Untitled

May 9, 2023

[2]:
```
#import codecademylib3

# Import pandas with alias
import pandas as pd
```

[4]:
```
# Read in the census dataframe
#census = pd.read_csv('census_data.csv', index_col=0)

#1
#print(census.head())

#2
#variable type assessment:
#names : nominal
#birth_year, marital_status, higher_tax : ordinal
#voted : binary
#num_children, income_year: discrete

#3
#print(census.dtypes)

#first_name, last_name are both objects and should be strings
#may make additional changes to ordinal varibales in making them 'category'␣
 ↪types or do a OHE and disregard ranking
```

[5]:
```
#4
#desire to take the average of the birth_year values
#thus we aim to reassign its data type to int

#print(census['birth_year'].unique())

#5
#replace 'missing' value for 1967 (integer)

#census['birth_year']=census['birth_year'].replace('missing', 1967)
```

```
#print(census['birth_year'].unique())

#6
#switch the data types from str to int
#census['birth_year'] = census['birth_year'].astype('int')
#print(census.dtypes)

#7
#calculating average with pandas method '.mean()'
#average = census.mean()
#birthyear_avg=average['birth_year']
#print(birthyear_avg)

#print(census['birth_year'].mean())
```

[6]:
```
#8
#set a ranking to the higher tax variable
#s-disagree < disagree < neutral < agree < s-agree

#census['higher_tax'] = pd.Categorical(census['higher_tax'], ['strongly␣
 ↪disagree', 'disagree', 'neutral', 'agree', 'strongly agree'], ordered=True)

#print(census['higher_tax'].unique())

#9
#desire median value under higher_tax variable
#need to label encode higher_tax and call .median() method

#census['higher_tax_codes']=census['higher_tax'].cat.codes
#median_sentiment=census['higher_tax_codes'].median()
#print(median_sentiment)
#median sentiment is neutral

#10

#census = pd.get_dummies(data=census, columns=["marital_status"])
#print(census.head())

#11

#marital_codes label encoding
#census["marital_status"] = pd.Categorical(census["marital_status"],␣
 ↪["widowed", "divorced", "single", "married"], ordered=True)
#census["marital_codes"]=census["marital_status"].cat.codes
#print(census.head())

#age_group
```

```
#print(census["birth_year"].unique())
ages=[]
age_groups=[]
#for i in census["birth_year"]:
#  ages.append(2023-i)
for i in ages:
  if i < 6:
    age_groups.append('0-5')
  if i < 11 and i > 5:
    age_groups.append('6-10')
  if i < 16 and i > 10:
    age_groups.append('11-15')
  if i < 21 and i > 15:
    age_groups.append('16-20')
  if i < 26 and i > 20:
    age_groups.append('21-25')
  if i < 31 and i > 25:
    age_groups.append('26-30')
  if i < 36 and i > 30:
    age_groups.append('31-35')
  if i < 41 and i > 35:
    age_groups.append('36-40')
  if i < 46 and i > 40:
    age_groups.append('41-45')
  if i < 51 and i > 45:
    age_groups.append('46-50')
  if i < 56 and i > 50:
    age_groups.append('51-55')
  if i < 61 and i > 55:
    age_groups.append('56-60')
  if i < 66 and i > 60:
    age_groups.append('61-65')
  if i < 71 and i > 65:
    age_groups.append('66-70')
  if i < 76 and i > 70:
    age_groups.append('71-75')
  if i < 81 and i > 75:
    age_groups.append('76-80')
  if i < 86 and i > 80:
    age_groups.append('81-85')
  if i < 91 and i > 85:
    age_groups.append('86-90')
  if i > 90:
    age_groups.append('90+')

#census["age_groups"] = age_groups
```

```
#print(census.head())
```

[ ]: