In [1]:
```
Workingdirectory = "C:/Users/Administrator/Documents/Master/MSIS-5223-70250 -
 Programming for Data Sci - 8282017 - 159 PM/Data for Tutorials and ICE/Data"
setwd(Workingdirectory)
data = paste(Workingdirectory,"\\ect_data.txt", sep ='')
df = read.table(data,header =T,sep ='\t')
```

In [3]:
```
df
```

attitude1_01	attitude1_02	attitude1_03	attitude1_04	intent1_01	intent1_02	intent1_03
6	6	6	6	6	6	6
7	7	7	7	7	7	7
6	6	6	6	7	7	7
7	7	7	7	6	6	7
7	7	6	6	7	7	7
6	6	5	5	6	7	7
6	5	5	6	6	6	6
7	7	7	7	7	7	7
7	7	6	6	7	7	7
5	4	4	5	6	5	6
7	6	7	7	7	7	7
7	7	7	7	7	7	7
5	6	4	3	6	6	6
7	7	5	6	7	7	7
6	6	6	6	6	6	6
6	6	5	5	7	6	7
6	6	6	6	7	7	7
7	7	6	6	7	7	7
3	3	4	5	3	3	3
7	7	6	7	7	7	7
6	5	6	5	5	6	5
7	6	4	5	6	6	6
6	7	5	6	6	6	7
6	6	6	6	7	7	7
7	7	7	7	5	5	7
6	6	6	6	6	6	6
5	5	4	4	5	5	6
7	7	7	7	7	7	7
6	5	7	7	7	6	7
7	7	7	7	7	7	7
...	...	...	...	...	...	...
5	5	4	4	5	5	5

attitude1_01	attitude1_02	attitude1_03	attitude1_04	intent1_01	intent1_02	intent1_03
6	6	6	6	6	6	6
7	7	6	6	6	6	6
7	7	7	7	7	7	7
7	7	7	7	7	7	7
6	6	6	6	4	4	4
7	7	7	7	6	6	6
6	6	5	6	6	6	6
7	7	7	7	7	7	7
7	7	7	7	7	7	7
6	6	6	6	7	7	7
6	6	6	6	7	7	7
5	5	5	5	3	3	4
6	7	6	6	6	6	6
6	6	5	5	6	6	7
7	7	7	7	7	7	7
7	5	4	5	7	6	7
6	6	6	6	5	5	5
7	7	7	7	7	7	7
7	7	7	7	7	7	7
5	6	5	5	6	6	6
6	5	6	6	6	5	6
7	7	6	6	7	7	7
7	7	6	6	7	7	7
5	7	6	7	7	7	7
6	6	6	7	7	6	7
5	6	6	6	7	7	7
6	5	6	6	5	6	6
6	6	6	6	6	7	6
7	7	6	7	7	7	7

In [1]:

In [17]:
```
names(df)
ncol(df)
```

'attitude1_01' 'attitude1_02' 'attitude1_03' 'attitude1_04' 'intent1_01' 'intent1_02' 'intent1_03' 'intent1_04' 'peruse01' 'peruse02' 'peruse03' 'peruse04' 'satis01' 'satis02' 'satis03' 'satis04'

16

In [2]:
```
reduction_data.pca = df[c('attitude1_01', 'attitude1_02', 'attitude1_03' ,'att
itude1_04' ,'intent1_01', 'intent1_02' ,'intent1_03' ,'intent1_04' ,'peruse01'
 ,'peruse02' ,'peruse03' ,'peruse04' ,'satis01', 'satis02', 'satis03', 'satis0
4')]
```

In [3]:
```
pcamodel_reduc = princomp(reduction_data.pca, cor=FALSE)
```

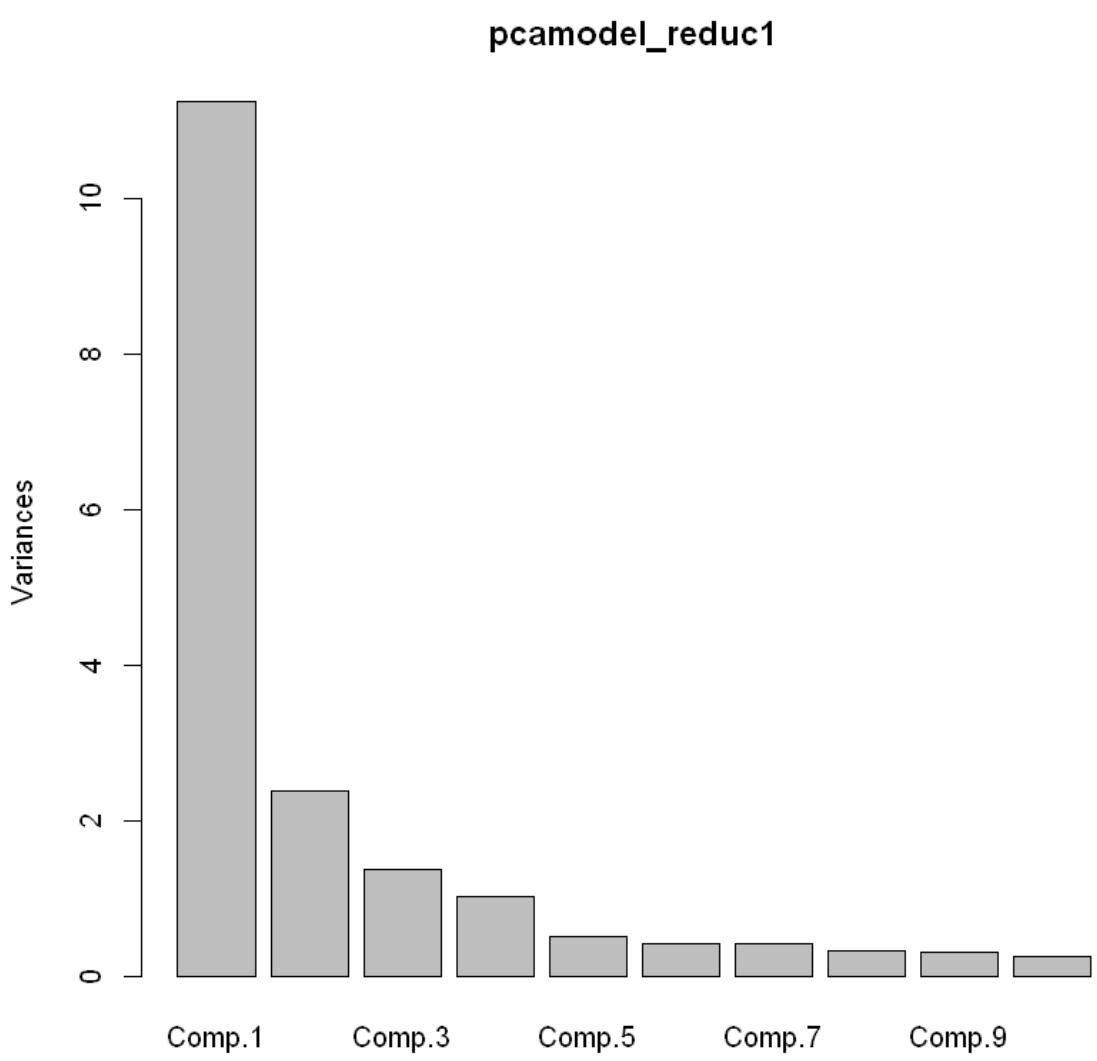
In [ ]:
```
#There are four components have a value greater than 1.0. Truly I am dealing w
ith 4 columns of data out of total of 16
```

In [4]:
```
pcamodel_reduc$sdev^2
```

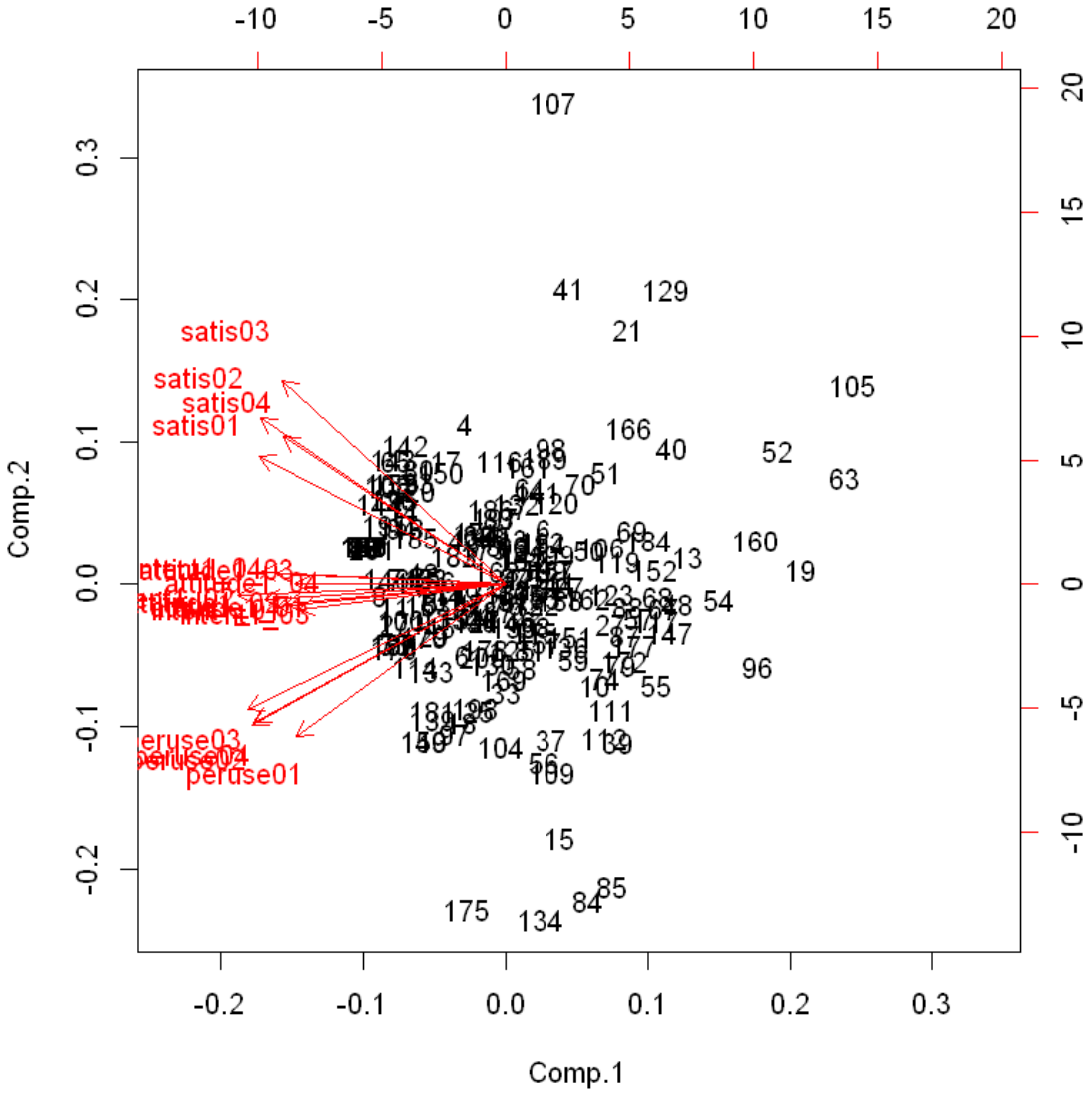
Comp.1	11.2597026603237
Comp.2	2.38326747224322
Comp.3	1.38374333827987
Comp.4	1.03135256437065
Comp.5	0.520880250139964
Comp.6	0.429449980794813
Comp.7	0.420624946685233
Comp.8	0.328492300505209
Comp.9	0.307023721729649
Comp.10	0.260842676791779
Comp.11	0.211696555770928
Comp.12	0.1760159163481
Comp.13	0.167807288994348
Comp.14	0.137635111845327
Comp.15	0.123826469302077
Comp.16	0.100924803150985

In [28]:
```
pcamodel_reduc1 = princomp(reduction_data.pca, cor=FALSE)
```

In [29]:
```
plot(pcamodel_reduc1)
```



In [30]:
```
biplot(pcamodel_reduc1)
```



In [31]:
```
reduction_data.FA = factanal(~attitude1_01+attitude1_02+attitude1_03+attitude1
_04+intent1_01+intent1_02+intent1_03+intent1_04+peruse01+peruse02+peruse03+per
use04+satis01+satis02+satis03+satis04,factors=4,rotation="varimax",
  scores="none",data=df)
reduction_data.FA
```

```
Call:
factanal(x = ~attitude1_01 + attitude1_02 + attitude1_03 + attitude1_04 +
intent1_01 + intent1_02 + intent1_03 + intent1_04 + peruse01 +     peruse02 +
peruse03 + peruse04 + satis01 + satis02 + satis03 +     satis04, factors = 4,
data = df, scores = "none", rotation = "varimax")

Uniquenesses:
attitude1_01 attitude1_02 attitude1_03 attitude1_04   intent1_01   intent1_02

       0.262        0.359        0.071        0.210        0.139        0.081

  intent1_03   intent1_04     peruse01     peruse02     peruse03     peruse04

       0.210        0.283        0.288        0.283        0.189        0.237

     satis01      satis02      satis03      satis04
       0.136        0.111        0.303        0.297

Loadings:
             Factor1 Factor2 Factor3 Factor4
attitude1_01 0.307   0.412   0.443   0.527
attitude1_02 0.292   0.369   0.473   0.441
attitude1_03 0.301   0.270   0.238   0.842
attitude1_04 0.282   0.291   0.252   0.750
intent1_01   0.267   0.319   0.809   0.184
intent1_02   0.255   0.254   0.847   0.267
intent1_03   0.258   0.353   0.760   0.145
intent1_04   0.361   0.280   0.633   0.328
peruse01             0.762   0.309   0.175
peruse02     0.205   0.761   0.242   0.192
peruse03     0.215   0.788   0.303   0.231
peruse04     0.199   0.798   0.189   0.225
satis01      0.833   0.302   0.230   0.160
satis02      0.874   0.195   0.201   0.216
satis03      0.778           0.229   0.186
satis04      0.743   0.159   0.259   0.244

               Factor1 Factor2 Factor3 Factor4
SS loadings      3.432   3.409   3.385   2.315
Proportion Var   0.214   0.213   0.212   0.145
Cumulative Var   0.214   0.428   0.639   0.784

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 132.27 on 62 degrees of freedom.
The p-value is 5.26e-07
```

In [2]:
```
#yes: peruse01 and satis03
#factor 1: retains satis01, satis02,satis04, satis03
#factor2:retains peruse01:peruse04
#factor3:intent1_01:intent1_04
#factor4: attitude1_01:attitude1_04
#because they have high percentage presentative for each factor
```

In [5]:
```
Workingdirectory1 = "C:/Users/Administrator/Documents/Master/MSIS-5223-70250 -
 Programming for Data Sci - 8282017 - 159 PM/Data for Tutorials and ICE/Data"
setwd(temptable)
template = paste(Workingdirectory1,"\\car.test.frame.txt", sep ='')
taxon_data = read.table(template,header =T,sep ='\t')
```

In [6]:
```
df2 <- na.omit(taxon_data)
df2
```

	Price	Country	Reliability	Mileage	Type	Weight	Disp.	HP
1	8895	USA	4	33	Small	2560	97	113
2	7402	USA	2	33	Small	2345	114	90
3	6319	Korea	4	37	Small	1845	81	63
4	6635	Japan/USA	5	32	Small	2260	91	92
5	6599	Japan	5	32	Small	2440	113	103
6	8672	Mexico	4	26	Small	2285	97	82
7	7399	Japan/USA	5	33	Small	2275	97	90
8	7254	Korea	1	28	Small	2350	98	74
9	9599	Japan	5	25	Small	2295	109	90
11	8748	Japan/USA	5	29	Small	2390	97	102
12	6488	Japan	5	35	Small	2075	89	78
13	9995	Germany	3	26	Small	2330	109	100
14	11545	USA	1	20	Sporty	3320	305	170
15	9745	USA	1	27	Sporty	2885	153	100
16	12164	USA	1	19	Sporty	3310	302	225
17	11470	USA	3	30	Sporty	2695	133	110
18	9410	Japan	5	33	Sporty	2170	97	108
19	13945	Japan	5	27	Sporty	2710	125	140
20	13249	Japan	3	24	Sporty	2775	146	140
24	10565	USA	2	23	Compact	2640	151	110
25	10320	USA	1	26	Compact	2655	133	95
26	10945	USA	4	25	Compact	3065	181	141
27	9483	USA	2	24	Compact	2750	141	98
28	12145	Japan/USA	5	26	Compact	2920	132	125
29	12459	Japan/USA	4	24	Compact	2780	133	110
30	10989	Japan	5	25	Compact	2745	122	102
31	17879	Japan	4	21	Compact	3110	181	142
32	11650	Japan	5	21	Compact	2920	146	138
33	9995	USA	2	23	Compact	2645	151	110
35	11499	Japan/USA	5	23	Compact	2935	135	130
36	11588	Japan/USA	5	27	Compact	2920	122	115
37	18450	Sweden	3	23	Compact	2985	141	114

	Price	Country	Reliability	Mileage	Type	Weight	Disp.	HP
38	24760	Japan	5	20	Medium	3265	163	160
39	13150	USA	3	21	Medium	2880	151	110
40	12495	USA	2	22	Medium	2975	153	150
41	16342	USA	3	22	Medium	3450	202	147
42	15350	USA	2	22	Medium	3145	180	150
43	13195	USA	3	22	Medium	3190	182	140
44	14980	USA	1	23	Medium	3610	232	140
46	23300	Japan	5	21	Medium	3480	180	158
47	17899	Japan	5	22	Medium	3200	180	160
48	13150	USA	2	21	Medium	2765	151	110
50	21498	Japan	3	23	Medium	3480	180	190
51	16145	USA	3	23	Large	3325	231	165
52	14525	USA	1	18	Large	3855	305	170
53	17257	USA	3	20	Large	3850	302	150
55	15395	USA	3	18	Van	3735	202	150
56	12267	USA	3	18	Van	3665	182	145
57	14944	Japan	5	19	Van	3735	181	150

In [11]:
```
library(tree)
```

```
Error in library(tree): there is no package called 'tree'
Traceback:

1. library(tree)
2. stop(txt, domain = NA)
```

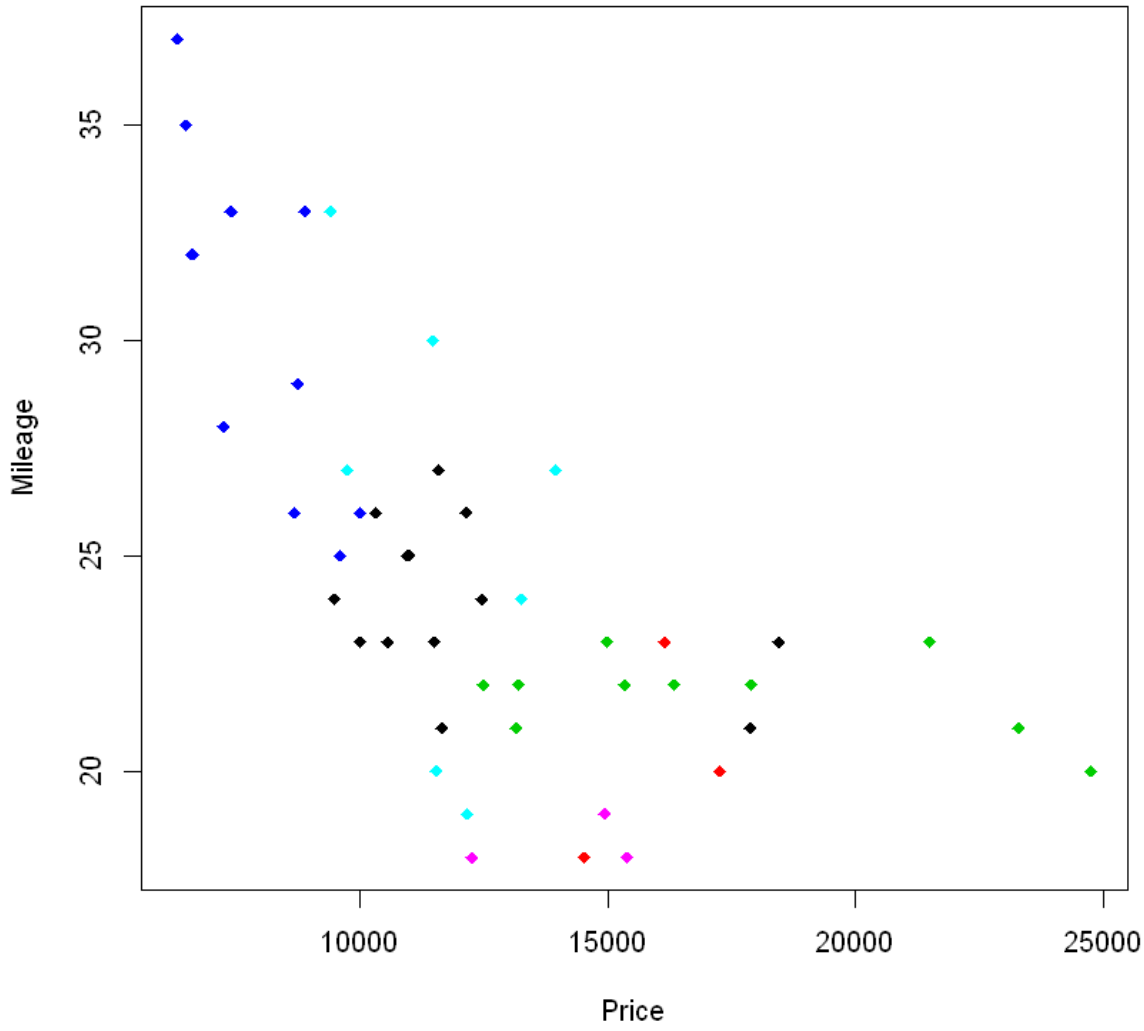
In [45]:
```
attach(df2)
```

```
The following objects are masked from df2 (pos = 5):

    Country, Disp., HP, Mileage, Price, Reliability, Type, Weight

The following objects are masked from df2 (pos = 6):

    Country, Disp., HP, Mileage, Price, Reliability, Type, Weight
```

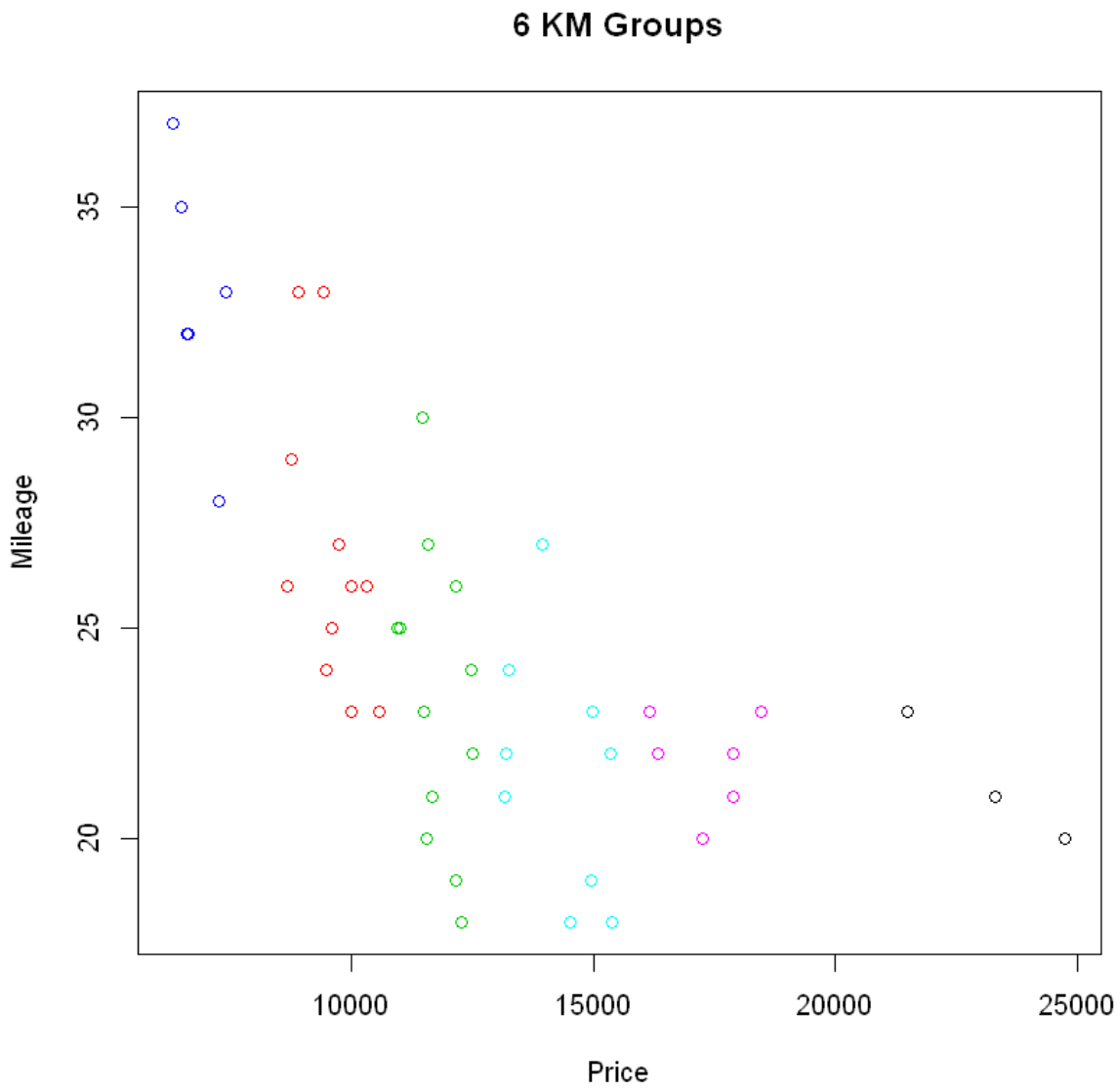
In [16]:
```
unique(Type)
```

Small Sporty Compact Medium Large Van

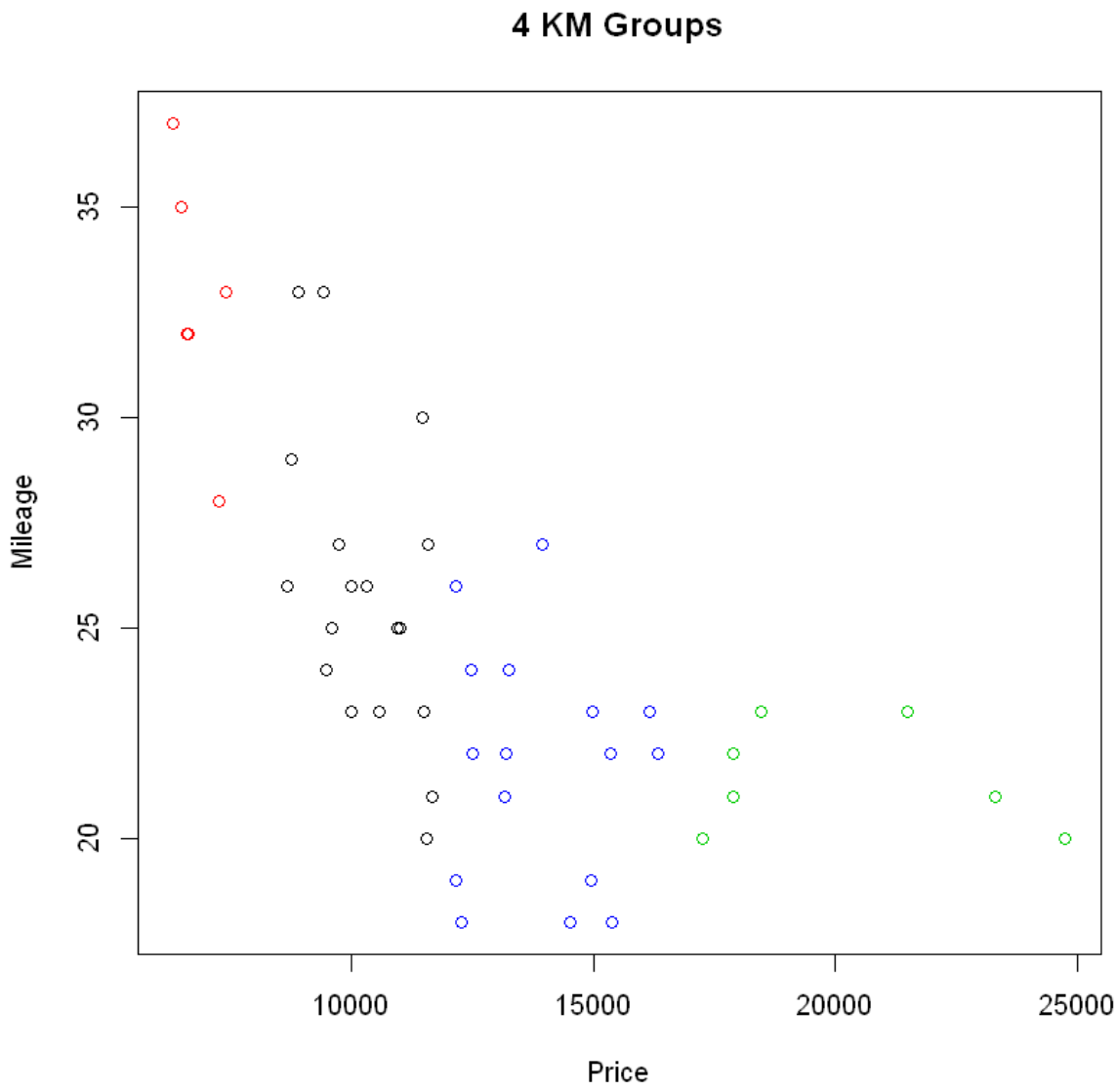
In [17]:
```
plot(Price,Mileage, pch =18, col =Type)
```



In [19]:
```
#### Now, let us use the K-Means algorithm to plot the data.
#### Restrict the data to 6 distinct groups
km = kmeans(data.frame(Price, Mileage), 6)
plot(Price, Mileage, col=km[[1]], main="6 KM Groups")
```



In [24]:
```
#### Now, let us use the K-Means algorithm to plot the data.
#### Restrict the data to 6 distinct groups
km = kmeans(data.frame(Price, Mileage), 4)
plot(Price, Mileage, col=km[[1]], main="4 KM Groups")
```



In [22]:
```
#### Allow the R Graphics window to display 4 graphics
#### with 2 on top and 2 on the bottom:
par(mfrow=c(2,2))
```

In [26]:
```
table(km[[1]], Type)
```

```
   Type
    Compact Large Medium Small Sporty Van
1        9     0      0     5      4   0
2        0     0      0     7      0   0
3        2     1      4     0      0   0
4        2     2      7     0      3   3
```

In [28]:
```
library(psych)
```

In [40]:
```
library(cluster)
```

In [47]:
```
k.means.fit <- kmeans(Type, 4)
k.means.fit$cluster
```

```
Warning message in storage.mode(x) <- "double":
"NAs introduced by coercion"
```

```
Error in kmeans(Type, 4): more cluster centers than distinct data points.
Traceback:

1. kmeans(Type, 4)
2. stop("more cluster centers than distinct data points.")
```

In [36]:
```
km_tax
```

```
Error in eval(expr, envir, enclos): object 'km_tax' not found
Traceback:
```