

بسم الله الرحمن الرحيم

پروژه آمار و احتمال مهندسی

استاد: دکتر فریبا نصیرزاده

ارائه دهنده ها:

یوسف اسدی - 9932135

متین کاظمی - 40030266

محمد رضا نظیری - 40030505

سلام و درود امیدوارم که حالتون خوب باشه.
تو این pdf من قصد داریم تمام قسمت های کد و همچنین متد های کتابخانه های مختلف رو براتون تفسیر کنیم.
بطور کلی این پروژه شامل چهار قسمت معیار های مرکزی، معیار های پراکندگی و معیار های چولگی - برجستگی و نمودار ها است.
در این پروژه از چهار کتابخانه math و matplotlib و numpy و scipy استفاده کردیم.
داده های گسسته رو با پایتون زدیم و داده های پیوسته را با برنامه R زدیم و در ادامه کد پایتون برای داده های گسسته را بررسی میکنیم.

معیار های مرکزی:

۱. میانگین حسابی:
برای محاسبه میانگین حسابی از متد mean از کتابخانه numpy استفاده کردیم.

```
# calculating the arithmetic mean value of input using numpy
arithmeticMean = numpy.mean(data)
print("\nArithmetic Mean = " + str(arithmeticMean))
# -----
```

```
Arithmetic Mean = 26.24137931034483
```

خروجی:

۲. میانگین هندسی:

برای محاسبه میانگین هندسی از متد `gmean` کتابخانه `scipy` استفاده کردیم؛ اما نکته ای که اینجا مهم از این است که ابتدا چک کردم که همه دیتا های ورودی مثبت باشد و اگر مثبت بود آنگاه میانگین هندسی را حساب کند و اگر مقدار منفی نیز در دیتا ها وجود داشت آنگاه مقدار `Unavailable` را نشان دهد.

```
# calculating the geometric mean value of input using scipy
if min(data) >= 0:
    geometricMean = scipy.stats.gmean(data)
    print("\nGeometric Mean = " + str(geometricMean))
else:
    print("\nGeometric Mean = Unavailable")
# -----
```

```
Geometric Mean = 21.41138877289215
```

خروجی:

۳. میانگین هارمونیک:

برای محاسبه میانگین هارمونیک از متد `hmean` کتابخانه `scipy` استفاده کردیم؛ اما نکته ای که اینجا مهم از این است که ابتدا چک کردیم که همه دیتا های ورودی مثبت باشد و اگر مثبت بود آنگاه میانگین هارمونیک را حساب کند و اگر مقدار منفی نیز در دیتا ها وجود داشت آنگاه مقدار `Unavailable` را نشان دهد.

```
# calculating the harmonic mean value of input using scipy
if min(data) >= 0:
    harmonicMean = scipy.stats.hmean(data)
    print("\nHarmonic Mean = " + str(harmonicMean))
else:
    print("\nHarmonic Mean = Unavailable")
# -----
```

```
Harmonic Mean = 14.779766868764561
```

خروجی:

۴. میانه:

برای محاسبه میانه از متد median از کتابخانه numpy استفاده کردیم.

```
# calculating the median value of input using numpy
Median = numpy.median(data)
print("\nMedian = " + str(Median))
# -----
```

```
Median = 25.0
```

خروجی:

۵. مد:

برای محاسبه مد از چند تا از متد های کتابخانه numpy استفاده کردیم. ابتدا با متد unique تعداد هر کدام از دیتا ها را می‌شماریم. و ذخیره می‌کنیم. در ادامه با متد argwhere بیشترین تکرار را پیدا می‌کنیم. و در ادامه دیتا ها با بیشترین تکرار را با متد های flatten و tolist پیدا و در Modes ذخیره می‌کنیم و در نهایت آن مقدار ها را پرینت می‌کنیم.

```
# calculating the modes value of input using numpy
# find unique values in array along with their counts
vals, counts = numpy.unique(data, return_counts=True)

# find mode
mode_value = numpy.argmax(counts == numpy.max(counts))

# print list of modes
Modes = vals[mode_value].flatten().tolist()
print("\nModes = " + str(Modes))
# -----
```

```
Modes = [28.0]
```

خروجی:

۶. چارک‌ها:

برای محاسبه چارک‌ها از متد `quantile` کتابخانه `numpy` استفاده میکنیم. که اگر مقادیر `[0, 0.25, 0.5, 0.75, 1]` را به عنوان ورودی به این متد بدهیم، یک آرایه از چارک‌ها به ما می‌دهد که در ایندکس ۱ چارک اول و در ایندکس ۲ میانه و در ایندکس ۳ چارک سوم وجود دارد.

```
# calculating the quantiles value of input using numpy
Quantiles = numpy.quantile(data, [0, 0.25, 0.5, 0.75, 1])
print("\n1st Quantile = " + str(Quantiles[1]))
print("2st Quantile = " + str(Quantiles[2]))
print("3st Quantile = " + str(Quantiles[3]))
# -----
```

```
1st Quantile = 16.0
2st Quantile = 25.0
3st Quantile = 37.0
```

خروجی:

معیارهای پراکندگی:

۱. دامنه تغییرات:

برای محاسبه دامنه تغییرات مقدار ماکزیمم را از مینیمم کم میکنیم.

```
# calculating the range value of input
Range = max(data) - min(data)
print("\nRange = " + str(Range))
# -----
```

```
Range = 48.0
```

خروجی:

۲. دامنه میان چارکی:

برای محاسبه دامنه میان چارکی چارک سوم را از چارک اول کم میکنیم.

```
# calculating the interquartile range value of input
```

```
Range = Quantiles[3] - Quantiles[1]
```

```
print("\nInterquartile Range = " + str(Range))
```

```
# -----
```

```
Interquartile Range = 21.0
```

خروجی:

۳. انحراف:

برای محاسبه انحراف هر مقدار را از میانگین کم میکنیم.

```
# calculating the Deviation value of input using numpy
```

```
Deviation = [abs(x - arithmeticMean) for x in data]
```

```
print("\nDeviation = " + str(Deviation))
```

```
# -----
```

خروجی:

```
Deviation = [24.24137931034483, 21.24137931034483, 19.24137931034483, 14.24137931034483, 14.24137931034483, 13.24137931034483, 13.24137931034483, 10.24137931034483, 10.24137931034483, 9.24137931034483, 7.241379310344829, 6.241379310344829, 4.241379310344829, 1.2413793103448292, 1.2413793103448292, 0.24137931034482916, 1.7586206896551708, 1.7586206896551708, 1.7586206896551708, 1.7586206896551708, 9.75862068965517, 10.75862068965517, 16.75862068965517, 18.75862068965517, 18.75862068965517, 18.75862068965517, 22.75862068965517, 23.75862068965517]
```

۴. انحراف متوسط:

برای محاسبه این مقدار حاصل جمع قدر مطلق اختلاف دیتا ها از میانگین را تقسیم بر تعداد دیتا ها میکنیم.

```
# calculating the Average Absolute Deviation value of input
```

```
medianAbsoluteDeviation = sum([abs(x - arithmeticMean) for x in data]) / len(data)
```

```
print("\nMedian Absolute Deviation = " + str(medianAbsoluteDeviation))
```

```
# -----
```

```
Median Absolute Deviation = 11.71462544589774
```

خروجی:

۵. واریانس:

برای محاسبه واریانس از متد var کتابخانه numpy استفاده میکنیم.

```
# calculating the variance value of input using numpy
Variance = numpy.var(data)
print("\nVariance = " + str(Variance))
# -----
```

```
Variance = 197.70035671819267
```

خروجی:

۶. انحراف معیار:

برای محاسبه انحراف معیار جذر واریانس را میگیریم.

```
# calculating the standard deviation value of input using statistics
standardDeviation = math.sqrt(Variance)
print("\nStandard Deviation = " + str(standardDeviation))
# -----
```

```
Standard Deviation = 14.060595887735081
```

خروجی:

۷. ضریب تغییرات:

برای محاسبه ضریب تغییرات حاصل انحراف معیار را تقسیم بر میانگین حسابی میکنیم.

```
# calculating the coefficient of variation value of input
coefficientOfVariation = standardDeviation / arithmeticMean
print("\nCoefficient of Variation = " + str(coefficientOfVariation))
# -----
```

```
Coefficient of Variation = 0.5358177145129006
```

خروجی:

معیار های چولگی و برجستگی:

۱. ضریب چولگی اول پیرسون:

برای محاسبه مقدار مد را از میانگین کم میکنیم و تقسیم بر انحراف معیار میکنیم.

```
# calculating the Pearson Mode Skewness value of input
pearsonModeSkewness = (arithmeticMean - Modes[0]) / standardDeviation
print("\nPearson Mode Skewness = " + str(pearsonModeSkewness))
# -----
```

```
Pearson Mode Skewness = -0.1250744067816641
```

خروجی:

۲. ضریب چولگی سوم پیرسون:

برای محاسبه مقدار میانه را از میانگین کم میکنیم و در ۳ ضرب میکنیم و سپس تقسیم بر انحراف معیار میکنیم.

```
# calculating the Pearson Median Skewness value of input
pearsonMedianSkewness = 3 * (arithmeticMean - Median) / standardDeviation
print("\nPearson Median Skewness = " + str(pearsonMedianSkewness))
# -----
```

```
Pearson Median Skewness = 0.26486344965528924
```

خروجی:

۳. ضریب گشتاور چولگی:

با استفاده از متد skew از کتابخانه scipy ضریب گشتاور چولگی را محاسبه میکنیم.

```
# calculating the Skewness value of input
Skewness = scipy.stats.skew(data)
print("\nSkewness = " + str(Skewness))
# -----
```

```
Skewness = 0.23170839189382642
```

خروجی:

۴. معیار چولگی با چارک ها:

برای محاسبه چارک اول و سوم را جمع و از سه برابر چارک دوم کم کردیم و سپس تقسیم بر حاصل جمع چارک اول و سوم میکنیم.

```
# calculating the Skewness Quantile Coefficient value of input
skewnessQuantileCoefficient = (Quantiles[3] + Quantiles[1] -
                              (2 * Quantiles[2])) / (Quantiles[3] - Quantiles[1])
print("\nSkewness Quantile Coefficient = " + str(skewnessQuantileCoefficient))
# -----
```

```
Skewness Quantile Coefficient = 0.14285714285714285
```

خروجی:

۵. معیار برجستگی (برآمدگی):

با استفاده از متد kurtosis از کتابخانه scipy معیار برجستگی را محاسبه میکنیم.

```
# calculating the kurtosis value of input
kurtosis = scipy.stats.kurtosis(data)
print("\nkurtosis = " + str(kurtosis))
# -----
```

```
kurtosis = -1.066431982012745
```

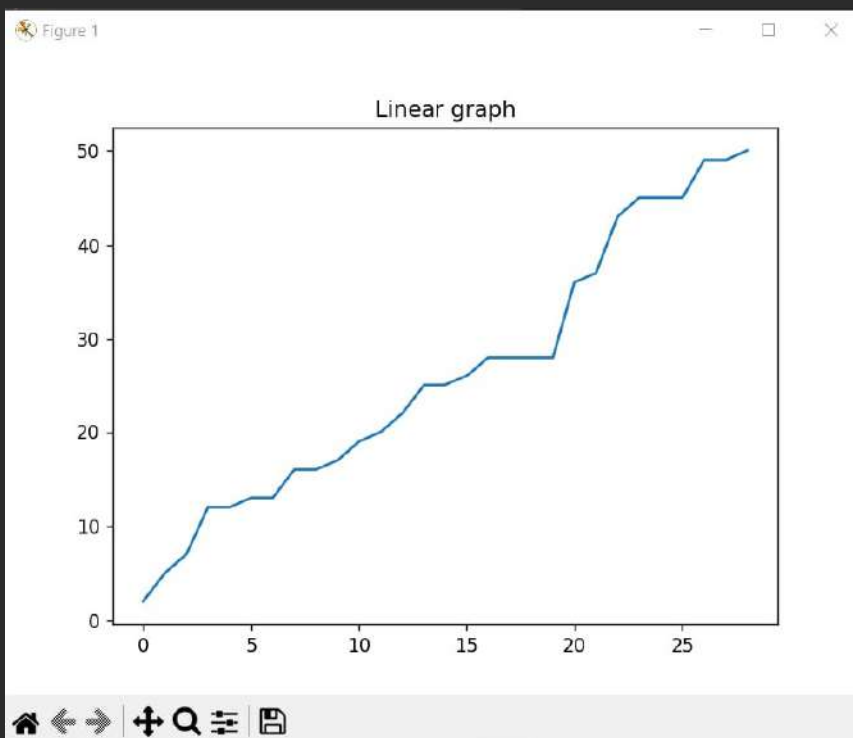
خروجی:

رسم نمودار ها:

نمودار خطی:

رسم نمودار خطی با متد plot از کتابخانه matplotlib استفاده می کنیم و برای نمایش از متد show استفاده میکنیم.

```
# drawing the Linear graph the data
plt.plot(data)
plt.title('Linear graph')
plt.show()
# -----
```

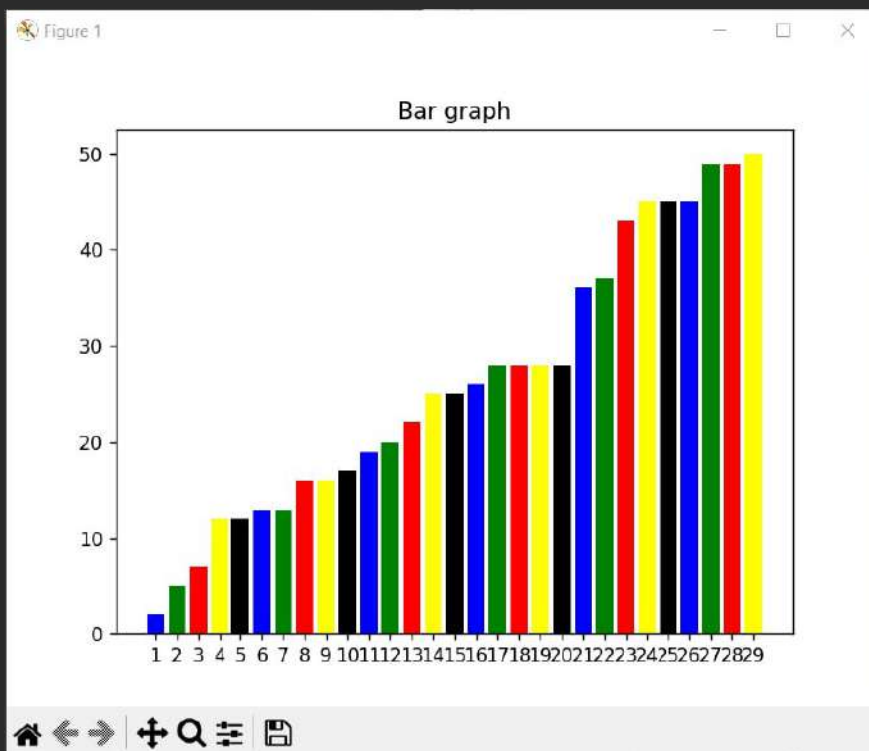


خروجی:

نمودار میله‌ای:

با استفاده از متد bar از کتابخانه matplotlib این کار را انجام می دهیم.

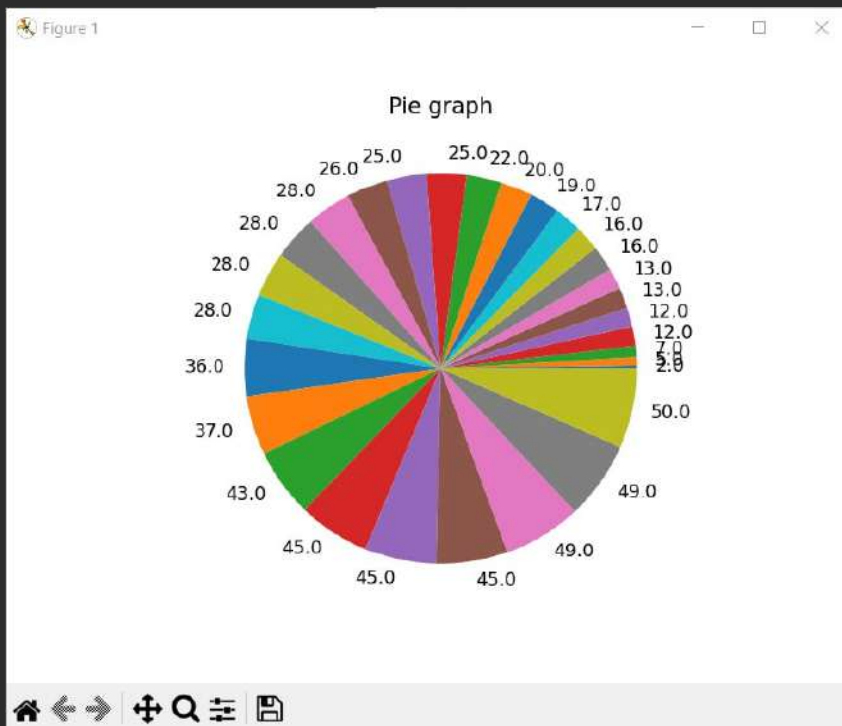
```
# drawing the Bar graph the data
left = [x for x in range(1, len(data) + 1)]
height = data
plt.bar(left, height, tick_label=left,
        width=0.8,
        color=['blue', 'green', 'red', 'yellow', 'black', 'blue',
               'red', 'yellow', 'black'])
plt.title('Bar graph')
plt.show()
# -----
```



خروجی:

نمودار دایره ای:
با استفاده از متد pie از کتابخانه matplotlib این کار را انجام می دهیم.

```
# drawing the Pie graph the data
plt.pie(data, labels = data)
plt.title('Pie graph')
plt.show()
# -----
```

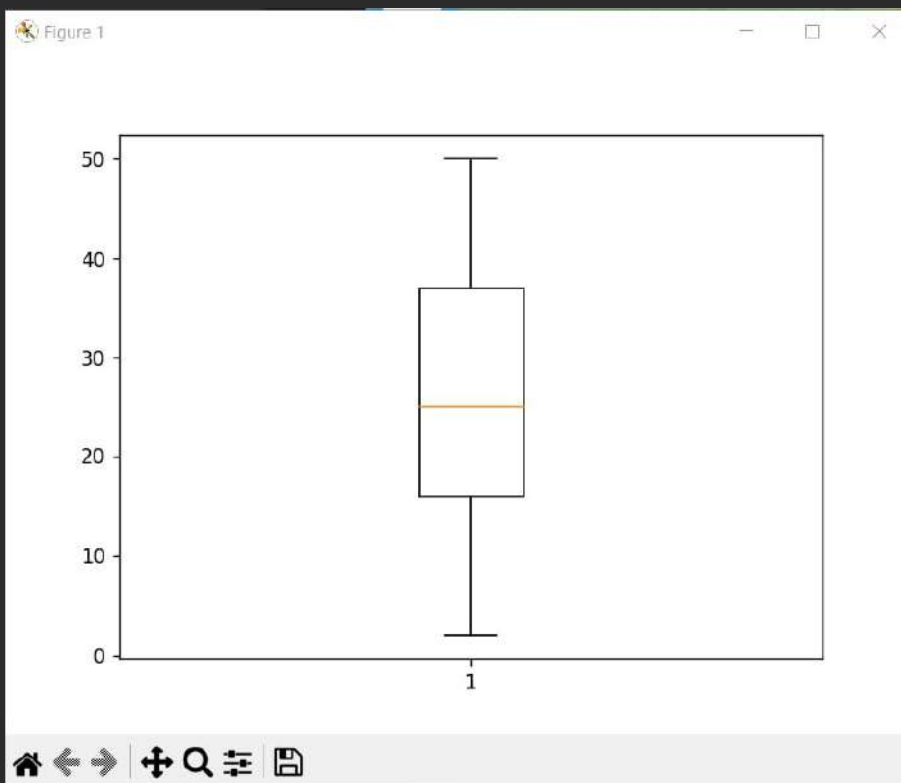


خروجی:

نمودار جعبه ای:

با استفاده از متد boxplot از کتابخانه matplotlib این کار را انجام می دهیم.

```
# drawing the Pie graph the data
plt.boxplot(data)
plt.show()
# -----
```

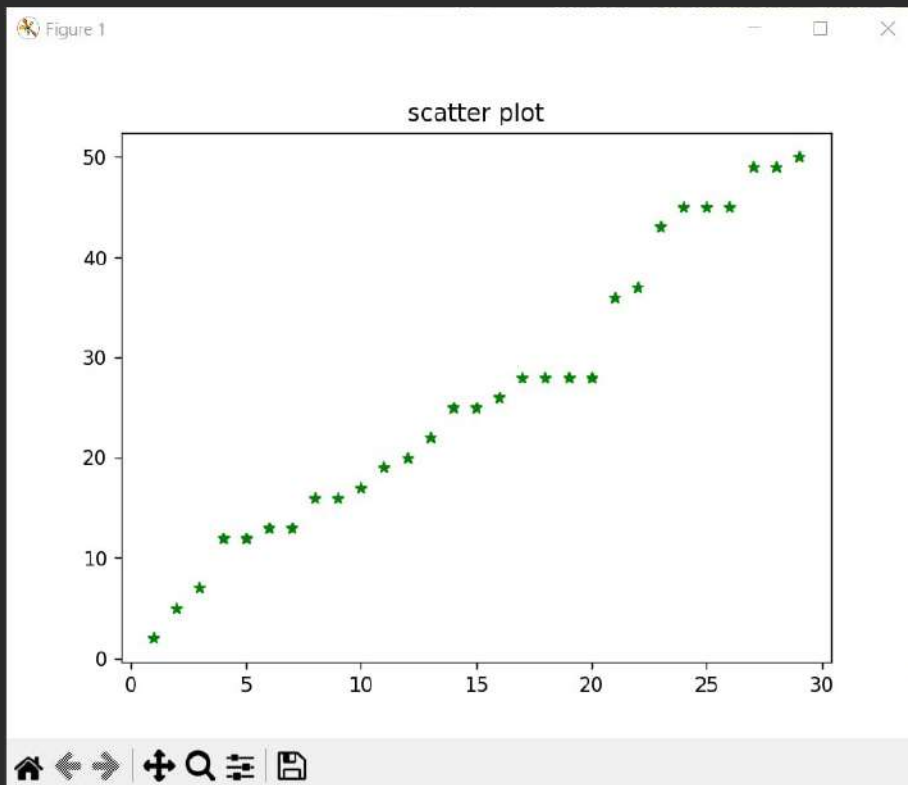


خروجی:

نمودار پراکندگی:

با استفاده از متد scatter از کتابخانه matplotlib این کار را انجام میدهیم.

```
# drawing the scatter plot the data
x = [x for x in range(1, len(data) + 1)]
y = data
plt.scatter(x, y, label="stars", color="green",
            marker="*", s=30)
plt.title('scatter plot')
plt.show()
# -----
```



خروجی:

خوب تا اینجا در رابطه با دیتا های گسسته محاسبات را انجام دادیم .
از اینجا به بعد با استفاده از زبان R داده های پیوسته را بررسی میکنیم.

```
1 ## frequency table
2 data<-read.table(file="C:/Users/ASUS/Desktop/University/Semester 3/amar/project/nahaii/continues data/input2.txt")
3 str(data)
4 data<-data$V1
5 head(data)
6
7 decimalplaces <- function(x) {
8   if ((x%%1) != 0) {
9     nchar(strsplit(sub('0+$', '', as.character(x)), ".", fixed=TRUE)[[1]][[2]])
10  } else {
11    return(0)
12  }
13 }
14 dvec<-vector()
15 for(i in 1:length(data)){
16   dvec[i]<-decimalplaces(data[i])
17 }
18 unig_d<-1/(10^max(dvec))
19 u_d<-0.5*unig_d
20 u_d
21 n<-length(data)
22 #redad_rade
23 num<-round(1+3.322*log10(n))
24 num
25 max_val<-max(data)
26 min_val<-min(data)
27 Range_d<-(max_val+u_d)-(min_val-u_d)
28 step_val<-ceiling(Range_d/num*10)/10
29 step_val
30 #rain_rade ha
31 data_break<-seq(min_val-u_d,max_val+u_d,step_val)
32 if(length(data_break)<(num+1)){data_break<-c(data_break,(data_break[num]+step_val))}
33 for(i in 1:length(data_break)){
34   data_break[i]<-data_break[i] + u_d
35 }
36 data_break
37 #mid-point(markaz_rade ha)
38 midpoint<-seq((min_val-u_d)+step_val/2,(max_val+u_d)+step_val/2,step_val)
39 midpoint
40 x<-cut(data,breaks=data_break, dig.lab = 4)
41 x
42 #rain_rade ha
43 y<-table(x)
44 y
45 #jadval faravani
46 df<-data.frame(y)
47 df
48 #add mid-point to table
49 df$midpnt<-midpoint
50 df
51 #add relative frequency to table
52 df$rF<-df$Freq/n
53 df
54 view(df)
55 #####
56 #add cumulative frequency to table
57 df$F<-cumsum(df$Freq)
58 df
59 #add relative cumulative frequency to table
60 df$rF<-df$F/n
61 df
62 view(df)
63 #####
64 #Pie
65 df$pie<-round(360*df$rF,1)
66 df
67 #####
68 # mean
69 wmean<-sum(df$midpnt*df$Freq)/n
70 wmean
71 Gmean<-(prod(df$midpnt^df$Freq))^(1/n)
72 Gmean
73 Hmean<-n/sum(df$Freq/df$midpnt)
74 Hmean
75
76 #variance
77 Var<-sum(df$Freq*(df$midpnt-wmean)^2)/(n-1)
78 Var
79 Sd<-sqrt(Var)
80 Sd
81 CV<-Sd/wmean
82 CV
83
84 ## Histogram
85 data<-read.table(file="C:/Users/ASUS/Desktop/University/Semester 3/amar/project/nahaii/continues data/input2.txt")
86 str(data)
87 data<-data$V1
88 hist(data, nclass=num, main="Histogram plot", col="blue")
89
90
91 ##Box Plot
92 boxplot(data)
93
```



```
> u_d
[1] 5e-15
```

0.5 واحد داده :

```
> num
[1] 8
```

تعداد رده :

```
> step_val
[1] 12.4
```

طول رده :

تعیین رده ها :

```
> data_break
[1] -49.9456696 -37.5456696 -25.1456696 -12.7456696 -0.3456696 12.0543304 24.4543304 36.8543304 49.2543304
```

مرکز رده ها :

```
> midpoint
[1] -43.74567 -31.34567 -18.94567 -6.54567 5.85433 18.25433 30.65433 43.05433
```

رده ها :

```
> x<-cut(data,breaks=data_break, dig.lab = 4)
> x
[1] (12.05,24.45] (12.05,24.45] (36.85,49.25] (-12.75,-0.3457] (-12.75,-0.3457] (-37.55,-25.15]
[7] (12.05,24.45] (24.45,36.85] (-37.55,-25.15] (36.85,49.25] (12.05,24.45] (-49.95,-37.55]
[13] (-12.75,-0.3457] (24.45,36.85] (-0.3457,12.05] (-12.75,-0.3457] (-49.95,-37.55] (24.45,36.85]
[19] (24.45,36.85] (-37.55,-25.15] (12.05,24.45] (-0.3457,12.05] (-25.15,-12.75] (36.85,49.25]
[25] (-25.15,-12.75] (-0.3457,12.05] (-0.3457,12.05] (36.85,49.25] (36.85,49.25] (-0.3457,12.05]
[31] (36.85,49.25] (36.85,49.25] (12.05,24.45] (-49.95,-37.55] (-25.15,-12.75] (-49.95,-37.55]
[37] (-37.55,-25.15] (12.05,24.45] (-25.15,-12.75] (24.45,36.85] (24.45,36.85] (-25.15,-12.75]
[43] (-0.3457,12.05] (36.85,49.25] (12.05,24.45] (-12.75,-0.3457] (12.05,24.45] (24.45,36.85]
[49] (-12.75,-0.3457] (-25.15,-12.75] (24.45,36.85] (-12.75,-0.3457] (-37.55,-25.15] (-0.3457,12.05]
[55] (-37.55,-25.15] (-37.55,-25.15] (24.45,36.85] (36.85,49.25] (24.45,36.85] (24.45,36.85]
[61] (-25.15,-12.75] (-25.15,-12.75] (-12.75,-0.3457] (-0.3457,12.05] (-0.3457,12.05] (-12.75,-0.3457]
[67] (-49.95,-37.55] (36.85,49.25] (24.45,36.85] (-37.55,-25.15] (-25.15,-12.75] (24.45,36.85]
[73] (12.05,24.45] (36.85,49.25] (-25.15,-12.75] (-25.15,-12.75] (36.85,49.25] (24.45,36.85]
[79] (-49.95,-37.55] (36.85,49.25] (12.05,24.45] (-49.95,-37.55] (-37.55,-25.15] (-49.95,-37.55]
[85] (24.45,36.85] (-49.95,-37.55] (-12.75,-0.3457] (-37.55,-25.15] (-12.75,-0.3457] (-25.15,-12.75]
[91] (36.85,49.25] (24.45,36.85] (36.85,49.25] (-25.15,-12.75] <NA> (12.05,24.45]
[97] (-37.55,-25.15] (-12.75,-0.3457] (-25.15,-12.75]
8 Levels: (-49.95,-37.55] (-37.55,-25.15] (-25.15,-12.75] (-12.75,-0.3457] (-0.3457,12.05] ... (36.85,49.25]
```

تعیین رده ها :

```
> y
x
(-49.95,-37.55] (-37.55,-25.15] (-25.15,-12.75] (-12.75,-0.3457] (-0.3457,12.05] (12.05,24.45]
          9          11          14          12          9          12
(24.45,36.85] (36.85,49.25]
          16          15
```

```
> df
```

	x	Freq
1	(-49.95,-37.55]	9
2	(-37.55,-25.15]	11
3	(-25.15,-12.75]	14
4	(-12.75,-0.3457]	12
5	(-0.3457,12.05]	9
6	(12.05,24.45]	12
7	(24.45,36.85]	16
8	(36.85,49.25]	15

جدول فراوانی :

```
> df
```

	x	Freq	midpnt
1	(-49.95,-37.55]	9	-43.74567
2	(-37.55,-25.15]	11	-31.34567
3	(-25.15,-12.75]	14	-18.94567
4	(-12.75,-0.3457]	12	-6.54567
5	(-0.3457,12.05]	9	5.85433
6	(12.05,24.45]	12	18.25433
7	(24.45,36.85]	16	30.65433
8	(36.85,49.25]	15	43.05433

اضاف کردن مرکز رده ها
به جدول فراوانی :

```
> df
```

	x	Freq	midpnt	rf
1	(-49.95,-37.55]	9	-43.74567	0.09090909
2	(-37.55,-25.15]	11	-31.34567	0.11111111
3	(-25.15,-12.75]	14	-18.94567	0.14141414
4	(-12.75,-0.3457]	12	-6.54567	0.12121212
5	(-0.3457,12.05]	9	5.85433	0.09090909
6	(12.05,24.45]	12	18.25433	0.12121212
7	(24.45,36.85]	16	30.65433	0.16161616
8	(36.85,49.25]	15	43.05433	0.15151515

اضاف کردن فراوانی نسبی
به جدول فراوانی :

```
> df
```

	x	Freq	midpnt	rf	F	rF
1	(-49.95,-37.55]	9	-43.74567	0.09090909	9	0.09090909
2	(-37.55,-25.15]	11	-31.34567	0.11111111	20	0.20202020
3	(-25.15,-12.75]	14	-18.94567	0.14141414	34	0.34343434
4	(-12.75,-0.3457]	12	-6.54567	0.12121212	46	0.46464646
5	(-0.3457,12.05]	9	5.85433	0.09090909	55	0.55555556
6	(12.05,24.45]	12	18.25433	0.12121212	67	0.67676768
7	(24.45,36.85]	16	30.65433	0.16161616	83	0.83838384
8	(36.85,49.25]	15	43.05433	0.15151515	98	0.98989899

اضاف کردن فراوانی
تجمعی و فراوانی
تجمعی نسبی
به جدول فراوانی :

جدول فراوانی :

	x	Freq	midpnt	rf	F	rF
1	(-49.95,-37.55]	9	-43.74567	0.09090909	9	0.09090909
2	(-37.55,-25.15]	11	-31.34567	0.11111111	20	0.20202020
3	(-25.15,-12.75]	14	-18.94567	0.14141414	34	0.34343434
4	(-12.75,-0.3457]	12	-6.54567	0.12121212	46	0.46464646
5	(-0.3457,12.05]	9	5.85433	0.09090909	55	0.55555556
6	(12.05,24.45]	12	18.25433	0.12121212	67	0.67676768
7	(24.45,36.85]	16	30.65433	0.16161616	83	0.83838384
8	(36.85,49.25]	15	43.05433	0.15151515	98	0.98989899

معیار های مرکزی:

```
> wmean<-sum(df$midpnt*df$Freq)/n
> wmean
[1] 3.290145
```

۱. میانگین حسابی:

```
> Gmean<-(prod(df$midpnt^df$Freq))^(1/n)
> Gmean
[1] 20.20516
```

۲. میانگین هندسی:

```
> Hmean<-n/sum(df$Freq/df$midpnt)
> Hmean
[1] -1550.736
```

۳. میانگین هارمونیک:

معیارهای پراکندگی:

```
> var<-sum(df$Freq*(df$midpnt-wmean)^2)/(n-1)
> var
[1] 812.6045
```

۱. واریانس:

```
> sd<-sqrt(var)
> sd
[1] 28.50622
```

۲. انحراف معیار:

```
> CV<-sd/wmean
> CV
[1] 8.664121
```

۳. ضریب تغییرات:

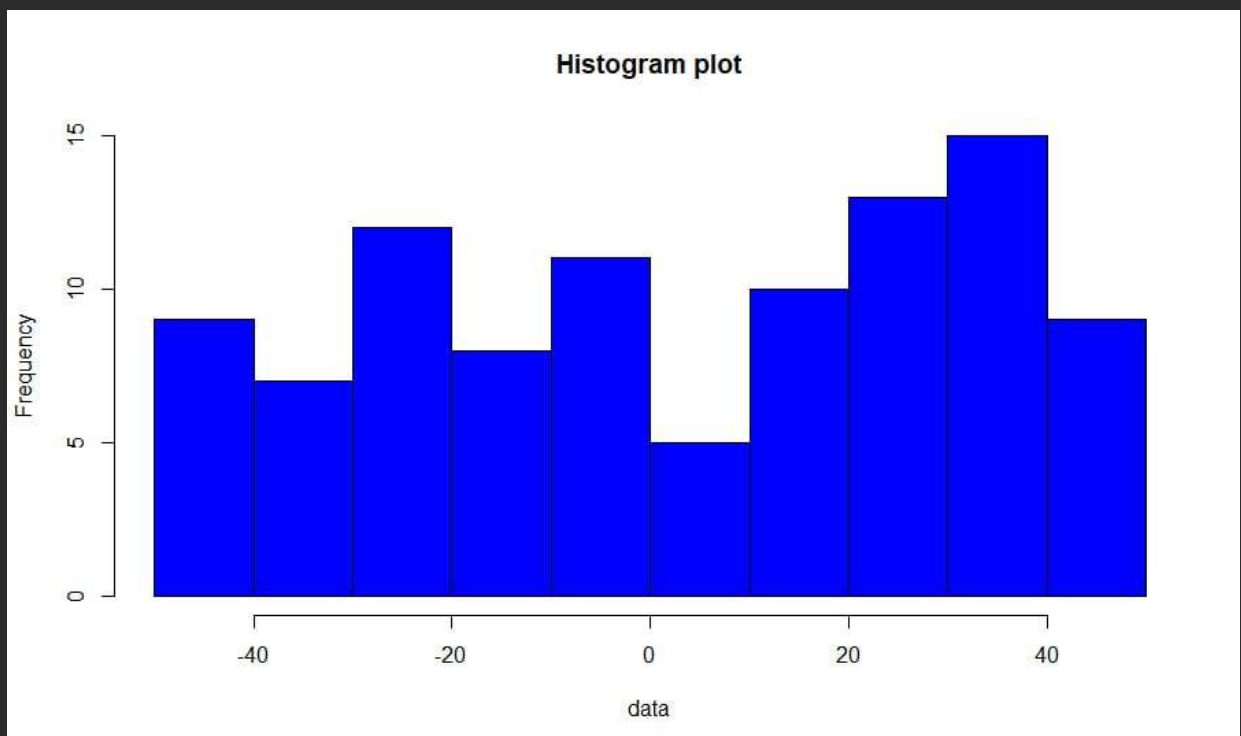
نمودار:

داده ها را با این دستورات میاوریم.

```
> data<-read.table(file="C:/Users/ASUS/Desktop/University/Semester 3/amar/project/nahai/continues data/input2.txt")
> str(data)
'data.frame': 99 obs. of 1 variable:
 $ v1: num 20.9 17.82 41.77 -5.45 -9.82 ...
> data<-data$v1
```

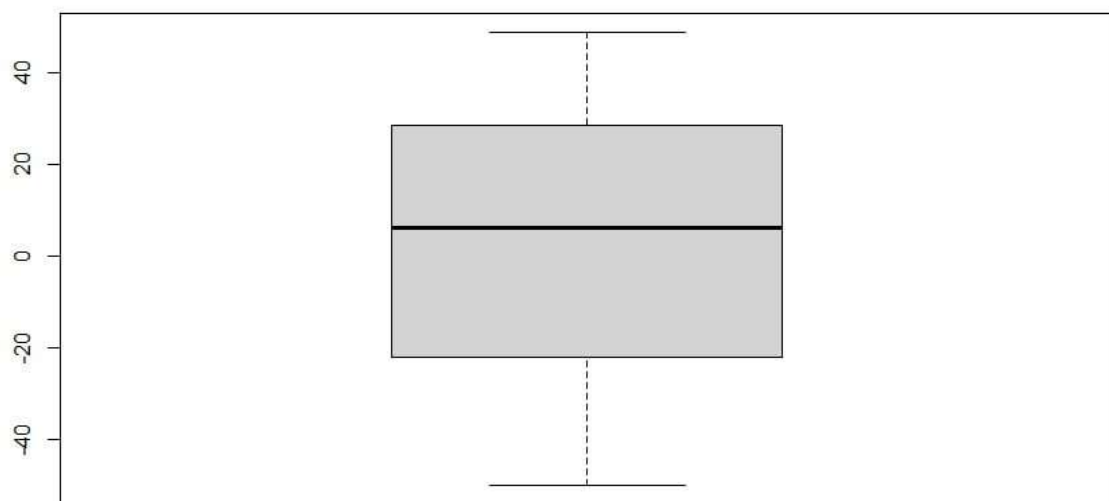
۱. نمودار هیستوگرام:

```
88 hist(data, nclass=num, main="Histogram plot", col="blue")
89
```



۲. نمودار جعبه ای:

```
91 ##Box Plot  
92 boxplot(data)
```



پایان