मौलाना आज़ाद
राष्ट्रीय प्रौद्योगिकी संस्थान भोपाल (म.प्र.) भारत
MAULANA AZAD
NATIONAL INSTITUTE OF TECHNOLOGY BHOPAL (M. P.) INDIA

Data Warehousing & Mining Lab Assignment

Lab - 5

Sub Code: CSE-326

Date: 16-02-2022

Vivek Kumar Ahirwar

191112419

CSE - 3

Department:

Computer Science and Engineering

Vivek Kumar Ahirwar 191112419

Assignment - 5

Assignment Problem

Q1: Using the data for age and fare attribute given in Titanic dataset,

```
a) WAP to Plot histogram using singleton bucket.
b) WAP to Plot an equal-width histogram of width 10.
```

Q2: Using the data for age attribute given in Titanic dataset, WAP to perform following sampling techniques (Select 30% samples with following methods)

```
a) Simple Random Sampling With Replacement.
b) Simple Random Sampling Without Replacement.
c) Stratified Sampling. (use three intervals as per the range of attribute)
d) Calculate mean and standard deviation after sampling and compare it with mean
and standard deviation of original data.
```

Q3: Using the data for age and fare attribute given in Titanic dataset,

```
a) WAP for min-max normalization onto the range [0, 1].
b) WAP for z-score normalization.
c) WAP to perform decimal scaling.
d) Calculate mean and standard deviation after all types of normalization and
compare it with mean and standard deviation of original data.
```

Approach Used

Using dataframe of pandas library to store the excel titanic data and numpy libraries to perform basic functions such as mean, median, mode, sum, etc. Also, pyplot to plot the various data generated in the process.

Code & Output

Import all libraries

In [1]:

```
import math
import random
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Import the data from the excel sheet

In [2]:

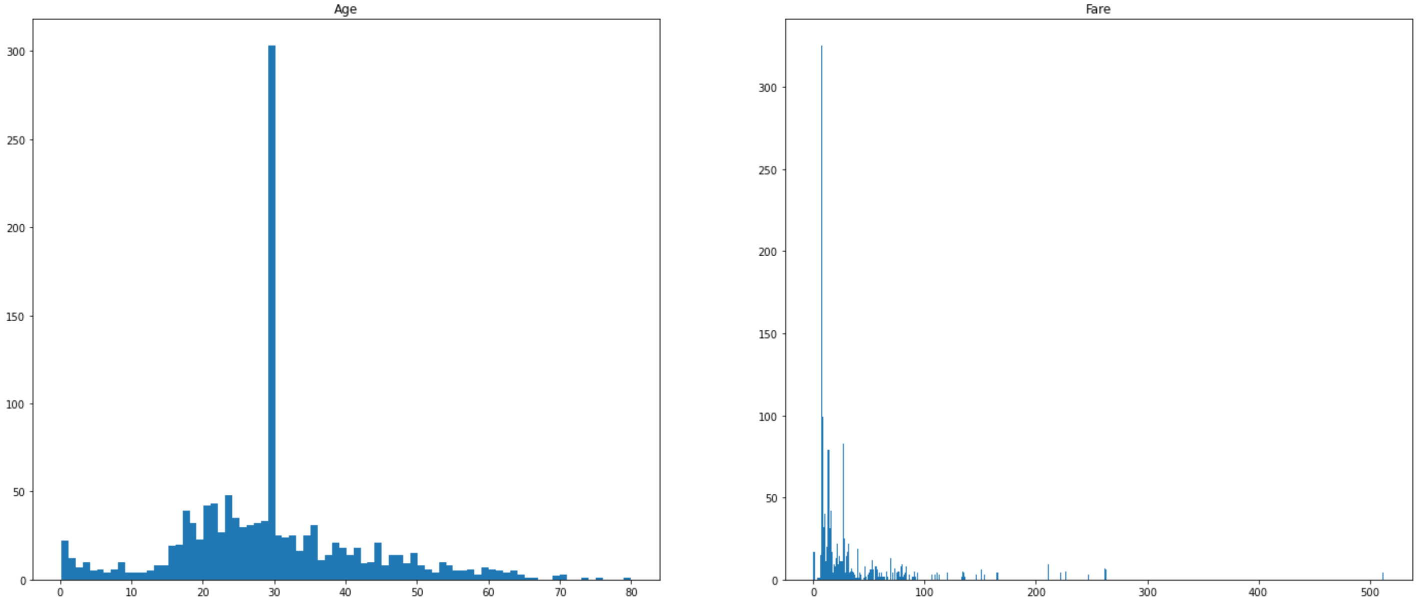
```
data = pd.read_excel('titanic.xls')
display(data)
```

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	embarked
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	S
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	S
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	S
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	S
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	S
...	...	...	...	...	...	...	...	...	...	...
1304	3	0	Zabour, Miss. Hileni	female	14.5000	1	0	2665	14.4542	C
1305	3	0	Zabour, Miss. Thamine	female	NaN	1	0	2665	14.4542	C
1306	3	0	Zakarian, Mr. Mapriededer	male	26.5000	0	0	2656	7.2250	C
1307	3	0	Zakarian, Mr. Ortin	male	27.0000	0	0	2670	7.2250	C
1308	3	0	Zimmerman, Mr. Leo	male	29.0000	0	0	315082	7.8750	S

1309 rows × 10 columns

In [3]:

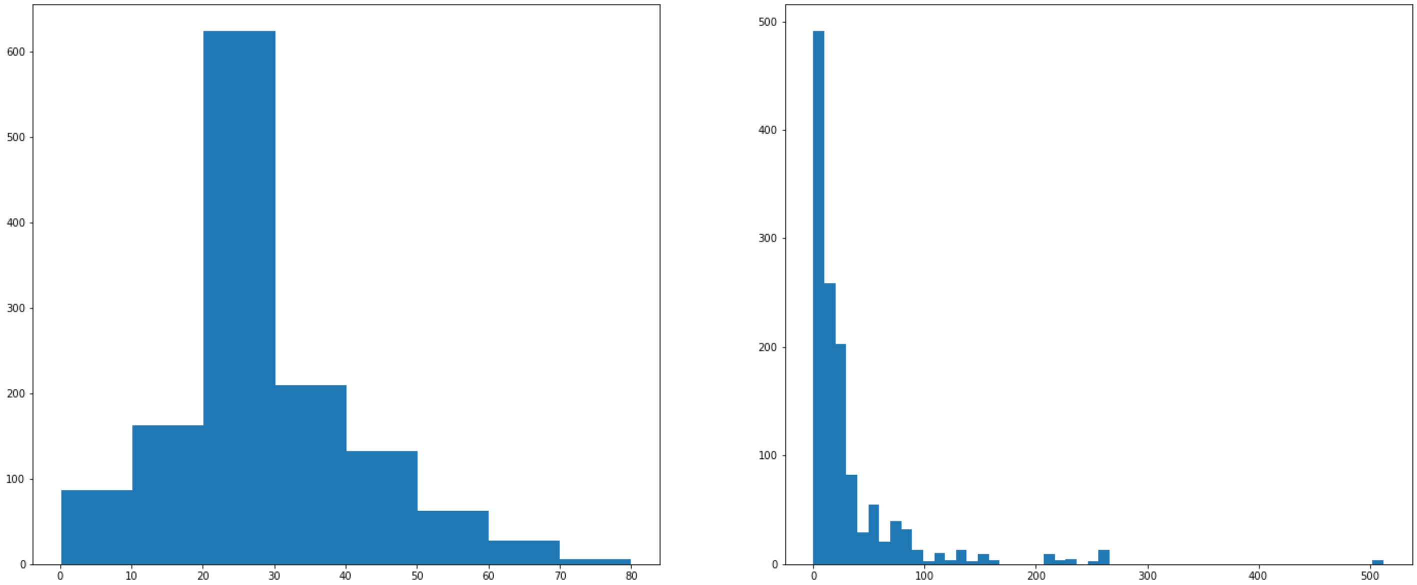
```
data['age'].fillna(value=data['age'].mean(axis=0, skipna=True), inplace=True)
data['fare'].fillna(value=data['fare'].mean(axis=0, skipna=True), inplace=True)

fig, axs = plt.subplots(1, 2, figsize=(24, 10))
axs[0].hist(data['age'], bins=math.ceil(data['age'].max()))
axs[0].set_title('Age')
axs[1].hist(data['fare'], bins=math.ceil(data['fare'].max()))
axs[1].set_title('Fare')
plt.show()
```


Age

Fare

In [4]:

```
fig, axs = plt.subplots(1, 2, figsize=(24, 10))
axs[0].hist(data['age'], bins=math.ceil(data['age'].max()/10))
axs[1].hist(data['fare'], bins=math.ceil(data['fare'].max()/10))
plt.show()
```



In [5]:

```
def random_sampling_replacement(arr):
    sz = len(arr)
    sample_size = math.ceil(sz * 0.3)
    sample = []
    for i in range(sample_size + 1):
        sample.append(arr[random.randint(0, sz-1)])
    return sample


def random_sampling_noreplacement(arr):
    sz = len(arr)
    sample_size = math.ceil(sz * 0.3)
    sample_index = []
    sample = []
    for i in range(sample_size + 1):
        index = random.randint(0, sz-1)
        if index not in sample_index:
            sample_index.append(index)
            sample.append(arr[index])
    return sample


def stratified_sampling(arr):
    sz = len(arr)
    clusters = [[], [], []]
    mn = min(arr)
    mx = max(arr)
    w = (mx - mn) / 3
    for i in range(sz):
        if arr[i] <= mn + w:
            clusters[0].append(arr[i])
        elif arr[i] <= mn + 2 * w:
            clusters[1].append(arr[i])
        elif arr[i] <= mn + 3 * w:
            clusters[2].append(arr[i])
    sample = []
    for i in range(3):
        sz = len(clusters[i])
        sample_size = math.ceil(sz * 0.3)
        sample_index = []
        for j in range(sample_size + 1):
            index = random.randint(0, sz-1)
            if index not in sample_index:
                sample_index.append(index)
                sample.append(clusters[i][index])
    return sample
```

In [6]:

```
def calculation(x):
    sz = len(x)
    xm = round(sum(x)/sz, 3)
    s = 0
    for i in range(sz):
        s += ((x[i] - xm)**2)
    xvar = s / sz
    xdev = xvar**0.5
    return (xm, round(xdev, 3))
```

In [7]:

```
sample_random_replacement = random_sampling_replacement(list(data['age']))
sample_random_noreplacement = random_sampling_noreplacement(list(data['age']))
sample_stratified = stratified_sampling(list(data['age']))

print(f"Sample type
      (Mean, Standard Deviation)", end='\n\n')
print(f"Simple Random Sampling With Replacement
      {calculation(sample_random_replacement)}")
print(f"Simple Random Sampling Without Replacement
      {calculation(sample_random_noreplacement)}")
print(f"Stratified Sampling
      {calculation(sample_stratified)}")
print(f"Original Data
      {calculation(list(data['age']))}")
```

```
Sample type                                 (Mean, Standard Deviation)

Simple Random Sampling With Replacement     (29.529, 13.516)
Simple Random Sampling Without Replacement  (30.115, 13.315)
Stratified Sampling                         (30.511, 13.196)
Original Data                               (29.881, 12.878)
```

In [8]:

```
def min_max_normalize(x):
    mn = min(x)
    mx = max(x)
    for i in range(len(x)):
        x[i] = (x[i] - mn) / (mx - mn)
    return x


def z_score_normalize(x):
    mean, stddev = calculation(x)
    for i in range(len(x)):
        x[i] = (x[i] - mean) / stddev
    return x


def decimal_scaling(x):
    num = math.floor(math.log(max(x), 10)) + 1
    for i in range(len(x)):
        x[i] /= num
    return x
```

In [9]:

```
min_max_data = min_max_normalize(list(data['age']))
z_score_data = z_score_normalize(list(data['age']))
decimal_data = decimal_scaling(list(data['age']))
print(f"Normalization                    (Mean, Standard Deviation)", end='\n\n')
print(f"Min Max Normalization            {calculation(min_max_data)}")
print(f"Z Score Normalization            {calculation(z_score_data)}")
print(f"Decimal Scaling                  {calculation(decimal_data)}")
print(f"Original Data                    {calculation(list(data['age']))}")
```

```
Normalization                 (Mean, Standard Deviation)

Min Max Normalization         (0.372, 0.161)
Z Score Normalization         (0.0, 1.0)
Decimal Scaling               (14.941, 6.439)
Original Data                 (29.881, 12.878)
```