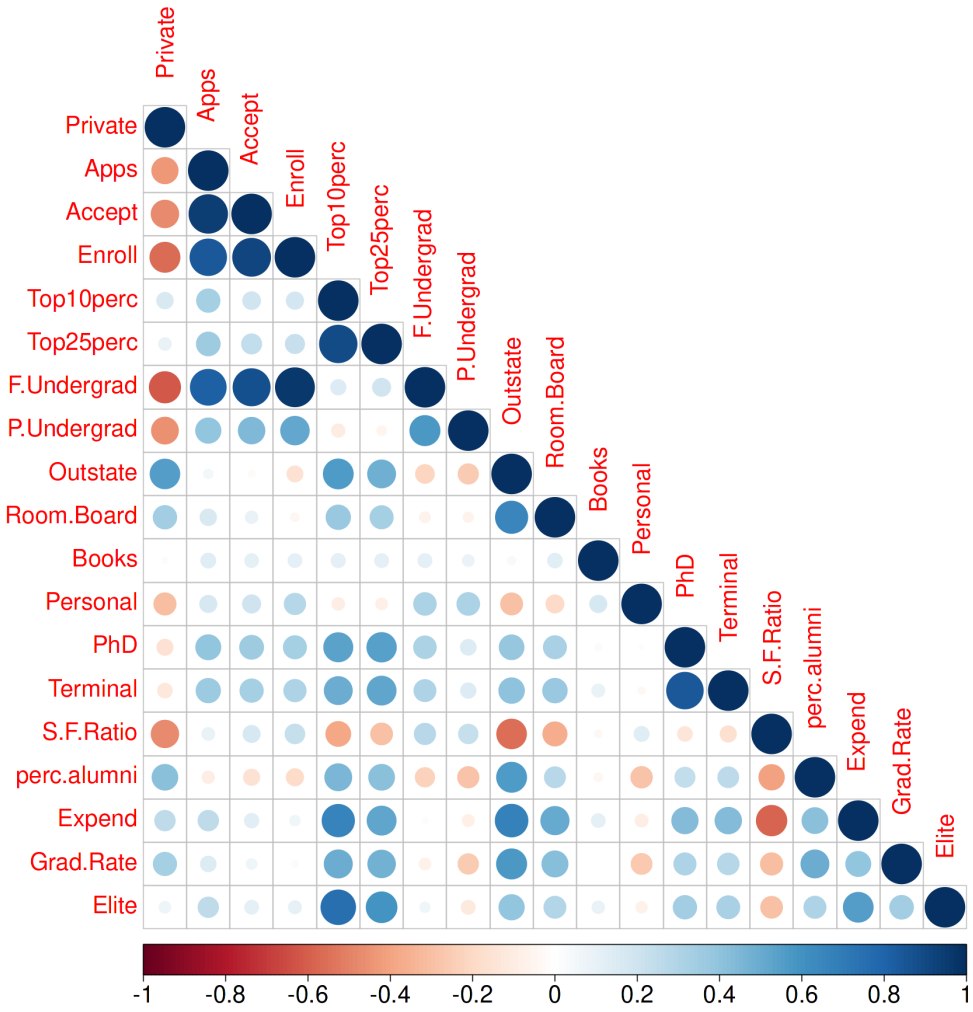








[81]:
```
college$Private = ifelse(college$Private == 'Yes', 1, 0)
college$Elite = ifelse(college$Elite == 'Yes', 1, 0)
correlation_matrix = cor(college)
my_corrplot(correlation_matrix)
```



This would be another good tool for understanding correlations.

Exercise 9 - The Auto Dataset

[82]:
```
auto = read.csv("/kaggle/input/cs5565-datasets/ALL CSV FILES - 2nd Edition/Auto.c
```

[83]:
```
auto = na.omit(auto)
```

[84]:
```
summary(auto)
```

```
      mpg          cylinders      displacement     horsepower       
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Length:397        
 1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.0   Class :character  
 Median :23.00   Median :4.000   Median :146.0   Mode  :character  
 Mean   :23.52   Mean   :5.458   Mean   :193.5                     
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0                     
 Max.   :46.60   Max.   :8.000   Max.   :455.0                     
     weight      acceleration        year           origin      
 Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000  
 1st Qu.:2223   1st Qu.:13.80   1st Qu.:73.00   1st Qu.:1.000  
 Median :2800   Median :15.50   Median :76.00   Median :1.000  
 Mean   :2970   Mean   :15.56   Mean   :75.99   Mean   :1.574  
 3rd Qu.:3609   3rd Qu.:17.10   3rd Qu.:79.00   3rd Qu.:2.000  
 Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000  
     name          
 Length:397        
 Class :character  
 Mode  :character  
```

[85]:
```
auto[!grepl("^\\d+$", auto$horsepower),]
```

A data.frame: 5 × 9

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
	<dbl>	<int>	<dbl>	<chr>	<int>	<dbl>	<int>	<int>	<chr>
33	25.0	4	98	?	2046	19.0	71	1	ford pinto
127	21.0	6	200	?	2875	17.0	74	1	ford maverick
331	40.9	4	85	?	1835	17.3	80	2	renault lecar deluxe
337	23.6	4	140	?	2905	14.3	80	1	ford mustang cobra
355	34.5	4	100	?	2320	15.8	81	2	renault 18i

I noticed horsepower was a "<chr>", and I thought that was odd. It appears question marks have been used for NA values. Let's fix that.

[86]:
```
auto[auto== "?"] = NA
auto = na.omit(auto)
```

```
auto$horsepower = as.numeric(auto$horsepower)
```

(a) Which of the predictors are quantitative, and which are qualitative?

The features origin, name, and year are qualitative. While year might appear to be quantitative, it isn't measuring anything. Depending on what I'm trying to predict or infer, I might also treat cylinders as a qualitative since it could be seen as different types of engines. All other predictors are clearly measured values and are quantitative.

(b) What is the range of each quantitative predictor? You can answer this using the range() function.

[87]:
```
quantitative_columns = c("mpg", "cylinders", "displacement", "horsepower", "weigh
find_range = function(auto) {
    range_df = data.frame()
    for (val in quantitative_columns) {
        range_val = range(auto[[val]])
        range_df = rbind(
            range_df,
            data.frame(
                Column = val,
                Minimum = range_val[1],
                Maximum = range_val[2],
                Range = range_val[2] - range_val[1]
            )
        )
    }
    return(range_df)
}
find_range(auto)
```

A data.frame: 6 × 4

Column	Minimum	Maximum	Range
<chr>	<dbl>	<dbl>	<dbl>
mpg	9	46.6	37.6
cylinders	3	8.0	5.0
displacement	68	455.0	387.0
horsepower	46	230.0	184.0
weight	1613	5140.0	3527.0
acceleration	8	24.8	16.8

(c) What is the mean and standard deviation of each quantitative predictor?

[88]:
```
find_mean_sd = function(auto) {
    mean_sd_df = data.frame()
    for (val in quantitative_columns) {
        mean_val = mean(auto[[val]])
        std_dev_val = sd(auto[[val]])
        mean_sd_df = rbind(
            mean_sd_df,
            data.frame(
                Column = val,
                Mean = mean_val,
                std_dev_val = std_dev_val
            )
        )
    }
    return(mean_sd_df)
}
find_mean_sd(auto)
```

A data.frame: 6 × 3

Column	Mean	std_dev_val
<chr>	<dbl>	<dbl>
mpg	23.445918	7.805007
cylinders	5.471939	1.705783
displacement	194.411990	104.644004
horsepower	104.469388	38.491160
weight	2977.584184	849.402560
acceleration	15.541327	2.758864

(d) Now remove the 10th through 85th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains?

[89]:
```
temp_auto = auto[-c(10:85)]
find_range(temp_auto)
find_mean_sd(temp_auto)
```

A data.frame: 6 × 4

Column	Minimum	Maximum	Range
<chr>	<dbl>	<dbl>	<dbl>
mpg	9	46.6	37.6
cylinders	3	8.0	5.0
displacement	68	455.0	387.0

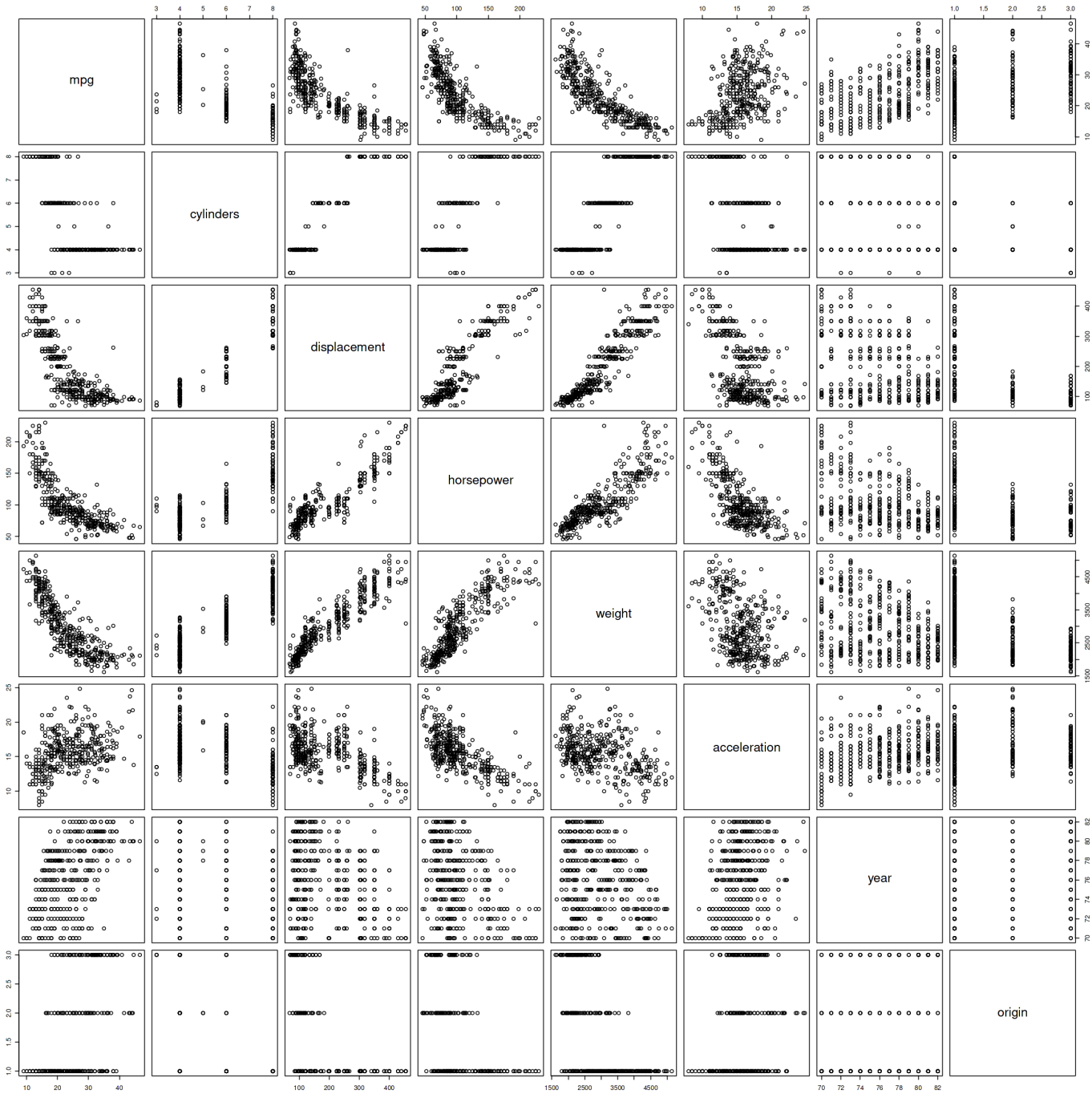
Column	Minimum	Maximum	Range
<chr>	<dbl>	<dbl>	<dbl>
horsepower	46	230.0	184.0
weight	1613	5140.0	3527.0
acceleration	8	24.8	16.8

A data.frame: 6 × 3

Column	Mean	std_dev_val
<chr>	<dbl>	<dbl>
mpg	23.445918	7.805007
cylinders	5.471939	1.705783
displacement	194.411990	104.644004
horsepower	104.469388	38.491160
weight	2977.584184	849.402560
acceleration	15.541327	2.758864

(e) Using the full data set, investigate the predictors graphically,using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

[90]:
```
temp_auto = subset(auto, select = -c(name))
pairs(temp_auto, cex.labels = 2)
```

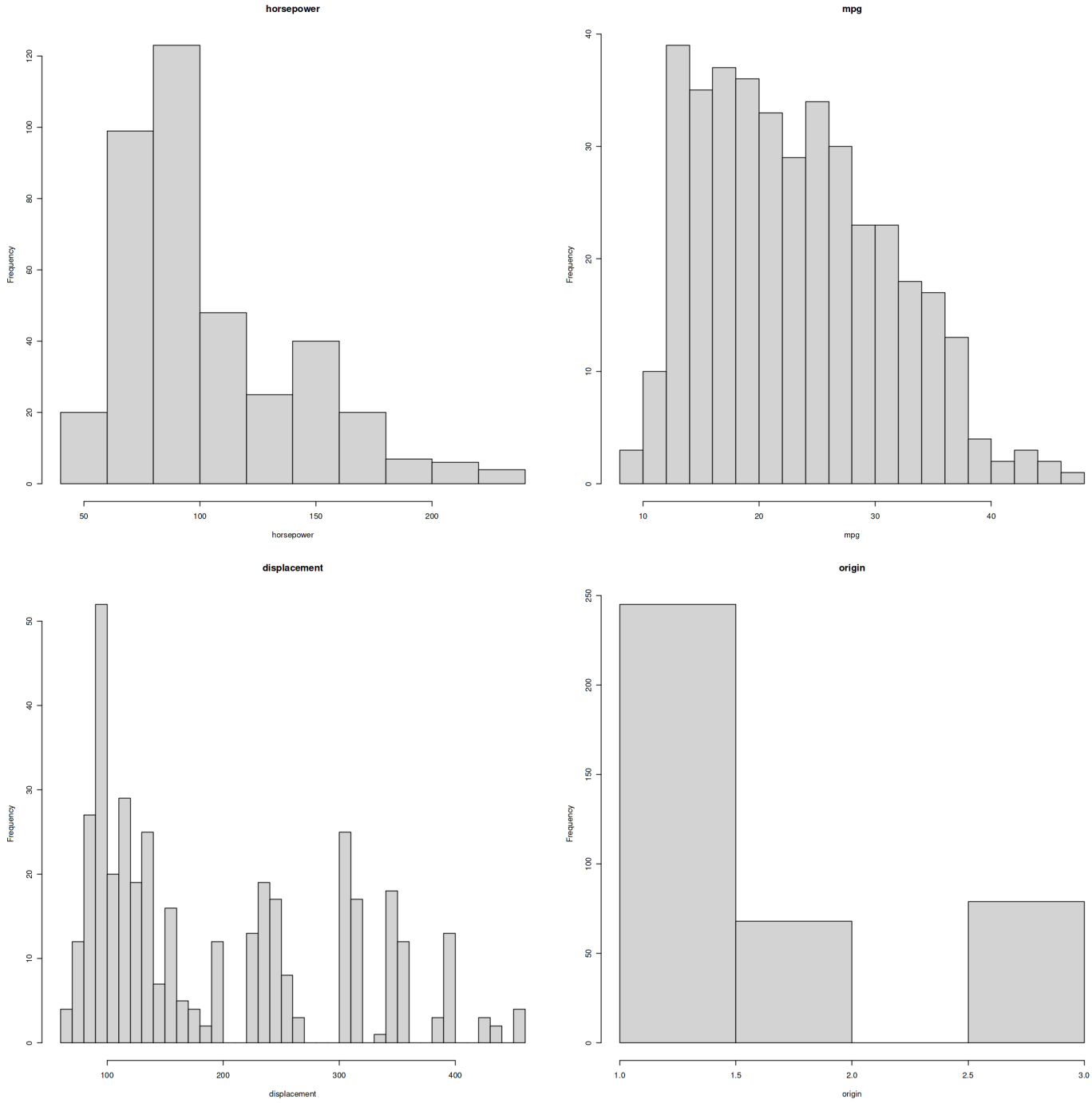


There appears to be a positive, linear relationship between the variables cylinders, weight, displacement, and horsepower. Conversely, miles per gallon (MPG) and acceleration show a negative correlation with these four features. Newer cars tend to be more fuel-efficient. While country of origin may seem to affect fuel economy, it's important to note that it also has a strong correlation with cylinders, weight, displacement, and horsepower.

[91]:

```
par(mfrow = c(2, 2))
hist(
    auto$horsepower,
    main = "horsepower",
    xlab = "horsepower",
```

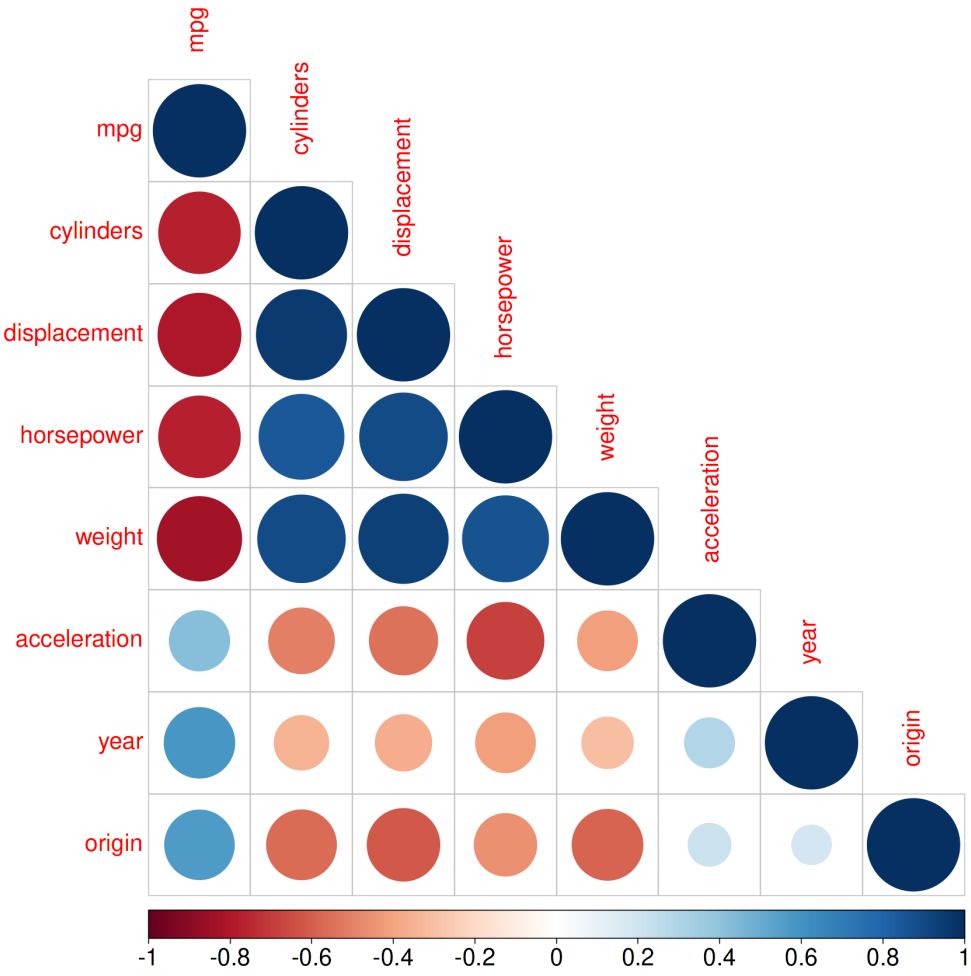
```
    breaks = 10,
)
hist(
    auto$mpg,
    main = "mpg",
    xlab = "mpg",
    breaks = 20,
)
hist(
    auto$displacement,
    main = "displacement",
    xlab = "displacement",
    breaks = 30,
)
hist(
    auto$origin,
    main = "origin",
    xlab = "origin",
    breaks = 3,
)
```



These histograms illustrate the distribution of the population across these domains. There are comparatively fewer vehicles with high displacement, high horsepower, and high miles per gallon (mpg). Moreover, domestic vehicles constitute the majority of this population.

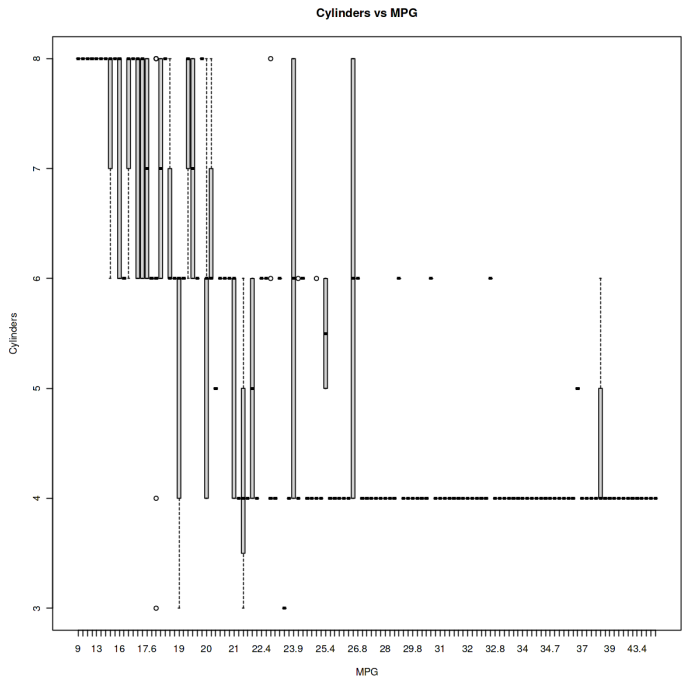
[92]:

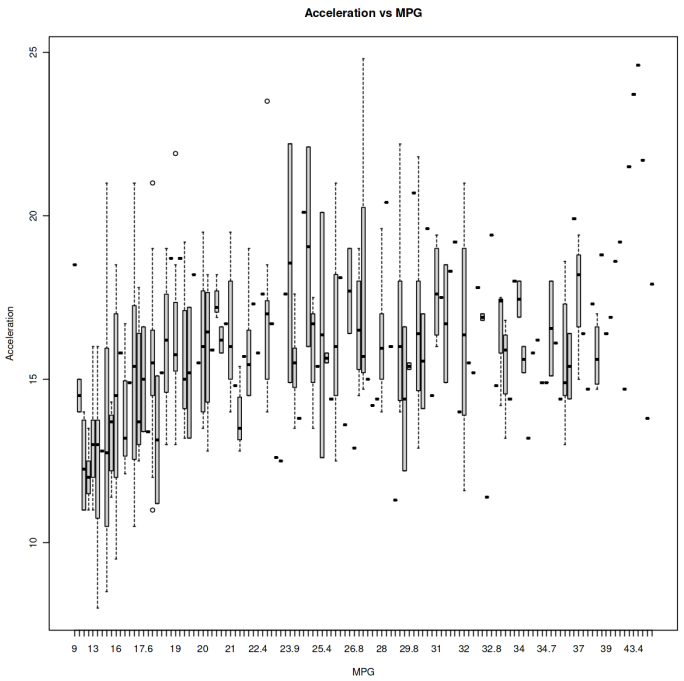
```
correlation_matrix = cor(temp_auto)
my_corrplot(correlation_matrix)
```

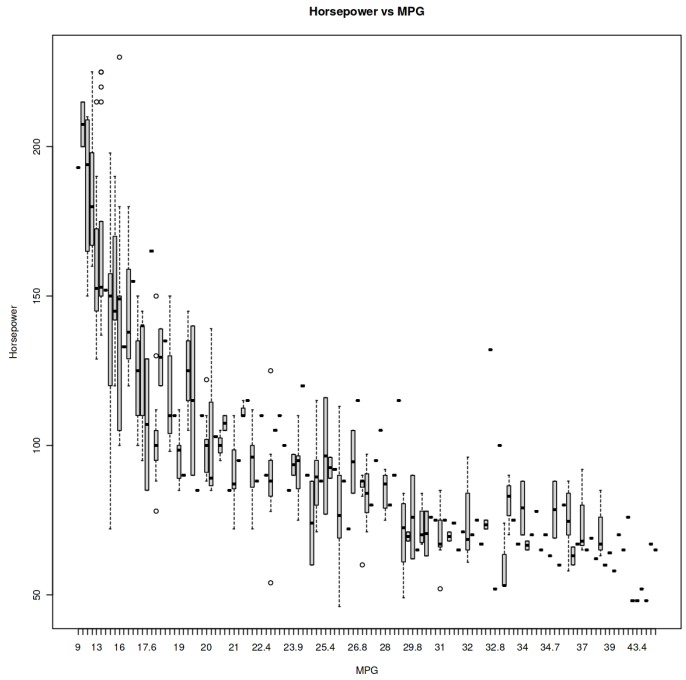


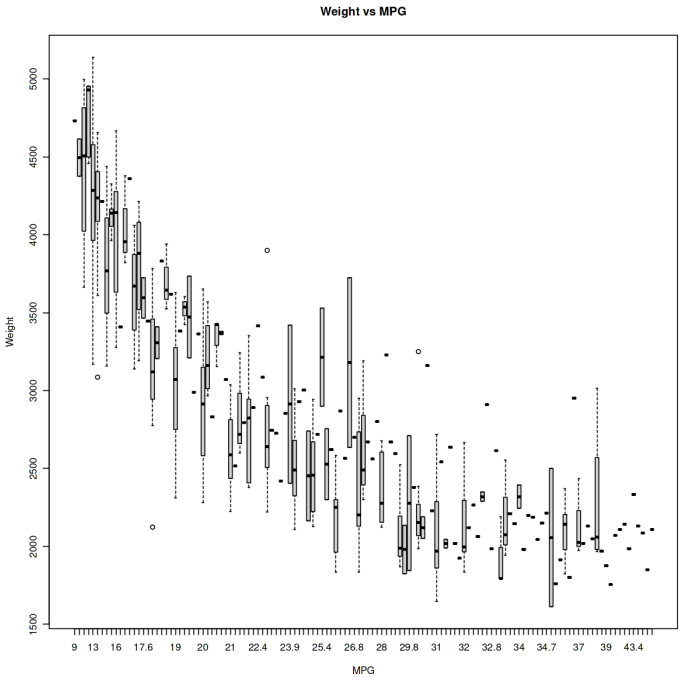
[93]:
```
par(mfrow = c(2, 2))
boxplot(auto$cylinders ~ auto$mpg,
        main = 'Cylinders vs MPG',
        ylab = "Cylinders",
        xlab = 'MPG',
       )
boxplot(auto$acceleration ~ auto$mpg,
        main = 'Acceleration vs MPG',
        ylab = "Acceleration",
        xlab = 'MPG',
       )
```

```
boxplot(auto$horsepower ~ auto$mpg,
        main = 'Horsepower vs MPG',
        ylab = "Horsepower",
        xlab = 'MPG',
       )
boxplot(auto$weight ~ auto$mpg,
        main = 'Weight vs MPG',
        ylab = "Weight",
        xlab = 'MPG',
       )
```









The boxplots above reinforce previous observations. Initially, I was taken aback by the correlation between acceleration and miles per gallon (mpg), but it made sense when I understood that acceleration is also negatively correlated with weight, cylinders, and displacement.

(f) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer

This dataset has many metrics that encapsulate similar data. Although not completely correlated, cylinders and displacement are essentially measuring the same attribute. In addition, origin likely doesn't contribute any useful information to miles per gallon (mpg), as it could potentially be a reflection of different types of vehicles being built in different countries. Similarly, acceleration is largely a product of weight, displacement, and horsepower. However, the 'year' variable deserves further investigation, even though it's negatively correlated with factors that influence low mpg.

Exercise 10 - The Boston Dataset

(a) To begin, load in the Boston data set.

[94]:
```
Boston
```

A data.frame: 506 × 13

	crime_rate	percent_25k_ft	percent_non-retail	On_River	percent_NO2	rooms_per_home	built_prior_1940	dist_to_e
	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	
1	0.00632	18.0	2.31	0	0.538	6.575	65.2	
2	0.02731	0.0	7.07	0	0.469	6.421	78.9	
3	0.02729	0.0	7.07	0	0.469	7.185	61.1	
4	0.03237	0.0	2.18	0	0.458	6.998	45.8	
5	0.06905	0.0	2.18	0	0.458	7.147	54.2	
6	0.02985	0.0	2.18	0	0.458	6.430	58.7	
7	0.08829	12.5	7.87	0	0.524	6.012	66.6	
8	0.14455	12.5	7.87	0	0.524	6.172	96.1	
9	0.21124	12.5	7.87	0	0.524	5.631	100.0	
10	0.17004	12.5	7.87	0	0.524	6.004	85.9	
11	0.22489	12.5	7.87	0	0.524	6.377	94.3	
12	0.11747	12.5	7.87	0	0.524	6.009	82.9	
13	0.09378	12.5	7.87	0	0.524	5.889	39.0	
14	0.62976	0.0	8.14	0	0.538	5.949	61.8	
15	0.63796	0.0	8.14	0	0.538	6.096	84.5	

	crime_rate	percent_25k_ft	percent_non-retail	On_River	percent_NO2	rooms_per_home	built_prior_1940	dist_to_e
	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	
16	0.62739	0.0	8.14	0	0.538	5.834	56.5	
17	1.05393	0.0	8.14	0	0.538	5.935	29.3	
18	0.78420	0.0	8.14	0	0.538	5.990	81.7	
19	0.80271	0.0	8.14	0	0.538	5.456	36.6	
20	0.72580	0.0	8.14	0	0.538	5.727	69.5	
21	1.25179	0.0	8.14	0	0.538	5.570	98.1	
22	0.85204	0.0	8.14	0	0.538	5.965	89.2	
23	1.23247	0.0	8.14	0	0.538	6.142	91.7	
24	0.98843	0.0	8.14	0	0.538	5.813	100.0	
25	0.75026	0.0	8.14	0	0.538	5.924	94.1	
26	0.84054	0.0	8.14	0	0.538	5.599	85.7	
27	0.67191	0.0	8.14	0	0.538	5.813	90.3	
28	0.95577	0.0	8.14	0	0.538	6.047	88.8	
29	0.77299	0.0	8.14	0	0.538	6.495	94.4	
30	1.00245	0.0	8.14	0	0.538	6.674	87.3	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
477	4.87141	0	18.10	0	0.614	6.484	93.6	
478	15.02340	0	18.10	0	0.614	5.304	97.3	
479	10.23300	0	18.10	0	0.614	6.185	96.7	
480	14.33370	0	18.10	0	0.614	6.229	88.0	
481	5.82401	0	18.10	0	0.532	6.242	64.7	
482	5.70818	0	18.10	0	0.532	6.750	74.9	
483	5.73116	0	18.10	0	0.532	7.061	77.0	
484	2.81838	0	18.10	0	0.532	5.762	40.3	
485	2.37857	0	18.10	0	0.583	5.871	41.9	
486	3.67367	0	18.10	0	0.583	6.312	51.9	
487	5.69175	0	18.10	0	0.583	6.114	79.8	
488	4.83567	0	18.10	0	0.583	5.905	53.2	
489	0.15086	0	27.74	0	0.609	5.454	92.7	
490	0.18337	0	27.74	0	0.609	5.414	98.3	
491	0.20746	0	27.74	0	0.609	5.093	98.0	
492	0.10574	0	27.74	0	0.609	5.983	98.8	
493	0.11132	0	27.74	0	0.609	5.983	83.5	

	crime_rate	percent_25k_ft	percent_non-retail	On_River	percent_NO2	rooms_per_home	built_prior_1940	dist_to_e
	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	
494	0.17331	0	9.69	0	0.585	5.707	54.0	
495	0.27957	0	9.69	0	0.585	5.926	42.6	
496	0.17899	0	9.69	0	0.585	5.670	28.8	
497	0.28960	0	9.69	0	0.585	5.390	72.9	
498	0.26838	0	9.69	0	0.585	5.794	70.6	
499	0.23912	0	9.69	0	0.585	6.019	65.3	
500	0.17783	0	9.69	0	0.585	5.569	73.5	
501	0.22438	0	9.69	0	0.585	6.027	79.7	
502	0.06263	0	11.93	0	0.573	6.593	69.1	
503	0.04527	0	11.93	0	0.573	6.120	76.7	
504	0.06076	0	11.93	0	0.573	6.976	91.0	
505	0.10959	0	11.93	0	0.573	6.794	89.3	
506	0.04741	0	11.93	0	0.573	6.030	80.8	

[95]:
```
?Boston
```

Boston {ISLR2} R Documentation

Boston Data
Description
A data set containing housing values in 506 suburbs of Boston.

Usage
Boston
Format
A data frame with 506 rows and 13 variables.

crim	per capita crime rate by town.
zn	proportion of residential land zoned for lots over 25,000 sq.ft.
indus	proportion of non-retail business acres per town.
chas	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
nox	nitrogen oxides concentration (parts per 10 million).
rm	average number of rooms per dwelling.
age	proportion of owner-occupied units built prior to 1940.

dis	weighted mean of distances to five Boston employment centres.
rad	index of accessibility to radial highways.
tax	full-value property-tax rate per $10,000.
ptratio	pupil-teacher ratio by town.
lstat	lower status of the population (percent).
medv	median value of owner-occupied homes in $1000s.

Source
This dataset was obtained from, and is slightly modified from, the Boston dataset that is part of the MASS library. References are available in the MASS library.

References
James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013) *An Introduction to Statistical Learning with applications in R*, https://www.statlearning.com, Springer-Verlag, New York

Examples
lm(medv ~ crim + rm, data=Boston)

[Package *ISLR2* version 1.3-2]

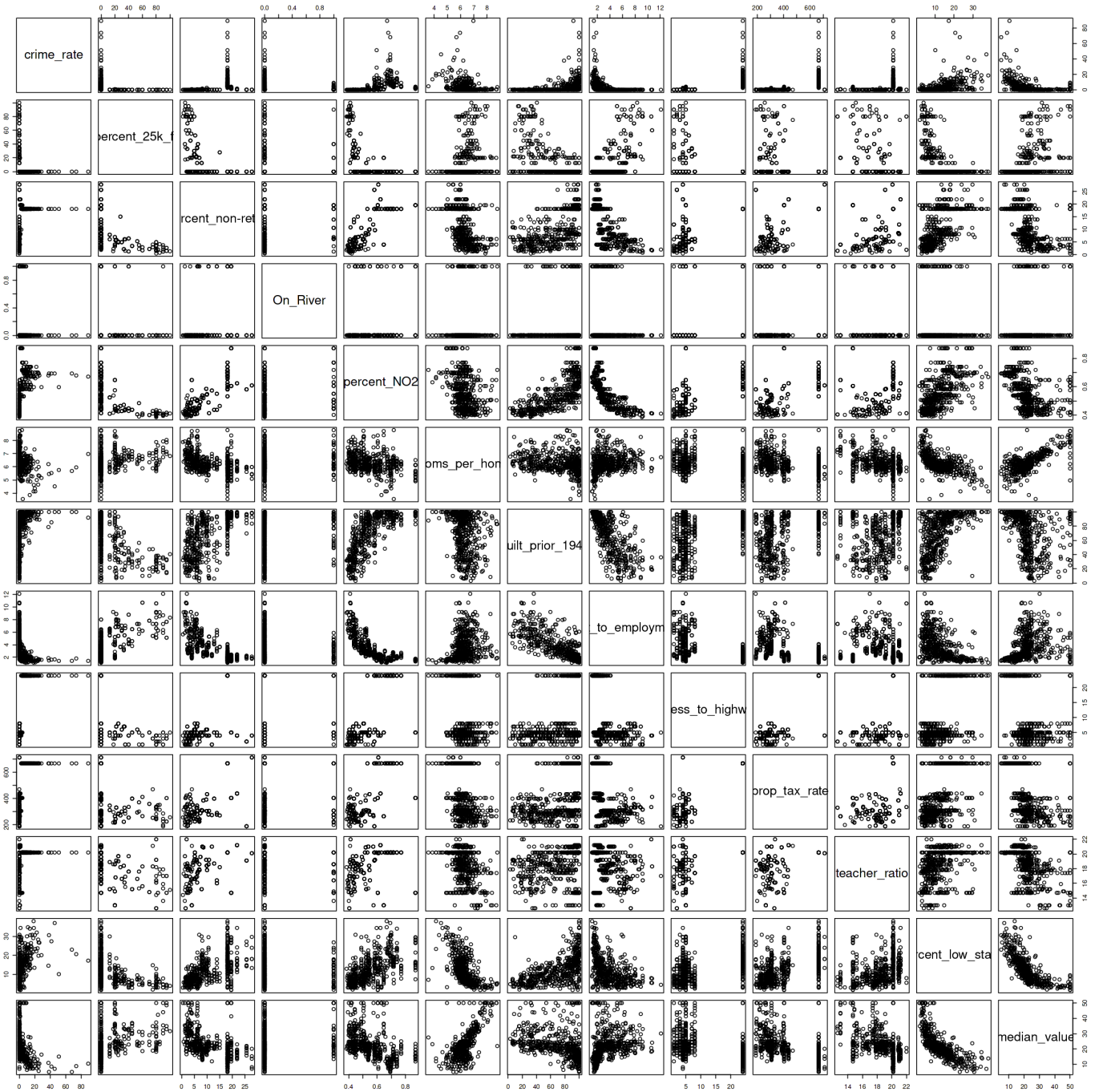
[96]:
```
# assuming your original data is 'Boston' data frame
names(Boston) <- c(
    "crime_rate",
    "percent_25k_ft",
    "percent_non-retail",
    "On_River",
    "percent_NO2",
    "rooms_per_home",
    "built_prior_1940",
    "dist_to_employment",
    "access_to_highways",
    "prop_tax_rate",
    "teacher_ratio",
    "percent_low_status",
    "median_value"
)
```

How many rows are in this data set? How many columns? What do the rows and columns represent?

This dataset contains 506 rows with 13 columns. Each row represents a suburb in the Boston metropolitan area, and each column represents a specific statistic about that suburb.

(b) Make some pairwise scatterplots of the predictors (columns) in this data set. Describe your findings.

[97]:
```
pairs(Boston, cex.labels = 2.0)
```



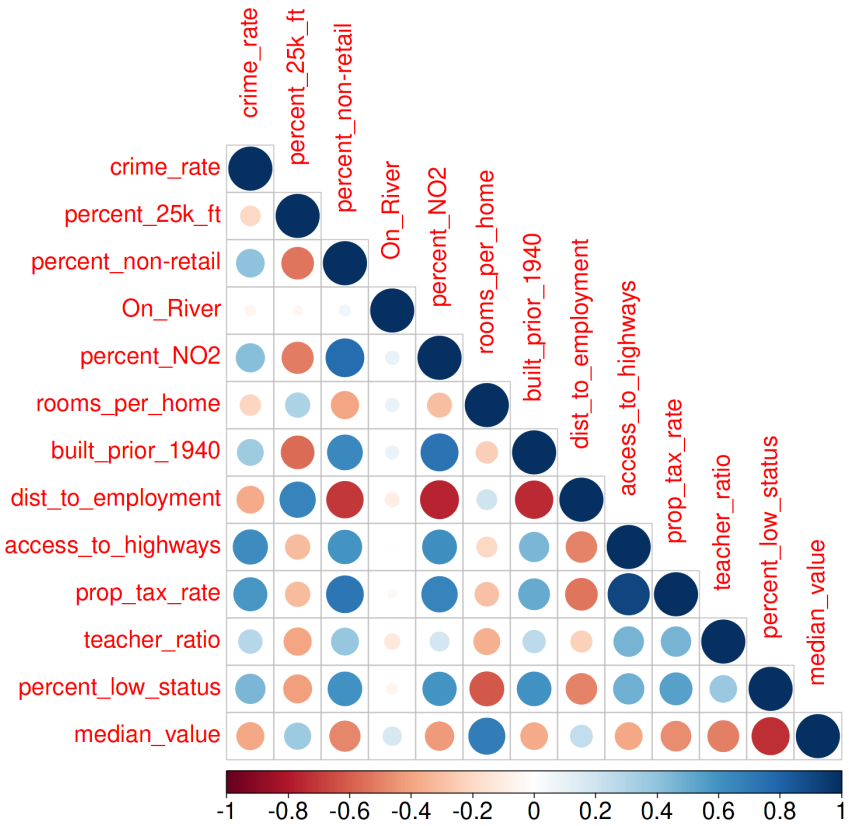
Several notable correlations stand out. The weighted mean distance to employment centers is highly correlated with most other variables, including NO2 emissions, the percentage of non-retail zoning, the percentage of low-status population, and the number of homes built prior to 1940. Property tax rates and access to highways also show a significant correlation, but this appears to be influenced by outliers, potentially a suburb with exceptionally high rates and access to radial highways. Finally, the crime rate data reveals a significant proportion of suburbs with low crime rates, contrasted by a relatively few number with high crime rates.

(c) Are any of the predictors associated with per capita crime rate? If so, explain the relationship.

[98]:
```
correlation_matrix = cor(Boston)
new_df = as.data.frame(correlation_matrix[,"crime_rate"])
colnames(new_df) = "crime_rate"
sorted_df = new_df[order(new_df["crime_rate"], decreasing = TRUE), , drop = FALSE
sorted_df
my_corrplot(correlation_matrix)
```

A data.frame: 13 × 1

	crime_rate
	<dbl>
crime_rate	1.00000000
access_to_highways	0.62550515
prop_tax_rate	0.58276431
percent_low_status	0.45562148
percent_NO2	0.42097171
percent_non-retail	0.40658341
built_prior_1940	0.35273425
teacher_ratio	0.28994558
On_River	-0.05589158
percent_25k_ft	-0.20046922
rooms_per_home	-0.21924670
dist_to_employment	-0.37967009
median_value	-0.38830461



[99]:

```
model <- lm(crime_rate ~ ., data = Boston)
summary(model)
```

Call:
lm(formula = crime_rate ~ ., data = Boston)

Residuals:
   Min   1Q Median   3Q   Max
-8.534 -2.248 -0.348 1.087 73.923

Coefficients:

```
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         13.7783938  7.0818258   1.946 0.052271 .
percent_25k_ft       0.0457100  0.0187903   2.433 0.015344 *
`percent_non-retail` -0.0583501  0.0836351  -0.698 0.485709
On_River            -0.8253776  1.1833963  -0.697 0.485841
percent_NO2         -9.9575865  5.2898242  -1.882 0.060370 .
rooms_per_home       0.6289107  0.6070924   1.036 0.300738
built_prior_1940    -0.0008483  0.0179482  -0.047 0.962323
dist_to_employment  -1.0122467  0.2824676  -3.584 0.000373 ***
access_to_highways   0.6124653  0.0875358   6.997 8.59e-12 ***
prop_tax_rate       -0.0037756  0.0051723  -0.730 0.465757
teacher_ratio       -0.3040728  0.1863598  -1.632 0.103393
percent_low_status   0.1388006  0.0757213   1.833 0.067398 .
median_value        -0.2200564  0.0598240  -3.678 0.000261 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.46 on 493 degrees of freedom
Multiple R-squared:  0.4493,    Adjusted R-squared:  0.4359
F-statistic: 33.52 on 12 and 493 DF,  p-value: < 2.2e-16
```

The dataframe and the model above reveal several strong correlations with high certainty. Leading the group are the percentage of nitrous oxides, distance to employment centers, access to radial highways, and number of rooms per dwelling. Nitrous oxides and access to highways exhibit strong positive correlations, while distance to employment centers and the number of rooms per dwelling are strongly negatively correlated.

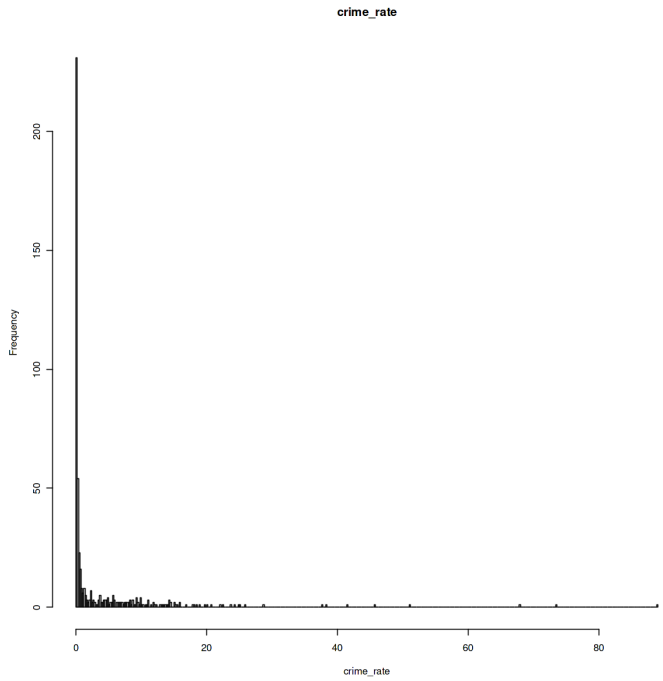
d) Do any of the census tracts of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher ratios? Comment on the range of each predictor.

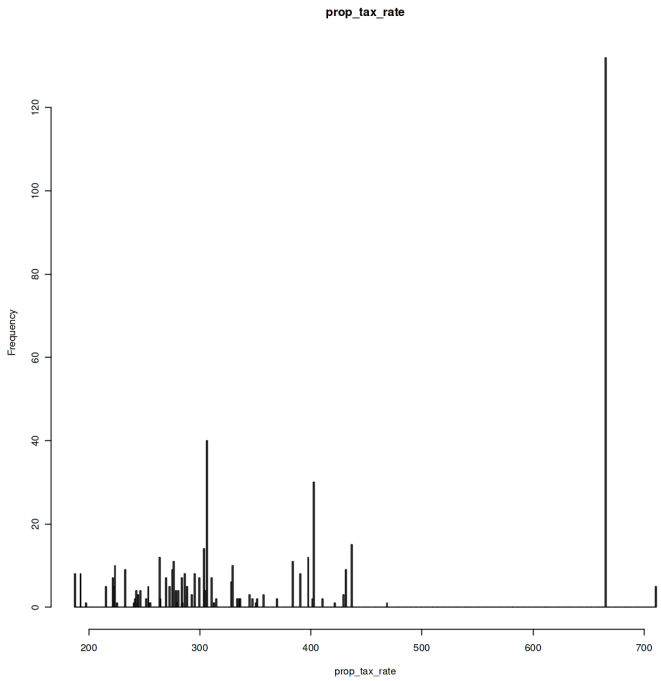
[100]:

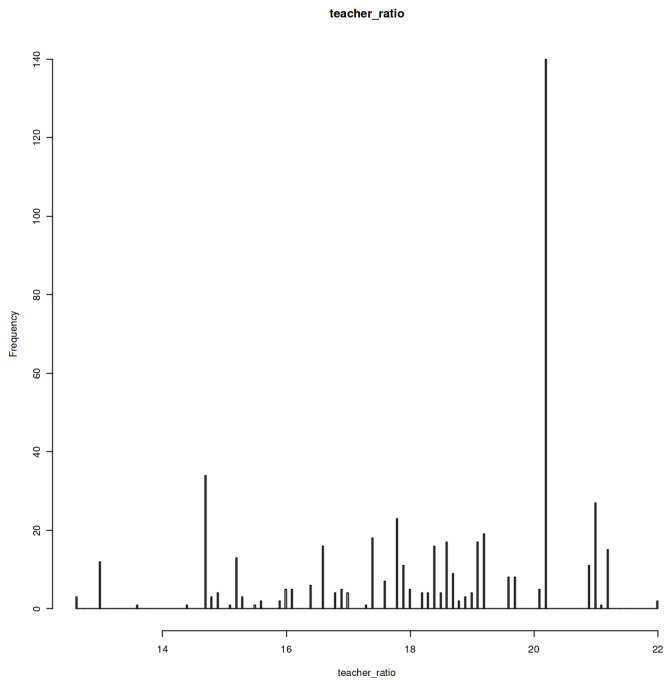
```
summary(Boston[c('crime_rate', 'prop_tax_rate', 'teacher_ratio')])
par(mfrow = c(2, 2))
hist(
    Boston$'crime_rate',
    main = "crime_rate",
    xlab = "crime_rate",
    breaks = length(Boston$'crime_rate'),
)
hist(
    Boston$'prop_tax_rate',
    main = "prop_tax_rate",
    xlab = "prop_tax_rate",
    breaks = length(Boston$'prop_tax_rate'),
)
hist(
    Boston$'teacher_ratio',
    main = "teacher_ratio",
```

```
      xlab = "teacher_ratio",
      breaks = length(Boston$'teacher_ratio'),
  )
```

```
  crime_rate       prop_tax_rate   teacher_ratio
Min.   : 0.00632   Min.   :187.0   Min.   :12.60
1st Qu.: 0.08205   1st Qu.:279.0   1st Qu.:17.40
Median : 0.25651   Median :330.0   Median :19.05
Mean   : 3.61352   Mean   :408.2   Mean   :18.46
3rd Qu.: 3.67708   3rd Qu.:666.0   3rd Qu.:20.20
Max.   :88.97620   Max.   :711.0   Max.   :22.00
```







The range of crime rates is likely the most striking statistic in this dataset. The vast majority of suburbs experience crime rates under ten percent. The third quartile is just under four percent. However, a few outliers

greatly skew the data, with the maximum exceeding 80 percent. This is further evidenced by a significant discrepancy between the median of 0.25% and the mean, 3.61%.

Property tax is another feature that appears skewed. This isn't reflected in a difference between the median and mean, but there's a clearly high mode. Speculatively, perhaps a large number of these suburbs are under one high-tax structure, while some suburbs can establish their own, lower-tax structures.

Teacher-pupil ratios are more uniformly distributed. There seems to be a relatively even spread between 18 and 21. However, there are some intriguing aspects. There's a small concentration near 12.5 that would be interesting to compare with the percentage of low-income households, median house values, and crime rate. Additionally, the concentration just above 20 might be influenced by a bureaucratic organization, such as a large school district that spans many suburbs, and thus, reports the same ratio for all the suburbs.

(e) How many of the census tracts in this data set bound the Charles River?

[101]:
```
cat("There are", sum(Boston$On_River == 1), "tracts on the Charles River.\n")
```

There are 35 tracts on the Charles River.

(f) What is the median pupil-teacher ratio among the towns in this data set?

[102]:
```
cat("The median pupil-teacher ratio among the towns in this dataset is", median(
```

The median pupil-teacher ratio among the towns in this dataset is 19.05

(g) Which census tract of Boston has lowest median value of owneroccupied homes? What are the values of the other predictors for that census tract, and how do those values compare to the overall ranges for those predictors? Comment on your findings

[103]:
```
cat("The census tract with the lowest median_value is:", which.min(Boston$median
```

The census tract with the lowest median_value is: 399

The relationship of Tract 399 with the overall ranges is striking. The crime rate is over 12 times the mean, the percentage of nitrous oxides is in the upper 25%, all the homes were built prior to 1940, and the access to highways is the highest. Additionally, both property tax rates and the percentage of low-status residents are exceptionally high.

(h) In this data set, how many of the census tracts average more than seven rooms per dwelling? More than eight rooms per dwelling? Comment on the census tracts that average more than eight rooms per dwelling.

[104]:
```
cat("There are", sum(Boston$rooms_per_home > 7), "tracts with greater than 7 roo
cat("There are", sum(Boston$rooms_per_home > 8), "tracts with greater than 8 roo
big_houses = Boston[Boston$rooms_per_home > 8, ]
```

There are 64 tracts with greater than 7 rooms per dwelling
There are 13 tracts with greater than 8 rooms per dwelling

[105]:
```
summary_big_houses = summary(big_houses)
summary_whole_dataset = summary(Boston)
summary_big_houses
summary_whole_dataset
```

```
  crime_rate      percent_25k_ft  percent_non-retail    On_River
 Min.   :0.02009   Min.   : 0.00   Min.   : 2.680     Min.   :0.0000
 1st Qu.:0.33147   1st Qu.: 0.00   1st Qu.: 3.970     1st Qu.:0.0000
 Median :0.52014   Median : 0.00   Median : 6.200     Median :0.0000
 Mean   :0.71879   Mean   :13.62   Mean   : 7.078     Mean   :0.1538
 3rd Qu.:0.57834   3rd Qu.:20.00   3rd Qu.: 6.200     3rd Qu.:0.0000
 Max.   :3.47428   Max.   :95.00   Max.   :19.580     Max.   :1.0000
  percent_NO2     rooms_per_home  built_prior_1940 dist_to_employment
 Min.   :0.4161   Min.   :8.034   Min.   : 8.40     Min.   :1.801
 1st Qu.:0.5040   1st Qu.:8.247   1st Qu.:70.40     1st Qu.:2.288
 Median :0.5070   Median :8.297   Median :78.30     Median :2.894
 Mean   :0.5392   Mean   :8.349   Mean   :71.54     Mean   :3.430
 3rd Qu.:0.6050   3rd Qu.:8.398   3rd Qu.:86.50     3rd Qu.:3.652
 Max.   :0.7180   Max.   :8.780   Max.   :93.90     Max.   :8.907
 access_to_highways prop_tax_rate   teacher_ratio   percent_low_status
 Min.   : 2.000     Min.   :224.0   Min.   :13.00   Min.   :2.47
 1st Qu.: 5.000     1st Qu.:264.0   1st Qu.:14.70   1st Qu.:3.32
 Median : 7.000     Median :307.0   Median :17.40   Median :4.14
 Mean   : 7.462     Mean   :325.1   Mean   :16.36   Mean   :4.31
 3rd Qu.: 8.000     3rd Qu.:307.0   3rd Qu.:17.40   3rd Qu.:5.12
 Max.   :24.000     Max.   :666.0   Max.   :20.20   Max.   :7.44
  median_value
 Min.   :21.9
 1st Qu.:41.7
 Median :48.3
 Mean   :44.2
 3rd Qu.:50.0
 Max.   :50.0
  crime_rate       percent_25k_ft   percent_non-retail    On_River
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46      Min.   :0.00000
 1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19      1st Qu.:0.00000
 Median : 0.25651   Median :  0.00   Median : 9.69      Median :0.00000
 Mean   : 3.61352   Mean   : 11.36   Mean   :11.14      Mean   :0.06917
 3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10      3rd Qu.:0.00000
 Max.   :88.97620   Max.   :100.00   Max.   :27.74      Max.   :1.00000
  percent_NO2     rooms_per_home  built_prior_1940 dist_to_employment
```

```
Min.   :0.3850   Min.   :3.561   Min.   : 2.90   Min.   : 1.130
1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
access_to_highways prop_tax_rate   teacher_ratio   percent_low_status
Min.   : 1.000     Min.   :187.0   Min.   :12.60   Min.   : 1.73
1st Qu.: 4.000     1st Qu.:279.0   1st Qu.:17.40   1st Qu.: 6.95
Median : 5.000     Median :330.0   Median :19.05   Median :11.36
Mean   : 9.549     Mean   :408.2   Mean   :18.46   Mean   :12.65
3rd Qu.:24.000     3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:16.95
Max.   :24.000     Max.   :711.0   Max.   :22.00   Max.   :37.97
 median_value
Min.   : 5.00
1st Qu.:17.02
Median :21.20
Mean   :22.53
3rd Qu.:25.00
Max.   :50.00
```

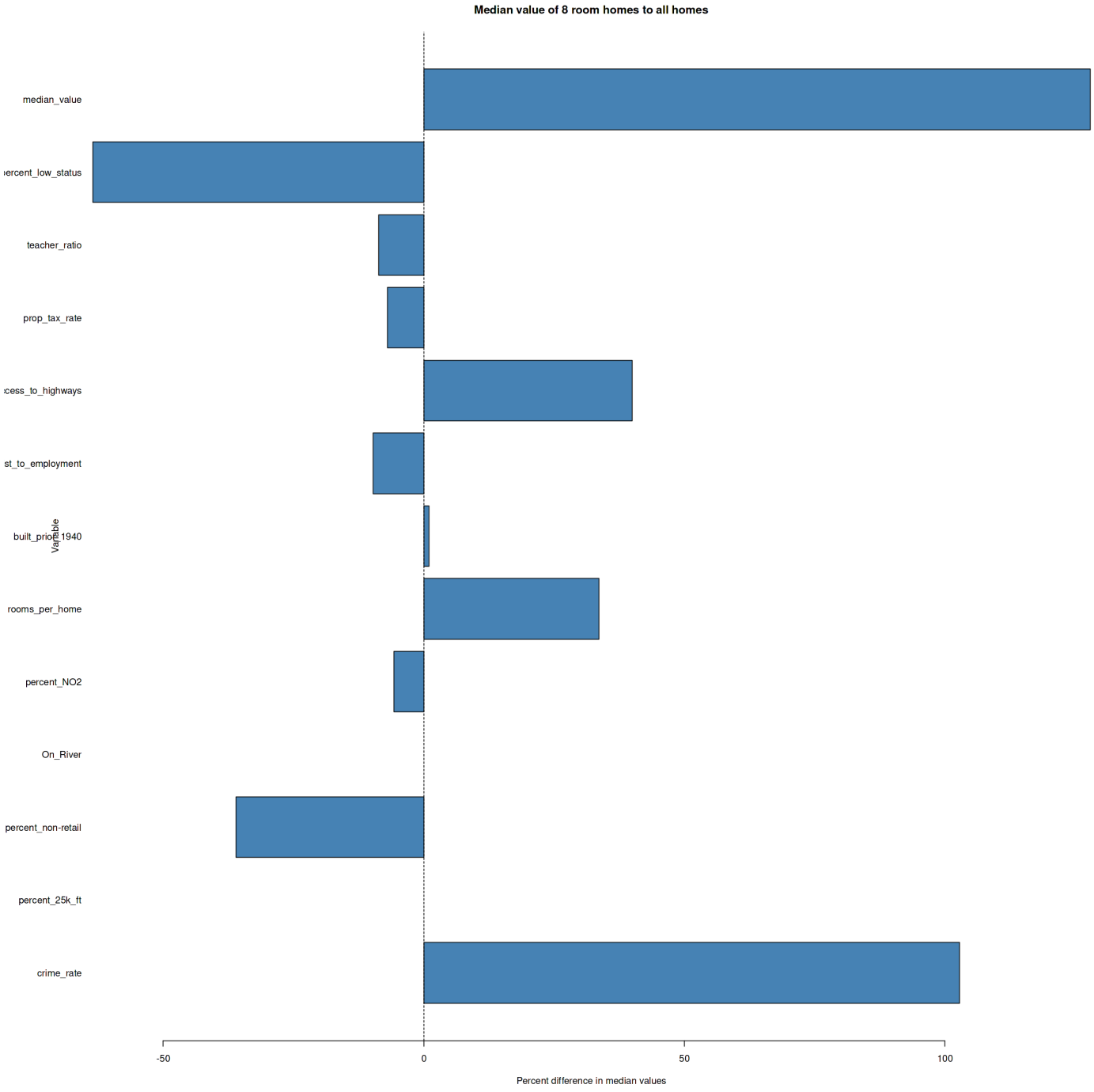
[106]:

```
graph_df = data.frame()
for (column_name in names(Boston)) {
  if (is.numeric(Boston[[column_name]])) {
    median_difference =  median(big_houses[[column_name]]) - median(Boston[[colu
    percentage_difference = (median_difference / median(Boston[[column_name]]))
    graph_df[1, column_name] = percentage_difference
  }
}
View(graph_df)
graph_vec <- unlist(graph_df)
names(graph_vec) <- colnames(graph_df)
par(mar = c(5, 8, 4, 2))
barplot(
    height = graph_vec,
    main = "Median value of 8 room homes to all homes",
    xlab = "Percent difference in median values",
    ylab = "Variable",
    horiz = TRUE,
    las = 1,
    col = "steelblue"
)
abline(v = 0, lty = 2)
```

A data.frame: 1 × 13

crime_rate	percent_25k_ft	percent_non-retail	On_River	percent_NO2	rooms_per_home	built_prior_1940	dist_to_em
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	

	crime_rate	percent_25k_ft	percent_non-retail	On_River	percent_NO2	rooms_per_home	built_prior_1940	dist_to_em
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	102.7757	NaN	-36.01651	NaN	-5.762082	33.63937	1.032258	



This has been one of the most surprising observations in the dataset. Homes with more than 8 rooms per dwelling have much higher median values and better access to highways. They have a very low percentage of low-status residents and large non-retail spaces. However, the crime rate is double the median of the overall dataset. This would be an interesting topic to explore. One hypothesis could be the inclusion of minor crimes in the dataset, combined with increased policing in affluent areas.

3.7 Exercises - Conceptual

1. Describe the null hypotheses to which the p-values given in Table 3.4 correspond. Explain what conclusions you can draw based on these p-values. Your explanation should be phrased in terms of sales, TV, radio, and newspaper, rather than in terms of the coefficients of the linear model.

Table 3.4

[107]:
```
Table_3_4 = data.frame(
    'Coefficient' = c(2.939, 0.046, 0.189, -0.001),
    'Std. error' = c(0.3119, 0.0014, 0.0086, 0.0059),
    't-statistic' = c(9.42, 32.81, 21.89, -0.18),
    'p-value' = c('<0.0001', '<0.0001', '<0.0001', '0.8599')
)
row.names(Table_3_4) = c('Intercept', 'TV', 'radio', 'newspaper')
View(Table_3_4)
```

A data.frame: 4 × 4

	Coefficient	Std..error	t.statistic	p.value
	<dbl>	<dbl>	<dbl>	<chr>
Intercept	2.939	0.3119	9.42	<0.0001
TV	0.046	0.0014	32.81	<0.0001
radio	0.189	0.0086	21.89	<0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

The null hypothesis for the intercept suggests that if all spending on TV, radio and newspaper marketing was zero, then sales would also be zero. The null hypothesis for TV, radio and newspapers suggests that spending in these categories does not have an effect on sales.

Based upon the p-values in the table, it seems that Intercept, TV, and radio have low values and the null hypothesis can be rejected. For TV and radio, this means that spending in these categories has a significant correlation with sales. For the intercept, this means that even if all marketing budgets were set to zero, sales would not be zero.

However, the p-value for the newspaper is significantly above the normal threshold of 0.05. This suggests it would not be appropriate to reject the null hypothesis, and it would be reasonable to assume that money spent on newspaper advertising does not have a significant impact on sales.

3. Suppose we have a data set with five predictors, X1 = GPA, X2 = IQ, X3 = Level (1 for College and 0 for High School), X4 = Interaction between GPA and IQ, and X5 = Interaction between GPA and Level. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to fit the model, and get $\hat{\beta}0 = 50$, $\hat{\beta}1 = 20$, $\hat{\beta}2 = 0.07$, $\hat{\beta}3 = 35$, $\hat{\beta}4 = 0.01$, $\hat{\beta}5 = -10$.
(a) Which answer is correct, and why?
i. For a fixed value of IQ and GPA, high school graduates earn more, on average, than college graduates.
ii. For a fixed value of IQ and GPA, college graduates earn more, on average, than high school graduates.

iii. For a fixed value of IQ and GPA, high school graduates earn more, on average, than college graduates provided that the GPA is high enough.
iv. For a fixed value of IQ and GPA, college graduates earn more, on average, than high school graduates provided that the GPA is high enough.
The correct answer is iv. The Level generally increases salary. However, the interaction between GPA and Level being negative indicates that GPA has a greater impact on salary for high school graduates than college graduates. This suggests that the advantage conferred by a college degree decreases as GPA increases. Therefore, in some cases, a high-GPA high school graduate would earn more than a low-GPA college graduate.

(b) Predict the salary of a college graduate with IQ of 110 and a GPA of 4.0.
Salary = βˆ0 + βˆ1 * GPA + βˆ2 * IQ + βˆ3 * Level + βˆ4 * GPA * IQ + βˆ5 * GPA * Level = 50 + 20 * 4.0 + 0.07 * 110 + 35 * 1 + 0.01 * 4.0 * 110 - 10 * 4.0 * 1 = $137,100

(c) True or false: Since the coefficient for the GPA/IQ interaction term is very small, there is very little evidence of an interaction effect. Justify your answer.
False. The size of the coefficient is not an indicator of the interaction effect. We would need data concerning the standard error of the coefficient or p-values to justify a null hypothesis.

Chapter 3.7 Exercises - Applied

8. This question involves the use of simple linear regression on the Auto data set.

(a) Use the lm() function to perform a simple linear regression with mpg as the response and horsepower as the predictor. Use the summary() function to print the results. Comment on the output.

[108]:
```
model = lm(mpg ~ horsepower, data = Auto)
summary_fit = summary(model)
summary_fit
```

```
Call:
lm(formula = mpg ~ horsepower, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max 
-13.5710  -3.2592  -0.3435   2.7630  16.9240 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 39.935861   0.717499   55.66   <2e-16 ***
horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.906 on 390 degrees of freedom
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.6049 
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

i. Is there a relationship between the predictor and the response?

[109]:
```
summary_fit$coefficients
```

A matrix: 2 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	39.9358610	0.717498656	55.65984	1.220362e-187
horsepower	-0.1578447	0.006445501	-24.48914	7.031989e-81

There appears to be a strong relationship between horsepower and mpg with high certainty. While it hardly can be used solely to predict mpg, it should certainly be kept in a predictive model.

ii. How strong is the relationship between the predictor and the response?

[110]:
```
cat("The relationship between the predictor and the response is very high.  The
```

The relationship between the predictor and the response is very high. The R-squared is 0.6059483

iii. Is the relationship between the predictor and the response positive or negative?

[111]:
```
cat("The relationship between the predictor and the response is negative. Coeffi
```

The relationship between the predictor and the response is negative. Coefficient for horsepower: -0.1578447

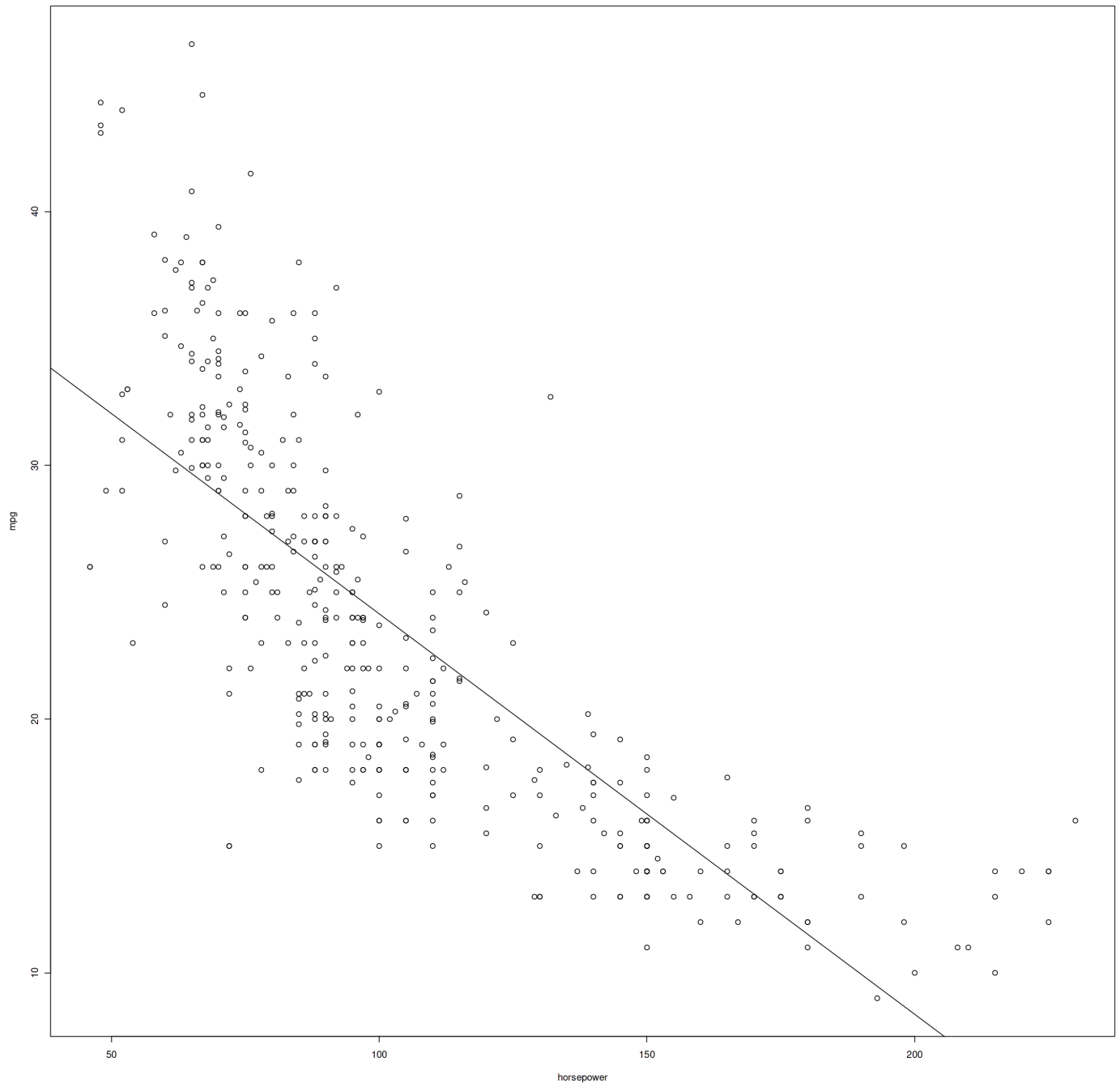
iv. What is the predicted mpg associated with a horsepower of98? What are the associated 95 % confidence and prediction intervals?

[112]:
```
hp98 = data.frame(horsepower = 98)
pred = predict(model, newdata = hp98, interval = 'prediction' )
cat("The predicted value for mpg given a 98 horsepower vehicle is ", pred[1,1],
conf = predict(model, newdata = hp98, interval = 'confidence' )
cat("We have a 95% confidence interval between ", conf[1,2], " and ", conf[1,3])
```

The predicted value for mpg given a 98 horsepower vehicle is 24.46708 with a 95% prediction interval between 1
4.8094 and 34.12476
We have a 95% confidence interval between 23.97308 and 24.96108

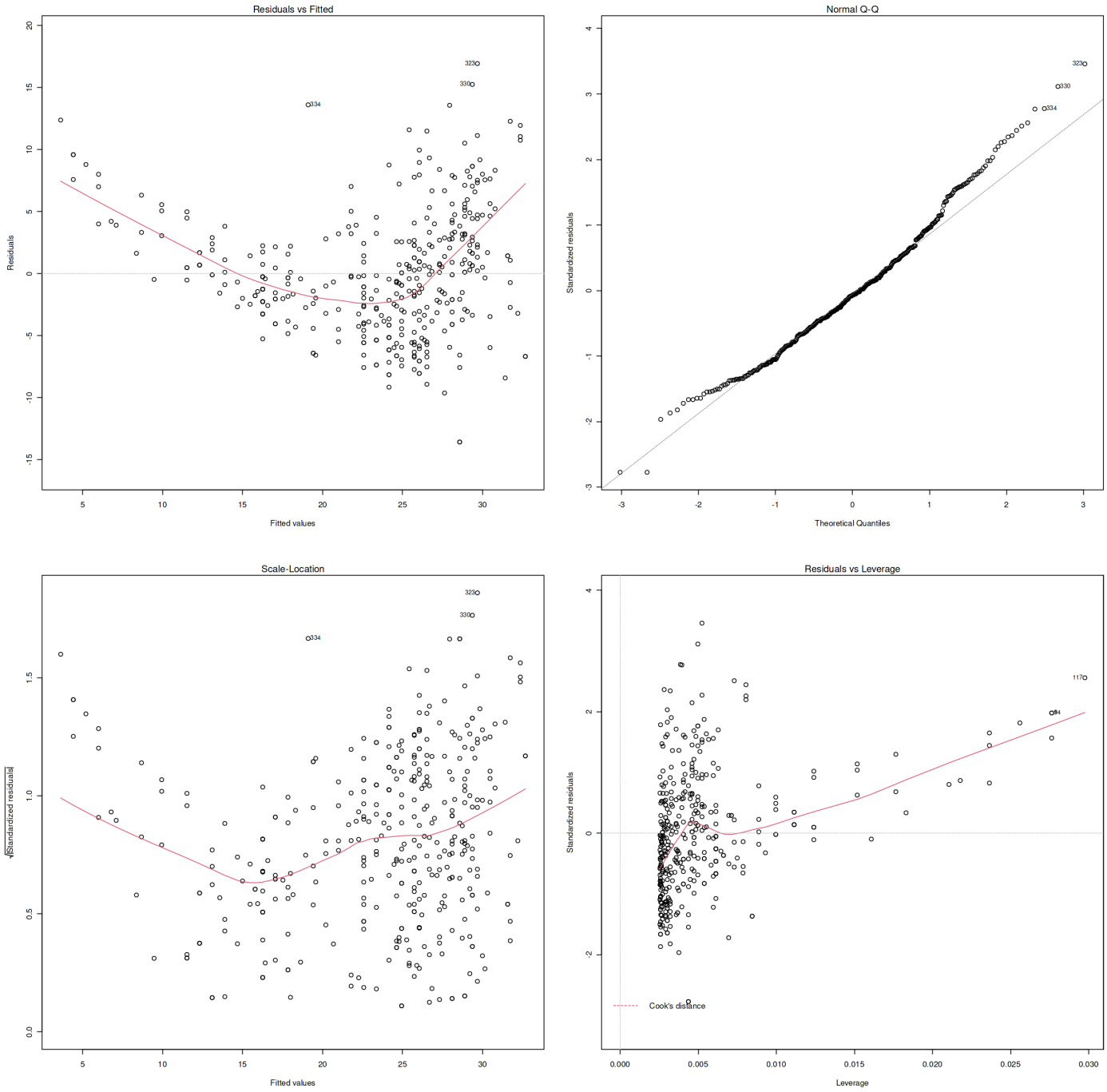
(b) Plot the response and the predictor. Use the abline() function to display the least squares regression line.

[113]:
```
plot(Auto$horsepower, Auto$mpg, xlab = 'horsepower', ylab = 'mpg')
abline(model)
```



(c) Use the plot() function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.

[114]:
```
par(mfrow=c(2,2))
```



The non-linearity displayed in the Residuals vs. Fitted plot is a cause for concern. The changing spread of residuals suggests that a transformation might be needed for the linear model.

Furthermore, the deviation in the Normal Q-Q plot at higher horsepower levels speaks against the suitability of a linear model of horsepower for predicting mpg.

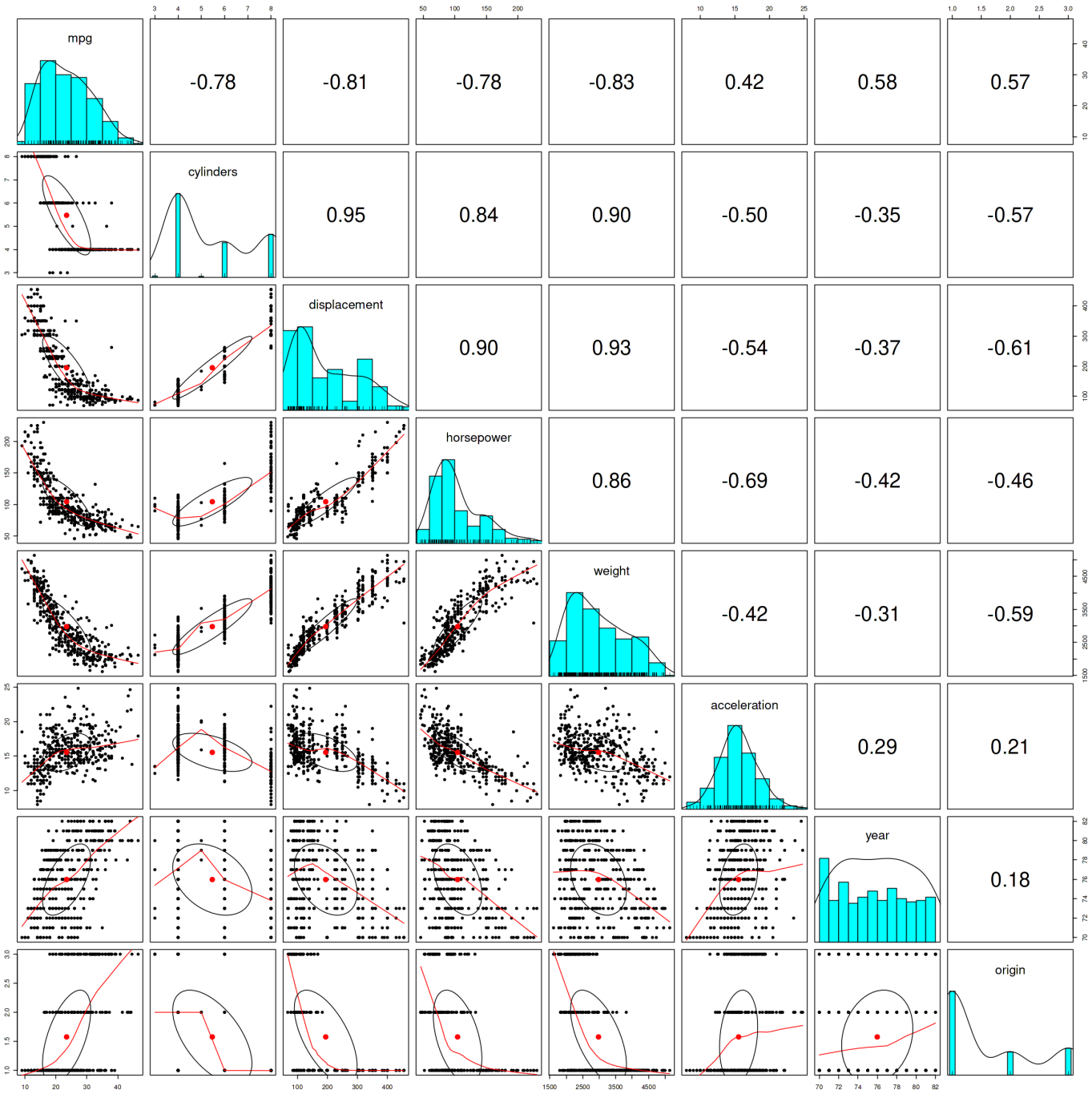
In the Scale-Location plot, an ideal scenario would be a horizontal, flat line. However, we observe a concave curve, indicating a need for a different model.

Lastly, the Residuals vs. Leverage plot shows that there are influential, high-leverage data points that push the Cook's distance above 1, which is another sign of a poor model selection.

9. This question involves the use of multiple linear regression on the Auto data set.

(a) Produce a scatterplot matrix which includes all of the variables in the data set.

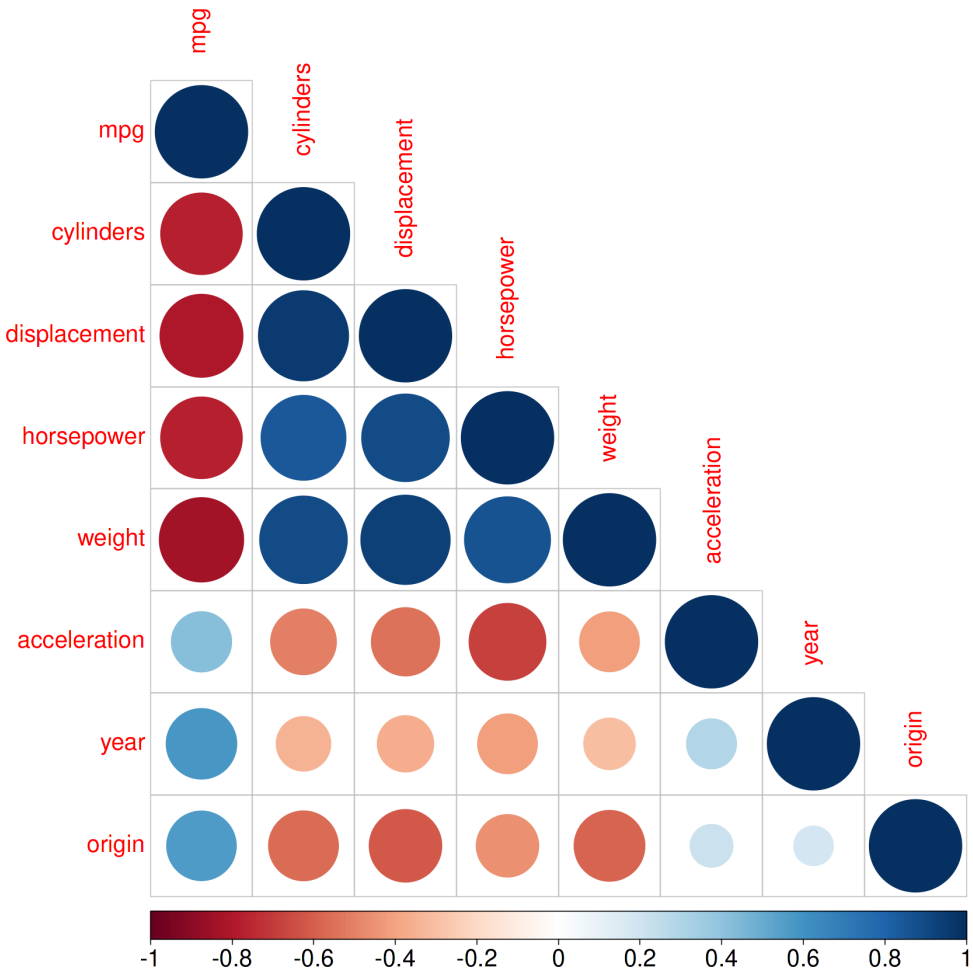
[115]:
```
temp_auto = subset(Auto, select = -c(name))
pairs.panels(temp_auto, cex.labels = 2)
```



Note: I didn't encode model names

(b) Compute the matrix of correlations between the variables using the function cor(). You will need to exclude the name variable, cor() which is qualitative

[116]:
```
correlation_matrix = cor(temp_auto)
my_corrplot(correlation_matrix)
```



[117]:
```
model = lm(mpg ~ ., data = Auto)
summary(model)
```

```
Call:
lm(formula = mpg ~ ., data = Auto)

Residuals:
   Min     1Q Median     3Q    Max
-5.646  0.000  0.000  0.000  5.646

Coefficients: (1 not defined because of singularities)
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                      0.187305  12.773943   0.015 0.988335
cylinders                       -0.918096   0.616653  -1.489 0.140231
displacement                     0.003041   0.015621   0.195 0.846109
horsepower                      -0.042342   0.029070  -1.457 0.148919
weight                          -0.004193   0.001209  -3.467 0.000829
acceleration                    -0.481449   0.171537  -2.807 0.006206
year                             0.636498   0.112195   5.673 1.89e-07
origin                           1.324264   4.243221   0.312 0.755737
nameamc ambassador dpl           3.371358   3.245610   1.039 0.301870
nameamc ambassador sst           3.364264   3.270924   1.029 0.306616
nameamc concord                 -0.122500   3.275058  -0.037 0.970251
nameamc concord d/l             -1.686548   3.422955  -0.493 0.623483
nameamc concord dl 6            -0.529687   3.514452  -0.151 0.880556
nameamc gremlin                 -0.426606   2.958293  -0.144 0.885679
nameamc hornet                   0.269026   2.902972   0.093 0.926382
nameamc hornet sportabout (sw)  -0.403111   3.527112  -0.114 0.909278
nameamc matador                  0.462233   2.667735   0.173 0.862853
nameamc matador (sw)             3.118929   2.894160   1.078 0.284233
nameamc pacer                    0.012163   3.519402   0.003 0.997251
nameamc pacer d/l               -1.682001   3.517154  -0.478 0.633716
nameamc rebel sst                2.875941   3.284138   0.876 0.383658
nameamc spirit dl                0.713000   4.063462   0.175 0.861131
nameaudi 100 ls                  0.936254   3.449074   0.271 0.786702
nameaudi 100ls                  -3.133448   2.984726  -1.050 0.296773
nameaudi 4000                    4.004843   3.244193   1.234 0.220431
nameaudi 5000                   -4.109082   3.450859  -1.191 0.237070
nameaudi 5000s (diesel)         11.652927   3.494256   3.335 0.001266
nameaudi fox                     3.199486   3.328970   0.961 0.339226
namebmw 2002                     2.082893   3.429803   0.607 0.545276
namebmw 320i                    -5.320674   3.378167  -1.575 0.118967
namebuick century                2.213275   3.073805   0.720 0.473473
namebuick century 350            2.163058   3.250910   0.665 0.507615
namebuick century limited        1.154506   3.653147   0.316 0.752754
namebuick century luxus (sw)     3.701574   3.397404   1.090 0.278999
namebuick century special        0.471663   3.468124   0.136 0.892143
namebuick electra 225 custom     5.565899   3.481596   1.599 0.113606
namebuick estate wagon (sw)      2.288361   2.883255   0.794 0.429596
namebuick lesabre custom         3.878880   3.350669   1.158 0.250252
namebuick opel isuzu deluxe      2.360903   4.090583   0.577 0.565360
namebuick regal sport coupe (turbo) -0.770758   3.673221  -0.210 0.834301
namebuick skyhawk                1.178015   3.453774   0.341 0.733885
namebuick skylark                0.485925   3.292621   0.148 0.883024
```

namebuick skylark 320	3.220548	3.239745	0.994	0.323008
namebuick skylark limited	2.526638	3.945353	0.640	0.523630
namecadillac eldorado	7.506805	3.434596	2.186	0.031592
namecadillac seville	4.705913	3.307504	1.423	0.158454
namecapri ii	0.213795	3.974233	0.054	0.957225
namechevroelt chevelle malibu	1.190886	3.501889	0.340	0.734643
namechevrolet bel air	3.526745	3.332996	1.058	0.292994
namechevrolet camaro	1.016970	3.965053	0.256	0.798197
namechevrolet caprice classic	1.486191	2.705678	0.549	0.584249
namechevrolet cavalier	1.711757	4.062490	0.421	0.674558
namechevrolet cavalier 2-door	6.060983	4.085007	1.484	0.141584
namechevrolet cavalier wagon	0.377051	4.065117	0.093	0.926318
namechevrolet chevelle concours (sw)	1.497987	3.363217	0.445	0.657161
namechevrolet chevelle malibu	2.508523	3.012061	0.833	0.407277
namechevrolet chevelle malibu classic	2.148872	2.973671	0.723	0.471889
namechevrolet chevette	2.796588	3.583400	0.780	0.437307
namechevrolet citation	0.835024	3.277093	0.255	0.799488
namechevrolet concours	-0.962069	3.436400	-0.280	0.780185
namechevrolet impala	3.901975	2.655000	1.470	0.145343
namechevrolet malibu	0.338930	2.908792	0.117	0.907516
namechevrolet malibu classic (sw)	1.566909	3.369001	0.465	0.643052
namechevrolet monte carlo	1.755561	3.510683	0.500	0.618323
namechevrolet monte carlo landau	1.360482	2.830125	0.481	0.631955
namechevrolet monte carlo s	2.817323	3.294556	0.855	0.394875
namechevrolet monza 2+2	1.960822	3.418832	0.574	0.567797
namechevrolet nova	0.013367	2.932203	0.005	0.996373
namechevrolet nova custom	-0.583803	3.483183	-0.168	0.867291
namechevrolet vega	-0.123809	3.451390	-0.036	0.971468
namechevrolet vega (sw)	0.835785	3.957409	0.211	0.833241
namechevrolet vega 2300	5.309133	3.978383	1.334	0.185606
namechevrolet woody	0.242401	4.035002	0.060	0.952237
namechevy c10	-1.686820	3.292205	-0.512	0.609722
namechevy c20	4.381943	3.755983	1.167	0.246612
namechevy s-10	4.822281	4.016031	1.201	0.233181
namechrysler cordoba	3.158331	3.288955	0.960	0.339636
namechrysler lebaron medallion	-2.791925	4.039754	-0.691	0.491378
namechrysler lebaron salon	-4.524894	3.582800	-1.263	0.210061
namechrysler lebaron town @ country (sw)	2.084281	3.302393	0.631	0.529641
namechrysler new yorker brougham	5.282481	3.390324	1.558	0.122924
namechrysler newport royal	4.391942	3.291715	1.334	0.185688
namedatsun 1200	7.452422	6.449055	1.156	0.251090
namedatsun 200-sx	-5.232326	6.424023	-0.814	0.417638
namedatsun 200sx	2.817519	6.441163	0.437	0.662913
namedatsun 210	5.606103	6.195917	0.905	0.368126
namedatsun 210 mpg	5.070308	6.390048	0.793	0.429714
namedatsun 280-zx	5.896050	6.830585	0.863	0.390466
namedatsun 310	4.643895	6.277737	0.740	0.461496
namedatsun 310 gx	4.043465	6.266014	0.645	0.520470
namedatsun 510	-2.468844	6.400190	-0.386	0.700649
namedatsun 510 (sw)	3.062356	6.557580	0.467	0.641698
namedatsun 510 hatchback	6.552699	6.395446	1.025	0.308465

namedatsun 610	-3.382103	6.545652	-0.517	0.606711
namedatsun 710	1.331250	6.306882	0.211	0.833331
namedatsun 810	-3.215340	6.816828	-0.472	0.638366
namedatsun 810 maxima	-2.442319	6.860787	-0.356	0.722736
namedatsun b-210	2.371920	6.357360	0.373	0.710004
namedatsun b210	3.331328	6.462449	0.515	0.607549
namedatsun b210 gx	9.604655	6.404554	1.500	0.137408
namedatsun f-10 hatchback	2.950464	6.329051	0.466	0.642281
namedatsun pl510	0.981674	6.254240	0.157	0.875648
namedodge aries se	0.403815	4.041675	0.100	0.920649
namedodge aries wagon (sw)	-2.256829	4.019898	-0.561	0.575992
namedodge aspen	-0.206883	3.088343	-0.067	0.946748
namedodge aspen 6	0.366429	3.465245	0.106	0.916035
namedodge aspen se	3.002155	3.493627	0.859	0.392579
namedodge challenger se	2.064684	3.290611	0.627	0.532048
namedodge charger 2.2	5.309608	4.166685	1.274	0.206030
namedodge colt	1.482396	3.584355	0.414	0.680228
namedodge colt (sw)	3.716950	4.113570	0.904	0.368771
namedodge colt hardtop	1.522048	4.067987	0.374	0.709223
namedodge colt hatchback custom	5.628627	4.197231	1.341	0.183482
namedodge colt m/m	6.221646	4.105872	1.515	0.133405
namedodge coronet brougham	2.669663	3.290052	0.811	0.419382
namedodge coronet custom	1.606868	3.247032	0.495	0.621967
namedodge coronet custom (sw)	3.302820	3.364562	0.982	0.329059
namedodge d100	-1.672691	3.254823	-0.514	0.608647
namedodge d200	5.074896	3.746461	1.355	0.179140
namedodge dart custom	-0.700127	3.265096	-0.214	0.830727
namedodge diplomat	2.561880	3.282848	0.780	0.437333
namedodge magnum xe	2.349068	3.293521	0.713	0.477652
namedodge monaco (sw)	5.409949	3.475789	1.556	0.123313
namedodge monaco brougham	1.448836	3.286662	0.441	0.660460
namedodge omni	2.600955	4.128988	0.630	0.530433
namedodge rampage	0.321131	4.244538	0.076	0.939869
namedodge st. regis	1.874870	3.310142	0.566	0.572613
namefiat 124 sport coupe	0.843793	3.356728	0.251	0.802132
namefiat 124 tc	-1.284412	3.299112	-0.389	0.698013
namefiat 124b	4.234430	3.350724	1.264	0.209780
namefiat 128	-0.302820	2.911282	-0.104	0.917402
namefiat 131	1.571897	3.313206	0.474	0.636407
namefiat strada custom	6.505740	3.221602	2.019	0.046595
namefiat x1.9	3.420877	3.286517	1.041	0.300882
nameford country	3.810815	3.447980	1.105	0.272178
nameford country squire (sw)	2.588023	2.891203	0.895	0.373242
nameford escort 2h	2.903207	4.052921	0.716	0.475754
nameford escort 4w	3.832150	4.172436	0.918	0.360985
nameford f108	-1.507274	3.300893	-0.457	0.649105
nameford f250	5.376492	3.649336	1.473	0.144369
nameford fairmont	1.688480	3.942005	0.428	0.669494
nameford fairmont (auto)	-2.420856	3.577091	-0.677	0.500392
nameford fairmont (man)	-0.267333	3.979340	-0.067	0.946596
nameford fairmont 4	-2.076328	3.945762	-0.526	0.600108

nameford fairmont futura	-2.654573	4.013149	-0.661	0.510101
nameford fiesta	5.590179	4.227911	1.322	0.189646
nameford futura	-1.916806	3.362257	-0.570	0.570117
nameford galaxie 500	4.442480	2.696732	1.647	0.103177
nameford gran torino	2.395149	2.717190	0.881	0.380544
nameford gran torino (sw)	4.305909	3.032697	1.420	0.159315
nameford granada	0.802555	3.476415	0.231	0.817981
nameford granada ghia	1.260545	3.617154	0.348	0.728333
nameford granada gl	-3.179138	3.572976	-0.890	0.376099
nameford granada l	-3.195561	3.585987	-0.891	0.375378
nameford ltd	2.931338	2.921112	1.004	0.318467
nameford ltd landau	-0.237362	3.326697	-0.071	0.943286
nameford maverick	0.203527	3.006394	0.068	0.946185
nameford mustang	-0.086763	3.673287	-0.024	0.981211
nameford mustang gl	-0.608234	4.034452	-0.151	0.880522
nameford mustang ii	-5.296511	3.364527	-1.574	0.119151
nameford mustang ii 2+2	1.150829	3.960070	0.291	0.772059
nameford pinto	-0.775120	3.301082	-0.235	0.814923
nameford pinto (sw)	-0.652033	4.001814	-0.163	0.870957
nameford pinto runabout	-2.119866	3.996017	-0.530	0.597152
nameford ranger	0.908753	4.018900	0.226	0.821651
nameford thunderbird	3.220573	3.297455	0.977	0.331497
nameford torino	2.803510	3.333644	0.841	0.402722
nameford torino 500	2.078087	3.646390	0.570	0.570247
namehi 1200d	6.272325	3.984511	1.574	0.119162
namehonda accord	1.948521	6.116848	0.319	0.750849
namehonda accord cvcc	2.063973	6.429582	0.321	0.748989
namehonda accord lx	-1.109939	6.347929	-0.175	0.861613
namehonda civic	1.053256	6.055926	0.174	0.862341
namehonda civic (auto)	-2.323039	6.246699	-0.372	0.710906
namehonda civic 1300	0.480561	6.224808	0.077	0.938645
namehonda civic 1500 gl	10.153049	6.196347	1.639	0.105003
namehonda civic cvcc	3.182648	6.123490	0.520	0.604593
namehonda prelude	0.704854	6.280545	0.112	0.910907
namemaxda glc deluxe	1.418177	6.244770	0.227	0.820893
namemaxda rx3	-10.867477	6.267079	-1.734	0.086533
namemazda 626	1.984386	6.282271	0.316	0.752875
namemazda glc	15.147599	6.344260	2.388	0.019175
namemazda glc 4	0.684088	6.266174	0.109	0.913324
namemazda glc custom	-2.344981	6.320355	-0.371	0.711545
namemazda glc custom l	4.174484	6.349602	0.657	0.512674
namemazda glc deluxe	2.292567	6.386457	0.359	0.720505
namemazda rx-4	-6.598216	6.446482	-1.024	0.308958
namemazda rx-7 gs	-8.439964	6.251072	-1.350	0.180548
namemazda rx2 coupe	-8.070898	6.367885	-1.267	0.208460
namemercedes benz 300d	4.052324	3.714017	1.091	0.278317
namemercedes-benz 240d	6.431350	3.628762	1.772	0.079923
namemercedes-benz 280s	-0.574813	3.970931	-0.145	0.885247
namemercury capri 2000	-0.712146	4.074233	-0.175	0.861658
namemercury capri v6	-0.303572	3.691241	-0.082	0.934648
namemercury cougar brougham	1.589959	3.361893	0.473	0.637471

namemercury grand marquis	-0.237277	3.337834	-0.071	0.943496
namemercury lynx l	5.872369	4.147537	1.416	0.160468
namemercury marquis	3.228385	3.350958	0.963	0.338069
namemercury marquis brougham	4.746644	3.423970	1.386	0.169282
namemercury monarch	-1.952366	3.676929	-0.531	0.596819
namemercury monarch ghia	2.483829	3.298276	0.753	0.453489
namemercury zephyr	-0.947324	3.554276	-0.267	0.790476
namemercury zephyr 6	-2.197060	3.553076	-0.618	0.537995
namenissan stanza xe	2.717783	6.321214	0.430	0.668323
nameoldsmobile cutlass ciera (diesel)	13.431975	3.705632	3.625	0.000492
nameoldsmobile cutlass ls	9.023570	3.661744	2.464	0.015743
nameoldsmobile cutlass salon brougham	4.510530	3.094646	1.458	0.148656
nameoldsmobile cutlass supreme	3.859509	3.468811	1.113	0.269004
nameoldsmobile delta 88 royale	2.897729	3.327483	0.871	0.386290
nameoldsmobile omega	-2.416124	3.226729	-0.749	0.456054
nameoldsmobile omega brougham	2.387769	3.687973	0.647	0.519087
nameoldsmobile starfire sx	-0.102617	3.905251	-0.026	0.979098
nameoldsmobile vista cruiser	2.806911	3.342458	0.840	0.403391
nameopel 1900	0.364179	2.888955	0.126	0.899983
nameopel manta	-1.463586	2.886584	-0.507	0.613446
namepeugeot 304	6.452724	3.479899	1.854	0.067166
namepeugeot 504	2.256143	3.020582	0.747	0.457171
namepeugeot 504 (sw)	1.205706	3.655813	0.330	0.742359
namepeugeot 505s turbo diesel	3.702626	3.568596	1.038	0.302418
namepeugeot 604sl	-3.734443	3.870878	-0.965	0.337405
nameplymouth 'cuda 340	0.001994	3.312175	0.001	0.999521
nameplymouth arrow gs	-0.550125	4.028316	-0.137	0.891698
nameplymouth champ	7.809841	4.225532	1.848	0.068044
nameplymouth cricket	3.723019	4.089471	0.910	0.365191
nameplymouth custom suburb	4.243663	3.356462	1.264	0.209569
nameplymouth duster	3.100163	3.013252	1.029	0.306472
nameplymouth fury	2.614640	3.537123	0.739	0.461822
nameplymouth fury gran sedan	3.498384	3.316570	1.055	0.294497
nameplymouth fury iii	4.299562	2.679985	1.604	0.112352
nameplymouth grand fury	5.319669	3.369828	1.579	0.118138
nameplymouth horizon	4.301083	4.214230	1.021	0.310336
nameplymouth horizon 4	4.113045	4.189513	0.982	0.329010
nameplymouth horizon miser	6.302919	4.228303	1.491	0.139755
nameplymouth horizon tc3	5.209914	4.143729	1.257	0.212088
nameplymouth reliant	-0.545007	3.754920	-0.145	0.884940
nameplymouth sapporo	-0.765473	3.911418	-0.196	0.845310
nameplymouth satellite	4.364495	3.282401	1.330	0.187186
nameplymouth satellite custom	0.448326	3.526696	0.127	0.899143
nameplymouth satellite custom (sw)	3.223333	3.292706	0.979	0.330392
nameplymouth satellite sebring	1.749802	3.497976	0.500	0.618204
nameplymouth valiant	1.232906	3.073459	0.401	0.689319
nameplymouth valiant custom	-0.014077	3.503255	-0.004	0.996803
nameplymouth volare	1.061859	3.474219	0.306	0.760626
nameplymouth volare custom	1.277611	3.488038	0.366	0.715064
nameplymouth volare premier v8	-1.282210	3.255780	-0.394	0.694696
namepontiac astro	0.074570	3.909875	0.019	0.984828

namepontiac catalina	5.445037	2.738420	1.988	0.049987
namepontiac catalina brougham	5.087948	3.326698	1.529	0.129871
namepontiac firebird	2.261616	3.549778	0.637	0.525762
namepontiac grand prix	6.401045	3.428059	1.867	0.065314
namepontiac grand prix lj	2.265087	3.274738	0.692	0.491020
namepontiac j2000 se hatchback	2.822025	4.099435	0.688	0.493079
namepontiac lemans v6	0.400000	3.461834	0.116	0.908285
namepontiac phoenix	3.126592	3.641875	0.859	0.393024
namepontiac phoenix lj	1.358450	3.465997	0.392	0.696086
namepontiac safari (sw)	7.162903	3.555491	2.015	0.047108
namepontiac sunbird coupe	0.108434	3.930485	0.028	0.978055
namepontiac ventura sj	1.102220	3.450690	0.319	0.750192
namerenault 12 (sw)	1.482167	3.395489	0.437	0.663573
namerenault 12tl	-0.731646	3.266427	-0.224	0.823303
namerenault 5 gtl	6.648358	3.298178	2.016	0.046983
namesaab 99e	3.370839	3.531246	0.955	0.342500
namesaab 99gle	-3.431693	3.516847	-0.976	0.331939
namesaab 99le	0.426670	3.063299	0.139	0.889555
namesubaru	-0.233280	6.193441	-0.038	0.970043
namesubaru dl	0.927894	6.146221	0.151	0.880357
nametoyota carina	-4.818346	6.613451	-0.729	0.468269
nametoyota celica gt	0.811953	6.436095	0.126	0.899906
nametoyota celica gt liftback	-7.749581	6.454951	-1.201	0.233256
nametoyota corolla	0.502682	6.023789	0.083	0.933690
nametoyota corolla 1200	4.597137	6.314043	0.728	0.468565
nametoyota corolla 1600 (sw)	0.864045	6.491090	0.133	0.894419
nametoyota corolla liftback	-2.358642	6.455302	-0.365	0.715735
nametoyota corolla tercel	6.264712	6.367373	0.984	0.327968
nametoyota corona	-1.217090	6.101896	-0.199	0.842379
nametoyota corona hardtop	-1.623374	6.517263	-0.249	0.803893
nametoyota corona liftback	0.624486	6.480690	0.096	0.923461
nametoyota corona mark ii	-0.196994	6.567408	-0.030	0.976141
nametoyota cressida	-2.182111	6.794927	-0.321	0.748894
nametoyota mark ii	-3.168040	6.744488	-0.470	0.639756
nametoyota starlet	4.766154	6.246630	0.763	0.447578
nametoyota tercel	4.934523	6.305873	0.783	0.436079
nametoyouta corona mark ii (sw)	-2.085503	6.540427	-0.319	0.750612
nametriumph tr7 coupe	6.023330	3.279325	1.837	0.069741
namevokswagen rabbit	-2.827106	3.226492	-0.876	0.383382
namevolkswagen 1131 deluxe sedan	1.497675	3.578202	0.419	0.676597
namevolkswagen 411 (sw)	-0.947433	3.453190	-0.274	0.784470
namevolkswagen dasher	-0.858245	2.683260	-0.320	0.749866
namevolkswagen jetta	1.112715	3.228297	0.345	0.731190
namevolkswagen model 111	1.727602	3.449948	0.501	0.617833
namevolkswagen rabbit	-1.074894	2.814449	-0.382	0.703474
namevolkswagen rabbit custom	-1.051324	3.224945	-0.326	0.745228
namevolkswagen rabbit custom diesel	14.722011	3.409156	4.318	4.24e-05
namevolkswagen rabbit l	3.125355	3.244626	0.963	0.338159
namevolkswagen scirocco	0.942324	3.219495	0.293	0.770470
namevolkswagen super beetle	0.311060	3.498963	0.089	0.929370
namevolkswagen type 3	0.764483	3.649570	0.209	0.834581

```
namevolvo 144ea                -2.766457   3.580135  -0.773 0.441829
namevolvo 145e (sw)            -3.338886   3.609192  -0.925 0.357530
namevolvo 244dl                -1.790859   3.478340  -0.515 0.607987
namevolvo 245                  -2.848126   3.553844  -0.801 0.425122
namevolvo 264gl                -5.464384   3.628736  -1.506 0.135809
namevolvo diesel                7.278640   3.656834   1.990 0.049758
namevw dasher (diesel)         16.275630   3.503824   4.645 1.23e-05
namevw pickup                  15.324526   3.586036   4.273 5.00e-05
namevw rabbit                   4.756880   2.789326   1.705 0.091774
namevw rabbit c (diesel)       15.164570   3.399743   4.461 2.49e-05
namevw rabbit custom                  NA         NA      NA       NA

(Intercept)
cylinders
displacement
horsepower
weight                          ***
acceleration                     **
year                            ***
origin
nameamc ambassador dpl
nameamc ambassador sst
nameamc concord
nameamc concord d/l
nameamc concord dl 6
nameamc gremlin
nameamc hornet
nameamc hornet sportabout (sw)
nameamc matador
nameamc matador (sw)
nameamc pacer
nameamc pacer d/l
nameamc rebel sst
nameamc spirit dl
nameaudi 100 ls
nameaudi 100ls
nameaudi 4000
nameaudi 5000
nameaudi 5000s (diesel)          **
nameaudi fox
namebmw 2002
namebmw 320i
namebuick century
namebuick century 350
namebuick century limited
namebuick century luxus (sw)
namebuick century special
namebuick electra 225 custom
namebuick estate wagon (sw)
namebuick lesabre custom
namebuick opel isuzu deluxe
```

namebuick regal sport coupe (turbo)
namebuick skyhawk
namebuick skylark
namebuick skylark 320
namebuick skylark limited
namecadillac eldorado *
namecadillac seville
namecapri ii
namechevroelt chevelle malibu
namechevrolet bel air
namechevrolet camaro
namechevrolet caprice classic
namechevrolet cavalier
namechevrolet cavalier 2-door
namechevrolet cavalier wagon
namechevrolet chevelle concours (sw)
namechevrolet chevelle malibu
namechevrolet chevelle malibu classic
namechevrolet chevette
namechevrolet citation
namechevrolet concours
namechevrolet impala
namechevrolet malibu
namechevrolet malibu classic (sw)
namechevrolet monte carlo
namechevrolet monte carlo landau
namechevrolet monte carlo s
namechevrolet monza 2+2
namechevrolet nova
namechevrolet nova custom
namechevrolet vega
namechevrolet vega (sw)
namechevrolet vega 2300
namechevrolet woody
namechevy c10
namechevy c20
namechevy s-10
namechrysler cordoba
namechrysler lebaron medallion
namechrysler lebaron salon
namechrysler lebaron town @ country (sw)
namechrysler new yorker brougham
namechrysler newport royal
namedatsun 1200
namedatsun 200-sx
namedatsun 200sx
namedatsun 210
namedatsun 210 mpg
namedatsun 280-zx
namedatsun 310
namedatsun 310 gx

namedatsun 510
namedatsun 510 (sw)
namedatsun 510 hatchback
namedatsun 610
namedatsun 710
namedatsun 810
namedatsun 810 maxima
namedatsun b-210
namedatsun b210
namedatsun b210 gx
namedatsun f-10 hatchback
namedatsun pl510
namedodge aries se
namedodge aries wagon (sw)
namedodge aspen
namedodge aspen 6
namedodge aspen se
namedodge challenger se
namedodge charger 2.2
namedodge colt
namedodge colt (sw)
namedodge colt hardtop
namedodge colt hatchback custom
namedodge colt m/m
namedodge coronet brougham
namedodge coronet custom
namedodge coronet custom (sw)
namedodge d100
namedodge d200
namedodge dart custom
namedodge diplomat
namedodge magnum xe
namedodge monaco (sw)
namedodge monaco brougham
namedodge omni
namedodge rampage
namedodge st. regis
namefiat 124 sport coupe
namefiat 124 tc
namefiat 124b
namefiat 128
namefiat 131
namefiat strada custom *
namefiat x1.9
nameford country
nameford country squire (sw)
nameford escort 2h
nameford escort 4w
nameford f108
nameford f250
nameford fairmont

nameford fairmont (auto)
nameford fairmont (man)
nameford fairmont 4
nameford fairmont futura
nameford fiesta
nameford futura
nameford galaxie 500
nameford gran torino
nameford gran torino (sw)
nameford granada
nameford granada ghia
nameford granada gl
nameford granada l
nameford ltd
nameford ltd landau
nameford maverick
nameford mustang
nameford mustang gl
nameford mustang ii
nameford mustang ii 2+2
nameford pinto
nameford pinto (sw)
nameford pinto runabout
nameford ranger
nameford thunderbird
nameford torino
nameford torino 500
namehi 1200d
namehonda accord
namehonda accord cvcc
namehonda accord lx
namehonda civic
namehonda civic (auto)
namehonda civic 1300
namehonda civic 1500 gl
namehonda civic cvcc
namehonda prelude
namemaxda glc deluxe
namemaxda rx3 .
namemazda 626
namemazda glc *
namemazda glc 4
namemazda glc custom
namemazda glc custom l
namemazda glc deluxe
namemazda rx-4
namemazda rx-7 gs
namemazda rx2 coupe
namemercedes benz 300d
namemercedes-benz 240d .
namemercedes-benz 280s

namemercury capri 2000
namemercury capri v6
namemercury cougar brougham
namemercury grand marquis
namemercury lynx l
namemercury marquis
namemercury marquis brougham
namemercury monarch
namemercury monarch ghia
namemercury zephyr
namemercury zephyr 6
namenissan stanza xe
nameoldsmobile cutlass ciera (diesel) ***
nameoldsmobile cutlass ls *
nameoldsmobile cutlass salon brougham
nameoldsmobile cutlass supreme
nameoldsmobile delta 88 royale
nameoldsmobile omega
nameoldsmobile omega brougham
nameoldsmobile starfire sx
nameoldsmobile vista cruiser
nameopel 1900
nameopel manta
namepeugeot 304 .
namepeugeot 504
namepeugeot 504 (sw)
namepeugeot 505s turbo diesel
namepeugeot 604sl
nameplymouth 'cuda 340
nameplymouth arrow gs
nameplymouth champ .
nameplymouth cricket
nameplymouth custom suburb
nameplymouth duster
nameplymouth fury
nameplymouth fury gran sedan
nameplymouth fury iii
nameplymouth grand fury
nameplymouth horizon
nameplymouth horizon 4
nameplymouth horizon miser
nameplymouth horizon tc3
nameplymouth reliant
nameplymouth sapporo
nameplymouth satellite
nameplymouth satellite custom
nameplymouth satellite custom (sw)
nameplymouth satellite sebring
nameplymouth valiant
nameplymouth valiant custom
nameplymouth volare

nameplymouth volare custom
nameplymouth volare premier v8
namepontiac astro
namepontiac catalina *
namepontiac catalina brougham
namepontiac firebird
namepontiac grand prix .
namepontiac grand prix lj
namepontiac j2000 se hatchback
namepontiac lemans v6
namepontiac phoenix
namepontiac phoenix lj
namepontiac safari (sw) *
namepontiac sunbird coupe
namepontiac ventura sj
namerenault 12 (sw)
namerenault 12tl
namerenault 5 gtl *
namesaab 99e
namesaab 99gle
namesaab 99le
namesubaru
namesubaru dl
nametoyota carina
nametoyota celica gt
nametoyota celica gt liftback
nametoyota corolla
nametoyota corolla 1200
nametoyota corolla 1600 (sw)
nametoyota corolla liftback
nametoyota corolla tercel
nametoyota corona
nametoyota corona hardtop
nametoyota corona liftback
nametoyota corona mark ii
nametoyota cressida
nametoyota mark ii
nametoyota starlet
nametoyota tercel
nametoyouta corona mark ii (sw)
nametriumph tr7 coupe .
namevokswagen rabbit
namevolkswagen 1131 deluxe sedan
namevolkswagen 411 (sw)
namevolkswagen dasher
namevolkswagen jetta
namevolkswagen model 111
namevolkswagen rabbit
namevolkswagen rabbit custom
namevolkswagen rabbit custom diesel ***
namevolkswagen rabbit l

```
namevolkswagen scirocco
namevolkswagen super beetle
namevolkswagen type 3
namevolvo 144ea
namevolvo 145e (sw)
namevolvo 244dl
namevolvo 245
namevolvo 264gl
namevolvo diesel                *
namevw dasher (diesel)            ***
namevw pickup                   ***
namevw rabbit                   .
namevw rabbit c (diesel)          ***
namevw rabbit custom
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.272 on 85 degrees of freedom
Multiple R-squared:  0.9816,    Adjusted R-squared:  0.9153
F-statistic:  14.8 on 306 and 85 DF,  p-value: < 2.2e-16
```

i. Is there a relationship between the predictors and the response?

[118]:
```
pval = pf(summary(model)$fstatistic[1], summary(model)$fstatistic[2], summary(mo
cat("The F-statistic is ", summary(model)$fstatistic, "\n")
cat("The p-value is ", pval)
```

```
The F-statistic is  14.80296 306 85
The p-value is  1.835635e-32
```

In this case, the extremely low p-value (approximately 1.836 x 10^-32), indicates strong evidence against the null hypothesis. Usually, if the p-value falls below a predetermined significance level (often 0.05), it's considered statistically significant.

Therefore, with such a small p-value, we have strong evidence to reject the null hypothesis, indicating a significant relationship between the predictors and the response variable.

ii. Which predictors appear to have a statistically significant relationship to the response?

[119]:
```
summary(model)$coefficients
```

A matrix: 307 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.187305388	12.773942994	0.014663083	9.883353e-01

	Estimate	Std. Error	t value	Pr(>|t|)
cylinders	-0.918096032	0.616652670	-1.488838170	1.402310e-01
displacement	0.003041110	0.015621221	0.194678132	8.461095e-01
horsepower	-0.042342228	0.029069767	-1.456572665	1.489185e-01
weight	-0.004192648	0.001209473	-3.466509749	8.292887e-04
acceleration	-0.481448874	0.171537286	-2.806671865	6.205556e-03
year	0.636497759	0.112194538	5.673161734	1.894602e-07
origin	1.324263763	4.243220865	0.312089284	7.557370e-01
nameamc ambassador dpl	3.371358012	3.245610483	1.038743876	3.018697e-01
nameamc ambassador sst	3.364264053	3.270924003	1.028536294	3.066156e-01
nameamc concord	-0.122500355	3.275058082	-0.037404025	9.702506e-01
nameamc concord d/l	-1.686548268	3.422955121	-0.492717026	6.234829e-01
nameamc concord dl 6	-0.529686512	3.514451759	-0.150716683	8.805563e-01
nameamc gremlin	-0.426605723	2.958292761	-0.144206729	8.856786e-01
nameamc hornet	0.269026085	2.902971677	0.092672652	9.263817e-01
nameamc hornet sportabout (sw)	-0.403110903	3.527111551	-0.114289241	9.092780e-01
nameamc matador	0.462233110	2.667735270	0.173267983	8.628527e-01
nameamc matador (sw)	3.118929445	2.894159858	1.077663156	2.842330e-01
nameamc pacer	0.012162863	3.519401608	0.003455946	9.972507e-01
nameamc pacer d/l	-1.682001190	3.517154057	-0.478227898	6.337157e-01
nameamc rebel sst	2.875941050	3.284137921	0.875706538	3.836576e-01
nameamc spirit dl	0.713000196	4.063462043	0.175466188	8.611307e-01
nameaudi 100 ls	0.936254043	3.449073648	0.271450870	7.867025e-01
nameaudi 100ls	-3.133447775	2.984726465	-1.049827450	2.967732e-01
nameaudi 4000	4.004843450	3.244192524	1.234465409	2.204315e-01
nameaudi 5000	-4.109081904	3.450858732	-1.190741848	2.370697e-01
nameaudi 5000s (diesel)	11.652926764	3.494255701	3.334880948	1.266297e-03
nameaudi fox	3.199486346	3.328969754	0.961104060	3.392260e-01
namebmw 2002	2.082892923	3.429802557	0.607292370	5.452758e-01
namebmw 320i	-5.320674111	3.378167436	-1.575017879	1.189672e-01
⋮	⋮	⋮	⋮	⋮
nametoyota corona mark ii	-0.1969944	6.567408	-0.02999576	9.761408e-01
nametoyota cressida	-2.1821109	6.794927	-0.32113825	7.488939e-01
nametoyota mark ii	-3.1680400	6.744488	-0.46972284	6.397559e-01
nametoyota starlet	4.7661544	6.246630	0.76299616	4.475781e-01
nametoyota tercel	4.9345230	6.305873	0.78252816	4.360789e-01

	Estimate	Std. Error	t value	Pr(>|t|)
nametoyouta corona mark ii (sw)	-2.0855028	6.540427	-0.31886340	7.506123e-01
nametriumph tr7 coupe	6.0233301	3.279325	1.83675931	6.974069e-02
namevokswagen rabbit	-2.8271061	3.226492	-0.87621665	3.833818e-01
namevolkswagen 1131 deluxe sedan	1.4976746	3.578202	0.41855505	6.765970e-01
namevolkswagen 411 (sw)	-0.9474326	3.453190	-0.27436443	7.844703e-01
namevolkswagen dasher	-0.8582449	2.683260	-0.31985156	7.498657e-01
namevolkswagen jetta	1.1127150	3.228297	0.34467555	7.311897e-01
namevolkswagen model 111	1.7276020	3.449948	0.50076179	6.178330e-01
namevolkswagen rabbit	-1.0748945	2.814449	-0.38192007	7.034738e-01
namevolkswagen rabbit custom	-1.0513236	3.224945	-0.32599734	7.452276e-01
namevolkswagen rabbit custom diesel	14.7220109	3.409156	4.31837372	4.235706e-05
namevolkswagen rabbit l	3.1253548	3.244626	0.96324032	3.381593e-01
namevolkswagen scirocco	0.9423239	3.219495	0.29269311	7.704700e-01
namevolkswagen super beetle	0.3110603	3.498963	0.08890070	9.293700e-01
namevolkswagen type 3	0.7644832	3.649570	0.20947214	8.345809e-01
namevolvo 144ea	-2.7664569	3.580135	-0.77272410	4.418291e-01
namevolvo 145e (sw)	-3.3388859	3.609192	-0.92510607	3.575297e-01
namevolvo 244dl	-1.7908586	3.478340	-0.51486014	6.079870e-01
namevolvo 245	-2.8481258	3.553844	-0.80142127	4.251225e-01
namevolvo 264gl	-5.4643838	3.628736	-1.50586424	1.358093e-01
namevolvo diesel	7.2786399	3.656834	1.99042094	4.975796e-02
namevw dasher (diesel)	16.2756298	3.503824	4.64510441	1.228964e-05
namevw pickup	15.3245265	3.586036	4.27338950	5.002966e-05
namevw rabbit	4.7568795	2.789326	1.70538689	9.177441e-02
namevw rabbit c (diesel)	15.1645700	3.399743	4.46050422	2.487447e-05

To ascertain which predictor shares the strongest relationship with the response, we would again examine the p-value information. Based on the matrix provided, the variable with the smallest p-value is "namevw dasher (diesel)" with a p-value of approximately 1.229 x 10^-5. This p-value, being the closest to 0, suggests that this variable has the most substantial influence on the model, and therefore, it would be considered the strongest predictor among the other variables.

iii. What does the coefficient for the year variable suggest?

[120]:
```
coef(model)['year']
```

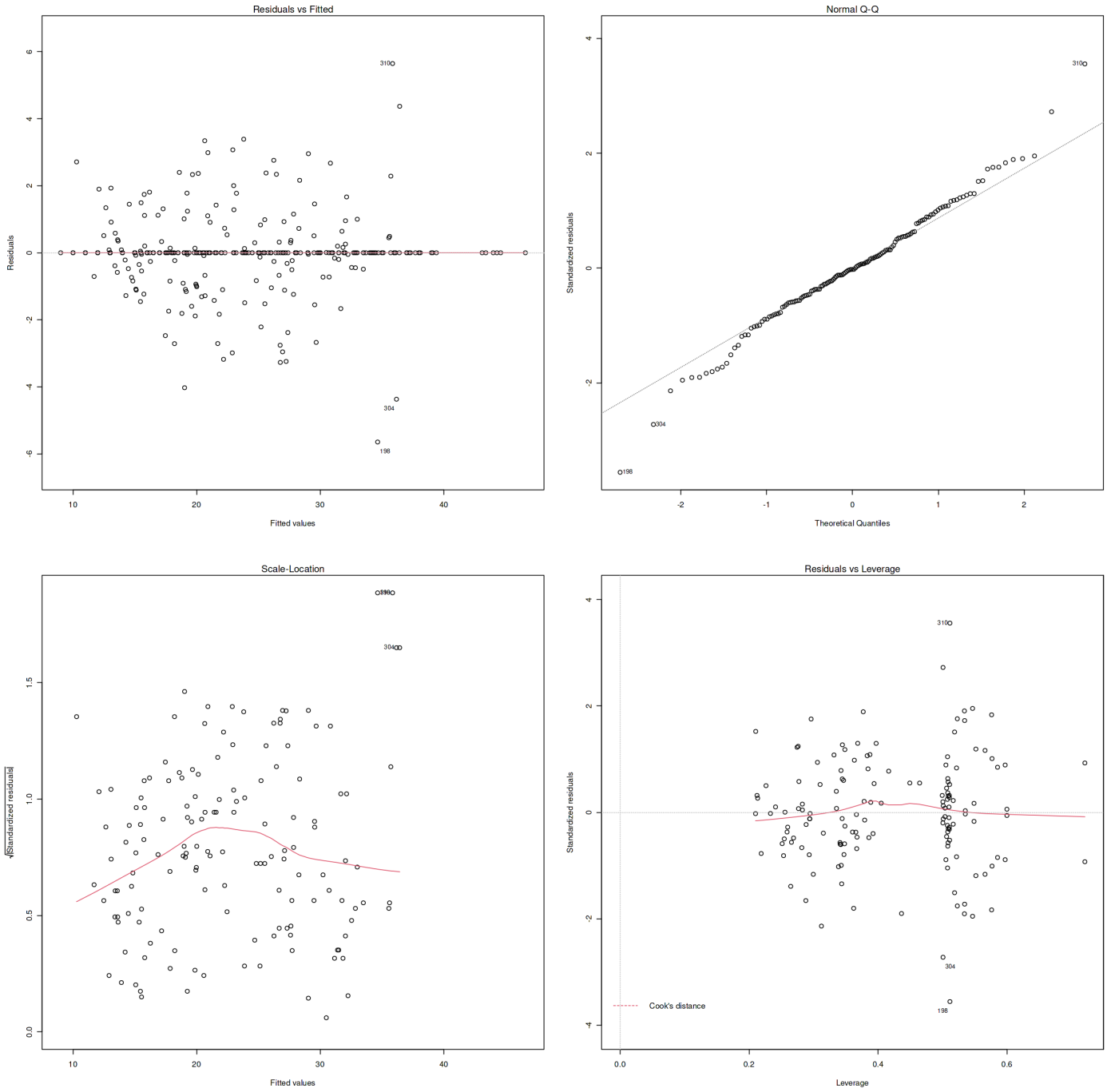
year: 0.636497758875279

Based on the coefficient, we can infer that the 'year' variable has a positive relationship with the response. To contextualize this in practical terms, for each passing year, we expect an increase of approximately 0.64 in the response variable, assuming that other variables remain constant.

(d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

[121]:
```
par(mfrow=c(2,2))
plot(model)
```

Warning message:
"not plotting observations with leverage one:
2, 3, 4, 5, 10, 11, 12, 13, 15, 20, 22, 23, 24, 26, 27, 28, 29, 31, 34, 36, 39, 42, 44, 45, 46, 47, 48, 49, 51, 52, 54, 55, 56, 57, 58, 59, 61, 66, 67, 68, 69, 70, 71, 73, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 89, 90, 93, 94, 95, 96, 98, 102, 104, 105, 106, 108, 110, 111, 113, 114, 115, 116, 120, 121, 124, 128, 134, 136, 137, 140, 147, 150, 151, 153, 156, 157, 160, 162, 163, 164, 165, 169, 175, 178, 181, 183, 185, 187, 195, 198, 199, 200, 201, 203, 206, 207, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 221, 222, 223, 224, 226, 227, 228, 230, 231, 232, 233, 234, 235, 237, 240, 241, 242, 243, 244, 245, 246, 249, 250, 251, 253, 254, 255, 257, 258, 260, 262, 263, 264, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 285, 286, 287, 290, 291, 292, 293, 294, 295, 296, 297, 300, 301, 303, 304, 306, 309, 311, 313, 316, 317, 319, 321, 324, 325, 326, 327, 328, 330, 331, 332, 333, 337, 340, 341, 342, 344, 345, 346, 347, 348, 349, 350, 351, 353, 355, 356, 357, 358, 360, 361, 362, 363, 364, 365, 366, 367, 369, 370, 371, 372, 373, 374, 375, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392"



For a majority of the plots, the data is quite dispersed, and there appear to be a considerable number of outliers. Outliers can significantly skew the data, leading to potentially inaccurate representations of the model. We might want to consider removing some outliers to improve the fit of our model and achieve a more accurate response. The leverage plots reveal several instances of high leverage, and as previously mentioned, if these outliers are impacting the overall response or outcome, it might be beneficial to either remove them or explore alternative methods to better accommodate these outliers within the model.

(e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

[122]:
```
pairwise_model = lm(mpg ~ . * ., data = temp_auto)
summary(pairwise_model)
```

```
Call:
lm(formula = mpg ~ . * ., data = temp_auto)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.6303 -1.4481  0.0596  1.2739 11.1386 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)             3.548e+01  5.314e+01   0.668  0.50475    
cylinders               6.989e+00  8.248e+00   0.847  0.39738    
displacement           -4.785e-01  1.894e-01  -2.527  0.01192 *  
horsepower              5.034e-01  3.470e-01   1.451  0.14769    
weight                  4.133e-03  1.759e-02   0.235  0.81442    
acceleration           -5.859e+00  2.174e+00  -2.696  0.00735 ** 
year                    6.974e-01  6.097e-01   1.144  0.25340    
origin                 -2.090e+01  7.097e+00  -2.944  0.00345 ** 
cylinders:displacement  -3.383e-03  6.455e-03  -0.524  0.60051    
cylinders:horsepower    1.161e-02  2.420e-02   0.480  0.63157    
cylinders:weight        3.575e-04  8.955e-04   0.399  0.69000    
cylinders:acceleration  2.779e-01  1.664e-01   1.670  0.09584 .  
cylinders:year         -1.741e-01  9.714e-02  -1.793  0.07389 .  
cylinders:origin        4.022e-01  4.926e-01   0.816  0.41482    
displacement:horsepower -8.491e-05  2.885e-04  -0.294  0.76867    
displacement:weight     2.472e-05  1.470e-05   1.682  0.09342 .  
displacement:acceleration -3.479e-03  3.342e-03  -1.041  0.29853    
displacement:year       5.934e-03  2.391e-03   2.482  0.01352 *  
displacement:origin     2.398e-02  1.947e-02   1.232  0.21875    
horsepower:weight      -1.968e-05  2.924e-05  -0.673  0.50124    
horsepower:acceleration -7.213e-03  3.719e-03  -1.939  0.05325 .  
horsepower:year        -5.838e-03  3.938e-03  -1.482  0.13916    
horsepower:origin       2.233e-03  2.930e-02   0.076  0.93931    
weight:acceleration     2.346e-04  2.289e-04   1.025  0.30596    
weight:year            -2.245e-04  2.127e-04  -1.056  0.29182    
weight:origin          -5.789e-04  1.591e-03  -0.364  0.71623    
acceleration:year       5.562e-02  2.558e-02   2.174  0.03033 *  
acceleration:origin     4.583e-01  1.567e-01   2.926  0.00365 ** 
year:origin             1.393e-01  7.399e-02   1.882  0.06062 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.695 on 363 degrees of freedom
Multiple R-squared:  0.8893,    Adjusted R-squared:  0.8808 
F-statistic: 104.2 on 28 and 363 DF,  p-value: < 2.2e-16
```

[123]:
```
custom_model_0 = lm(mpg ~ year:acceleration:origin, data = temp_auto)
```

```
summary(custom_model_0)
pval = pf(
    summary(custom_model_0)$fstatistic[1],
    summary(custom_model_0)$fstatistic[2],
    summary(custom_model_0)$fstatistic[3],
    lower.tail = F
)
```

```
Call:
lm(formula = mpg ~ year:acceleration:origin, data = temp_auto)

Residuals:
   Min     1Q  Median     3Q    Max
-13.729  -4.045  -1.170   3.395  18.207

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           1.473e+01  5.882e-01   25.04   <2e-16 ***
year:acceleration:origin 4.567e-03  2.657e-04   17.19   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.895 on 390 degrees of freedom
Multiple R-squared:  0.4311,    Adjusted R-squared:  0.4296
F-statistic: 295.5 on 1 and 390 DF,  p-value: < 2.2e-16
```

[124]:

```
custom_model_1 = lm(mpg ~ displacement:year:weight, data = temp_auto)
summary(custom_model_1)
pval = pf(
    summary(custom_model_1)$fstatistic[1],
    summary(custom_model_1)$fstatistic[2],
    summary(custom_model_1)$fstatistic[3],
    lower.tail = F
)
```

```
Call:
lm(formula = mpg ~ displacement:year:weight, data = temp_auto)

Residuals:
   Min     1Q  Median     3Q    Max
-11.661  -3.457  -0.751   2.679  17.527

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)             3.140e+01  4.016e-01    78.2   <2e-16 ***
displacement:year:weight -1.606e-07  6.425e-09   -25.0   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Residual standard error: 4.845 on 390 degrees of freedom
Multiple R-squared: 0.6157, Adjusted R-squared: 0.6147
F-statistic: 624.7 on 1 and 390 DF, p-value: < 2.2e-16

[125]:
```
# Create dataframe
interaction_p_values <- data.frame(
    Interaction = c(
        "cylinders:acceleration",
        "cylinders:year",
        "displacement:weight",
        "displacement:year",
        "acceleration:year",
        "acceleration:origin",
        "year:origin"
    ),
    P_value = c(
        0.09584,
        0.07389,
        0.09342,
        0.01352,
        0.03033,
        0.00365,
        0.06062
    )
)

interaction_p_values
```

A data.frame: 7 × 2

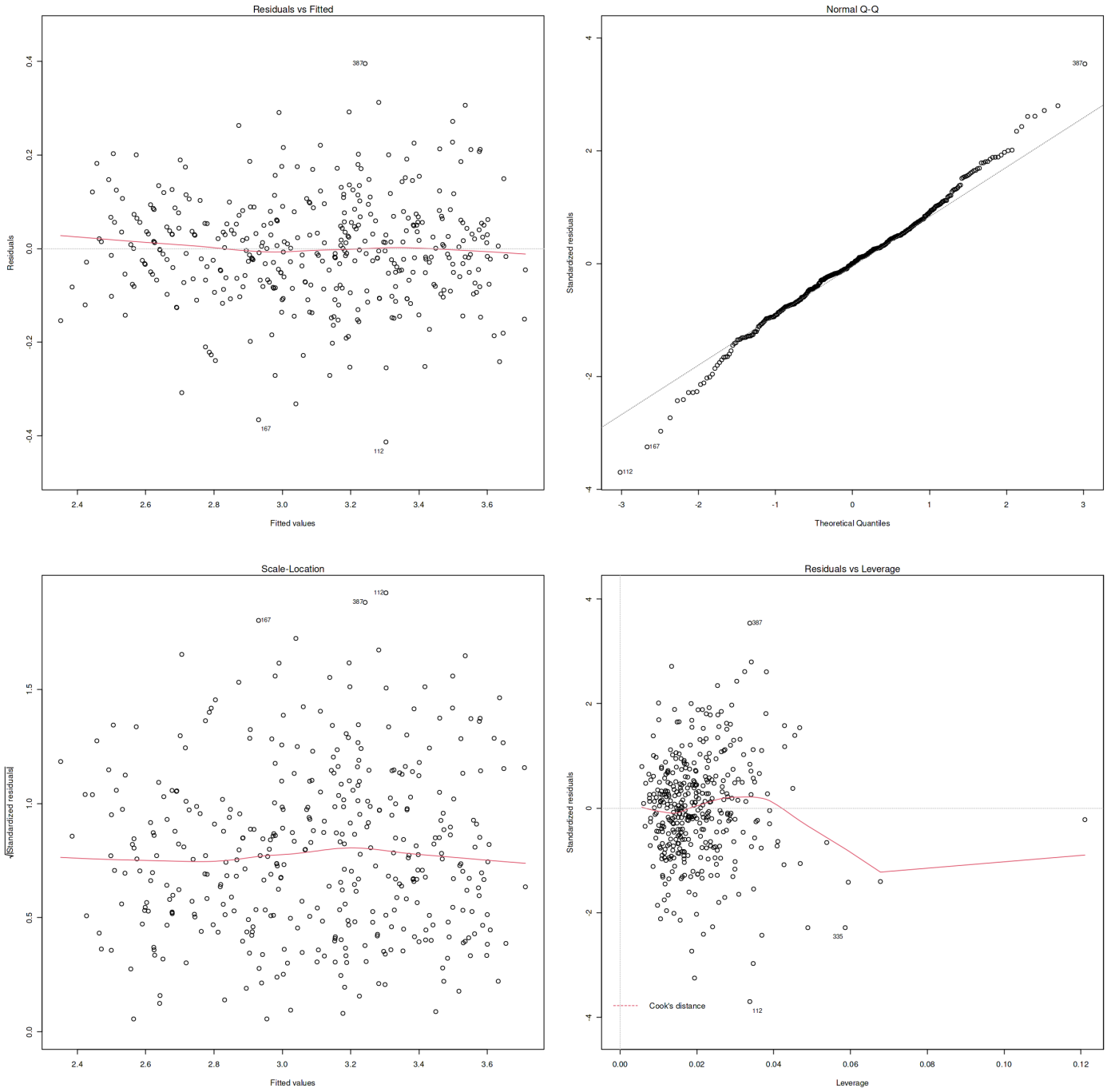
Interaction	P_value
<chr>	<dbl>
cylinders:acceleration	0.09584
cylinders:year	0.07389
displacement:weight	0.09342
displacement:year	0.01352
acceleration:year	0.03033
acceleration:origin	0.00365
year:origin	0.06062

Several interactions exhibit p-values close to 0 or near the common threshold of 0.05. These significant interactions suggest that the combined effect of the interacting variables on the response variable (assuming other variables are kept constant) is statistically significant and thus, should be incorporated into the model.

(f) Try a few different transformations of the variables, such aslog(X), √X, X2. Comment on your findings.

[126]:
```
x_auto = temp_auto
x_auto[] = lapply(Auto, function(x) if(is.numeric(x)) log(x) else x)

modelx  = lm(mpg ~ ., data = x_auto)
summary(modelx)
pval = pf(
    summary(modelx)$fstatistic[1],
    summary(modelx)$fstatistic[2],
    summary(modelx)$fstatistic[3],
    lower.tail = F
)
par(mfrow=c(2,2))
plot(modelx)
```

```
Warning message in `[<-.data.frame`(`*tmp*`, , value = list(mpg = c(2.89037175789616, :
"provided 9 variables to replace 8 variables"
Call:
lm(formula = mpg ~ ., data = x_auto)

Residuals:
     Min       1Q   Median       3Q      Max
-0.41298 -0.07098  0.00055  0.06150  0.39532

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.155391   0.648230  -0.240  0.81068
cylinders    -0.082815   0.061429  -1.348  0.17841
displacement  0.006625   0.056970   0.116  0.90748
horsepower   -0.294389   0.057652  -5.106 5.18e-07 ***
weight       -0.569666   0.082397  -6.914 1.98e-11 ***
acceleration -0.179239   0.059536  -3.011  0.00278 **
year          2.243989   0.131661  17.044  < 2e-16 ***
origin        0.044848   0.018821   2.383  0.01767 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1136 on 384 degrees of freedom
Multiple R-squared:  0.8903,    Adjusted R-squared:  0.8883
F-statistic: 445.3 on 7 and 384 DF,  p-value: < 2.2e-16
```



[127]:

```
x_auto = temp_auto
x_auto[] = lapply(Auto, function(x) if(is.numeric(x)) sqrt(x) else x)

modelx  = lm(mpg ~ ., data = x_auto)
summary(modelx)
pval = pf(
    summary(modelx)$fstatistic[1],
    summary(modelx)$fstatistic[2],
    summary(modelx)$fstatistic[3],
    lower.tail = F
)
```

```
par(mfrow=c(2,2))
plot(modelx)
```

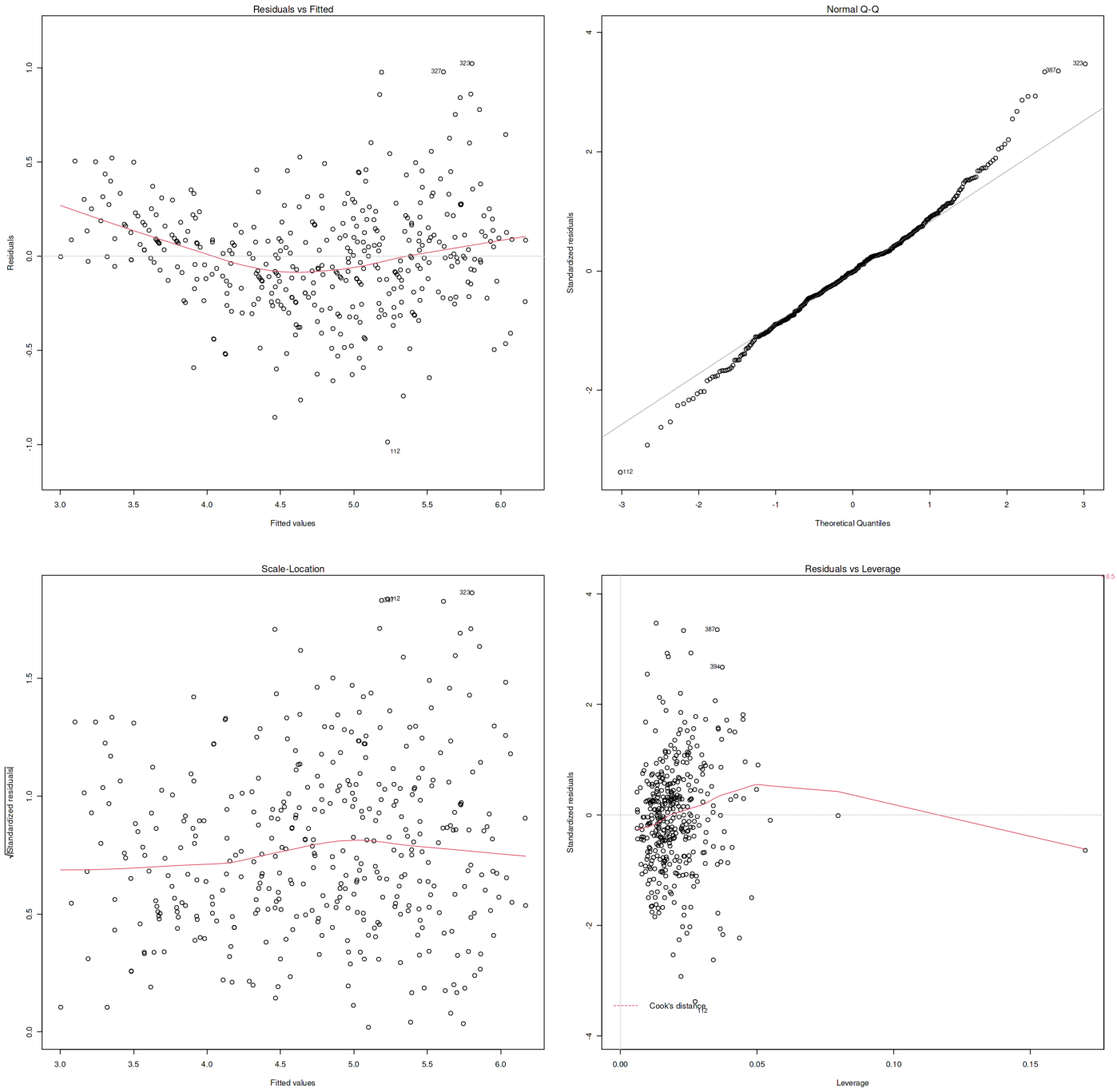
Warning message in `[<-.data.frame`(`*tmp*`, , value = list(mpg = c(4.24264068711928, :
"provided 9 variables to replace 8 variables"
Call:
lm(formula = mpg ~ ., data = x_auto)

Residuals:
    Min   1Q Median   3Q   Max
-0.98667 -0.17280 -0.00315 0.16145 1.02245

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.949286   0.847481 -2.300 0.021979 *
cylinders   -0.108552   0.141968 -0.765 0.444964
displacement 0.019707   0.021182   0.930 0.352752
horsepower   -0.090896   0.028428 -3.197 0.001502 **
weight   -0.061414   0.007292 -8.422 7.48e-16 ***
acceleration -0.107258   0.077048 -1.392 0.164699
year   1.266015   0.079308 15.963 < 2e-16 ***
origin   0.272324   0.070883   3.842 0.000143 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2964 on 384 degrees of freedom
Multiple R-squared: 0.8662,   Adjusted R-squared: 0.8638
F-statistic: 355.1 on 7 and 384 DF, p-value: < 2.2e-16



[128]:

```
x_auto = temp_auto
x_auto[] = lapply(Auto, function(x) if(is.numeric(x)) x**2 else x)

modelx  = lm(mpg ~ ., data = x_auto)
summary(modelx)
pval = pf(
    summary(modelx)$fstatistic[1],
    summary(modelx)$fstatistic[2],
    summary(modelx)$fstatistic[3],
    lower.tail = F
)
```

```
par(mfrow=c(2,2))
plot(modelx)
```

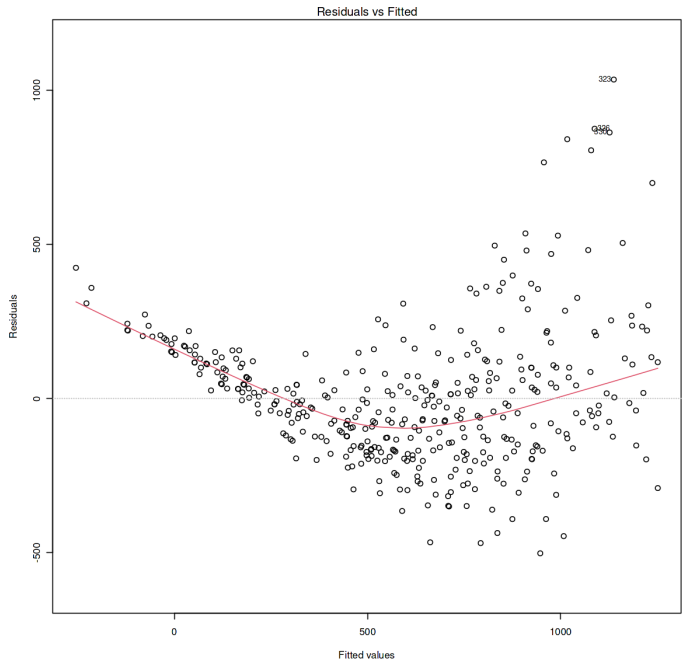
Warning message in `[<-.data.frame`(`*tmp*`, , value = list(mpg = c(324, 225, 324, :
"provided 9 variables to replace 8 variables"
Call:
lm(formula = mpg ~ ., data = x_auto)

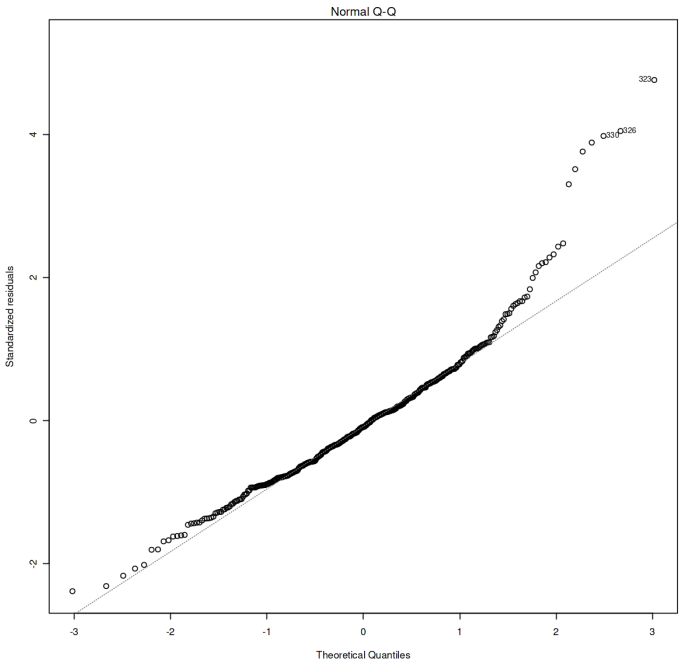
Residuals:
    Min   1Q Median   3Q   Max
-501.89 -145.36 -18.91 111.41 1034.08

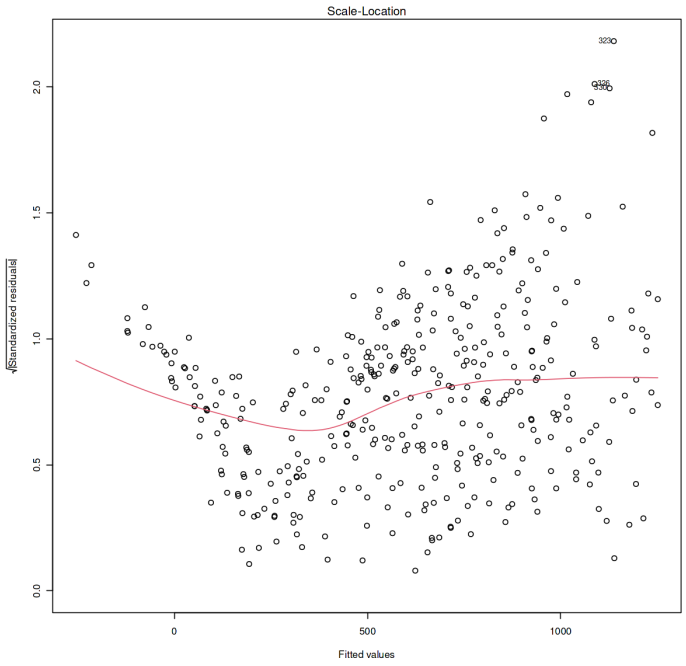
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.523e+02 1.456e+02 -5.165 3.87e-07 ***
cylinders   -3.746e+00 1.559e+00 -2.403 0.016713 *
displacement 3.356e-03 8.547e-04   3.926 0.000102 ***
horsepower   1.279e-04 3.076e-03   0.042 0.966851
weight   -4.833e-05 5.551e-06 -8.707 < 2e-16 ***
acceleration 4.892e-01 1.663e-01   2.941 0.003474 **
year   2.731e-01 2.183e-02 12.513 < 2e-16 ***
origin   2.608e+01 4.275e+00   6.101 2.57e-09 ***

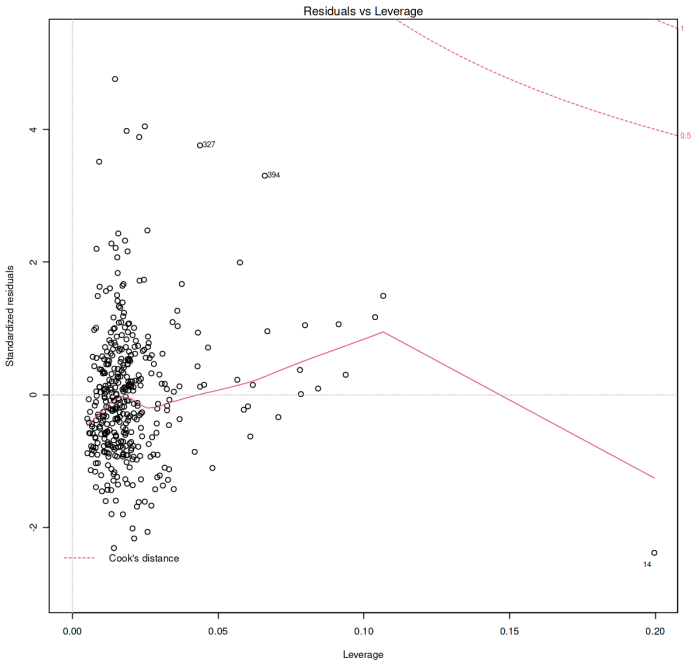
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 218.8 on 384 degrees of freedom
Multiple R-squared: 0.7055,   Adjusted R-squared: 0.7001
F-statistic: 131.4 on 7 and 384 DF, p-value: < 2.2e-16









[129]:

```
x_auto = temp_auto
x_auto[] = lapply(Auto, function(x) if(is.numeric(x)) x**3 else x)

modelx  = lm(mpg ~ ., data = x_auto)
summary(modelx)
pval = pf(
    summary(modelx)$fstatistic[1],
    summary(modelx)$fstatistic[2],
    summary(modelx)$fstatistic[3],
    lower.tail = F
)
```

```
par(mfrow=c(2,2))
plot(modelx)
```

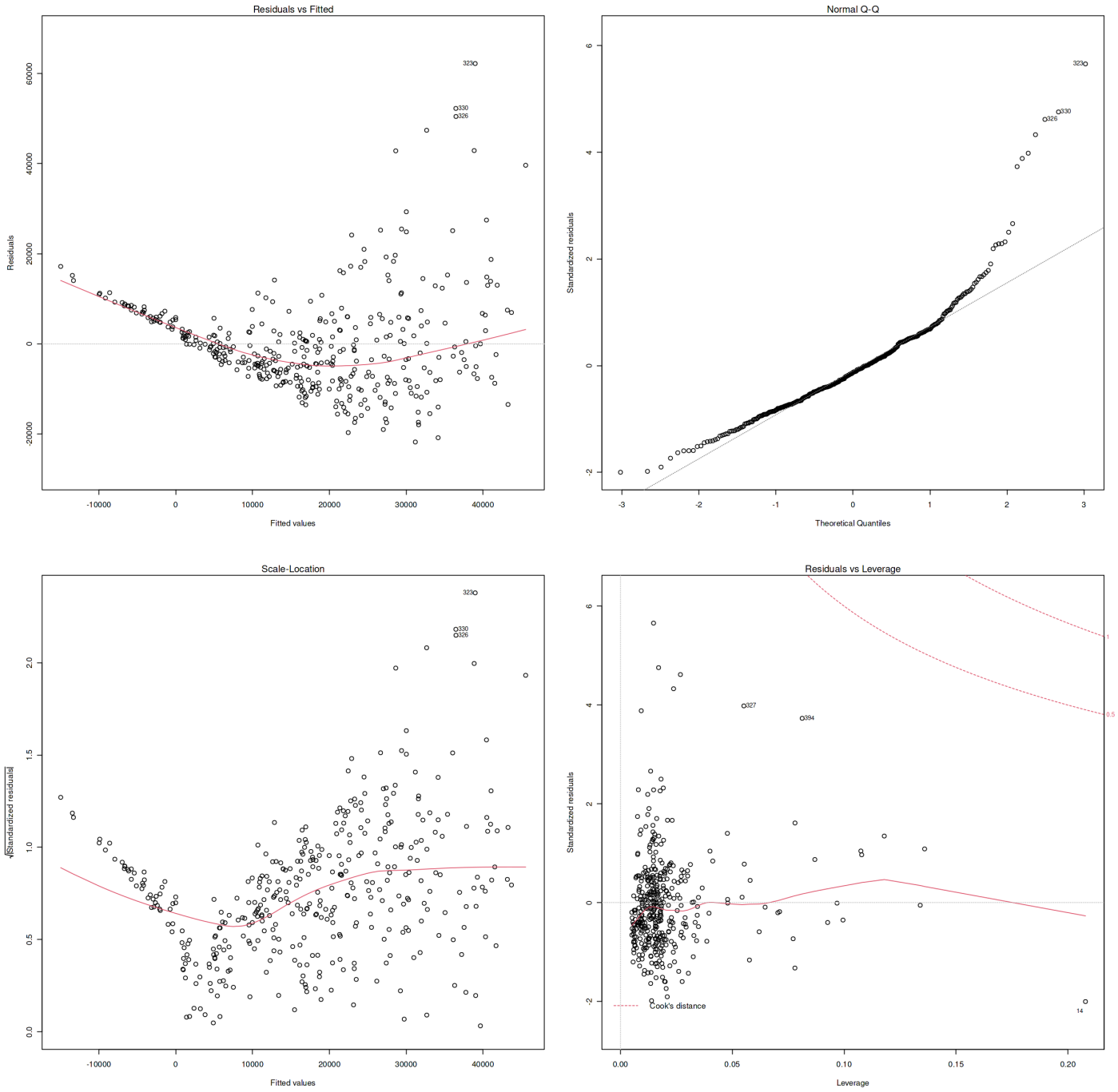
Warning message in `[<-.data.frame`(`*tmp*`, , value = list(mpg = c(5832, 3375, :
“provided 9 variables to replace 8 variables”
Call:
lm(formula = mpg ~ ., data = x_auto)

Residuals:
   Min   1Q Median   3Q   Max
-21793 -7195 -1400   5067 62213

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.726e+04 4.812e+03 -5.664 2.90e-08 ***
cylinders   -1.685e+01 7.298e+00 -2.308 0.021516 *
displacement 2.938e-04 8.944e-05   3.285 0.001113 **
horsepower   1.820e-04 6.479e-04   0.281 0.778916
weight   -2.800e-07 4.837e-08 -5.789 1.47e-08 ***
acceleration 1.099e+00 2.982e-01   3.686 0.000261 ***
year   1.030e-01 9.612e-03 10.712 < 2e-16 ***
origin   3.959e+02 6.383e+01   6.202 1.44e-09 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11080 on 384 degrees of freedom
Multiple R-squared: 0.5788,   Adjusted R-squared: 0.5711
F-statistic: 75.37 on 7 and 384 DF, p-value: < 2.2e-16



Depending on the interactions, we can observe that variables have varying effects on the responses, whether positive or negative. Let's consider the first interaction:

1. The coefficient for Horsepower is negative (-0.294389), indicating that as horsepower increases, mpg decreases.
2. The coefficient for Weight is negative (-0.569666), implying that as weight increases, mpg decreases.
3. The coefficient for Acceleration is negative (-0.179239), suggesting that as acceleration increases, mpg decreases.
4. The coefficient for Year is positive (2.243989), indicating that as the year of the car increases, mpg also increases.
5. The coefficient for Origin is positive (0.044848), suggesting that cars from a specific origin tend to have higher average mpg.

However, when we compare this to the last interaction:

1. The coefficient for Cylinders is negative (-16.85), implying that as the number of cylinders increases, mpg decreases.
2. The coefficient for Displacement is positive (0.0002938), indicating that as displacement increases, mpg also increases.
3. The coefficient for Weight is negative (-2.8e-07), suggesting that as weight increases, mpg decreases.
4. The coefficient for Acceleration is positive (1.099), indicating that as acceleration increases, mpg also increases.
5. The coefficient for Year is positive (0.103), suggesting that as the year of the car increases, mpg increases.
6. The coefficient for Origin is positive (395.9), indicating that cars from a specific origin tend to have higher average mpg.

This reveals that the choice of variables can lead to different outcomes for the response variable. It highlights the importance of understanding various situations and scenarios in which we would want to investigate the effects of different variables on the response