DATA ANALYSIS

APPLICATION BY

NURALIEV SALOKHIDDIN

NORMURODOV JAKHON

INTRODUCTION

METHODS:

DESCRIPTIVE STATISTICS

STATISTICAL ESTIMATION

STATISTICAL TESTING

ANALYSIS OF VARIANCE

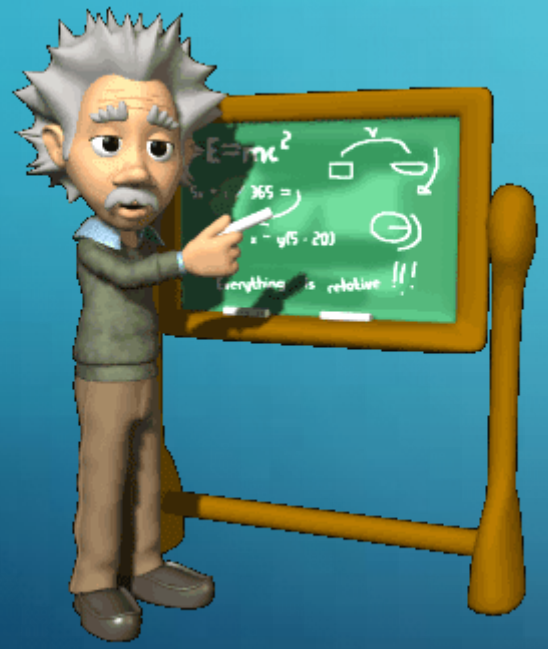
DESCRIPTIVE STATISTICS

- BASIC DSCRIPTIVE STATISTICS
- HISTOGRAM
- BOXPLOT

PYTHON


Этот компьютер
Barrons Ielts
Camera
Новая папка
Books
Ulsan
Topics in IT
Computee network
Data analyse
Assignment Computer...
Assignment Data Anal...
Assignment Topics in IT
Панель управления
Internet Explorer
AIDA64
Oracle VM VirtualBox
Python for...
Ashampoo WinOptimi...
PC
Python

PROGRAMMING CODE



my_project.py

Functions.py

```
from Functions import Bd
from Functions import Bp
from Functions import Hg
import sys
import os
def clear():
    clear = lambda: os.system('cls')
    clear()
clear()
def mainMenu():
    print("\n              ****** WELCOME TO MY PROJECT *****\n\n")
    print("          ***** GLOBAL SMART IT CONVERGENCE\n\n\n\n\n")
    print("COURSE: DATA ANALYSIS ")
    print("STUDENTS: SATTOROV SAMARIDDIN, BATIROV SULTONBEK")
    input()
    return 0
mainMenu()
clear()
```

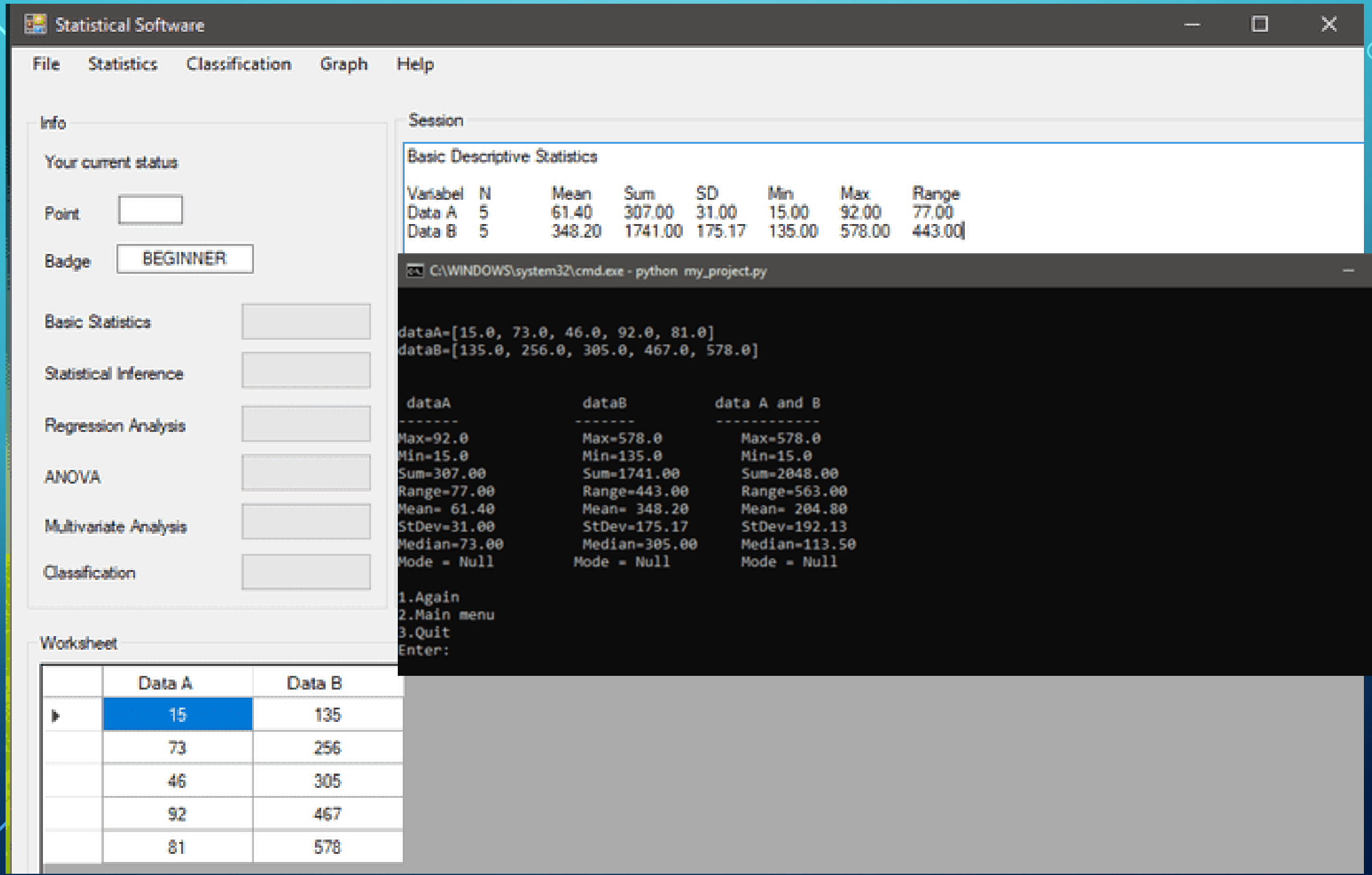
```
def menu():
    print("                    ***********MAIN MENU**************")
    #time.sleep(1)
    print()
    choice = input("""
                      A: Basic Descriptive Statistics
                      B: Histogram
                      C: Boxplot
                      D: Quit
                      Please enter your choice: """)
    if choice == "A" or choice =="a":
        clear()
        Bd()
        print("\n")
        print("1.Again")
        print("2.Main menu")
        print("3.Quit")
        chose = int(input("Enter:"))
        if chose == 1:
            clear()
            Bd()
            input()
            clear()
            menu()
        elif chose == 2:
            clear()
            menu()
        else:   sys.exit()
```

```
elif choice == "B" or choice =="b":
    clear()
    Hg()
    print("\n")
    print("1.Again")
    print("2.Main menu")
    print("3.Quit")
    chose = int(input("Enter:"))
    if chose == 1:
        clear()
        Hg()
        clear()
        menu()
    elif chose == 2:
        clear()
        menu()
    else:
        sys.exit()
```

```
elif choice == "C" or choice =="c":
    clear()
    Bp()
    print("\n")
    print("1.Again")
    print("2.Main menu")
    print("3.Quit")
    chose = int(input("Enter:"))
    if chose == 1:
        clear()
        Bp()
        clear()
        menu()
    elif chose == 2:
        clear()
        menu()
    else:
        sys.exit()
```

```
elif choice=="D" or choice=="d":
        sys.exit
    else:
        clear()
        print("You must only select either A,B or C.")
        print("Please try again")
        menu()
menu()
```

BASIC DESCRIPTIVE STATISTICS



SUPERIORITY OF OUR SOFTWARE

C:\WINDOWS\system32\cmd.exe - python my_project.py

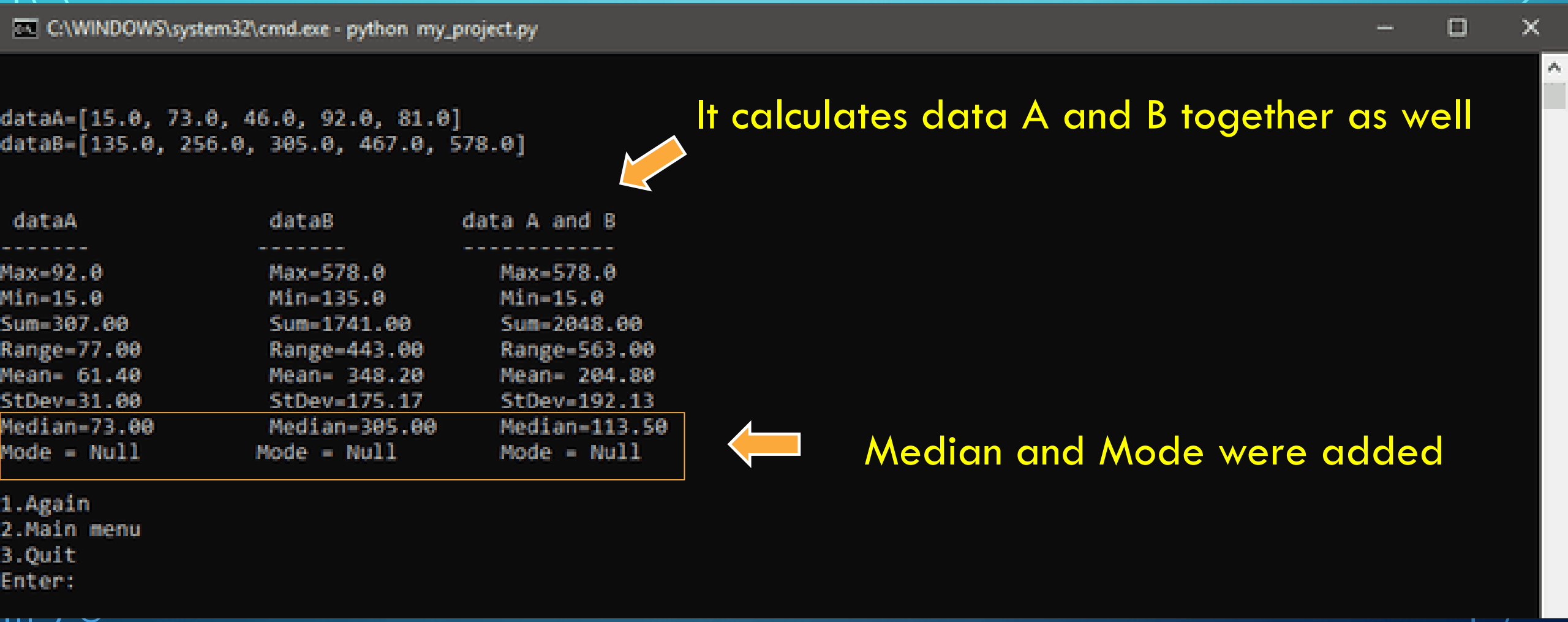
```
How much data do you want to enter?
For dataA = 5


Number1=15
Number2=73
Number3=46
Number4=92
Number5=81


For dataB = 5


Number1=135
Number2=256
Number3=305
Number4=467
Number5=578


dataA=[15.0, 73.0, 46.0, 92.0, 81.0]
dataB=[135.0, 256.0, 305.0, 467.0, 578.0]


 dataA              dataB          data A and B
-------            -------         ------------
Max=92.0           Max=578.0          Max=578.0
Min=15.0           Min=135.0          Min=15.0
Sum=307.00         Sum=1741.00        Sum=2048.00
```

You can choose how much data you want to enter

You can enter data directly in the app

(You don`t have to extract it from excel)

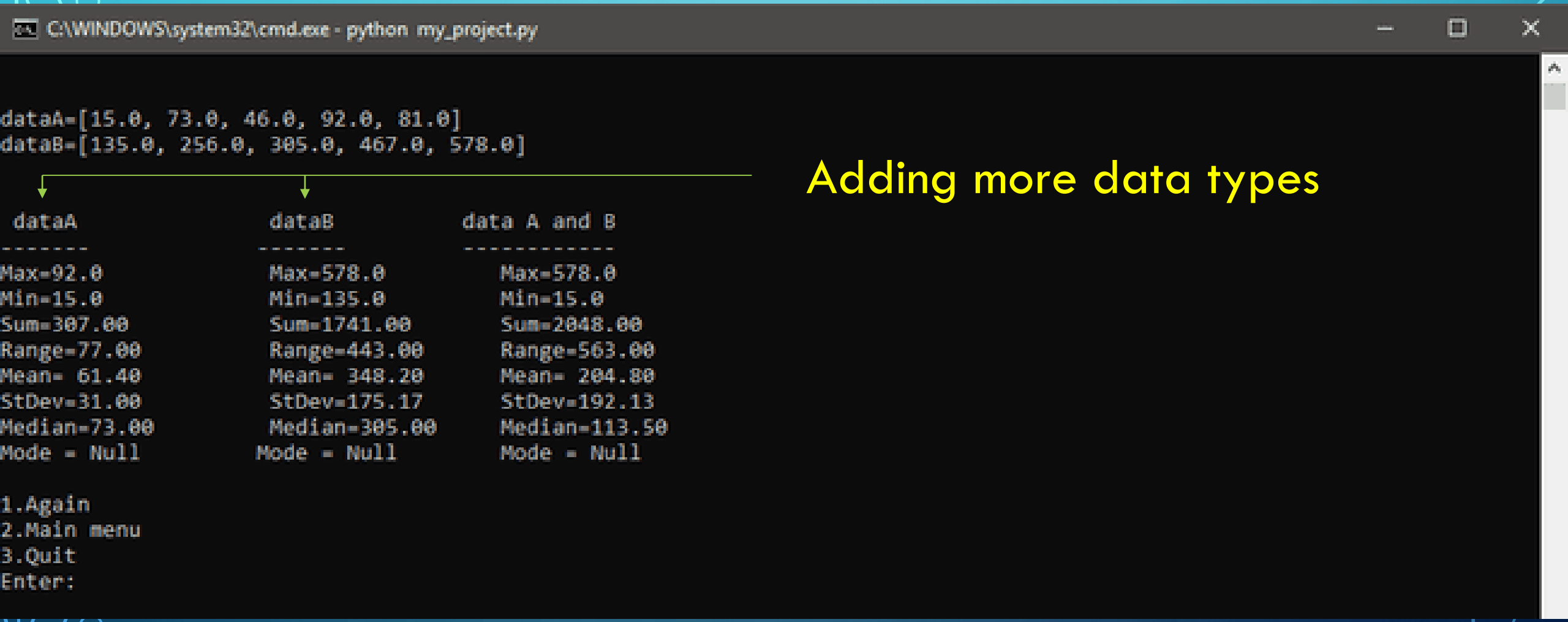
SUPERIORITY OF OUR SOFTWARE



It calculates data A and B together as well

Median and Mode were added

FUTURE UPDATES



Adding more data types

HISTOGRAM

Statistical Software

File Statistics Classification Graph Help

Figure 1: Histogram

File

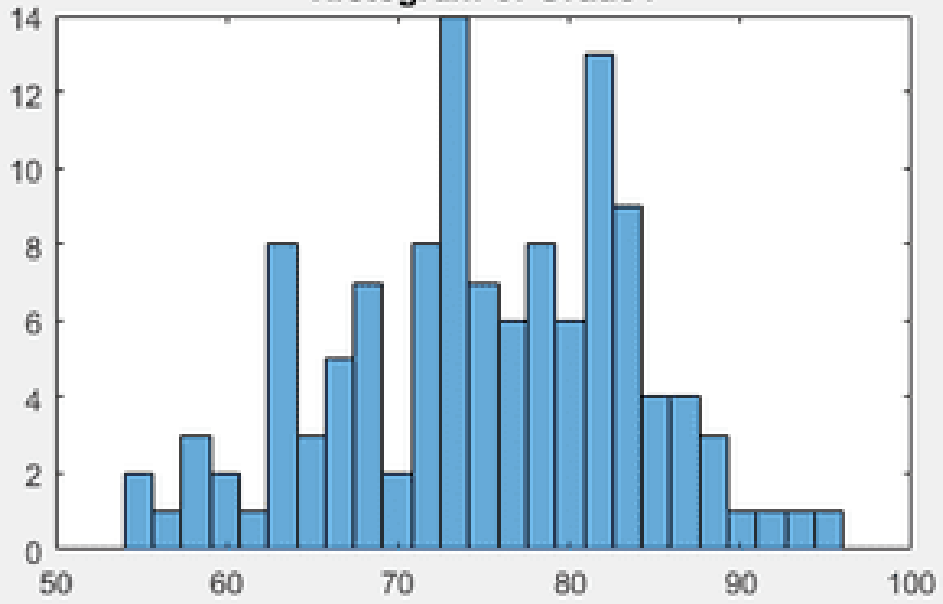
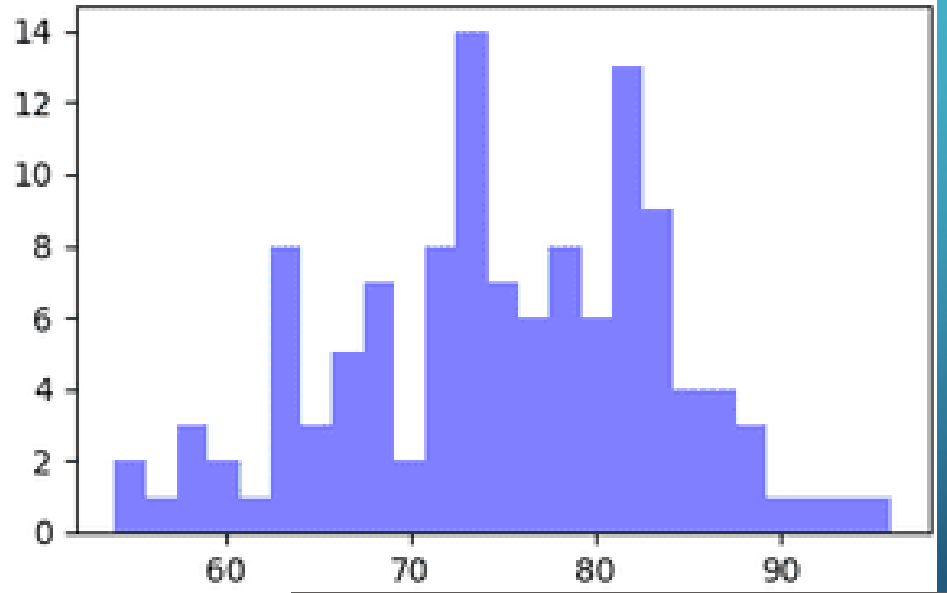


Figure 1



Histogram

Variable Grade1

Default no. of bins

Specify number of bins 25

OK Cancel

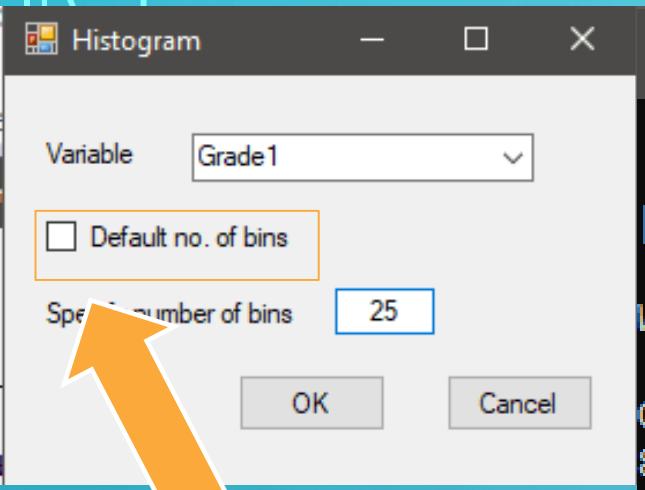
Worksheet

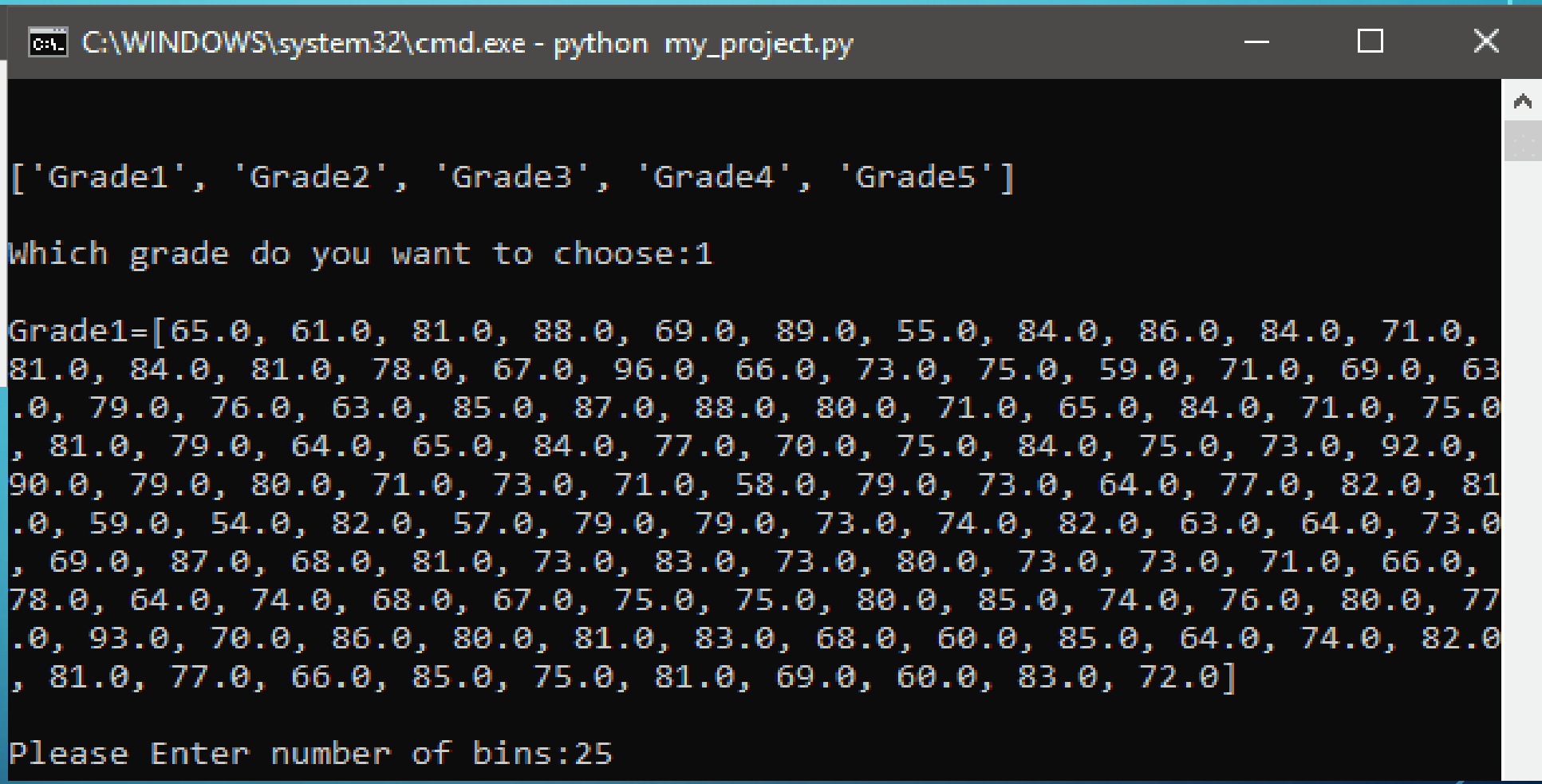
	Grade1	Grade2	Grade3	Grade4	Grade5
	65	77	69	75	69
	61	74	70	66	68
	81	80	71	74	79
	88	76	80	88	79
	69	77	74	69	76
	89	93	78	77	80

C:\WINDOWS\system32\cmd.exe - python my_project.py

```
['Grade1', 'Grade2', 'Grade3', 'Grade4', 'Grade5']

Which grade do you want to choose:1

Grade1=[65.0, 61.0, 81.0, 88.0, 69.0, 89.0, 55.0, 84.0, 86.0, 84.0, 71.0,
81.0, 84.0, 81.0, 78.0, 67.0, 96.0, 66.0, 73.0, 75.0, 59.0, 71.0, 69.0, 63
.0, 79.0, 76.0, 63.0, 85.0, 87.0, 88.0, 80.0, 71.0, 65.0, 84.0, 71.0, 75.0
, 81.0, 79.0, 64.0, 65.0, 84.0, 77.0, 70.0, 75.0, 84.0, 75.0, 73.0, 92.0,
90.0, 79.0, 80.0, 71.0, 73.0, 71.0, 58.0, 79.0, 73.0, 64.0, 77.0, 82.0, 81
.0, 59.0, 54.0, 82.0, 57.0, 79.0, 79.0, 73.0, 74.0, 82.0, 63.0, 64.0, 73.0
, 69.0, 87.0, 68.0, 81.0, 73.0, 83.0, 73.0, 80.0, 73.0, 73.0, 71.0, 66.0,
78.0, 64.0, 74.0, 68.0, 67.0, 75.0, 75.0, 80.0, 85.0, 74.0, 76.0, 80.0, 77
.0, 93.0, 70.0, 86.0, 80.0, 81.0, 83.0, 68.0, 60.0, 85.0, 64.0, 74.0, 82.0
, 81.0, 77.0, 66.0, 85.0, 75.0, 81.0, 69.0, 60.0, 83.0, 72.0]

Please Enter number of bins:25
```

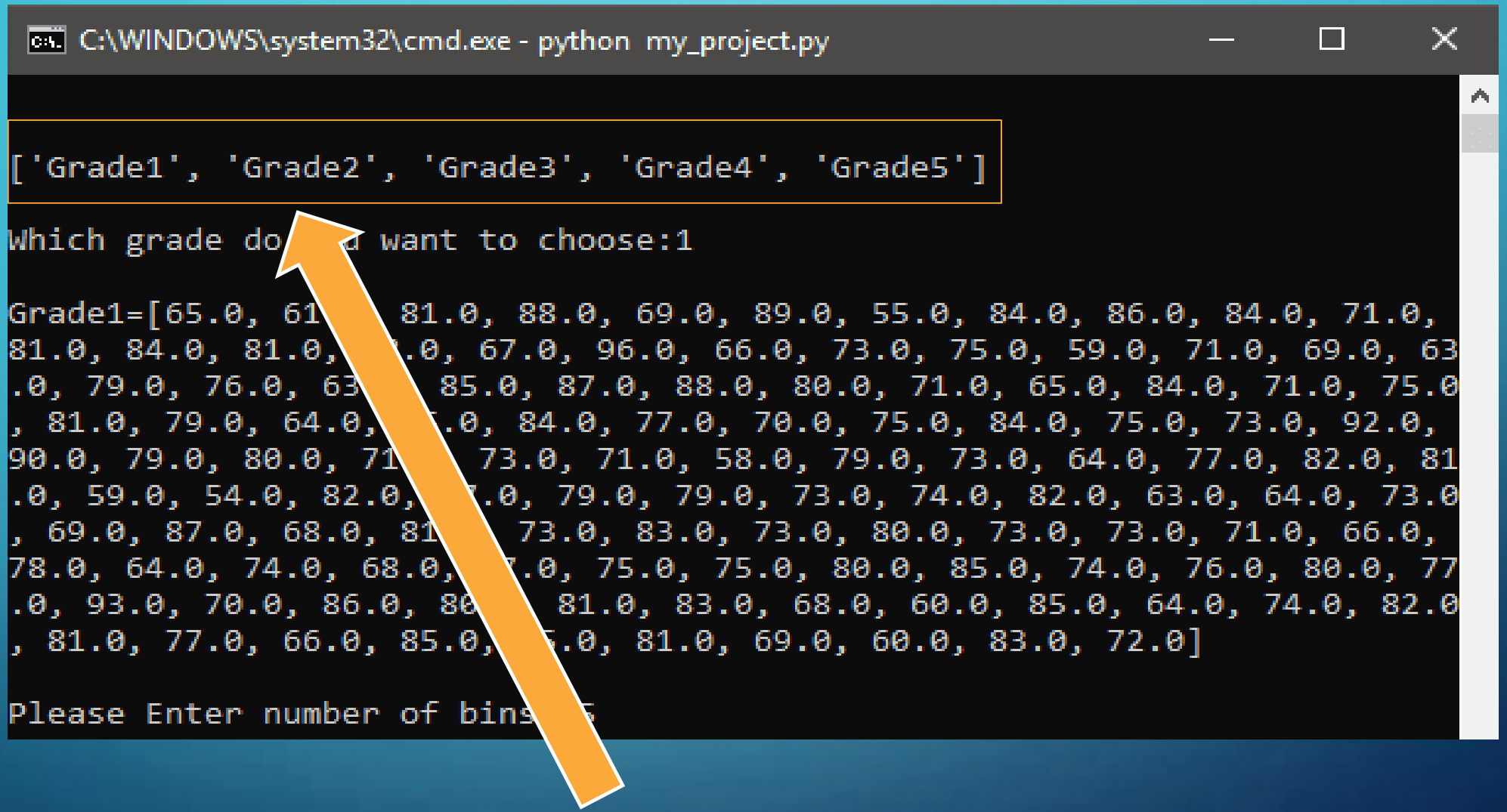
INFERIORITY OF OUR SOFTWARE





Uncapable to select default number of bins

FUTURE UPDATES



Input data yourself (not read it from excel file)

BOXPLOT

Statistical Software

Figure 1: Boxplot

File

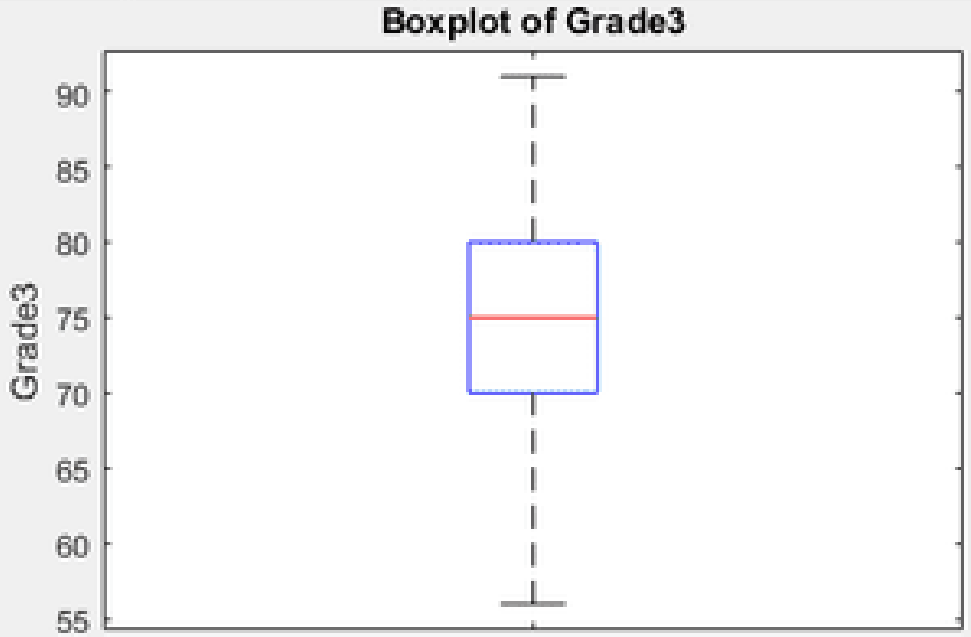
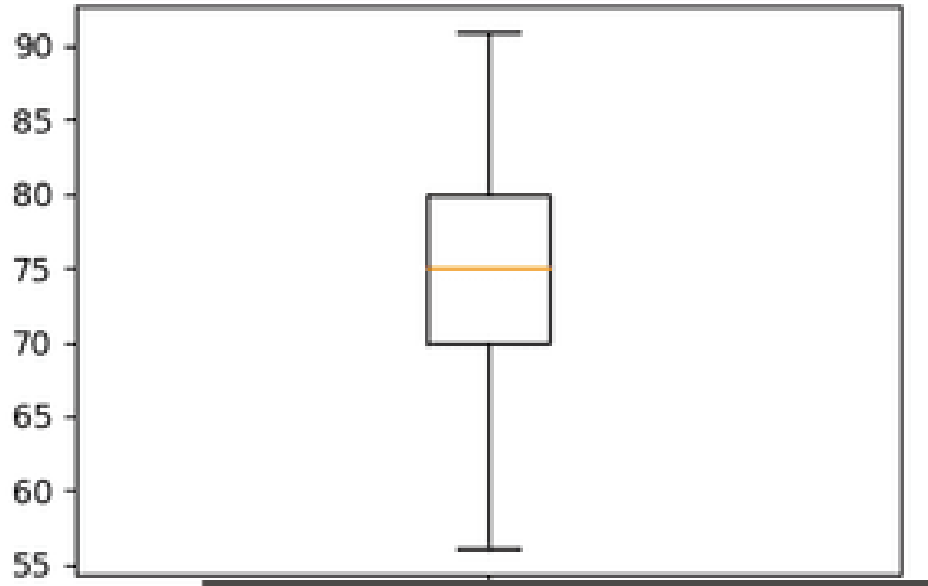


Figure 1



Boxplot

Variable: Grade3

OK | Cancel

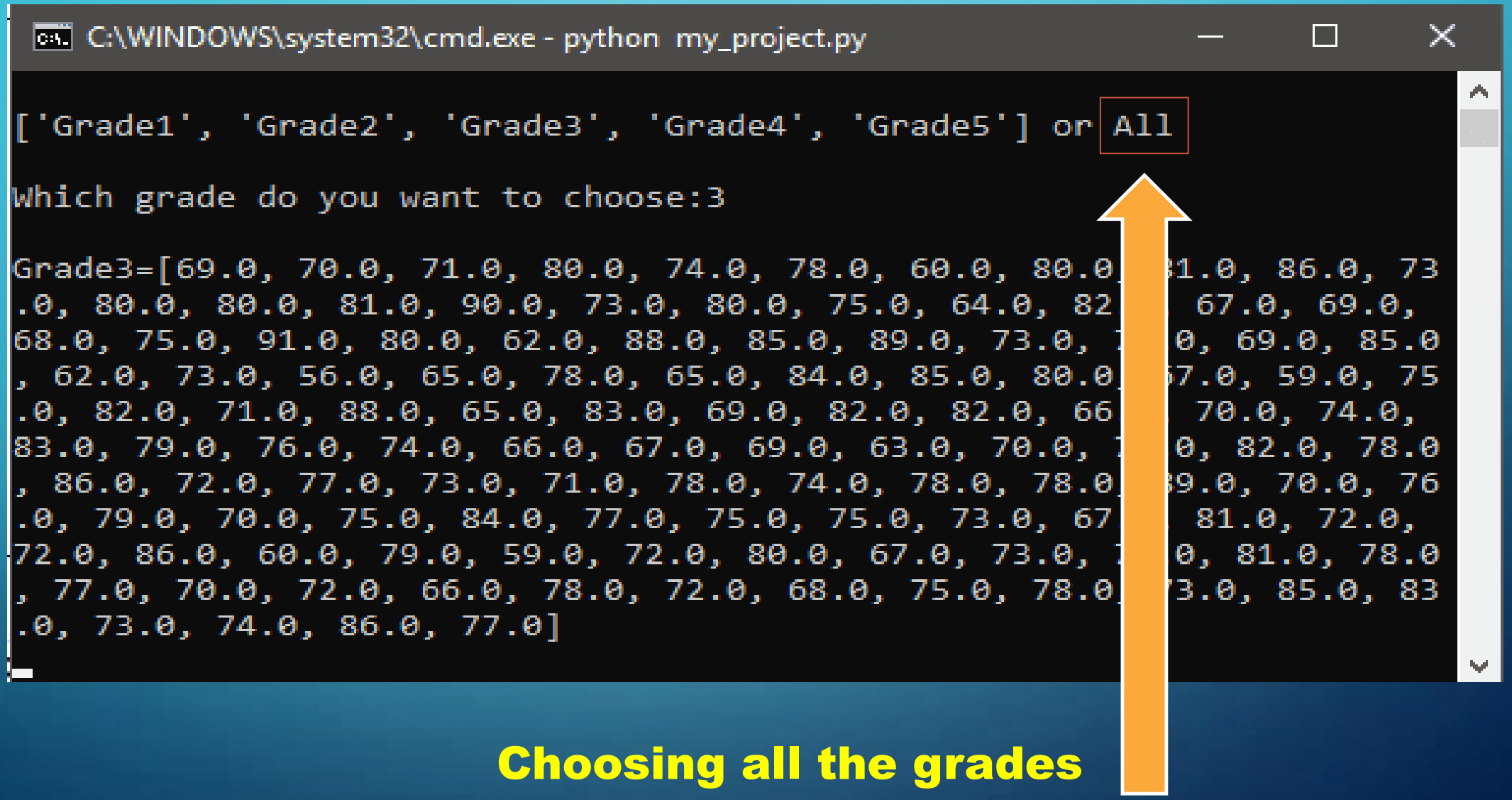
C:\WINDOWS\system32\cmd.exe - python my_project.py

```
['Grade1', 'Grade2', 'Grade3', 'Grade4', 'Grade5'] or All

Which grade do you want to choose:3

Grade3=[69.0, 70.0, 71.0, 80.0, 74.0, 78.0, 60.0, 80.0, 81.0, 86.0, 73
.0, 80.0, 80.0, 81.0, 90.0, 73.0, 80.0, 75.0, 64.0, 82.0, 67.0, 69.0,
68.0, 75.0, 91.0, 80.0, 62.0, 88.0, 85.0, 89.0, 73.0, 76.0, 69.0, 85.0
, 62.0, 73.0, 56.0, 65.0, 78.0, 65.0, 84.0, 85.0, 80.0, 67.0, 59.0, 75
.0, 82.0, 71.0, 88.0, 65.0, 83.0, 69.0, 82.0, 82.0, 66.0, 70.0, 74.0,
83.0, 79.0, 76.0, 74.0, 66.0, 67.0, 69.0, 63.0, 70.0, 71.0, 82.0, 78.0
, 86.0, 72.0, 77.0, 73.0, 71.0, 78.0, 74.0, 78.0, 78.0, 89.0, 70.0, 76
.0, 79.0, 70.0, 75.0, 84.0, 77.0, 75.0, 75.0, 73.0, 67.0, 81.0, 72.0,
72.0, 86.0, 60.0, 79.0, 59.0, 72.0, 80.0, 67.0, 73.0, 72.0, 81.0, 78.0
, 77.0, 70.0, 72.0, 66.0, 78.0, 72.0, 68.0, 75.0, 78.0, 73.0, 85.0, 83
.0, 73.0, 74.0, 86.0, 77.0]
```

Worksheet

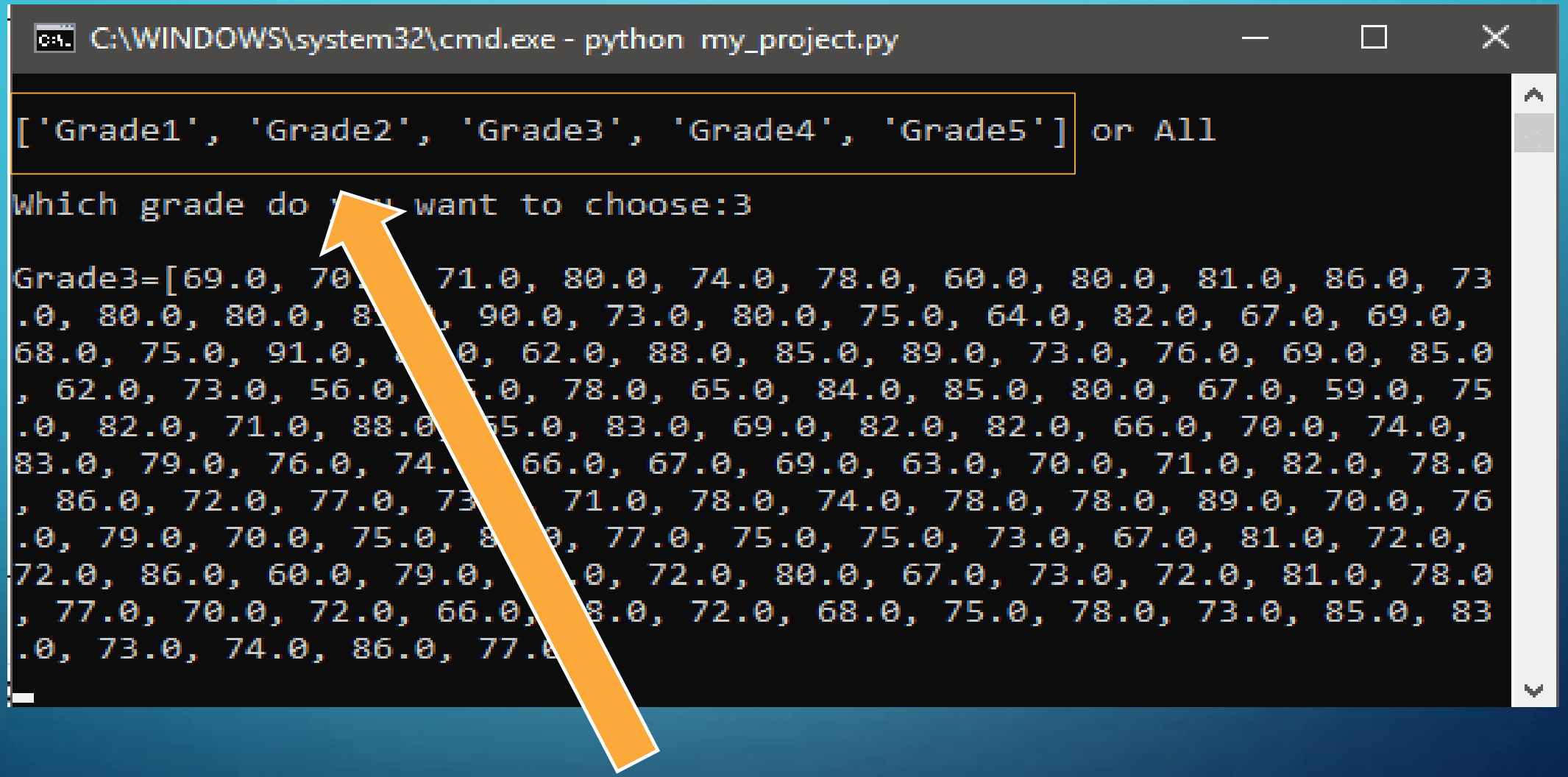
	Grade1	Grade2	Grade3	Grade4	Grade5
	65	77	69	75	69
	61	74	70	66	68
	81	80	71	74	79
	88	76	80	88	79
	69	77	74	69	76
	89	93	78	77	80

SUPERIORITY OF OUR SOFTWARE



Choosing all the grades

FUTURE UPDATES



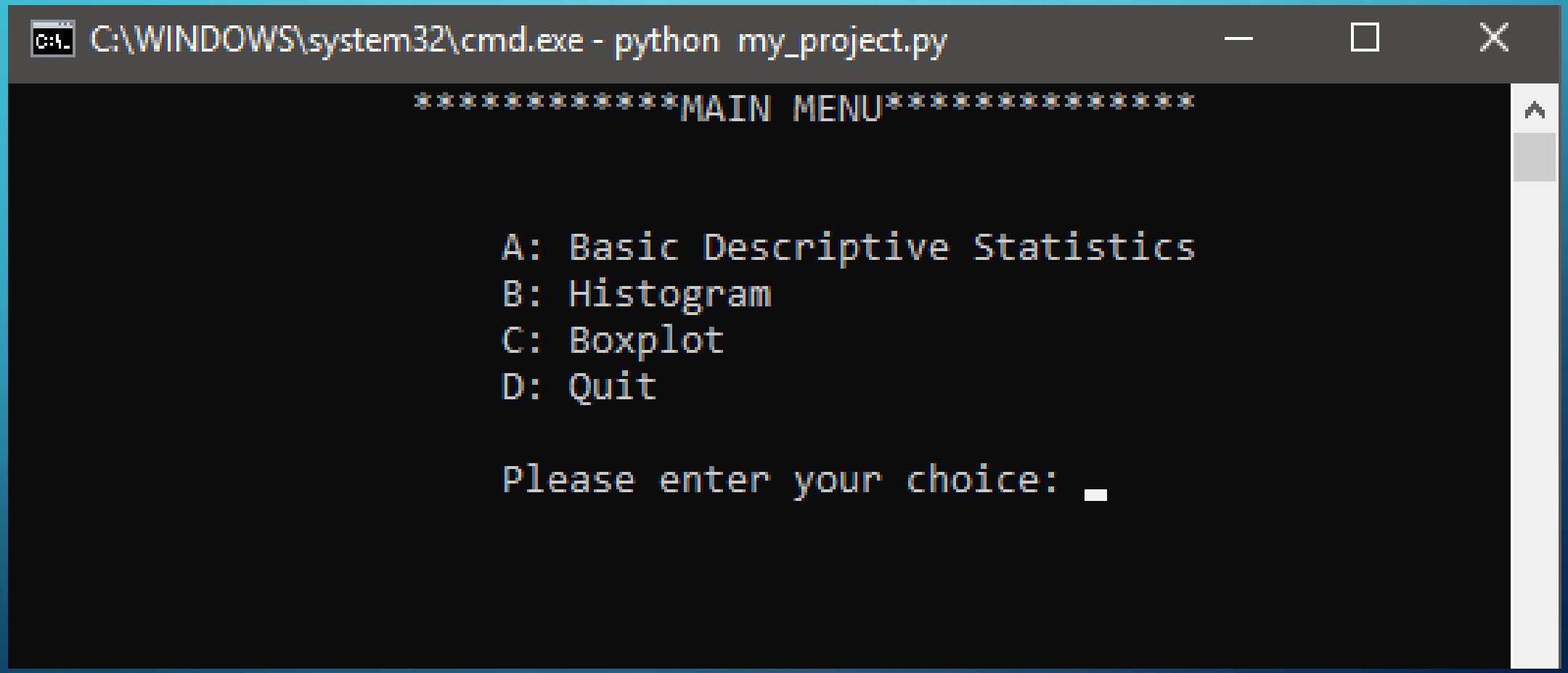
Input data yourself (not read it from excel file)

STATISTICAL TESTING

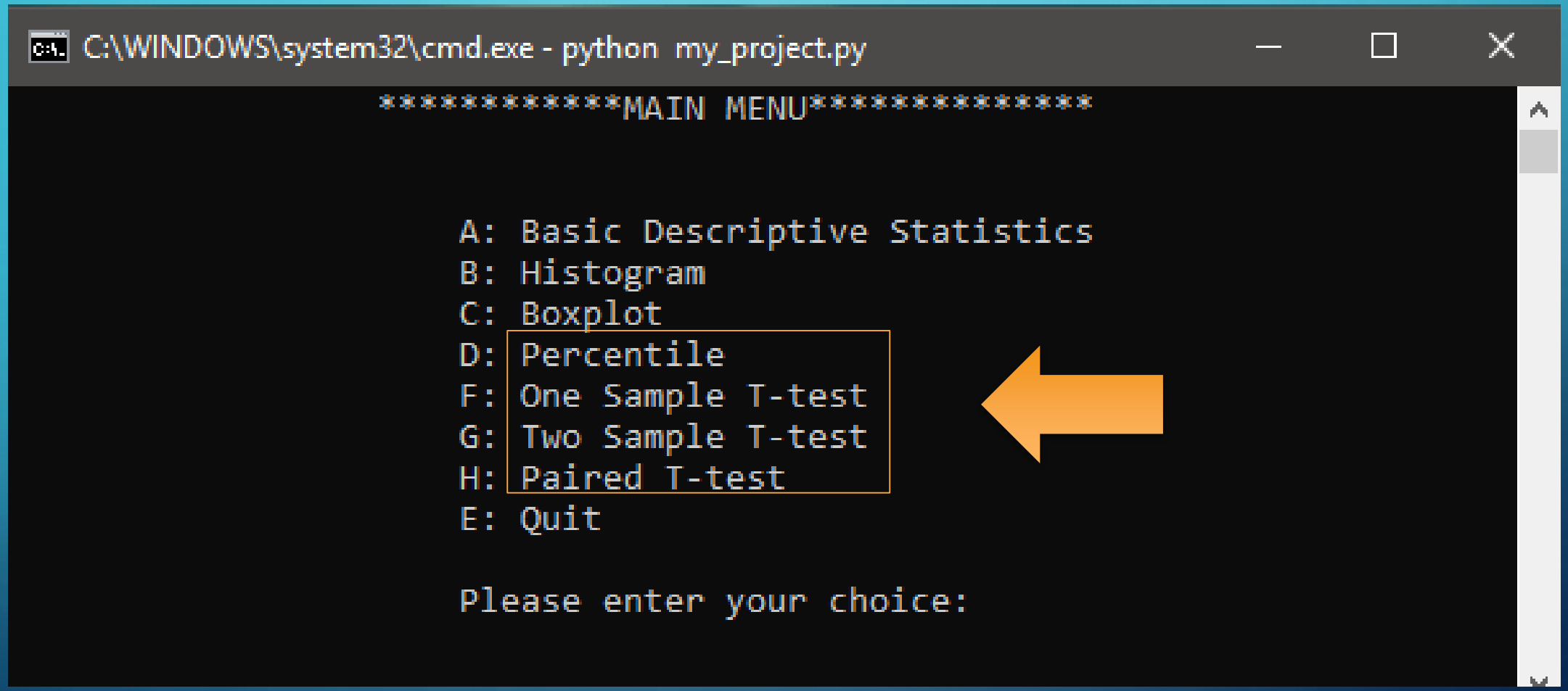
- One-sample t test
- Two-sample t test
- Paired t test

PYTHON

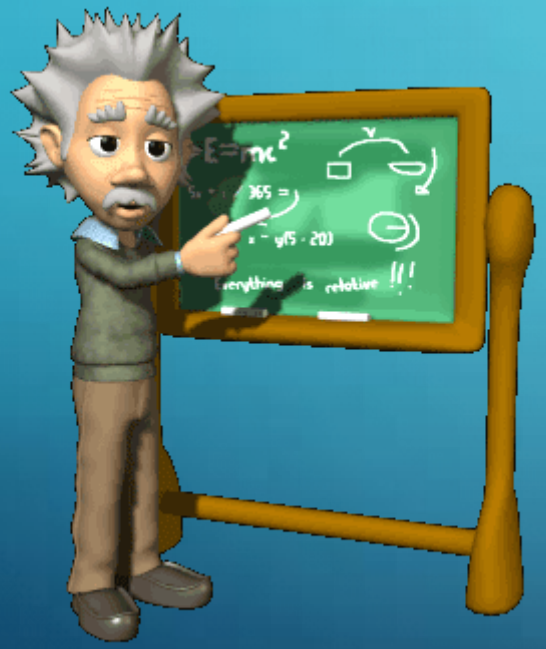
MINI PROJECT - 1



MINI PROJECT - 2



PROGRAMMING CODE



my_project.py

Functions.py

Two-sample *t* Test

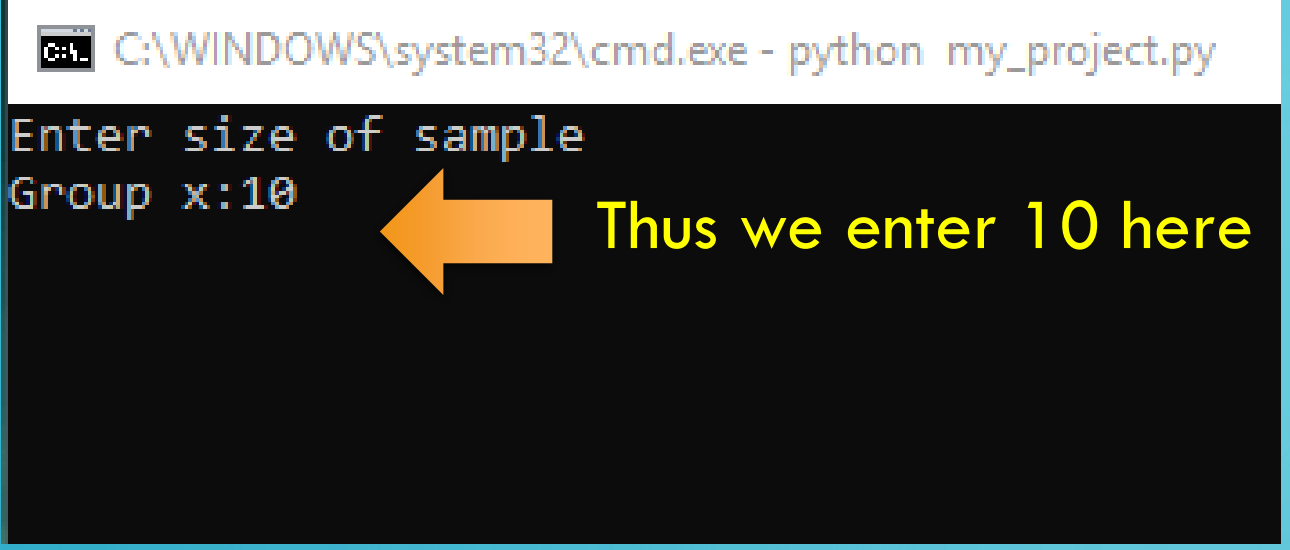
- Test whether means of two samples are the same or not.

- Example: An engineer want to test whether weight of products manufactured by two different machines is the same or not. Take two groups of sample (n_1 = 10, n_2=10) from the two machines.

  Machine X: 4, 6, 2, 3, 8, 9, 4, 3, 6, 5

  Machine Y: 7, 9, 5, 8, 7, 8, 8, 5, 6, 7

TWO-SAMPLE T TEST



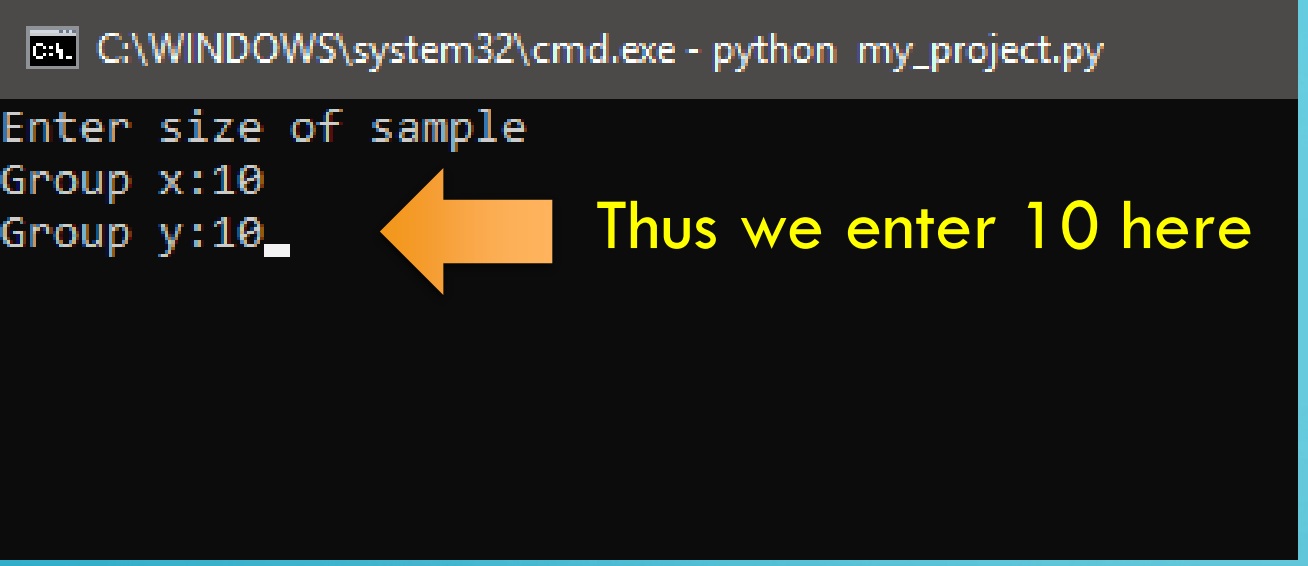
Thus we enter 10 here

Machine X: 4, 6, 2, 3, 8, 9, 4, 3, 6, 5

There are 10 data (sample size is 10)

PYTHON CODE

```
print("Enter size of sample")
sample1 = float(input("Group x:"))
```



Thus we enter 10 here

Machine Y: 7, 9, 5, 8, 7, 8, 8, 5, 6, 7

There are 10 data (sample size is 10)

PYTHON CODE

```
print("Enter size of sample")
sample1 = float(input("Group x:"))
sample2 = float(input("Group y:"))
```

C:\WINDOWS\system32\cmd.exe - python my_project.py

```
Enter size of sample
Group x:10
Group y:10

Group x
Data(1)=4
Data(2)=6
Data(3)=2
Data(4)=3
Data(5)=8
Data(6)=9
Data(7)=4
Data(8)=3
Data(9)=6
Data(10)=5
```

Machine X: 4, 6, 2, 3, 8, 9, 4, 3, 6, 5

We input all the data

```
list_sample1 = []
list_sample2 = []
i = 0
print("\nGroup x")
while i < sample1:
    list_sample1.append(i)
    list_sample1[i] = float(input("Data(" + str(i + 1) + ")" + "="))
    i = i + 1
```

```
C:\WINDOWS\system32\cmd.exe - python  my_project.py
Group y
Data(1)=7
Data(2)=9
Data(3)=5
Data(4)=8
Data(5)=7
Data(6)=8
Data(7)=8
Data(8)=5
Data(9)=6
Data(10)=7
```

Machine Y: 7, 9, 5, 8, 7, 8, 8, 5, 6, 7

We input all the data

```
I = 0
print("\nGroup y")
while I < sample2:
    list_sample2.append(I)
    list_sample2[I] = float(input("Data(" + str(I + 1) + ")" + "="))
    I = I + 1
```

C:\WINDOWS\system32\cmd.exe - python my_project.py

```
Group y
Data(1)=7
Data(2)=9
Data(3)=5
Data(4)=8
Data(5)=7
Data(6)=8
Data(7)=8
Data(8)=5
Data(9)=6
Data(10)=7

H0: Mx=My
Please choose alternative hypothesis
1) H1: Mx≠My (Two-tailed)
2) H1: Mx<My (Left-tailed)
3) H1: Mx>My (Right-Tailed)
Enter:1
```

- Step 1. Set hypothesis
  - Null: $H_0 : \mu_x = \mu_y$
  - Alternative:

$H_1 : \mu_x \neq \mu_y$	Two-tailed
$H_1 : \mu_x < \mu_y$	Left-tailed
$H_1 : \mu_x > \mu_y$	Right-tailed

```
print("\nH0: Mx=My")
print("Please choose alternative hypothesis")
print("1) H1: Mx≠My (Two-tailed)")
print("2) H1: Mx<My (Left-tailed)")
print("3) H1: Mx>My (Right-Tailed)")
alt_hypothesis = int(input("Enter:"))
```

```
H0: Mx=My
Please choose alternative hypothesis
1) H1: Mx≠My (Two-tailed)
2) H1: Mx<My (Left-tailed)
3) H1: Mx>My (Right-Tailed)
Enter:1

Assume equal variance
1)Yes
2)No
Enter:1
```

- Same (equal) variance: $\bar{x}-\bar{y}\sim N(\mu_x-\mu_y, \sigma^2_{\square}(\frac{1}{n}+\frac{1}{m}))$

$$T=\frac{(\bar{x}-\bar{y})-(\mu_x-\mu_y)}{\sqrt{s^2_{\square}(\frac{1}{n}+\frac{1}{m})}}\sim t(n+m-2) \qquad s^2_{\square}=\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2+\sum_{j=1}^{m}(y_j-\bar{y})^2}{n+m-2}$$

- Different (unequal) variance: $\bar{x}-\bar{y}\sim N(\mu_x-\mu_y, \frac{\sigma^2_x}{n}+\frac{\sigma^2_y}{m})$

$$T=\frac{(\bar{x}-\bar{y})-(\mu_x-\mu_y)}{\sqrt{\frac{s_x^2}{n}+\frac{s_y^2}{m}}}\sim t(\emptyset) \qquad \phi=\frac{(\frac{s_x^2}{n}+\frac{s_y^2}{m})^2}{\frac{(s_x^2/n)^2}{n-1}+\frac{(s_y^2/m)^2}{m-1}}$$

```
variance = float(input("\nAssume equal variance\n1)Yes\n2)No\nEnter:"))
```

C:\WINDOWS\system32\cmd.exe - python my_project.py

```
Group(x):[4.0, 6.0, 2.0, 3.0, 8.0, 9.0, 4.0, 3.0, 6.0, 5.0]
Group(y):[7.0, 9.0, 5.0, 8.0, 7.0, 8.0, 8.0, 5.0, 6.0, 7.0]
H0: Mx=My
H1: Mx≠My


Variable    Sample size     Mean     SD     Variance     SE Mean
Group(x):       10          5.00     2.26     5.11        0.71
Group(y):       10          7.00     1.33     1.78        0.42
Difference:                -2.00     0.93     3.33        0.29

Difference: Mu(Group(x)-Mu(Group(y))
Estimate for difference:-2.0

T value:-2.41
Degree of freedom:18
Significance level:0.05
t(18,0.025)=2.1


for T(-2.41)>t(2.1)/T(-2.41)<-t(-2.1)
H1(Mx≠My)rejected
H0(Mx=My) accepted


1.Again
2.Main menu
3.Quit
Enter:
```

```
import math
import statistics
from scipy import stats
from scipy.stats import sem
```

Libraries

```
variance = float(input("\nAssume equal variance\n1)Yes\n2)No\nEnter:"))
Xx = statistics.mean(list_sample1)
Xy = statistics.mean(list_sample2)
n = len(list_sample1)
m = len(list_sample2)
Sx = statistics.variance(list_sample1)
Sy = statistics.variance(list_sample2)
SDx = statistics.stdev(list_sample1)
SDy = statistics.stdev(list_sample2)
```

Some statistical functions

```
variance = float(input("\nAssume equal variance\n1)Yes\n2)No\nEnter:"))
Xx = statistics.mean(list_sample1)
Xy = statistics.mean(list_sample2)
n = len(list_sample1)
m = len(list_sample2)
Sx = statistics.variance(list_sample1)
Sy = statistics.variance(list_sample2)
SDx = statistics.stdev(list_sample1)
SDy = statistics.stdev(list_sample2)

Meandif = Xx - Xy
Variancedif = Sx - Sy
SDdif = SDx - SDy
SDerdif = sem(list_sample1) - sem(list_sample2)
```

Finding differences

```
variance = float(input("\nAssume equal variance\n1)Yes\n2)No\nEnter:"))
```

```
if variance == 1:
    Sx1 = 0
    for i in list_sample1:
        Sx1 = pow((i - Xx), 2) + Sx1
    Sx2 = 0
    for x in list_sample2:
        Sx2 = pow((x - Xy), 2) + Sx2
    S = (Sx1 + Sx2) / ((n + m) - 2)
    T = (Xx - Xy) / math.sqrt(S * (1 / n + 1 / m))
    o = n + m - 2
    a = 0.05
    t = abs(float("%.2f" % stats.t.ppf(a, o)))
```

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2 + \sum_{j=1}^{m}(y_j - \bar{y})^2}{n + m - 2}$$

$$T = \frac{(\bar{x} - \bar{y}) - (\mu_x - \mu_y)}{\sqrt{s^2\left(\frac{1}{n} + \frac{1}{m}\right)}}$$

$$\emptyset = n + m - 2$$

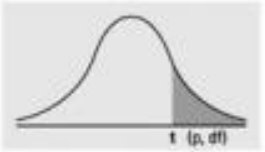
5% significance level (α)

- Two-tailed: $T > t(\phi, \alpha/2)$ or $T < -t(\phi, \alpha/2)$
- Left-tailed: $T < -t(\phi, \alpha)$
- Right-tailed: $T > t(\phi, \alpha)$

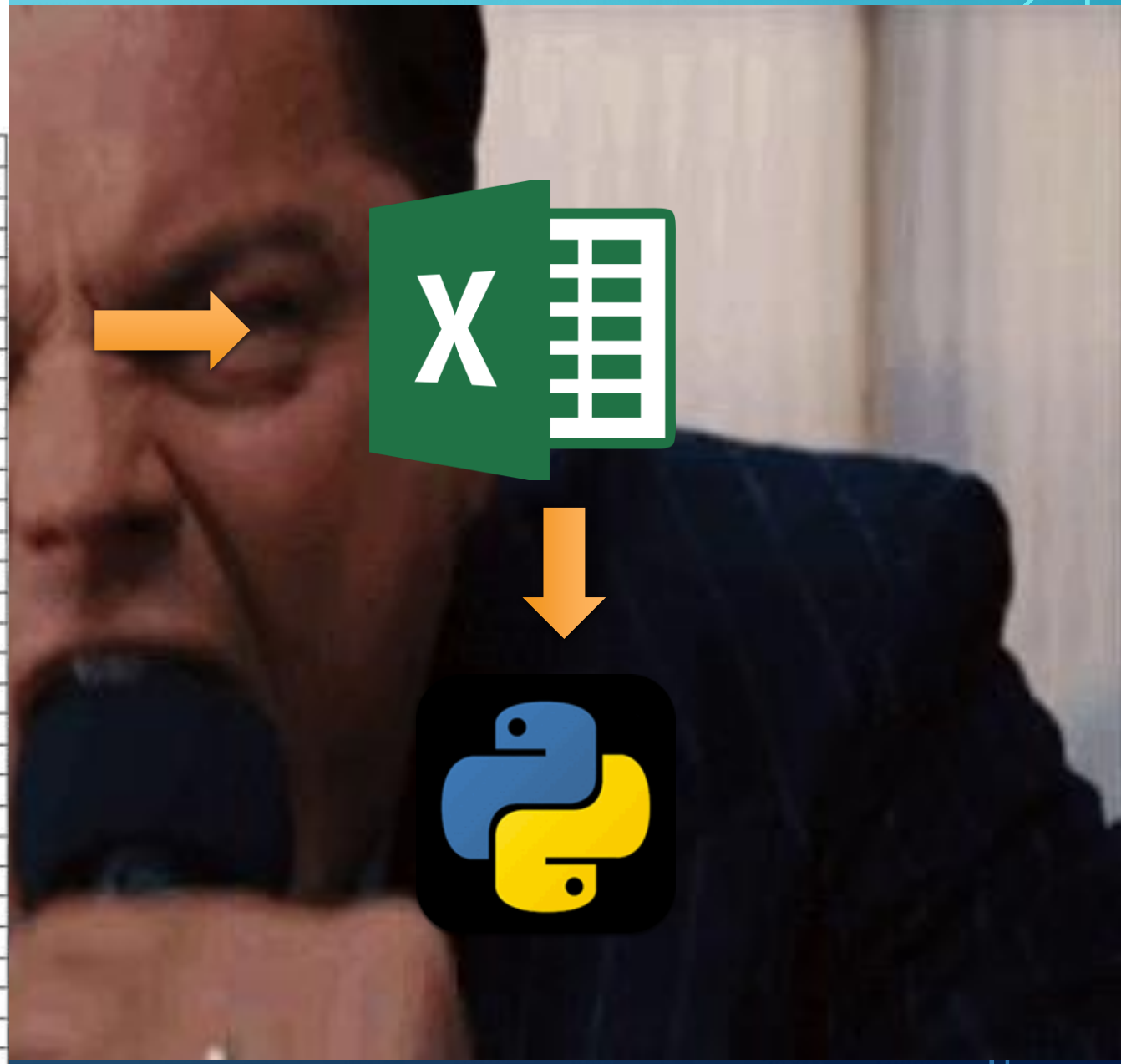
} Reject H_0 otherwise accept H_0

t = ?

Numbers in each row of the table are values on a *t*-distribution with (*df*) degrees of freedom for selected right-tail (greater-than) probabilities (*p*).



df/p	0.40	0.25	0.10	0.05	0.025	0.01	0.005	0.0005
1	0.324920	1.000000	3.077684	6.313752	12.70620	31.82052	63.65674	636.6192
2	0.288675	0.816497	1.885618	2.919986	4.30265	6.96456	9.92484	31.5991
3	0.276671	0.764892	1.637744	2.353363	3.18245	4.54070	5.84091	12.9240
4	0.270722	0.740697	1.533206	2.131847	2.77645	3.74695	4.60409	8.6103
5	0.267181	0.726687	1.475884	2.015048	2.57058	3.36493	4.03214	6.8688
6	0.264835	0.717558	1.439756	1.943180	2.44691	3.14267	3.70743	5.9588
7	0.263167	0.711142	1.414924	1.894579	2.36462	2.99795	3.49948	5.4079
8	0.261921	0.706387	1.396815	1.859548	2.30600	2.89646	3.35539	5.0413
9	0.260955	0.702722	1.383029	1.833113	2.26216	2.82144	3.24984	4.7809
10	0.260185	0.699812	1.372184	1.812461	2.22814	2.76377	3.16927	4.5869
11	0.259556	0.697445	1.363430	1.795885	2.20099	2.71808	3.10581	4.4370
12	0.259033	0.695483	1.356217	1.782288	2.17881	2.68100	3.05454	43178
13	0.258591	0.693829	1.350171	1.770933	2.16037	2.65031	3.01228	4.2208
14	0.258213	0.692417	1.345030	1.761310	2.14479	2.62449	2.97684	4.1405
15	0.257885	0.691197	1.340606	1.753050	2.13145	2.60248	2.94671	4.0728
16	0.257599	0.690132	1.336757	1.745884	2.11991	2.58349	2.92078	4.0150
17	0.257347	0.689195	1.333379	1.739607	2.10982	2.56693	2.89823	3.9651
18	0.257123	0.688364	1.330391	1.734064	2.10092	2.55238	2.87844	3.9216
19	0.256923	0.687621	1.327728	1.729133	2.09302	2.53948	2.86093	3.8834
20	0.256743	0.686954	1.325341	1.724718	2.08596	2.52798	2.84534	3.8495
21	0.256580	0.686352	1.323188	1.720743	2.07961	2.51765	2.83136	3.8193
22	0.256432	0.685805	1.321237	1.717144	2.07387	2.50832	2.81876	3.7921
23	0.256297	0.685306	1.319460	1.713872	2.06866	2.49987	2.80734	3.7676
24	0.256173	0.684850	1.317836	1.710882	2.06390	2.49216	2.79694	3.7454
25	0.256060	0.684430	1.316345	1.708141	2.05954	2.48511	2.78744	3.7251
26	0.255955	0.684043	1.314972	1.705618	2.05553	2.47863	2.77871	3.7066
27	0.255858	0.683685	1.313703	1.703288	2.05183	2.47266	2.77068	3.6896
28	0.255768	0.683353	1.312527	1.701131	2.04841	2.46714	2.76326	3.6739
29	0.255684	0.683044	1.311434	1.699127	2.04523	2.46202	2.75639	3.6594
30	0.255605	0.682756	1.310415	1.697261	2.04227	2.45726	2.75000	3.6460
z	0.253347	0.674490	1.281552	1.644854	1.95996	2.32635	2.57583	3.2905
CI	———	———	80%	90%	95%	98%	99%	99.9%



```
if variance == 1:
    Sx1 = 0
    for i in list_sample1:
        Sx1 = pow((i - Xx), 2) + Sx1
    Sx2 = 0
    for x in list_sample2:
        Sx2 = pow((x - Xy), 2) + Sx2
    S = (Sx1 + Sx2) / ((n + m) - 2)
    T = (Xx - Xy) / math.sqrt(S * (1 / n + 1 / m))
    o = n + m - 2
    a = 0.05
    t = abs(float("%.2f" % stats.t.ppf(a, o)))
```

t value function (scipy)

```
print("\nH0: Mx=My")
print("Please choose alternative hypothesis")
print("1) H1: Mx≠My (Two-tailed)")
print("2) H1: Mx<My (Left-tailed)")
print("3) H1: Mx>My (Right-Tailed)")
alt_hypothesis = int(input("Enter:"))
```

```
if alt_hypothesis == 1:
    clear()
    print("Group(x):" + str(list_sample1))
    print("Group(y):" + str(list_sample2))
    print("H0: Mx=My")
    print("H1: Mx≠My")
```

```
C:\WINDOWS\system32\cmd.exe - python my_project.py
Group(x):[4.0, 6.0, 2.0, 3.0, 8.0, 9.0, 4.0, 3.0, 6.0, 5.0]
Group(y):[7.0, 9.0, 5.0, 8.0, 7.0, 8.0, 8.0, 5.0, 6.0, 7.0]
H0: Mx=My
H1: Mx≠My
```

```
for T(-2.41)>t(2.1)/T(-2.41)<-t(-2.1)
H1(Mx≠My)rejected
H0(Mx=My) accepted
```

```
a = a / 2
t = abs(float("%.2f" % stats.t.ppf(a, o)))
print("t(" + str(o) + "," + str(a) + ")=" + str(t))
print("\n")
if T > t and T < -t:
    print("for T(" + str("%.2f" % T) + ")" + ">" + "t(" + str(t) + ")" + "/" + "T(" + str(
        "%.2f" % T) + ")" + "<" + "-t(" + str(-t) + ")")
    print("H0(Mx=My)rejected" + "\nH1(Mx≠My)accepted")
else:
    print("for T(" + str("%.2f" % T) + ")" + ">" + "t(" + str(t) + ")" + "/" + "T(" + str(
        "%.2f" % T) + ")" + "<" + "-t(" + str(-t) + ")")
    print("H1(Mx≠My)rejected\nH0(Mx=My) accepted")
```

```
elif alt_hypothesis == 2:
```

```
if T < -t:
    print("for T(" + str("%.2f" % T) + ")" + "<" + "-t(" + str(-t) + ")")
    print("H0(Mx=My)rejected\nH1(Mx<My)accepted")
else:
    print("for T(" + str("%.2f" % T) + ")" + "<" + "-t(" + str(-t) + ")")
    print("H1(Mx<My)rejected\nH0(Mx=My)accepted")
```

```
elif alt_hypothesis == 3:
```

```
if T > t:
    print("for T(" + str("%.2f" % T) + ")" + ">" + "t(" + str(t) + ")")
    print("H0(Mx=My)rejected" + "\nH1(Mx>My)accepted")
else:
    print("for T(" + str("%.2f" % T) + ")" + ">" + "t(" + str(t) + ")")
    print("H1(Mx>My)rejected\nH0(Mx=My)accepted")
```

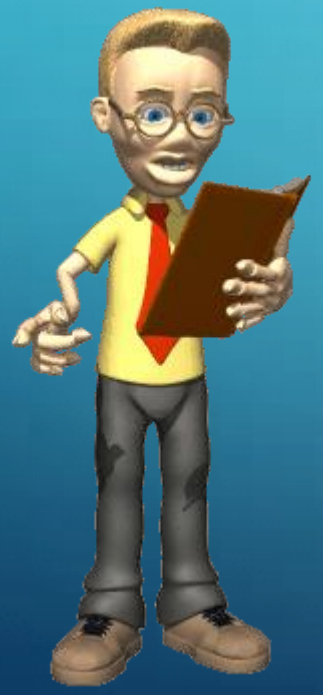
```
variance = float(input("\nAssume equal variance\n1)Yes\n2)No\nEnter:"))
```

```
elif variance == 2:

    T = (Xx - Xy) / math.sqrt(Sx / n + Sy / m)
    o = int(pow((Sx / n + Sy / m), 2) / (pow((Sx / n), 2) / (n - 1) + pow((Sy / n), 2) / (m - 1)))
    a = 0.05
    t = abs(float("%.2f" % stats.t.ppf(a, o)))
```

$$\phi = \frac{(\frac{s_x^2}{n} + \frac{s_y^2}{m})^2}{\frac{(s_x^2/n)^2}{n-1} + \frac{(s_y^2/m)^2}{m-1}}$$

$$T = \frac{(\bar{x} - \bar{y}) - (\mu_x - \mu_y)}{\sqrt{\frac{s_x^2}{n} + \frac{s_y^2}{m}}}$$

PAIRED T TEST



```
Enter size of sample
Group x:10
Group y:10

Group x
Data(1)=4
Data(2)=6
Data(3)=2
Data(4)=3
Data(5)=8
Data(6)=9
Data(7)=4
Data(8)=3
Data(9)=6
Data(10)=5

Group y
Data(1)=7
Data(2)=9
Data(3)=5
Data(4)=8
Data(5)=7
Data(6)=8
Data(7)=8
Data(8)=5
Data(9)=6
Data(10)=7

H0: Md=0
Please choose alternative hypothesis
1) H1: Md≠0 (Two-tailed)
2) H1: Md<0 (Left-tailed)
3) H1: Md>0 (Right-Tailed)
Enter:2
```

C:\WINDOWS\system32\cmd.exe - python my_project.py

```
Group(x-y):[-3.0, -3.0, -3.0, -5.0, 1.0, 1.0, -4.0, -2.0, 0.0, -2.0]
H0: Md=0
H1: Md<0
Variable    Sample size      Mean      SD      Variance      SE Mean
Group(x-y):      10          -2.00      2.05      4.22          0.65

T value:-3.08
Degree of freedom:9
Significance level:0.05
t(9,0.05)=1.83


for T(-3.08)<-t(-1.83)
H0(Md=0)rejected
H1(Md<0)accepted


1.Again
2.Main menu
3.Quit
Enter:
```

- Step 1. Set hypothesis
  - Null: $H_0 : \mu_d = 0$
  - Alternative:

$H_1 : \mu_d \neq 0$	Two-tailed
$H_1 : \mu_d < 0$	Left-tailed
$H_1 : \mu_d > 0$	Right-tailed

```
print("\nH0: Md=0")
print("Please choose alternative hypothesis")
print("1) H1: Md≠0 (Two-tailed)")
print("2) H1: Md<0 (Left-tailed)")
print("3) H1: Md>0 (Right-Tailed)")
alt_hypothesis = int(input("Enter:"))
```

THANK YOU FOR YOUR ATTENTION