

## Make sense of Census

### Instructions :

**Step 1:** In this first task, we will load the data to a numpy array and add a new record to it.

- The path to the data set has been stored in the variable named `path`
- Load the dataset and store it in a variable called `data` using `np.genfromtxt()`

---

Example of `genfromtxt` function

---

```
data_file='file.csv' # path for the file
data=np.genfromtxt(data_file, delimiter=",", skip_header=1)

print("\nData: \n\n", data)

print("\nType of data: \n\n", type(data))
```

### Output

Data:

```
[39. 13.  4. ...  0. 40.  0.]
[50. 13.  4. ...  0. 13.  0.]
[38.  9.  4. ...  0. 40.  0.]
...
[48. 13.  4. ...  0. 58.  1.]
[40. 10.  4. ...  0. 40.  0.]
[39. 13.  4. ...  0. 50.  1.]]
```

Type of data:

```
<class 'numpy.ndarray'>
```

---

- Append '`new_record`' (given) to '`data`' using "`np.concatenate()`" and store the new array in a variable called `census`

```
new_record=[[50, 9, 4, 1, 0, 0, 40, 0]]
```

---

The shape of `data` should be (1000, 8) and that of `census` should be (1001,8).

---

**Step 2:** We often associate the potential of a country based on the age distribution of the people residing there. We too want to do a simple analysis of the age distribution

### Instructions :

- Create a new array called '`age`' by taking only age column(age is the column with index 0) of '`census`' array.
- Find the max age and store it in a variable called '`max_age`'.
- Find the min age and store it in a variable called '`min_age`'.
- Find the mean of the age and store it in a variable called '`age_mean`'.
- Find the standard deviation of the age and store it in a variable called '`age_std`'.

**Ponder whether based on the above statistics, would you classify the country as 'young' or 'old'?**

---

`max_age` should be 90.

`min_age` should be 17.

`age_mean` should be 38.06.

`age_std` should be 13.34.

---

**Step 3:** The constitution of the country tries it's best to ensure that people of all races are able to live harmoniously. Let's check the country's race distribution to identify the minorities so that the government can help them.

- Create four different arrays by subsetting 'census' array by Race column(Race is the column with index 2) and save them in 'race\_0','race\_1','race\_2','race\_3' and 'race\_4' respectively(Meaning: Store the array where 'race' column has value 0 in 'race\_0', so on and so forth)
- Store the length of the above created arrays in 'len\_0','len\_1','len\_2','len\_3' and 'len\_4' respectively
- Find out which is the race with the minimum no. of citizens
- Store the number associated with the minority race in a variable called 'minority\_race'(For eg: if "len(race\_5)" is the minimum, store 5 in 'minority\_race' because that is the index of the race having the least no. of citizens )

---

minority\_race should be 3.

---

**Step 4:** As per the new govt. policy, all citizens above age 60 should not be made to work more than 25 hours per week. Let us look at the data and see if that policy is followed.

- Create a new subset array called 'senior\_citizens' by filtering 'census' according to age>60 (age is the column with index 0)
- Add all the working hours(working hours is the column with index 6) of 'senior\_citizens' and store it in a variable called 'working\_hours\_sum'
- Find the length of 'senior\_citizens' and store it in a variable called 'senior\_citizens\_len'
- Finally find the average working hours of the senior citizens by dividing 'working\_hours\_sum' by 'senior\_citizens\_len' and store it in a variable called 'avg\_working\_hours'.
- Print 'avg\_working\_hours' and see if the govt. policy is followed.

---

working\_hours\_sum should be 1917.

avg\_working\_hours should be 31.43.

---