ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων

(ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2022-23)

ΕΡΓΑΣΙΑ 1 - Ιστογράμματα

Προθεσμία: 31 Μαρτίου 2023

Ονοματεπώνυμο: Evangelos Iliadis 3117 (Ευάγγελος Ηλιάδης 3117)

Τα ιστογράμματα χρησιμοποιούνται από συστήματα βάσεων δεδομένων για την προσεγγιστική αναπαράσταση της κατανομής τιμών σε πεδία πινάκων. Για τις ανάγκες του πρώτου σετ ασκήσεων του μαθήματος "Διαχείριση Σύνθετων Δεδομένων" κληθήκαμε λοιπόν να αναπτύξουμε κώδικα σε C, C++, Java, Python με σκοπό να δημιουργεί ιστογράμματα και να τα χρησιμοποιεί για την εκτίμηση του πλήθους των αποτελεσμάτων τελεστών επιλογής.

Οποιαδήποτε αριθμητικά αποτελέσματα που παρατίθενται παρακάτω αφορούν το αρχείο “acs2015_census_tract_data.csv” από τη διεύθυνση https://www.kaggle.com/datasets/muonneutrino/us-census-demographic-data. Ωστόσο ο κώδικας έχει αναπτυχθεί έτσι ώστε να μπορεί να δουλέψει για οποιαδήποτε αρχεία περιέχουν τη στήλη “Income” αρκεί να αλλάζει κάθε φορά η μεταβλητή path στο αντίστοιχο μονοπάτι της επιθυμητής εισόδου.

Για παράδειγμα εάν θέλαμε να το τρέξουμε σε έναν από τους υπολογιστές του εργαστηρίου του τμήματός μας, θα ήταν κάπως έτσι:

```
path = '/usr/home/students/stud15/cse53117/Desktop/ddtry/acs2015_census_tract_data.csv'
```

Μέσα στο πρόγραμμα υπάρχουν επιπλέον επεξηγηματικά σχόλια για την κατανόηση του ρόλου που επιτελούν τα διάφορα blocks κώδικα, από έναν απλό αναγνώστη που δεν εμπλέκεται με την ανάπτυξη του εν λόγω κώδικα. Καθώς και γραμμές κώδικα, σε σχόλια, που μπορούν να φανούν χρήσιμες κατά την αποσφαλμάτωσή του.

Αρχικά είναι απαραίτητη η εξαγωγή των ζητούμενων δεδομένων “Income”.

```
#Data extraction

with open(path, mode = 'r') as inf: #Open csv file for reading.
    reader = csv.reader(inf, delimiter = ',') #Determine the delimiter that seperates each cell.
    firstRow = next(reader) #Get first row of the opened csv file.


    #Find the column that contains the 'Income data.
    for col in firstRow:
        if col == 'Income':
            break #Found 'Income' column. Break.
        else:
            columnCounter += 1 #For each column that isn't 'Income', columnCounter++.
    #print(columnCounter)

    #Add income data to array
    for row in reader:
        if row[columnCounter].strip(): #We need only the valid entries.
            #print(row[columnCounter].strip())
            #We use float() function because we need to find min/max later on.
            #So we need float values in order to compare them.
            incomes.append(float(row[columnCounter]))
    #print(len(incomes), 'valid income values')
```

Αρχικά με ανοιχτό το csv αρχείο διαβάζουμε την πρώτη γραμμή η οποία περιέχει τα headers. Έπειτα διατρέχουμε αυτή τη γραμμή για το κελί το οποίο περιέχει τον όρο “Income”. Αυτή η διαδικασία γίνεται δεδομένου ότι για διαφορετικά csv αρχεία η στήλη με τα “Incomes” μπορεί να είναι διαφορετική. Έπειτα απομονώνουμε τα στοιχεία και τα βάζουμε σε μία δομή.

Μέρος 1ο

Σε αυτό το μέρος μας ζητήθηκε να δημιουργήσουμε δύο ιστογράμματα από τα δεδομένα που μας παρήχθησαν με τη μορφή csv αρχείων. Το πρώτο θα είναι τύπου equi-width και το δεύτερου τύπου equi-depth.

1) equi-width: πρόκειται για ένα ιστόγραμμα του οποίου τα bins έχουν όλα ίσο εύρος. Προκειμένου να επιτευχθεί αυτό αρχικά χρειάζεται να γνωρίζουμε το συνολικό φάσμα που καλύπτουν οι τιμές μας. Γι’ αυτό χρειάζεται να βρούμε τις min/ max τιμές μέσα από όλα τα στοιχεία “Income” που απομονώσαμε.

```
minIncome = min(incomes) #Find min income value.
maxIncome = max(incomes) #Find max income value.
```

Έπειτα βρίσκουμε το συνολικό εύρος τιμών με το “maxIncome – minIncome”. Το εύρος που θα αποδοθεί στο εκάστοτε bin γίνεται διαιρώντας το συνολικό εύρος διά 100.

```
binWidth = (maxIncome - minIncome)/100 #Each bin needs to have the same width. So we divide the min/max spectrum into 100 equal bins.
```

Προκειμένου να έχουμε 100 bins χρειαζόμαστε 101 edges (άκρα). Μιας και γνωρίζουμε ότι όλα τα bins έχουν το ίδιο εύρος αρκεί μια απλή δομή επανάληψης, ξεκινώντας από το ελάχιστον στοιχείο “minIncome” στο οποίο θα προσθέτουμε σε κάθε επανάληψη το “binWidth”, μιας και αυτή είναι η απόσταση μεταξύ δύο διαδοχικών άκρων. Τα αποτελέσματα τα κρατάμε σε έναν πίνακα μιας και θα χρησιμοποιηθούν αργότερα ως άκρα για τα bins.

```
binEdges = [] #Here will be stored the histogram edges for each bin.
for i in range(101): #101 cause with 100 we don't cover the full income value spectrum.
    roundedBinEdge = minIncome + i*binWidth #Find i'th bin edge.
    binEdges.append(roundedBinEdge) #Add the i'th bin edge to the list.
```

Έπειτα δημιουργούμε τα bins, ως εξής

```
binRanges = [] #Each element is a histogram bin range.
binRange = [0,0] #Dummy for bin range. Will be replaced later on with the real edges.
for i in range(100):
    binRange = [binEdges[i],binEdges[i+1]]
    binRanges.append(binRange)
```

Τέλος, έμεινε να δούμε πόσα στοιχεία “Incomes” ανήκουν σε κάθε bin. Για αυτό θα δημιουργήσουμε ένα πίνακα εκατό μηδενικών και διατρέχοντας τα “Incomes”, “bins” θα αυξάνουμε +1 το κελί που αντιστοιχεί στο bin που ανήκει το εκάστοτε income.

```
numtuples = [0]*100 #Create an array full of one hundred 0's.

#Find the number of tuples in each histogram bin.
for i in incomes: #For each income element.
    numtuplesIndex = 0 #Will help us determine which numtuples element should be increased by 1.
    counter = 0 #Helper counter. Keeps track of the bin we're in.
    for j in binRanges: #For each bin.
        if counter < 99:
            if i >= j[0] and i< j[1]: #If income value is between the two bin edges.
                numtuples[numtuplesIndex] += 1 #Increase it by 1.
            else:
                numtuplesIndex += 1 #Else keep searching. Increase helper index by 1.
        else:
            if i >= j[0] and i<= j[1]: #If income value is between the two bin edges.
                numtuples[numtuplesIndex] += 1 #Increase it by 1.
            else:
                numtuplesIndex += 1 #Else keep searching. Increase helper index by 1.
        counter += 1
```

Τέλος, έμεινε να γίνει το print του ιστογράμματος equi-width.

```
#equiwidth print
print('equiwidth:')
for i in range(100):
      print('range:', '[', round(binEdges[i],2), ',', round(binEdges[i+1],2), ')', ',', 'numtuples:', numtuples[i])
```

Δεδομένου ότι τα bin edges είναι 101 κάνουμε ένα loop 100 φορών προκειμένου να μπορούμε να μιλήσουμε για i+1. Επίσης τα στρογγυλοποιούμε στο 2ο δεκαδικό ψηφίο, ανάλογο του υποδειγματικού παραδείγματος της εκφώνησης.

Τα αποτελέσματα του equi-width ιστογράμματος παρατίθενται με αντίστοιχο τρόπο με αυτόν στο υπόδειγμα της εκφώνησης της άσκησης:

equiwidth:

range: [2611.0 , 5072.39) , numtuples: 13

range: [5072.39 , 7533.78) , numtuples: 23

range: [7533.78 , 9995.17) , numtuples: 85

range: [9995.17 , 12456.56) , numtuples: 240

range: [12456.56 , 14917.95) , numtuples: 386

range: [14917.95 , 17379.34) , numtuples: 687

range: [17379.34 , 19840.73) , numtuples: 799

range: [19840.73 , 22302.12) , numtuples: 1286

range: [22302.12 , 24763.51) , numtuples: 1413

range: [24763.51 , 27224.9) , numtuples: 1938

range: [27224.9 , 29686.29) , numtuples: 1936

range: [29686.29 , 32147.68) , numtuples: 2715

range: [32147.68 , 34609.07) , numtuples: 2740

range: [34609.07 , 37070.46) , numtuples: 3167

range: [37070.46 , 39531.85) , numtuples: 3081

range: [39531.85 , 41993.24) , numtuples: 3676

range: [41993.24 , 44454.63) , numtuples: 3331

range: [44454.63 , 46916.02) , numtuples: 3384

range: [46916.02 , 49377.41) , numtuples: 3175

range: [49377.41 , 51838.8) , numtuples: 3465

range: [51838.8 , 54300.19) , numtuples: 2996

range: [54300.19 , 56761.58) , numtuples: 2738

range: [56761.58 , 59222.97) , numtuples: 2358

range: [59222.97 , 61684.36) , numtuples: 2613

range: [61684.36 , 64145.75) , numtuples: 2199

range: [64145.75 , 66607.14) , numtuples: 1951

range: [66607.14 , 69068.53) , numtuples: 1618

range: [69068.53 , 71529.92) , numtuples: 1649

range: [71529.92 , 73991.31) , numtuples: 1500

range: [73991.31 , 76452.7) , numtuples: 1362

range: [76452.7 , 78914.09) , numtuples: 1155

range: [78914.09 , 81375.48) , numtuples: 1133

range: [81375.48 , 83836.87) , numtuples: 1115

range: [83836.87 , 86298.26) , numtuples: 960

range: [86298.26 , 88759.65) , numtuples: 817

range: [88759.65 , 91221.04) , numtuples: 773

range: [91221.04 , 93682.43) , numtuples: 741

range: [93682.43 , 96143.82) , numtuples: 696

range: [96143.82 , 98605.21) , numtuples: 613

range: [98605.21 , 101066.6) , numtuples: 593

range: [101066.6 , 103527.99) , numtuples: 624

range: [103527.99 , 105989.38) , numtuples: 502

range: [105989.38 , 108450.77) , numtuples: 403

range: [108450.77 , 110912.16) , numtuples: 405

range: [110912.16 , 113373.55) , numtuples: 379

range: [113373.55 , 115834.94) , numtuples: 294

range: [115834.94 , 118296.33) , numtuples: 292

range: [118296.33 , 120757.72) , numtuples: 262

range: [120757.72 , 123219.11) , numtuples: 273

range: [123219.11 , 125680.5) , numtuples: 218

range: [125680.5 , 128141.89) , numtuples: 211

range: [128141.89 , 130603.28) , numtuples: 159

range: [130603.28 , 133064.67) , numtuples: 148

range: [133064.67 , 135526.06) , numtuples: 121

range: [135526.06 , 137987.45) , numtuples: 122

range: [137987.45 , 140448.84) , numtuples: 117

range: [140448.84 , 142910.23) , numtuples: 108

range: [142910.23 , 145371.62) , numtuples: 83

range: [145371.62 , 147833.01) , numtuples: 91

range: [147833.01 , 150294.4) , numtuples: 87

range: [150294.4 , 152755.79) , numtuples: 105

range: [152755.79 , 155217.18) , numtuples: 68

range: [155217.18 , 157678.57) , numtuples: 69

range: [157678.57 , 160139.96) , numtuples: 52

range: [160139.96 , 162601.35) , numtuples: 59

range: [162601.35 , 165062.74) , numtuples: 49

range: [165062.74 , 167524.13) , numtuples: 44

range: [167524.13 , 169985.52) , numtuples: 44

range: [169985.52 , 172446.91) , numtuples: 36

range: [172446.91 , 174908.3) , numtuples: 22

range: [174908.3 , 177369.69) , numtuples: 40

range: [177369.69 , 179831.08) , numtuples: 23

range: [179831.08 , 182292.47) , numtuples: 22

range: [182292.47 , 184753.86) , numtuples: 18

range: [184753.86 , 187215.25) , numtuples: 15

range: [187215.25 , 189676.64) , numtuples: 21

range: [189676.64 , 192138.03) , numtuples: 20

range: [192138.03 , 194599.42) , numtuples: 6

range: [194599.42 , 197060.81) , numtuples: 19

range: [197060.81 , 199522.2) , numtuples: 11

range: [199522.2 , 201983.59) , numtuples: 14

range: [201983.59 , 204444.98) , numtuples: 19

range: [204444.98 , 206906.37) , numtuples: 16

range: [206906.37 , 209367.76) , numtuples: 16

range: [209367.76 , 211829.15) , numtuples: 10

range: [211829.15 , 214290.54) , numtuples: 10

range: [214290.54 , 216751.93) , numtuples: 10

range: [216751.93 , 219213.32) , numtuples: 7

range: [219213.32 , 221674.71) , numtuples: 3

range: [221674.71 , 224136.1) , numtuples: 2

range: [224136.1 , 226597.49) , numtuples: 2

range: [226597.49 , 229058.88) , numtuples: 4

range: [229058.88 , 231520.27) , numtuples: 2

range: [231520.27 , 233981.66) , numtuples: 5

range: [233981.66 , 236443.05) , numtuples: 4

range: [236443.05 , 238904.44) , numtuples: 3

range: [238904.44 , 241365.83) , numtuples: 2

range: [241365.83 , 243827.22) , numtuples: 2

range: [243827.22 , 246288.61) , numtuples: 4

range: [246288.61 , 248750.0) , numtuples: 4

2) equi-depth: πρόκειται για ένα τύπο ιστογράμματος, του οποίου τα bins έχουν όλα το ίδιο πλήθος στοιχείων. Επομένως προκειμένου να γίνει αυτό θα πρέπει τα στοιχεία μας να είναι διατεταγμένα σε αύξουσα σειρά προ κειμένου να ανατεθούν σωστά στα ανάλογα bins.

Αρχικά το κομμάτι της ταξινόμησης γίνεται με τη βοήθεια της βιβλιοθήκης numpy και πιο συγκεκριμένα της συνάρτησης sort().

```
#equidepth
sortedIncomes = np.sort(incomes) #Sort incomes in an ascending order, then save them in a new array.
```

Έπειτα πρέπει να εξετασθεί πόσο θα είναι το πλήθος πλειάδων σε κάθε bin. Δεδομένου ότι για το εν λόγω csv αρχείο έχουμε 72901 έγκυρα στοιχεία “Income”, η ακέραια διαίρεση τους διά 100 bins μας δίνει 729 ανά bin.

```
actualBinSize = len(sortedIncomes)/100 #Float division.
binSize = int(len(sortedIncomes)/100) #Int value of bin size.
```

Στη συνέχεια ομαδοποιούμε τα διατεταγμένα (κατά αύξουσα σειρά) αυτά στοιχεία σε 100 γκρουπ των 729 στοιχείων. Κάθε ένα από αυτά τα γκρουπ είναι ουσιαστικά και ένα bin με τα στοιχεία του. Για την εκτύπωση του ιστογράμματος χρησιμοποιούμε αντίστοιχο κομμάτι κώδικα με αυτό για το equiwidth:

```
#equidepth print
print('equidepth:')
for i in range(100):
    if i < 99:
        print('range:', '[', groupedIncomes[i][0], ',', groupedIncomes[i+1][0],')', ',', 'numtuples:', len(groupedIncomes[i]))
    else:
        print('range:', '[', groupedIncomes[i][0], ',', groupedIncomes[i][len(groupedIncomes[i])-1],']', ',', 'numtuples:', len(groupedIncomes[i]))
```

Τα αποτελέσματα του equi-depth ιστογράμματος παρατίθενται με αντίστοιχο τρόπο με αυτόν στο υπόδειγμα της εκφώνησης της άσκησης:

equidepth:

range: [2611.0 , 14814.0) , numtuples: 729

range: [14814.0 , 17456.0) , numtuples: 729

range: [17456.0 , 19731.0) , numtuples: 729

range: [19731.0 , 21265.0) , numtuples: 729

range: [21265.0 , 22475.0) , numtuples: 729

range: [22475.0 , 23874.0) , numtuples: 729

range: [23874.0 , 24978.0) , numtuples: 729

range: [24978.0 , 26015.0) , numtuples: 729

range: [26015.0 , 26923.0) , numtuples: 729

range: [26923.0 , 27765.0) , numtuples: 729

range: [27765.0 , 28750.0) , numtuples: 729

range: [28750.0 , 29620.0) , numtuples: 729

range: [29620.0 , 30345.0) , numtuples: 729

range: [30345.0 , 31047.0) , numtuples: 729

range: [31047.0 , 31679.0) , numtuples: 729

range: [31679.0 , 32267.0) , numtuples: 729

range: [32267.0 , 32870.0) , numtuples: 729

range: [32870.0 , 33583.0) , numtuples: 729

range: [33583.0 , 34223.0) , numtuples: 729

range: [34223.0 , 34884.0) , numtuples: 729

range: [34884.0 , 35466.0) , numtuples: 729

range: [35466.0 , 36020.0) , numtuples: 729

range: [36020.0 , 36581.0) , numtuples: 729

range: [36581.0 , 37123.0) , numtuples: 729

range: [37123.0 , 37683.0) , numtuples: 729

range: [37683.0 , 38294.0) , numtuples: 729

range: [38294.0 , 38879.0) , numtuples: 729

range: [38879.0 , 39461.0) , numtuples: 729

range: [39461.0 , 40000.0) , numtuples: 729

range: [40000.0 , 40496.0) , numtuples: 729

range: [40496.0 , 40990.0) , numtuples: 729

range: [40990.0 , 41462.0) , numtuples: 729

range: [41462.0 , 41905.0) , numtuples: 729

range: [41905.0 , 42344.0) , numtuples: 729

range: [42344.0 , 42891.0) , numtuples: 729

range: [42891.0 , 43484.0) , numtuples: 729

range: [43484.0 , 44007.0) , numtuples: 729

range: [44007.0 , 44591.0) , numtuples: 729

range: [44591.0 , 45119.0) , numtuples: 729

range: [45119.0 , 45643.0) , numtuples: 729

range: [45643.0 , 46188.0) , numtuples: 729

range: [46188.0 , 46715.0) , numtuples: 729

range: [46715.0 , 47222.0) , numtuples: 729

range: [47222.0 , 47774.0) , numtuples: 729

range: [47774.0 , 48366.0) , numtuples: 729

range: [48366.0 , 48917.0) , numtuples: 729

range: [48917.0 , 49554.0) , numtuples: 729

range: [49554.0 , 50125.0) , numtuples: 729

range: [50125.0 , 50618.0) , numtuples: 729

range: [50618.0 , 51094.0) , numtuples: 729

range: [51094.0 , 51590.0) , numtuples: 729

range: [51590.0 , 52107.0) , numtuples: 729

range: [52107.0 , 52636.0) , numtuples: 729

range: [52636.0 , 53274.0) , numtuples: 729

range: [53274.0 , 53917.0) , numtuples: 729

range: [53917.0 , 54583.0) , numtuples: 729

range: [54583.0 , 55240.0) , numtuples: 729

range: [55240.0 , 55882.0) , numtuples: 729

range: [55882.0 , 56507.0) , numtuples: 729

range: [56507.0 , 57192.0) , numtuples: 729

range: [57192.0 , 57925.0) , numtuples: 729

range: [57925.0 , 58750.0) , numtuples: 729

range: [58750.0 , 59545.0) , numtuples: 729

range: [59545.0 , 60266.0) , numtuples: 729

range: [60266.0 , 60880.0) , numtuples: 729

range: [60880.0 , 61573.0) , numtuples: 729

range: [61573.0 , 62257.0) , numtuples: 729

range: [62257.0 , 63125.0) , numtuples: 729

range: [63125.0 , 63988.0) , numtuples: 729

range: [63988.0 , 64932.0) , numtuples: 729

range: [64932.0 , 65833.0) , numtuples: 729

range: [65833.0 , 66732.0) , numtuples: 729

range: [66732.0 , 67775.0) , numtuples: 729

range: [67775.0 , 68951.0) , numtuples: 729

range: [68951.0 , 70117.0) , numtuples: 729

range: [70117.0 , 71146.0) , numtuples: 729

range: [71146.0 , 72222.0) , numtuples: 729

range: [72222.0 , 73413.0) , numtuples: 729

range: [73413.0 , 74760.0) , numtuples: 729

range: [74760.0 , 76065.0) , numtuples: 729

range: [76065.0 , 77448.0) , numtuples: 729

range: [77448.0 , 79196.0) , numtuples: 729

range: [79196.0 , 80833.0) , numtuples: 729

range: [80833.0 , 82241.0) , numtuples: 729

range: [82241.0 , 83929.0) , numtuples: 729

range: [83929.0 , 85772.0) , numtuples: 729

range: [85772.0 , 87750.0) , numtuples: 729

range: [87750.0 , 90259.0) , numtuples: 729

range: [90259.0 , 92476.0) , numtuples: 729

range: [92476.0 , 95146.0) , numtuples: 729

range: [95146.0 , 97679.0) , numtuples: 729

range: [97679.0 , 100847.0) , numtuples: 729

range: [100847.0 , 103750.0) , numtuples: 729

range: [103750.0 , 107703.0) , numtuples: 729

range: [107703.0 , 112150.0) , numtuples: 729

range: [112150.0 , 117984.0) , numtuples: 729

range: [117984.0 , 125156.0) , numtuples: 729

range: [125156.0 , 136250.0) , numtuples: 729

range: [136250.0 , 154583.0) , numtuples: 729

range: [154583.0 , 247868.0] , numtuples: 729

Οπτικοποίηση Ιστογραμμάτων(plots)

Προκειμένου να καταλήξουμε με σιγουριά στο ποια μέθοδος υπερτερεί της άλλης αναφορικά με την ακρίβεια έχει δημιουργηθεί και το plot της εκάστοτε προσεγγιστικής μεθόδου. Για τη δημιουργία αυτή χρησιμοποιείται η pyplot της matplotlib.

```
from matplotlib import pyplot as plt
```

Ωστόσο η matplotlib δεν έρχεται προεγκατεστημένη με την έκδοση της βασικής python3. Χρειάζεται να την εγκαταστήσουμε ξεχωριστά. Επομένως για να αποφευχθούν τυχών σφάλματα κατά την εκτέλεση στους ηλεκτρονικούς υπολογιστές της σχολής, ο κώδικας που είναι υπεύθυνος για την παραγωγή των ακόλουθων plots έχει τεθεί σε σχόλια και είναι διαθέσιμος ανά πάσα στιγμή για χρήση.

Οι παράμετροι για το plot του equiwidth:

```
# equiwidth plot creation
plt.hist(incomes, binEdges, alpha = 0.5, edgecolor = 'black', label = 'equiwidth')
```

Οι παράμετροι για το plot του equidepth:

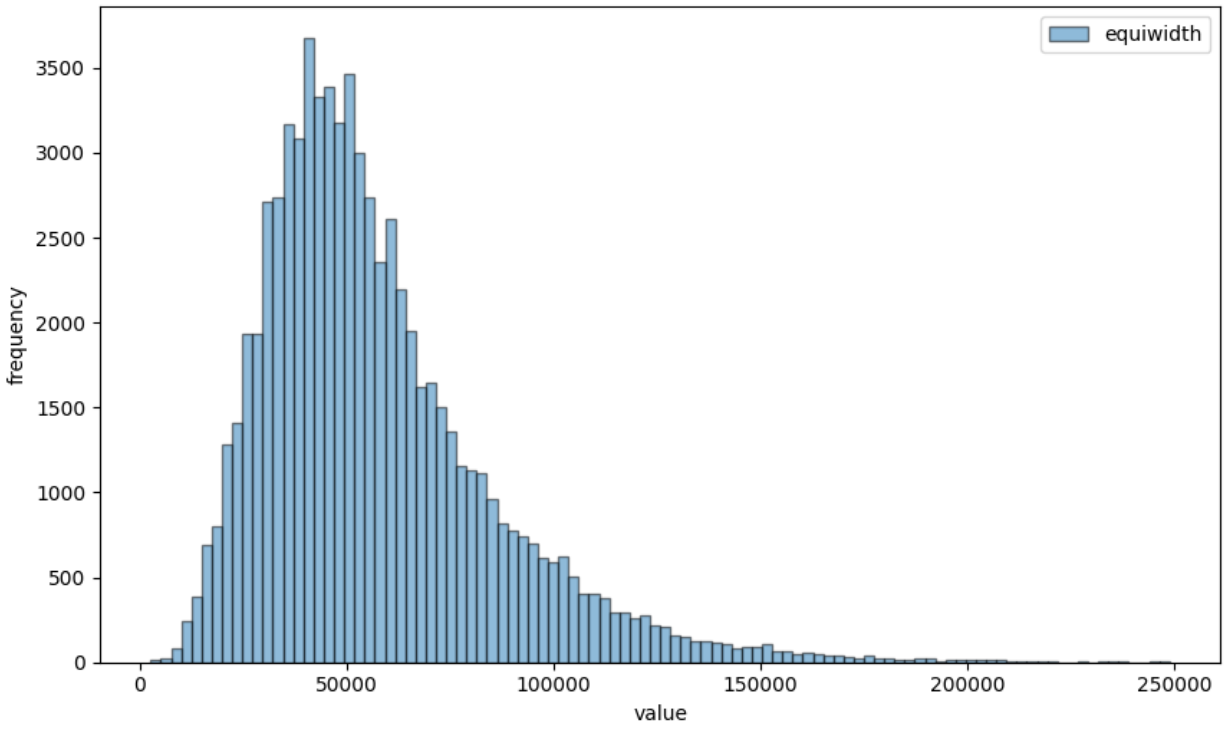
```
# equidepth plot creation
plt.hist(sortedIncomes, groupEdges, alpha = 0.5, edgecolor = 'black', label = 'equidepth')
```

Οι κοινές παράμετροι για την οπτικοποίηση και των δύο:

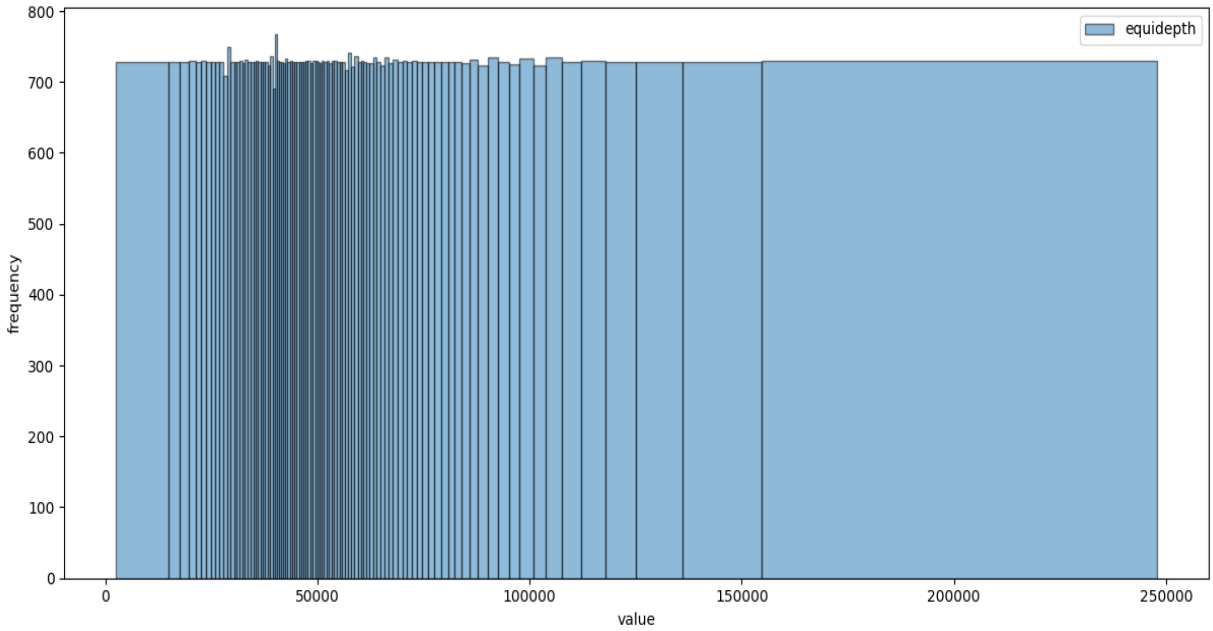
```
# diagram creation
plt.xlabel('value')
plt.ylabel('frequency')
plt.legend(loc='upper right')
plt.show()
```

Plots

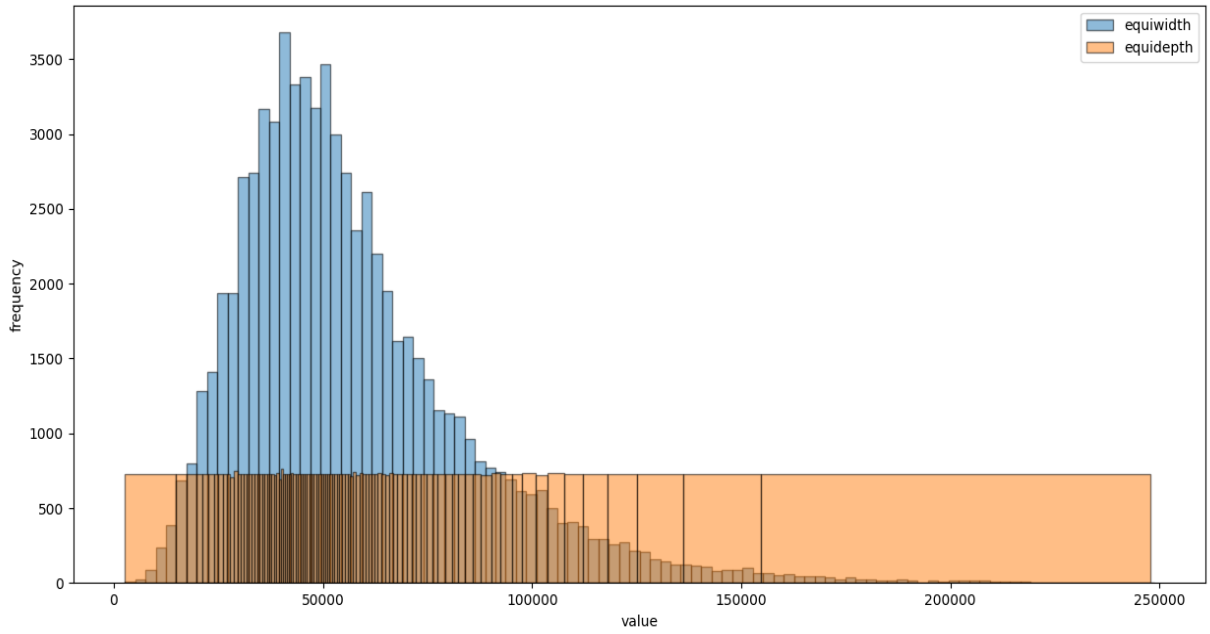
Equiwidth:



Equidepth:



Και τα δύο μαζί στους ίδιους άξονες:



Μέρος 2ο

Βλέποντας κάποιος των κώδικα του προγράμματος για το δεύτερο μέρος, διαπιστώνει πως ένα μεγάλο μέρος του είναι ακριβώς ίδιο με του πρώτου. Αυτό γιατί χρειαζόμαστε και εδώ την δημιουργία των ιστογραμμάτων, με σκοπό να τεστάρουμε την ακρίβειά τους. Για το λόγο αυτό δεν θα εξηγηθεί ξανά αυτό το κομμάτι κώδικα. Αντ' αυτού θα ξεκινήσουμε κατευθείαν με τον νέο.

Αρχικά έχουμε την είσοδο των (α,β). Ωστόσο πρέπει να υπάρχουν συγκεκριμένοι έλεγχοι εγκυρότητας, καθώς θέλουμε να δεχόμαστε μόνο θετικούς ακεραίους(φυσικούς) ως είσοδο.

```
#input
while(1):
    a = input('Give a positive integer for a:')
    try:
        a = int(a) #save input as int
        if(a < 0): #input should be a positive integer
            print('Input is not positive! Try again:')
            continue
        break
    except ValueError:
        print('Invalid input! Try again:')
while(1):
    b = input('Give a positive integer for b:')
    try:
        b = int(b) #save input as int
        if(b < 0): #input should be a positive integer
            print('Input is not positive! Try again:')
            continue
        break
    except ValueError:
        print('Invalid input! Try again:')
```

Προκειμένου να μπορούμε να ελέγξουμε εάν είναι θετικός αριθμός, χρησιμοποιούμε την συνάρτηση int(), καθώς το input από μόνο του περνιέται ως αλφαριθμητικό και δεν θα ήταν εφικτός ο έλεγχος για το αν είναι θετικός αριθμός. Έπειτα ακόμα και αν είναι "valid inputs" υπάρχει το ενδεχόμενο το a να είναι μεγαλύτερο του b. Επομένως ακολουθεί ένας δεύτερος έλεγχος και αν ισχύει ότι a > b, τότε τα κάνουμε "swap".

```
if(a > b):
    a, b = b, a #if a>b, swap a with b.
```

Επίσης επειδή κατά τη μεθοδολογία που ακολουθείται κατά τον υπολογισμό των equiwidth/ equidepth estimations τα a,b θα αλλάξουν, αντ' αυτού θα χρησιμοποιηθούν οι tempa, tempb, έτσι ώστε τα a,b να παραμείνουν άθικτα για τον υπολογισμό των "actual results".

```
equiWEstRes = 0.0
tempa = a
tempb = b
```

```
equiDEstRes = 0.0
tempa = a
tempb = b
```

<u>Μεθοδολογία για τον υπολογισμό των equiwidth estimated results:</u>

```
for i in range(100): #for every bin
    if (tempa >= binEdges[i] and tempa < binEdges[i+1]) and b < binEdges[i+1]: #check if the given range [a,b] belongs within a single bin.
        percentage = (tempb - tempa)/ (binEdges[i+1] - binEdges[i]) #calculate the percentage of the bin that is the same as the given range.
        equiWEstRes += percentage*numtuples[i] #add the new results to the existing ones.
        #print(percentage*numtuples[i])
        break #this will be the last calculation. Exit loop.
    elif (tempa >= binEdges[i] and tempa < binEdges[i+1]) and b >= binEdges[i+1]: #if the given a is within the current bin range, but b surpasses it.
        percentage = (binEdges[i+1] - tempa)/ (binEdges[i+1] - binEdges[i]) #calculate the percentage of the bin that is the same as the given range.
        equiWEstRes += percentage*numtuples[i]  #add the new results to the existing ones.
        tempa = binEdges[i+1] #set a as the current bin's second edge.
        #print(percentage*numtuples[i])
```

Μεθοδολογία για τον υπολογισμό των equidepth estimated results:

```
for i in range(100):
    print(i)
    if i == 99:
        secondEdge = groupedIncomes[i][len(groupedIncomes[i])-1] #If it's the last bin, secondEdge(b) is equal to the last element of it.
    else:
        secondEdge = groupedIncomes[i+1][0] #If it's the last bin, secondEdge(b) is equal to the first element of the next bin.

    if ((tempa >= groupedIncomes[i][0] and tempa < secondEdge) and tempb < secondEdge): #check if the given range [a,b] belongs within a single bin.
        percentage = (tempb - tempa)/ (secondEdge - groupedIncomes[i][0]) #calculate the percentage of the bin that is the same as the given range.
        equiDEstRes += percentage*len(groupedIncomes[i]) #add the new results to the existing ones.
        break #Last calculation is done. Exit loop.
    elif (tempa >= groupedIncomes[i][0] and tempa < secondEdge) and b >= secondEdge: #if the given a is within the current bin range, but b surpasses it.
        percentage = (secondEdge - tempa)/ (secondEdge - groupedIncomes[i][0]) #calculate the percentage of the bin that is the same as the given range.
        equiDEstRes += percentage*len(groupedIncomes[i]) #add the new results to the existing ones.
        tempa = secondEdge #set a as the current bin's second edge.

    #If there are duplicates of the lower edge.
    if (a >= groupedIncomes[i][0] and a < secondEdge) and i>0 and n[i-1] != 0 and actualResults > equiDEstRes:
        equiDEstRes = equiDEstRes + n[i-1] - 1

    #If there are duplicates of the higher edge, that are not included in the current bin.
    if (a >= groupedIncomes[i][0] and a < secondEdge) and b == secondEdge and actualResults < equiDEstRes:
        equiDEstRes = equiDEstRes - n[i] + 1
```

Μέτρα πρόληψης για ειδικές περιπτώσεις κατά τον υπολογισμό των equiwidth/ equidepth:

- Περίπτωση όπου το input a < χαμηλότερο 1ο άκρο **και** το b > χαμηλότερο 1ο άκρο για την equiwidth. Σε αυτή την περίπτωση υπάρχει διάστημα επικάλυψης των bins με το δοσμένο εύρος εισόδου. Προκειμένου να αποφευχθεί σφάλμα κατά τον υπολογισμό θέτουμε ως tempa = binEdges[0] (το χαμηλότερο bin όριο).

```
#If a is lower than the first edge, but b is higher.
if tempa < binEdges[0] and tempb >= binEdges[0] and actualResults != 0:
    tempa = binEdges[0]
```

- Για την equidepth. Σε αυτή την περίπτωση υπάρχει διάστημα επικάλυψης των bins με το δοσμένο εύρος εισόδου. Προκειμένου να αποφευχθεί σφάλμα κατά τον υπολογισμό θέτουμε ως tempa = groupedIncomes[0][0] (το χαμηλότερο bin όριο).

```
#If a is lower than the first edge, but b is higher.
if tempa < groupedIncomes[0][0] and tempb >= groupedIncomes[0][0] and actualResults != 0:
    tempa = groupedIncomes[0][0]
```

Σκεπτικό πίσω από την υλοποίηση για την equiwidth: Έστω το παράδειγμα εξόδου της εκφώνησης για α=19000 και β=55000. Τα bins τα οποία μας αφορούν είναι τα:

range: [17379.34 , 19840.73) , numtuples: 799

range: [19840.73 , 22302.12) , numtuples: 1286

range: [22302.12 , 24763.51) , numtuples: 1413

range: [24763.51 , 27224.9) , numtuples: 1938

range: [27224.9 , 29686.29) , numtuples: 1936

range: [29686.29 , 32147.68) , numtuples: 2715

range: [32147.68 , 34609.07) , numtuples: 2740

range: [34609.07 , 37070.46) , numtuples: 3167

range: [37070.46 , 39531.85) , numtuples: 3081

range: [39531.85 , 41993.24) , numtuples: 3676

range: [41993.24 , 44454.63) , numtuples: 3331

range: [44454.63 , 46916.02) , numtuples: 3384

range: [46916.02 , 49377.41) , numtuples: 3175

range: [49377.41 , 51838.8) , numtuples: 3465

range: [51838.8 , 54300.19) , numtuples: 2996

range: [54300.19 , 56761.58) , numtuples: 2738

Στο 1^{ο} bin μας ενδιαφέρει μόνο το διάστημα 19000~19840.73. Μετά τον υπολογισμό της διαφοράς 19840.73 – 19000 = 840.73. Το εύρος του bin είναι 19840.73 – 17379.34 = 2461.39. Επομένως το percentage = 840.73/ 2461.39 = 0.341567. Άρα το προσεγγιστικό πλήθος των equiwidth results για το συγκεκριμένο bin θα είναι 799* 0.341567 = 272.912033. Για τα υπόλοιπα ενδιάμεσα είναι ολόκληρο το πλήθος πλειάδων. Κάθε φορά θέτουμε ως tempa το 2^{ο} binEdge, έτσι ώστε να χρησιμοποιηθεί κατά τον υπολογισμό του επόμενου bin και για να μπορούμε να μπούμε στο επόμενο if/ elif condition. Το τελευταίο είναι πάλι κατά αντιστοιχία με το πρώτο. Βλέποντας τον κώδικα για το input a=19000, b=55000 από 19000~54300.19 θα μπαίνουμε στο elif. Στο τελευταίο bin θα μπούμε στο if condition.

Σκεπτικό πίσω από την υλοποίηση για την equidepth: Είναι η ίδια λογική μόνο που το εύρος του εκάστοτε bin δεν είναι ίδιο για όλα και το πλήθος numtuples είναι ίδιο και ίσο με 729(για το συγκεκριμένο csv αρχείο που αναφέρθηκε στην αρχή).

Για την εκτύπωση των αποτελεσμάτων είναι ισχύει ο ακόλουθος κώδικας:

```
print('equiwidth estimated results:', equiWEstRes)
print('equidepth estimated results:', equiDEstRes)

actualResults = 0
for i in incomes:
    if i >= a and i < b:
        actualResults += 1
print('actual results:', actualResults)
```

Τα αποτελέσματα για το εν λόγω csv και inputs a = 19000 και b = 55000:

```
Give a positive integer for a:19000
Give a positive integer for b:55000
equiwidth estimated results: 39354.366524606026
equidepth estimated results: 39333.939948818304
actual results: 39361
```

Άλλα αποτελέσματα και για τα 4 csv αρχεία που μας δόθηκαν για inputs a=19000 και b=55000:

1. acs2015 census:

   a=19000, b=55000

   equiwidth estimated results: 39354.366524606026

   equidepth estimated results: 39333.939948818304

   actual results: 39361

   a=2611, b=248750

   equiwidth estimated results: 72901.0

equidepth estimated results: 72900.0

actual results: 72900

a=0, b=2611

equiwidth estimated results: 0.0

equidepth estimated results: 0.0

actual results: 0

a=400, b=3000

equiwidth estimated results: 2.054530163850508

equidepth estimated results: 23.23862984512005

actual results: 1

a=1500, b=5000

equiwidth estimated results: 12.617667253056204

equidepth estimated results: 142.71744652954192

actual results: 12

a=41905, b=42344

equiwidth estimated results: 606.4669962907136

equidepth estimated results: 729.0

actual results: 724

a=42344, b=42891

equiwidth estimated results: 740.2553028979562

equidepth estimated results: 729.0

actual results: 733

a=23456, b=98765

equiwidth estimated results: 62384.09360158285

equidepth estimated results: 62432.71756205732

actual results: 62416

a=2610, b=2612

equiwidth estimated results: 0.005281568544602848

equidepth estimated results: 0.059739408342210934

actual results: 1

a=248749, b=248751

equiwidth estimated results: 0.0016250980137239436

equidepth estimated results: 0.0

actual results: 1

a=25000, b=75000

equiwidth estimated results: 52601.95168583605

equidepth estimated results: 52606.603198882716

actual results: 52599

2. acs2017 census:

   a=19000, b=55000

   equiwidth estimated results: 35548.0734402448

   equidepth estimated results: 35602.14949317666

   actual results: 35613

   a=0, b=5000

   equiwidth estimated results: 10.27612949186021

equidepth estimated results: 128.50661567877628

actual results: 8

a=0, b=150000

equiwidth estimated results: 71614.10925369752

equidepth estimated results: 71593.58505031145

actual results: 71607

3. acs2015 county:

   a=19000, b=55000

   equiwidth estimated results: 2572.445154664731

   equidepth estimated results: 2575.244511278195

   actual results: 2575

   a=0, b=5000

   equiwidth estimated results: 0.0

   equidepth estimated results: 0.0

   actual results: 0

   a=0, b=100000

   equiwidth estimated results: 3208.473360837155

   equidepth estimated results: 3200.0

   actual results: 3208

   a=10000, b=60000

   equiwidth estimated results: 2857.7210368822707

   equidepth estimated results: 2857.2631578947367

actual results: 2854

4. acs2017 county:
   a=19000, b=55000

   equiwidth estimated results: 2326.22814397666

   equidepth estimated results: 2325.988096994737

   actual results: 2322

   a=0, b=5000

   equiwidth estimated results: 0.0

   equidepth estimated results: 0.0

   actual results: 0

   a=0, b=100000

   equiwidth estimated results: 3198.9058588051703

   equidepth estimated results: 3199.310695136864

   actual results: 3199

Συμπέρασμα

Ύστερα από μία πληθώρα εισόδων a, b και για τα 4 datasets που μας παρήχθησαν στα πλαίσια της συγκεκριμένης εργασίας, γεννάται το εξής συμπέρασμα:

Δεν υπάρχει αδιαμφησβήτητος "νικητής" μεταξύ των equiwidth/ equidepth ιστογραμμάτων, αναφορικά με την ακρίβεια. Η επιλογή του ενός εκ των δύο και η απόρριψη του άλλου έχουν να κάνουν με τις ανάγκες που έχουμε ανάλογα με το dataset το οποίο πρόκειται να αναλύσουμε. Αυτό γίνεται πιο προφανές με τα plots που παρατίθενται παραπάνω σε συνδυασμό με τα διαφορετικά inputs/ datasets που χρησιμοποιήθηκαν. Ουσιαστικά, η χρησιμότητα του καθενός εκ των ιστογραμμάτων είναι άρρηκτα συνδεδεμένη με τις ιδιαιτερότητες/ χαρακτηριστικά του.

- Το equiwidth χωρίζει τα δεδομένα σε bins ίσου εύρους. Επομένως καθίσταται πιο χρήσιμο και ακριβές όταν η κατανομή των στοιχείων του dataset είναι "σχετικά" ομοιόμορφη. Ωστόσο παρατηρούμε και από τα διαγράμματα παραπάνω πως η χρήση του equiwidth μπορεί να οδηγήσει σε δημιουργία bins που έχουν πάρα πολλά ή ελάχιστα στοιχεία. Έχει να κάνει καθαρά και μόνο με την κατανομή των στοιχείων του εκάστοτε dataset που τίθεται προς ανάλυση. Επίσης από τα plots παραπάνω είναι εμφανές ότι ένα equiwidth διάγραμμα είναι και πιο ευανάγνωστο.
- Το equidepth χωρίζει τα δεδομένα σε bins με "σχετικά ίσο" αριθμό στοιχείων. Επομένως είναι πολύ πιο ακριβές όταν η κατανομή των στοιχείων είναι ανομοιόμορφη, καθώς και όταν θέλουμε ακρίβεια εντός των ορίων ενός bin ανεξαρτήτως της κατανομής των στοιχείων σε εκείνη την περιοχή.

Συμπερασματικά, η επιλογή ενός εκ των δύο, αναφορικά με την ακρίβεια, έχει να κάνει με τους εξής παράγοντες:

1. Την κατανομή των δεδομένων του εκάστοτε dataset.
2. Τα inputs(περιοχή) που θέλουμε να αναλύσουμε.