PROJECT 2 DATA

PART A:

STEP 1)

- I am making a table for probability mass function. For this table, I am utilizing with Python and I am sharing source code of the table function. First of all, womens' data will be shown and than, mans' data will be shown.

```
In [6]: import pandas as pd

        # Kadınlar için olasılık kütle fonksiyonu tablosu
        womenTable = pd.DataFrame({
            'Height (inch)': ['< 58', '58-59', '59-60', '60-61', '61-62', '62-63', '63-64', '64-65', '65-66', '66-67', '67-68', '68-69',
            'Probability Mass Function (PMS)': [0.0422, 0.0674, 0.0750, 0.0769, 0.0792, 0.0820, 0.0818, 0.0843, 0.0861, 0.0879, 0.0873, 0
        })

        # Erkekler için olasılık kütle fonksiyonu tablosu
        manTable = pd.DataFrame({
            'Height (inch)': ['< 58', '58-59', '59-60', '60-61', '61-62', '62-63', '63-64', '64-65', '65-66', '66-67', '67-68', '68-69',
            'Probability Mass Function (PMS)': [0.0158, 0.0269, 0.0296, 0.0324, 0.0350, 0.0379, 0.0403, 0.0424, 0.0444, 0.0458, 0.0457, 0
        })

        print("Probability Mass Function Table Based on Women's Data: ")
        print(womenTable)

        print("\nProbability Mass Function Table Based on Man Data: ")
        print(manTable)
```

```
Probability Mass Function Table Based on Women's Data:
   Height (inch)  Probability Mass Function (PMS)
0           < 58                          0.0422
1          58-59                          0.0674
2          59-60                          0.0750
3          60-61                          0.0769
4          61-62                          0.0792
5          62-63                          0.0820
6          63-64                          0.0818
7          64-65                          0.0843
8          65-66                          0.0861
9          66-67                          0.0879
10         67-68                          0.0873
11         68-69                          0.0817
12         69-70                          0.0702
13          > 70                          0.0599
```

```
Probability Mass Function Table Based on Man Data:
   Height (inch)  Probability Mass Function (PMS)
0           < 58                          0.0158
1          58-59                          0.0269
2          59-60                          0.0296
3          60-61                          0.0324
4          61-62                          0.0350
5          62-63                          0.0379
6          63-64                          0.0403
7          64-65                          0.0424
8          65-66                          0.0444
9          66-67                          0.0458
10         67-68                          0.0457
11         68-69                          0.0457
12         69-70                          0.0447
13          > 70                          0.0420
```

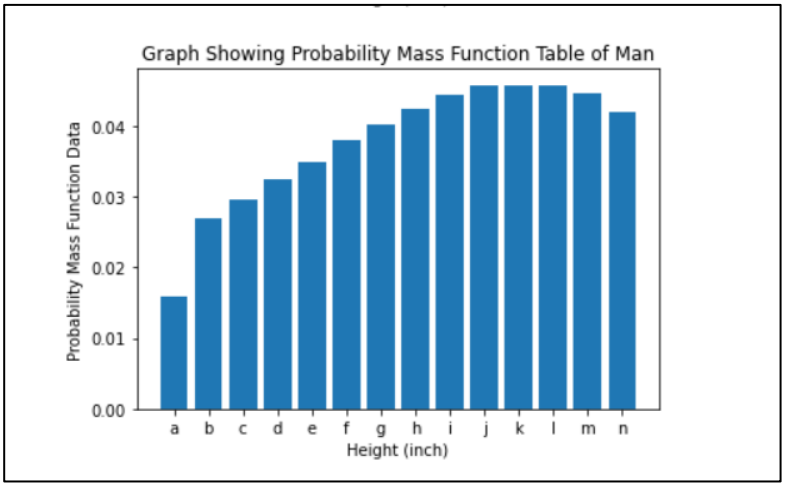
STEP 2)

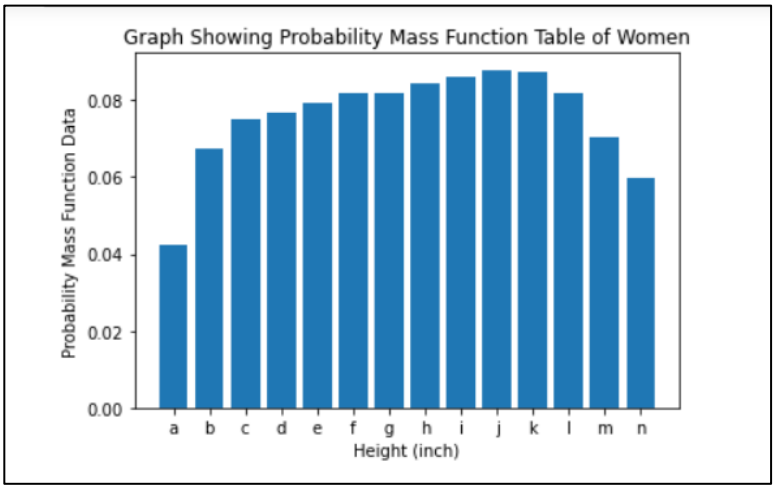
- I'm plotting the probability mass functions of men and women on separate graphs. I am using pyhton for this graphs. Here are the snapshots of the codes:

```
In [3]: #First, we import the library to be able to draw the graph.
        import matplotlib.pyplot as plt

        #Since we created the Probability Mass Function (PMS) table for women and men separately,
        #we will graph both separately. First we enter the women's data, then the men's data.
        #We enter the height and probability mass function data for women one by one.

        """
        Since the graph is narrow, the numbers appear to be mixed. I created characters to describe each age range.
        < 58 = a
        58-59 = b
        59-60 = c
        60-61 = d
        61-62 = e
        62-63 = f
        63-64 = g
        64-65 = h
        65-66 = i
        66-67 = j
        67-68 = k
        68-69 = l
        69-70 = m
        > 70 = n

        """

        wHeight = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n']
        wProbability = [0.0422, 0.0674, 0.0750, 0.0769, 0.0792, 0.0820, 0.0818, 0.0843, 0.0861, 0.0879, 0.0873, 0.0817, 0.0702, 0.0599]

        #We enter the height and probability mass function data for men one by one.
        mBoy = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n']
        mProbability = [0.0158, 0.0269, 0.0296, 0.0324, 0.0350, 0.0379, 0.0403, 0.0424, 0.0444, 0.0458, 0.0457, 0.0457, 0.0447, 0.0420]
```

```
#We are creating a bar chart based on women's data.
plt.bar(wHeight, wProbability)
plt.xlabel('Height (inch)')
plt.ylabel('Probability Mass Function Data')
plt.title('Graph Showing Probability Mass Function Table of Women')
plt.show()

#We are creating a bar chart based on man data.
plt.bar(mBoy, mProbability)
plt.xlabel('Height (inch)')
plt.ylabel('Probability Mass Function Data')
plt.title('Graph Showing Probability Mass Function Table of Man')
plt.show()
```

After showing the source codes, lets show graphs. While left side is showing the PMF of Man, right side is showing the PMS of Women





STEP 3)

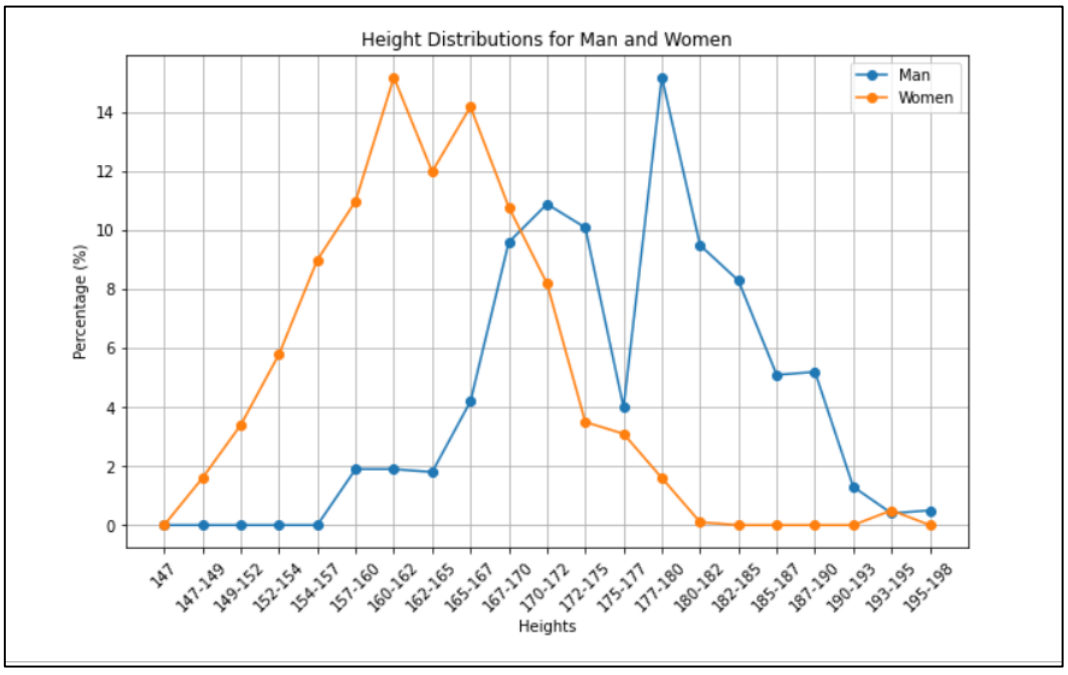
- Below are the average heights for men and women according to age distributions. I calculated separately for each column value. Expected values are shown below:

25	Age Distributions	Male	Female
26	20-29	175.78	164.52
27	30-39	174.88	162.79
28	40-49	158.45	163.20
29	50-59	158.39	161.74
30	60-69	165.63	161.40
31	70-79	172.74	158.92

STEP 4)

- I am using python to draw this graph. First, I define the height ranges and enter the male and female percentiles corresponding to the heights. I will show the code first and then the graph. I am defining the code for 40-49 aged both male and female.

```
In [10]: import matplotlib.pyplot as plt

         # Verileri tanımlama
         boyAraliklari = ['147', '147-149', '149-152', '152-154', '154-157', '157-160', '160-162', '162-165',
                          '165-167', '167-170', '170-172', '172-175', '175-177', '177-180', '180-182', '182-185',
                          '185-187', '187-190', '190-193', '193-195', '195-198']

         erkekYuzdelikler = [0, 0, 0, 0, 0, 1.9,1.9,1.8,4.2,9.6,10.9,10.1,4,15.2,9.5,8.3,5.1,5.2,1.3,0.4,0.5]
         print(len(erkekYuzdelikler))
         kadinYuzdelikler = [0, 1.6, 3.4, 5.8, 9, 11, 15.2, 12, 14.2, 10.8, 8.2, 3.5, 3.1, 1.6, 0.1, 0, 0, 0, 0, 0.5, 0]
         print(len(kadinYuzdelikler))

         # Grafik oluşturma
         plt.figure(figsize=(10, 6))
         plt.plot(boyAraliklari, erkekYuzdelikler, label='Man', marker='o')
         plt.plot(boyAraliklari, kadinYuzdelikler, label='Women', marker='o')
         plt.title('Height Distributions for Man and Women')
         plt.xlabel('Heights')
         plt.ylabel('Percentage (%)')
         plt.legend()
         plt.xticks(rotation=45)
         plt.grid()
         plt.show()
```



- To use the GraphPad T test calculator after drawing the graph, the values we need are the mean and standard deviation values. Therefore, for this, I first use a fixed parameter since the height values of men and women are the same. Then I calculate by entering the percentages of men and women. Finally, I enter the system with the N value along with the standard deviation and mean values I found and calculate the p-value.

```
In [12]: import math

sabitBoy = [147, 148, 150.5, 153, 155.5, 158.5, 161, 163.5, 166, 168.5, 171, 173.5, 176, 178.5, 181, 183.5, 186, 188.5, 191.5, 19

# Kadınlar
kadınYüzdelikler = [0, 1.6, 3.4, 5.8, 9, 11, 15.2, 12, 14.2, 10.8, 8.2, 3.5, 3.1, 1.6, 0.1, 0, 0, 0, 0.5, 0]
#print(len(kadınYüzdelikler))
kadınOrtalama = sum([ort * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, kadınYüzdelikler)])
kadınStandartSapma = math.sqrt(sum([((ort - kadınOrtalama) ** 2) * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, kadınYüzdeli

# Erkekler
erkekYüzdelikler = [0, 0, 0, 0, 0, 1.9, 1.9,1.8 ,4.2, 9.6, 10.9, 10.1, 4, 15.2, 9.5, 8.3, 5.1, 5.2, 1.3, 0.4, 0.5]
#print(len(erkekYüzdelikler))
erkekOrtalama = sum([ort * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, erkekYüzdelikler)])
erkekStandartSapma = math.sqrt(sum([((ort - erkekOrtalama) ** 2) * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, erkekYüzdeli


# Sonuçları yazdırma
print("Kadınlar:")
print("Ortalama (Mean): {:.2f}".format(kadınOrtalama))
print("Standart Sapma: {:.2f}".format(kadınStandartSapma))
print("Erkekler:")
print("Ortalama (Mean): {:.2f}".format(erkekOrtalama))
print("Standart Sapma: {:.2f}".format(erkekStandartSapma))
```

```
Kadınlar:
Ortalama (Mean): 163.21
Standart Sapma: 7.10
Erkekler:
Ortalama (Mean): 158.46
Standart Sapma: 18.40
```

- I enter the system by changing the N values. N values must be 10,100,1000.

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value equals 0.4515

By conventional criteria, this difference is considered to be not statistically significant.

Confidence interval:

The mean of Male 40-49 minus Female 40-49 equals -4.8000

95% confidence interval of this difference: From -17.9029 to 8.3029

Intermediate values used in calculations:

t = 0.7696

df = 18

standard error of difference = 6.237

Review your data:

Group	Male 40-49	Female 40-49
Mean	158.4000	163.2000
SD	18.4000	7.1000
SEM	5.8186	2.2452
N	10	10

N = 10 → P value equals 0.4515

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value equals 0.0158

By conventional criteria, this difference is considered to be statistically significant.

Confidence interval:

The mean of Male 40-49 minus Female 40-49 equals -4.8000

95% confidence interval of this difference: From -8.6893 to -0.9107

Intermediate values used in calculations:

t = 2.4338

df = 198

standard error of difference = 1.972

Review your data:

Group	Male 40-49	Female 40-49
Mean	158.4000	163.2000
SD	18.4000	7.1000
SEM	1.8400	0.7100
N	100	100

N = 100 → P value equals 0.0158

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value is less than 0.0001

By conventional criteria, this difference is considered to be extremely statistically significant.

Confidence interval:

The mean of Male 40-49 minus Female 40-49 equals -4.8000

95% confidence interval of this difference: From -6.0253 to -3.5747

Intermediate values used in calculations:

t = 7.6963

df = 1998

standard error of difference = 0.624

Review your data:

Group	Male 40-49	Female 40-49
Mean	158.4000	163.2000
SD	18.4000	7.1000
SEM	0.5819	0.2245
N	1000	1000

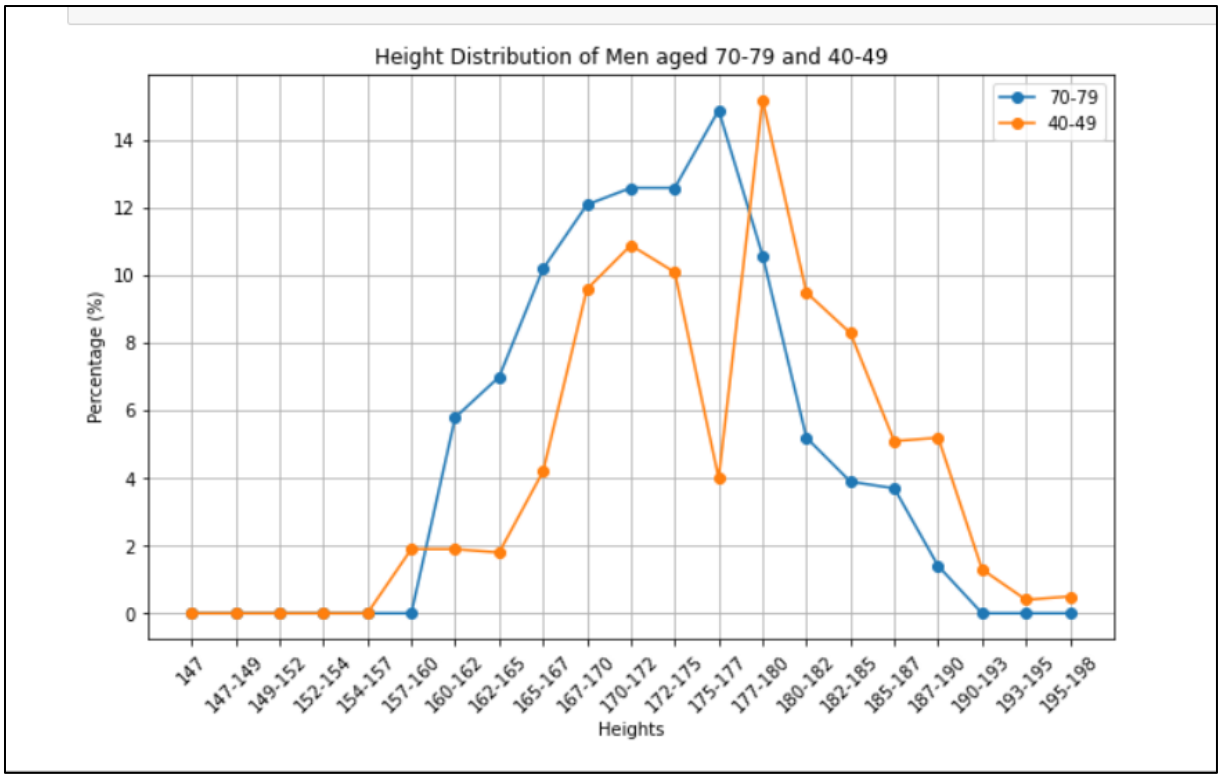
N = 1000 → P = 0.0001

STEP 5)

- I'm using python to graph men aged 40-49 and 70-79. For this, I first define fixed height ranges for men, and then indicate the percentages of men in the 40-49 and 70-79 age ranges.

```
In [5]: import matplotlib.pyplot as plt

boyAraliklari = ['147', '147-149', '149-152', '152-154', '154-157', '157-160', '160-162', '162-165',
                 '165-167', '167-170', '170-172', '172-175', '175-177', '177-180', '180-182', '182-185',
                 '185-187', '187-190', '190-193', '193-195', '195-198']

# 70-79 yaş aralığındaki erkeklerin dağılımı
firstPercentages = [0, 0,0,0,0,0,5.8,7,10.2,12.1,12.6,12.6,14.9,10.6,5.2,3.9,3.7,1.4,0,0,0]

# 40-49 yaş aralığındaki erkeklerin dağılımı
secondPercentages = [0, 0, 0, 0, 0, 1.9, 1.9,1.8 ,4.2, 9.6, 10.9, 10.1, 4, 15.2, 9.5, 8.3, 5.1, 5.2, 1.3, 0.4, 0.5]

plt.figure(figsize=(10, 6))
plt.plot(boyAraliklari, firstPercentages, label='70-79', marker='o')
plt.plot(boyAraliklari, secondPercentages, label='40-49', marker='o')
plt.title('Height Distribution of Men aged 70-79 and 40-49')
plt.xlabel('Heights')
plt.ylabel('Percentage (%)')
plt.legend()
plt.xticks(rotation=45)
plt.grid()
plt.show()
```

- The output of the code will be given below:



- After plotting the graph, what we need to calculate the GraphPad T test is the mean and standard deviation of men in the 40-49 age range and 70-79 age range. We will enter the N values as 10, 100, 1000, respectively, as in the previous question. I calculate the standard deviation and mean values.

```
In [2]: import math

sabitBoy = [147, 148, 150.5, 153, 155.5, 158.5, 161, 163.5, 166, 168.5, 171, 173.5, 176, 178.5, 181, 183.5, 186, 188.5, 191.5, 19

# 70-79 yaş aralığındaki erkeklerin dağılımı
firstPercentages = [0, 0,0,0,0,0,5.8,7,10.2,12.1,12.6,12.6,14.9,10.6,5.2,3.9,3.7,1.4,0,0,0]
#print(len(kadınYüzdelikler))
firstOrtalama = sum([ort * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, firstPercentages)])
firstStandartSapma = math.sqrt(sum([((ort - firstOrtalama) ** 2) * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, firstPercent

# 40-49 yaş aralığındaki erkeklerin dağılımı
secondPercentages = [0, 0, 0, 0, 0, 1.9, 1.9,1.8 ,4.2, 9.6, 10.9, 10.1, 4, 15.2, 9.5, 8.3, 5.1, 5.2, 1.3, 0.4, 0.5]
#print(len(erkekYüzdelikler))
secondOrtalama = sum([ort * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, secondPercentages)])
secondStandartSapma = math.sqrt(sum([((ort - secondOrtalama) ** 2) * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, secondPerc


# Sonuçları yazdırma
print("70-79:")
print("Ortalama (Mean): {:.2f}".format(firstOrtalama))
print("Standart Sapma: {:.2f}".format(firstStandartSapma))
print("40-49:")
print("Ortalama (Mean): {:.2f}".format(secondOrtalama))
print("Standart Sapma: {:.2f}".format(secondStandartSapma))
```

```
70-79:
Ortalama (Mean): 172.75
Standart Sapma: 6.66
40-49:
Ortalama (Mean): 158.46
Standart Sapma: 18.40
```

Unpaired *t* test results

P value and statistical significance:
The two-tailed P value equals 0.0330
By conventional criteria, this difference is considered to be statistically significant.

Confidence interval:
The mean of Male 40-49 minus Male 70-79 equals -14.2900
95% confidence interval of this difference: From -27.2905 to -1.2895

Intermediate values used in calculations:
t = 2.3093
df = 18
standard error of difference = 6.188

Review your data:

Group	Male 40-49	Male 70-79
Mean	158.4600	172.7500
SD	18.4000	6.6600
SEM	5.8186	2.1061
N	10	10

N = 10 → P = 0.0330

Unpaired *t* test results

P value and statistical significance:
The two-tailed P value is less than 0.0001
By conventional criteria, this difference is considered to be extremely statistically significant.

Confidence interval:
The mean of Male 40-49 minus Male 70-79 equals -14.2900
95% confidence interval of this difference: From -18.1489 to -10.4311

Intermediate values used in calculations:
t = 7.3027
df = 198
standard error of difference = 1.957

Review your data:

Group	Male 40-49	Male 70-79
Mean	158.4600	172.7500
SD	18.4000	6.6600
SEM	1.8400	0.6660
N	100	100

N = 100 → P = 0.0001

Unpaired *t* test results

P value and statistical significance:
The two-tailed P value is less than 0.0001
By conventional criteria, this difference is considered to be extremely statistically significant.

Confidence interval:
The mean of Male 40-49 minus Male 70-79 equals -14.2900
95% confidence interval of this difference: From -15.5058 to -13.0742

Intermediate values used in calculations:
t = 23.0930
df = 1998
standard error of difference = 0.619

Review your data:

Group	Male 40-49	Male 70-79
Mean	158.4600	172.7500
SD	18.4000	6.6600
SEM	0.5819	0.2106
N	1000	1000

N = 1000 → P = 0.0001

STEP 8)

Qualifications for the question can be shown as following:

- A = 288; B = 30
- Formula: Z = ((h/288) + (1/30)*((h*h)/1000))
- Percentages showing the distribution of men aged 50-59 by age and height. The image in right side shows probabilities of distributions of Man according to the heights, aged 50-59.

	Height Distributions	*Dist. Percentage of Man*
60		
61	162 cm below	%4.3
62	162-165 arası	%3.3
63	165-167 arası	%4.6
64	167-170 arası	%6.4
65	170-172 arası	%11.7
66	172-175 arası	%10.9
67	175-177 arası	%13.1
68	177-180 arası	%15.7
69	180-182 arası	%11.2
70	182-185 arası	%10.4
71	185-187 arası	%2.1
72	187-190 arası	%2.9
73	190-193 arası	%2.9
74	193-195 arası	%0.1
75	195-198 arası	%0.4

First of all, I define the height ranges and the probability percentage of each height range in a fixed way.

```
In [11]: # Given percentiles and height ranges
         heightPercentages = [4.3, 3.3, 4.6, 6.4, 11.7, 10.9, 13.1, 15.7, 11.2, 10.4, 2.1, 2.9, 2.9, 0.1, 0.4]
         heightBetweens = [(162, 165), (165, 167), (167, 170), (170, 172), (172, 175), (175, 177), (177, 180), (180, 182), (182, 185), (18

         varianceZ = 0
         expectedValue = 0

         for i in range(len(heightBetweens)):
             hMin = heightBetweens[i][0]
             hMax = heightBetweens[i][1]
             hMiddle = (hMin + hMax) / 2
             newBoy = heightPercentages[i]

             varianceZ += (((hMiddle / 288) + ((hMiddle ** 2) / (30 * 1000))) ** 2) * (newBoy / 100)
             expectedValue += ((hMiddle / 288) + ((hMiddle ** 2) / (30 * 1000))) * (newBoy / 100)

         print("Variance of Z: ", varianceZ)
         print("Expected value of Z: ", expectedValue)

         Variance of Z:  2.8314282687711723
         Expected value of Z:  1.6757792555555557
```

According to the code above, results are determined as following:

- Variance of Z: 2.83
- Expected Value of Z: 1.67

PART B:

STEP 1)

- My relatives' heights and eye colors are in below. I used excel for preparing this tables. I collected these data from 20 people.

	Heights	Eye colors
1	Heights	Eye colors
2	197	Black
3	194	Black
4	179	Brown
5	172	Light Brown
6	175	Black
7	162	Black
8	179	Dark brown
9	145	Light Brown
10	149	Hazel
11	169	Black
12	165	Brown
13	163	Black
14	184	Hazel
15	185	Light Brown
16	181	brown
17	170	Brown
18	160	Black
19	140	Brown
20	177	Green
21	173	Brown
22	RELATIVES	

STEP 2)

- My friends' heights and eye colors are in below. I used excel for preparing this tables. I collected data from 20 people.

Heights	Eye Color
196	Light Brown
170	Black
159	Hazel
183	Dark Brown
175	Black
170	Green
184	Black
186	Brown
192	Light Brown
188	Black
182	Brown
165	Brown
184	Black
166	Blue
173	Black
174	Brown
169	Light Brown
165	Light Brown
185	Brown
187	Black
FRIENDS	

STEP 3)

I need to create different tables based on eye color and length data that I collect from my relatives and friends.

- First of all, to create a cumulative percentage table, we need to find the percentage of people based on their data. For this, I first calculate how many people are in that

height within 20 people in certain height ranges. Then I convert this to a percentage. After making these values for all size ranges, I calculate the cumulative percentage.

I have worked as much on the papers as I can to calculate the cumulative percentage and I will show it in the report.

- Finally, we can create cumulative percentage tables for relative and non-relative. After this value, we find how many lengths relative and non-relatives contain between the specified lengths. This also refers to the percentage.
- First, I create the cumulative tables without eye color for relative and friends.

24	RELATIVES (HEIGHTS WITHOUT EYE COLORS)			
25	HEIGHTS	TOTAL NUMBER IN TWENTY PEOPLE	PERCENTAGES (%)	CUMULATIVE DISTRIBUTIONS (%)
26	140 - 150	3	15	15
27	151 - 160	1	5	20
28	161 - 170	5	25	45
29	171 - 180	6	30	75
30	181 - 190	3	15	90
31	191-200	2	10	100

33	NON-RELATIVES (FRIENDS) (HEIGHTS WITHOUT EYE COLORS)	TOTAL NUMBER IN TWENTY PEOPLE	PERCENTAGES (%)	CUMULATIVE DISTRIBUTIONS (%)
34	140 - 150	0	0	0
35	151 - 160	1	5	5
36	161 - 170	6	30	35
37	171 - 180	3	15	50
38	181 - 190	8	40	90
39	191-200	2	10	100

- Then I create separate tables for relative and friend tables, which include eye color and height lengths.

41	RELATIVES (HEIGHTS WITH EYE COLORS)	TOTAL NUMBERS IN TWENTY PEOPLE			
42	HEIGHTS	BROWN	HAZEL	BLACK	GREEN
43	140 - 150	2	1	0	0
44	151 - 160	0	0	1	0
45	161 - 170	2	0	3	0
46	171 - 180	4	0	1	1
47	181 - 190	2	1	0	0
48	191-200	0	0	2	0

50	NON-RELATIVES (HEIGHTS WITH EYE COLORS)	TOTAL NUMBERS IN TWENTY PEOPLE				
51	HEIGHTS	BROWN	HAZEL	BLACK	GREEN	BLUE
52	140 - 150	0	0	0	0	0
53	151 - 160	0	1	0	0	0
54	161 - 170	3	0	1	1	1
55	171 - 180	1	0	2	0	0
56	181 - 190	4	0	4	0	0
57	191-200	2	0	0	0	0

STEP 4)

- Since the frequencies of the heights I get for Relative and Friend are equal, I have equalized the lengths in only one parameter and I can use the same lengths for both relative and friend. Then I use the formula to calculate the mean and standard deviation. Let's view the code and its output below.

```
In [21]: import math

#Boyların ortalamasını yazıyorum çünkü hesaplama yapacak. Relatives ve Friends boyları değişkenlik gösterebilir ona bakacağız.
relativesHeigthsBoy = [145, 155.5, 165.5, 175.5, 185.5, 195.5] #relative's heights ortalama daha az sıklık.
#Relatives ve Friendslerin boy ortalamaları aynı olduğu için eşitliyorum.

#Relativesheights = [197,194,179,172,175,162,179,145,149,169,165,163,184,185,181,170,160,140,177,173]
#friendsHeights = [196,170,159,183,175,170,184,186,192,188,182,165,184,166,173,174,169,165,185,187]

# relatives
relativesPercentages = [15, 5, 25, 30, 15, 1] #relativelerin yüzde ortalaması
relativesOrtalama = sum([ort * yüzdelik / 100 for ort, yüzdelik in zip(relativesHeigthsBoy, relativesPercentages)])
relativesStandartSapma = math.sqrt(sum([((ort - relativesOrtalama) ** 2) * yüzdelik / 100 for ort, yüzdelik in zip(relativesHeigt

# friends
friendsPercentages = [0, 5, 30, 15, 40, 10]
friendsOrtalama = sum([ort * yüzdelik / 100 for ort, yüzdelik in zip(relativesHeigthsBoy, friendsPercentages)])
friendsStandartSapma = math.sqrt(sum([((ort - friendsOrtalama) ** 2) * yüzdelik / 100 for ort, yüzdelik in zip(relativesHeigthsBo

# Sonuçlar
print("Results of heights of my relatives' and friends': ")
print("Relatives:")
print("Ortalama (Mean): {:.2f}".format(relativesOrtalama))
print("Standart Sapma: {:.2f}".format(relativesStandartSapma))
print("Friends:")
print("Ortalama (Mean): {:.2f}".format(friendsOrtalama))
print("Standart Sapma: {:.2f}".format(friendsStandartSapma))
```

```
Results of heights of my relatives' and friends':
Relatives:
Ortalama (Mean): 153.33
Standart Sapma: 19.19
Friends:
Ortalama (Mean): 177.50
Standart Sapma: 11.22
```

- We will get the results by entering the standard deviation and mean values we found above, using GraphPad, separately for the relative and friend groups, and by entering the N values from 10 to 1000.

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value equals 0.0029

By conventional criteria, this difference is considered to be very statistically significant.

Confidence interval:

The mean of Relatives' Heights minus Friends' Heights equals -24.1700

95% confidence interval of this difference: From -38.9385 to -9.4015

Intermediate values used in calculations:

t = 3.4383

df = 18

standard error of difference = 7.030

Review your data:

Group	Relatives' Heights	Friends' Heights
Mean	153.3300	177.5000
SD	19.1900	11.2200
SEM	6.0684	3.5481
N	10	10

N = 10 → P = 0.0029

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value is less than 0.0001

By conventional criteria, this difference is considered to be extremely statistically significant.

Confidence interval:

The mean of Relatives' Heights minus Friends' Heights equals -24.1700

95% confidence interval of this difference: From -28.5537 to -19.7863

Intermediate values used in calculations:

t = 10.8730

df = 198

standard error of difference = 2.223

Review your data:

Group	Relatives' Heights	Friends' Heights
Mean	153.3300	177.5000
SD	19.1900	11.2200
SEM	1.9190	1.1220
N	100	100

N = 100 → P = 0.0001

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value is less than 0.0001

By conventional criteria, this difference is considered to be extremely statistically significant.

Confidence interval:

The mean of Relatives' Heights minus Friends' Heights equals -24.1700

95% confidence interval of this difference: From -25.5511 to -22.7889

Intermediate values used in calculations:

t = 34.3835

df = 1998

standard error of difference = 0.703

Review your data:

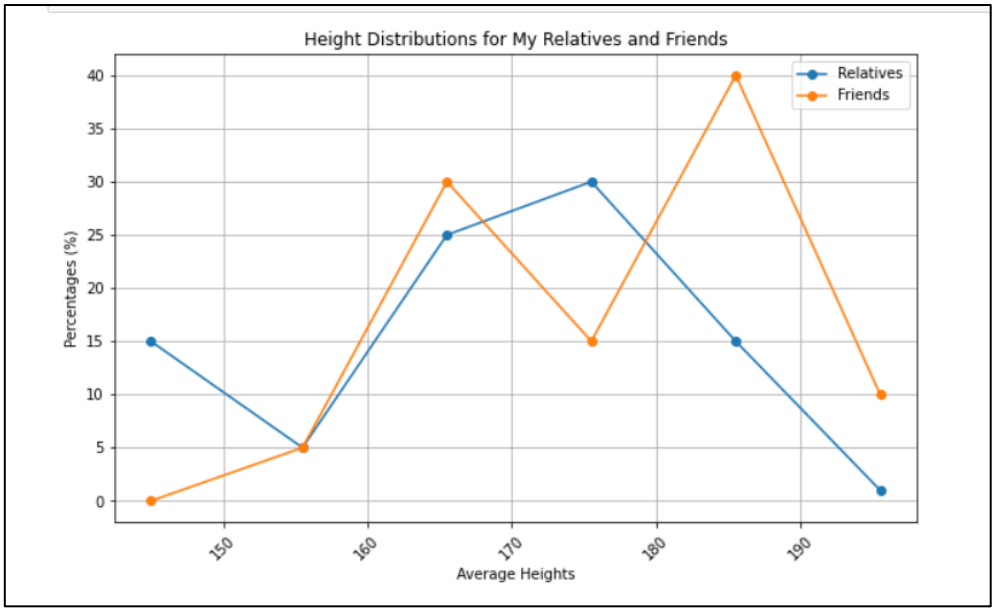
Group	Relatives' Heights	Friends' Heights
Mean	153.3300	177.5000
SD	19.1900	11.2200
SEM	0.6068	0.3548
N	1000	1000

N = 1000 → P = 0.0001

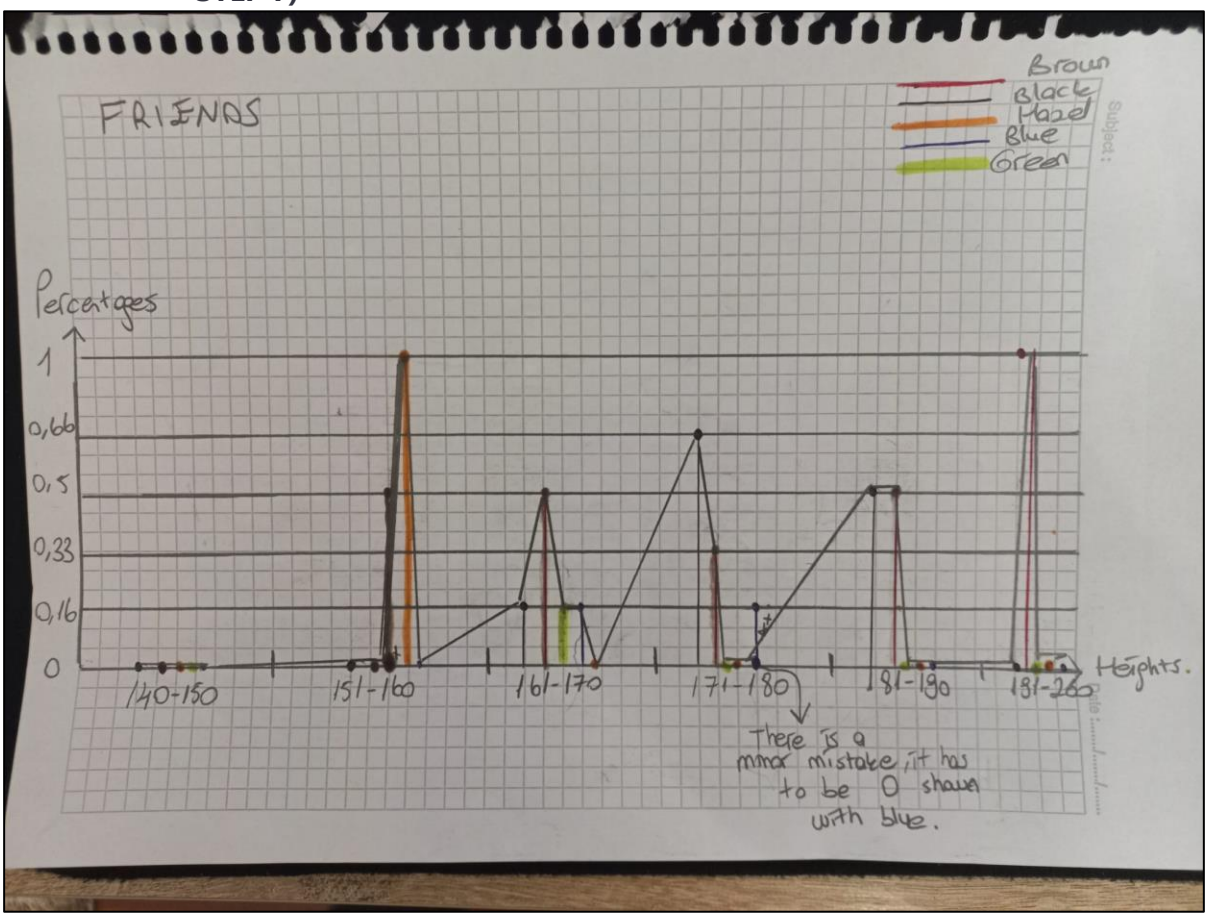
STEP 5)

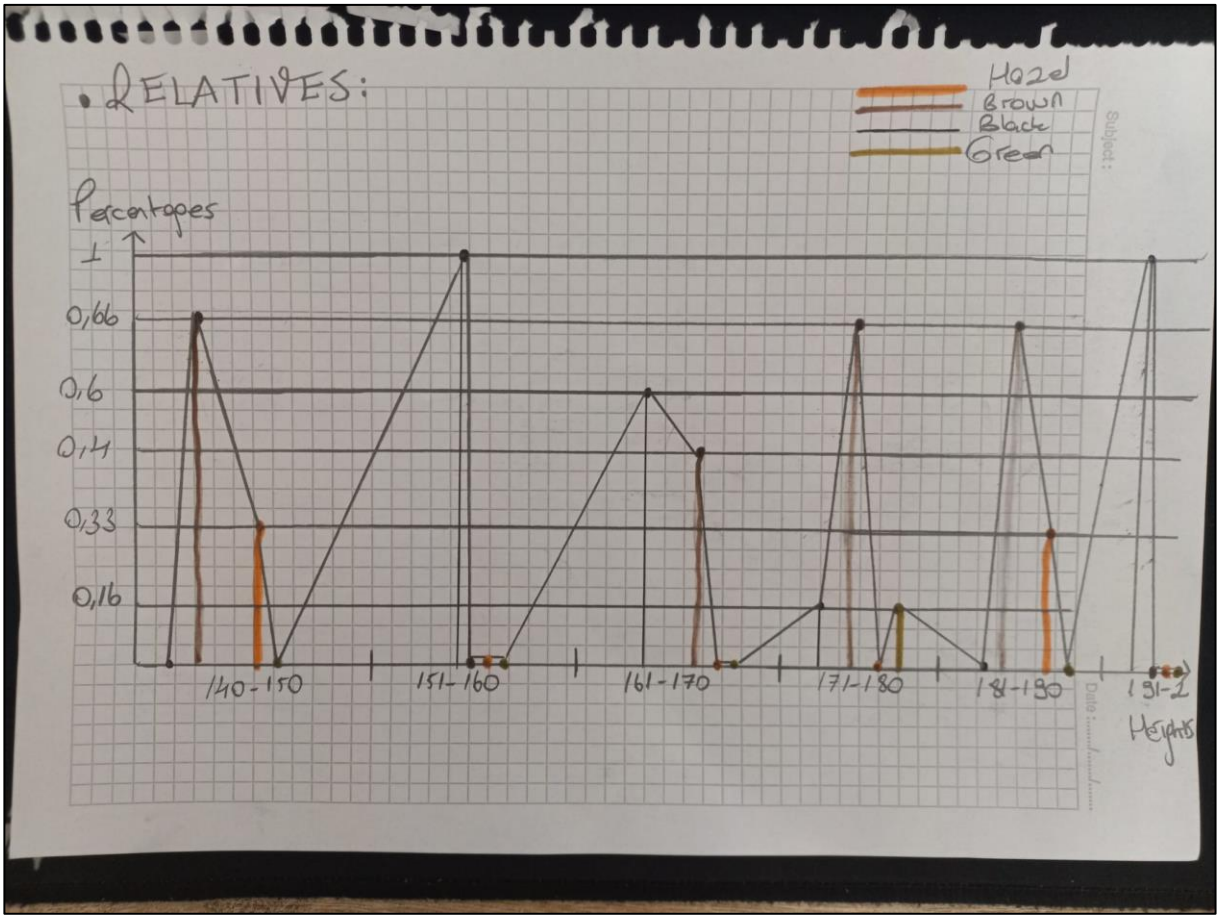
- Since I set the limits of the relative and non-relative heights equally, I can show the average height in only one parameter. Then I create a graph by entering the relative and friend percentages. Lets see the code and its ouptut below.

```
In [24]: import matplotlib.pyplot as plt
relativesHeigthsBoy = [145, 155.5, 165.5, 175.5, 185.5, 195.5]
relativesPercentages = [15, 5, 25, 30, 15, 1]
friendsPercentages = [0, 5, 30, 15, 40, 10]
plt.figure(figsize=(10, 6))
plt.plot(relativesHeigthsBoy,relativesPercentages , label='Relatives', marker='o')
plt.plot(relativesHeigthsBoy,friendsPercentages , label='Friends', marker='o')
plt.title('Height Distributions for My Relatives and Friends')
plt.xlabel('Average Heights')
plt.ylabel('Percentages (%)')
plt.legend()
plt.xticks(rotation=45)
plt.grid()
plt.show()
```



STEP 7)


FRIENDS
Brown
Black
Hazel
Blue
Green
Subject:
Percentages
1
0,66
0,5
0,33
0,16
0
140-150
151-160
161-170
171-180
181-190
181-200
Heights.
Date:
There is a minor mistake, it has to be 0 shaded with blue.


RELATIVES:
Hazel
Brown
Black
Green
Subject:
Percentages
1
0,66
0,6
0,4
0,33
0,16
140-150
151-160
161-170
171-180
181-190
191-2
Heights
Date:

<u>PART A DATASET</u> (PERCENTAGES OF DISTRIBUTIONS- %):

- It shows the distribution of men and women according to their height as a percentage between the specified ages. The table referenced within the project shows its cumulative distribution. However, we need percentage values to solve the stages in Part A. Therefore, I calculated the following percentage values by looking at the cumulative values. The distribution percentage of men and women at certain ages according to their height is shown below.

109		MALE AND FEMALE AGE DISTRIBUTIONS											
110		MALES						FEMALES					
111	HEIGHTS (CM)	20-29	30-39	40-49	50-59	60-69	70-79	20-29	30-39	40-49	50-59	60-69	70-79
112	147	0	0	0	0	0	0	0	1,7	0	1	0	3,3
113	147-149	0	0	0	0	0	0	2,6	1,4	1,6	1,1	3,6	5,4
114	149-152	0	0	0	0	0	0	3,1	2,9	3,4	5,9	5,4	7,3
115	152-154	0	0	0	0	0,4	0	6,6	5,6	5,8	8,7	5,7	10
116	154-157	0	0	0	0	0	0	8,5	8,1	9	6,6	8,7	10,9
117	157-160	0	3,1	1,9	0	1,9	0	10,4	11,6	11	13	15	15
118	160-162	3,7	1,3	1,9	4,3	2,1	5,8	13,1	15,3	15,2	14,4	14,4	18
119	162-165	3,5	2,3	1,8	3,3	3,4	7	10,6	14,6	12	17,7	13,8	12,9
120	165-167	4,4	6,4	4,2	4,6	6,9	10,2	18,3	12,8	14,2	11,3	16,7	6,5
121	167-170	9	6,5	9,6	6,4	9	12,1	9,9	10,9	10,8	8,7	10	6,1
122	170-172	12,5	12,6	10,9	11,7	14	12,6	8	6,9	8,2	6,8	3,7	3
123	172-175	9,1	13,2	10,1	10,9	12,5	12,6	3,8	4,3	3,5	2,1	0,8	1,2
124	175-177	16,4	12,7	4	3,1	15	14,9	3,5	2,8	3,1	1,6	1,8	0
125	177-180	12,1	11,3	15,2	15,7	4,8	10,6	2	0	1,6	1,1	0,2	0,4
126	180-182	9,2	9,1	9,5	11,2	9,3	5,2	0,4	0,5	0,1	0	0,1	0
127	182-185	9,1	10,5	8,3	10,4	9,3	3,9	0	0,5	0	0	0	0
128	185-187	5,1	5	5,1	2,1	4,2	3,7	0	0,1	0	0	0,1	0
129	187-190	4,2	1,8	5,2	2,9	2,1	1,4	0	0	0	0	0	0
130	190-193	1,7	1,8	1,3	2,9	0,1	0	0	0	0	0	0	0
131	193-195	0	1,8	0,4	0,1	0	0	0	0	0,5	0	0	0
132	195-198	0	0,1	0,5	0,4	0	0	0	0	0	0	0	0

<u>PART B DATASET</u> (INCLUDING RELATIVES NON-RELATIVES HEIGHTS AND EYE COLORS'):

- First 20 data belongs to relatives. After from 20 belongs to friends.

1	197	Black
2	194	Black
3	179	Brown
4	172	Light Brown
5	175	Black
6	162	Black
7	179	Dark brown
8	145	Light Brown
9	149	Hazel
10	169	Black
11	165	Brown
12	163	Black
13	184	Hazel
14	185	Light Brown
15	181	brown
16	170	Brown
17	160	Black
18	140	Brown
19	177	Green
20	173	Brown
21	196	Light Brown
22	170	Black
23	159	Hazel
24	183	Dark Brown
25	175	Black
26	170	Green
27	184	Black
28	186	Brown
29	192	Light Brown
30	188	Black
31	182	Brown
32	165	Brown
33	184	Black
34	166	Blue
35	173	Black
36	174	Brown
37	169	Light Brown
38	165	Light Brown
39	185	Brown
40	187	Black

MISSING PLACE IN PART A:

- I was supposed to do the same for women as I did for men aged 40-49 and 70-79, but I missed that part at first. I'm adding that part now. I repeat the same process for women aged 40-49 and 70-79.

```
In [25]: import math

sabitBoy = [147, 148, 150.5, 153, 155.5, 158.5, 161, 163.5, 166, 168.5, 171, 173.5, 176, 178.5, 181, 183.5, 186, 188.5, 191.5, 19

# 70-79 yaş aralığındaki kadınların dağılımı
firstPercentages = [3.3,5.4,7.3,10,10.9,15,18,12.9,6.5,6.1,3,1.2,0,0.4,0,0,0,0,0,0,0]
#print(len(kadınYüzdelikler))
firstOrtalama = sum([ort * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, firstPercentages)])
firstStandartSapma = math.sqrt(sum([((ort - firstOrtalama) ** 2) * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, firstPercent

# 40-49 yaş aralığındaki kadınların dağılımı
secondPercentages = [0, 1.6, 3.4, 5.8, 9, 11, 15.2,12 ,14.2, 10.8, 8.2, 3.5, 3.1, 1.6, 0.1, 0, 0, 0, 0, 0.5, 0]
#print(len(erkekYüzdelikler))
secondOrtalama = sum([ort * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, secondPercentages)])
secondStandartSapma = math.sqrt(sum([((ort - secondOrtalama) ** 2) * yüzdelik / 100 for ort, yüzdelik in zip(sabitBoy, secondPerc


# Sonuçları yazdırma
print("70-79:")
print("Ortalama (Mean): {:.2f}".format(firstOrtalama))
print("Standart Sapma: {:.2f}".format(firstStandartSapma))
print("40-49:")
print("Ortalama (Mean): {:.2f}".format(secondOrtalama))
print("Standart Sapma: {:.2f}".format(secondStandartSapma))
```

```
70-79:
Ortalama (Mean): 158.92
Standart Sapma: 6.47
40-49:
Ortalama (Mean): 163.21
Standart Sapma: 7.10
```

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value equals 0.1749

By conventional criteria, this difference is considered to be not statistically significant.

Confidence interval:

The mean of 70-79 FEMALE minus 40-49 FEMALE equals -4.2900

95% confidence interval of this difference: From -10.6718 to 2.0918

Intermediate values used in calculations:

t = 1.4123

df = 18

standard error of difference = 3.038

Review your data:

Group	70-79 FEMALE	40-49 FEMALE
Mean	158.9200	163.2100
SD	6.4700	7.1000
SEM	2.0460	2.2452
N	10	10

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value is less than 0.0001

By conventional criteria, this difference is considered to be extremely statistically significant.

Confidence interval:

The mean of 70-79 FEMALE minus 40-49 FEMALE equals -4.2900

95% confidence interval of this difference: From -6.1843 to -2.3957

Intermediate values used in calculations:

t = 4.4661

df = 198

standard error of difference = 0.961

Review your data:

Group	70-79 FEMALE	40-49 FEMALE
Mean	158.9200	163.2100
SD	6.4700	7.1000
SEM	0.6470	0.7100
N	100	100

Unpaired *t* test results

P value and statistical significance:

The two-tailed P value is less than 0,0001

By conventional criteria, this difference is considered to be extremely statistically significant.

Confidence interval:

The mean of 70-79 FEMALE minus 40-49 FEMALE equals -4,2900

95% confidence interval of this difference: From -4,8868 to -3,6932

Intermediate values used in calculations:

t = 14,1229

df = 1998

standard error of difference = 0,304

Review your data:

Group	70-79 FEMALE	40-49 FEMALE
Mean	158,9200	163,2100
SD	6,4700	7,1000
SEM	0,2046	0,2245
N	1000	1000

In [26]:
```
import matplotlib.pyplot as plt

boyAraliklari = ['147', '147-149', '149-152', '152-154', '154-157', '157-160', '160-162', '162-165',
                 '165-167', '167-170', '170-172', '172-175', '175-177', '177-180', '180-182', '182-185',
                 '185-187', '187-190', '190-193', '193-195', '195-198']

# 70-79 yaş aralığındaki kadınların dağılımı
firstPercentages = [3.3,5.4,7.3,10,10.9,15,18,12.9,6.5,6.1,3,1.2,0,0.4,0,0,0,0,0,0]

# 40-49 yaş aralığındaki kadınların dağılımı
secondPercentages = [0, 1.6, 3.4, 5.8, 9, 11, 15.2,12 ,14.2, 10.8, 8.2, 3.5, 3.1, 1.6, 0.1, 0, 0, 0, 0, 0.5, 0]

plt.figure(figsize=(10, 6))
plt.plot(boyAraliklari, firstPercentages, label='70-79', marker='o')
plt.plot(boyAraliklari, secondPercentages, label='40-49', marker='o')
plt.title('Height Distribution of Women aged 70-79 and 40-49')
plt.xlabel('Heights')
plt.ylabel('Percentage (%)')
plt.legend()
plt.xticks(rotation=45)
plt.grid()
plt.show()
```