Data Science in Statistical Methods using R

Md Sayeef Alam

21/09/2020

Day 1

Session 1: Application of Regression and Multiple Regression in Data Science

Dr. R. K. Jana, IIM Raipur

Simple addition in R

```
1+1
```

```
## [1] 2
```

Some packages to be installed

```
install.packages("matlib", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("corpcor", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("GPArotation", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("psych", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("FactoMineR", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("tseries", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("corrplot", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("tseries", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("ggpubr", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("tidyverse", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("Hmisc", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("dplyr", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("ggplot2", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("lattice", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("grid", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("DMwR", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("stats", dependencies = T,repos = "http://cran.us.r-project.org")
install.packages("nortest", dependencies = T,repos = "http://cran.us.r-project.org")
```

Adding the libraries corresponding to packages.

```
library(dplyr)
library(tseries)
library(matlib)
library(corpcor)
library(GPArotation)
library(psych)
library(FactoMineR)
library(corrplot)
library(ggpubr)
library(lattice)
```

```
library(grid)
library(nortest)
library(stats)
library(DMwR)
library(ggplot2)
```

Reading xls and xlsx files

```
install.packages("gdata", dep = T,repos = "http://cran.us.r-project.org")
library(gdata)
xls.data = read.xls("file.xls")
```

You need to specify the sheetIndex (sheet number)

```
install.packages("xlsx", dep = T,repos = "http://cran.us.r-project.org")
library(xlsx)
xlsx.data = read.xlsx("file.xlsx", sheetIndex = 1)
```

Linear Regression

Simple Linear Regression

1 dependent (y)

1 independent (x)

Assumptions

1. Relationships between the above two must be linear
2. Residuals should be normally distributed
3. Residuals should be homoscedastic
4. Residuals should be independent

Homoscedasticity means same variance, error term (i.e. distance of the points from the fitted line) should be same across all values of the independent variables.

Heteroscedasticity is when the error varies with the values of the independent variables.

Several measures are there to check for homoscedasticity

```
library(datasets)
data(cars)
```

Lets check the variables inside the dataset

```
names(cars)
```

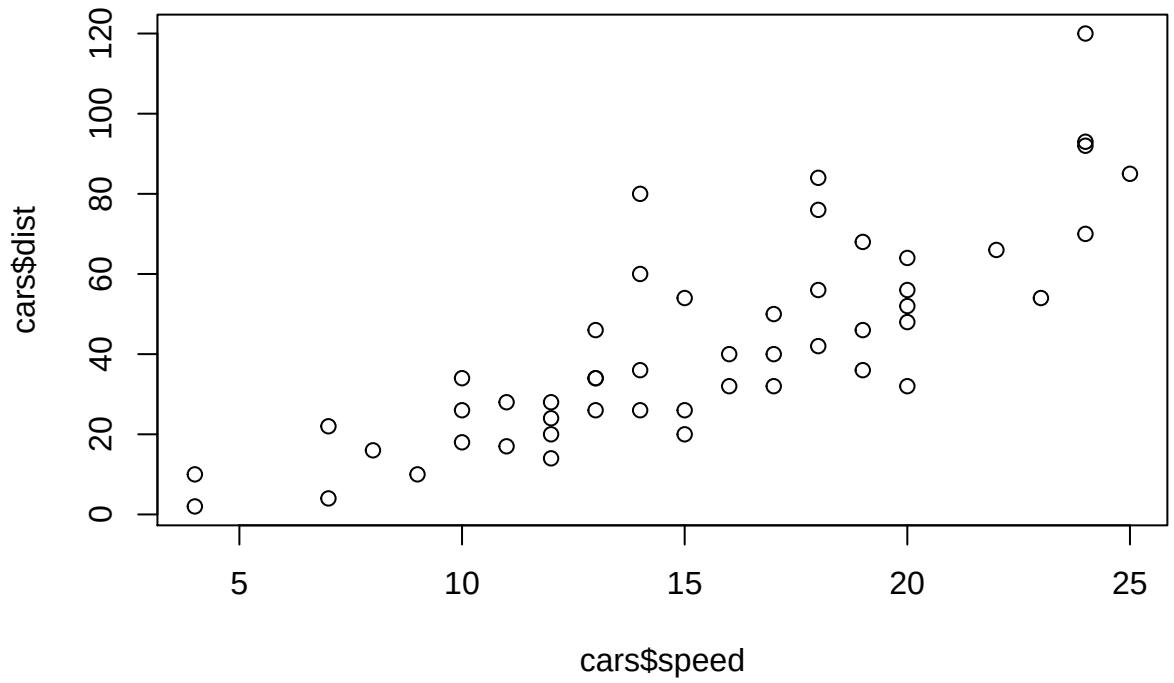
```
## [1] "speed" "dist"
```

```
head(cars)
```

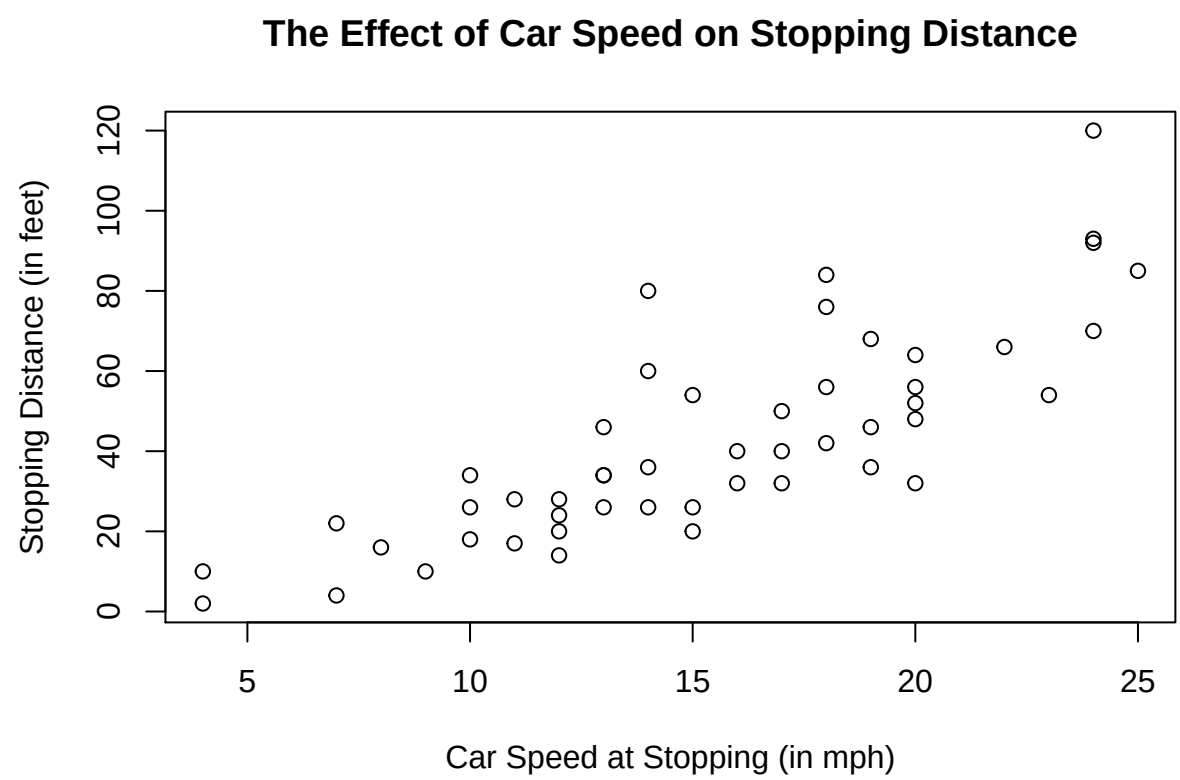
```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

Lets plot some parameters specifically speed vs distance

```r
plot(cars$speed, cars$dist)
```

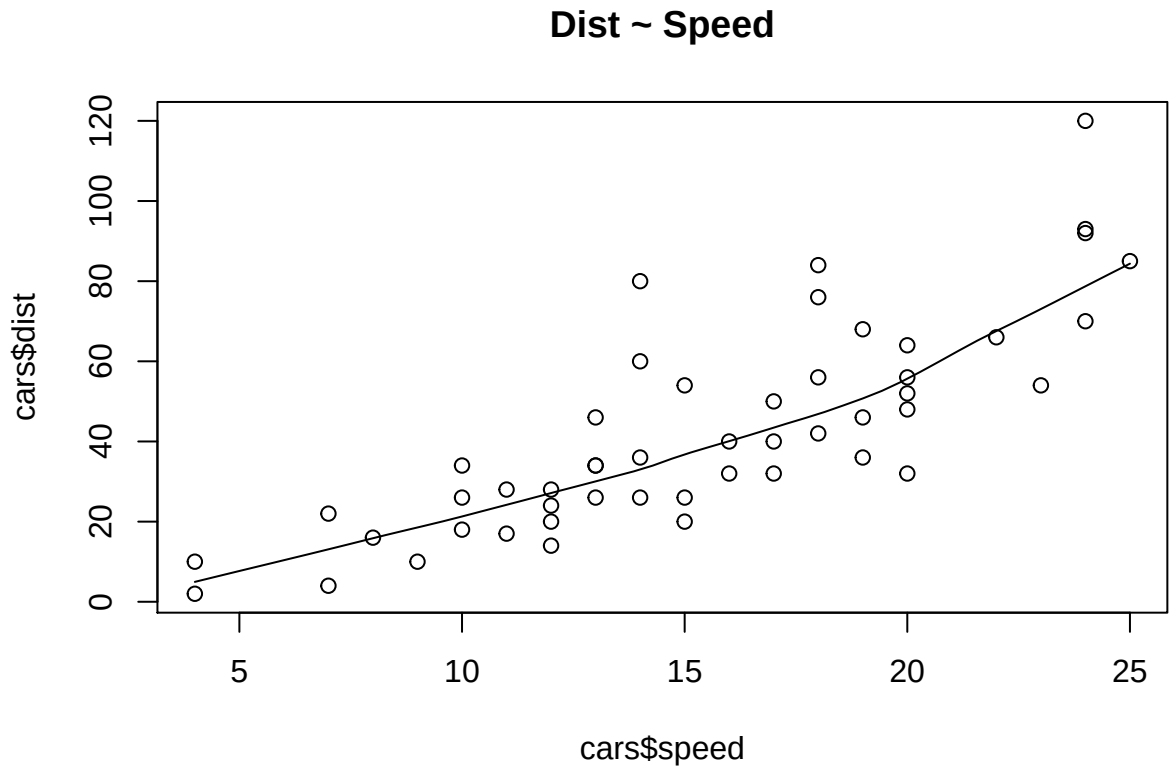



```r
plot(cars$speed, cars$dist, xlab = "Car Speed at Stopping (in mph)",
     ylab = "Stopping Distance (in feet)", main = "The Effect of Car Speed on Stopping Distance")
```



Fitting a smooth line

```
scatter.smooth(x=cars$speed, y=cars$dist, main="Dist ~ Speed")
```

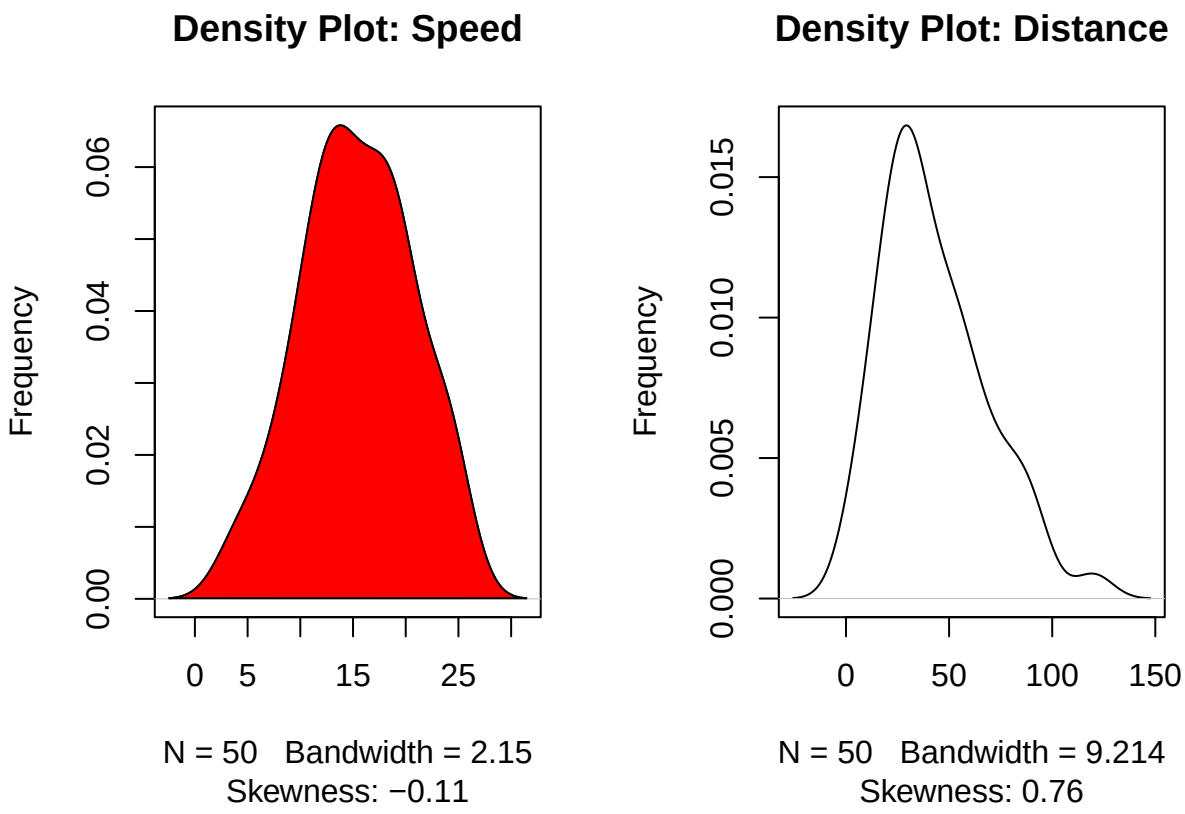


Density plots for speed and distance

```
library(e1071)
par(mfrow=c(1, 2))

plot(density(cars$speed), main="Density Plot: Speed", ylab="Frequency", sub=paste("Skewness:", round(e1
polygon(density(cars$speed), col="red")

plot(density(cars$dist), main="Density Plot: Distance", ylab="Frequency", sub=paste("Skewness:", round(
```



Linear regression model fitting

```
carmod <- lm(dist ~ speed, data = cars)
summary(carmod)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

95% CI

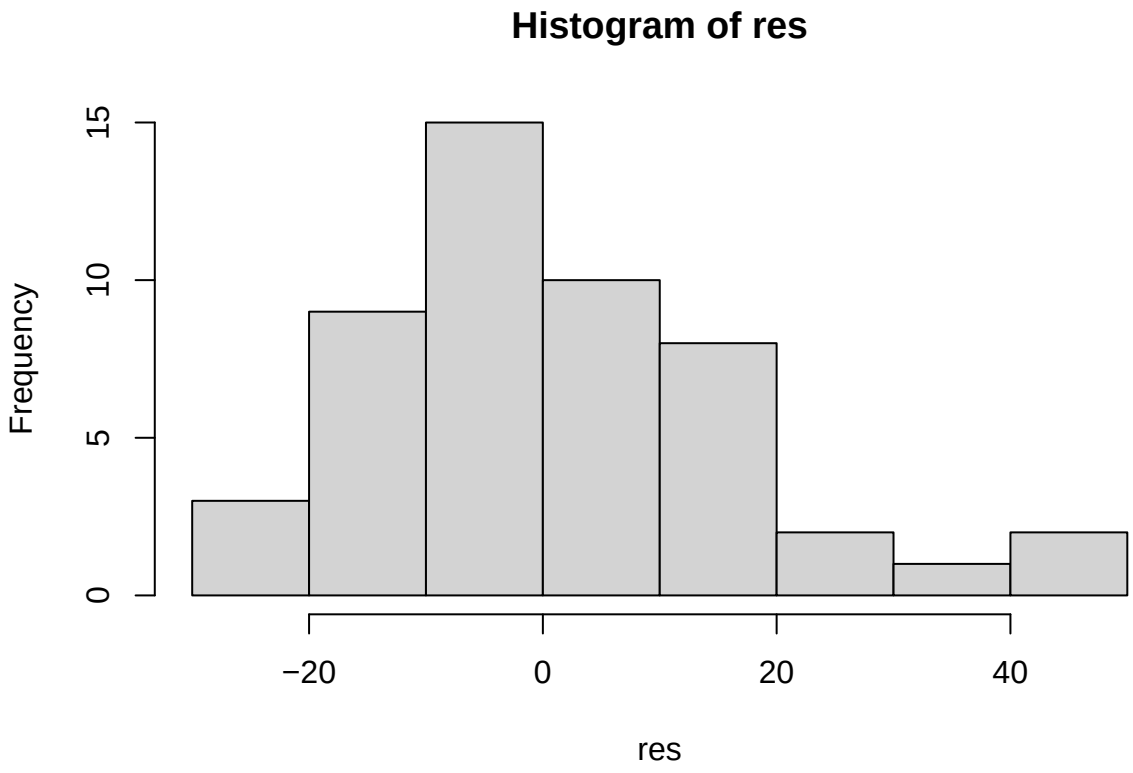
```
confint(carmod, level = 0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) -31.167850 -3.990340
```

```
## speed         3.096964  4.767853
```

Normality of residuals check

```
res = carmod$residuals
hist(res)
```



Interpretation

The coefficients in linear regression model states that with a unit change in x how much change is expected in y.

Session 2: Data Science & Sample Survey

Prof. G. N. Singh, IIT (ISM) Dhanbad

Word Statistics

In a literal sense

Plural sense some sort of data numerical figures in our day to day arising, runs and all figures are called statistics

In singular collection of methods and principles in a book,

Procedure to collection, analyse and interpret the data is called statistics

Statistics never claims 100% accuracy

Statistics is the science of decision making. As no decision is free from error.

Hope that PPTs will be provided soon.

Day 2

Session 3: Introduction to R

Prof. G. N. Singh, IIT (ISM) Dhanbad

Theory and PPT will be available.

Session 4: Introduction to R

Dr. Anup Kumar Sharma, NIT Raipur

Theory and PPT will be available.

Session 5: Graphical representation and normality testing in R

Dr. Dhaval Maheta,

```
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
```

```
attach(mtcars)
```

```
## The following object is masked from package:ggplot2:
```

```
##
##     mpg
```

Find the mean for all columns

the number in between is the parameter denoting the 1 = row and 2 = column. row mean is useless so we are looking at column mean

```
apply(mtcars,2,mean)
```

```
##        mpg        cyl       disp         hp       drat         wt       qsec
##  20.090625   6.187500 230.721875 146.687500   3.596563   3.217250  17.848750
##         vs         am       gear       carb
##   0.437500   0.406250   3.687500   2.812500
```

Now similarly for median and mode

```
apply(mtcars,2,median)
```

```
##     mpg     cyl    disp      hp    drat      wt    qsec      vs      am    gear
##  19.200   6.000 196.300 123.000   3.695   3.325  17.710   0.000   0.000   4.000
##    carb
##   2.000
```

```
apply(mtcars,2,mode)
```

```
##       mpg       cyl      disp        hp      drat        wt      qsec        vs
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##        am      gear      carb
## "numeric" "numeric" "numeric"
```

aggregate function helps to calculate the required function (mean/median/mode/sd) for each category of independent variable

```
aggregate(mpg~am,FUN = mean)
```

```
##   am      mpg
## 1  0 17.14737
## 2  1 24.39231
```

```
aggregate(mpg~am,FUN = median)
```

```
##   am  mpg
## 1  0 17.3
## 2  1 22.8
```

```
aggregate(mpg~am,FUN = mode)
```

```
##   am     mpg
## 1  0 numeric
## 2  1 numeric
```

```
aggregate(mpg~am,FUN = sd)
```

```
##   am      mpg
## 1  0 3.833966
## 2  1 6.166504
```

Find 3 way table to summary statistics and describeBy (available in psych library)

```
aggregate(mpg~am+vs,FUN = mean)
```

```
##   am vs      mpg
## 1  0  0 15.05000
## 2  1  0 19.75000
## 3  0  1 20.74286
## 4  1  1 28.37143
```

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
describeBy(mpg,am)
```

```
##
##  Descriptive statistics by group
## group: 0
##    vars  n  mean   sd median trimmed  mad  min  max range skew kurtosis   se
## X1    1 19 17.15 3.83   17.3   17.12 3.11 10.4 24.4    14 0.01     -0.8 0.88
## ------------------------------------------------------------
## group: 1
##    vars  n  mean   sd median trimmed  mad min  max range skew kurtosis   se
## X1    1 13 24.39 6.17   22.8   24.38 6.67  15 33.9  18.9 0.05    -1.46 1.71
```

best descriptive summarizer called the stargazer, the flip = T command helps to transpose the rows and columns

```
install.packages("stargazer", dependencies = T,repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yp/0237rgk11t35swrh_2f9h_200000gn/T//RtmpedLe6o/downloaded_packages
```

```
library(stargazer)

stargazer(mtcars,type = "text", title = "Descriptive Stats", digits = 1)
```

```
##
## Descriptive Stats
```

```
## ========================================================
## Statistic N  Mean  St. Dev. Min  Pctl(25) Pctl(75) Max
## --------------------------------------------------------
## mpg       32 20.1    6.0     10    15.4     22.8    34
## cyl       32  6.2    1.8     4      4        8      8
## disp      32 230.7  123.9    71   120.8     326    472
## hp        32 146.7   68.6    52    96.5     180    335
## drat      32  3.6    0.5    2.8    3.1      3.9    4.9
## wt        32  3.2    1.0    1.5    2.6      3.6    5.4
## qsec      32 17.8    1.8    14.5   16.9     18.9   22.9
## vs        32  0.4    0.5     0      0        1      1
## am        32  0.4    0.5     0      0        1      1
## gear      32  3.7    0.7     3      3        4      5
## carb      32  2.8    1.6     1      2        4      8
## --------------------------------------------------------
```

```
stargazer(mtcars,type = "text", title = "Descriptive Stats", digits = 1, flip = T)
```

```
##
## Descriptive Stats
## ============================================================
## Statistic mpg  cyl disp   hp   drat wt  qsec vs  am  gear carb
## ------------------------------------------------------------
## N          32  32   32    32    32  32   32  32  32   32   32
## Mean      20.1 6.2 230.7 146.7 3.6  3.2 17.8 0.4 0.4 3.7  2.8
## St. Dev.  6.0  1.8 123.9 68.6  0.5  1.0 1.8  0.5 0.5 0.7  1.6
## Min        10   4   71    52   2.8  1.5 14.5  0   0   3    1
## Pctl(25)  15.4  4  120.8 96.5  3.1  2.6 16.9  0   0   3    2
## Pctl(75)  22.8  8   326   180  3.9  3.6 18.9  1   1   4    4
## Max        34   8   472   335  4.9  5.4 22.9  1   1   5    8
## ------------------------------------------------------------
```

Try the following codes to obtain data like SPSS

```
install.packages("summarytools", dependencies = T,repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yp/0237rgk11t35swrh_2f9h_200000gn/T//RtmpedLe6o/downloaded_packages
```

```
install.packages("ellipsis", dependencies = T,repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yp/0237rgk11t35swrh_2f9h_200000gn/T//RtmpedLe6o/downloaded_packages
```

```
library(summarytools)
library(ellipsis)
```

```
attach(mtcars)
```

```
## The following objects are masked from mtcars (pos = 6):
##
##     am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt

## The following object is masked from package:ggplot2:
##
##     mpg
```

```
summarytools::descr(mtcars)
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## Descriptive Statistics
## mtcars
## N: 32
##
##                       am     carb      cyl     disp     drat     gear       hp      mpg     qsec
## ----------------- -------- -------- -------- -------- -------- -------- -------- -------- --------
##              Mean     0.41     2.81     6.19   230.72     3.60     3.69   146.69    20.09    17.85
##           Std.Dev     0.50     1.62     1.79   123.94     0.53     0.74    68.56     6.03     1.79
##               Min     0.00     1.00     4.00    71.10     2.76     3.00    52.00    10.40    14.50
##                Q1     0.00     2.00     4.00   120.65     3.08     3.00    96.00    15.35    16.88
##            Median     0.00     2.00     6.00   196.30     3.70     4.00   123.00    19.20    17.71
##                Q3     1.00     4.00     8.00   334.00     3.92     4.00   180.00    22.80    18.90
##               Max     1.00     8.00     8.00   472.00     4.93     5.00   335.00    33.90    22.90
##               MAD     0.00     1.48     2.97   140.48     0.70     1.48    77.10     5.41     1.42
##               IQR     1.00     2.00     4.00   205.18     0.84     1.00    83.50     7.38     2.01
##                CV     1.23     0.57     0.29     0.54     0.15     0.20     0.47     0.30     0.10
##          Skewness     0.36     1.05    -0.17     0.38     0.27     0.53     0.73     0.61     0.37
##       SE.Skewness     0.41     0.41     0.41     0.41     0.41     0.41     0.41     0.41     0.41
##          Kurtosis    -1.92     1.26    -1.76    -1.21    -0.71    -1.07    -0.14    -0.37     0.34
##           N.Valid    32.00    32.00    32.00    32.00    32.00    32.00    32.00    32.00    32.00
##         Pct.Valid   100.00   100.00   100.00   100.00   100.00   100.00   100.00   100.00   100.00
##
## Table: Table continues below
##
##
##
##                       vs       wt
## ----------------- -------- --------
##              Mean     0.44     3.22
##           Std.Dev     0.50     0.98
##               Min     0.00     1.51
##                Q1     0.00     2.54
##            Median     0.00     3.33
##                Q3     1.00     3.65
##               Max     1.00     5.42
##               MAD     0.00     0.77
##               IQR     1.00     1.03
##                CV     1.15     0.30
```

```
##          Skewness     0.24     0.42
##       SE.Skewness     0.41     0.41
##          Kurtosis    -2.00    -0.02
##           N.Valid    32.00    32.00
##         Pct.Valid   100.00   100.00
```

```
summarytools::freq(am)
```

```
## Frequencies
## am
## Type: Numeric
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##           0     19     59.38          59.38     59.38          59.38
##           1     13     40.62         100.00     40.62         100.00
##        <NA>      0                               0.00         100.00
##       Total     32    100.00         100.00    100.00         100.00
```

```
summarytools::ctable(am,vs)
```

```
## Cross-Tabulation, Row Proportions
## am * vs
##
## ------- ---- ------------ ------------ -------------
##           vs            0            1         Total
##      am
##       0        12 (63.2%)    7 (36.8%)   19 (100.0%)
##       1         6 (46.2%)    7 (53.8%)   13 (100.0%)
##   Total        18 (56.2%)   14 (43.8%)   32 (100.0%)
## ------- ---- ------------ ------------ -------------
```

```
summarytools::dfSummary(mtcars)
```

```
## Data Frame Summary
## mtcars
## Dimensions: 32 x 11
## Duplicates: 0
##
## ---------------------------------------------------------------------------------------------------------------
## No   Variable    Stats / Values               Freqs (% of Valid)   Graph                     Valid
## ---- ----------- ---------------------------- -------------------- ------------------------- --------
## 1    mpg         Mean (sd) : 20.1 (6)         25 distinct values     :                       32
##      [numeric]   min < med < max:                                    : .                     (100%)
##                  10.4 < 19.2 < 33.9                               . : :
##                  IQR (CV) : 7.4 (0.3)                             : : :   .
##                                                                   : : : : :
##
## 2    cyl         Mean (sd) : 6.2 (1.8)        4 : 11 (34.4%)       IIIIII                  32
##      [numeric]   min < med < max:             6 :  7 (21.9%)       IIII                    (100%)
##                  4 < 6 < 8                    8 : 14 (43.8%)       IIIIIIII
##                  IQR (CV) : 4 (0.3)
##
## 3    disp        Mean (sd) : 230.7 (123.9)    27 distinct values     :                     32
##      [numeric]   min < med < max:                                    . :                   (100%)
##                  71.1 < 196.3 < 472                               : : :   : : :
```

```
##                 IQR (CV) : 205.2 (0.5)                          : : :   : : :   .
##                                                                 : : : . : : : : . :
##
## 4    hp         Mean (sd) : 146.7 (68.6)    22 distinct values   . :                    32
##      [numeric]  min < med < max:                                 : :                    (100%)
##                 52 < 123 < 335                                  : : : .
##                 IQR (CV) : 83.5 (0.5)                           : : : :
##                                                                 : : : : . .
##
## 5    drat       Mean (sd) : 3.6 (0.5)       22 distinct values      :                   32
##      [numeric]  min < med < max:                                  : :                   (100%)
##                 2.8 < 3.7 < 4.9                                  : : .
##                 IQR (CV) : 0.8 (0.1)                          . : : :
##                                                               : : : : .
##
## 6    wt         Mean (sd) : 3.2 (1)         29 distinct values        :                 32
##      [numeric]  min < med < max:                                     : :                (100%)
##                 1.5 < 3.3 < 5.4                                     : :
##                 IQR (CV) : 1 (0.3)                            : : : : :     .
##                                                               : : : : : : .   :
##
## 7    qsec       Mean (sd) : 17.8 (1.8)      30 distinct values        :                 32
##      [numeric]  min < med < max:                                     :                 (100%)
##                 14.5 < 17.7 < 22.9                                  : :
##                 IQR (CV) : 2 (0.1)                              . : : : :
##                                                               : : : : : : :   .
##
## 8    vs         Min  : 0                    0 : 18 (56.2%)      IIIIIIIIIII           32
##      [numeric]  Mean : 0.4                  1 : 14 (43.8%)      IIIIIIII              (100%)
##                 Max  : 1
##
## 9    am         Min  : 0                    0 : 19 (59.4%)      IIIIIIIIIII           32
##      [numeric]  Mean : 0.4                  1 : 13 (40.6%)      IIIIIIII              (100%)
##                 Max  : 1
##
## 10   gear       Mean (sd) : 3.7 (0.7)       3 : 15 (46.9%)      IIIIIIIII             32
##      [numeric]  min < med < max:            4 : 12 (37.5%)      IIIIIII               (100%)
##                 3 < 4 < 5                   5 :  5 (15.6%)      III
##                 IQR (CV) : 1 (0.2)
##
## 11   carb       Mean (sd) : 2.8 (1.6)       1 :  7 (21.9%)      IIII                  32
##      [numeric]  min < med < max:            2 : 10 (31.2%)      IIIIII                (100%)
##                 1 < 2 < 8                   3 :  3 ( 9.4%)      I
##                 IQR (CV) : 2 (0.6)          4 : 10 (31.2%)      IIIIII
##                                             6 :  1 ( 3.1%)
##                                             8 :  1 ( 3.1%)
## ----------------------------------------------------------------------------------------------------
```

Graphical representation of data

Using a new data set called “Orange”

```
Orange
```

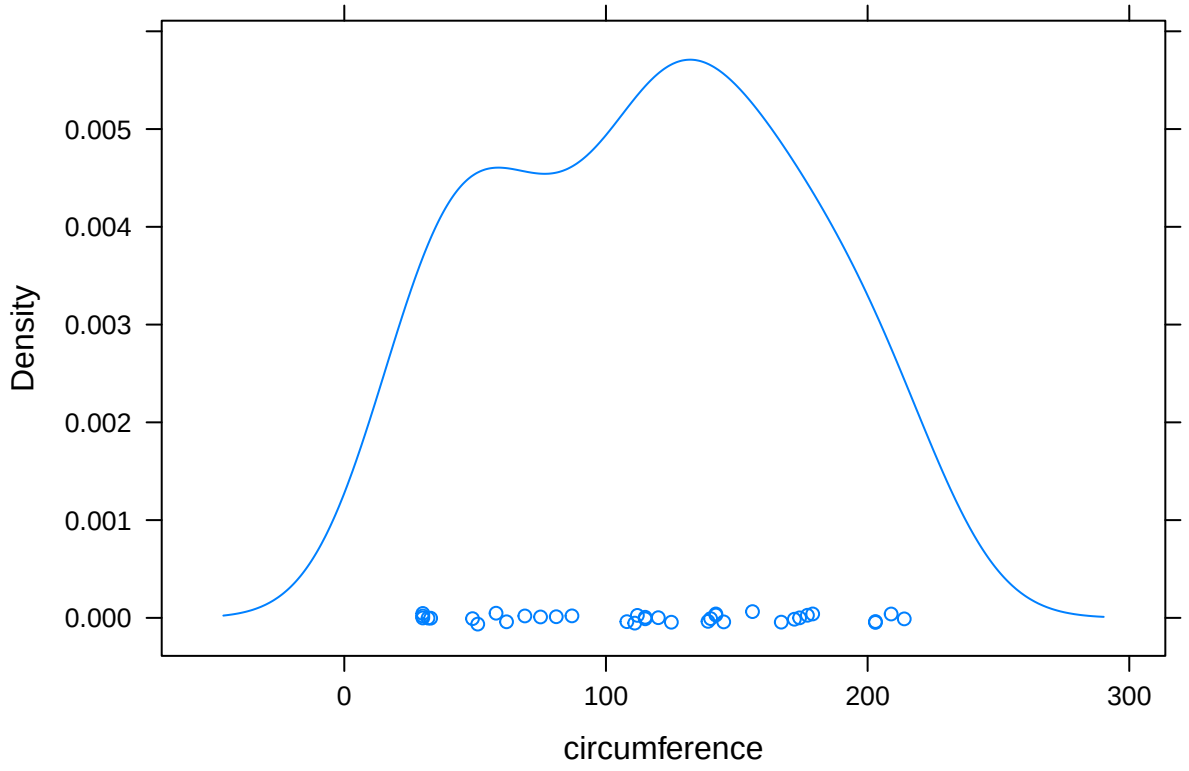
```
## Grouped Data: circumference ~ age | Tree
##    Tree  age circumference
```

```
## 1      1  118             30
## 2      1  484             58
## 3      1  664             87
## 4      1 1004            115
## 5      1 1231            120
## 6      1 1372            142
## 7      1 1582            145
## 8      2  118             33
## 9      2  484             69
## 10     2  664            111
## 11     2 1004            156
## 12     2 1231            172
## 13     2 1372            203
## 14     2 1582            203
## 15     3  118             30
## 16     3  484             51
## 17     3  664             75
## 18     3 1004            108
## 19     3 1231            115
## 20     3 1372            139
## 21     3 1582            140
## 22     4  118             32
## 23     4  484             62
## 24     4  664            112
## 25     4 1004            167
## 26     4 1231            179
## 27     4 1372            209
## 28     4 1582            214
## 29     5  118             30
## 30     5  484             49
## 31     5  664             81
## 32     5 1004            125
## 33     5 1231            142
## 34     5 1372            174
## 35     5 1582            177
```

```
attach(Orange)
```

Density plot of circumference

```
densityplot(~circumference)
```

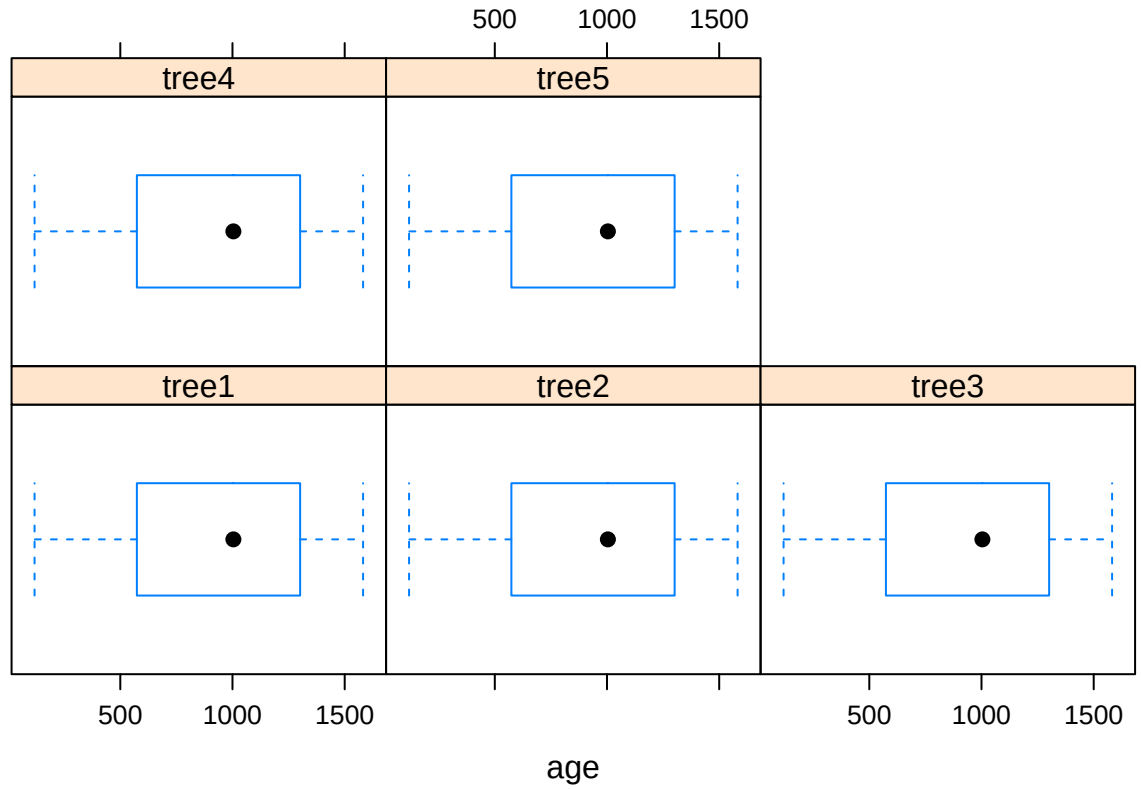


Converting into categorical/factor variable

```
tree.f = factor(Tree, levels = c(1,2,3,4,5), labels = c("tree1","tree2","tree3","tree4","tree5"))
```

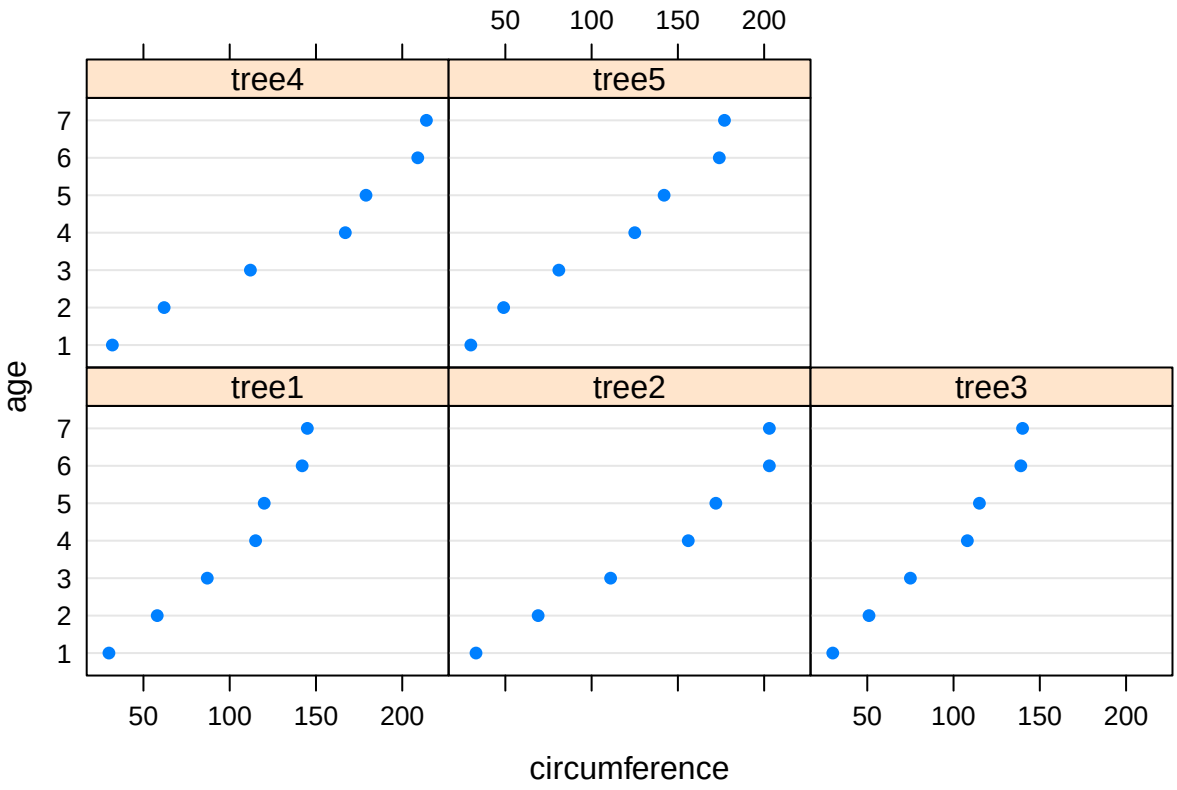
Boxplot of age of trees

```
bwplot(~age|tree.f)
```



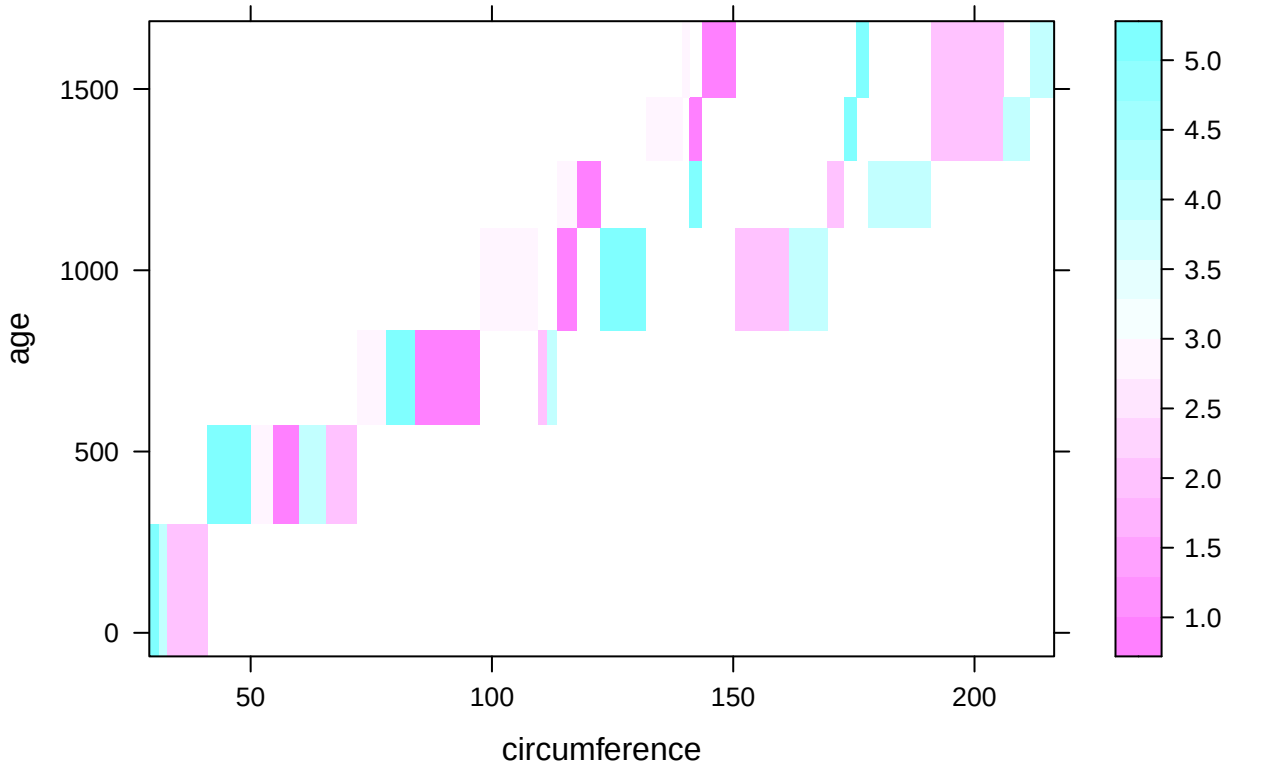
Dot plot

```
dotplot(age~circumference|tree.f)
```



Level plot

```
levelplot(tree.f~circumference*age)
```

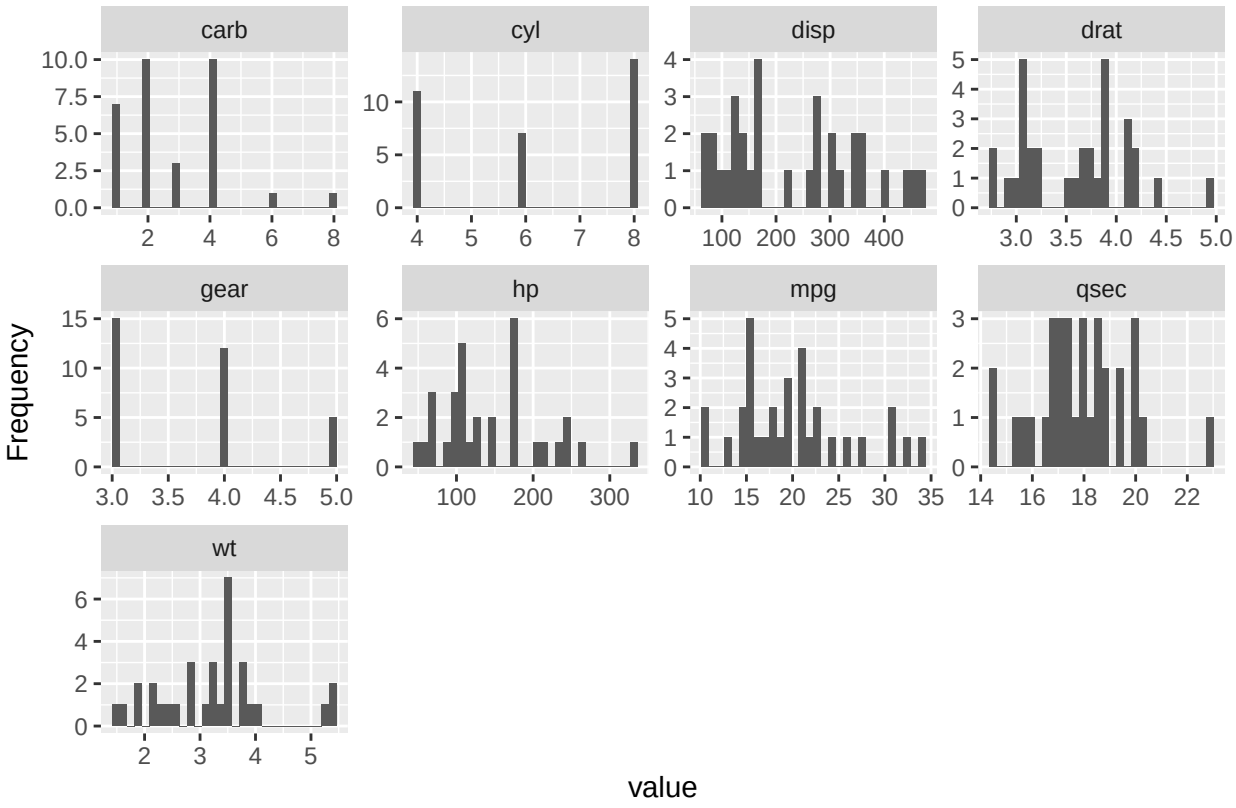



```
install.packages("DataExplorer", dependencies = T,repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yp/0237rgk11t35swrh_2f9h_200000gn/T//RtmpedLe6o/downloaded_packages
```

```
library(DataExplorer)

plot_histogram(mtcars)
```



Not even interested to write a single line of command, this is very sexy and appealing for data cleaning

```
install.packages("esquisse", dependencies = T,repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yp/0237rgk11t35swrh_2f9h_200000gn/T//RtmpedLe6o/downloaded_packages
```

```
library(esquisse)

esquisse::esquisser(mtcars)
```

Even new packages click and play, contingency tables, summary stats

```
install.packages("Rcmdr", dependencies = T,repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yp/0237rgk11t35swrh_2f9h_200000gn/T//RtmpedLe6o/downloaded_packages
```

```
library(Rcmdr)
```

Lets use our own dataset Employee dataset but for now use mtcars

```
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
```

```
attach(mtcars)
```

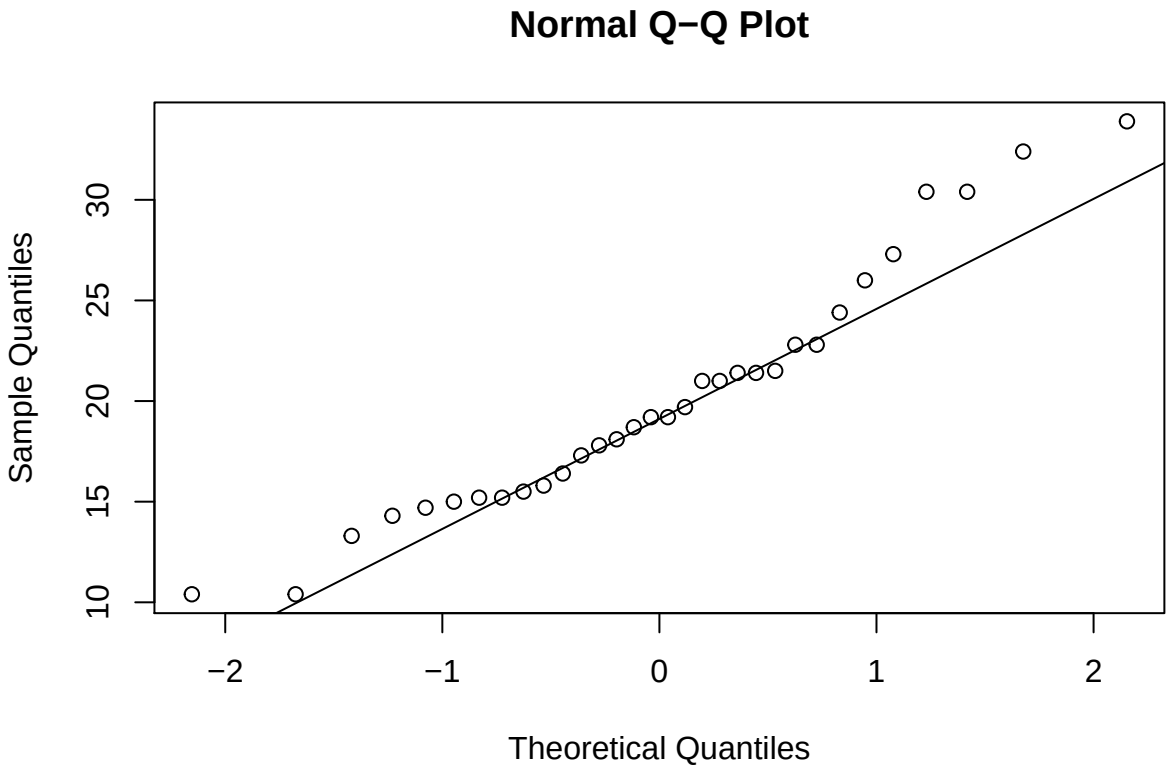
```
## The following objects are masked from mtcars (pos = 14):
##
##     am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt

## The following objects are masked from mtcars (pos = 18):
##
##     am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt

## The following object is masked from package:ggplot2:
##
##     mpg
```

Normality Checks using graphics but graphics is not 100% so we use test for rejection of h0: normal distribution and h1: not normal ; hence if p value < 0.05 then the data is not normal

```
qqnorm(mpg)
qqline(mpg)
```




```
shapiro.test(mpg)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mpg
## W = 0.94756, p-value = 0.1229
```

tm, quanteda: for unstructured data

tseries: for timeseries

animate: can be used to animate any plot type, written by Yihui Xie

gganimate: used to specifically animate ggplot graphics, written by Thomas Lin Pedersen

plotly: an interactive plotting library which has animation features

googlevis: has a flash based motion chart option

plspm for SEM