Nama : Kevin Aryo Wicaksono

NIM : 175150201111039

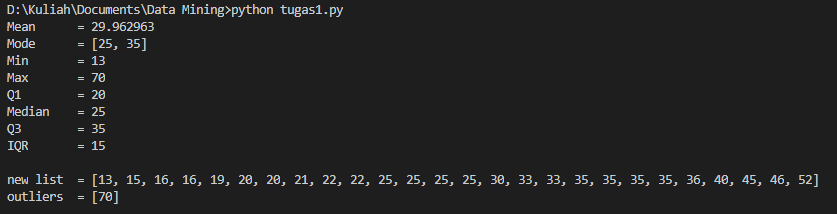
Kelas : Data Mining – C

Given an age data :  13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70

1. Write a program (without any libraries) that computes
2. Mean
3. Mode
4. Five-number summary
5. Detect outliers using IQR. Do outliers exist?

Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90 | ageList = [13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70]  ageListLength = len(ageList)  #Mean  def mean(listNum, listLen):  totalNumber = sum(listNum)  mean = totalNumber/listLen  print("%-10s= %f" %("Mean", mean))  #Mode  def mode(listNum):  modeNum = 0  listMode = []  for num in listNum:  if num in listMode:  continue  counter = listNum.count(num)  if counter > modeNum:  del listMode[:]  listMode.append(num)  modeNum = counter  elif counter == modeNum:  listMode.append(num)  print("%-10s= %s" %("Mode", str(listMode)))  # Index of the median  def Qindex(listNum, l, r):  n = r - l + 1  n = (n + 1) // 2 - 1  return n + l  #Maximum and Minimum Value  def maxminVal(listNum, a):  if a=="min":  minimum = listNum[0]  for num in listNum:  if num < minimum:  minimum = num  return minimum  if a=="max":  maximum = listNum[0]  for num in listNum:  if num > maximum:  maximum = num  return maximum  #Five Number Summary  def fiveNumSummary(listNum, listLen):  # listNum = listNum.sort()  #middle index  mid\_index = Qindex(listNum, 0, listLen)  Q1 = listNum[Qindex(listNum, 0, mid\_index)]  median = listNum[(Qindex(listNum, 0, listLen))]  Q3 = listNum[Qindex(listNum, mid\_index + 1, listLen)]  print("%-10s= %d" %("Min", maxminVal(listNum,"min")))  print("%-10s= %d" %("Max", maxminVal(listNum,"max")))  print("%-10s= %d" %("Q1", Q1))  print("%-10s= %d" %("Median", median))  print("%-10s= %d" %("Q3", Q3))  #Outlier - IQR  def outIQR(listNum, listLen):  mid\_index = Qindex(listNum, 0, listLen)  Q1 = listNum[Qindex(listNum, 0, mid\_index)]  Q3 = listNum[Qindex(listNum, mid\_index + 1, listLen)]  IQR = Q3 - Q1  lowerBound = Q1 - 1.5\*IQR  upperBound = Q3 + 1.5\*IQR  #list without outliers  newList = []  #list of outliers  outliers = []  for num in listNum:  if num>lowerBound and num<upperBound:  newList.append(num)  else:  outliers.append(num)  print("%-10s= %d" %("IQR", IQR))  print("\nnew list = " + str(newList))  print("outliers = " + str(outliers))  mean(ageList, ageListLength)  mode(ageList)  fiveNumSummary(ageList, ageListLength)  outIQR(ageList, ageListLength) |



1. Draw a boxplot describing the five-number summary of the data

