

14th July 2020

```
> # Power of a number
> a=8
> print(a**2)  # 8*8-->64
[1] 64
> print(a^2)   # 8*8-->64
[1] 64
>
> print(c(1,2,3,4,5)^2)      # All the values are powered in terms of 2
[1] 1 4 9 16 25
> print(c(1,2,3,4)*c(3,4))  # 3 8 9 16
[1] 3 8 9 16
> print(c(2,4,6,8)*c(-2,-4,-6,-8)) # The values are multiplied element-wise -4 -16 -36 -64
[1] -4 -16 -36 -64
> print(c(1,2,3,4)+5)        # All the values are added with 5
[1] 6 7 8 9
>
> # Integer division (quotient)
> print(c(2,4,6,8)%/c(2,3)) # 1 1 3 2
[1] 1 1 3 2
>
> # Modulo division (remainder)
> print(c(2,4,6,8)%%c(2,3)) # 0 1 0 2
[1] 0 1 0 2
> # Maximum and minimum function
> print(max(c(2,4,5,1))) # 5
[1] 5
> print(min(c(2,4,5,1))) # 1
[1] 1
>
> # abs(), round(), sqrt(), sum(), prod()
> print(abs(-2))           # 2
[1] 2
> print(round(12.78))      # 13
[1] 13
> print(sqrt(c(2,4,6,8))) # 1.414214 2.000000 2.449490 2.828427
[1] 1.414214 2.000000 2.449490 2.828427
> print(sum(c(2,4,6,8)))  # 20
[1] 20
> print(prod(c(2,4,6,8))) # 384
[1] 384
```

```

> ##### Logarithmic function #####
> # Natural log (ln --> log to the base e)
> print(log(5))    # 1.609438
[1] 1.609438
>
> # Common log (log --> log to the base 10)
> print(log10(5))      # 0.69897
[1] 0.69897
> print(log(5,base=10)) # 0.69897
[1] 0.69897
>
> # log(number,base=<number>) --> We can find the log of any number with any base
> print(log(9,base=4))  # 1.584963
[1] 1.584963
'
> ##### Complex functions #####
> a = 3+5i
> print(Re(a))      # real part of a --> 3
[1] 3
> print(Im(a))      # imaginary part of a --> 5
[1] 5
> print(Conj(a))    # conjugate of a --> 3-5i
[1] 3-5i
> print(Mod(a))     # modulus of a --> 5.830952
[1] 5.830952
> print(Arg(a))     # argument of a --> 1.030377
[1] 1.030377

> ##### Matrix #####
> x = matrix(nrow=3,ncol=3,data=c(2,4,6,3,6,9,5,10,15))  # Creating a matrix
> y = matrix(nrow=3,ncol=3,data=c(2,4,6,3,6,9,5,10,15),byrow=TRUE)  # Creating a matrix row wise
> t = matrix(nrow=2,ncol=3,data=100) ## Creating a matrix of single data
> d = diag(1,nrow=2,ncol=2)          ## Creating a diagonal matrix
> x[2,3] # Accessing the matrix elements
[1] 10
>
> ## Properties of a matrix ##
> print(dim(x))
[1] 3 3
> print(attributes(x))
$dim
[1] 3 3

```

```

> print(nrow(x))
[1] 3
> print(ncol(x))
[1] 3
> print(mode(x)) # types of storage
[1] "numeric"
> print(t(x)) # diagonal of a matrix
[,1] [,2] [,3]
[1,]    2    4    6
[2,]    3    6    9
[3,]    5   10   15
[4,]   15   30   45
[5,]   20   40   60
[6,]   25   50   75
[7,]   30   60   90
[8,]   35   70   105
[9,]   40   80   120
[10,]  45   90   135
[11,]  50   100  150
[12,]  55   110  165
[13,]  60   120  180
[14,]  65   130  195
[15,]  70   140  210
[16,]  75   150  225
[17,]  80   160  240
[18,]  85   170  255
[19,]  90   180  270
[20,]  95   190  285
[21,] 100   200  300
[22,] 105   210  315
[23,] 110   220  330
[24,] 115   230  345
[25,] 120   240  360
[26,] 125   250  375
[27,] 130   260  390
[28,] 135   270  405
[29,] 140   280  420
[30,] 145   290  435
[31,] 150   300  450
[32,] 155   310  465
[33,] 160   320  480
[34,] 165   330  495
[35,] 170   340  510
[36,] 175   350  525
[37,] 180   360  540
[38,] 185   370  555
[39,] 190   380  570
[40,] 195   390  585
[41,] 200   400  600
[42,] 205   410  615
[43,] 210   420  630
[44,] 215   430  645
[45,] 220   440  660
[46,] 225   450  675
[47,] 230   460  690
[48,] 235   470  705
[49,] 240   480  720
[50,] 245   490  735
[51,] 250   500  750
[52,] 255   510  765
[53,] 260   520  780
[54,] 265   530  795
[55,] 270   540  810
[56,] 275   550  825
[57,] 280   560  840
[58,] 285   570  855
[59,] 290   580  870
[60,] 295   590  885
[61,] 300   600  900
[62,] 305   610  915
[63,] 310   620  930
[64,] 315   630  945
[65,] 320   640  960
[66,] 325   650  975
[67,] 330   660  990
[68,] 335   670  1005
[69,] 340   680  1020
[70,] 345   690  1035
[71,] 350   700  1050
[72,] 355   710  1065
[73,] 360   720  1080
[74,] 365   730  1095
[75,] 370   740  1110
[76,] 375   750  1125
[77,] 380   760  1140
[78,] 385   770  1155
[79,] 390   780  1170
[80,] 395   790  1185
[81,] 400   800  1200
[82,] 405   810  1215
[83,] 410   820  1230
[84,] 415   830  1245
[85,] 420   840  1260
[86,] 425   850  1275
[87,] 430   860  1290
[88,] 435   870  1305
[89,] 440   880  1320
[90,] 445   890  1335
[91,] 450   900  1350
[92,] 455   910  1365
[93,] 460   920  1380
[94,] 465   930  1395
[95,] 470   940  1410
[96,] 475   950  1425
[97,] 480   960  1440
[98,] 485   970  1455
[99,] 490   980  1470
[100,] 495   990  1485
[101,] 500   1000 1500
[102,] 505   1010 1515
[103,] 510   1020 1530
[104,] 515   1030 1545
[105,] 520   1040 1560
[106,] 525   1050 1575
[107,] 530   1060 1590
[108,] 535   1070 1605
[109,] 540   1080 1620
[110,] 545   1090 1635
[111,] 550   1100 1650
[112,] 555   1110 1665
[113,] 560   1120 1680
[114,] 565   1130 1695
[115,] 570   1140 1710
[116,] 575   1150 1725
[117,] 580   1160 1740
[118,] 585   1170 1755
[119,] 590   1180 1770
[120,] 595   1190 1785
[121,] 600   1200 1800
[122,] 605   1210 1815
[123,] 610   1220 1830
[124,] 615   1230 1845
[125,] 620   1240 1860
[126,] 625   1250 1875
[127,] 630   1260 1890
[128,] 635   1270 1905
[129,] 640   1280 1920
[130,] 645   1290 1935
[131,] 650   1300 1950
[132,] 655   1310 1965
[133,] 660   1320 1980
[134,] 665   1330 1995
[135,] 670   1340 2010
[136,] 675   1350 2025
[137,] 680   1360 2040
[138,] 685   1370 2055
[139,] 690   1380 2070
[140,] 695   1390 2085
[141,] 700   1400 2100
[142,] 705   1410 2115
[143,] 710   1420 2130
[144,] 715   1430 2145
[145,] 720   1440 2160
[146,] 725   1450 2175
[147,] 730   1460 2190
[148,] 735   1470 2205
[149,] 740   1480 2220
[150,] 745   1490 2235
[151,] 750   1500 2250
[152,] 755   1510 2265
[153,] 760   1520 2280
[154,] 765   1530 2295
[155,] 770   1540 2310
[156,] 775   1550 2325
[157,] 780   1560 2340
[158,] 785   1570 2355
[159,] 790   1580 2370
[160,] 795   1590 2385
[161,] 800   1600 2400
[162,] 805   1610 2415
[163,] 810   1620 2430
[164,] 815   1630 2445
[165,] 820   1640 2460
[166,] 825   1650 2475
[167,] 830   1660 2490
[168,] 835   1670 2505
[169,] 840   1680 2520
[170,] 845   1690 2535
[171,] 850   1700 2550
[172,] 855   1710 2565
[173,] 860   1720 2580
[174,] 865   1730 2595
[175,] 870   1740 2610
[176,] 875   1750 2625
[177,] 880   1760 2640
[178,] 885   1770 2655
[179,] 890   1780 2670
[180,] 895   1790 2685
[181,] 900   1800 2700
[182,] 905   1810 2715
[183,] 910   1820 2730
[184,] 915   1830 2745
[185,] 920   1840 2760
[186,] 925   1850 2775
[187,] 930   1860 2790
[188,] 935   1870 2805
[189,] 940   1880 2820
[190,] 945   1890 2835
[191,] 950   1900 2850
[192,] 955   1910 2865
[193,] 960   1920 2880
[194,] 965   1930 2895
[195,] 970   1940 2910
[196,] 975   1950 2925
[197,] 980   1960 2940
[198,] 985   1970 2955
[199,] 990   1980 2970
[200,] 995   1990 2985
[201,] 1000  2000 3000
[202,] 1005  2010 3015
[203,] 1010  2020 3030
[204,] 1015  2030 3045
[205,] 1020  2040 3060
[206,] 1025  2050 3075
[207,] 1030  2060 3090
[208,] 1035  2070 3105
[209,] 1040  2080 3120
[210,] 1045  2090 3135
[211,] 1050  2100 3150
[212,] 1055  2110 3165
[213,] 1060  2120 3180
[214,] 1065  2130 3195
[215,] 1070  2140 3210
[216,] 1075  2150 3225
[217,] 1080  2160 3240
[218,] 1085  2170 3255
[219,] 1090  2180 3270
[220,] 1095  2190 3285
[221,] 1100  2200 3300
[222,] 1105  2210 3315
[223,] 1110  2220 3330
[224,] 1115  2230 3345
[225,] 1120  2240 3360
[226,] 1125  2250 3375
[227,] 1130  2260 3390
[228,] 1135  2270 3405
[229,] 1140  2280 3420
[230,] 1145  2290 3435
[231,] 1150  2300 3450
[232,] 1155  2310 3465
[233,] 1160  2320 3480
[234,] 1165  2330 3495
[235,] 1170  2340 3510
[236,] 1175  2350 3525
[237,] 1180  2360 3540
[238,] 1185  2370 3555
[239,] 1190  2380 3570
[240,] 1195  2390 3585
[241,] 1200  2400 3600
[242,] 1205  2410 3615
[243,] 1210  2420 3630
[244,] 1215  2430 3645
[245,] 1220  2440 3660
[246,] 1225  2450 3675
[247,] 1230  2460 3690
[248,] 1235  2470 3705
[249,] 1240  2480 3720
[250,] 1245  2490 3735
[251,] 1250  2500 3750
[252,] 1255  2510 3765
[253,] 1260  2520 3780
[254,] 1265  2530 3795
[255,] 1270  2540 3810
[256,] 1275  2550 3825
[257,] 1280  2560 3840
[258,] 1285  2570 3855
[259,] 1290  2580 3870
[260,] 1295  2590 3885
[261,] 1300  2600 3900
[262,] 1305  2610 3915
[263,] 1310  2620 3930
[264,] 1315  2630 3945
[265,] 1320  2640 3960
[266,] 1325  2650 3975
[267,] 1330  2660 3990
[268,] 1335  2670 4005
[269,] 1340  2680 4020
[270,] 1345  2690 4035
[271,] 1350  2700 4050
[272,] 1355  2710 4065
[273,] 1360  2720 4080
[274,] 1365  2730 4095
[275,] 1370  2740 4110
[276,] 1375  2750 4125
[277,] 1380  2760 4140
[278,] 1385  2770 4155
[279,] 1390  2780 4170
[280,] 1395  2790 4185
[281,] 1400  2800 4200
[282,] 1405  2810 4215
[283,] 1410  2820 4230
[284,] 1415  2830 4245
[285,] 1420  2840 4260
[286,] 1425  2850 4275
[287,] 1430  2860 4290
[288,] 1435  2870 4305
[289,] 1440  2880 4320
[290,] 1445  2890 4335
[291,] 1450  2900 4350
[292,] 1455  2910 4365
[293,] 1460  2920 4380
[294,] 1465  2930 4395
[295,] 1470  2940 4410
[296,] 1475  2950 4425
[297,] 1480  2960 4440
[298,] 1485  2970 4455
[299,] 1490  2980 4470
[300,] 1495  2990 4485
[301,] 1500  3000 4500
[302,] 1505  3010 4515
[303,] 1510  3020 4530
[304,] 1515  3030 4545
[305,] 1520  3040 4560
[306,] 1525  3050 4575
[307,] 1530  3060 4590
[308,] 1535  3070 4605
[309,] 1540  3080 4620
[310,] 1545  3090 4635
[311,] 1550  3100 4650
[312,] 1555  3110 4665
[313,] 1560  3120 4680
[314,] 1565  3130 4695
[315,] 1570  3140 4710
[316,] 1575  3150 4725
[317,] 1580  3160 4740
[318,] 1585  3170 4755
[319,] 1590  3180 4770
[320,] 1595  3190 4785
[321,] 1600  3200 4800
[322,] 1605  3210 4815
[323,] 1610  3220 4830
[324,] 1615  3230 4845
[325,] 1620  3240 4860
[326,] 1625  3250 4875
[327,] 1630  3260 4890
[328,] 1635  3270 4905
[329,] 1640  3280 4920
[330,] 1645  3290 4935
[331,] 1650  3300 4950
[332,] 1655  3310 4965
[333,] 1660  3320 4980
[334,] 1665  3330 4995
[335,] 1670  3340 5010
[336,] 1675  3350 5025
[337,] 1680  3360 5040
[338,] 1685  3370 5055
[339,] 1690  3380 5070
[340,] 1695  3390 5085
[341,] 1700  3400 5100
[342,] 1705  3410 5115
[343,] 1710  3420 5130
[344,] 1715  3430 5145
[345,] 1720  3440 5160
[346,] 1725  3450 5175
[347,] 1730  3460 5190
[348,] 1735  3470 5205
[349,] 1740  3480 5220
[350,] 1745  3490 5235
[351,] 1750  3500 5250
[352,] 1755  3510 5265
[353,] 1760  3520 5280
[354,] 1765  3530 5295
[355,] 1770  3540 5310
[356,] 1775  3550 5325
[357,] 1780  3560 5340
[358,] 1785  3570 5355
[359,] 1790  3580 5370
[360,] 1795  3590 5385
[361,] 1800  3600 5400
[362,] 1805  3610 5415
[363,] 1810  3620 5430
[364,] 1815  3630 5445
[365,] 1820  3640 5460
[366,] 1825  3650 5475
[367,] 1830  3660 5490
[368,] 1835  3670 5505
[369,] 1840  3680 5520
[370,] 1845  3690 5535
[371,] 1850  3700 5550
[372,] 1855  3710 5565
[373,] 1860  3720 5580
[374,] 1865  3730 5595
[375,] 1870  3740 5610
[376,] 1875  3750 5625
[377,] 1880  3760 5640
[378,] 1885  3770 5655
[379,] 1890  3780 5670
[380,] 1895  3790 5685
[381,] 1900  3800 5700
[382,] 1905  3810 5715
[383,] 1910  3820 5730
[384,] 1915  3830 5745
[385,] 1920  3840 5760
[386,] 1925  3850 5775
[387,] 1930  3860 5790
[388,] 1935  3870 5805
[389,] 1940  3880 5820
[390,] 1945  3890 5835
[391,] 1950  3900 5850
[392,] 1955  3910 5865
[393,] 1960  3920 5880
[394,] 1965  3930 5895
[395,] 1970  3940 5910
[396,] 1975  3950 5925
[397,] 1980  3960 5940
[398,] 1985  3970 5955
[399,] 1990  3980 5970
[400,] 1995  3990 5985
[401,] 2000  4000 6000
[402,] 2005  4010 6015
[403,] 2010  4020 6030
[404,] 2015  4030 6045
[405,] 2020  4040 6060
[406,] 2025  4050 6075
[407,] 2030  4060 6090
[408,] 2035  4070 6105
[409,] 2040  4080 6120
[410,] 2045  4090 6135
[411,] 2050  4100 6150
[412,] 2055  4110 6165
[413,] 2060  4120 6180
[414,] 2065  4130 6195
[415,] 2070  4140 6210
[416,] 2075  4150 6225
[417,] 2080  4160 6240
[418,] 2085  4170 6255
[419,] 2090  4180 6270
[420,] 2095  4190 6285
[421,] 2100  4200 6300
[422,] 2105  4210 6315
[423,] 2110  4220 6330
[424,] 2115  4230 6345
[425,] 2120  4240 6360
[426,] 2125  4250 6375
[427,] 2130  4260 6390
[428,] 2135  4270 6405
[429,] 2140  4280 6420
[430,] 2145  4290 6435
[431,] 2150  4300 6450
[432,] 2155  4310 6465
[433,] 2160  4320 6480
[434,] 2165  4330 6495
[435,] 2170  4340 6510
[436,] 2175  4350 6525
[437,] 2180  4360 6540
[438,] 2185  4370 6555
[439,] 2190  4380 6570
[440,] 2195  4390 6585
[441,] 2200  4400 6600
[442,] 2205  4410 6615
[443,] 2210  4420 6630
[444,] 2215  4430 6645
[445,] 2220  4440 6660
[446,] 2225  4450 6675
[447,] 2230  4460 6690
[448,] 2235  4470 6705
[449,] 2240  4480 6720
[450,] 2245  4490 6735
[451,] 2250  4500 6750
[452,] 2255  4510 6765
[453,] 2260  4520 6780
[454,] 2265  4530 6795
[455,] 2270  4540 6810
[456,] 2275  4550 6825
[457,] 2280  4560 6840
[458,] 2285  4570 6855
[459,] 2290  4580 6870
[460,] 2295  4590 6885
[461,] 2300  4600 6900
[462,] 2305  4610 6915
[463,] 2310  4620 6930
[464,] 2315  4630 6945
[465,] 2320  4640 6960
[466,] 2325  4650 6975
[467,] 2330  4660 6990
[468,] 2335  4670 7005
[469,] 2340  4680 7020
[470,] 2345  4690 7035
[471,] 2350  4700 7050
[472,] 2355  4710 7065
[473,] 2360  4720 7080
[474,] 2365  4730 7095
[475,] 2370  4740 7110
[476,] 2375  4750 7125
[477,] 2380  4760 7140
[478,] 2385  4770 7155
[479,] 2390  4780 7170
[480,] 2395  4790 7185
[481,] 2400  4800 7200
[482,] 2405  4810 7215
[483,] 2410  4820 7230
[484,] 2415  4830 7245
[485,] 2420  4840 7260
[486,] 2425  4850 7275
[487,] 2430  4860 7290
[488,] 2435  4870 7305
[489,] 2440  4880 7320
[490,] 2445  4890 7335
[491,] 2450  4900 7350
[492,] 2455  4910 7365
[493,] 2460  4920 7380
[494,] 2465  4930 7395
[495,] 2470  4940 7410
[496,] 2475  4950 7425
[497,] 2480  4960 7440
[498,] 2485  4970 7455
[499,] 2490  4980 7470
[500,] 2495  
```

(25th July 2020)

```
1 empid = c(1,2,3,4,5,6,7,8,9,10) # vector empid
2 age = c(30,37,45,32,50,31,53,21,54,42) # vector age
3 gender = c(0,1,1,0,1,0,0,1,0,1)
4 status = c(1,2,1,1,2,2,1,2,1,2)
5
6 # data.frame
7 empinfo = data.frame(empid,age,gender,status)
8
9 # factor() --> will convert the numbers provided into certain specific labels
10 empinfo$gender = factor(empinfo$gender,labels=c("male","female")) # 0-->male and 1-->female
11 empinfo$status = factor(empinfo$status,labels=c("staff","faculty")) # 1-->staff and 2-->faculty
12
13
14 # subset() --> will take a subset of data from the main frame with specific condition
15 genderm = subset(empinfo,empinfo$gender=='male') # Taking the rows whose gender is male
16 genderf = subset(empinfo,empinfo$gender=='female') # Taking the rows whose gender is female
17
18 # summary() --> summary statiscs of empinfo
19 summary(empinfo)
20 summary(genderm) # Summary statiscs of male employ
21 summary(genderf) # Summary statiscs of female employ
22 summary(empinfo$age) # Summary statiscs of age
23
24 # table()
25 # One way table
26 table1 = table(empinfo$gender) # In gender, how many male and female are present???
27 table2 = table(empinfo$status) # In status, how many staff's and faculty's are present???
28 # Two way table
29 table3 = table(empinfo$gender,empinfo$status) # gender --> row and status --> column
30
31 ## Graphical Representations in R
32 plot(empinfo$age,type="l",main="Age of employ",
      xlab="emp id",ylab="age in years",col="red") # main --> heading
33 # type = 'l' --> type of the line
34 # xlab --> x-label and ylab --> Y-label
35 # col --> color of the line
36 # Pie-chart for the table
37 table4 = table(empinfo$gender)
38 pie(table4)
39
40 ### barplot
41 barplot(table5,beside=T,xlim=c(1,15),ylim=c(0,5),
        col=c("blue","red"),legend=rownames(table5))
42
43
44 ### boxplot
45 boxplot(empinfo$age~empinfo$status,col=c("red","blue"))
46
```

```
> empinfo
  empid age gender  status
1     1  30   male   staff
2     2  37 female faculty
3     3  45 female   staff
4     4  32   male   staff
5     5  50 female faculty
6     6  31   male faculty
7     7  53   male   staff
8     8  21 female faculty
9     9  54   male   staff
10    10 42 female faculty
> empinfo$gender
 [1] male   female  female male   female male   male   female male   female
Levels: male female
> empinfo$status
 [1] staff   faculty staff   staff   faculty faculty staff   faculty staff   faculty
Levels: staff faculty
```

Select the snip mode using
New button.



In a future update, Sr
new home. Try impro
with Snip & Sketch (c
Windows logo key +

Snip & Sketch)

```

> genderm
  empid age gender status
1     1  30   male staff
4     4  32   male staff
6     6  31   male faculty
7     7  53   male staff
9     9  54   male staff
> genderf
  empid age gender status
2     2  37 female faculty
3     3  45 female staff
5     5  50 female faculty
8     8  21 female faculty
10    10 42 female faculty
> |
> summary(empinfo)
      empid           age         gender        status
Min. : 1.00   Min. :21.00   male :5   staff :5
1st Qu.: 3.25  1st Qu.:31.25  female:5  faculty:5
Median : 5.50   Median :39.50
Mean   : 5.50   Mean   :39.50
3rd Qu.: 7.75  3rd Qu.:48.75
Max.  :10.00   Max.  :54.00
> summary(genderm) # Summary statiscs of male employ
      empid           age         gender        status
Min. :1.0   Min. :30   male :5   staff :4
1st Qu.:4.0  1st Qu.:31   female:0  faculty:1
Median :6.0   Median :32
Mean   :5.4   Mean   :40
3rd Qu.:7.0  3rd Qu.:53
Max.  :9.0   Max.  :54

```

```

> summary(genderf) # Summary statiscs of female employ
    empid          age       gender      status
  Min.   : 2.0   Min.   :21   male   :0   staff  :1
  1st Qu.: 3.0   1st Qu.:37   female:5   faculty:4
  Median  : 5.0   Median  :42
  Mean   : 5.6   Mean   :39
  3rd Qu.: 8.0   3rd Qu.:45
  Max.   :10.0   Max.   :50
> summary(empinfo$age) # Summary statiscs of age
  Min. 1st Qu. Median  Mean 3rd Qu. Max.
  21.00 31.25 39.50 39.50 48.75 54.00

```

> table1

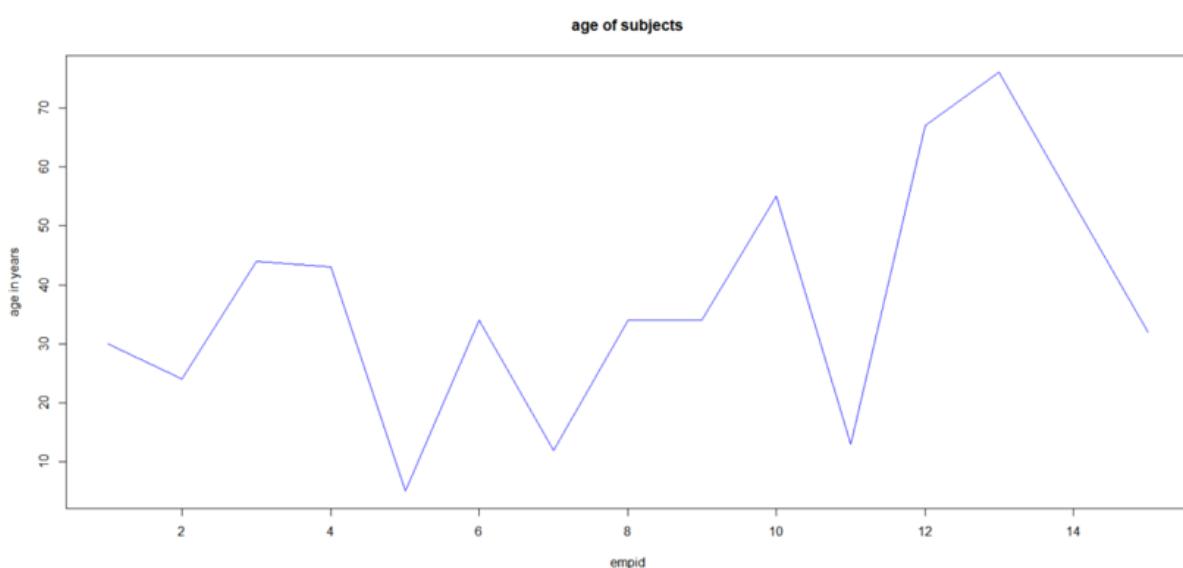
male	female
5	5

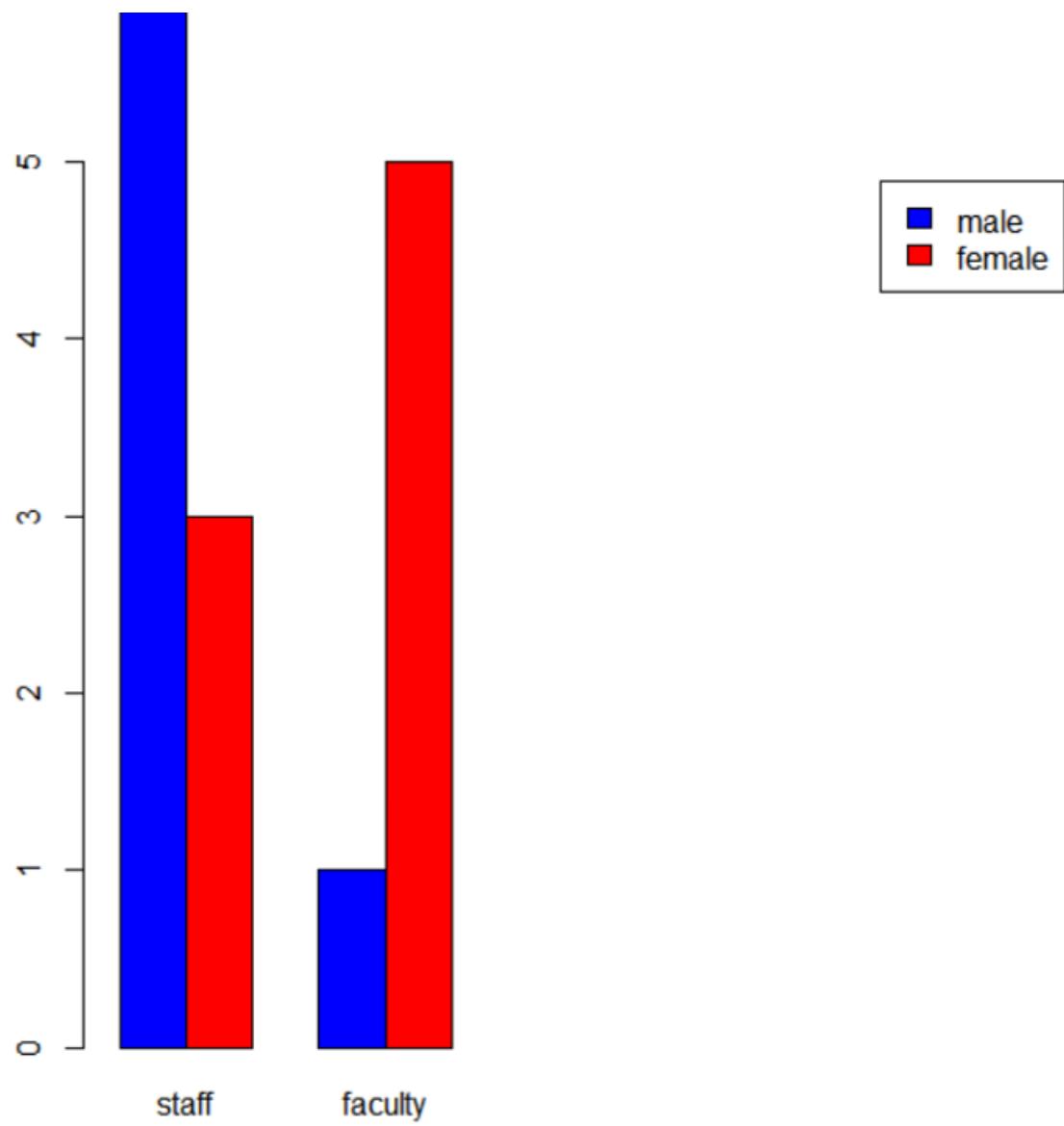
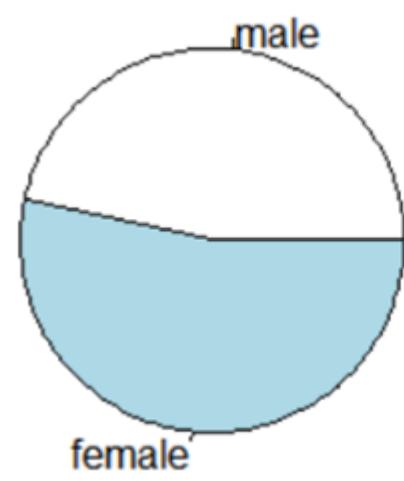
> table2

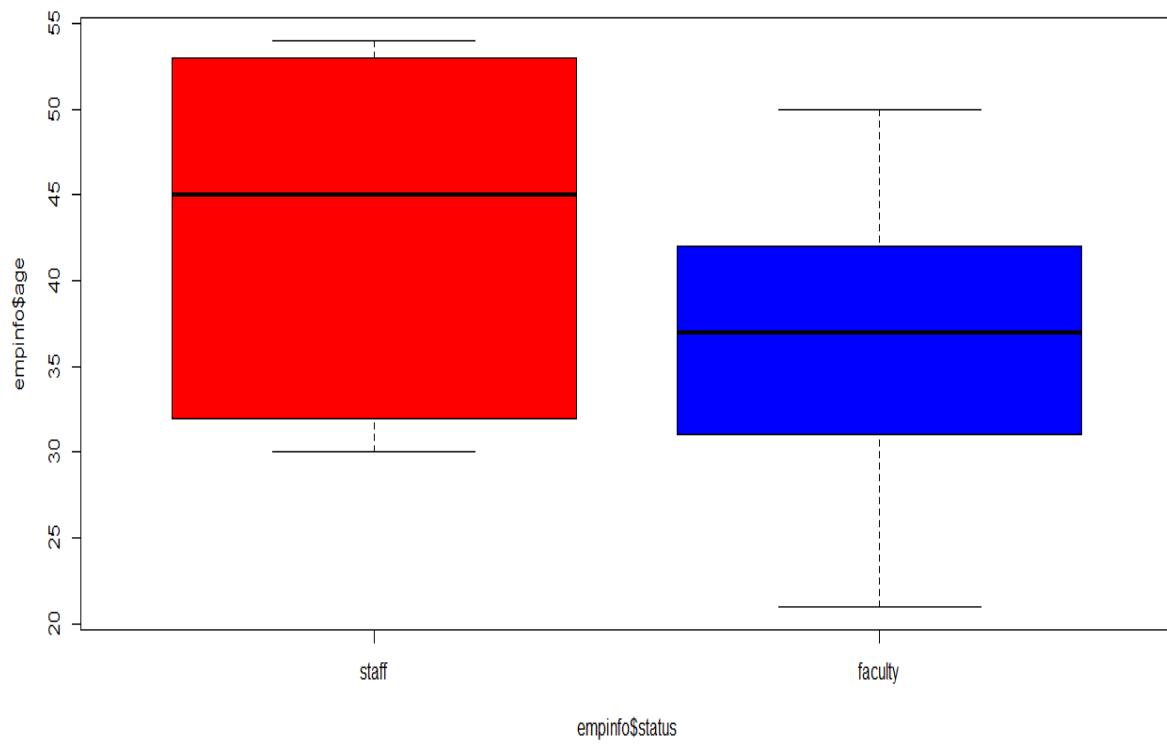
staff	faculty
5	5

> table3

	staff	faculty
male	4	1
female	1	4







(18th August 2020)

Problem1: Twenty students , graduates and undergraduates, were enrolled in a statistics course. Their ages were

18,19,19,19,19,20,20,20,20,21,21,21,22,23,24,27,30,36.

- a) Find Mean and Median of all students
- b) Find median age of all students under 25 years.
- c) Find modal age of all student

```
1 x = c(18,19,19,19,19,20,20,20,20,21,21,21,21,22,23,24,27,30,36)
2 cat('The mean of all the students : ',mean(x))
3 cat('The median of all the students : ',median(x))
4
5 under_25 = x[x<25]
6 cat('The median age of all students under 25 : ',median(under_25))
7
8 xr = table(x)
9 print(xr)
10 mode = which(xr==max(xr))
11 cat('The modal age of all the students : ',mode)
> cat('The mean of all the students : ',mean(x))
The mean of all the students : 22
> cat('The median of all the students : ',median(x))
The median of all the students : 20.5
> under_25 = x[x<25]
> cat('The median age of all students under 25 : ',median(under_25))
The median age of all students under 25 : 20
> xr = table(x)
> print(xr)
x
18 19 20 21 22 23 24 27 30 36
 1  4  5  4  1  1  1  1  1
> mode = which(xr==max(xr))
> cat('The modal age of all the students : ',mode)
The modal age of all the students : 3
> mode
20
 3
```

Problem 2 : A survey of 25 faculty members is taken in a college to study their vocational mobility. They were asked the question “In addition to your present position ,at how many educational institutions have served on the faculty?.Following is the frequency distribution of their responses .

<i>X</i>	0	1	2	3
<i>f</i>	8	11	5	1

Find mean and median of the distribution

```

14 ## 2nd problem
15 x = c(0,1,2,3)
16 f = c(8,11,5,1)
17 y = rep(x,f)      # rep(value,number_of_times)
18 print(y)
19 mean   = (sum(y)/length(y))
20 median = median(y)
21 cat('The mean of the distribution : ',mean)
22 cat('The median of the distribution : ',median)
??
> x = c(0,1,2,3)
> f = c(8,11,5,1)
> y = rep(x,f)      # rep(value,number_of_times)
> print(y)
[1] 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3
> mean   = (sum(y)/length(y))
> median = median(y)
> cat('The mean of the distribution : ',mean)
The mean of the distribution : 0.96
> cat('The median of the distribution : ',median)
The median of the distribution : 1

```

Problem 3 : Compute mean ,median , 1st Quartile, 3rd Quartile and mode of for the following frequency Distribution:

<i>Height in Cm</i>	<i>145-150</i>	<i>150-155</i>	<i>155-160</i>	<i>160-165</i>	<i>165-170</i>	<i>170-175</i>	<i>175-180</i>	<i>180-185</i>
<i>No. of Adult men</i>	4	6	28	58	64	30	5	5

```

24 ## 3rd problem (Continuous frequency distribution)
25 # Mean
26 x = seq(147.5,182.5,5)
27 freq = c(4,6,28,58,64,30,5,5)
28 mean = sum(x*f)/sum(f)
29 cat('The mean is : ',mean)
30
31 # Median
32 cf = cumsum(freq)
33 N = sum(freq)           # total frequency
34 median_class = min(which(cf>N/2))
35 c      = 5             # width of the median class
36 m      = cf[median_class-1] # cumulative frequency of the preceding median class
37 f      = freq[median_class] # frequency of the median class
38 L      = x[median_class]-c/2 # lower limit of the median class
39 median = L+((N/2)-m)/f*c
40 cat('The median is : ',median)
41
42 # 1st quartile
43 quartile_class = min(which(cf>3*N/4))
44 c    = 5
45 m    = cf[quartile_class-1]
46 f    = freq[quartile_class]
47 L    = x[quartile_class]-c/2
48 q11 = L+((N/4)-m)/f*c
49 cat('1st quartile : ',q11)
50
51 # 3rd quartile
52 quartile_class = min(which(cf>3*N/4))
53 c    = 5
54 m    = cf[quartile_class-1]
55 f    = freq[quartile_class]
56 L    = x[quartile_class]-c/2
57 q13 = L+((3*N/4)-m)/f*c
58 cat('3rd quartile : ',q13)
59
60 # Mode
61 modal_class = which(freq==max(freq))
62 f0    = freq[modal_class]   # frequency of the modal class
63 f1    = freq[modal_class-1] # frequency of the preceding modal class
64 f2    = freq[modal_class+1] # frequency of the succeeding modal class
65 L    = x[modal_class]-c/2 # lower limit of the modal class
66 mode = L+((f0-f1)/(2*f0-f1-f2))*c
67 cat('The mode is : ',mode)
--
```

```

> x = seq(147.5,182.5,5)
> freq = c(4,6,28,58,64,30,5,5)
> mean = sum(x*f)/sum(f)
> cat('The mean is : ',mean)
The mean is : 1320
> cf = cumsum(freq)
> N = sum(freq)           # total frequency
> median_class = min(which(cf>N/2))
> c      = 5             # width of the median class
> m      = cf[median_class-1] # cumulative frequency of the preceding median class
> f      = freq[median_class] # frequency of the median class
> L      = x[median_class]-c/2 # lower limit of the median class
> median = L+((N/2)-m)/f*c
> cat('The median is : ',median)
The median is : 165.3125
> quartile_class = min(which(cf>3*N/4))
> c    = 5
> m    = cf[quartile_class-1]
> f    = freq[quartile_class]
> L    = x[quartile_class]-c/2
> q13 = L+((3*N/4)-m)/f*c
> cat('1st quartile : ',q13)
1st quartile : 161.4062

> quartile_class = min(which(cf>3*N/4))
> c    = 5
> m    = cf[quartile_class-1]
> f    = freq[quartile_class]
> L    = x[quartile_class]-c/2
> q13 = L+((3*N/4)-m)/f*c
> cat('3rd quartile : ',q13)
3rd quartile : 169.2188
> modal_class = which(freq==max(freq))
> f0    = freq[modal_class] # frequency of the modal class
> f1    = freq[modal_class-1] # frequency of the preceding modal class
> f2    = freq[modal_class+1] # frequency of the succeeding modal class
> L    = x[modal_class]-c/2 # lower limit of the modal class
> mode = L+((f0-f1)/(2*f0-f1-f2))*c
> cat('The mode is : ',mode)
The mode is : 165.75

```

```

74 * ##### Range #####
75 a=c(1,2,3,4,5,5,6,7,8)
76 cat('Range of a ',range(a))
77 cat('Difference between maximum and minimum : ',diff(range(a)))
78
79 ## Using max() and min()
80 cat('Range : ',max(a)-min(a))
> a=c(1,2,3,4,5,5,6,7,8)
> cat('Range of a ',range(a))
Range of a 1 8> cat('Difference between maximum and minimum : ',diff(range(a)))
Difference between maximum and minimum : 7>
> ## Using max() and min()
> cat('Range : ',max(a)-min(a))
Range : 7

```

```
83 a = c(1,2,3,4,5,5,5,6,7,8)
84 cat('Interquantile range : ')
85 print(IQR(a))
86
87 cat('Quantile : ')
88 print(quantile(a))
89
90 cat('1st quantile')
91 print(quantile(a,0.25))
92
93 cat('2nd quantile')
94 print(quantile(a,0.75))

> a = c(1,2,3,4,5,5,5,6,7,8)
> cat('Interquantile range : ')
Interquantile range : > print(IQR(a))
[1] 2.5
> cat('Quantile : ')
Quantile : > print(quantile(a))
 0% 25% 50% 75% 100%
1.00 3.25 5.00 5.75 8.00
> cat('1st quantile')
1st quantile> print(quantile(a,0.25))
 25%
3.25
> cat('2nd quantile')
2nd quantile> print(quantile(a,0.75))
 75%
5.75
```

Example:

An entomologist studying morphological variation in species of mosquito recorded the following data on body length: 1.2,1.4,1.3,1.6,1.0,1.5,1.7,1.1,1.2,1.3. Compute all the measures of dispersion.

```
97 - ##### Measures of Dispersion #####
98 x=c(1.2,1.4,1.3,1.6,1.0,1.5,1.7,1.1,1.2,1.3)
99 cat('Range : ',range(x))
100 cat('Difference of Range : ',diff(range(x)))
101 cat('Variance : ',var(x))
102 cat('Standard Deviation : ',sd(x))
103 cat('Quantile ')
104 print(quantile(x))
105 cat('Inter quantile Range : ',IQR(x))
106
107 - ### Raw data #####
108 y = abs(x-mean(x))
109 cat('Mean Deviation about Mean : ',sum(y)/length(y))
110
111 y = abs(x-median(x))
112 cat('Mean Deviation about Median : ',sum(y)/length(y))

> x=c(1.2,1.4,1.3,1.6,1.0,1.5,1.7,1.1,1.2,1.3)
> cat('Range : ',range(x))
Range : 1 1.7
> cat('Difference of Range : ',diff(range(x)))
Difference of Range : 0.7
> cat('Variance : ',var(x))
Variance : 0.049
> cat('Standard Deviation : ',sd(x))
Standard Deviation : 0.2213594
> cat('Quantile ')
Quantile > print(quantile(x))
 0% 25% 50% 75% 100%
1.000 1.200 1.300 1.475 1.700
> cat('Inter quantile Range : ',IQR(x))
Inter quantile Range : 0.275
> y = abs(x-mean(x))
> cat('Mean Deviation about Mean : ',sum(y)/length(y))
Mean Deviation about Mean : 0.176
> y = abs(x-median(x))
> cat('Mean Deviation about Median : ',sum(y)/length(y))
Mean Deviation about Median : 0.17
> y = abs(x-mode(x))
```

1st September 2020

```
1 ## Karl Pearson(Correlation coefficient)
2 x=c(78,36,98,25,75,82,90,62,65,39)
3 y=c(84,51,91,60,68,62,86,58,53,47)
4
5 rxy = var(x,y)/(sqrt(var(x))*sqrt(var(y)))
6 cat('Karl Pearson coefficient : ',rxy)
7
8 ## Direction function
9 cat('Karl Pearson coefficient : ',cor(x,y))
10 print(cor.test(x,y))
11 print(cor.test(x,y,method='pearson'))|
```

```
> ## Karl Pearson(Correlation coefficient)
> x=c(78,36,98,25,75,82,90,62,65,39)
> y=c(84,51,91,60,68,62,86,58,53,47)
>
> rxy = var(x,y)/(sqrt(var(x))*sqrt(var(y)))
> cat('Karl Pearson coefficient : ',rxy)
Karl Pearson coefficient :  0.7804101
> ## Direction function
> cat('Karl Pearson coefficient : ',cor(x,y))
Karl Pearson coefficient :  0.7804101
> print(cor.test(x,y))

Pearson's product-moment correlation

data: x and y
t = 3.5302, df = 8, p-value = 0.00773
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.2964488 0.9454659
sample estimates:
cor
0.7804101
```

```

> print(cor.test(x,y,method='pearson'))

Pearson's product-moment correlation

data: x and y
t = 3.5302, df = 8, p-value = 0.00773
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.2964488 0.9454659
sample estimates:
cor
0.7804101

14 - ##### Problem-2 #####
15 x1=c(10,14,18,22,26,30)
16 y1=c(18,12,24,6,30,36)
17 print(cor(x1,y1))
18
19 - ##### Problem-3 #####
20 x2=c(10,15,12,17,13,16,24,14,22)
21 y2=c(30,42,45,46,33,34,40,35,39)
22 print(cor(x2,y2))
23 print(cor.test(x2,y2,method='pearson')) # Karl Pearson's solution
24 print(cor.test(x2,y2,method="spearman")) # Spearan's solution |
25
> x1=c(10,14,18,22,26,30)
> y1=c(18,12,24,6,30,36)
> print(cor(x1,y1))
[1] 0.6
> x2=c(10,15,12,17,13,16,24,14,22)
> y2=c(30,42,45,46,33,34,40,35,39)
> print(cor(x2,y2))
[1] 0.3307218
> print(cor.test(x2,y2,method='pearson')) # Karl Pearson's solution

Pearson's product-moment correlation

data: x2 and y2
t = 0.92718, df = 7, p-value = 0.3847
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.4272383 0.8156865
sample estimates:
cor
0.3307218

> print(cor.test(x2,y2,method="spearman")) # Spearan's solution

Spearman's rank correlation rho

data: x2 and y2
S = 72, p-value = 0.2912
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.4

```

```

26 - ##### Problem-4 #####
27 x3=c(68,64,75,50,64,80,75,40,55,64)
28 y3=c(62,58,68,45,81,60,68,48,50,70)
29 print(cor(x3,y3))
30 print(cor.test(x3,y3,method="pearson"))
31 print(cor.test(x3,y3,method="spearman",exact=F)) # without rectification
32

> x3=c(68,64,75,50,64,80,75,40,55,64)
> y3=c(62,58,68,45,81,60,68,48,50,70)
> print(cor(x3,y3))
[1] 0.6240869
> print(cor.test(x3,y3,method="pearson"))

    Pearson's product-moment correlation

data: x3 and y3
t = 2.2591, df = 8, p-value = 0.0538
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.009124948  0.900047265
sample estimates:
      cor
0.6240869

> print(cor.test(x3,y3,method="spearman",exact=F)) # without rectification

    Spearman's rank correlation rho

data: x3 and y3
S = 73.326, p-value = 0.09542
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.5555979

33 - ##### Problem-5 #####
34 - ## Linear Regression #####
35 a=1:9 # a=seq(1,8)
36 b=c(9,8,10,12,11,13,14,16,15)
37 re1 = lm(a~b) # a on b
38 print(re1)
39
40 re2 = lm(b~a) # b on a
41 print(re2)
42
43 print(summary(re1))
44 print(summary(re2))
45

```

```

> ##### Problem-5 #####
> ## Linear Regression #####
> a=1:9 # a=seq(1,8)
> b=c(9,8,10,12,11,13,14,16,15)
> re1 = lm(a~b) # a on b
> print(re1)

call:
lm(formula = a ~ b)

Coefficients:
(Intercept)          b
              -6.40        0.95

> re2 = lm(b~a) # b on a
> print(re2)

call:
lm(formula = b ~ a)

Coefficients:
(Intercept)          a
              7.25        0.95

>
> print(summary(re1))

call:
lm(formula = a ~ b)

Residuals:
    Min     1Q Median     3Q    Max 
-1.15  -0.80   0.05   0.80   1.15 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -6.400     1.449  -4.418  0.00309 **  
b            0.950     0.118   8.050 8.76e-05 *** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

```

Residual standard error: 0.9142 on 7 degrees of freedom
Multiple R-squared:  0.9025,    Adjusted R-squared:  0.8886
F-statistic: 64.79 on 1 and 7 DF,  p-value: 8.763e-05

> print(summary(re2))

call:
lm(formula = b ~ a)

Residuals:
    Min      1Q  Median      3Q     Max 
 -1.15   -0.80   0.05   0.80   1.15 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.2500    0.6641  10.92 1.20e-05 ***
a            0.9500    0.1180   8.05 8.76e-05 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

Residual standard error: 0.9142 on 7 degrees of freedom
 Multiple R-squared: 0.9025, Adjusted R-squared: 0.8886
 F-statistic: 64.79 on 1 and 7 DF, p-value: 8.763e-05

```

46 ##### Simple Linear regression
47 e=c(65,66,67,67,68,69,70,72)
48 f=c(67,68,65,68,72,72,69,71)
49 print(lm(e~f)) # e on f
50 print(lm(f~e)) # f on e
51 |
> e=c(65,66,67,67,68,69,70,72)
> f=c(67,68,65,68,72,72,69,71)
> print(lm(e~f)) # e on f

```

```

call:
lm(formula = e ~ f)

```

```

Coefficients:
(Intercept)          f
            30.3636     0.5455

```

```
> print(lm(f~e)) # f on e
```

```

call:
lm(formula = f ~ e)

```

```

Coefficients:
(Intercept)          e
            23.6667     0.6667

```

```
52 ##### Multiple Regression
53 f=c(20,23,25,27,21,29,22,24,27,35)
54 g=c(8,13,8,18,23,16,10,12,14,20)
55 h=c(28,23,38,16,20,28,23,30,26,32)
56 print(lm(f~(g+h)))
57
> f=c(20,23,25,27,21,29,22,24,27,35)
> g=c(8,13,8,18,23,16,10,12,14,20)
> h=c(28,23,38,16,20,28,23,30,26,32)
> print(lm(f~(g+h)))

call:
lm(formula = f ~ (g + h))

Coefficients:
(Intercept)                  g                  h
               5.1483        0.6190        0.4304
```

8th September 2020

Binomial and Poisson Distribution

> **Problem 1:**

1. Suppose $X \sim \text{Bin}(10, 0.4)$, what is $P(X = 7)$?

```
> ## Binomial Distribution
> #dbinom(x,size,probability)
> # This function gives the probability density function at each point
> dbinom(7,size=10,prob=0.4)
[1] 0.04246733
```

> **Problem 2:**

2. Suppose $Y \sim \text{Bin}(8, 0.15)$, what is $P(Y < 3)$?

```
> # pbinom(x,size,probability)
> # This function gives the cumulative probability of an event.
ability.
> pbinom(3,8,0.15)
[1] 0.9786475
```

> **Problem 3:**

3. Suppose $W \sim \text{Bin}(50, 0.12)$, what is $P(W > 2)$?

```
> #P(X>2)=1-P(X<=2)
> cat('P(X>2) : ',1-pbinom(2,size=50,prob=0.12))
P(X>2) : 0.9487358
> |
```

> Problem 4:

- > In a box of switches it is known 10% of the switches are faulty. A technician is wiring 30 circuits, each of which needs one switch. What is the probability that (a) all 30 work, (b) at most 2 of the circuits do not work?

c) at-most 2 of the circuit works : $P(X \leq 2)$

Probability of failure(i.e switches are not working) $\rightarrow 10\% (0.1)$

Probability of success(i.e switches are working) $\rightarrow 90\% (0.9)$

```
## Problem-4
cat('All 30 works : ',dbinom(30,size=30,prob=0.9))
cat('At most 2 circuit do not work P(X>=27) : ',1-pbinom(27,size=30,prob=0.9))
cat('At most 2 circuit do not work P(X>=27) : ',pbinom(2,size=30,prob=0.1))
cat('At most 2 circuit works P(X<=2) : ',pbinom(2,size=30,prob=0.9))

> ## Problem-4
> cat('All 30 works : ',dbinom(30,size=30,prob=0.9))
All 30 works : 0.04239116
> cat('At most 2 circuit do not work P(X>=27) : ',1-pbinom(27,size=30,prob=0.9))
At most 2 circuit do not work P(X>=27) : 0.4113512
> cat('At most 2 circuit do not work P(X<=2) : ',pbinom(2,size=30,prob=0.1))
At most 2 circuit do not work P(X<=2) : 0.4113512
> cat('At most 2 circuit works P(X<=2) : ',pbinom(2,size=30,prob=0.9))
At most 2 circuit works P(X<=2) : 3.5506e-26
```

> Problem 5:

- > If 10% of the Screws produced by an automatic machine are defective, find the probability that out of 20 screws selected at random, there are
- > (i) Exactly 2 defective (ii) At least 2 defectives
- > (iii) Between 1 and 3 defectives (inclusive)

Probability of success (i.e defective screws) $\rightarrow 0.1$

```
> ## Problem-5
> cat('Exactly 2 defective P(X=2) : ',dbinom(2,size=20,prob=0.1))
Exactly 2 defective P(X=2) : 0.2851798
> cat('Atleast 2 defectives P(X>=2) : ',1-dbinom(1,size=20,prob=0.1))
Atleast 2 defectives P(X>=2) : 0.7298297
> cat('Between 1 and 3 are defective P(1>=X<=3) : ',sum(dbinom(1:3,size=20,prob=0.1)))
Between 1 and 3 are defective P(1>=X<=3) : 0.74547
` |
```

1) In a large consignment of electric bulbs 10% are defective. A random sample of 20 is taken for inspection. Find the probability that

- (i) All are good bulbs,
- (ii) At most there are 3 defective bulbs,
- (iii) Exactly there are three defective bulbs.

(Solution: (i) 0.1216 (ii) 0.8666 (iii) 0.19)

Probability of defective \rightarrow 10% (i.e 0.1)

Probability of good bulbs \rightarrow 90% (i.e 0.9)

```
> ## Problem-6
> cat('All are good bulbs P(X=20) : ',dbinom(20,size=20,prob=0.9))
All are good bulbs P(X=20) : 0.1215767
> cat('Atmost 3 bulbs are defective P(X<=3) : ',pbisom(3,size=20,prob=0.1))
Atmost 3 bulbs are defective P(X<=3) : 0.8670467
> cat('Exactly 3 bulbs are defective P(X=3) : ',dbinom(3,size=20,prob=0.1))
Exactly 3 bulbs are defective P(X=3) : 0.1901199
```

For a random variable X with a binomial (20,1/2) distribution, find the following probabilities.

- (i). Find $Pr(X < 8)$
- (ii). Find $Pr(X > 12)$
- (iii). Find $Pr(8 \leq X \leq 1)$

```
> ### Problem-7
> cat('P(X<8) : ',pbisom(7,size=20,prob=0.5))
P(X<8) : 0.131588
> cat('P(X>12) = 1-P(X<=12) : ',1-pbisom(12,size=20,prob=0.5))
P(X>12) = 1-P(X<=12) : 0.131588
> cat('P(8<=X<=1) : ',sum(dbinom(1:8,size=20,prob=0.5)))
P(8<=X<=1) : 0.2517214
```

Poisson Distribution

Consider a computer system with Poisson job-arrival stream at an average of 2 per minute. Determine the probability that in any one-minute interval there will be

- i. 0 jobs;
- ii. exactly 2 jobs
- iii. at most 3 arrivals.

```
> cat('0-jobs P(X=0) : ',dpois(0,lambda = 2))
0-jobs P(X=0) : 0.1353353
> cat('Exaclty 2 jobs P(X=2) ',dpois(2,lambda=2))
Exaclty 2 jobs P(X=2) 0.2706706
> cat('At-most 3 arrivals P(X<=3) ',ppois(3,lambda=2))
At-most 3 arrivals P(X<=3) 0.8571235
|
```

Let X be the number of heads in 10 tosses of a fair coin.

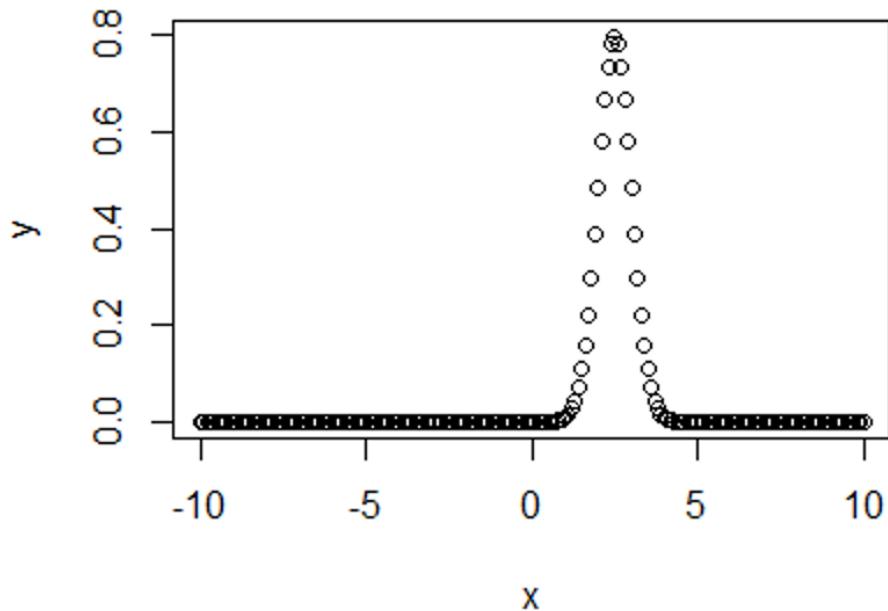
1. Find the probability of getting at least 5 heads (that is, 5 or more).
2. Find the probability of getting exactly 5 heads.
3. Find the probability of getting between 4 and 6 heads, inclusive

```
> ### Problem-2
> cat('Probability of getting at-least 5 heads P(X>=5)=1-P(X<5) : ',1-ppois(4,lambda=0.5))
Probability of getting at-least 5 heads P(X>=5)=1-P(X<5) : 0.0001721156
> cat('Probability of getting exactly 5 heads P(X=5) : ',dpois(5,lambda=0.5))
Probability of getting exactly 5 heads P(X=5) : 0.0001579507
> cat('Probability of getting between 4 and 6 heads : ',sum(dpois(4:6,lambda=0.5)))
Probability of getting between 4 and 6 heads : 0.00175062
|
```

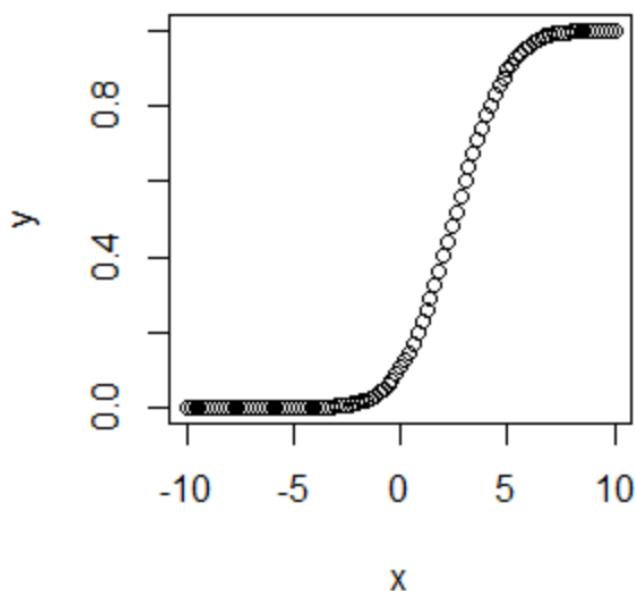
22th September

Normal Distribution

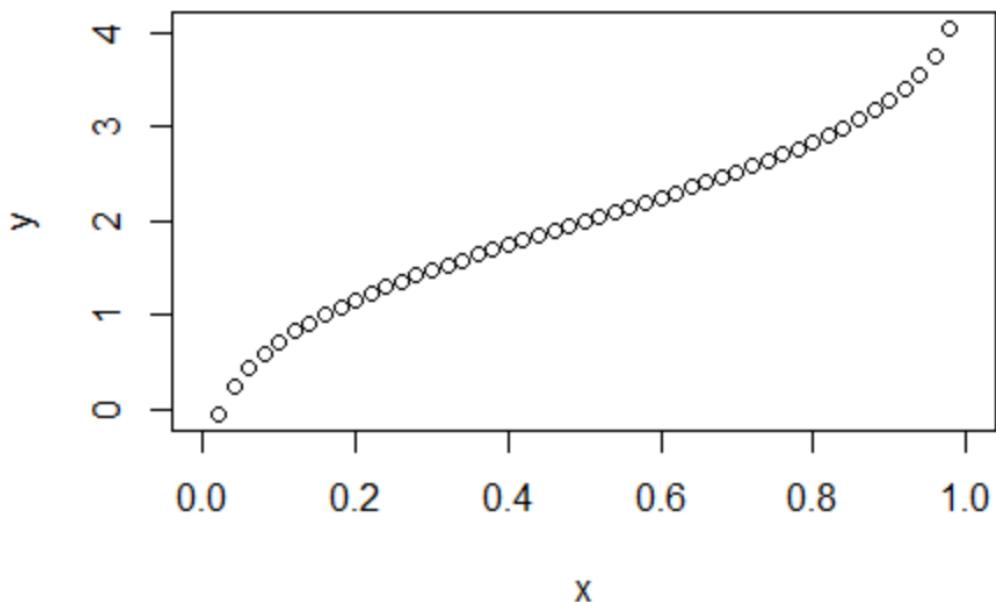
```
1 # Sequence of numbers between -10 and 10 incrementing by 0.1
2 x=seq(-10,10,by=.1)
3 y=dnorm(x,mean=2.5,sd=0.5) # mean=2.5 and standard deviation=0.5
4 plot(x,y)
```



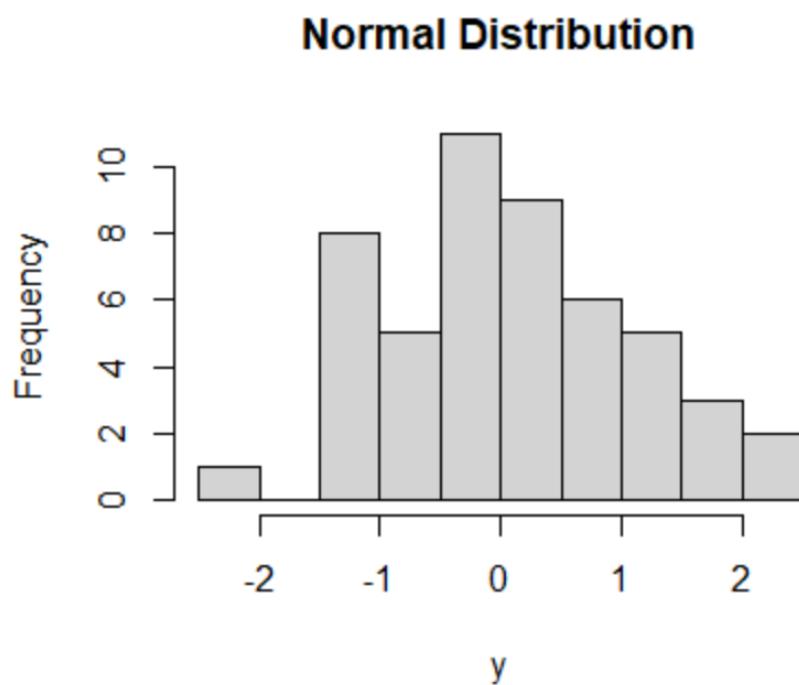
```
6 x=seq(-10,10,by=0.2)
7 y=pnorm(x,mean=2.5,sd=2)
8 plot(x,y) # pnorm graph
```



```
10 x=seq(0,1,by=0.02)
11 y=qnorm(x,mean=2,sd=1)
12 plot(x,y) # qnorm graph
```



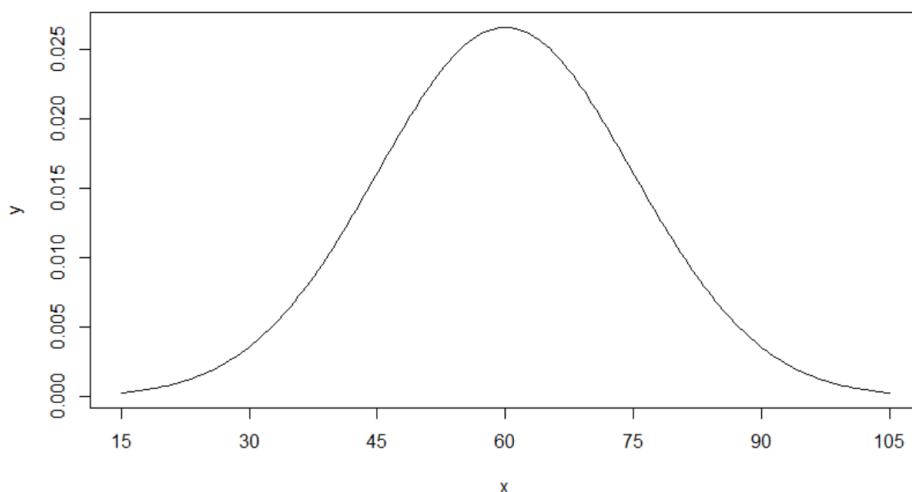
```
14 y=rnorm(50)
15 hist(y,main='Normal Distribution')
```



Program-1

Draw a normal distribution with the mean=60 and standard deviation=15.

```
17 ## Program-1 ##
18 x = seq(15,105,length=200)
19 y = dnorm(x,mean=60,sd=15)
20 plot(x,y,type="l",xaxt="n")
21 axis(1,at=c(15,30,45,60,75,90,105))|
```



1)

Suppose X is normal with mean 527 and standard deviation 105. Compute $P(X \geq 310)$

```
> ## Problem-1
> cat('P(X>=310) = 1-P(X<310) : ',1-pnorm(309,mean=527,sd=105))
P(X>=310) = 1-P(X<310) :  0.9810618
```

2)

Find $P(80 \text{ pts} < x < 95 \text{ pts.})$

```
> ## Problem-2
> cat('P(80<x<95) : ',sum(dnorm(81:94,mean=527,sd=105)))
P(80<x<95) :  8.446068e-06
```

3)

The weekly wages of 1000 workmen are normally distributed around a mean of Rs. 70 with S.D of Rs 5. Estimate the number of workers whose weekly wages will be

- (i) Between Rs 69 and Rs 72
- (ii) Less than Rs 69
- (iii) More than Rs 72

```
> ## Problem-3
> cat('P(69<=X<=72) = ',(pnorm(72,mean=70,sd=5)-pnorm(69,mean=70,sd=5))*1000)
P(69<=X<=72) = 234.6815
> cat('P(x<=69) = ',(pnorm(69,mean=70,sd=5))*1000)
P(x<=69) = 420.7403
> cat('P(x>72)=1-P(x<=72) = ',(1-pnorm(72,mean=70,sd=5))*1000)
P(x>72)=1-P(x<=72) = 344.5783
```

4)

In a test on 2000 Electric bulbs ,it was found that the life of particular make, was normally distributed with an average life of 2040 hours and S.D of 60 hours. Estimate the number of bulbs likely to burn for

- (i) More than 2150 hours
- (ii) Less than 1950 hours
- (iii) More than 1920 hours but less than 2160 hours
- (iv) More than 2150 hours

```
> ## Problem-4
> cat('P(X>2150)=1-P(X<=2150) : ',(1-pnorm(2150,mean=2040,sd=60))*2000)
P(X>2150)=1-P(X<=2150) : 66.75302
> cat('P(X<=1950) : ',pnorm(1950,mean=2040,sd=60)*2000)
P(X<=1950) : 133.6144
> cat('P(1920<=X<=2160) : ',(pnorm(2160,mean=2040,sd=60)-pnorm(1920,mean=2040,sd=60))*2000)
P(1920<=X<=2160) : 1908.999
```

6th October

```
1  ### Sum-1
2  xbar=9900
3  mu=10000
4  sig=120
5  n=30
6  z=(xbar-mu)/(sig/sqrt(n))
7  print(z)
8
9  z_alpha1 = qnorm(0.05)  # left tailed
10 print(z_alpha1)
11
12 z_alpha = qnorm(1-0.05,lower.tail = F)
13 print(z_alpha)
14 print(pnorm(z))
15
16 z_alphar = qnorm(1-0.05)          # right tailed
17 print(z_alphar)
18 print(pnorm(z,lower.tail = F))
19 #####
20
> xbar=9900
> mu=10000
> sig=120
> n=30
> z=(xbar-mu)/(sig/sqrt(n))
> print(z)
[1] -4.564355
> z_alpha1 = qnorm(0.05)  # left tailed
> print(z_alpha1)
[1] -1.644854
> z_alpha = qnorm(1-0.05,lower.tail = F)
> print(z_alpha)
[1] -1.644854
> print(pnorm(z))
[1] 2.505166e-06
> z_alphar = qnorm(1-0.05)          # right tailed
> print(z_alphar)
[1] 1.644854
> print(pnorm(z,lower.tail = F))
[1] 0.9999975
> |
```

```

21  #### Sum-2
22  xbar=14.6
23  mu=15.4
24  sig=2.5
25  n=35
26  z=(xbar-mu)/(sig/sqrt(n))
27  print("z : ")
28  print(z)
29
30  print("z alpha : ")
31  print(qnorm(1-0.05/2))    # /2 two tailed test
32  print(2*pnorm(z))
33

> xbar=14.6
> mu=15.4
> sig=2.5
> n=35
> print("z alpha : ")
[1] "z alpha : "
> print(qnorm(1-0.05/2))    # /2 two tailed test
[1] 1.959964
> print(2*pnorm(z))
[1] 5.010332e-06

34  #####
35  # Sum-3
36  P=0.12
37  Q=1-P
38  p=30/214
39  n=214
40  z=(p-P)/(sqrt((P*Q/n)))
41  print(qnorm(1-0.05))          # right tailed
42  print(pnorm(z,lower.tail=F))
43  print(prop.test(30,214,0.12,alt = "greater",correct=FALSE))
44  #####

```

```
> P=0.12
> Q=1-P
> p=30/214
> n=214
> z=(p-P)/(sqrt((P*Q/n)))
> print(qnorm(1-0.05))
[1] 1.644854
> print(pnorm(z,lower.tail=F))
[1] 0.1817408
> print(prop.test(30,214,0.12,alt = "greater",correct=FALSE))
```

1-sample proportions test without continuity correction

```
data: 30 out of 214, null probability 0.12
X-squared = 0.82583, df = 1, p-value = 0.1817
alternative hypothesis: true p is greater than 0.12
95 percent confidence interval:
 0.1056274 1.0000000
sample estimates:
      p 
0.1401869
```

13th October

```
> x=c(5,6,8,1,12,4,3,9,6,10)
> y=c(2,3,6,8,10,1,2,8)
> t.test(x,y) # t-test

Welch Two Sample t-test

data: x and y
t = 0.8679, df = 15.041, p-value = 0.3991
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-2.0374 4.8374
sample estimates:
mean of x mean of y
6.4      5.0

> t.test(x,y,alt="greater") # right tail

Welch Two Sample t-test

data: x and y
t = 0.8679, df = 15.041, p-value = 0.1995
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
-1.42732 Inf
sample estimates:
mean of x mean of y
6.4      5.0

> t.test(x,y,alt="less") # left tail

Welch Two Sample t-test

data: x and y
t = 0.8679, df = 15.041, p-value = 0.8005
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
-Inf 4.22732
sample estimates:
mean of x mean of y
6.4      5.0

> x = c(12.9,13.5,12.8,15.6,17.2,19.2,12.6,15.3,14.4,11.3)
> y = c(12.0,12.2,11.2,13.0,15.0,15.8,12.2,13.4,12.9,11.0)
> t.test(x,y)

Welch Two Sample t-test

data: x and y
t = 1.7889, df = 15.367, p-value = 0.09335
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.3042834 3.5242834
sample estimates:
mean of x mean of y
14.48    12.87
```

```

> t.test(x,y,alt='less')

    Welch Two Sample t-test

data: x and y
t = 1.7889, df = 15.367, p-value = 0.9533
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
-Inf 3.185243
sample estimates:
mean of x mean of y
14.48     12.87

> plants = c(20,10,7,4)
> chisq.test(plants,p=c(9/16,3/16,3/16,1/16))

Chi-squared test for given probabilities

data: plants
X-squared = 1.9702, df = 3, p-value = 0.5786

Warning message:
In chisq.test(plants, p = c(9/16, 3/16, 3/16, 1/16)) :
  Chi-squared approximation may be incorrect
```

Inbox Gmail Mail

Diamond Daily Digest

```

> # Good ness of fit (chi-square)
> x = c(5,4,3,2,1,0)
> n=5
> N=320
> P=0.5
> obf=c(14,56,110,88,40,12)
> exf=dbinom(x,n,P)*320    # expected frequency
> sum(obf)
[1] 320
> sum(exf)
[1] 320
> chisq=sum((obf-exf)^2/exf)
> qchisq(0.95,5)
[1] 11.0705
```

```
> x = c(0,1,2,3,4,5,6)
> f = c(275,72,30,7,5,2,1)
> Efx = sum(x*obf)
Warning message:
In x * obf :
  longer object length is not a multiple of shorter object length
> Ef = sum(obf)
> lambda = Efx/Ef
> expois = dpois(x,lambda)*sum(f)
> f1 = round(expois)
> sum(f)
[1] 392
> sum(f1)
[1] 386
> f-f1
[1] 247 -2 -68 -79 -52 -28 -12
> E = c(242,117,28,5) # omitted unwanted data and added
> o = c(275,72,30,15)
> chisq = sum((o-E)^2/E)
> df=2
> qchisq(0.95,df)
[1] 5.991465
> print('H0 is accepted')
[1] "H0 is accepted"
```

```
> x = c(0,1,2,3,4,5,6)
> f = c(109,53,131,12,35,21,12)
> Ef = sum(f)
> Efx = sum(f*x)
> lambda = Efx/Ef
> expois = dpois(x,lambda)*sum(f)
> f1      = round(expois)
> print(sum(f))
[1] 373
> print(sum(f1))
[1] 373
> print(f-f1)
[1] 47 -58  31 -48   8  11   9
> O=c(109,53,131,12,35,33)
> E=c(62,111,100,60,27,13)
> cat('chi^square : ',chisq = sum((O-E)^2/E))
chi^square : 147.0849> df=5
> cat('chi^square_alpha : ',qchisq(0.95,df))
chi^square_alpha : 11.0705> print('H0 is rejected')
[1] "H0 is rejected"
```

20th October

```
1  ### 1-way classification
2  df=read.csv("F:\\2) Second Year 2020-2021\\Fall semester 2020-2021"
3  r=c(t(as.matrix(df)))
4  f=c("Tech-A","Tech-B","Tech-C","Tech-D")
5  k=4 # number of columns
6  n=5 # number of rows
7  te = gl(k,1,k*n,factor(f))
8  cr = aov(r~te)
9  summary(cr)
> ### 1-way classification
> df=read.csv("F:\\2) Second Year 2020-2021\\Fall semester 2020-2021\\Practice\\anova.csv")
> r=c(t(as.matrix(df)))
> f=c("Tech-A","Tech-B","Tech-C","Tech-D")
> k=4 # number of columns
> n=5 # number of rows
> te = gl(k,1,k*n,factor(f))
> cr = aov(r~te)
> summary(cr)
      Df Sum Sq Mean Sq F value Pr(>F)
te          3 12.95   4.317    0.68  0.577
Residuals   16 101.60   6.350
> |
```



```
11  ### 2-way classification
12  df1=read.csv("F:\\2) Second Year 2020-2021\\Fall semester 2020-2021\\Practice\\anova_2way.csv")
13  r1=c(t(as.matrix(df1)))
14  f=c("A","B","C","D","E","F")
15  blocks = gl(4,6)
16  treatments = gl(6,1,24)
17  rbf1 = aov(r1~blocks+treatments)
18  summary(rbf1)
19
> df1=read.csv("F:\\2) Second Year 2020-2021\\Fall semester 2020-2021\\Practice\\anova_2way.csv")
> r1=c(t(as.matrix(df1)))
> f=c("A","B","C","D","E","F")
> blocks = gl(4,6)
> treatments = gl(6,1,24)
> rbf1 = aov(r1~blocks+treatments)
> summary(rbf1)
      Df Sum Sq Mean Sq F value    Pr(>F)
blocks      3 219.4   73.14   4.778   0.0157 *
treatments  5 901.2  180.24  11.773 9.28e-05 ***
Residuals   15 229.6   15.31
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
`|
```