Data Statistics and Processing using R

R scripting language is widely used amongst researchers for performing statistical analysis and plotting results. It has a very good community and university support. It is a GNU project. Measures of object similarity and dissimilarity and relations help learn more about data to start with.

https://www.r-project.org/

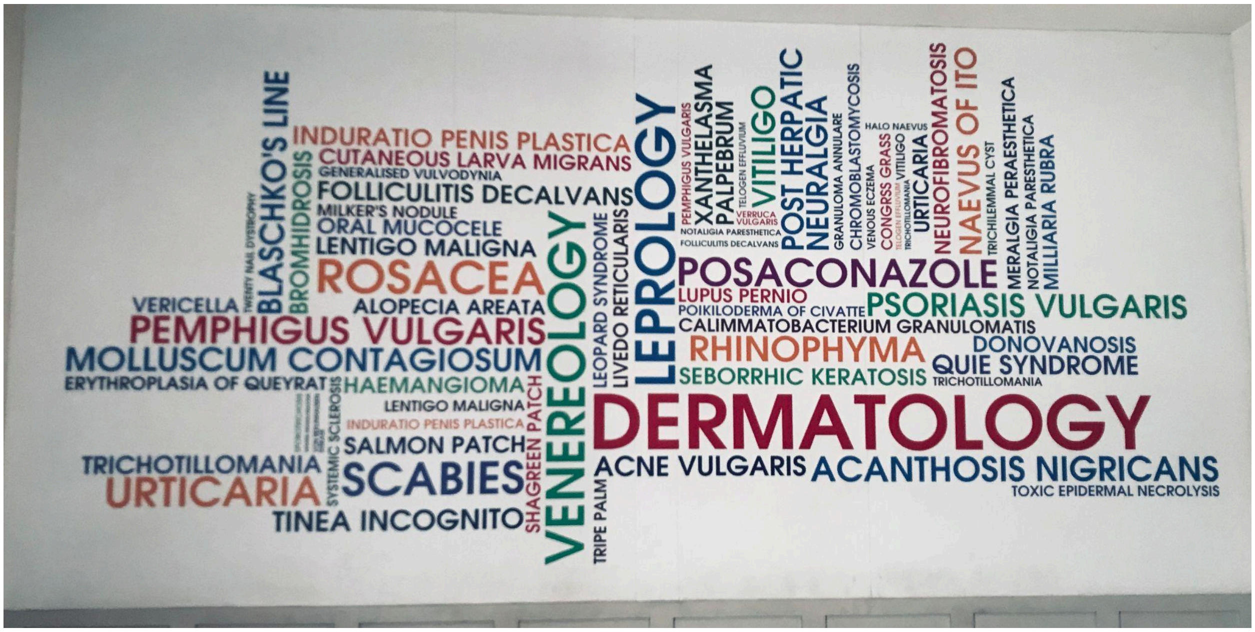
Wordcloud/Tag cloud

An application of plotting simply the trends with eye catching visualization is Word Cloud also known as Tag Cloud. It is an image showcasing trending words from a given domain based on their popularity. The dataset after certain cleaning applied is referred to find out words with their frequencies and given as input to the plotting library. The word cloud plotting displays beautifully the word having higher importance (i.e. frequencies/occurances) with relatively more bold and bigger font size compared to other words having lesser importance. Pictorial representation of analytics is always awesome.

Many times the fancy background of words can be used to highlight trends while celebrities are walking the red carpet or giving a pose. Many architects/interior designers use this technology to create giant wallpapers which actually convey achievements of the company in general or such important things.

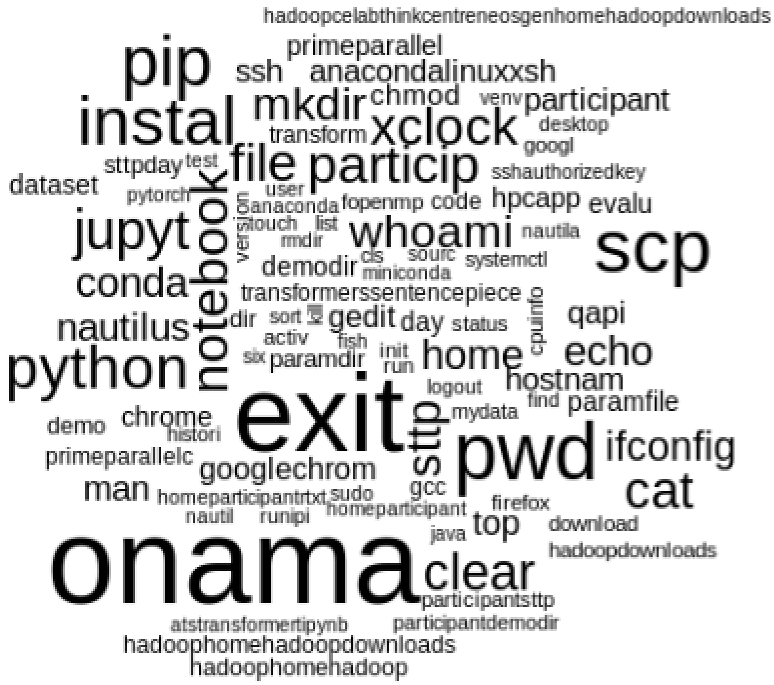
Impatient Patients learn Dermatology

For example in the waiting lounge/behind reception a wall poster containing at a Skin Clinic, Word Cloud of dermatology keywords will make some sense to patients constructively.



Participants' Interest

Another example showcasing participants' interest during a technical workshop based on commands they have practiced towards the end of the workshop using a Word Cloud during the closing ceremony.



Trending IT JOBS

Another real life example where trending jobs in IT sector can be better displayed as below instead of just a table.

https://www.kaggle.com/datasets/saurav0507/it-job-opportunities-dataset-2019-20h
ttps://www.kaggle.com/datasets/saurav0507/it-job-opportunities-dataset-2019-2

https://www.kaggle.com/datasets/saurav0507/it-job-opportunities-dataset-2019-2023

R code to practice Word Cloud

```
library("tm")
library("corpus")
library("SnowballC")
cname_jr <-
file.path("/home/jigarpandya/my-dev/R/tagcloud/it-job-opportunities-dataset/data")
print(cname_jr)
print(dir(cname_jr))

text_corpus_jr <- Corpus(DirSource(cname_jr))
```

```
text_corpus_jr <- tm_map(text_corpus_jr, stripWhitespace)
text_corpus_jr <- tm_map(text_corpus_jr, content_transformer(tolower))
text_corpus_jr <- tm_map(text_corpus_jr, removeWords, stopwords("english"))
text_corpus_jr <- tm_map(text_corpus_jr, stemDocument)
text_corpus_jr <- tm_map(text_corpus_jr, removeNumbers)
text_corpus_jr <- tm_map(text_corpus_jr, removePunctuation)
dtm_jr <- DocumentTermMatrix(text_corpus_jr)
freq_jr <- sort(colSums(as.matrix(dtm_jr)), decreasing=TRUE)
head(freq_jr)
library(wordcloud)
wordcloud(names(freq_jr), freq_jr, min.freq=100)
```

Exercises on Word Cloud/Tag Cloud:

1. Computer Engineers, relate the concept of Histogram as well as Hadoop wordcount program and establish connection to WordCloud/TagCloud example technically/logically.The orientation is for now just for adjustment it seems.
2. Either using the given R code or over the web do create Word Cloud/Tag cloud for your domain of interest dataset.
3. Try to see more aspects like orientation, colors, etc to bring more insight to reality from the same single presentation.

<u>Know your data at a glance</u>

The basic statistics like mean, median, mode, IQR and others let us learn about data at a glance. These facts convey information about the dataset/sample at hand.

- Overall dispersion of data points
- Data quality and biasedness, outliers if any

R provides various library functions to perform certain operations to record tabular as well as graphical representations of same data or statistics/results.

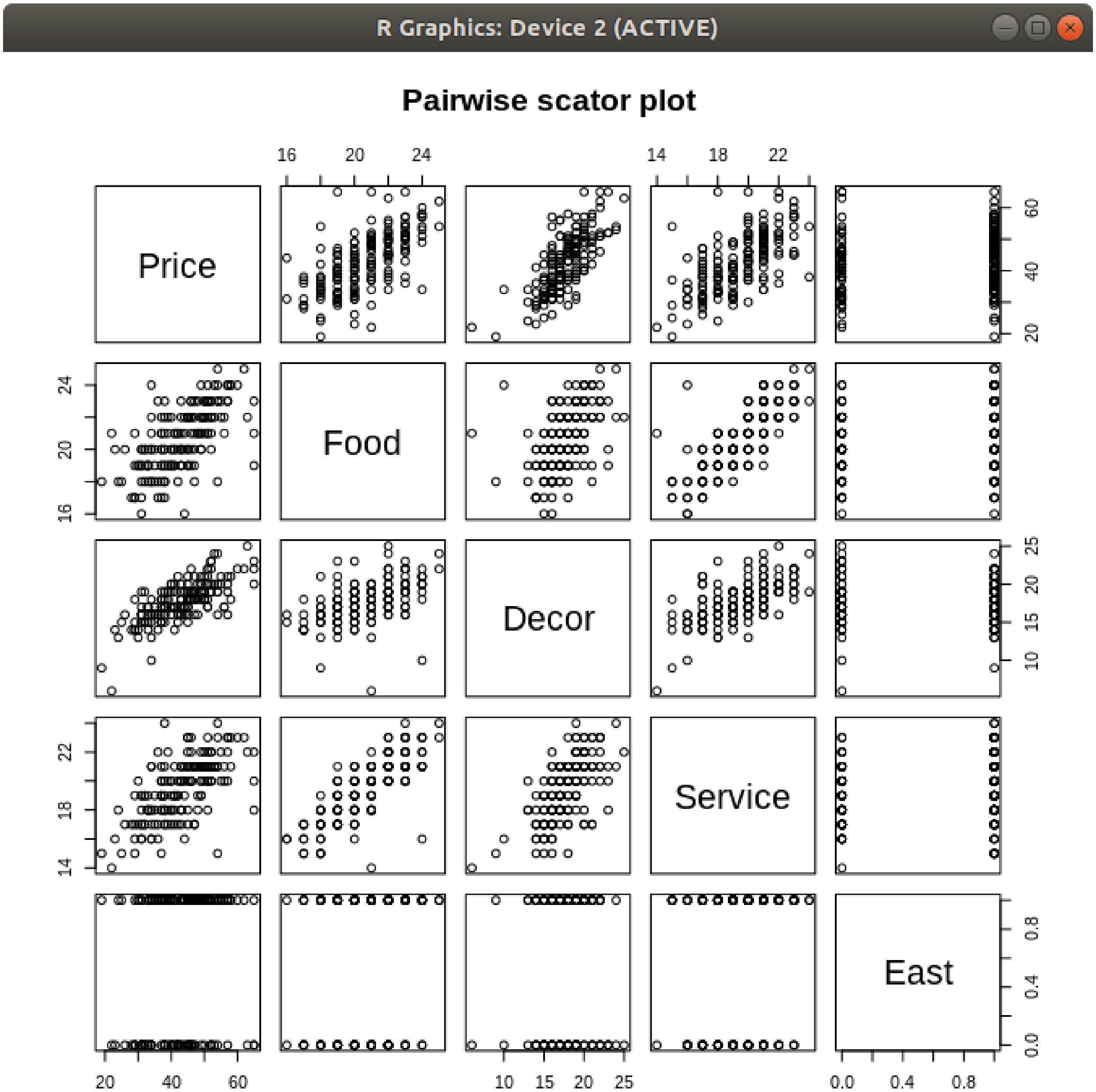
Practices

#1

nyc=read.csv("nyc.csv")
nyc
summary(nyc)

```
> summary(nyc)
     Price           Food           Decor           Service           East
 Min.   :19.0   Min.   :16.0   Min.   : 6.00   Min.   :14.0   Min.   :0.000
 1st Qu.:36.0   1st Qu.:19.0   1st Qu.:16.00   1st Qu.:18.0   1st Qu.:0.000
 Median :43.0   Median :20.5   Median :18.00   Median :20.0   Median :1.000
 Mean   :42.7   Mean   :20.6   Mean   :17.69   Mean   :19.4   Mean   :0.631
 3rd Qu.:50.0   3rd Qu.:22.0   3rd Qu.:19.00   3rd Qu.:21.0   3rd Qu.:1.000
 Max.   :65.0   Max.   :25.0   Max.   :25.00   Max.   :24.0   Max.   :1.000
>
```

#2
plot(nyc,main="Pairwise scator plot")


R Graphics: Device 2 (ACTIVE)
Pairwise scator plot
Price
Food
Decor
Service
East
16
20
24
14
18
22
60
40
20
25
15
10
0.8
0.4
0.0
10
15
20
25
0.0
0.4
0.8

#3

linearMod <- lm(dist ~ speed, data=cars) # build linear regression model on full data

print(linearMod)

```
Call:
lm(formula = dist ~ speed, data = cars)

Coefficients:
(Intercept)        speed
    -17.579        3.932
```

#4

lm(Price~Food+Decor+Service+East,data=nyc)

```
> lm(Price~Food+Decor+Service+East,data=nyc)

Call:
lm(formula = Price ~ Food + Decor + Service + East, data = nyc)

Coefficients:
(Intercept)         Food        Decor      Service         East
 -24.023800     1.538120     1.910087    -0.002727     2.068050
```

#5

Rank

The maximum number of linearly independent columns (or rows) of a given matrix is called the rank of the matrix. A higher rank indicates more independence amongst the vectors within a given matrix.

install.packages("pracma")

library(pracma)

A=matrix(data=c(1,2,8,2,9,4,5,6,8,7,3,0,5,6,6,5,5,1,10,5,6,1,0,12,1),nrow=5,ncol=5,byrow=FALSE)

A

Rank(A)

```
> library(pracma)
> A
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    3    5    6
[2,]    2    5    0    5    1
[3,]    8    6    5    1    0
[4,]    2    8    6   10   12
[5,]    9    7    6    5    1
> Rank(A)
[1] 4
>
```

#6

Eigen

install.packages("matlab")   #optional

library(matlab)

me=matrix(data=c(1,0,0,0,2,0,0,0,3),nrow=3,ncol=3,byrow=F)

me

mei=eigen(me)

mei

```
> me=matrix(data=c(1,0,0,0,2,0,0,0,3),nrow=3,ncol=3,byrow=F)
> me
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    2    0
[3,]    0    0    3
> mei=eigen(me)
> mei
eigen() decomposition
$values
[1] 3 2 1

$vectors
     [,1] [,2] [,3]
[1,]    0    0    1
[2,]    0    1    0
[3,]    1    0    0

>
```

#7

Central tendency of data

Mean (average value)

data = c(30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110)

mean(data)

```
> data
 [1]  30  36  47  50  52  52  56  60  63  70  70 110
> mean(data)
[1] 58
>
```

Median (middle value)

median(data)

```
> data
 [1]  30  36  47  50  52  52  56  60  63  70  70 110
> median(data)
[1] 54
>
```

Mode (most common value)
package:pracma

```
> Mode(data)
[1] 52
>
```

data = c(30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110)

range(data)

```
> range(data)
[1]  30 110
>
```

quantile(data)

```
> quantile(data)
    0%    25%    50%    75%   100%
 30.00  49.25  54.00  64.75 110.00
> quantile
```

fivenum(data)

help(fivenum)
Returns Tukey's five number summary (minimum, lower-hinge, median, upper-hinge, maximum) for the input data.

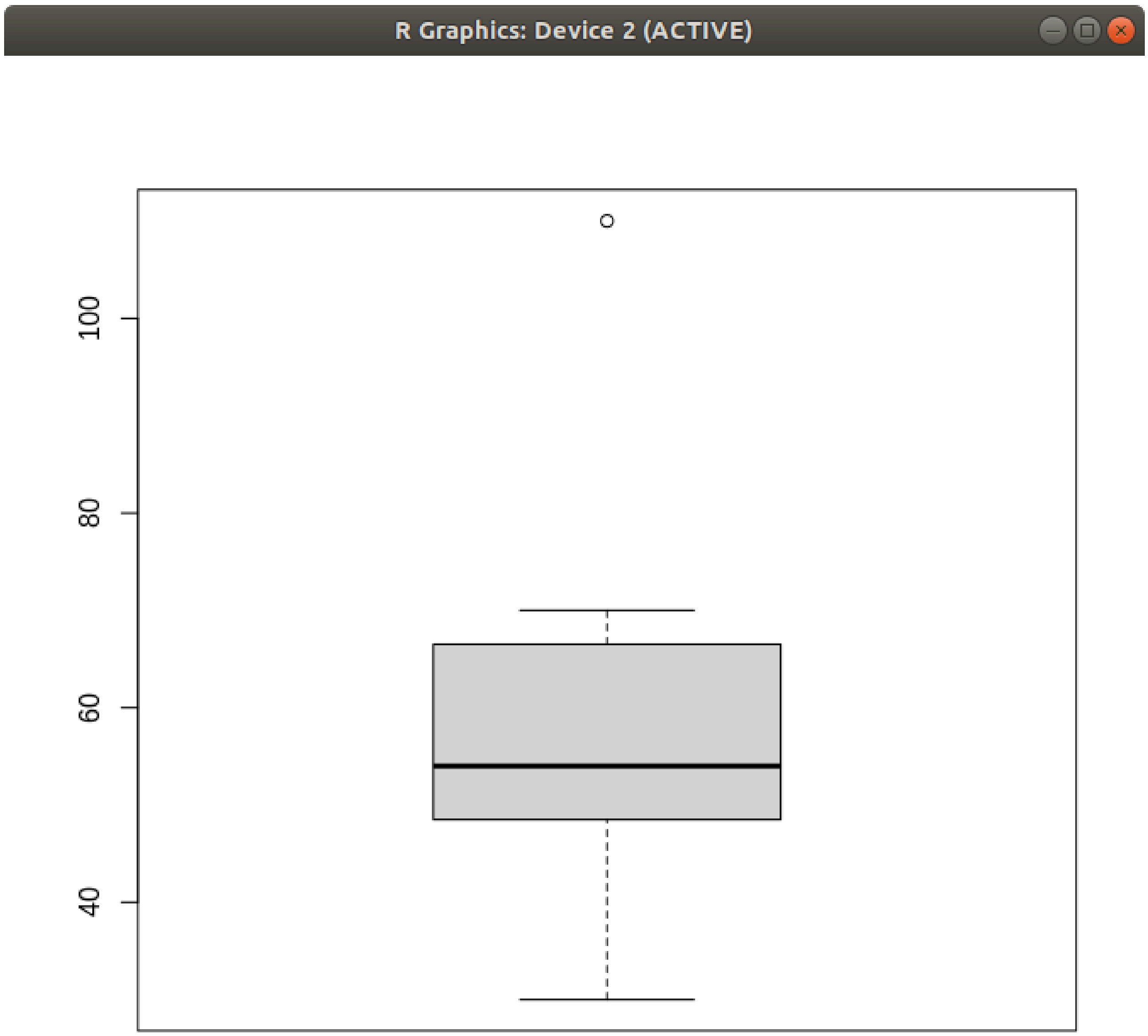
```
> fivenum(data)
[1]  30.0  48.5  54.0  66.5 110.0
> 
```

summary(data)

```
> summary(data)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  30.00   49.25   54.00   58.00   64.75  110.00 
> 
```

boxplot(data)

tab



table(data)

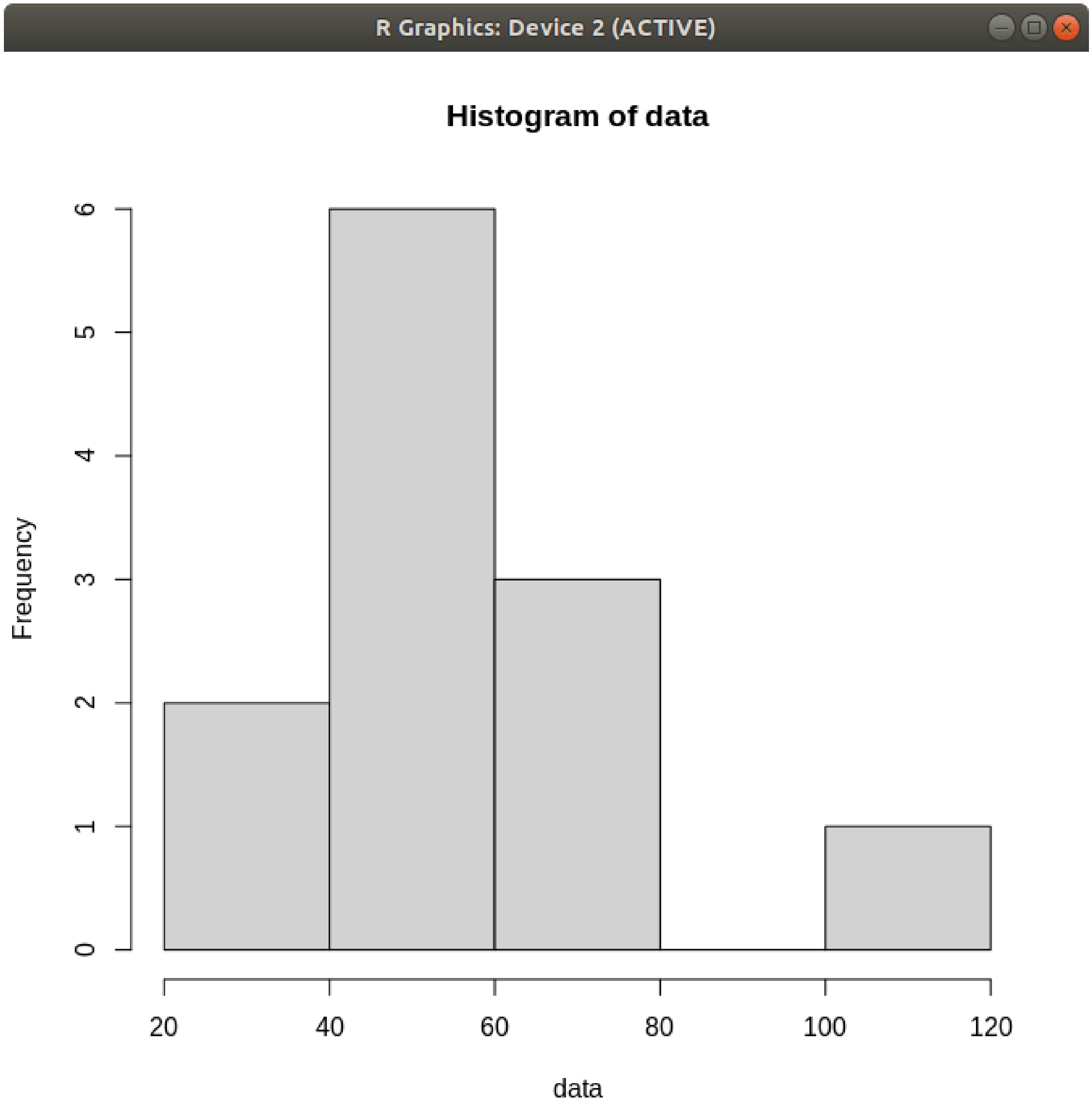
```
> table(data)
data
 30  36  47  50  52  56  60  63  70 110
  1   1   1   1   2   1   1   1   2   1
> 
```

names(table(data))[table(data)==max(table(data))]

```
> names(table(data))[table(data)==max(table(data))]
[1] "52" "70"
> 
```

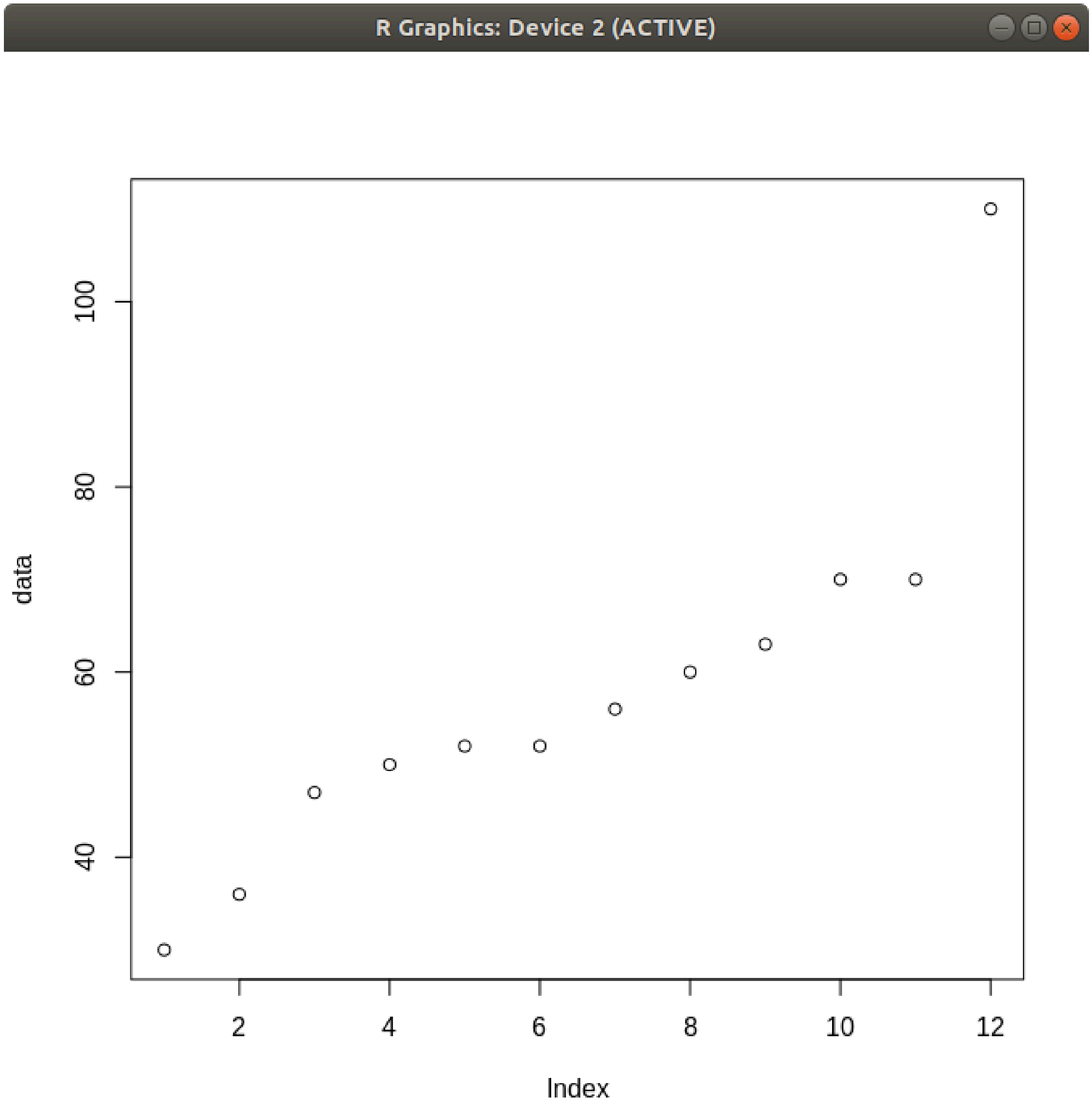
#8
Histogram

hist(data)



#9

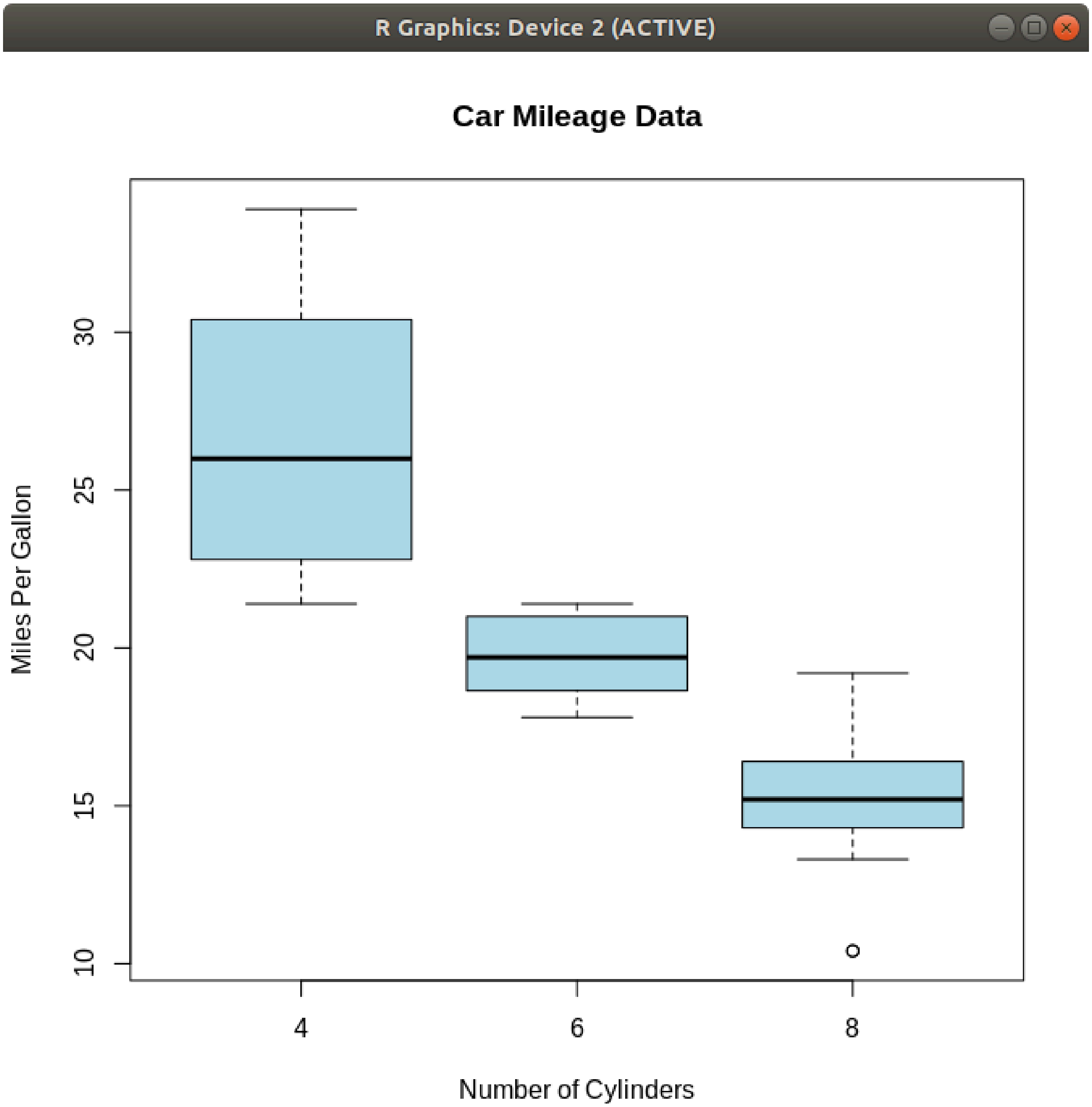
Scatter plot



#10

```
# Boxplot of MPG by Car Cylinders using the built-in mtcars dataset
boxplot(mpg ~ cyl, data = mtcars,
        main = "Car Mileage Data",
        xlab = "Number of Cylinders",
```

ylab = "Miles Per Gallon",
col = "lightblue")



#11

Distances

Dissimilarity Matrix Calculation

Euclidean distances are root sum-of-squares of differences,

and manhattan distances are the sum of absolute differences.

```
library(cluster)

ll=matrix(c(22,20,1,0,42,36,10,8),ncol=4)

distances=daisy(ll,metric=c("euclidean"))
```

square root (square(22-20) + square(1-0) + square (42-36) + square(10-8))
square root (4+1+36+ 4)
square root(45)
6.70

```
distances=daisy(ll,metric=c("manhattan"))
```

| 22-20| + |1-0| + |42-36| + |0-8|
2 + 1 + 6 + 2
11

```
> ll=matrix(c(22,20,1,0,42,36,10,8),ncol=4)
> distances=daisy(ll,metric=c("euclidean"))
> distances
Dissimilarities :
         1
2 6.708204

Metric :  euclidean
Number of objects : 2
> distances=daisy(ll,metric=c("manhattan"))
> distances
Dissimilarities :
   1
2 11

Metric :  manhattan
Number of objects : 2
> 
```

Exercises on basic statistics API and pots:

- Find out standard deviation
- Find out outliers theoretical and practically from your choice of observations and perform five-number summary using box plot/s
- Find out confusion matrix for your choice of
- Utilize cosine-similarity / Cosine measure
- Identify whether matrix is sparse or dense
- How do you detect noise in data?
- Draw QQ plot
- Prove by plot whether chosen dataset is symmetric or skewed (positive/negative)

Dataset

nyc.csv

```
Price,Food,Decor,Service,East
43,22,18,20,0
32,20,19,19,0
34,21,13,18,0
41,20,20,17,0
54,24,19,21,0
52,22,22,21,0
34,22,16,21,0
34,20,18,21,1
39,22,19,22,1
44,21,17,19,1
45,19,17,20,1
47,21,19,21,1
52,21,19,20,1
35,19,17,19,1
47,20,18,21,1
37,21,19,21,1
45,22,18,23,1
57,24,21,22,1
38,19,17,18,1
51,22,20,22,1
54,23,20,23,1
51,23,17,21,1
38,20,18,18,1
49,22,21,21,1
45,22,20,22,1
37,19,17,19,1
50,22,19,22,1
43,20,16,18,1
49,22,19,20,1
65,21,20,20,1
```

34,20,16,18,1
51,21,20,18,1
49,20,19,19,1
51,23,22,23,1
62,25,22,23,1
50,23,21,21,1
51,21,18,20,1
52,22,19,22,1
57,24,20,23,1
49,21,20,21,1
33,19,17,18,1
43,19,17,17,1
41,21,17,18,1
58,24,21,23,1
56,22,17,21,1
44,22,17,21,1
37,20,15,20,1
56,21,17,20,1
58,24,18,21,1
44,16,16,16,1
46,20,18,20,1
40,20,20,20,1
39,19,18,17,1
36,17,14,17,1
34,18,15,16,1
54,18,16,15,1
51,23,17,20,1
41,20,14,19,1
40,22,17,20,1
24,18,13,18,1
53,24,20,21,1
31,19,16,17,1
35,18,16,17,1
49,20,19,19,1
38,19,15,17,1

48,21,16,18,1
43,20,19,20,1
29,17,14,15,1
37,17,18,15,1
55,22,21,20,1
37,22,18,20,1
55,23,20,22,1
49,24,20,22,1
33,19,14,18,1
52,23,20,22,1
47,22,16,21,1
43,21,16,20,1
33,18,17,18,1
38,18,16,19,1
48,21,18,19,1
50,21,18,21,1
46,23,19,21,1
38,23,19,24,1
33,20,16,19,1
46,23,19,23,1
37,19,15,19,1
50,23,18,20,1
54,25,24,24,1
41,21,19,21,1
37,21,15,18,1
50,22,18,21,1
60,24,22,23,1
36,23,16,22,1
54,23,19,21,1
39,19,18,18,1
35,20,16,18,1
30,19,13,20,1
41,19,17,19,1
30,19,14,17,1
25,18,15,15,1

43,19,18,21,1
45,20,15,17,1
57,23,16,20,1
32,18,15,17,0
51,24,21,21,1
48,23,20,21,1
36,18,16,16,1
37,20,17,19,0
31,20,19,19,1
47,23,19,21,1
40,19,16,19,1
37,18,16,18,1
43,23,20,21,1
51,23,22,21,1
19,18,9,15,1
28,17,14,17,0
22,21,6,14,0
41,19,17,19,0
33,19,15,18,0
29,19,15,17,0
33,19,17,18,0
45,19,16,18,0
38,17,16,17,0
52,20,23,20,0
38,20,17,19,0
47,18,18,17,0
46,22,18,20,0
40,20,17,18,0
32,20,15,19,0
65,19,23,18,0
47,19,21,17,0
65,23,22,22,0
45,20,17,21,0
46,22,22,20,0
44,20,19,20,0

40,19,19,18,0
46,19,18,20,0
32,19,15,17,0
23,20,14,16,0
42,18,21,17,0
29,21,18,19,0
49,21,18,20,0
53,22,24,21,0
45,22,19,21,0
63,22,25,22,0
52,23,23,21,0
40,19,20,17,0
45,22,21,23,0
38,21,17,20,0
38,18,17,18,0
42,21,16,20,0
57,23,19,23,0
39,21,19,20,0
43,20,18,18,0
29,17,14,16,0
42,20,16,19,0
50,22,19,21,0
34,18,16,17,0
31,16,15,16,0
31,20,17,19,0
46,21,19,22,0
42,21,15,19,0
31,19,16,18,0
31,17,15,16,0
26,20,16,17,0
31,18,16,17,0
38,22,17,21,0
34,24,10,16,0

Dataset

Cars (Dataset is available in R Shell by default)

cars

	speed	dist
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10
7	10	18
8	10	26
9	10	34
10	11	17
11	11	28
12	12	14
13	12	20
14	12	24
15	12	28
16	13	26
17	13	34
18	13	34
19	13	46
20	14	26
21	14	36
22	14	60
23	14	80
24	15	20
25	15	26
26	15	54
27	16	32
28	16	40
29	17	32
30	17	40

31	17	50
32	18	42
33	18	56
34	18	76
35	18	84
36	19	36
37	19	46
38	19	68
39	20	32
40	20	48
41	20	52
42	20	56
43	20	64
44	22	66
45	23	54
46	24	70
47	24	92
48	24	93
49	24	120
50	25	85

Dataset

mtcars **(Dataset is available in R Shell by default)**

```
> mtcars
                   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
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
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
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
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
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
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
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
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2