REPORT OF DATA MINING

Lab1

He NI

BI2

2015042

Part A Discrete series

1. *Generate a discrete series of 1000 random data (values included between 0 and 10):*

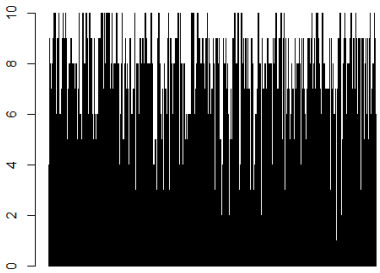
In order to don't change the value: I used "set.seed"

```
set.seed(250)
A = round(runif(1000,0,10))
```

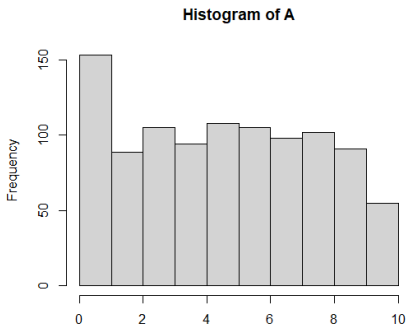
values	
A	num [1:1000] 3 8 2 8 8 10 8 1 5 8 ...

2. *Represent this series in the form of a histogram: to do so, you may use the R functions barplot or hist. See R help to find the right parameters.*

barplot(A) :



hist(A) :



3. Determine the mode, the median and the mean of this series without using the predefined R functions.

Mode:

names(which.max(table(A)))

```
> names(which.max(table(A)))
[1] "5"
```

Mean:

sum(A)/1000

```
> sum(A)/1000
[1] 4.99
```

Median:

sum(sort(A)[500:501])/2

```
> sum(sort(A)[500:501])/2
[1] 5
```

4. Verify the mean and the median value of your series using the functions mean(·) and median(·). The results should be identical with these of question 3.

Mean:

```
> mean(A)
[1] 4.99
```

Median:

```
> median(A)
[1] 5
```

5. Explain why the mean and median values of this series may be very different.

Because outlier values in our data can distort the results and visualizations.

6. Determine the range, the variance and the standard deviation:

• Without using the predefined R functions.

Range:

min(A),max(A)

```
> min(A)
[1] 0
> max(A)
[1] 10
```

Var:

```
ans = 0
i = 0
for(v in A)
{
   ans = ans + (v -mean(A))^2
   print(ans)
   i = i+1
}
var = sum(ans)/1000
print(var)
```

```
> print(var)
[1] 8.2959
```

Sd:

```
ans = 0
i = 0
for(v in A){
   ans = ans + (v-mean(A))^2
   print(ans)
   i = i+1
}
sd = sqrt(sum(ans)/1000)
print(sd)
```

```
> print(sd)
[1] 2.88026
```

• Using the predefined R functions: range(·), var(·) and sd(·).

Range:

```
> range(A)
[1]  0 10
```

Var:

```
> var(A)
[1] 8.304204204
```

Sd:

```
> sd(A)
[1] 2.881701616
```

• *Comment the results.*

For variance

Using the formula: $S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$, The result is $\boldsymbol{S^2}$, which is **biased estimate**

Calculate the expected value of the sample variance to estimate the difference between the biased variance S^2 and the unbiased variance σ^2

$$E[S^2] = E[\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}]$$

By deducing the formula, the final result is $S^2 = \frac{n-1}{n}\sigma^2$, Bias is $\frac{1}{n}\sigma^2$,

Then I can get the formula of **unbiased estimates** σ2.

And also using var(A) can get σ 2 , i.e., **unbiased estimates** , The formula is $\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$

I have : S^2

```
> print(var)
[1] 8.2959
```

and

```
> var(A)
[1] 8.304204204
```

: σ^2

So according to $S^2 = \frac{n-1}{n}\sigma^2$ then $8.2959 \div \frac{1000-1}{1000} = 8.304204204$

Then the above formula can also be changed to:

```
ans = 0
i = 0
for(v in A)
{
  ans = ans + (v -mean(A))^2
  print(ans)
  i = i+1
}
var = sum(ans)/(1000-1)
print(var)
```

```
> print(var)
[1] 8.304204

> var(A)
[1] 8.304204204
```

Standard deviation ceteris paribus

```
set.seed(250)
A = round(runif(1000,0,10))
ans = 0
i = 0
for(v in A){
  ans = ans + (v-mean(A))^2
  print(ans)
  i = i+1
}
sd = sqrt(sum(ans)/(1000-1))
print(sd)
```

```
> print(sd)
[1] 2.987327

> sd(A)
[1] 2.881701616
```

Part B Grouped discrete series

1. *Input this series and respresent it as a histogram.*

• The R function c(v1, · · · , vN) creates a vector with N values. Use this function to generate the vectors for the marks and number of students having each mark.

```
Mark = c(5,8,9,10,11,12,13,14,16)
Number = c(10,12,48,23,24,48,9,7,13)
```

• The function plot(data1,data2,type="h") is the only one available to generate a histogram from 2 vectors. You can use the command "? plot" to learn more about this function.

```
plot(Mark,Number,type="h")
```

2. *Determine the position and dispersion measures.*

Measures of Position

Mode:

```
Mode = Mark[which.max(Number)]
```

```
> Mark[which.max(Number)]
[1] 9
```

Mean:

```
weighted.mean(Mark,Number)
```

```
> weighted.mean(Mark,Number)
[1] 10.67526
```

Median:

```
ans = 0
i = 0
 for(t in Number){
    if(ans<sum(Number)/2) {
        ans = ans + t
        print(ans)
        i = i+1
    }
 }
print(i)
Mark[i+1]
```

```
[1] 10
[1] 22
[1] 70
[1] 93
[1] 117
> print(i)
[1] 5
> Mark[i+1]
[1] 12
```

Measures of Dispersion

Range:

```
range(Mark)
```

```
> range(Mark)
[1]  5 16
```

Variance:

```
ans = 0
i = 0
for(c in Mark){
   for(j in Number){
      if (i <= Number){
      ans = ans + (c-weighted.mean(Mark,Number))^2
      print(ans)
      i = i+1
      }
      else {i = 0}
   }
}
var = sum(ans)/(sum(Number)-1)
print(var)
```

```
[1] 746.7111
[1] 775.064
There were 50 or more warnings (us
 50)
> var = sum(ans)/(sum(Number)-1)
> print(var)
[1] 4.015875
```

Standard Deviation:

```
for(c in Mark){
   for(j in Number){
      if (i <= Number){
      ans = ans + (c-weighted.mean(Mark,Number))^2
      print(ans)
      i = i+1
      }
      else {i = 0}
   }
}
sd = sqrt(sum(ans)/(sum(Number)-1))
print(sd)
```

```
[1] 1500.971
There were 50 or more warnings (use wa
 50)
> sd = sqrt(sum(ans)/(sum(Number)-1))
> print(sd)
[1] 2.788737
```

3. Explain why this series has a bimodal distribution.

It could be that one group of students is underprepared for the class (perhaps because of a lack of previous classes). The other group may have overprepared.

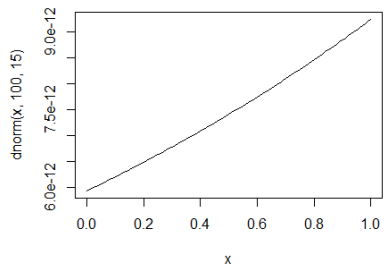
Part C Normal distributions

The R function rnorm(n,m,sd) generates a sample of n random variables that follow a normal distribution of mean m and standard deviation sd. In this exercise, we propose to generate a sample to simulate the human IQ. Human IQ has a mean value of 100 and a variance of 225.

```
set.seed(250)
x = rnorm(10, mean=100, sd=15)
```

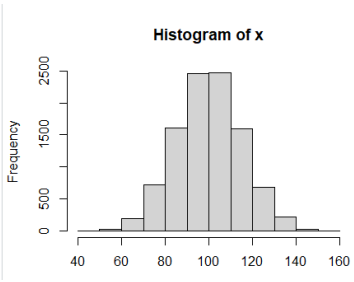
1． Use the function "curve(· · ·)" to display the probability density function of this distribution (dnorm(x, μ, σ) for a Gaussian distribution).

```
dnorm(x,100,15)
curve(dnorm(x,100,15))
```



2． Generate a sample of size 100000 and display its histogram.

hist(x)



3． Assess the mean value and the standard deviation of your sample. Comment.

mean(x)

```
> mean(x)
[1] 99.98217
```

sd(x)

```
> sd(x)
[1] 14.99234
```

4 . Find the percentage of your sample that has an IQ bellow 60.

```
v = paste(round(100*(length(x[x<60])/10000), 2), "%", sep="")
v
```

```
> v = paste(round(100
> v
[1] "0.26%"
```

5 . Find the percentage of your sample that has an IQ above 130.

```
v = paste(round(100*(length(x[x<60])/10000), 2), "%", sep="")
v
```

```
> v = paste(round(10
> v
[1] "2.5%"
```

6 . Find the range of values that contains 95 percent of your sample around the mean

```
min = mean(x)-1.96*sd(x)
max = mean(x)+1.96*sd(x)
print(min)
print(max)
```

```
> min = mean(x)-1.96*sd(x)
> max = mean(x)+1.96*sd(x)
> print(min)
[1] 70.59719
> print(max)
[1] 129.3672
```

Part D IQ analysis

In this exercise, we want to assess the affect of malnutrition on the human IQ. Knowing that the average IQ is of 100 with a standard deviation of 15, we will modelise the human population with random sample of different sizes and compare them with IQ sample data from people that suffered from malnutrition.

```
set.seed(250)
x1 = rnorm(10, mean=100, sd=15)
x2 = rnorm(1000, mean=100, sd=15)
x3 = rnorm(100000, mean=100, sd=15)
```

1 .Generate 3 different samples of size 10, 1000 and 100000 with a mean value of 100 and a standard deviation of 15 (function rnorm(·)).

• For each sample, evaluate its mean value and its standard deviation.

```
mean(x1)
mean(x2)
mean(x3)
sd(x1)
sd(x2)
sd(x3)
```

```
> mean(x1)
[1] 98.15036
> mean(x2)
[1] 99.95488
> mean(x3)
[1] 99.91877
> sd(x1)
[1] 9.121071
> sd(x2)
[1] 15.27824
> sd(x3)
[1] 14.99788
```

• Compare the values you found for the mean and standard deviation with the theoretical values.

If we have more samples, the SD will be more accurate

• Calculate the standard error and IC95 of the estimated mean values of each sample.

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}}$$

```
sd(x1)/sqrt(length(x1))
sd(x2)/sqrt(length(x2))
sd(x3)/sqrt(length(x3))
```

```
> sd(x1)/sqrt(length(x1))
[1] 2.884336
> sd(x2)/sqrt(length(x2))
[1] 0.4831403
> sd(x3)/sqrt(length(x3))
[1] 0.04742745
```

SE of x1 is 2.884336
SE of x2 is 0.4831403
SE of x3 is 0.04742745

```
min1 = mean(x1)-1.96*sd(x1)
max1 = mean(x1)+1.96*sd(x1)
```

```
print(min1)
print(max1)
min2 = mean(x2)-1.96*sd(x2)
max2 = mean(x2)+1.96*sd(x2)
print(min2)
print(max2)
min3 = mean(x3)-1.96*sd(x3)
max3 = mean(x3)+1.96*sd(x3)
print(min3)
print(max3)
```

```
> min1 = mean(x1)-1.96*sd(x1)
> max1 = mean(x1)+1.96*sd(x1)
> print(min1)
[1] 80.27306
> print(max1)
[1] 116.0277
> min2 = mean(x2)-1.96*sd(x2)
> max2 = mean(x2)+1.96*sd(x2)
> print(min2)
[1] 70.00953
> print(max2)
[1] 129.9002
> min3 = mean(x3)-1.96*sd(x3)
> max3 = mean(x3)+1.96*sd(x3)
> print(min3)
[1] 70.52293
> print(max3)
[1] 129.3146
```

IC_{95} of x1 is [80.27306, 116.0277]

IC_{95} of x2 is [70.00953, 129.9002]

IC_{95} of x3 is [70.52293, 129.3146]

• Comment on your previous results.

If we have more samples, the IC_{95} will be more accurate, and the Standard Error will be smaller

We now want to assess the effect of malnutrition on the IQ. To this end, we will analyze the data from a sample of people that suffered from malnutrition during their childhood.

2 .Using the command read.table(file), open the file malnutrition.csv.

```
data1=read.table("E:\\addons\\malnutrition.csv")
```

3 .Compute the mean and standard deviation of this new sample.

```
attach(data1)
mean(V1)
sd(V1)
```

```
> mean(V1)
[1] 87.98
> sd(V1)
[1] 9.677611
```

4 .Using the statistical measures at your disposal, what can you conclude on the effect of malnutrition on the IQ when comparing this sample to your previous sample of 100000 elements ?

• Compare the mean and standard deviation of both samples.

```
> mean(V1)
[1] 87.98
> sd(V1)
[1] 9.677611
```

```
> mean(x3)
[1] 99.91877
> sd(x3)
[1] 14.99788
```

Mean: IQ for malnutrition below 100,000 samples

SD: the smaller the SD, the closer the sample distribution is to the mean, and the variance of IQ for malnutrition is less than the variance of IQ for 100,000 samples.

• Compute the confidence intervals for both comparisons.

```
> print(min3)
[1] 70.52293
> print(max3)
[1] 129.3146
```

```
> print(minM)
[1] 69.01188
> print(maxM)
[1] 106.9481
```

• Comment on your results.

Malnutrition affects IQ, making it lower