Alexandria University
Faculty of Engineering
Computer and Systems Engineering
Probability

Final project Report

Num	Name	ID
1	حسن أحمد حسن محجوب	20010486
2	حسن أيمن عبد المنعم النجار	19015599

Loading and discovering Data (mtcars):

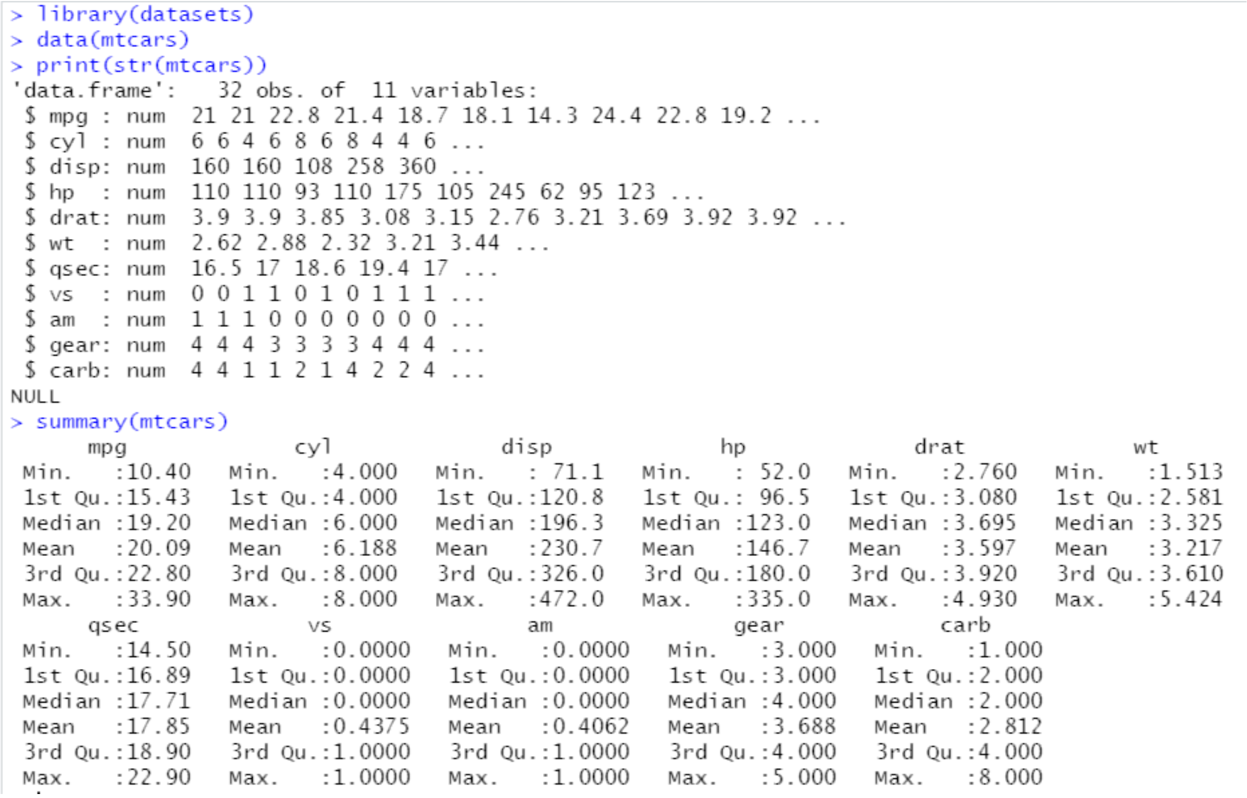
Algorithm:

Using the following built-in functions to load, discover and summary mtcars dataset:

library(datasets) –

data("mtcars") - str(mtcars) - summary(mtcars).

Output:



```
> library(datasets)
> data(mtcars)
> print(str(mtcars))
'data.frame':	32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
NULL
> summary(mtcars)
      mpg             cyl             disp             hp             drat             wt       
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0   Min.   :2.760   Min.   :1.513  
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:3.080   1st Qu.:2.581  
 Median :19.20   Median :6.000   Median :196.3   Median :123.0   Median :3.695   Median :3.325  
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7   Mean   :3.597   Mean   :3.217  
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.920   3rd Qu.:3.610  
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0   Max.   :4.930   Max.   :5.424  
      qsec             vs               am              gear            carb      
 Min.   :14.50   Min.   :0.0000   Min.   :0.0000   Min.   :3.000   Min.   :1.000  
 1st Qu.:16.89   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
 Median :17.71   Median :0.0000   Median :0.0000   Median :4.000   Median :2.000  
 Mean   :17.85   Mean   :0.4375   Mean   :0.4062   Mean   :3.688   Mean   :2.812  
 3rd Qu.:18.90   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
 Max.   :22.90   Max.   :1.0000   Max.   :1.0000   Max.   :5.000   Max.   :8.000  
```

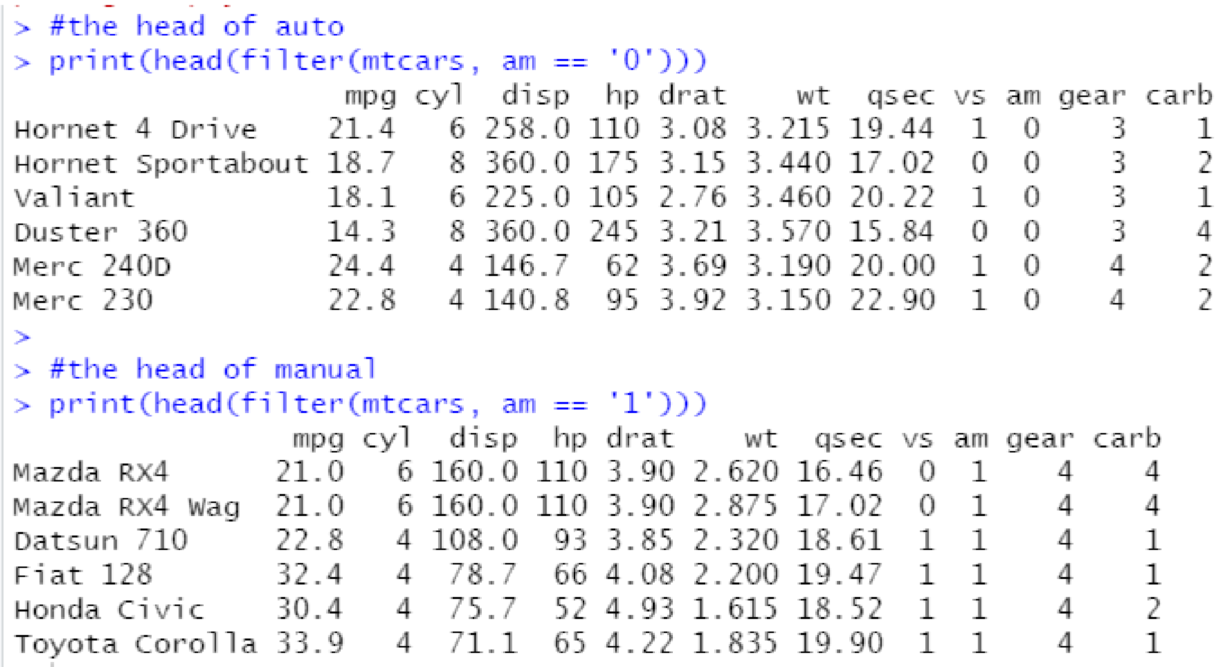
Extracting information:

Head of transmission:

Algorithm:

Using function: filter().

Output:



```
> #the head of auto
> print(head(filter(mtcars, am == '0')))
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
>
> #the head of manual
> print(head(filter(mtcars, am == '1')))
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
```

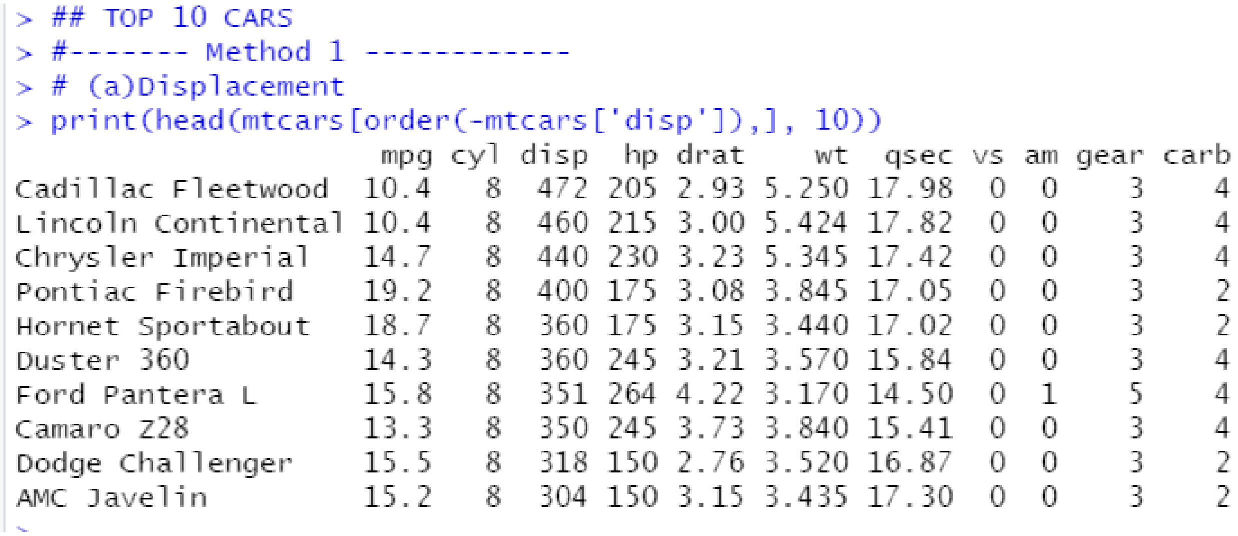
Top 10 cars according to displacement, horsepower, drat:

Algorithms:

1. Using function: order().
2. Getting 10th car then print all cars above it.

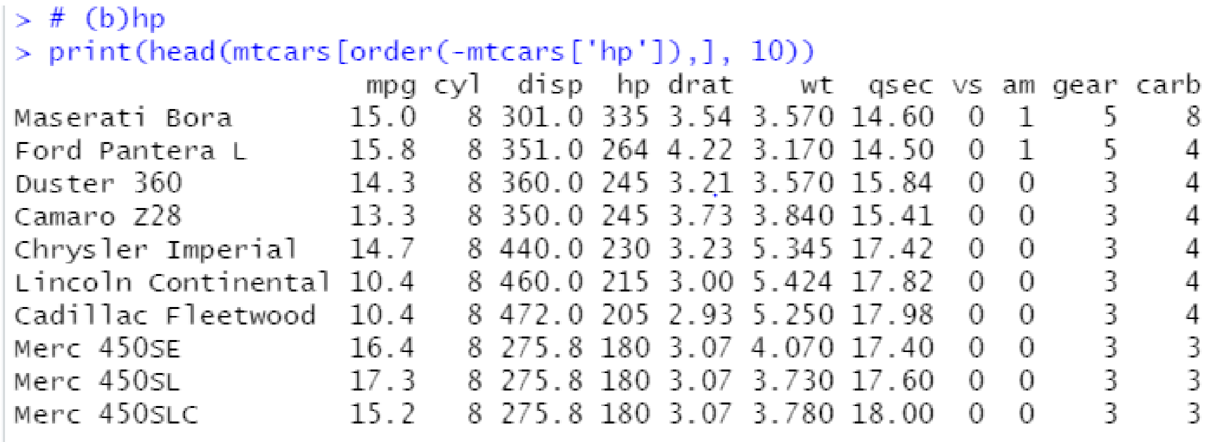
Outputs:

1- According to disp:



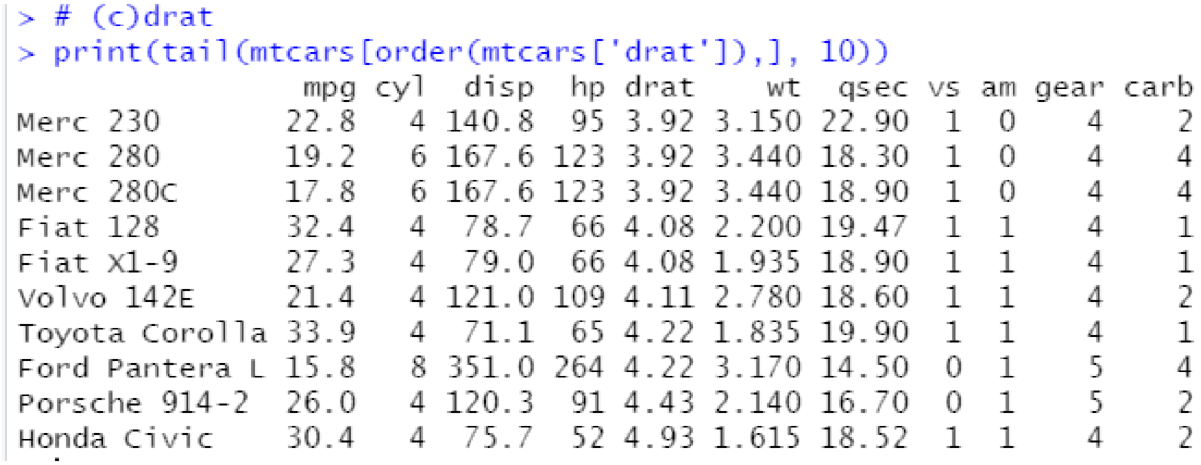
```
> ## TOP 10 CARS
> #------- Method 1 ------------
> # (a)Displacement
> print(head(mtcars[order(-mtcars['disp']),], 10))
                     mpg cyl disp  hp drat    wt  qsec vs am gear carb
Cadillac Fleetwood  10.4   8  472 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8  460 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8  440 230 3.23 5.345 17.42  0  0    3    4
Pontiac Firebird    19.2   8  400 175 3.08 3.845 17.05  0  0    3    2
Hornet Sportabout   18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Duster 360          14.3   8  360 245 3.21 3.570 15.84  0  0    3    4
Ford Pantera L      15.8   8  351 264 4.22 3.170 14.50  0  1    5    4
Camaro Z28          13.3   8  350 245 3.73 3.840 15.41  0  0    3    4
Dodge Challenger    15.5   8  318 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin         15.2   8  304 150 3.15 3.435 17.30  0  0    3    2
```

2- According to hp:



```
> # (b)hp
> print(head(mtcars[order(-mtcars['hp']),], 10))
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
```

3- According to drat:



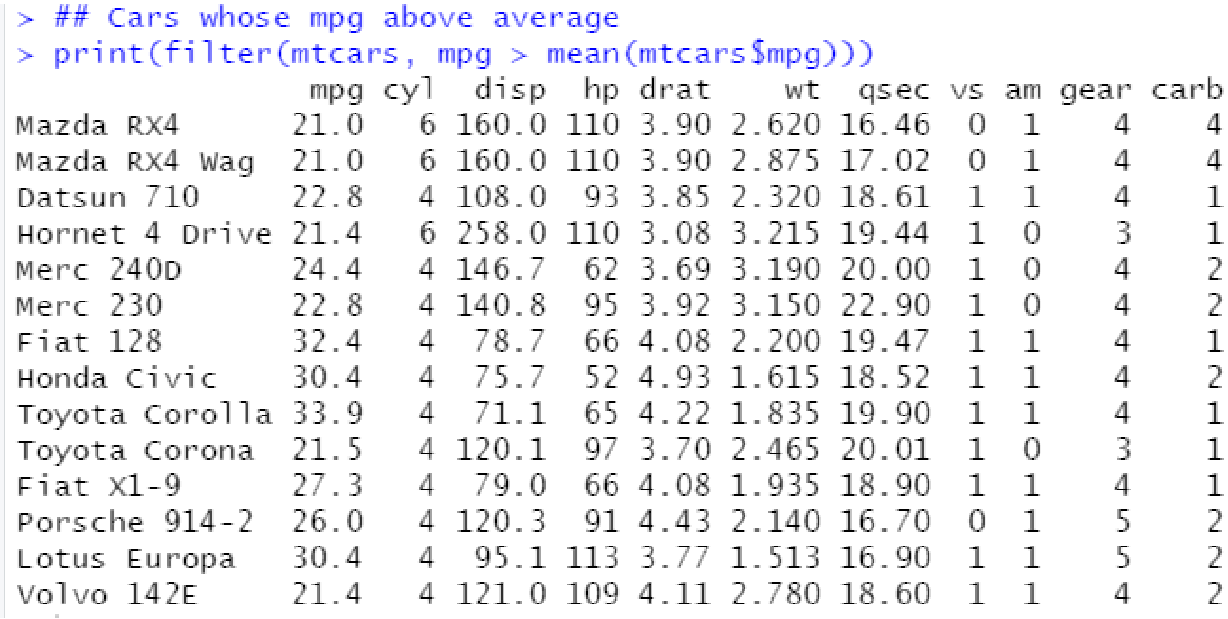
```
> # (c)drat
> print(tail(mtcars[order(mtcars['drat']),], 10))
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Merc 230       22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280       19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C      17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
```

Display cars whose mpg above average:

Algorithms:

Using functions: mean() and filter to get average and cars above average

Outputs:



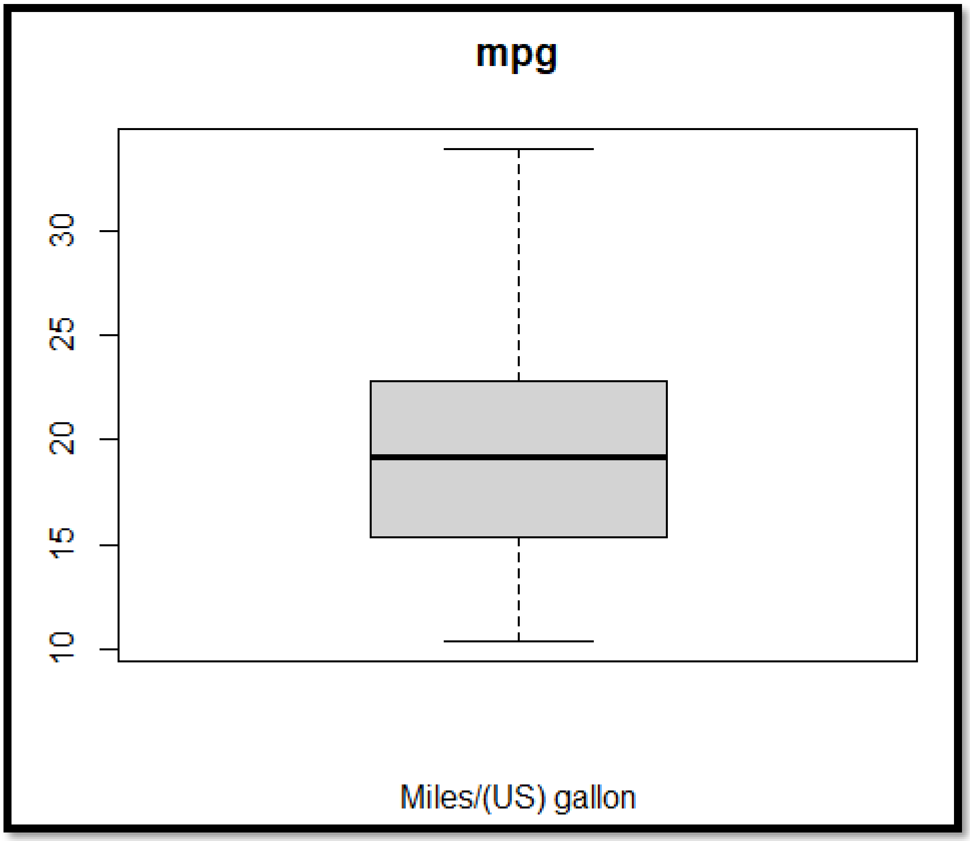
```
> ## Cars whose mpg above average
> print(filter(mtcars, mpg > mean(mtcars$mpg)))
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230       22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Toyota Corona  21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

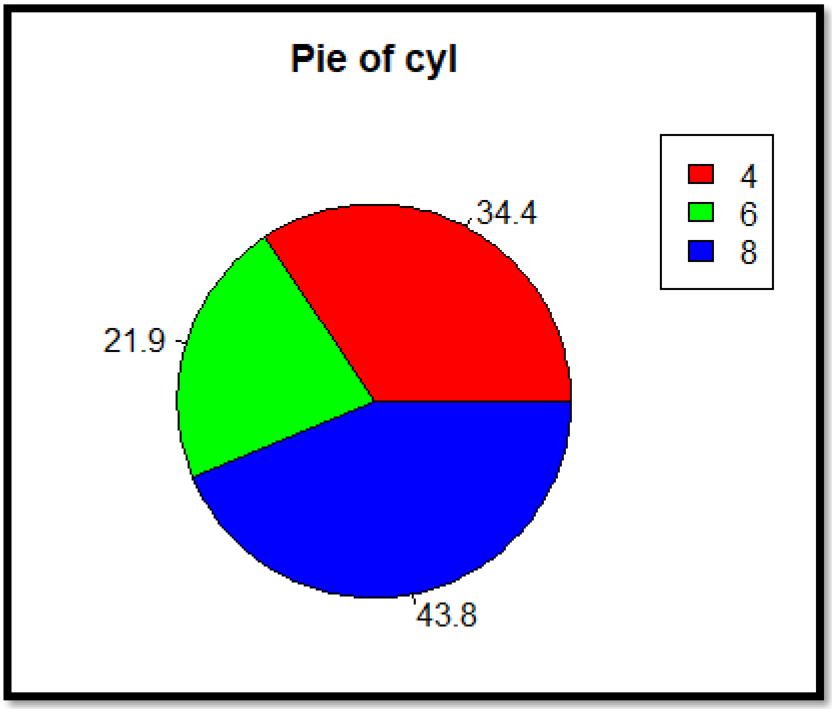
Dataset visualization:

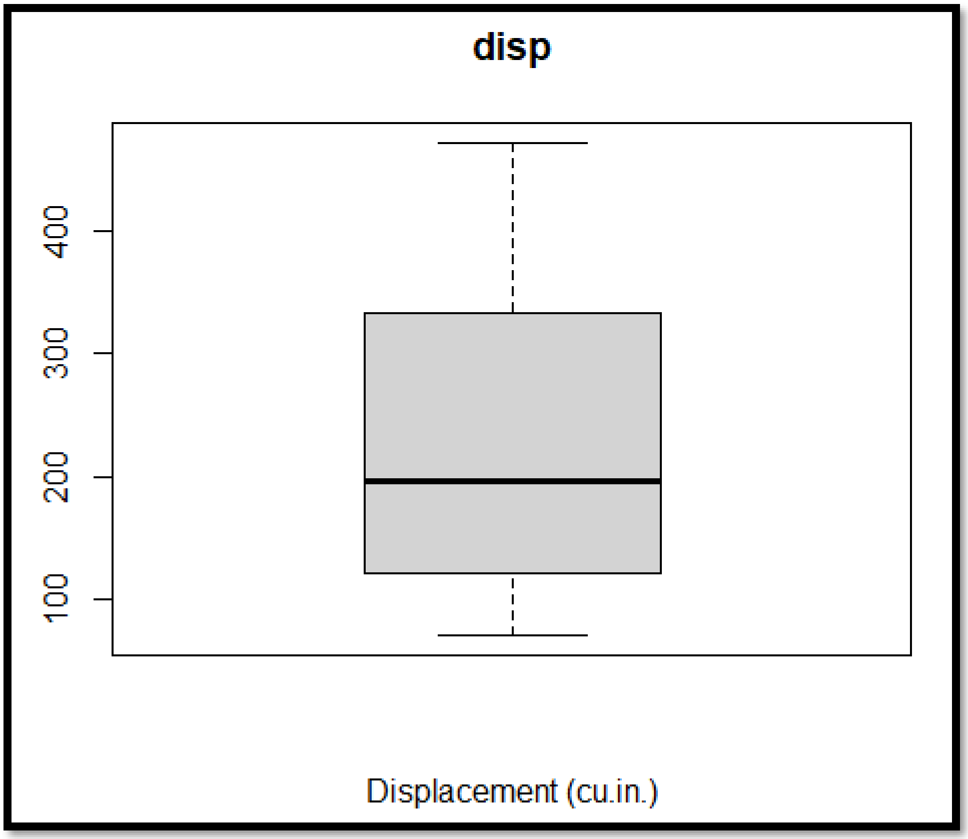
Histograms >> for discrete values spread over a small range

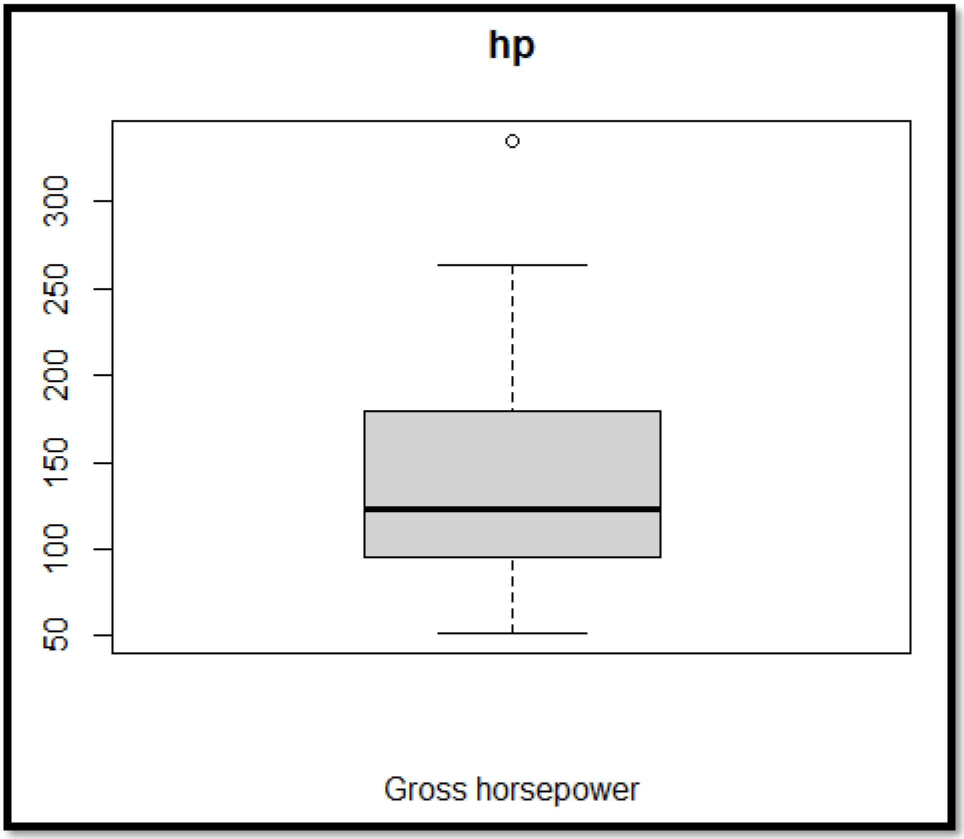
Box plots >> to provide a visual summary of a categorical variable

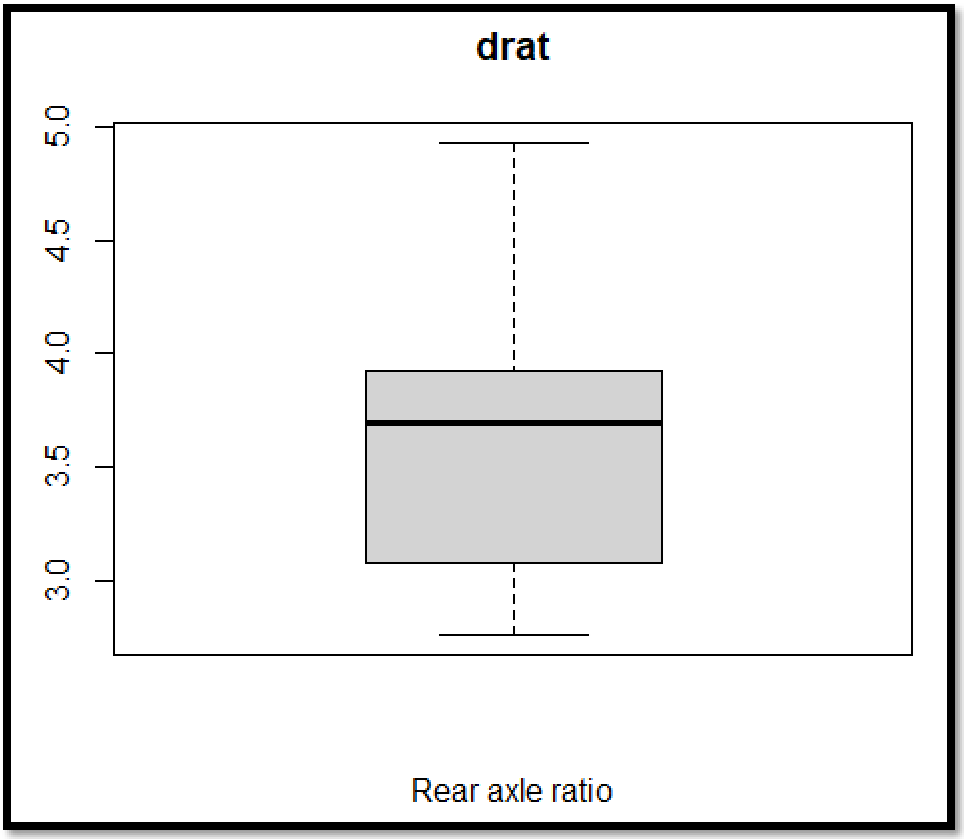
Pie chart >> to show percentage of single numerical variables

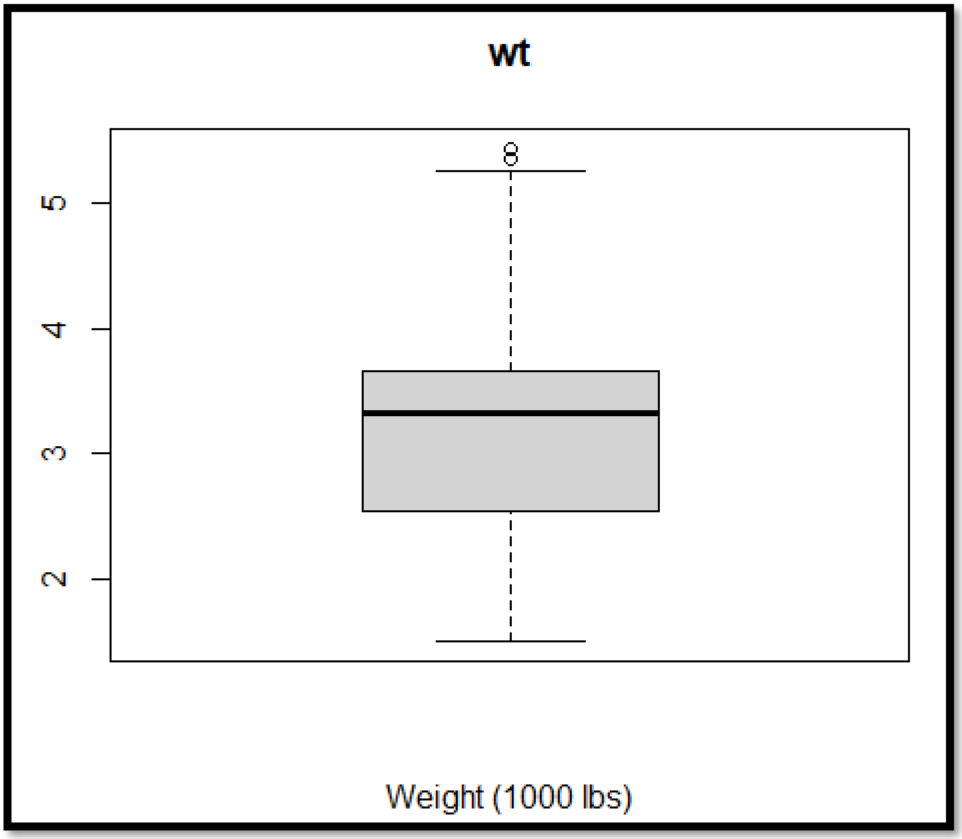

mpg
10
15
20
25
30
Miles/(US) gallon

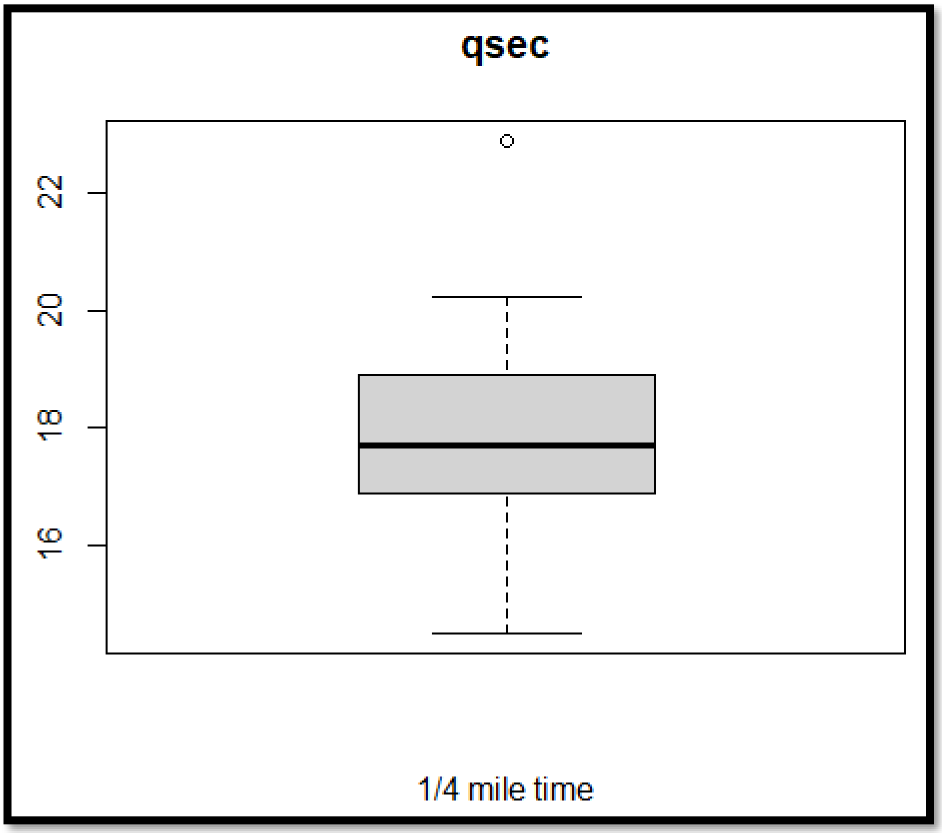

Pie of cyl
34.4
21.9
43.8
🟥 4
🟩 6
🟦 8

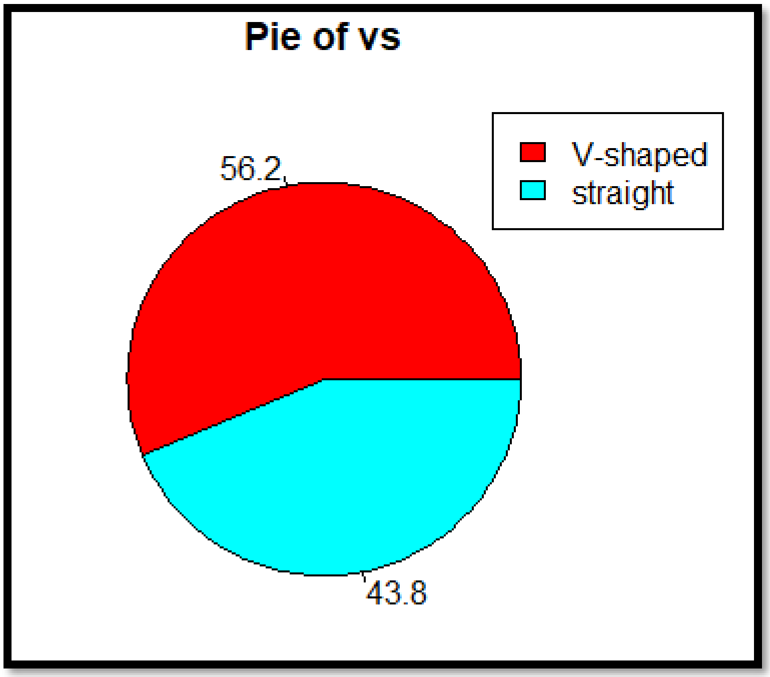

disp
400
300
200
100
Displacement (cu.in.)

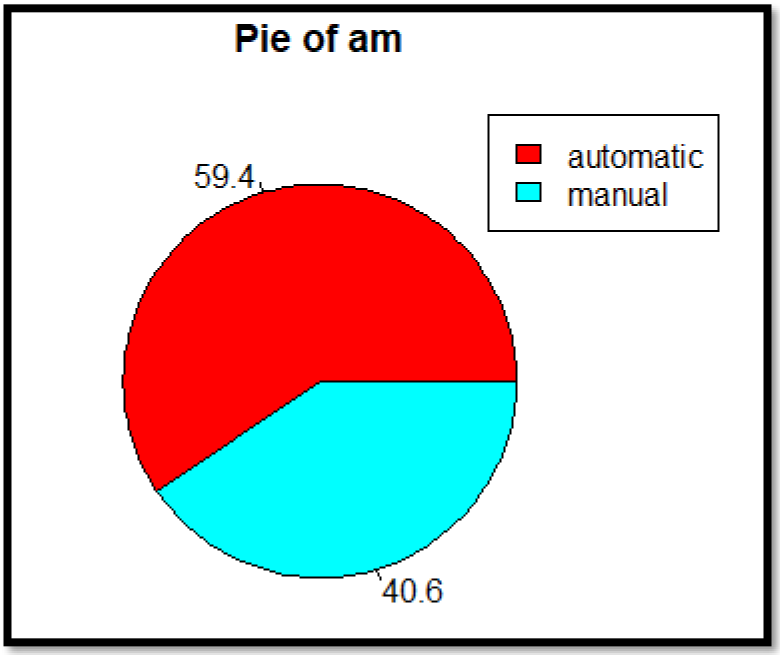

hp
300
250
200
150
100
50
Gross horsepower

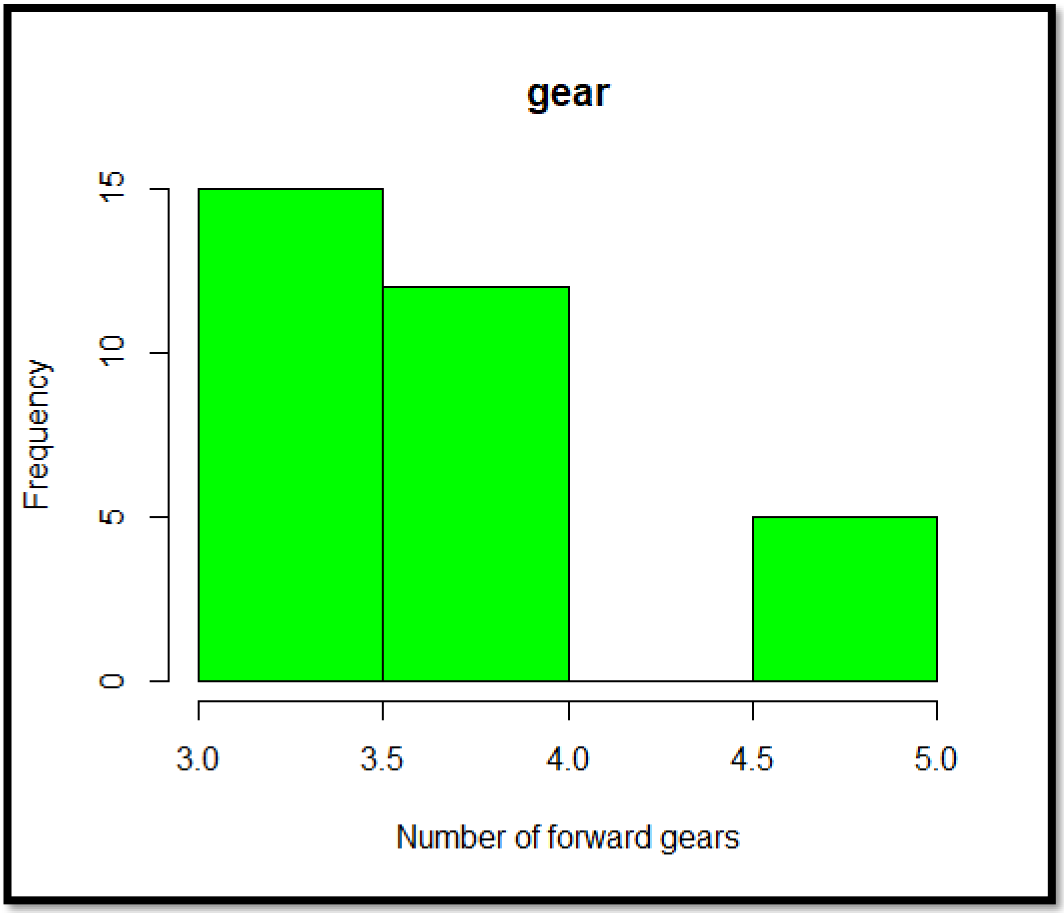

drat
5.0
4.5
4.0
3.5
3.0
Rear axle ratio

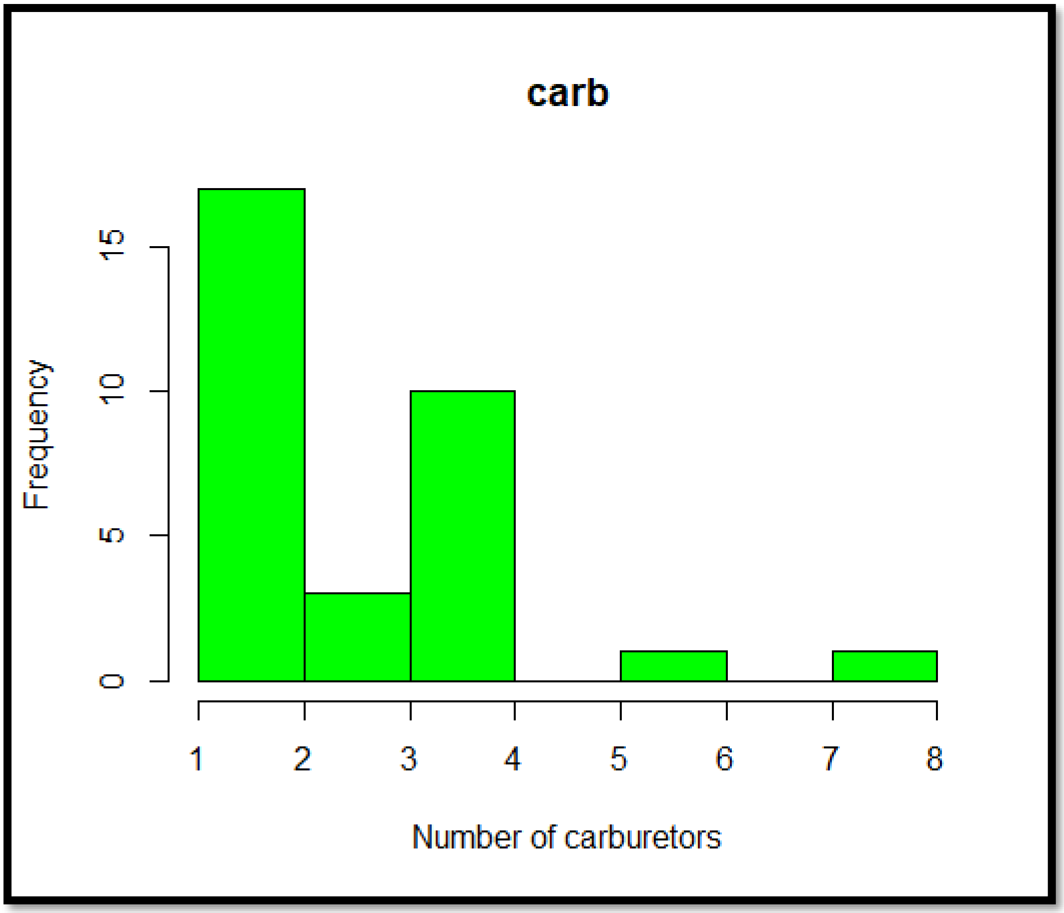

wt
5
4
3
2
Weight (1000 lbs)


qsec
22
20
18
16
1/4 mile time

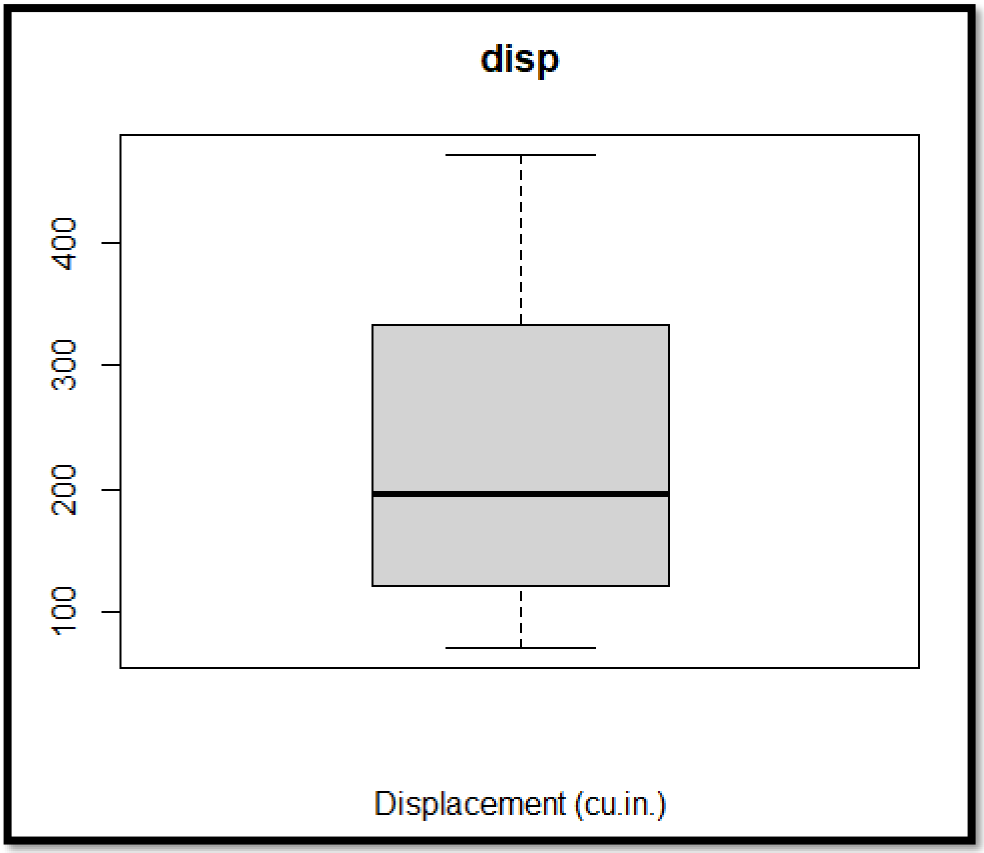

Pie of vs
56.2
43.8
🟥 V-shaped
🟦 straight


Pie of am
59.4
40.6
🟥 automatic
🟦 manual


gear
Frequency
0
5
10
15
3.0
3.5
4.0
4.5
5.0
Number of forward gears


carb
Frequency
0
5
10
15
1
2
3
4
5
6
7
8
Number of carburetors

Boxplot of (disp, hp, qsec)

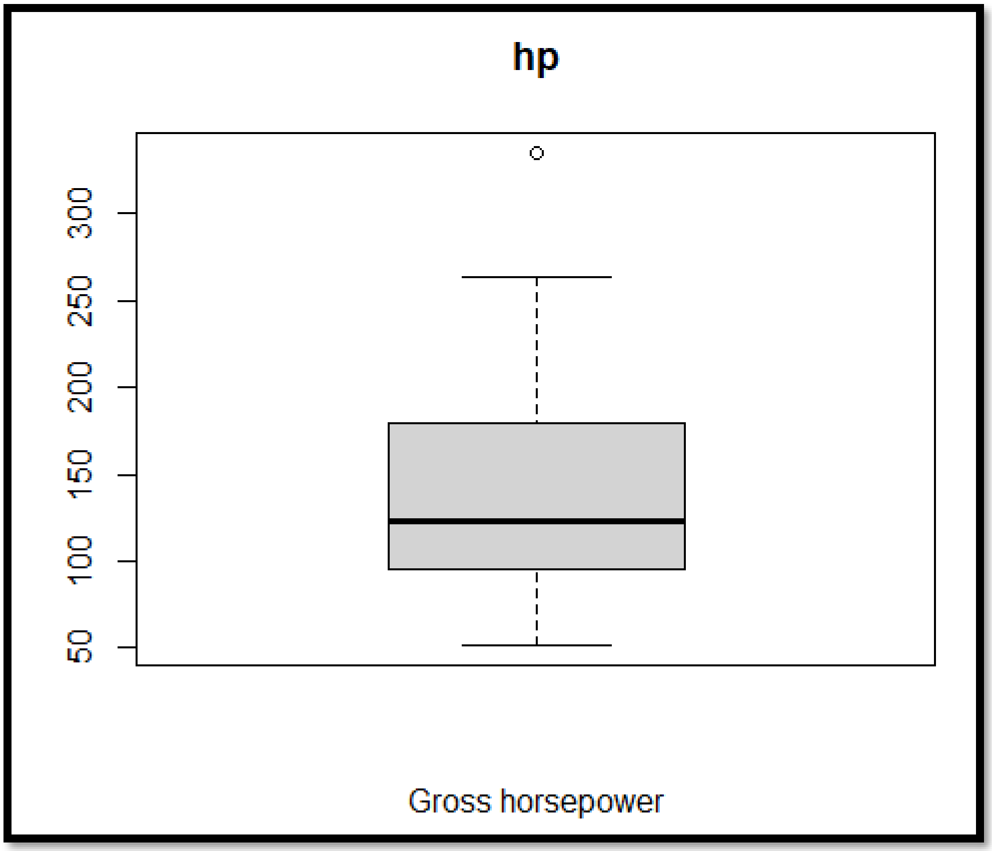


Disp percentile:	25%	50%	75%
	120.825	196.300	326.000

Deducing: 25% of cars have disp <= 120.825

50% of cars have disp <= 196.300

75% of cars have disp <= 326.000

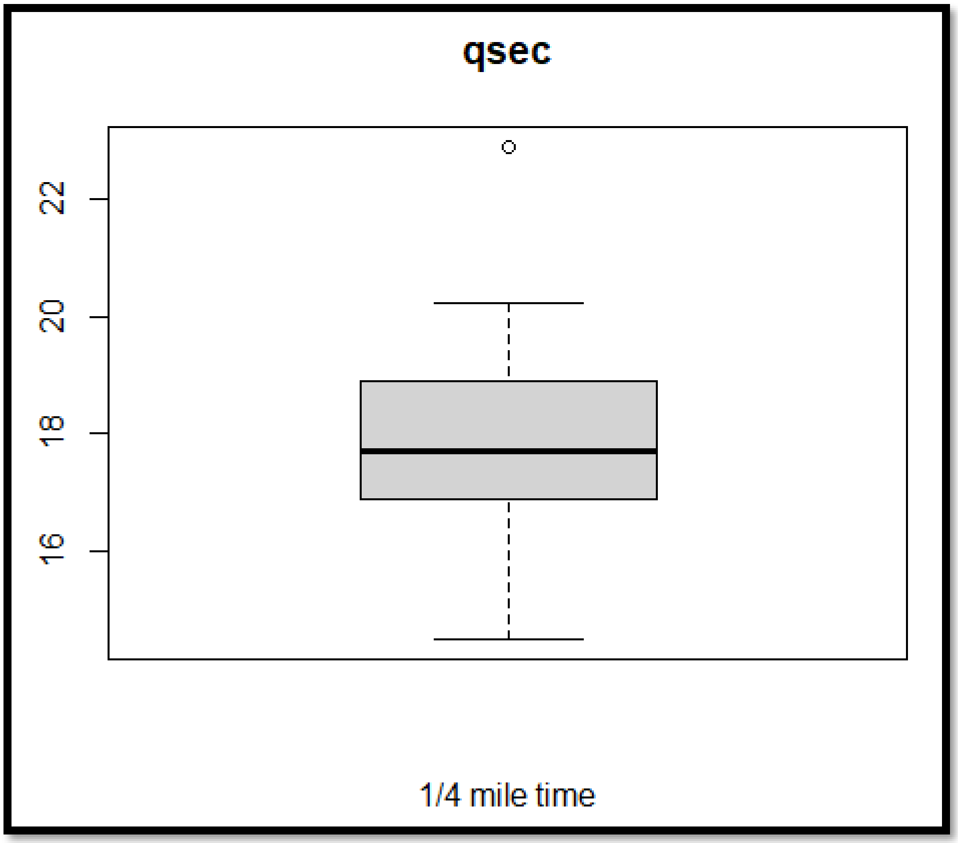


hp percentile:	25%	50%	75%
	96.5	123.0	180.0

Deducing: 25% of cars have hp <= 96.5

50% of cars have hp <= 123.0

75% of cars have hp <= 180.0



qsec percentile:	25%	50%	75%
	16.8925	17.7100	18.9000

Deducing: 25% of cars have qsec <= 16.8925

50% of cars have qsec<= 17.7100

75% of cars have qsec <= 18.9000

Distributions:

A:

Algorithm:

Get all occurrences of cars weighing 3.4 lbs. or more and dividing it by the total number of occurrences

Inputs:

X = filter(mtcars, wt >= 3.4)$wt, mean = mean(mtcars$wt) ,

sd = sd(mtcars$wt), result = sum(dnorm(a_x, mean = a_mean, sd = a_sd)) / sum(dnorm(mtcars$wt, mean = a_mean, sd = a_sd))

Output:

Result = 0.5137463

B:

Algorithm:

Prob = length(filter(mtcars, am == '1')$am) / length(mtcars$am)

Using function pbinom(x = 18, 32, Prob)

Inputs:

X = 13, n = 32 , p = Prob

Output:

Result = 0. 9751365

C:

Algorithm:

Using function pbinom(4, 12, 0.2)

Output:

Result = 0.9274445

Permutations & combinations:

A:

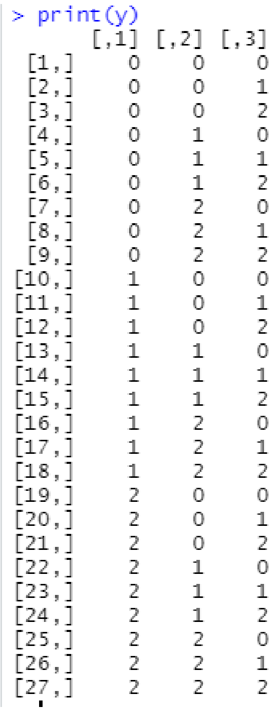
Algorithm:

1. Using function permutations().
2. Using loops.

Inputs:

N=3, r=3,repeats.allowed=T

Output:



```
> print(y)
      [,1] [,2] [,3]
 [1,]    0    0    0
 [2,]    0    0    1
 [3,]    0    0    2
 [4,]    0    1    0
 [5,]    0    1    1
 [6,]    0    1    2
 [7,]    0    2    0
 [8,]    0    2    1
 [9,]    0    2    2
[10,]    1    0    0
[11,]    1    0    1
[12,]    1    0    2
[13,]    1    1    0
[14,]    1    1    1
[15,]    1    1    2
[16,]    1    2    0
[17,]    1    2    1
[18,]    1    2    2
[19,]    2    0    0
[20,]    2    0    1
[21,]    2    0    2
[22,]    2    1    0
[23,]    2    1    1
[24,]    2    1    2
[25,]    2    2    0
[26,]    2    2    1
[27,]    2    2    2
```

27

B:

Algorithm:

1. Using function combinations().
2. Total = $^{9}C_{3}$ and n_ways = $^{2}C_{1}$ prob = n_ways/total
   Where $^{n}C_{r}$ = factorial(n)/(factorial(n-r)* factorial(r))

Inputs:

X = seq(1,9,by=1),n1=9,r1=3,n2=2,r2=1

Outputs:

2/84 = 0.02380952