harshitgupta5 Created using Colaboratory History

1 contributor

Raw Blame

1415 lines (1415 sloc) 48.3 KB

Aggregations

Harshit Gupta

Step 1. Import the necessary libraries

In [1]:
```
import pandas as pd
```

Step 2. Import the dataset occupation.csv from the folder

In [31]:
```
df = pd.read_csv('occupation.csv',delimiter = '|')
```

Step 3. Assign it to a variable called users.

In [32]:
```
users = df
users.head()
```

Out[32]:

	user_id	age	gender	occupation	zip_code
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213

Step 4. Discover what is the mean age per occupation

In [7]:
```
users.groupby('occupation').mean()
```

Out[7]:

	user_id	age
occupation		
administrator	430.949367	38.746835
artist	451.892857	31.392857
doctor	533.714286	43.571429
educator	466.905263	42.010526
engineer	456.328358	36.388060
entertainment	398.000000	29.222222

executive	422.312500	38.718750
healthcare	501.437500	41.562500
homemaker	443.000000	32.571429
lawyer	359.083333	36.750000
librarian	486.588235	40.000000
marketing	437.807692	37.615385
none	368.666667	26.555556
other	542.733333	34.523810
programmer	435.530303	33.121212
retired	515.714286	63.071429
salesman	494.916667	35.666667
scientist	465.129032	35.548387
student	484.954082	22.081633
technician	497.629630	33.148148
writer	495.711111	36.311111

Step 5. Discover the Male ratio per occupation and sort it from the most to the least.

Use numpy.where() to encode gender column.

In [37]:
```
#This task can be completed without encoding the gender column
so np.where() is not used
user_by_gender = df.pivot_table(index='occupation',columns =
'gender',aggfunc = 'size',fill_value = 0)
sums = user_by_gender[['F','M']].sum(axis = 1)
user_by_gender['Male Ratio'] = user_by_gender['M']/sums
user_by_gender
```

Out[37]:

gender	F	M	Male Ratio
occupation			
administrator	36	43	0.544304
artist	13	15	0.535714
doctor	0	7	1.000000
educator	26	69	0.726316
engineer	2	65	0.970149
entertainment	2	16	0.888889

executive	3	29	0.906250
healthcare	11	5	0.312500
homemaker	6	1	0.142857
lawyer	2	10	0.833333
librarian	29	22	0.431373
marketing	10	16	0.615385
none	4	5	0.555556
other	36	69	0.657143
programmer	6	60	0.909091
retired	1	13	0.928571
salesman	3	9	0.750000
scientist	3	28	0.903226
student	60	136	0.693878
technician	1	26	0.962963
writer	19	26	0.577778

0 For Female 1 For Male

Step 6. For each occupation, calculate the minimum and maximum ages

In [26]:
```
users[['age','occupation']].groupby('occupation').agg([min,max])
```

Out[26]:

	age	
	min	**max**
occupation		
administrator	21	70
artist	19	48
doctor	28	64
educator	23	63
engineer	22	70
entertainment	15	50
executive	22	69
healthcare	22	62
homemaker	20	50

lawyer	21	53
librarian	23	69
marketing	24	55
none	11	55
other	13	64
programmer	20	63
retired	51	73
salesman	18	66
scientist	23	55
student	7	42
technician	21	55
writer	18	60

Step 7. For each combination of occupation and gender, calculate the mean age

In [27]:
```
users[['occupation','gender','age']].groupby(['occupation','gender']).mean()
```

Out[27]:

		age
occupation	**gender**	
administrator	**F**	40.638889
	M	37.162791
artist	**F**	30.307692
	M	32.333333
doctor	**M**	43.571429
educator	**F**	39.115385
	M	43.101449
engineer	**F**	29.500000
	M	36.600000
entertainment	**F**	31.000000
	M	29.000000
executive	**F**	44.000000
	M	38.172414
healthcare	**F**	39.818182

healthcare	**M**	45.400000
homemaker	**F**	34.166667
	M	23.000000
lawyer	**F**	39.500000
	M	36.200000
librarian	**F**	40.000000
	M	40.000000
marketing	**F**	37.200000
	M	37.875000
none	**F**	36.500000
	M	18.600000
other	**F**	35.472222
	M	34.028986
programmer	**F**	32.166667
	M	33.216667
retired	**F**	70.000000
	M	62.538462
salesman	**F**	27.000000
	M	38.555556
scientist	**F**	28.333333
	M	36.321429
student	**F**	20.750000
	M	22.669118
technician	**F**	38.000000
	M	32.961538
writer	**F**	37.631579
	M	35.346154

Step 8. For each occupation present the percentage of women and men

In [38]:
```
user_by_gender = df.pivot_table(index='occupation',columns =
'gender',aggfunc = 'size',fill_value = 0)
sums = user_by_gender[['F','M']].sum(axis = 1)
user_by_gender['Female %age'] = user_by_gender['F']/sums*100
user_by_gender['Male %age'] = user_by_gender['M']/sums*100
```

```
user_by_gender['Male %age'] = user_by_gender['M']/sums*100
user_by_gender
```

Out[38]:

gender	F	M	Female %age	Male %age
occupation				
administrator	36	43	45.569620	54.430380
artist	13	15	46.428571	53.571429
doctor	0	7	0.000000	100.000000
educator	26	69	27.368421	72.631579
engineer	2	65	2.985075	97.014925
entertainment	2	16	11.111111	88.888889
executive	3	29	9.375000	90.625000