Final Paper

AUTHOR
Giovanni Rivera

PUBLISHED
August 23, 2023

Obtaining the data

```
if (!require(data.table)) {
  install.packages("data.table")
  library(data.table)
}
df <- fread("vehicles.csv")
```

Data Dictionary

Columns in the vehicles data set

position	name	description
1	id	entry ID
2	url	listing URL
3	region	craigslist region
4	region_url	region URL
5	price	entry price
6	year	entry year
7	manufacturer	manufacturer of vehicle
8	model	model of vehicle
9	condition	condition of vehicle
10	cylinders	number of cylinders
11	fuel	fuel type
12	odometer	miles traveled by vehicle
13	title_status	title status of vehicle
14	transmission	transmission of vehicle

position	name	description
15	vin	vehicle identification number
16	drive	type of drive
17	size	size of vehicle
18	type	generic type of vehicle
19	paint_color	color of vehicle
20	image_url	image URL
21	description	listed description of vehicle
22	county	useless column left in by mistake
23	state	state of listing
24	lat	latitude of listing
25	long	longitude of listing
26	posting_date	date of craigslist listing

Data Description

Part 1: Numerical Description

```
names(df)
```

```
 [1] "id"           "url"          "region"       "region_url"   "price"
 [6] "year"         "manufacturer" "model"        "condition"    "cylinders"
[11] "fuel"         "odometer"     "title_status" "transmission" "VIN"
[16] "drive"        "size"         "type"         "paint_color"  "image_url"
[21] "description"  "county"       "state"        "lat"          "long"
[26] "posting_date"
```

```
library(tidyverse)
df <- df |> select(-lat,-long,-id,-url,-region_url,-VIN)
```

```
names(df)
```

```
 [1] "region"       "price"        "year"         "manufacturer" "model"
 [6] "condition"    "cylinders"    "fuel"         "odometer"     "title_status"
```

```
[11] "transmission" "drive"       "size"         "type"         "paint_color"
[16] "image_url"    "description" "county"       "state"        "posting_date"
```

```
str(df)
```

```
Classes 'data.table' and 'data.frame':	426880 obs. of  20 variables:
 $ region      : chr  "prescott" "fayetteville" "florida keys" "worcester / central MA"
...
 $ price       :integer64 6000 11900 21000 1500 4900 1600 1000 15995 ...
 $ year        : int  NA NA NA NA NA NA NA NA NA NA ...
 $ manufacturer: chr  "" "" "" "" ...
 $ model       : chr  "" "" "" "" ...
 $ condition   : chr  "" "" "" "" ...
 $ cylinders   : chr  "" "" "" "" ...
 $ fuel        : chr  "" "" "" "" ...
 $ odometer    : int  NA NA NA NA NA NA NA NA NA NA ...
 $ title_status: chr  "" "" "" "" ...
 $ transmission: chr  "" "" "" "" ...
 $ drive       : chr  "" "" "" "" ...
 $ size        : chr  "" "" "" "" ...
 $ type        : chr  "" "" "" "" ...
 $ paint_color : chr  "" "" "" "" ...
 $ image_url   : chr  "" "" "" "" ...
 $ description : chr  "" "" "" "" ...
 $ county      : logi  NA NA NA NA NA NA ...
 $ state       : chr  "az" "ar" "fl" "ma" ...
 $ posting_date: POSIXct, format: NA NA ...
 - attr(*, ".internal.selfref")=<externalptr>
```

```
df$state <-as.factor(df$state)
```

```
str(df)
```

```
Classes 'data.table' and 'data.frame':	426880 obs. of  20 variables:
 $ region      : chr  "prescott" "fayetteville" "florida keys" "worcester / central MA"
...
 $ price       :integer64 6000 11900 21000 1500 4900 1600 1000 15995 ...
 $ year        : int  NA NA NA NA NA NA NA NA NA NA ...
 $ manufacturer: chr  "" "" "" "" ...
 $ model       : chr  "" "" "" "" ...
 $ condition   : chr  "" "" "" "" ...
 $ cylinders   : chr  "" "" "" "" ...
 $ fuel        : chr  "" "" "" "" ...
 $ odometer    : int  NA NA NA NA NA NA NA NA NA NA ...
 $ title_status: chr  "" "" "" "" ...
 $ transmission: chr  "" "" "" "" ...
 $ drive       : chr  "" "" "" "" ...
 $ size        : chr  "" "" "" "" ...
 $ type        : chr  "" "" "" "" ...
```

```
 $ paint_color : chr  "" "" "" "" ...
 $ image_url   : chr  "" "" "" "" ...
 $ description : chr  "" "" "" "" ...
 $ county      : logi  NA NA NA NA NA NA ...
 $ state       : Factor w/ 51 levels "ak","al","ar",..: 4 3 10 20 28 35 35 35 38 39 ...
 $ posting_date: POSIXct, format: NA NA ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Manufactures & Paint Color

This contingency table demonstrates the frequencies of colors for each vehicle, by manufacturer.

Additionally, it further highlights the most popular colors being white and black across all manufacturers.

```r
with(df,addmargins(table(paint_color,manufacturer)))
```

```
           manufacturer
paint_color        acura alfa-romeo aston-martin   audi    bmw  buick cadillac
              6051  1689        203            9   2067   3789   1799     2059
     black    1616  1004        217            5   1934   3806    675     1659
     blue     1160   307        164            3    719   1439    292      288
     brown     182    73          9            0     65    157    235      110
     custom    282    74         17            0     44    137    128      143
     green     597    26          4            2     51     97     57       39
     grey      687   367         11            1    563    892    218      224
     orange    155     1          1            0      4     36      2        5
     purple     67     2          0            0      2     13     18       10
     red      1642   239         94            0    185    376    508      435
     silver   1086   915         28            2    955   1501    550      686
     white    3785  1279        149            2    974   2436   1013     1275
     yellow    336     2          0            0     10     20      6       20
     Sum     17646  5978        897           24   7573  14699   5501     6953
           manufacturer
paint_color chevrolet chrysler datsun  dodge ferrari   fiat   ford    gmc
                16989     2160     25   4468      40    314  21549   5468
     black       7225      813      2   2253       9     89   8573   3005
     blue        3660      512      9    944       0     31   4711    870
     brown       1016       94      3     83       0     19    917    441
     custom       839      130      2    186       2     33   1173    240
     green        744       58      2    352       0     32   1295    144
     grey        2437      380      2    916       2     36   3282    670
     orange       251        4      4    215       0      4    330     12
     purple        95        9      0     70       0      0     77     13
     red         5281      336      7   1098      30     71   5167    979
     silver      4927      800      3   1087       3     24   4929   1078
     white      11075      713      4   1980       8    133  18609   3804
     yellow       525       22      0     55       1      6    373     61
     Sum        55064     6031     63  13707      95    792  70985  16785
           manufacturer
paint_color harley-davidson  honda hyundai infiniti jaguar   jeep    kia
```

```
                    40  6421    2982     1345    448   6107   2643
     black          86  2753    1242     1070    518   3266   1246
     blue            5  2106    1170      639    207   1008    494
     brown           0   414     155      178     16    249    221
     custom          4   440     167       74     19    313    146
     green           0   374      67       30     88    688    225
     grey            3  2111     733      280     49    975    490
     orange          2    82      63        2      0    195     38
     purple          0    75      11        3      1     24     42
     red             4  1134     805       97     85   1612    660
     silver          7  2990    1403      477    191   1904   1082
     white           0  2328    1530      591    322   2459   1154
     yellow          2    41      10       16      2    214     16
     Sum           153 21269   10338     4802   1946  19014   8457
           manufacturer
paint_color land rover  lexus lincoln  mazda mercedes-benz mercury   mini
                    12   2439    1323   1726          2933     409    603
     black           1   1251     912    827          3049      91    289
     blue            1    354     345    567           645     128    373
     brown           0    139      53     42           154      36     53
     custom          0    136      61     48           126      40     36
     green           1     81      39     67            48      57    119
     grey            3    472      94    441           632      53     85
     orange          0      4       2      6             4       1     30
     purple          0      1       1     10             8       2      3
     red             1    409     254    604           393      97    287
     silver          1   1377     396    461          1561     134    145
     white           1   1523     732    622          2235     130    322
     yellow          0     14       8      6            29       6     31
     Sum            21   8200    4220   5427         11817    1184   2376
           manufacturer
paint_color mitsubishi morgan nissan pontiac porsche    ram  rover saturn
                   828      1   5618     820     483   6500    623    395
     black         414      0   2805     208     250   2350    594    108
     blue          220      0   1183     183      99    924     70    115
     brown         128      1    295      32      26    175     18     18
     custom         30      0    289      47      12    240     32     21
     green          97      0    142      59       8    152     63     48
     grey          154      0   1656     125      75    832    117     47
     orange        190      0     45      22       3     65      3     13
     purple          3      0     43       7       2     18      0      2
     red           332      0   1354     290      58   1135     74    135
     silver        318      0   2438     228     133   1153    108     96
     white         570      1   3164     234     220   4771    405     83
     yellow          8      0     35      33      15     27      6      9
     Sum          3292      3  19067    2288    1384  18342   2113   1090
           manufacturer
paint_color subaru  tesla toyota volkswagen  volvo    Sum
              2858    215  10066       2494   1192 130203
     black     807     72   3584       1671    512  62861
     blue     1508     81   2312       1053    324  31223
```

brown	102	2	556	77	49	6593
custom	137	1	698	95	58	6700
green	422	3	820	102	43	7343
grey	693	28	2748	646	186	24416
orange	71	0	53	63	3	1984
purple	9	0	37	8	1	687
red	553	30	2853	610	159	30473
silver	1217	41	5120	1031	384	42970
white	1107	394	5267	1424	457	79285
yellow	11	1	88	71	6	2142
Sum	9495	868	34202	9345	3374	426880

Title Status and Contion

This contingency table demonstrates the frequencies of title status's of each vehicle based on their condition.

A vehicle's title status could directly correlate and be impacted by its condition. The most common title status is a clean status followed by a rebuilt status.

```
with(df,addmargins(table(condition,title_status)))
```

	title_status							
condition		clean	lien	missing	parts only	rebuilt	salvage	Sum
	2759	167445	107	299	70	2048	1376	174104
excellent	5369	91734	586	57	18	2823	880	101467
fair	0	6156	47	188	27	142	209	6769
good	114	118461	389	208	29	1362	893	121456
like new	0	19870	265	8	9	791	235	21178
new	0	1226	25	5	4	30	15	1305
salvage	0	225	3	49	41	23	260	601
Sum	8242	405117	1422	814	198	7219	3868	426880

Title Status and State

This contingency table demonstrates the frequency of title status's by State.

Across all states the most common title status's are clean and rebuilt.

```
with(df,addmargins(table(state,title_status)))
```

	title_status							
state		clean	lien	missing	parts only	rebuilt	salvage	Sum
ak	0	3313	30	6	2	112	11	3474
al	153	4668	20	7	1	101	5	4955
ar	13	3911	8	11	0	83	12	4038
az	145	8285	43	17	4	108	77	8679
ca	1620	47512	97	112	32	181	1060	50614
co	131	10694	53	17	3	127	63	11088
ct	10	5097	5	25	2	44	5	5188

dc	26	2899	4	7	2	18	14	2970
de	0	932	3	2	0	10	2	949
fl	493	27331	67	51	13	500	56	28511
ga	21	6887	26	17	2	33	17	7003
hi	91	2823	6	4	1	29	10	2964
ia	292	7380	25	5	3	764	163	8632
id	59	8603	35	10	5	188	61	8961
il	187	10009	26	12	0	129	24	10387
in	0	5509	19	5	1	157	13	5704
ks	78	5908	20	8	7	145	43	6209
ky	10	3880	16	6	2	217	18	4149
la	20	3070	12	4	0	62	28	3196
ma	138	7835	11	16	6	149	19	8174
md	26	4666	12	7	5	33	29	4778
me	1	2900	16	21	3	20	5	2966
mi	1036	15404	91	22	4	251	92	16900
mn	71	7212	32	18	6	228	149	7716
mo	1	4167	23	16	2	62	22	4293
ms	26	960	9	3	0	13	5	1016
mt	78	6069	47	20	3	44	33	6294
nc	606	13953	60	29	14	338	277	15277
nd	0	403	3	0	2	0	2	410
ne	1	973	4	3	0	23	32	1036
nh	2	2922	17	20	4	13	3	2981
nj	26	9511	8	13	3	130	51	9742
nm	23	4193	21	16	2	20	150	4425
nv	16	3073	9	9	0	65	22	3194
ny	79	18986	87	38	7	114	75	19386
oh	449	16735	36	8	7	363	98	17696
ok	1	6647	36	9	1	76	22	6792
or	192	16513	29	30	3	217	120	17104
pa	290	12902	20	12	9	294	226	13753
ri	0	2279	3	15	5	11	7	2320
sc	66	5975	25	4	2	60	195	6327
sd	0	1216	13	4	0	15	54	1302
tn	265	10301	34	24	8	380	54	11066
tx	112	21796	66	70	15	638	248	22945
ut	20	995	6	1	0	99	29	1150
va	181	10357	59	7	2	106	20	10732
vt	18	2417	9	28	1	35	5	2513
wa	1026	12584	22	12	1	174	42	13861
wi	143	10869	93	10	2	208	73	11398
wv	0	1015	2	2	1	11	21	1052
wy	0	578	4	1	0	21	6	610
Sum	8242	405117	1422	814	198	7219	3868	426880

```r
library(tidyverse)
df <- df[df$price<100000&df$price>0,]
dfd<-df |> select("price", "cylinders", "odometer", "size", "manufacturer")
dfd$size<-as.factor(dfd$size)
```

```
dfd$price<-as.factor(dfd$price)
dfd$odometer<-as.factor(dfd$odometer)
dfd$cylinders<-as.factor(dfd$cylinders)
```

Summary of price ranging from 1 to 99999.

```
df$price <- as.integer(df$price)
summary(df$price)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1    7000   15000   18709   27590   99999 
```

Before sorting the price range to be between 0 and 100000 the outrageously high prices dragged the mean above the median, however on setting a price range, the mean is now closer to the median and they provide a certain amount of accuracy.The mean here is 18709 and the median is 15000.

Stem plot of odometer.

```
dfd$odometer<- as.numeric(dfd$odometer)
stem(dfd$odometer[1:100])
```

```

  The decimal point is 4 digit(s) to the right of the |

  0 | 0111333344444555566677778888899
  1 | 02333444555578888999
  2 | 0111222888
  3 | 025669
  4 | 15699
  5 | 002
  6 | 7
  7 | 5
  8 | 6
  9 | 0
```

Count of cylinders

```
dfd |> count(cylinders,sort=TRUE)
```

```
      cylinders      n
1:               160154
2:   6 cylinders  88781
3:   4 cylinders  72995
4:   8 cylinders  66484
5:   5 cylinders   1668
6:  10 cylinders   1344
7:         other   1081
8:   3 cylinders    611
9:  12 cylinders    170
```

This representation shows that majority of the users did not fill in an answer for cylinders, however of those that did, “6 cylinders” are the most popular.

Contingency table between manufacturer and cylinders.

```
dfA<- subset (df, manufacturer %in% c("ford", "honda", "toyota", "chevrolet", "nissan"))
dfB<- subset (dfA, cylinders %in% c("10 cylinders", "12 cylinders", "3 cylinders", "4 cyl
tbl<- table(dfA$manufacturer, dfA$cylinders)
addmargins(tbl)
```

```
                  10 cylinders 12 cylinders 3 cylinders 4 cylinders
  chevrolet  18602           28            2          39        6414
  ford       26028          929            3         119        7514
  honda       7386            7            0          47        8686
  nissan      6600           26            0           5        5571
  toyota     11585            8            0          10        9067
  Sum        70201          998            5         220       37252

             5 cylinders 6 cylinders 8 cylinders  other    Sum
  chevrolet          144        7215       18001     91  50536
  ford                29       12922       17144    125  64813
  honda               14        3685          12     29  19866
  nissan               3        3998        1098     90  17391
  toyota               9        8092        2788     37  31596
  Sum                199       35912       39043    372 184202
```

On observing the cylinders of the most popular manufacturers, we can see that chevrolet and ford have 8 cylinders as their most produced whereas honda, nissan, and toyota have 4 cylinders as their most produced.

Contingency table between size and manufacturer.

```
dfX<- subset (df, size %in% c("compact", "full-size", "mid-size", "sub-compact"))
dfY<- subset (dfX, manufacturer %in% c("ford", "honda", "toyota", "chevrolet", "nissan"))
tbl<- table(dfY$size, dfY$manufacturer)
addmargins(tbl)
```

```
               chevrolet  ford honda nissan toyota   Sum
  compact           1685  1579  1650   1063   1631  7608
  full-size        10351 13748  2284   1969   3785 32137
  mid-size          3032  3835  2614   2097   3615 15193
  sub-compact        237   496   302    101    183  1319
  Sum              15305 19658  6850   5230   9214 56257
```

The entire data set has too many manufacturers to clearly analyze the data so I’ve chosen 5 of the most popular manufacturers. On short-listing the most popular manufacturers and comparing their respective sizes, it is clear to see that the most popular size in Chevrolet, Ford, and Toyota is “full-size” whereas in Honda and Nissan it is “mid-size” .

Number of vehicles from each year

```
df |> count(year,sort=TRUE)
```

```
     year     n
  1: 2018 32563
  2: 2017 32463
  3: 2013 28188
  4: 2015 27945
  5: 2016 27376
 ---
109: 1905     1
110: 1909     1
111: 1915     1
112: 1918     1
113: 1943     1
```

Number of Vehicles from each manufacturer

```
df |> count(manufacturer,sort=TRUE)
```

```
     manufacturer     n
 1:          ford 64813
 2:     chevrolet 50536
 3:        toyota 31596
 4:         honda 19866
 5:          jeep 17449
 6:        nissan 17391
 7:           ram 16443
 8:               15988
 9:           gmc 15420
10:           bmw 13738
11:         dodge 12325
12: mercedes-benz 10391
13:       hyundai  9374
14:        subaru  8984
15:    volkswagen  8896
16:         lexus  7739
17:           kia  7547
18:          audi  7150
19:      cadillac  6570
20:         acura  5702
21:      chrysler  5653
22:         buick  5181
23:         mazda  5048
24:      infiniti  4471
```

```
25:         lincoln  4033
26:           volvo  3276
27:      mitsubishi  3109
28:            mini  2260
29:         pontiac  2229
30:           rover  1976
31:          jaguar  1898
32:         porsche  1269
33:         mercury  1137
34:          saturn  1071
35:      alfa-romeo   870
36:           tesla   845
37:            fiat   772
38: harley-davidson   138
39:          datsun    63
40:         ferrari    39
41:    aston-martin    18
42:      land rover    11
43:          morgan     3
       manufacturer     n
```

Number of Vehicles from each model

```
df |> count(model,sort=TRUE)
```

```
                          model    n
    1:                    f-150 7115
    2:                          4600
    3:           silverado 1500 4546
    4:                     1500 3800
    5:                    camry 2827
   ---
28133:                      X5M    1
28134:               sorento lx    1
28135:                        ♿    1
28136:        𝓜𝓮𝓻𝓬𝓮𝓭𝓮𝓼 𝓫𝓮𝓷𝔃 𝓶𝓵 350    1
28137: 🔥GMC Sierra 1500 SLE🔥 4X4 🔥    1
```

Number of Vehicles from each type of drive

```
df |> count(drive,sort=TRUE)
```

```
   drive      n
1:   4wd 120500
2:       119969
3:   fwd  97602
4:   rwd  55217
```

contingency table comparing manufacturers and drive of vehicles

```
with(df,table(manufacturer,drive))
```

```
                 drive
manufacturer               4wd   fwd   rwd
                  6240  2141  3149  4458
  acura           2569   840  2255    38
  alfa-romeo       585    61     4   220
  aston-martin       2     0     0    16
  audi            4154  2218   750    28
  bmw             5719  2957   272  4790
  buick           1822   558  2610   191
  cadillac        2587  1514  1409  1060
  chevrolet      14563 16394  9819  9760
  chrysler        1699   264  2775   915
  datsun            24     0     0    39
  dodge           3603  1575  3525  3622
  ferrari            5     0    17    17
  fiat             164    13   440   155
  ford           18071 25123 10317 11302
  gmc             4519  8136  1126  1639
  harley-davidson   84    12     3    39
  honda           5869  3206 10633   158
  hyundai         2842   840  5269   423
  infiniti        1697  1103   576  1095
  jaguar          1113   116    28   641
  jeep            2728 13261  1002   458
  kia             2089   678  4627   153
  land rover         2     9     0     0
  lexus           2116  1758  2378  1487
  lincoln         1268   842  1467   456
  mazda           1202   481  2222  1143
  mercedes-benz   4246  2144   483  3518
  mercury          364   142   282   349
  mini             680   180  1381    19
  mitsubishi       892  1129   873   215
  morgan             2     0     0     1
  nissan          4396  4072  7558  1365
  pontiac          799    58   861   511
  porsche          573   336     9   351
  ram             3726 10606   677  1434
  rover            590  1369    10     7
  saturn           409    86   486    90
  subaru          3690  5058   113   123
  tesla            190   119    14   522
  toyota          8137  9657 11806  1996
  volkswagen      2500   554  5538   304
  volvo           1439   890   838   109
```

contingency table comparing year and drive of vehicles

```
with(df,table(year,drive))
```

```
      drive
year           4wd  fwd  rwd
  1900     9     1    2    0
  1901     2     0    0    1
  1905     0     0    0    1
  1909     1     0    0    0
  1910     2     0    0    0
  1913     0     0    0    2
  1915     0     0    0    1
  1916     1     0    0    1
  1918     0     0    0    1
  1920     1     0    0    1
  1921     1     0    0    1
  1922     1     0    0    2
  1923    10     0    0   26
  1924     2     0    0    7
  1925     3     0    0    5
  1926     7     0    0    9
  1927    10     0    3   23
  1928    10     1    1   24
  1929    24     0    0   32
  1930    38     0    0   29
  1931    35     0    0   22
  1932    20     0    3   31
  1933     8     0    1   14
  1934    19     0    1   23
  1935    10     0    0   13
  1936    23     0    0   20
  1937    31     0    1   38
  1938    15     1    1   20
  1939    13     0    4   30
  1940    39     1    3   36
  1941    22     1    1   41
  1942     6     0    0    8
  1943     1     0    0    0
  1944     0     3    0    0
  1945     0     2    0    0
  1946    23     5    4   25
  1947    27     0    1   35
  1948    43     6    3   47
  1949    35     3    2   44
  1950    38     2    3   59
  1951    36     4    2   54
  1952    42     7    2   57
  1953    44     6    4   48
  1954    44     3    5   46
  1955    77     6    2  125
```

1956	69	9	3	78
1957	70	8	0	80
1958	40	3	1	28
1959	33	4	3	44
1960	45	4	6	61
1961	39	3	3	37
1962	55	4	5	67
1963	87	17	5	120
1964	116	8	2	141
1965	149	5	5	194
1966	175	11	13	207
1967	145	8	9	186
1968	160	24	16	220
1969	159	27	9	203
1970	156	13	6	154
1971	100	18	11	170
1972	174	35	7	191
1973	99	33	11	179
1974	93	30	7	143
1975	70	27	5	91
1976	85	45	7	100
1977	81	48	4	128
1978	89	65	11	176
1979	133	61	13	175
1980	102	32	11	121
1981	68	31	17	95
1982	67	27	15	105
1983	84	41	9	122
1984	149	70	25	139
1985	149	80	40	195
1986	186	102	27	206
1987	177	92	32	221
1988	169	125	38	189
1989	157	155	53	195
1990	169	141	74	207
1991	164	152	77	210
1992	213	122	71	201
1993	185	178	117	206
1994	244	262	107	337
1995	366	365	147	355
1996	334	398	187	367
1997	480	497	233	473
1998	628	448	348	530
1999	822	1023	466	713
2000	969	1037	684	780
2001	1279	1280	879	890
2002	1625	1617	1111	1062
2003	1846	2250	1553	1258
2004	2467	2886	1926	1371
2005	3160	3129	2327	1579
2006	3542	3728	3050	1750

```
  2007  4306  4153  3716  1998
  2008  5033  5057  4224  1961
  2009  3527  3219  3544  1284
  2010  4706  4234  4384  1655
  2011  5818  6247  4566  2387
  2012  6836  6783  6181  2526
  2013  8619  7821  8039  3709
  2014  8377  8641  6791  3158
  2015  8150  9101  7558  3136
  2016  7196  9093  7760  3327
  2017  9463 11364  8287  3349
  2018 10325 10779  8188  3271
  2019  7115  7339  5428  2951
  2020  6613  4570  4311  2236
  2021   554   849   405    83
  2022    53    42     6     2
```

statistical summary of year

```r
with(df,summary(year))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   1900    2008    2013    2011    2017    2022    1171 
```

```r
library(tidyverse)
library(MASS)
library(pander)
```

```r
pander(addmargins(with(df,table(type,fuel))))
```

		diesel	electric	gas	hybrid	other	Sum
	791	8252	161	73953	815	1765	85737
bus	0	207	1	260	0	30	498
convertible	5	17	14	7008	37	292	7373
coupe	38	37	18	16355	39	1576	18063
hatchback	35	170	502	11302	1282	2626	15917
mini-van	41	18	0	4363	5	125	4552
offroad	0	31	0	561	0	1	593
other	47	888	52	14286	364	4139	19776
pickup	391	5851	3	28153	13	6944	41355
sedan	507	843	694	71005	1353	5877	80279
SUV	588	698	133	65163	494	3453	70529

		diesel	electric	gas	hybrid	other	Sum
truck	21	8739	2	21646	66	118	30592
van	116	297	1	6992	18	542	7966
wagon	10	193	42	8877	399	537	10058
Sum	2590	26241	1623	329924	4885	28025	393288

```
pander(addmargins(with(df,table(type,transmission))))
```

		automatic	manual	other	Sum
	328	76979	7238	1192	85737
bus	0	377	58	63	498
convertible	23	4915	1573	862	7373
coupe	50	9389	3028	5596	18063
hatchback	58	7941	1639	6279	15917
mini-van	15	4478	44	15	4552
offroad	0	356	231	6	593
other	376	8400	801	10199	19776
pickup	240	26619	1336	13160	41355
sedan	304	62065	3426	14484	80279
SUV	286	61868	1885	6490	70529
truck	20	28184	1802	586	30592
van	15	7261	98	592	7966
wagon	101	7445	593	1919	10058
Sum	1816	306277	23752	61443	393288

```
pander(addmargins(with(df,table(transmission,fuel))))
```

		diesel	electric	gas	hybrid	other	Sum
	334	115	19	1237	17	94	1816
automatic	2071	22964	781	266633	3800	10028	306277
manual	80	2293	12	20888	124	355	23752
other	105	869	811	41166	944	17548	61443

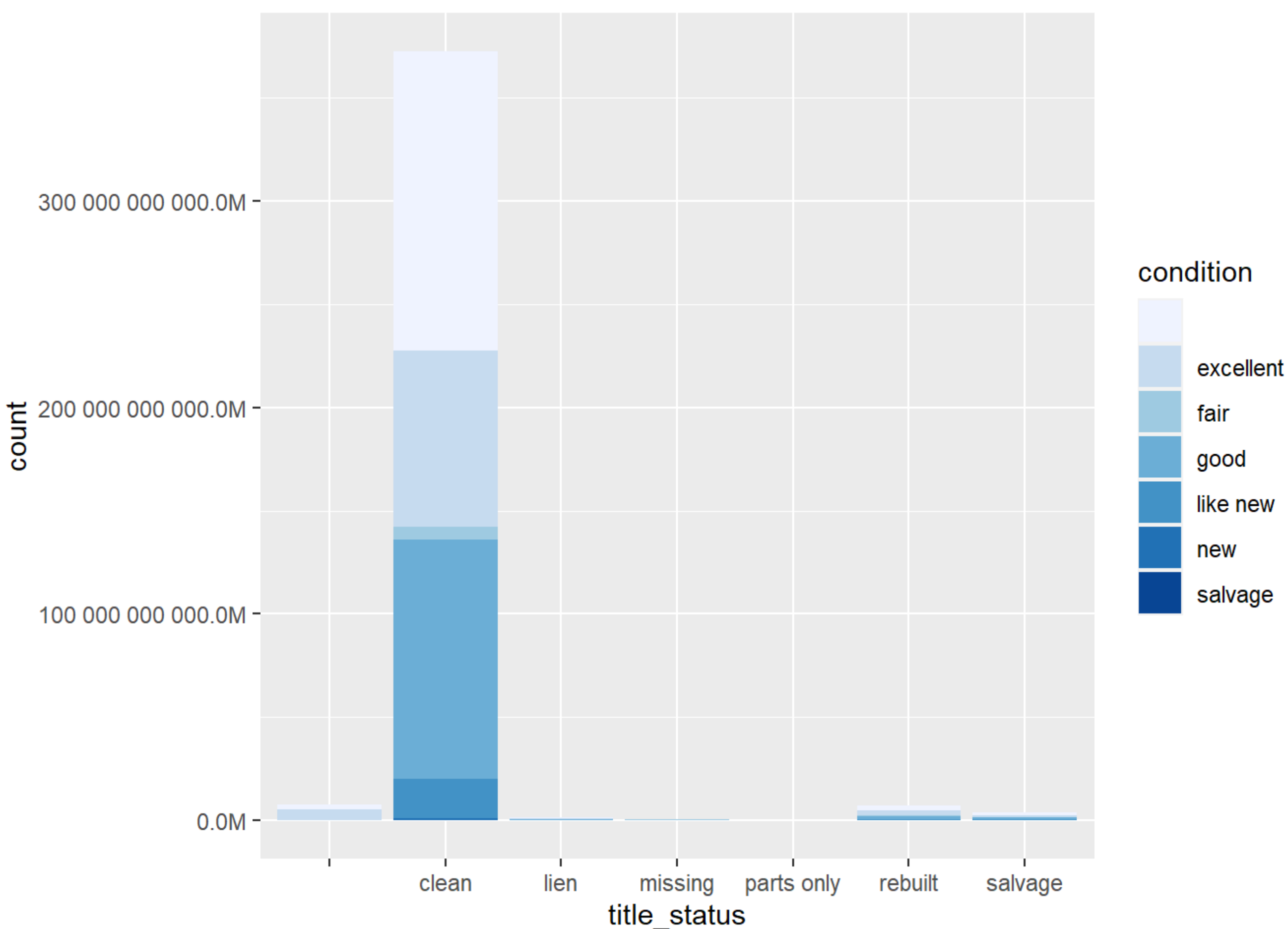
		diesel	electric	gas	hybrid	other	Sum
Sum	2590	26241	1623	329924	4885	28025	393288

Part 2: Visual Description

Title Status and Condition Bar Chart

This bar chart demonstrates the count of title status's for all vehicles which is disproportionately a clean title status, additionally it also demonstrates the condition of the vehicle.

```
df_bar <- ggplot(data = df, aes(title_status, fill = condition))
df_bar +
  geom_bar(stat = "count") +
  scale_fill_brewer() +
  scale_y_continuous(labels = scales::number_format(scale = 1e6, accuracy = 0.1, suffix =
```

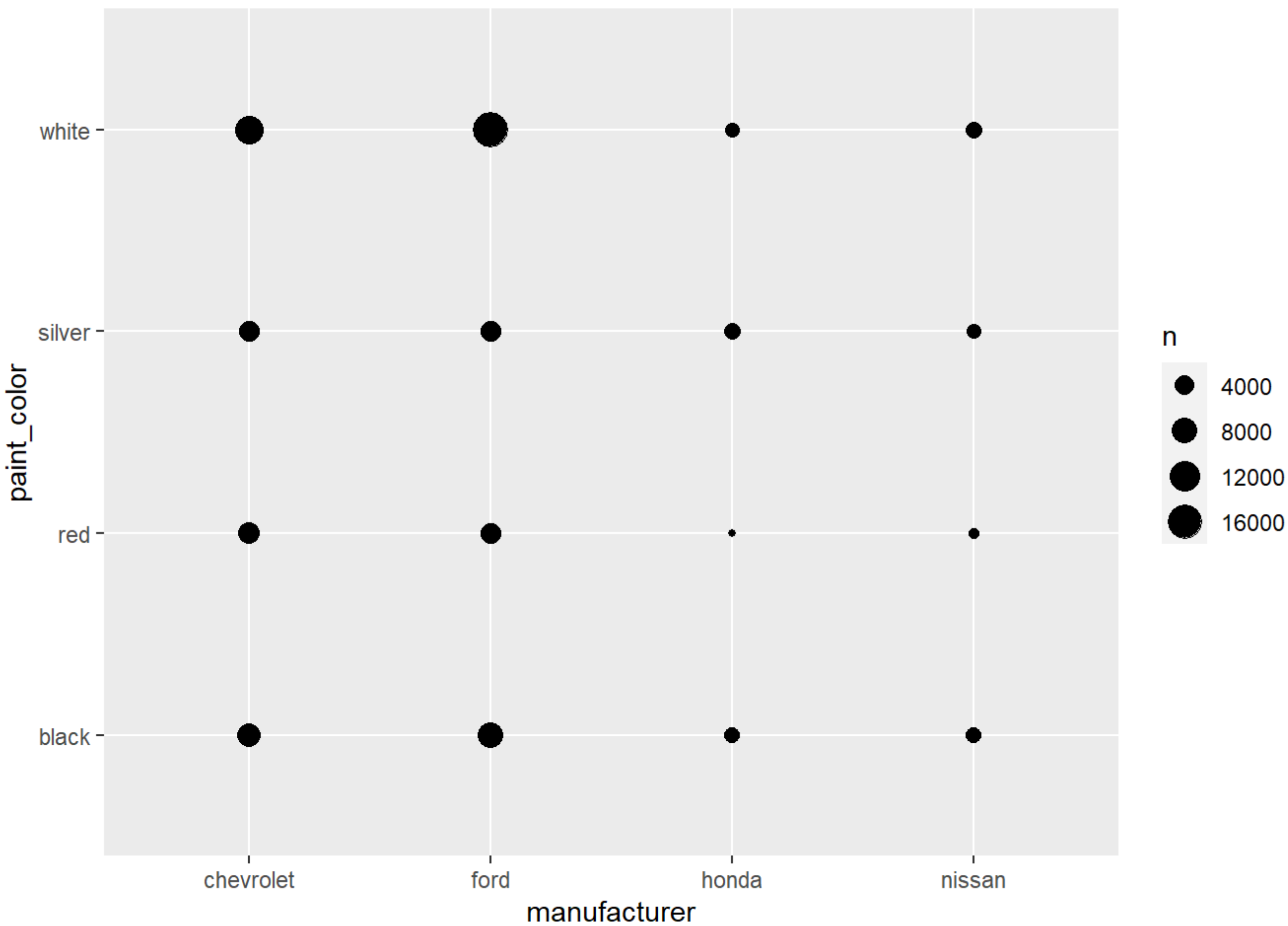


Manufacturer and Paint Color

Demonstrates the relationship between a vehicle's manufacturer and the vehicle's paint color. The most common types of relationships are a vehicle being manufactured by Ford and being the color white, a vehicle being manufactured by Chevrolet and being the color white, and a vehicle being manufactured by Ford and being the color black.

```
dfX_manufacturer<- subset(df, manufacturer %in% c("chevrolet", "ford", "honda", "nissan"))
dfY_manufacturer<- subset(dfX_manufacturer, paint_color %in% c("white", "black", "silver"
```

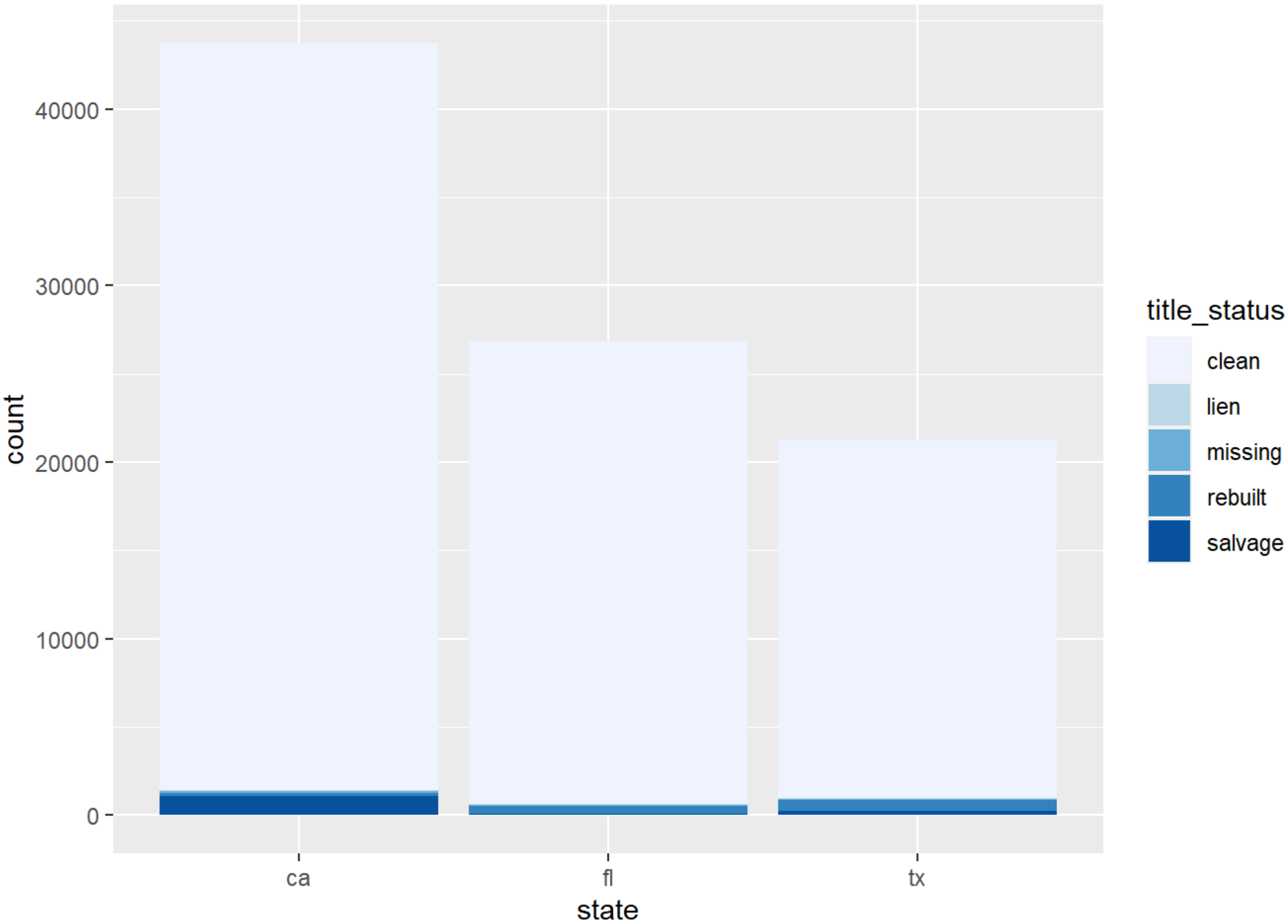
```
g_manufacturer <- ggplot(data = dfY_manufacturer, aes(manufacturer, paint_color))
g_manufacturer + geom_count()
```



Title Status and State

Demonstrates the title status of the top three states with the highest number of vehicles. The vehicles title statuses are overwhelmingly a clean title status.

```
dfX_state<- subset (df, state %in% c("tx", "ca", "fl"))
dfY_state<- subset (dfX_state, title_status %in% c("clean", "lien", "missing", "rebuilt",

df_state <- ggplot(data = dfY_state, aes(state,fill=title_status))
df_state + geom_bar(stat = "count") + scale_fill_brewer()
```



Number of Vehicles from each year

This is a visualization in the form of a stem plot of the number of vehicles from each year. Evidently, year the most cars are from is 2017 with 36420 cars.

```
stem(df$year)
```

```
  The decimal point is 1 digit(s) to the right of the |

  190 | 00000000000011159
  191 | 00335668
  192 | 00112223333333333333333333333333333333333334444444445555555566666666+124
  193 | 00000000000000000000000000000000000000000000000000000000000000000001+384
  194 | 00000000000000000000000000000000000000000000000000000000000000000000+387
  195 | 00000000000000000000000000000000000000000000000000000000000000000000+1109
  196 | 00000000000000000000000000000000000000000000000000000000000000000000+2670
  197 | 00000000000000000000000000000000000000000000000000000000000000000000+2964
  198 | 00000000000000000000000000000000000000000000000000000000000000000000+3838
  199 | 00000000000000000000000000000000000000000000000000000000000000000000+12537
  200 | 00000000000000000000000000000000000000000000000000000000000000000000+92977
  201 | 00000000000000000000000000000000000000000000000000000000000000000000+254578
  202 | 00000000000000000000000000000000000000000000000000000000000000000000+19644
```

Type of Model Visualization

This is a visualization in the form of a stem plot of the type of model of each vehicle. The model with the most cars is F-150 with 8009 cars.

```
df$model<-as.numeric(df$model)
```

```
Warning: NAs introduced by coercion
```

```
stem(df$model)
```

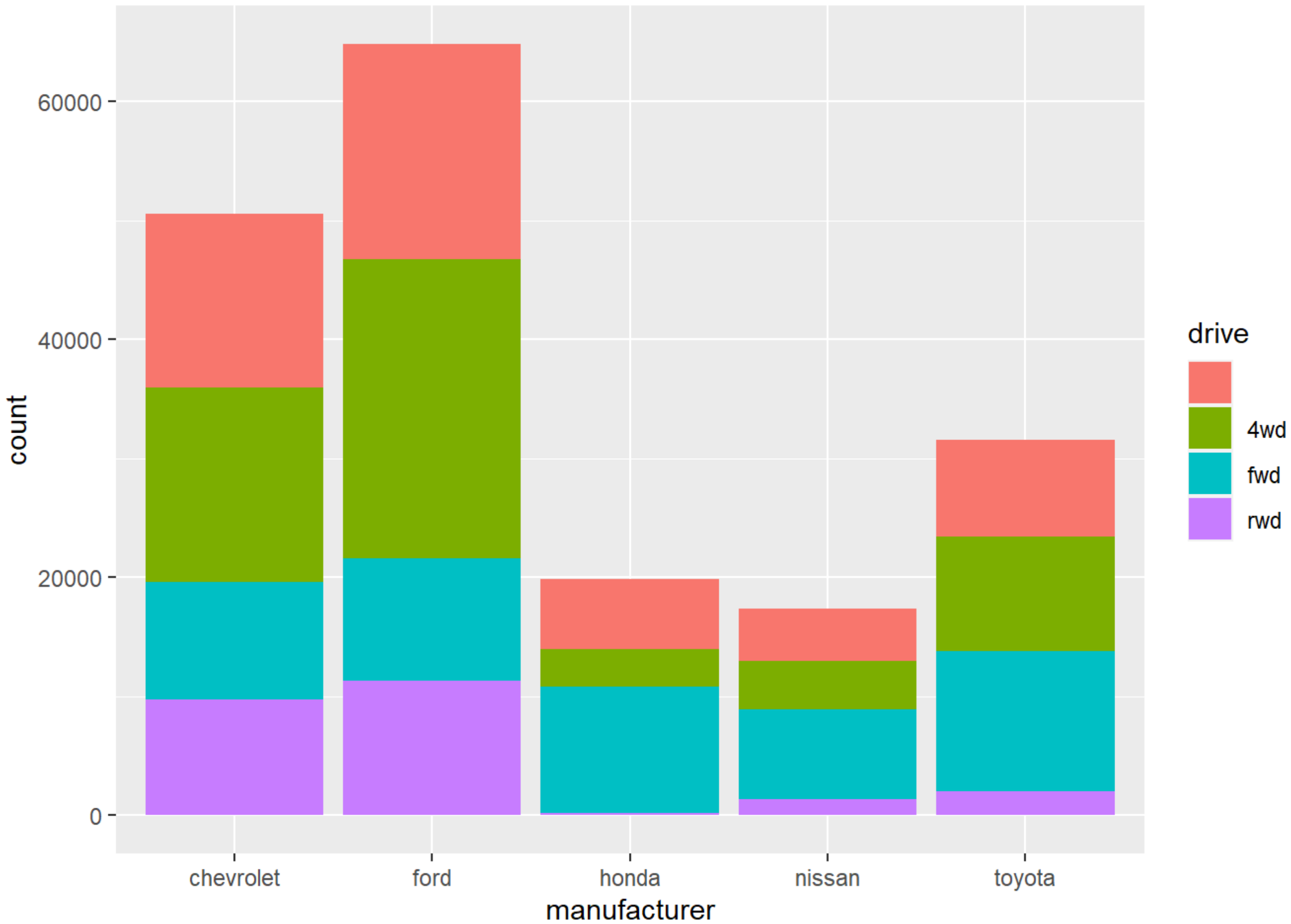
```

  The decimal point is 4 digit(s) to the right of the |

  -0 | 0
   0 | 0000000000000000000000000000000000000000000000000000000000000000000+11205
   1 | 
   2 | 15
   3 | 9
   4 | 
   5 | 
   6 | 
   7 | 
   8 | 
   9 | 3558
  10 | 08
  11 | 
  12 | 3
  13 | 35
  14 | 2
  15 | 8
  16 | 4
  17 | 0
  18 | 
  19 | 
  20 | 00
  21 | 
  22 | 
  23 | 
  24 | 8
```

Vehicle manufacturer and drive of the cars

This is a visualization in the form of a stacked bar chart comparing vehicle manufacturer and drive of the cars in the data set for the top manufacturers .

```
dfmanufacturerx <- subset(df, manufacturer %in% c("ford", "honda", "chevrolet", "nissan",
```

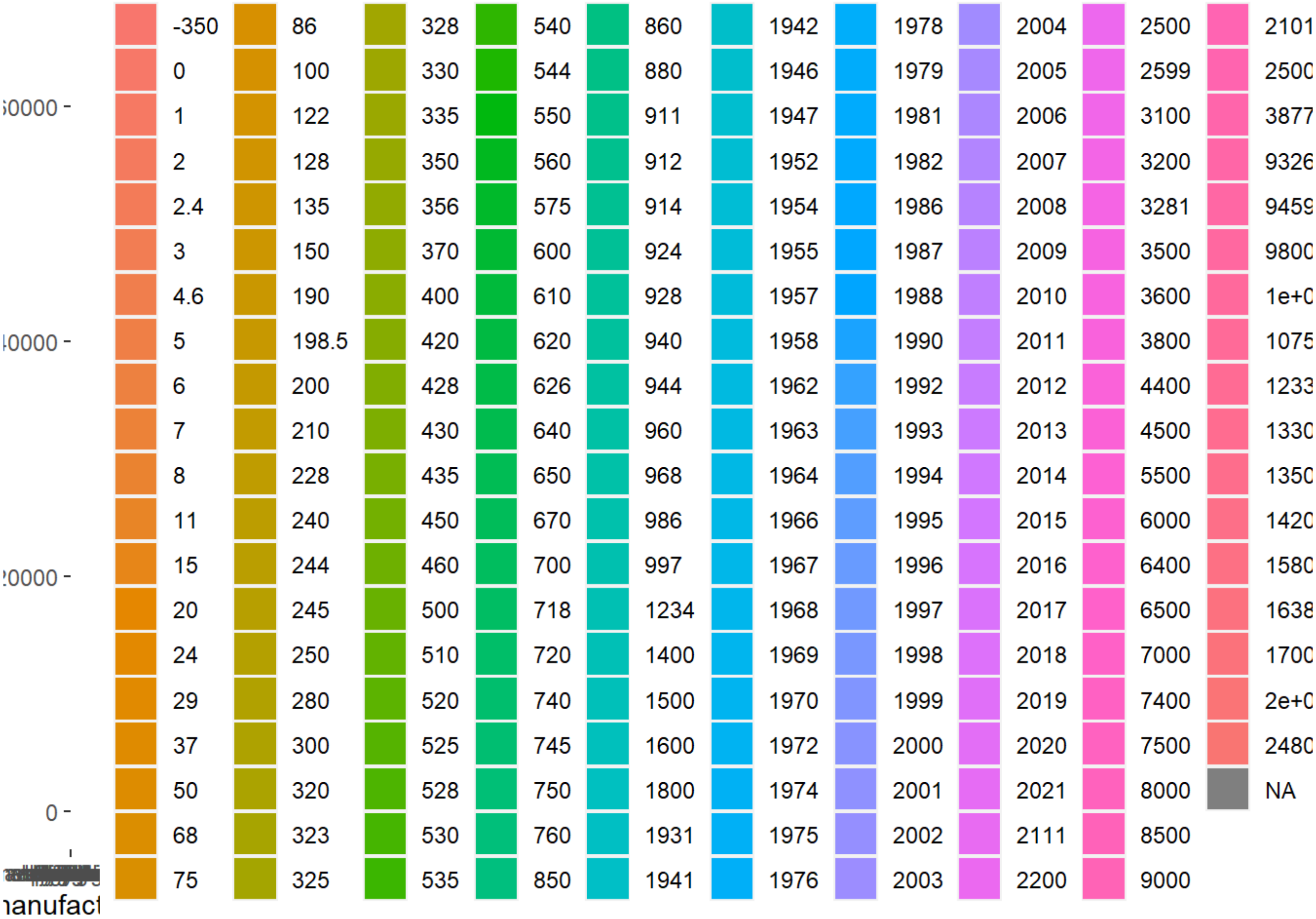
```
ggplot(data = dfmanufacturerx, aes(x = manufacturer, fill = drive)) +
    geom_bar(stat = "count")
```



Comparing vehicle manufacturer and model

This is a visualization in the form of a stacked bar chart comparing vehicle manufacturer and model of the cars in the data set.

```
df$model<-as.factor(df$model)
ggplot(data = df, aes(manufacturer, fill = model)) +
    geom_bar(stat = "count")
```

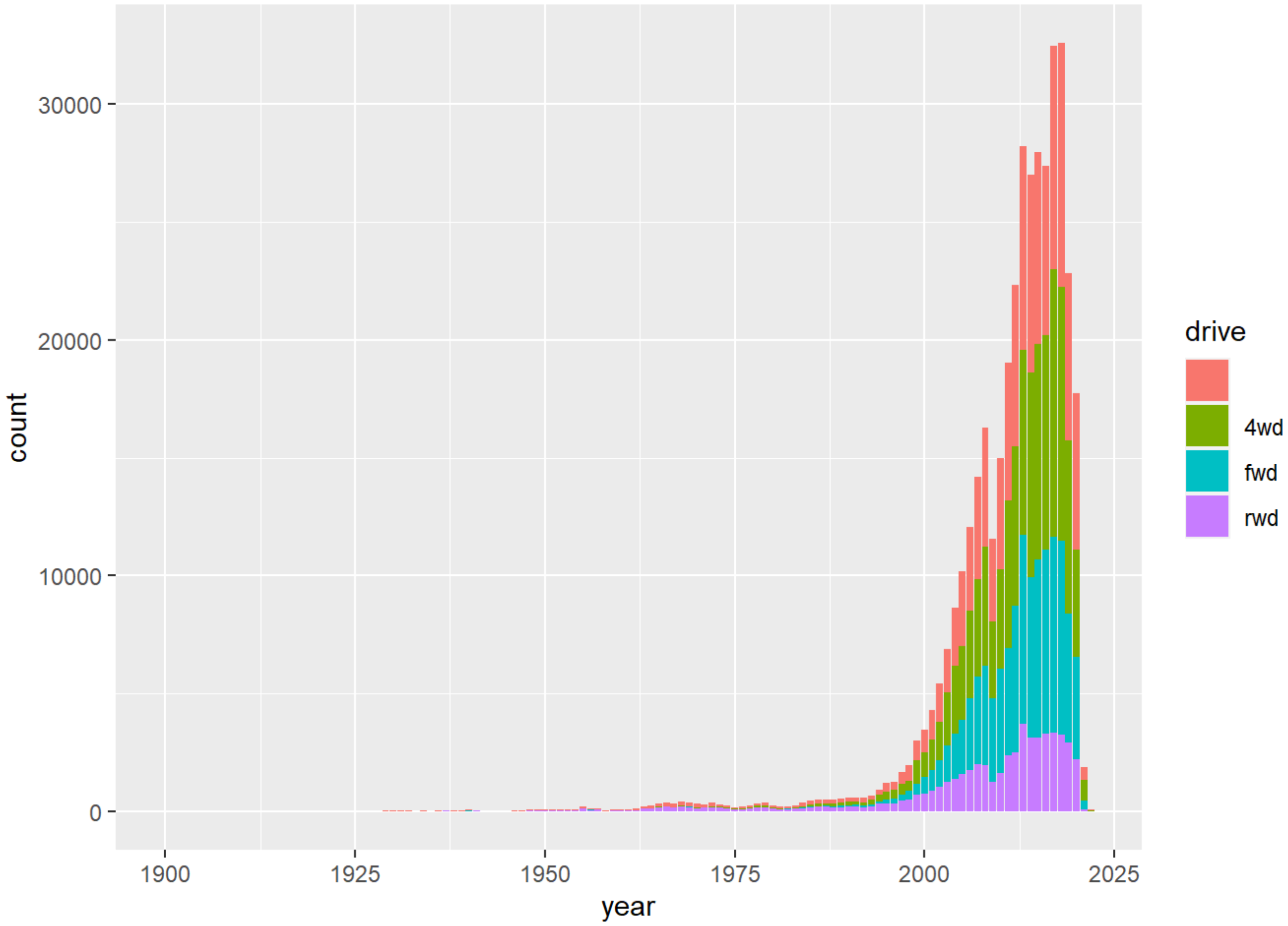


Comparing vehicle year and drive

This is a visualization in the form of a stacked bar chart comparing vehicle year and drive of the cars in the data set.

```
ggplot(data = df, aes(x = year, fill = drive)) +
    geom_bar(stat = "count")
```

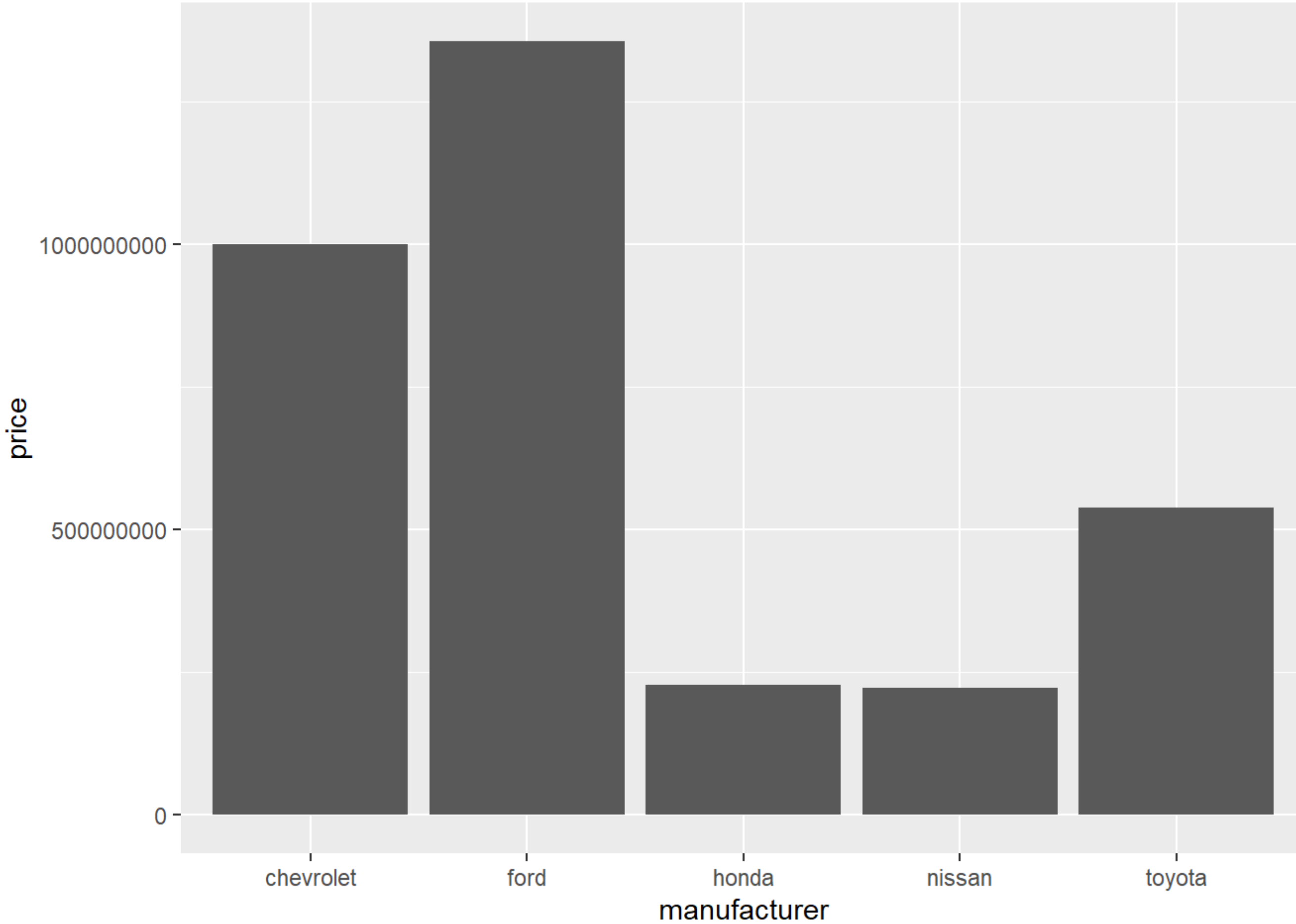
```
Warning: Removed 1171 rows containing non-finite values (`stat_count()`).
```



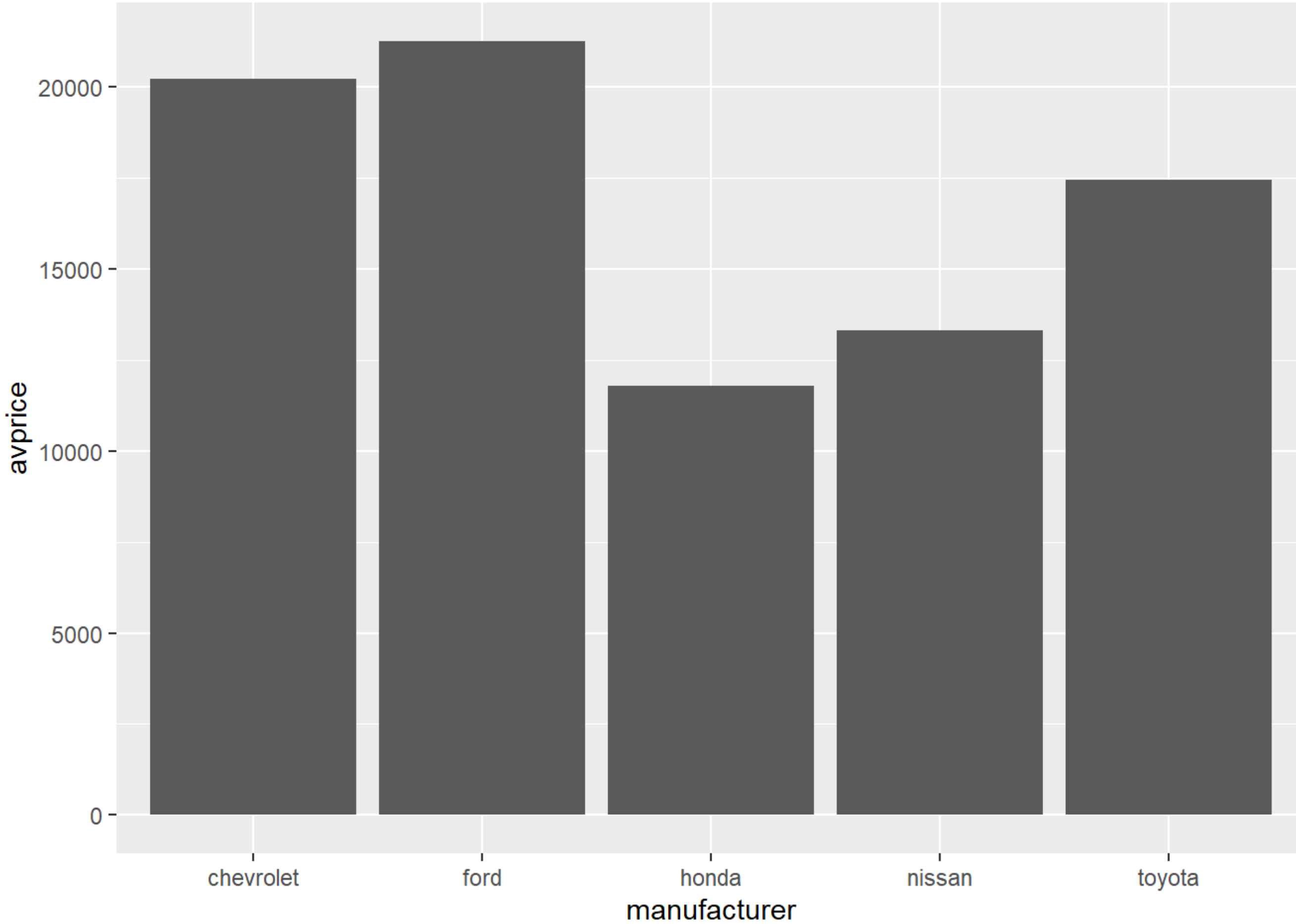
Manufacturer and Price

These are bar graphs showing the relationship between manufacturer and total price, and manufacturer and average price.

```
df <- df[df$price<500000&df$price>500,]
df<- subset (df, manufacturer %in% c("ford", "honda", "toyota", "chevrolet", "nissan"))
df$price<-as.numeric(df$price)
options(scipen=999)
f <- ggplot(df,aes(manufacturer,price))
f + geom_col() + scale_fill_manual(values=c(("lightblue"),("darkblue"),("pink"),("yellow"
```



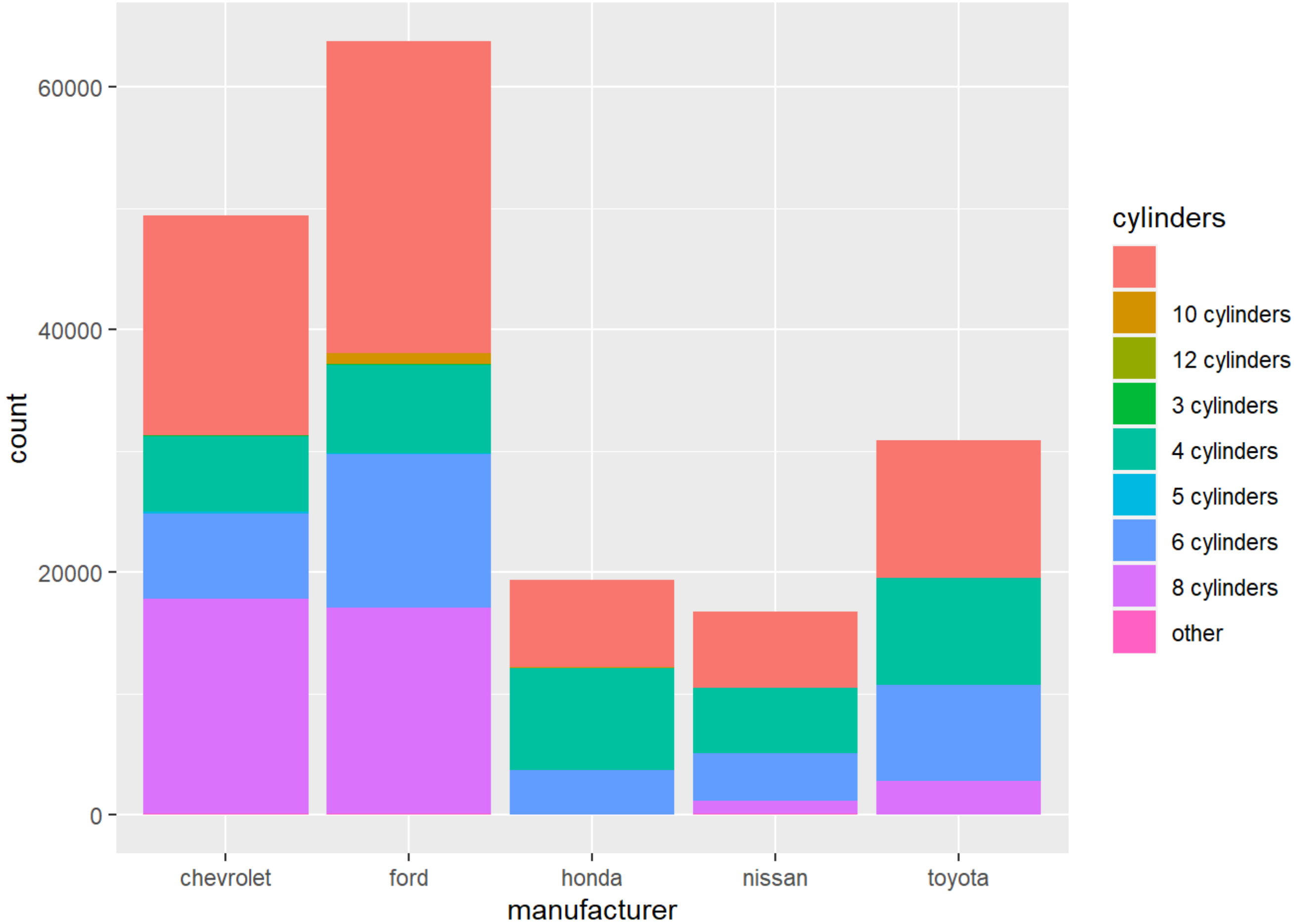

```
dfa <- df |> group_by(manufacturer) |>
  summarize(avprice = mean(price))
ggplot(dfa,aes(manufacturer,avprice))+geom_col() + scale_fill_manual(values=c(("lightblue
```



From this we can clearly see that a car manufactured by ford is on average more expensive than the others in comparisions, and one manufactured by honda is cheaper.

Manufacturer and Cylinders A bar graph showing the relationship between manufacturers and cylinders.

```
df<- subset (df, manufacturer %in% c("ford", "honda", "toyota", "chevrolet", "nissan"))
ggplot(df, aes(x = manufacturer, fill=cylinders)) +
    geom_bar(stat = "count")
```

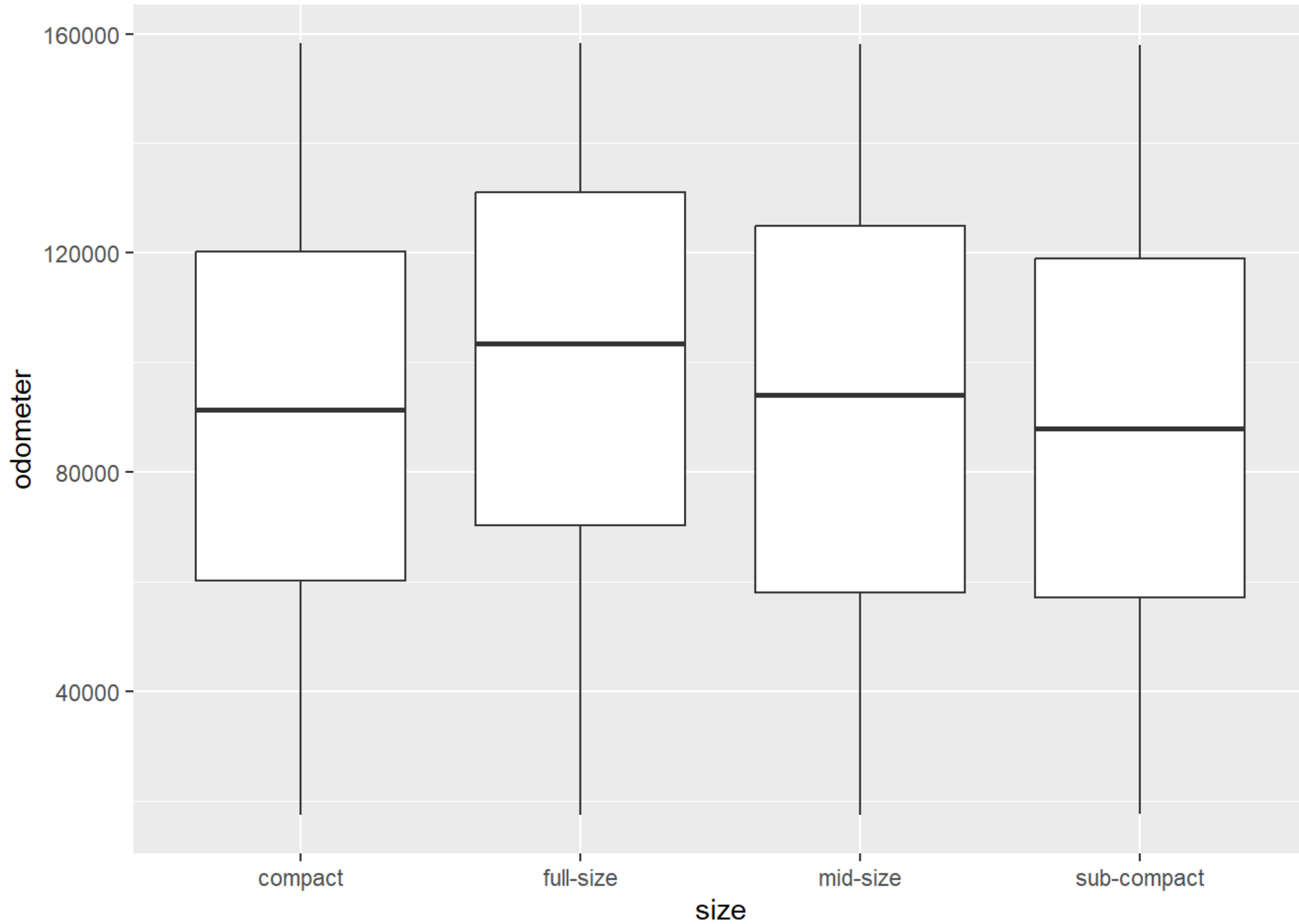


Chevrolet and Ford have produced a majority of cars their cars with 8 cylinders whereas cars with 3 cylinders are prevalent in honda,nissan, and toyota.We can also see that cylinders are often not mentioned from the glaring red patches in each column.

Odometer and Size A box plot showing the rlationship between type of car and odometer, with outliers removed for accuracy.

```
dfC<- subset (df, size %in% c("compact", "full-size", "mid-size", "sub-compact"))
dfC |> ggplot(aes(size,odometer))+geom_boxplot(outlier.shape=NA)+  scale_y_continuous(lim
```

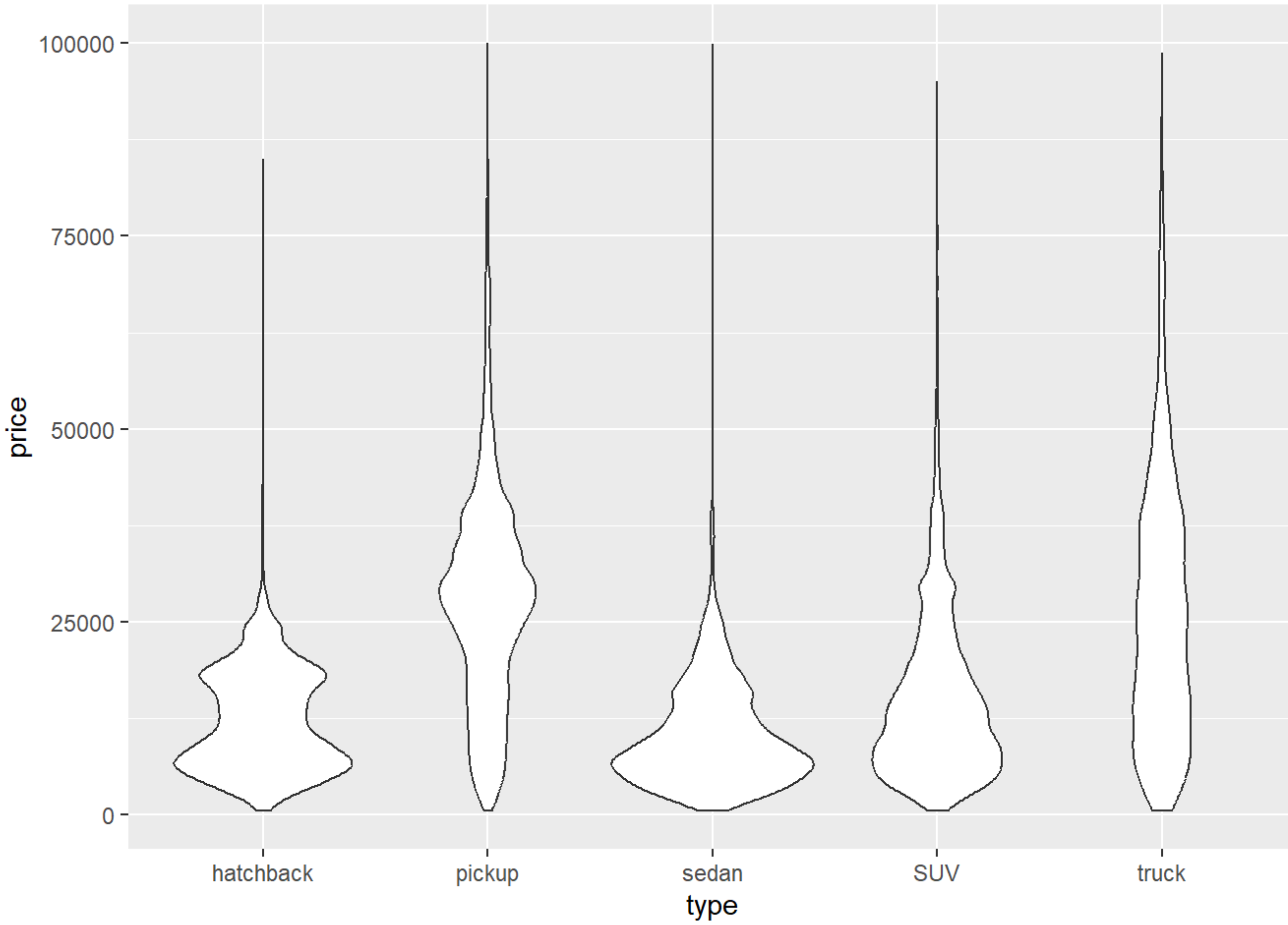
```
Warning: Removed 17285 rows containing non-finite values (`stat_boxplot()`).
```



We can see how the median of the odometer values for all types of cars is very similar.

Type and Price

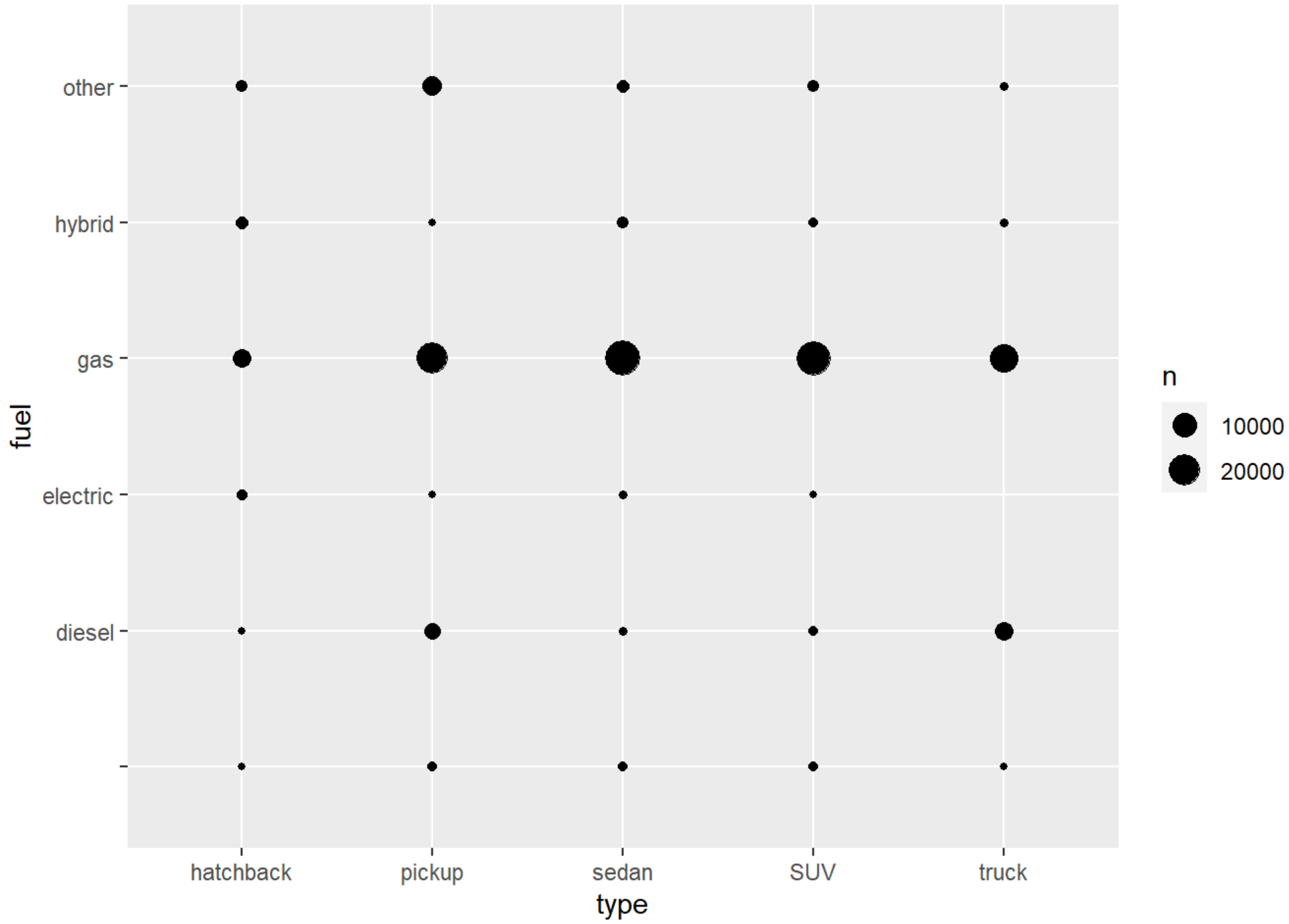
```
df <- df[df$price<500000&df$price>500,]
df<- subset (df, type %in% c("pickup", "sedan", "SUV", "truck", "hatchback"))
ggplot(df,aes(type,price)) + geom_violin(scale = "area")
```



The above violin plot demonstrates a visual representation of the distribution of the price of vehicle based on the type of vehicle, from the violin plot we can infer the median, interquartile range, and the shape of the distribution of prices of vehicles based on vehicle type. For example a pickup vehicle will have a higher mean and median price compared to a sedan.

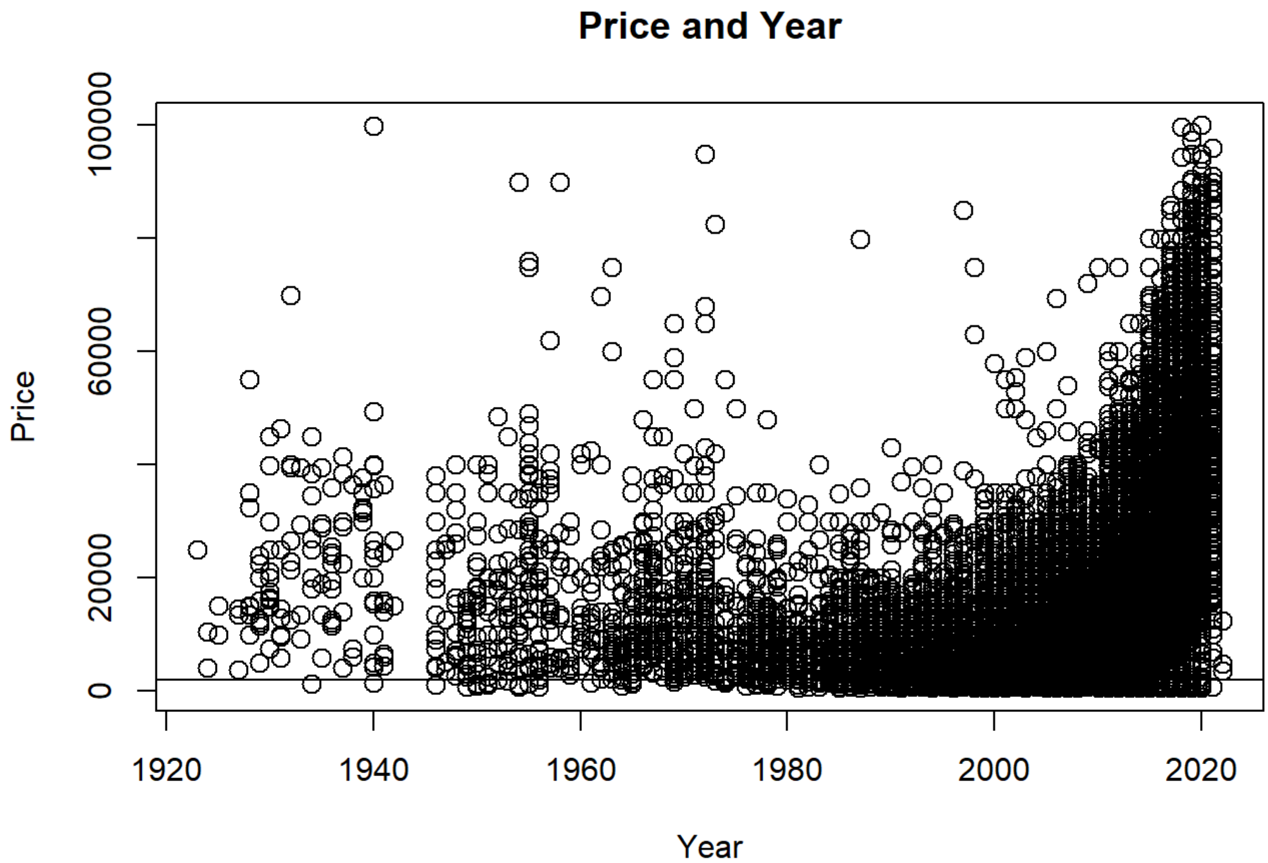
Vehicle type and Fuel Type

```
df<- subset (df, type %in% c("pickup", "sedan", "SUV", "truck", "hatchback"))
ggplot(df, aes(type, fuel)) + geom_count()
```



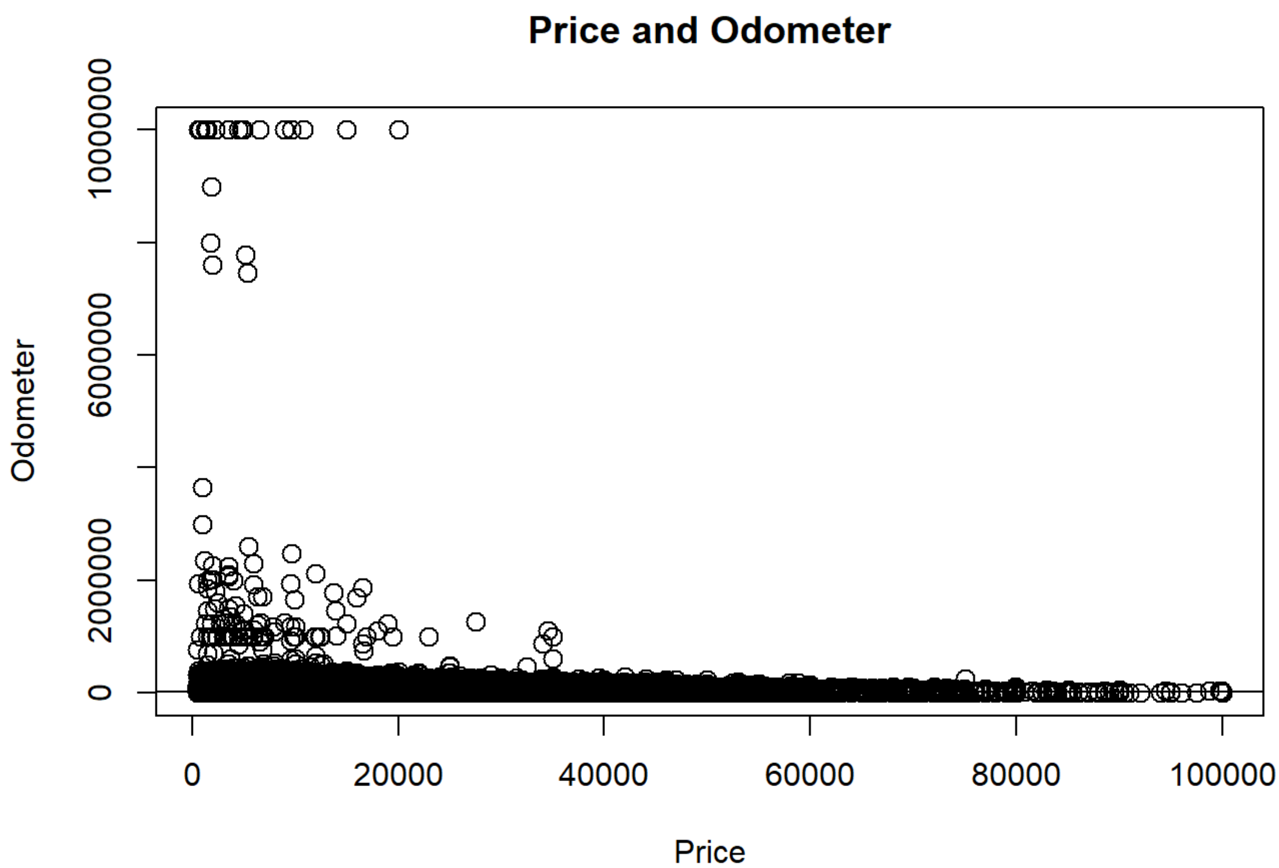
Regression Analysis

```
plot(df$year,df$price,col = "black",main = "Price and Year",
abline(lm(df$year~df$price)),cex = 1.3,pch = 1,xlab = "Year",ylab = "Price")
```



This linear model demonstrates the distribution of price and year. Furthermore, this linear model demonstrates that the distribution for price and odometer is significantly not normally distributed as there is a significant amount of outliers found throughout the model, giving the model a non linear appearance.

```
plot(df$price,df$odometer,col = "black",main = "Price and Odometer",
abline(lm(df$price~df$odometer)),cex = 1.3,pch = 1,xlab = "Price",ylab = "Odometer")
```



This linear model demonstrates the distribution of price and odometer. Furthermore, this linear model demonstrates that the distribution for price and odometer is significally normally distributed, however, there is a significant amount of outliers found mostly within the price range of 0 to 20000.

```
library(caret)
library(leaps)
```

```
library(ggplot2)

model <- lm(price ~ type + cylinders + drive, data = df)

summary(model)
```

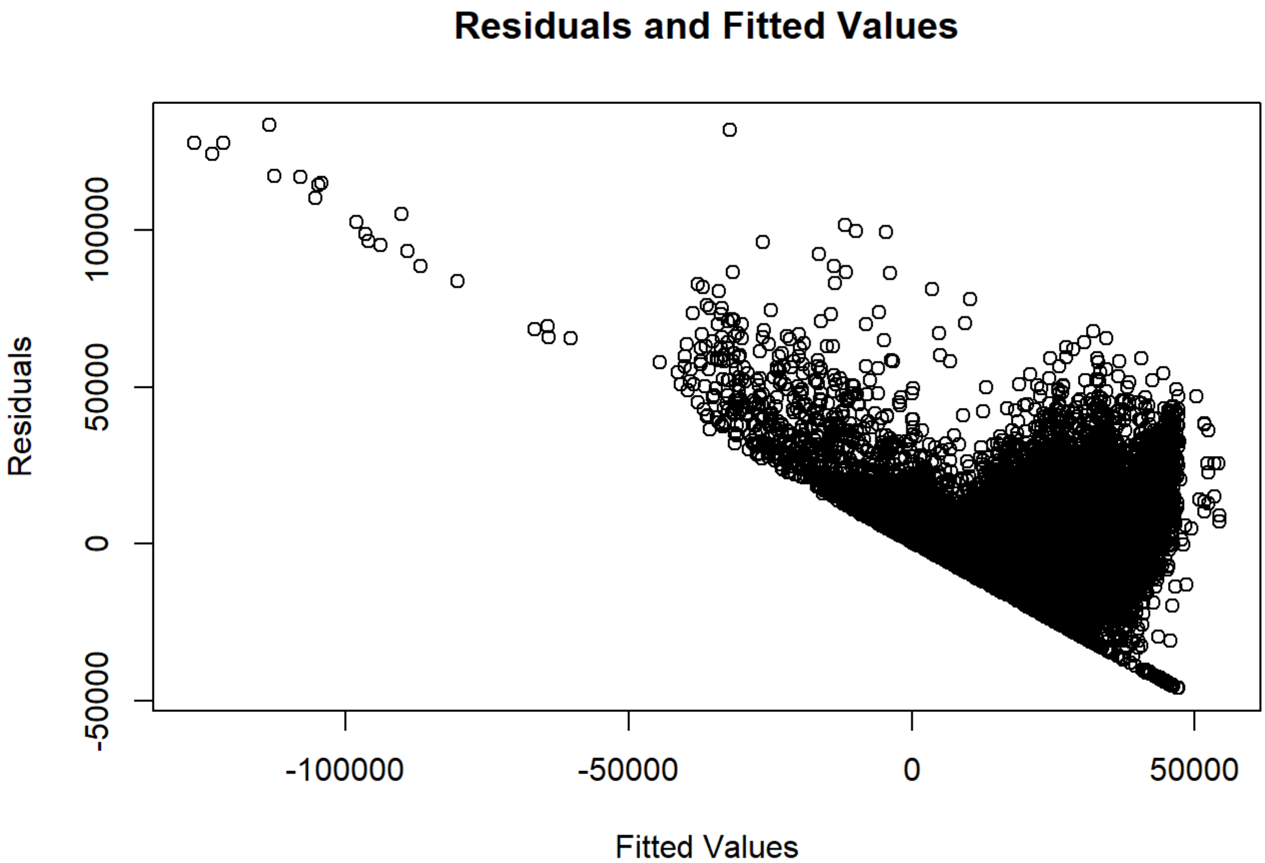
```

Call:
lm(formula = price ~ type + cylinders + drive, data = df)

Residuals:
   Min     1Q Median     3Q    Max 
-32312  -6834  -1159   5719  94014 

Coefficients:
                          Estimate Std. Error t value             Pr(>|t|)
```

```
(Intercept)             15067.4      178.0  84.632 < 0.0000000000000002 ***
typepickup              13234.1      184.2  71.833 < 0.0000000000000002 ***
typesedan                -581.3      161.4  -3.602             0.000316 ***
typeSUV                  1373.5      178.1   7.713   0.0000000000000124 ***
typetruck               12677.9      198.7  63.820 < 0.0000000000000002 ***
cylinders10 cylinders   -3563.8      489.7  -7.277   0.0000000000003433 ***
cylinders12 cylinders -14747.6     5803.2  -2.541             0.011046 *
cylinders3 cylinders    -3009.2      963.3  -3.124             0.001786 **
cylinders4 cylinders    -5294.1      104.2 -50.821 < 0.0000000000000002 ***
cylinders5 cylinders   -17038.9      947.9 -17.975 < 0.0000000000000002 ***
cylinders6 cylinders    -4552.9      102.4 -44.481 < 0.0000000000000002 ***
cylinders8 cylinders    -3780.7      109.7 -34.456 < 0.0000000000000002 ***
cylindersother          -3378.3      764.1  -4.421   0.0000098238119481 ***
drive4wd                 4522.6      112.9  40.057 < 0.0000000000000002 ***
drivefwd                 -325.8      123.6  -2.635             0.008410 **
driverwd                -4831.3      154.0 -31.368 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11600 on 111679 degrees of freedom
Multiple R-squared:  0.3368,    Adjusted R-squared:  0.3367
F-statistic:  3781 on 15 and 111679 DF,  p-value: < 0.00000000000000022
```

Regression Diagnostics

Residuals and Fitted Values

```
model <- lm(price ~ year + odometer + cylinders + condition + fuel + transmission + drive

plot(model$fitted.values, model$residuals,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residuals and Fitted Values")
```

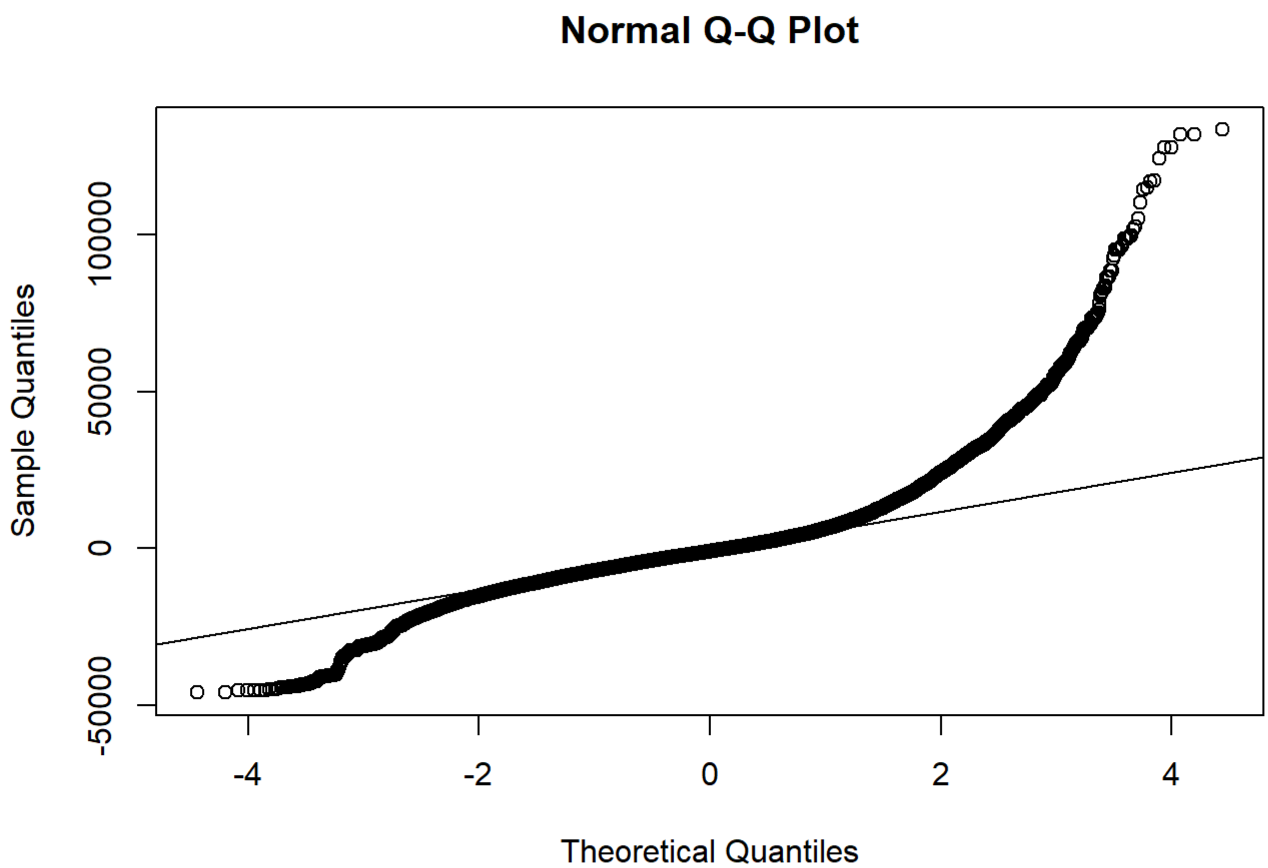


The Residuals and Fitted Values plot demonstrates the residuals in comparison to the the fitted values from the linear regression model. Some kind of a linear relationship is demonstrated by the plot as the points do not go below a certain defined line.

We expect to see no clear patterns in the plot, which indicates that the linear regression model is appropriate for the data. If there is a clear pattern, it suggests that the linear regression model may not be appropriate.

Normal Q Q Plot

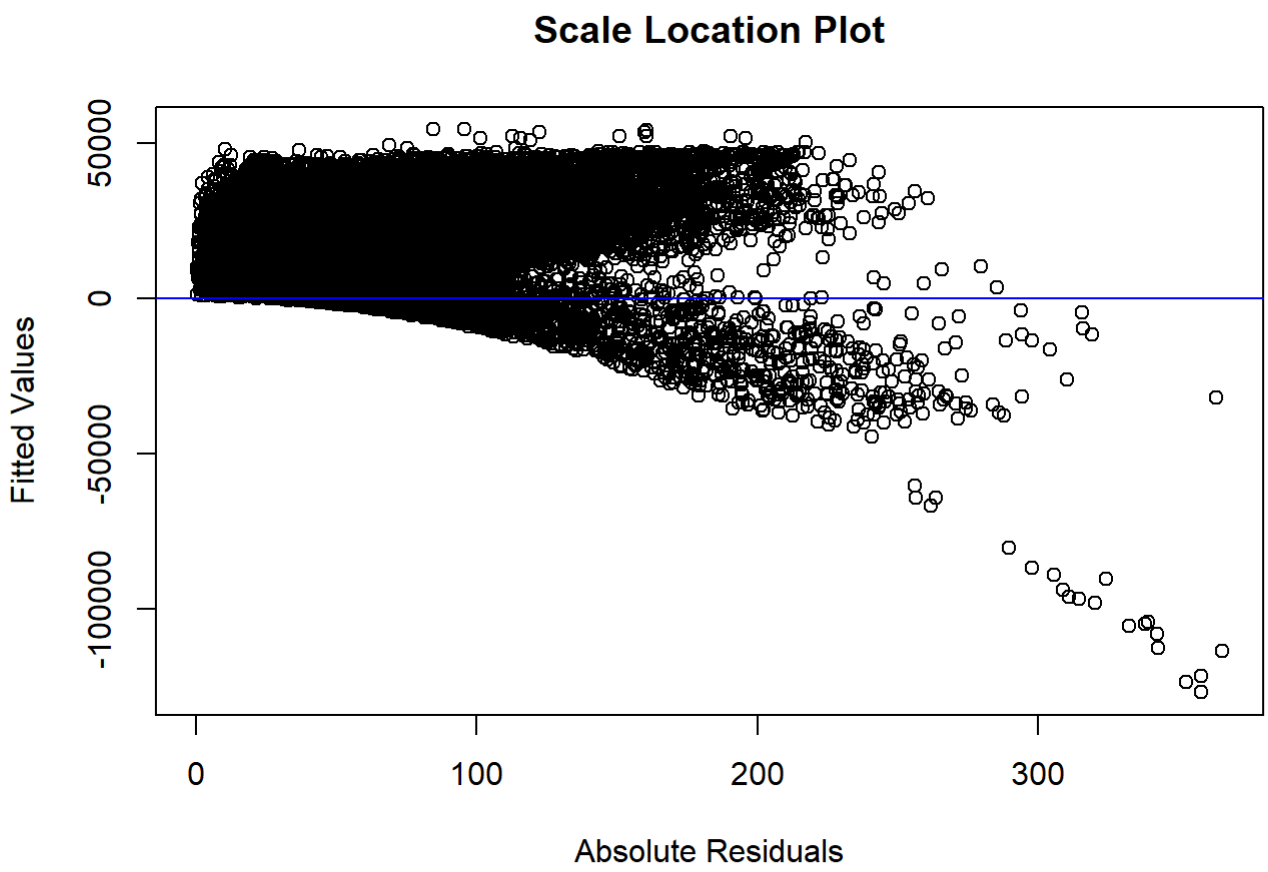
```
qqnorm(model$residuals)
qqline(model$residuals)
```



The Normal Q Q Plot demonstrates whether the residuals are normally distributed in the data. The relatively straight line between all the points indicates that the residuals are normally distributed. We only see significant deviation from the straight line at starting near the theoretical quantity of 2.

Scale Location Plot

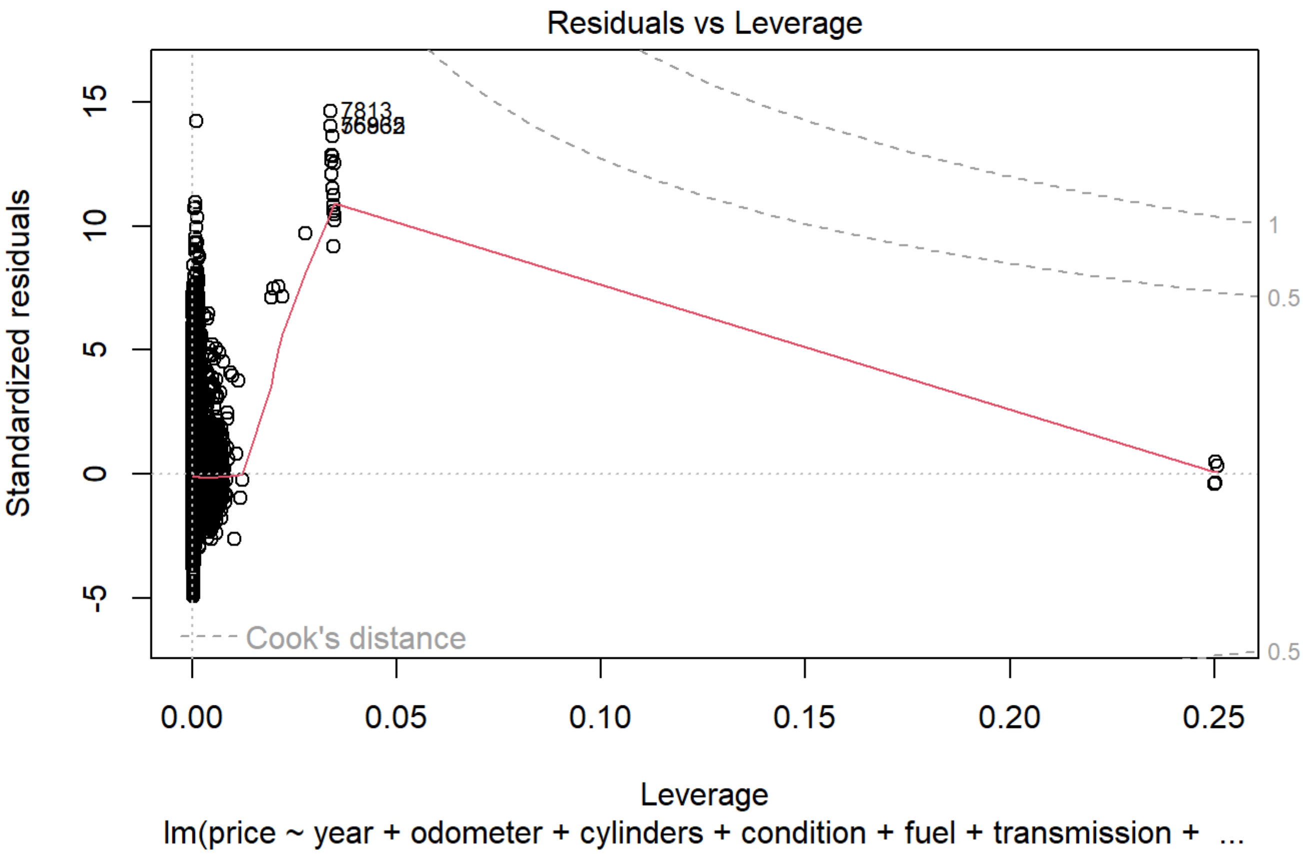
```
plot(sqrt(abs(model$residuals)), model$fitted.values,
     xlab = "Absolute Residuals", ylab = "Fitted Values",
     main = "Scale Location Plot")

abline(lm(sqrt(abs(model$residuals)) ~ model$fitted.values), col = "blue")
```



The Scale Location Plot demonstrates the square root of the absolute residuals against the fitted values. There is no clear pattern between fitted values and the absolute residuals demonstrating significant variance, however, the points seem to not go below a linear line that is concaving down.

Residuals and Leverage Plot

```
plot(model, which = 5)
```



The Residuals and Leverage plot shows the leverage of each point compared to its standardized residuals. Most points are found in between 0.0 and 0.002 on the x-axis demonstrating the leverage. Additionally, most points do not fall near the line. Ultimately, demonstrating that the points significantly impact the regression line.