CSE578FinalProject

February 27, 2024

CSE578 - DATA VISULIZATION - FINAL

[1]:
```
import numpy as np
import pandas as pd
pd.options.mode.chained_assignment = None  # default='warn'
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import math
from statsmodels.graphics.mosaicplot import mosaic
from itertools import product
import plotly.express as px
from pandas.plotting import parallel_coordinates, scatter_matrix
import plotly.offline as pyo
pyo.init_notebook_mode()
```

[ ]:

[ ]:

[2]:
```
#df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/
 ↪adult/adult.data', header=None)
df = pd.read_csv('/Desktop/Krithi/ASU/CSE578_Data_Visualization/project/
 ↪adult_data', header = None)
```

[3]:
```
df.head()
```

[3]:

	0	1	2	3	4	5	\
0	39	State-gov	77516	Bachelors	13	Never-married	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	
2	38	Private	215646	HS-grad	9	Divorced	
3	53	Private	234721	11th	7	Married-civ-spouse	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	

	6	7	8	9	10	11	12	\
0	Adm-clerical	Not-in-family	White	Male	2174	0	40	
1	Exec-managerial	Husband	White	Male	0	0	13	
2	Handlers-cleaners	Not-in-family	White	Male	0	0	40	

```
3   Handlers-cleaners         Husband   Black     Male     0   0  40
4      Prof-specialty            Wife   Black   Female     0   0  40

                13      14
0   United-States   <=50K
1   United-States   <=50K
2   United-States   <=50K
3   United-States   <=50K
4            Cuba   <=50K
```

[4]:
```
len(df.index)
```

[4]:
```
32561
```

[5]:
```
df.describe()
```

[5]:
```
                 0             2             4             10            11  \
count  32561.000000  3.256100e+04  32561.000000  32561.000000  32561.000000
mean      38.581647  1.897784e+05     10.080679   1077.648844     87.303830
std       13.640433  1.055500e+05      2.572720   7385.292085    402.960219
min       17.000000  1.228500e+04      1.000000      0.000000      0.000000
25%       28.000000  1.178270e+05      9.000000      0.000000      0.000000
50%       37.000000  1.783560e+05     10.000000      0.000000      0.000000
75%       48.000000  2.370510e+05     12.000000      0.000000      0.000000
max       90.000000  1.484705e+06     16.000000  99999.000000   4356.000000

                12
count  32561.000000
mean      40.437456
std       12.347429
min        1.000000
25%       40.000000
50%       40.000000
75%       45.000000
max       99.000000
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       32561 non-null  int64
 1   1       32561 non-null  object
 2   2       32561 non-null  int64
 3   3       32561 non-null  object
 4   4       32561 non-null  int64
```

```
 5   5       32561 non-null  object
 6   6       32561 non-null  object
 7   7       32561 non-null  object
 8   8       32561 non-null  object
 9   9       32561 non-null  object
 10  10      32561 non-null  int64
 11  11      32561 non-null  int64
 12  12      32561 non-null  int64
 13  13      32561 non-null  object
 14  14      32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

[7]:
```
df.isnull()
```

[7]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
32556	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
32557	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
32558	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
32559	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
32560	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

[32561 rows x 15 columns]

[8]:
```
df.shape
```

[8]:
```
(32561, 15)
```

[9]:
```
df.columns
```

[9]:
```
Index([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], dtype='int64')
```

[10]:
```
df.dtypes
df.columns
```

[10]:
```
Index([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], dtype='int64')
```

[11]:
```
df.rename(

    columns = {0: "age", 1: "workclass", 2: "workid",3: "Education" ,4:␣
 ↪"Education_number",5: "martialstatus",6: "occupation",
               7: "relationship", 8:"race",9:"gender",10: "capitalgain",11:␣
 ↪"capitalloss" ,12: "hoursPerweek",13: "nativecountry",
               14: "income"
              },inplace = True,
)
```

Data Peparation

[12]:
```
display(df.iloc[54])
```

```
age                             47
workclass              Self-emp-inc
workid                       109832
Education                   HS-grad
Education_number                  9
martialstatus            Divorced
occupation          Exec-managerial
relationship          Not-in-family
race                          White
gender                         Male
capitalgain                       0
capitalloss                       0
hoursPerweek                     60
nativecountry         United-States
income                        <=50K
Name: 54, dtype: object
```

[13]:
```
df.tail()
```

[13]:

	age	workclass	workid	Education	Education_number	\
32556	27	Private	257302	Assoc-acdm	12	
32557	40	Private	154374	HS-grad	9	
32558	58	Private	151910	HS-grad	9	
32559	22	Private	201490	HS-grad	9	
32560	52	Self-emp-inc	287927	HS-grad	9	

	martialstatus	occupation	relationship	race	gender	\

```
32556   Married-civ-spouse        Tech-support         Wife   White   Female
32557   Married-civ-spouse   Machine-op-inspct      Husband   White     Male
32558              Widowed        Adm-clerical    Unmarried   White   Female
32559        Never-married        Adm-clerical    Own-child   White     Male
32560   Married-civ-spouse     Exec-managerial         Wife   White   Female

       capitalgain  capitalloss  hoursPerweek   nativecountry  income
32556            0            0            38   United-States   <=50K
32557            0            0            40   United-States    >50K
32558            0            0            40   United-States   <=50K
32559            0            0            20   United-States   <=50K
32560        15024            0            40   United-States    >50K
```

[14]:
```
df.head()
```

[14]:
```
   age          workclass  workid   Education  Education_number  \
0   39          State-gov   77516   Bachelors                13
1   50   Self-emp-not-inc   83311   Bachelors                13
2   38            Private  215646     HS-grad                 9
3   53            Private  234721        11th                 7
4   28            Private  338409   Bachelors                13

         martialstatus          occupation    relationship    race   gender  \
0        Never-married        Adm-clerical   Not-in-family   White     Male
1   Married-civ-spouse     Exec-managerial         Husband   White     Male
2             Divorced   Handlers-cleaners   Not-in-family   White     Male
3   Married-civ-spouse   Handlers-cleaners         Husband   Black     Male
4   Married-civ-spouse      Prof-specialty            Wife   Black   Female

   capitalgain  capitalloss  hoursPerweek   nativecountry  income
0         2174            0            40   United-States   <=50K
1            0            0            13   United-States   <=50K
2            0            0            40   United-States   <=50K
3            0            0            40   United-States   <=50K
4            0            0            40            Cuba   <=50K
```

[15]:
```
df.shape[0]
```

[15]:
```
32561
```

[16]:
```
df[df['workclass'].isna()].head()
```

[16]:
```
Empty DataFrame
Columns: [age, workclass, workid, Education, Education_number, martialstatus,
occupation, relationship, race, gender, capitalgain, capitalloss, hoursPerweek,
nativecountry, income]
Index: []
```

[17]:
```
df[df['workclass'].isnull()].shape[0]
```

[17]:
```
0
```

[18]:
```
df['workclass'] = df['workclass'].apply(lambda x : x.strip() if x.strip() != '?
 ↪' else None)
```

[19]:
```
df.columns
```

[19]:
```
Index(['age', 'workclass', 'workid', 'Education', 'Education_number',
       'martialstatus', 'occupation', 'relationship', 'race', 'gender',
       'capitalgain', 'capitalloss', 'hoursPerweek', 'nativecountry',
       'income'],
      dtype='object')
```

[20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   age               32561 non-null  int64
 1   workclass         30725 non-null  object
 2   workid            32561 non-null  int64
 3   Education         32561 non-null  object
 4   Education_number  32561 non-null  int64
 5   martialstatus     32561 non-null  object
 6   occupation        32561 non-null  object
 7   relationship      32561 non-null  object
 8   race              32561 non-null  object
 9   gender            32561 non-null  object
 10  capitalgain       32561 non-null  int64
 11  capitalloss       32561 non-null  int64
 12  hoursPerweek      32561 non-null  int64
 13  nativecountry     32561 non-null  object
 14  income            32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

[21]:
```
df[df['occupation'].str.contains("\?")].shape[0]
```

[21]:
```
1843
```

[22]:
```
df['occupation'] = df['occupation'].apply(lambda x : x.strip() if x.strip() !=␣
 ↪'?' else None)
```

[23]:
```
df[df['occupation'].isna()].shape[0]
```

[23]:
```
1843
```

[24]:
```
df['relationship'] = df['relationship'].apply(lambda x : x.strip() if x.strip()␣
↪!= '?' else None)
```

[25]:
```
df['race'] = df['race'].apply(lambda x : x.strip() if x.strip() != '?' else␣
↪None)
```

[26]:
```
df['gender'] = df['gender'].apply(lambda x : x.strip() if x.strip() != '?' else␣
↪None)
```

[27]:
```
df['nativecountry'] = df['nativecountry'].apply(lambda x : x.strip() if x.
↪strip() != '?' else None)
```

[28]:
```
df['income'] = df['income'].apply(lambda x : x.strip() if x.strip() != '?' else␣
↪None)
```

[29]:
```
df.head()
```

[29]:

	age	workclass	workid	Education	Education_number	martialstatus	occupation	relationship	race	gender	capitalgain	capitalloss	hoursPerweek	nativecountry	income
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

[30]:
```
df.nunique()
```

[30]:
```
age                 73
workclass            8
workid           21648
Education           16
Education_number    16
martialstatus        7
```

```
occupation              14
relationship             6
race                     5
gender                   2
capitalgain            119
capitalloss             92
hoursPerweek            94
nativecountry           41
income                   2
dtype: int64
```

[31]:
```
country_income = df.groupby(by = ["nativecountry", "income", "gender"]).count()
```

[32]:
```
type(country_income)
```

[32]:
```
pandas.core.frame.DataFrame
```

[33]:
```
country_income.head(10)
```

[33]:

			age	workclass	workid	Education	\
nativecountry	income	gender					
Cambodia	<=50K	Female	3	2	3	3	
		Male	9	9	9	9	
	>50K	Male	7	7	7	7	
Canada	<=50K	Female	30	25	30	30	
		Male	52	46	52	52	
	>50K	Female	9	9	9	9	
		Male	30	27	30	30	
China	<=50K	Female	16	13	16	16	
		Male	39	35	39	39	
	>50K	Female	5	5	5	5	

			Education_number	martialstatus	occupation	\
nativecountry	income	gender				
Cambodia	<=50K	Female	3	3	2	
		Male	9	9	9	
	>50K	Male	7	7	7	
Canada	<=50K	Female	30	30	25	
		Male	52	52	46	
	>50K	Female	9	9	9	
		Male	30	30	27	
China	<=50K	Female	16	16	13	
		Male	39	39	35	
	>50K	Female	5	5	5	

			relationship	race	capitalgain	capitalloss	\
nativecountry	income	gender					

```
Cambodia      <=50K  Female             3     3             3             3
                     Male               9     9             9             9
              >50K   Male               7     7             7             7
Canada        <=50K  Female            30    30            30            30
                     Male              52    52            52            52
              >50K   Female             9     9             9             9
                     Male              30    30            30            30
China         <=50K  Female            16    16            16            16
                     Male              39    39            39            39
              >50K   Female             5     5             5             5

                                 hoursPerweek
nativecountry income gender
Cambodia      <=50K  Female             3
                     Male               9
              >50K   Male               7
Canada        <=50K  Female            30
                     Male              52
              >50K   Female             9
                     Male              30
China         <=50K  Female            16
                     Male              39
              >50K   Female             5
```

[34]:
```
df.groupby(['income'])['income'].count()
```

[34]:
```
income
<=50K    24720
>50K      7841
Name: income, dtype: int64
```

[35]:
```
df.groupby(['gender', 'age'])['income'].count()
```

[35]:
```
gender  age
Female  17     186
        18     268
        19     356
        20     363
        21     329
               …
Male    84       6
        85       2
        87       1
        88       2
        90      29
Name: income, Length: 144, dtype: int64
```

[36]:
```
'''The Greatest Generation - born 1901-1924.
The Silent Generation - born 1925-1945.
The Baby Boomer Generation - born 1946-1964.
Generation X - born 1965-1979.
Millennials - born 1980-1994.
Generation Z - born 1995-2012.
Gen Alpha - born 2013 - 2025.'''

bins = [18, 30, 40, 50, 60, 70, 80]
labels = ['18-29', '30-39', '40-49', '50-59', '60-69', '70+']
df['agerange'] = pd.cut(df.age, bins, labels = labels,include_lowest = True)
```

[37]:
```
df.groupby(['income','gender', 'agerange'])['income'].count()
```

```
/var/folders/s7/_4q3szcs7410hk3m0m3kb38r0000gn/T/ipykernel_77307/3842783767.py:1
: FutureWarning:

The default of observed=False is deprecated and will be changed to True in a
future version of pandas. Pass observed=False to retain current behavior or
observed=True to adopt the future default and silence this warning.
```

[37]:
```
income  gender  agerange
<=50K   Female  18-29       3915
                30-39       2157
                40-49       1688
                50-59        961
                60-69        514
                70+          140
        Male    18-29       5580
                30-39       3983
                40-49       2640
                50-59       1620
                60-69        821
                70+          221
>50K    Female  18-29        158
                30-39        409
                40-49        366
                50-59        189
                60-69         47
                70+            8
        Male    18-29        524
                30-39       1997
                40-49       2289
                50-59       1358
                60-69        410
                70+           72
Name: income, dtype: int64
```

[38]:
```
df['Education'].head(100)
```

[38]:
```
0         Bachelors
1         Bachelors
2           HS-grad
3              11th
4         Bachelors
           ...
95     Some-college
96        Doctorate
97     Some-college
98       Assoc-acdm
99          HS-grad
Name: Education, Length: 100, dtype: object
```

[39]:
```
uniqueEdu = df['Education'].unique()
print(sorted(uniqueEdu))
```

```
[' 10th', ' 11th', ' 12th', ' 1st-4th', ' 5th-6th', ' 7th-8th', ' 9th', ' Assoc-
acdm', ' Assoc-voc', ' Bachelors', ' Doctorate', ' HS-grad', ' Masters', '
Preschool', ' Prof-school', ' Some-college']
```

[40]:
```
grp = df.groupby(['Education','Education_number'])
grp.describe()
```

[40]:

		age					\
		count	mean	std	min	25%	
Education	Education_number						
10th	6	933.0	37.429796	16.720713	17.0	22.00	
11th	7	1175.0	32.355745	15.545485	17.0	18.00	
12th	8	433.0	32.000000	14.334625	17.0	19.00	
1st-4th	2	168.0	46.142857	15.615625	19.0	33.00	
5th-6th	3	333.0	42.885886	15.557285	17.0	29.00	
7th-8th	4	646.0	48.445820	16.092350	17.0	34.25	
9th	5	514.0	41.060311	15.946862	17.0	28.00	
Assoc-acdm	12	1067.0	37.381443	11.095177	19.0	29.00	
Assoc-voc	11	1382.0	38.553546	11.631300	19.0	30.00	
Bachelors	13	5355.0	38.904949	11.912210	19.0	29.00	
Doctorate	16	413.0	47.702179	11.784716	24.0	39.00	
HS-grad	9	10501.0	38.974479	13.541524	17.0	28.00	
Masters	14	1723.0	44.049913	11.068935	18.0	36.00	
Preschool	1	51.0	42.764706	15.126914	19.0	31.00	
Prof-school	15	576.0	44.746528	11.962477	25.0	36.00	
Some-college	10	7291.0	35.756275	13.474051	17.0	24.00	

					workid		…	\
		50%	75%	max	count	mean	…	
Education	Education_number						…	

10th	6	34.0	52.0	90.0	933.0	196832.465166	…
11th	7	28.0	43.0	90.0	1175.0	194928.077447	…
12th	8	28.0	41.0	79.0	433.0	199097.508083	…
1st-4th	2	46.0	57.0	90.0	168.0	239303.000000	…
5th-6th	3	42.0	54.0	84.0	333.0	232448.333333	…
7th-8th	4	50.0	61.0	90.0	646.0	188079.171827	…
9th	5	39.0	54.0	90.0	514.0	202485.066148	…
Assoc-acdm	12	36.0	44.0	90.0	1067.0	193424.093721	…
Assoc-voc	11	37.0	46.0	84.0	1382.0	181936.016643	…
Bachelors	13	37.0	46.0	90.0	5355.0	188055.914846	…
Doctorate	16	47.0	55.0	80.0	413.0	186698.760291	…
HS-grad	9	37.0	48.0	90.0	10501.0	189538.739739	…
Masters	14	43.0	51.0	90.0	1723.0	179852.362739	…
Preschool	1	41.0	53.5	75.0	51.0	235889.372549	…
Prof-school	15	43.0	51.0	90.0	576.0	185663.706597	…
Some-college	10	34.0	45.0	90.0	7291.0	188742.922370	…

		capitalloss			hoursPerweek		\
		75%	max		count	mean	
Education	Education_number						
10th	6	0.0	3770.0		933.0	37.052519	
11th	7	0.0	2824.0		1175.0	33.925957	
12th	8	0.0	2258.0		433.0	35.780600	
1st-4th	2	0.0	2603.0		168.0	38.255952	
5th-6th	3	0.0	2603.0		333.0	38.897898	
7th-8th	4	0.0	3900.0		646.0	39.366873	
9th	5	0.0	2231.0		514.0	38.044747	
Assoc-acdm	12	0.0	2824.0		1067.0	40.504217	
Assoc-voc	11	0.0	2603.0		1382.0	41.610709	
Bachelors	13	0.0	2824.0		5355.0	42.614006	
Doctorate	16	0.0	3683.0		413.0	46.973366	
HS-grad	9	0.0	4356.0		10501.0	40.575374	
Masters	14	0.0	2824.0		1723.0	43.836332	
Preschool	1	0.0	1719.0		51.0	36.647059	
Prof-school	15	0.0	2824.0		576.0	47.425347	
Some-college	10	0.0	4356.0		7291.0	38.852284	

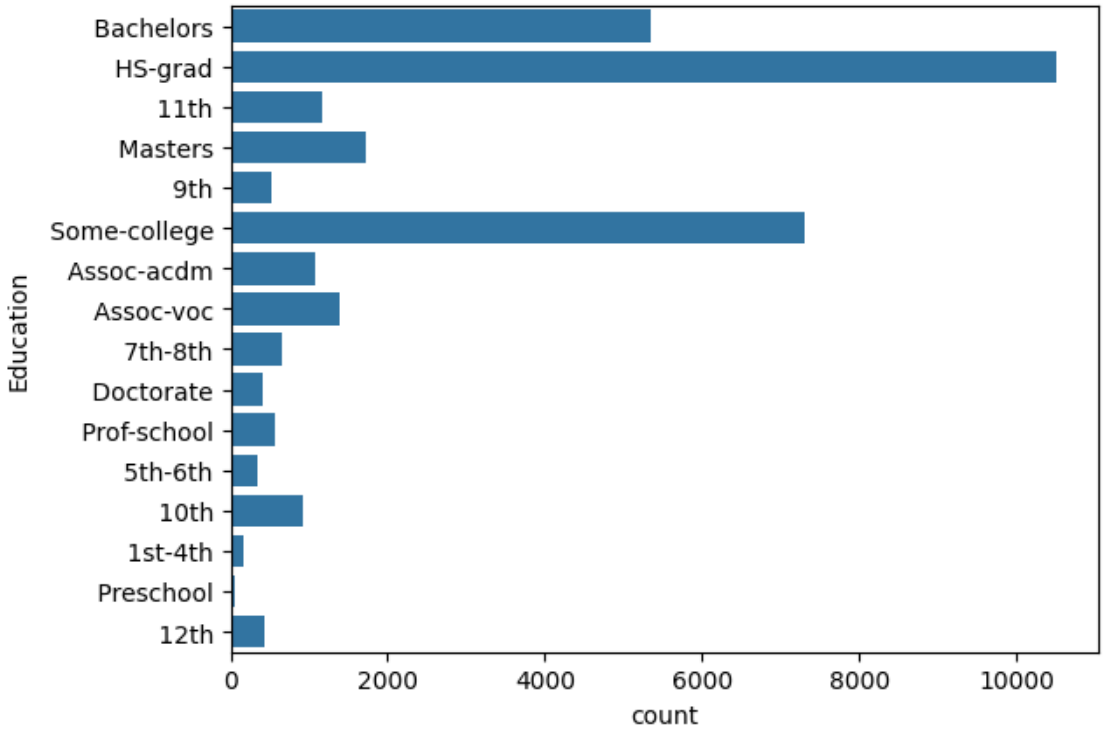
		std	min	25%	50%	75%	max
Education	Education_number						
10th	6	13.788112	1.0	30.0	40.0	40.0	99.0
11th	7	13.965416	2.0	20.0	40.0	40.0	99.0
12th	8	12.626412	6.0	30.0	40.0	40.0	99.0
1st-4th	2	12.848727	4.0	35.0	40.0	40.0	96.0
5th-6th	3	10.551727	3.0	40.0	40.0	40.0	84.0
7th-8th	4	14.201870	2.0	35.0	40.0	40.0	99.0
9th	5	11.064402	1.0	36.0	40.0	40.0	99.0

```
Assoc-acdm   12                12.196666   1.0  40.0  40.0  45.0  99.0
Assoc-voc    11                10.793384   1.0  40.0  40.0  45.0  99.0
Bachelors    13                11.446185   2.0  40.0  40.0  50.0  99.0
Doctorate    16                15.084447   1.0  40.0  45.0  55.0  99.0
HS-grad      9                 11.333757   1.0  40.0  40.0  42.0  99.0
Masters      14                12.277801   1.0  40.0  40.0  50.0  99.0
Preschool    1                 12.555196  10.0  30.0  40.0  40.0  75.0
Prof-school  15                14.806038   2.0  40.0  48.0  55.0  99.0
Some-college 10                12.761901   1.0  35.0  40.0  43.0  99.0

[16 rows x 40 columns]
```

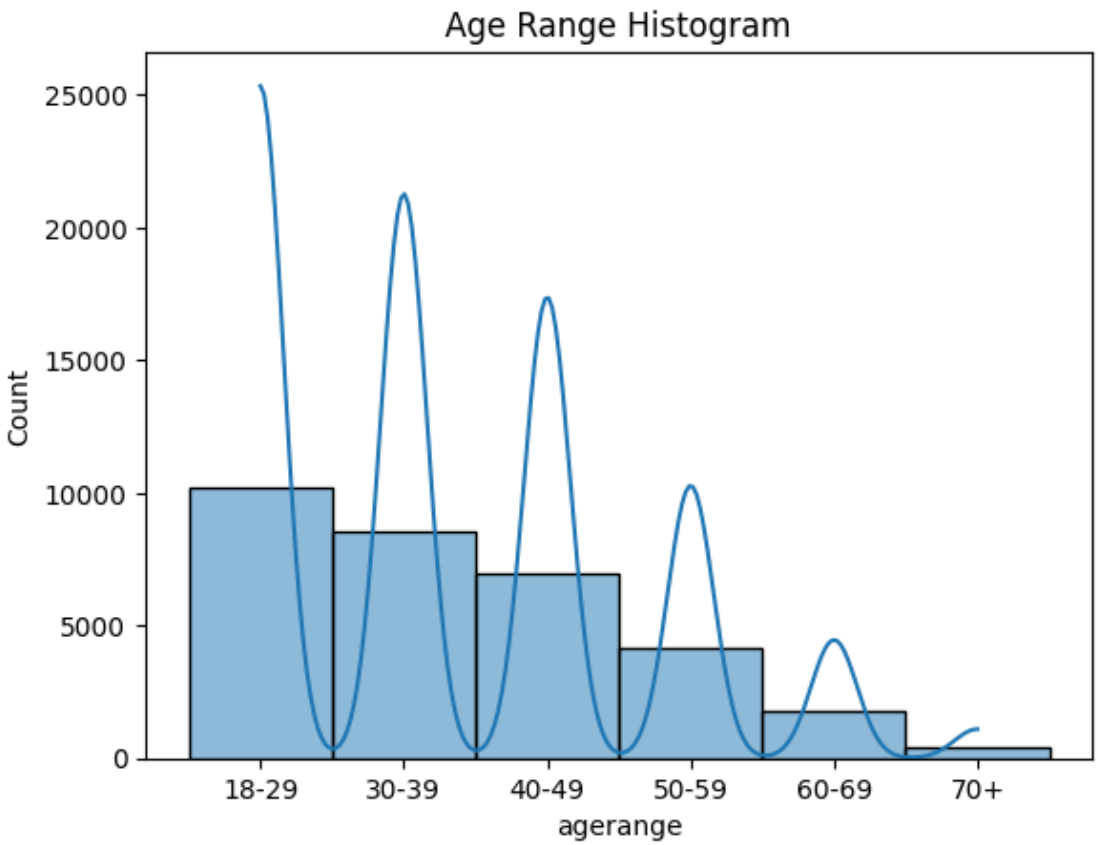
[41]:
```
# plot count plot for the education column
sns.countplot(df.Education)
```

[41]: <Axes: xlabel='count', ylabel='Education'>



[42]:
```
sns.histplot(df.agerange,kde=True).set(title='Age Range Histogram')
```

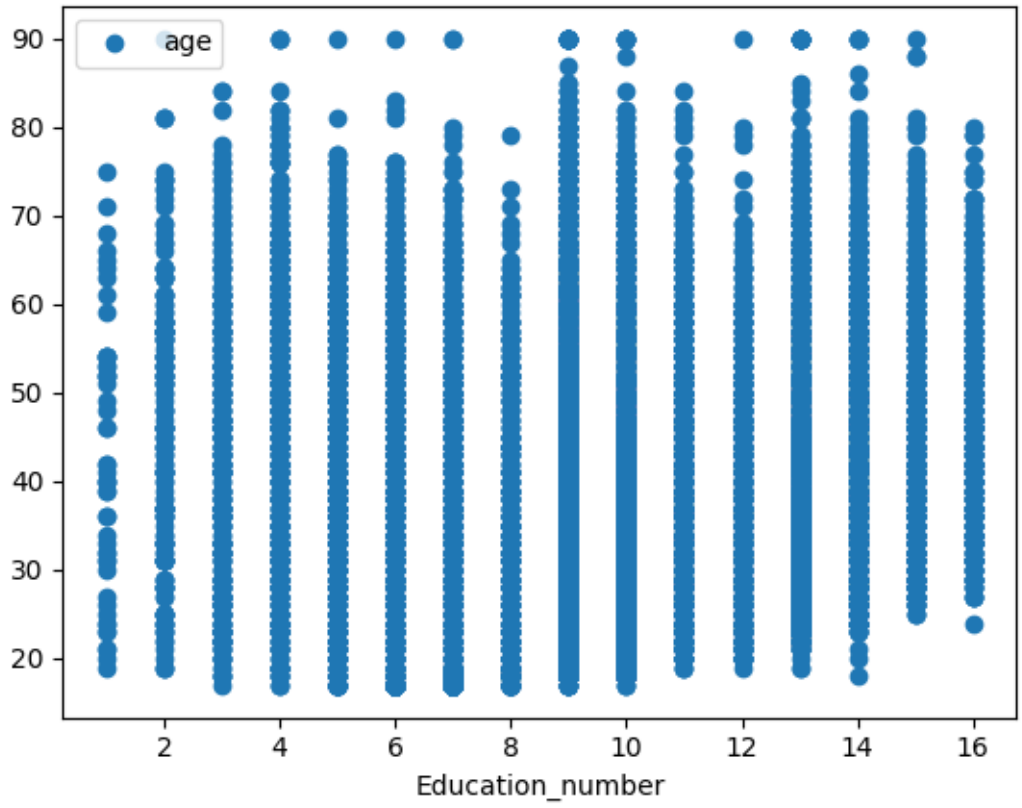
[42]: [Text(0.5, 1.0, 'Age Range Histogram')]



[43]:
```
#sns.scatterplot('income', 'agerange', data=g1);
```

[44]:
```
df.plot(x='Education_number', y='age', style='o')
```

[44]: <Axes: xlabel='Education_number'>



[45]:
```
df.head()
```

[45]:

	age	workclass	workid	Education	Education_number	\
0	39	State-gov	77516	Bachelors	13	
1	50	Self-emp-not-inc	83311	Bachelors	13	
2	38	Private	215646	HS-grad	9	
3	53	Private	234721	11th	7	
4	28	Private	338409	Bachelors	13	

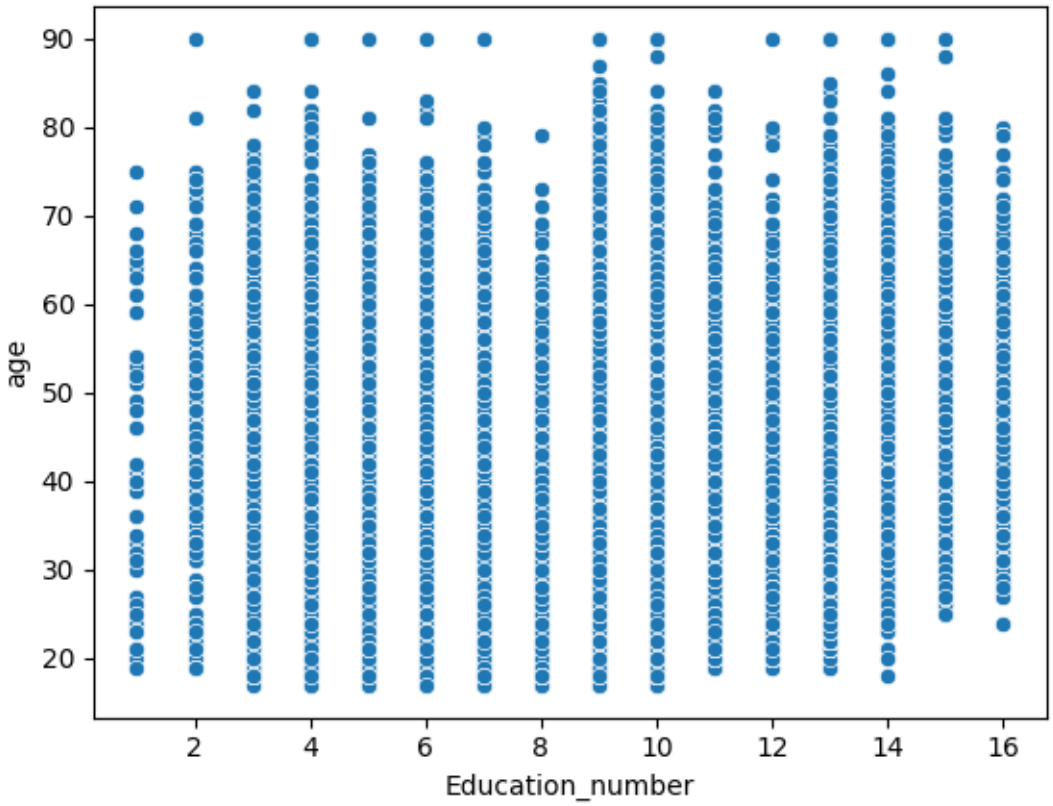
	martialstatus	occupation	relationship	race	gender	\
0	Never-married	Adm-clerical	Not-in-family	White	Male	
1	Married-civ-spouse	Exec-managerial	Husband	White	Male	
2	Divorced	Handlers-cleaners	Not-in-family	White	Male	
3	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	
4	Married-civ-spouse	Prof-specialty	Wife	Black	Female	

	capitalgain	capitalloss	hoursPerweek	nativecountry	income	agerange
0	2174	0	40	United-States	<=50K	30-39
1	0	0	13	United-States	<=50K	40-49
2	0	0	40	United-States	<=50K	30-39
3	0	0	40	United-States	<=50K	50-59

```
4              0              0              40             Cuba   <=50K     18-29
```

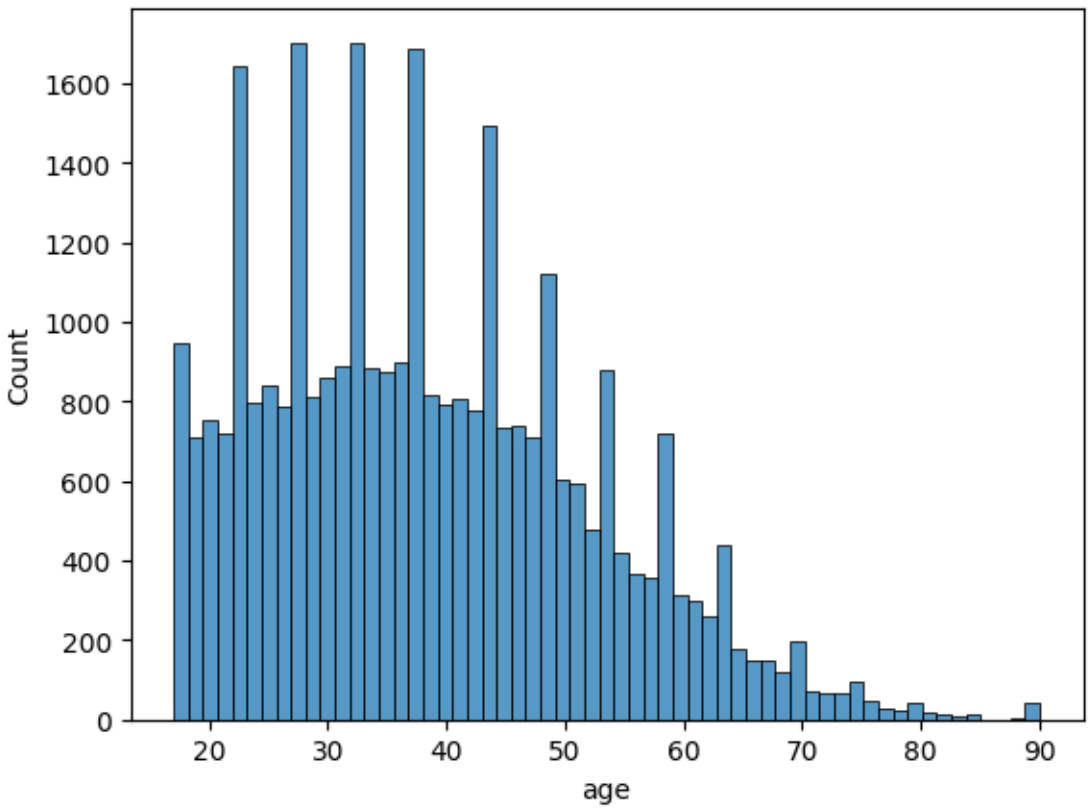
[46]:
```
sns.scatterplot(data=df, x="Education_number", y="age")
```

[46]:
```
<Axes: xlabel='Education_number', ylabel='age'>
```



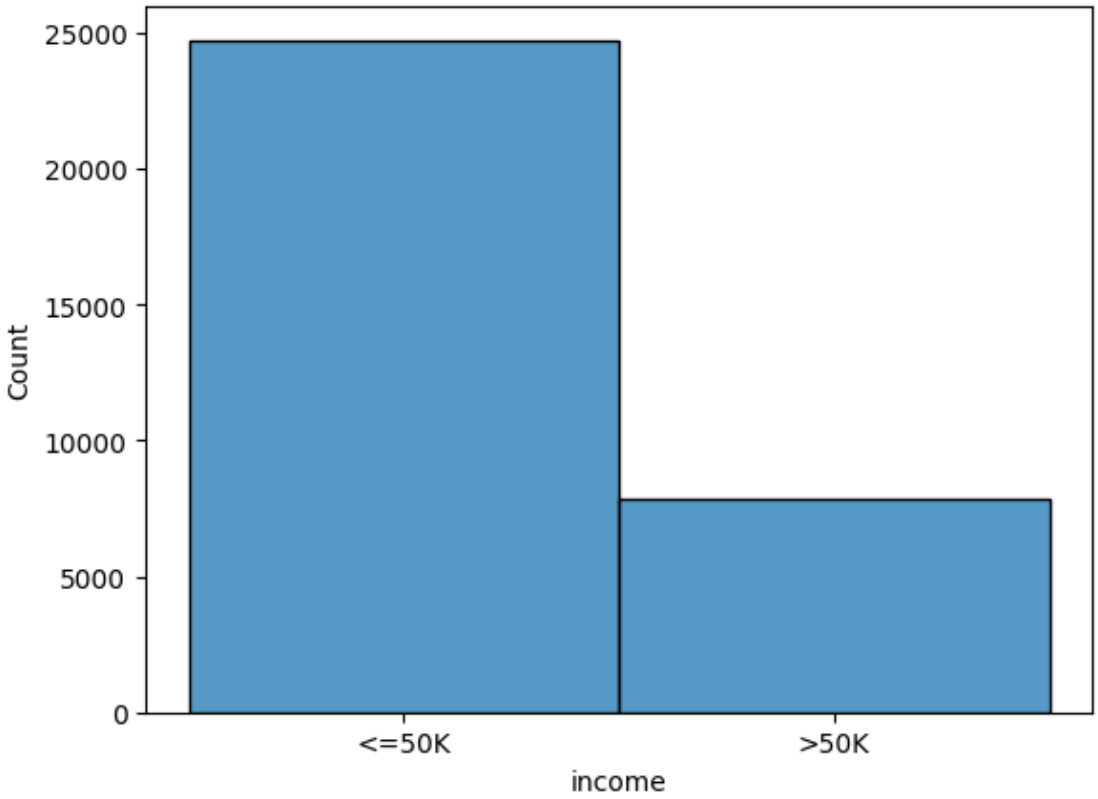
[47]:
```
# Distribution of age histogram
sns.histplot(data=df, x="age")
```

[47]:
```
<Axes: xlabel='age', ylabel='Count'>
```



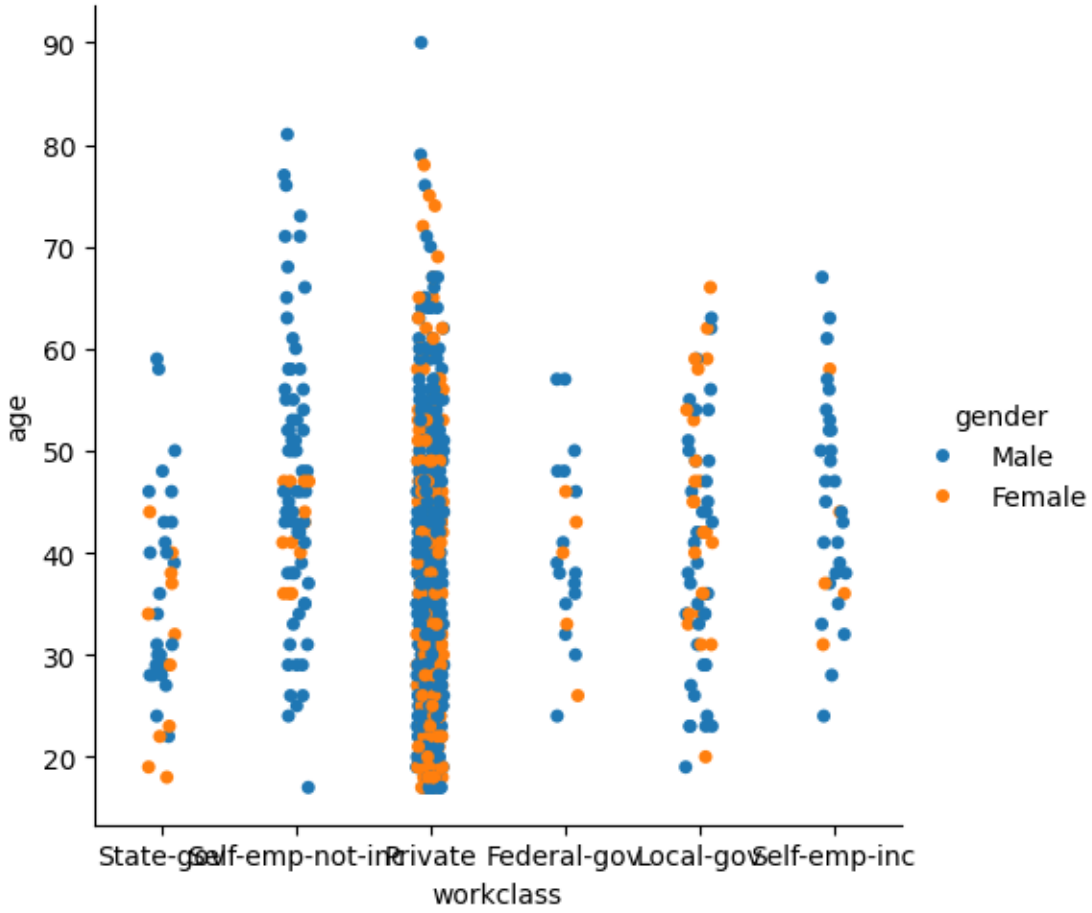
[48]:
```
sns.histplot(data=df, x="income")
```

[48]:
```
<Axes: xlabel='income', ylabel='Count'>
```



[49]:
```
sns.catplot(data=df.head(1000), x="workclass", y="age", hue = "gender")
```

[49]: <seaborn.axisgrid.FacetGrid at 0x16c5178e0>



[50]:
```
df.head()
```

[50]:

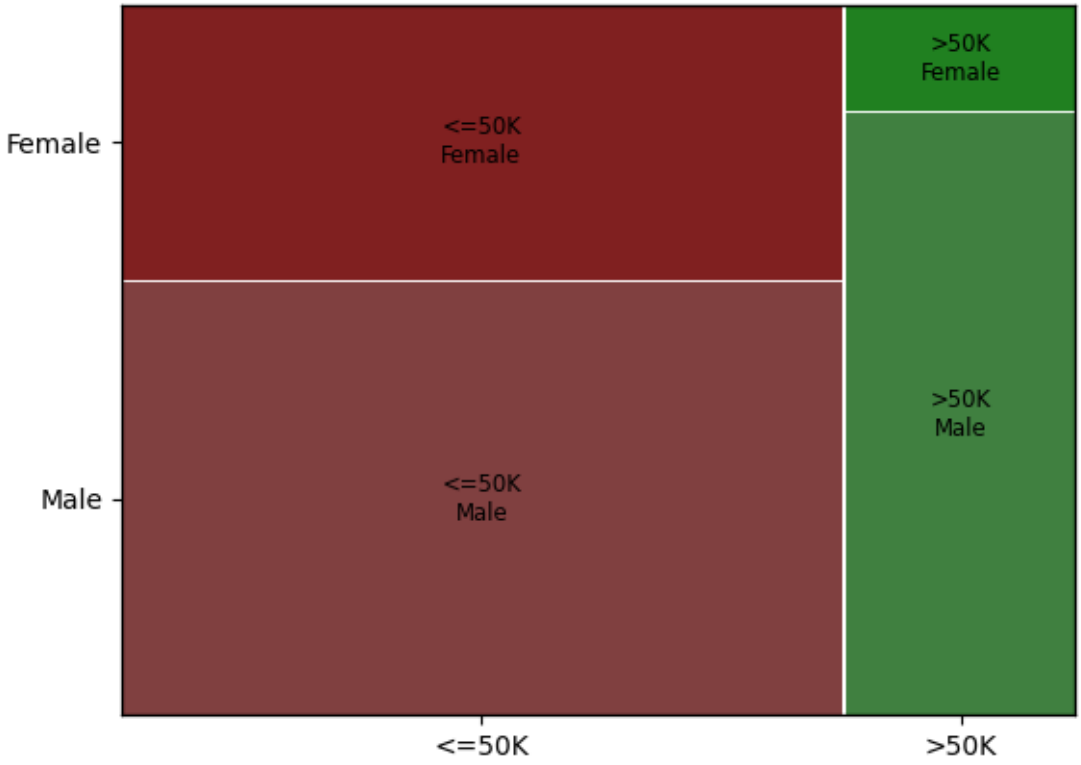
	age	workclass	workid	Education	Education_number	\
0	39	State-gov	77516	Bachelors	13	
1	50	Self-emp-not-inc	83311	Bachelors	13	
2	38	Private	215646	HS-grad	9	
3	53	Private	234721	11th	7	
4	28	Private	338409	Bachelors	13	

	martialstatus	occupation	relationship	race	gender	\
0	Never-married	Adm-clerical	Not-in-family	White	Male	
1	Married-civ-spouse	Exec-managerial	Husband	White	Male	
2	Divorced	Handlers-cleaners	Not-in-family	White	Male	
3	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	
4	Married-civ-spouse	Prof-specialty	Wife	Black	Female	

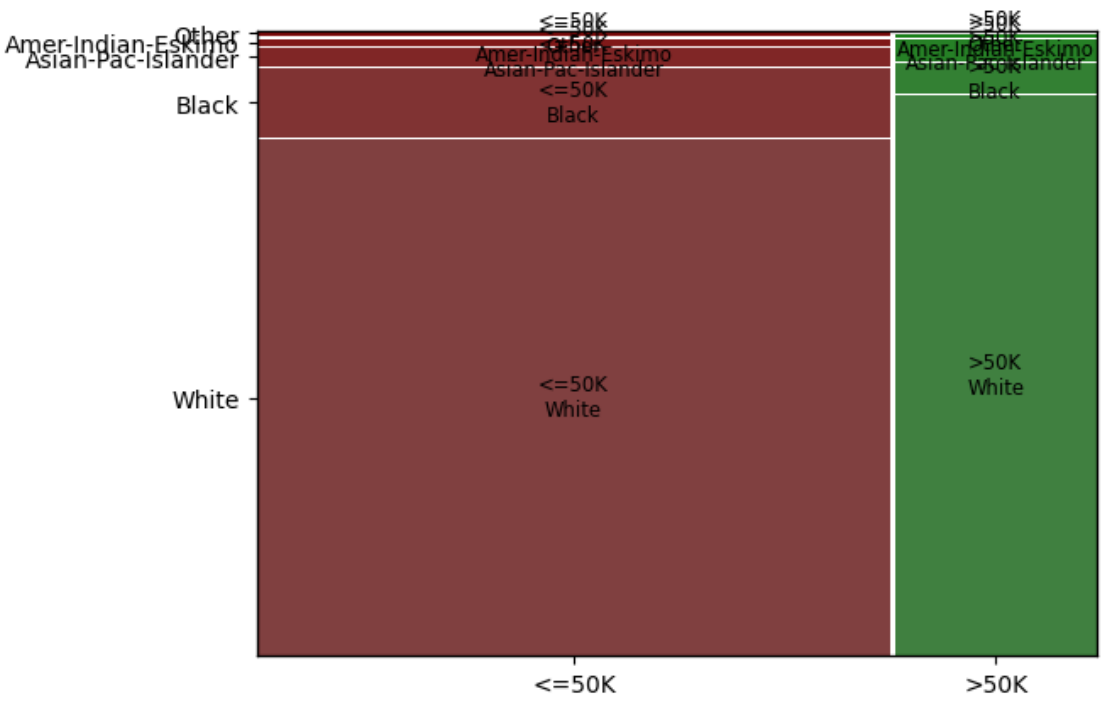
	capitalgain	capitalloss	hoursPerweek	nativecountry	income	agerange
0	2174	0	40	United-States	<=50K	30-39

1	0	0	13	United-States	<=50K	40-49
2	0	0	40	United-States	<=50K	30-39
3	0	0	40	United-States	<=50K	50-59
4	0	0	40	Cuba	<=50K	18-29

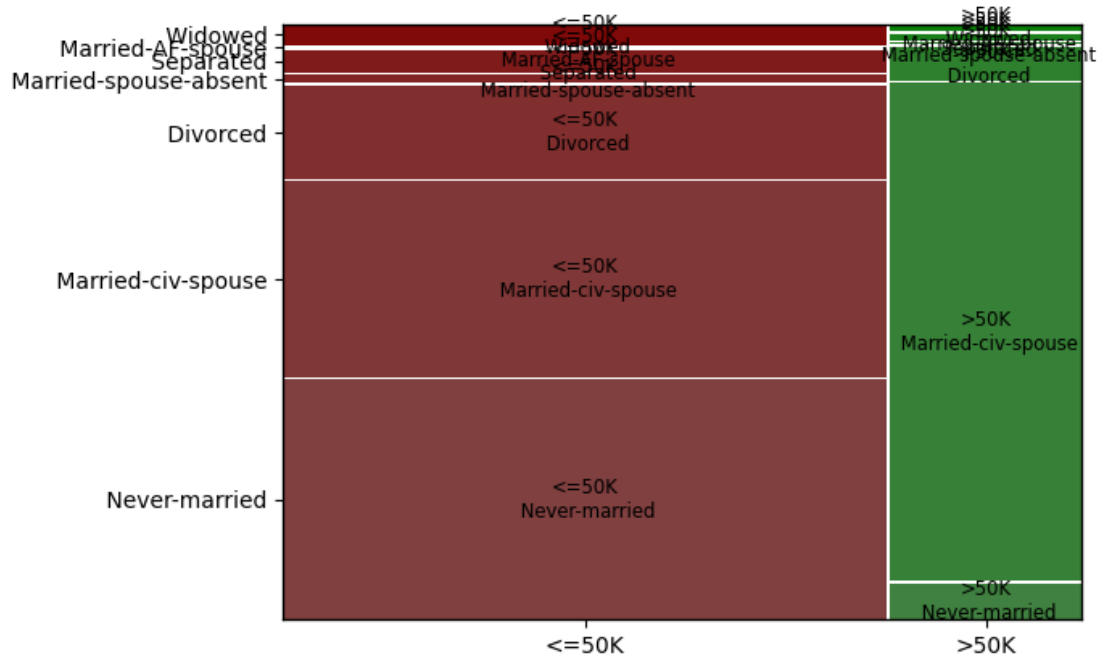
[51]:
```
mosaic(df, ['income', 'gender'])
plt.show()
```



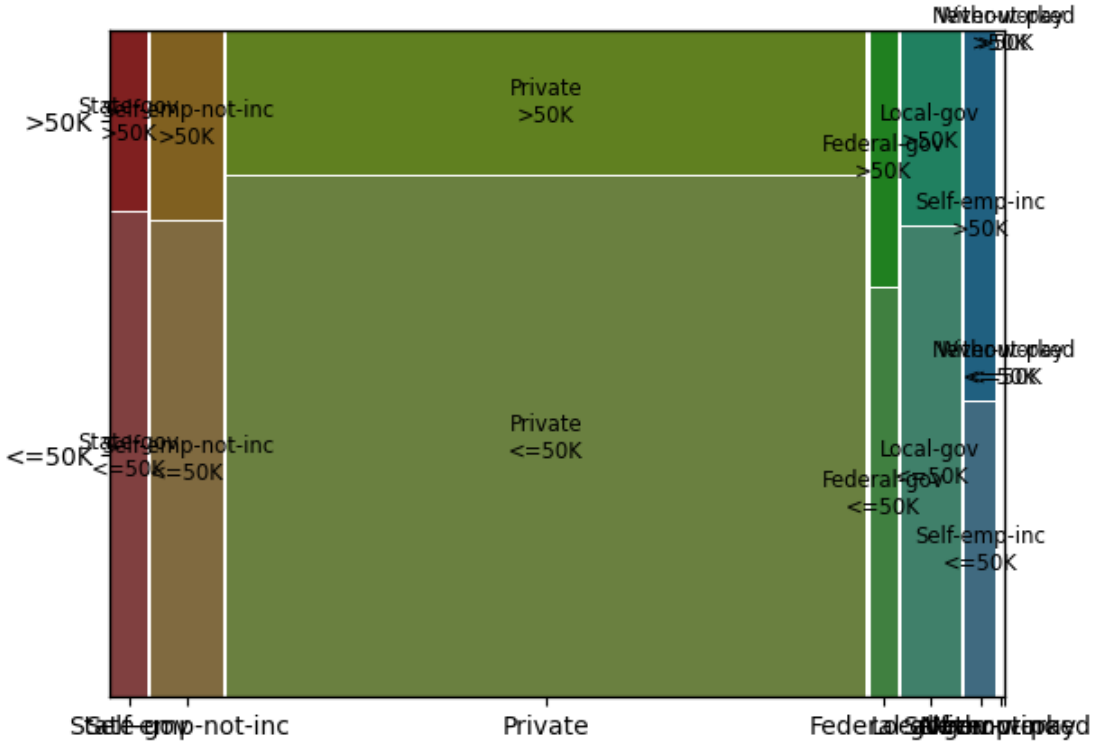
[52]:
```
mosaic(df, ['income', 'race'])
plt.show()
```



[53]:
```
mosaic(df, ['income', 'martialstatus'])
plt.show()
```



[54]:
```
mosaic(df, ['workclass', 'income'])
#plt.rcParams["figure.figsize"] = [7.00, 3.50]
plt.rcParams["figure.figsize"]=(20,20)
plt.rcParams['font.size'] = (9.0)
#plt.rcParams["figure.autolayout"] = True
plt.show()
```



[55]:
```
df.capitalgain.max()
```

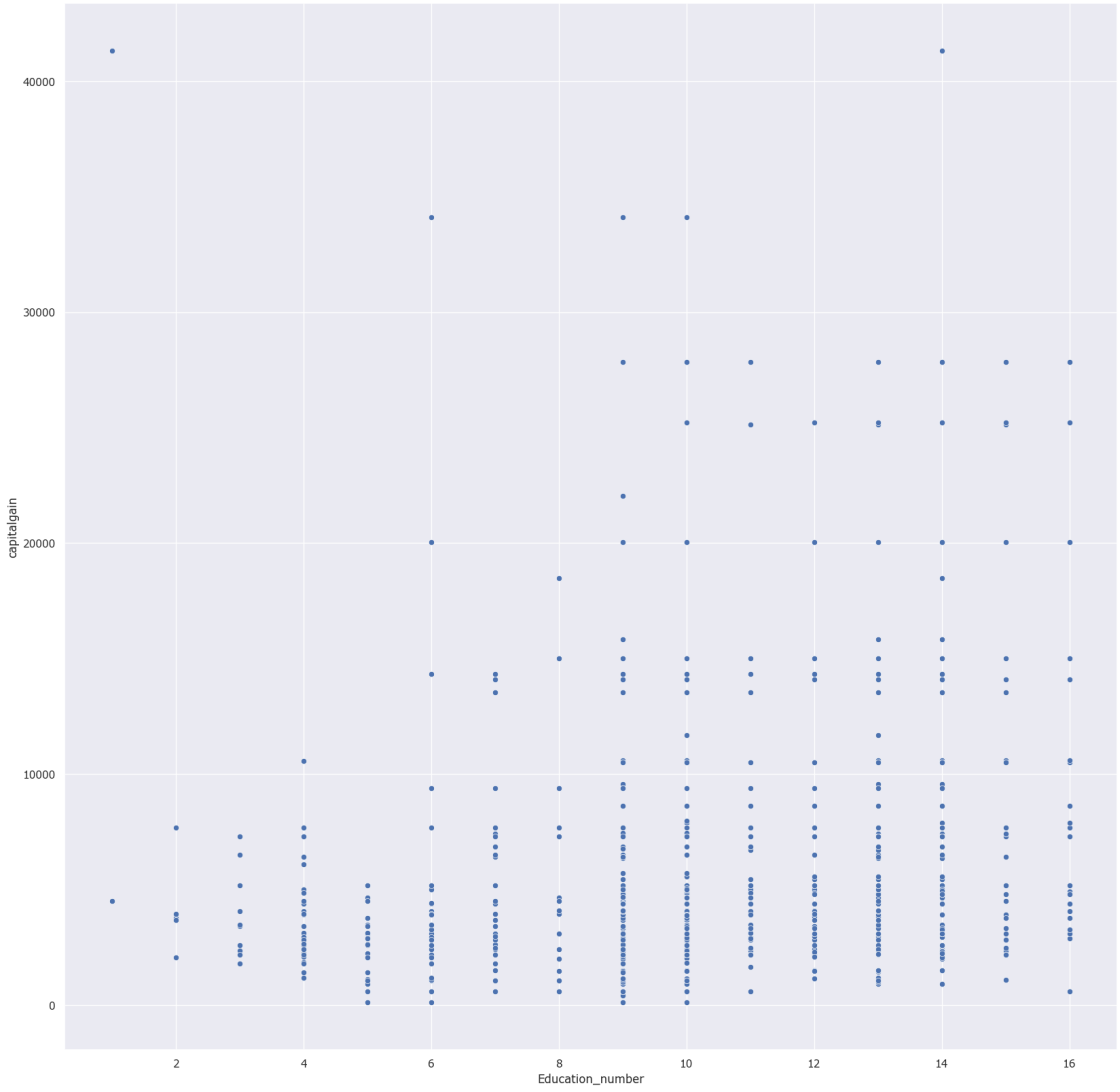
[55]:
```
99999
```

[56]:
```
df.capitalgain.mean()
```

[56]:
```
1077.6488437087312
```

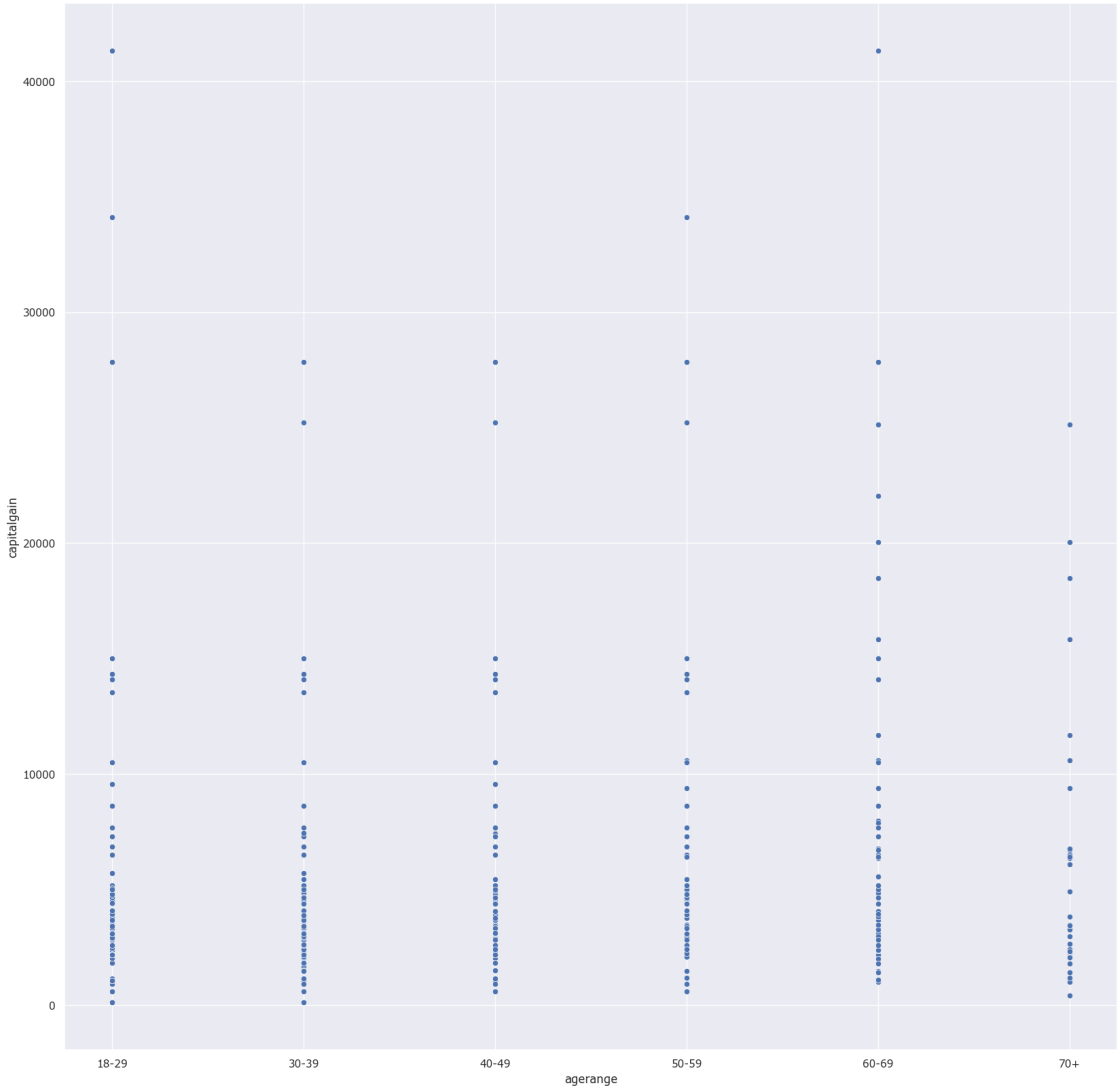
[57]:
```
sns.set(font="Verdana")
sns.scatterplot(data=df.query('capitalgain > 0 and capitalgain < 99999'),␣
↪x="Education_number", y="capitalgain")
```

[57]:
```
<Axes: xlabel='Education_number', ylabel='capitalgain'>
```



[58]:
```
sns.scatterplot(data=df.query('capitalgain > 0 and capitalgain < 99999'),␣
↪x="agerange", y="capitalgain")
```

[58]: <Axes: xlabel='agerange', ylabel='capitalgain'>



[ ]:
```
df.head()
df1 = df[['Education_number','capitalgain','age', 'income']]
df_capgain = df1.query('capitalgain > 0 and capitalgain < 99999')
#df_capgain["income"] = np.where(df_capgain["income"] == "<=50K", 0, 1)
df_capgain['income'].mask(df_capgain['income'] == '<=50K', 0, inplace=True)
df_capgain['income'].mask(df_capgain['income'] == '>50K', 1, inplace=True)

normalized_df=(df_capgain-df_capgain.min())/(df_capgain.max()-df_capgain.min())

df_pcp = df[['Education_number','age', 'income']]
```

[ ]:
```
normalized_df.head()
```

[ ]:
```
plt.figure(figsize = (12, 8))
```

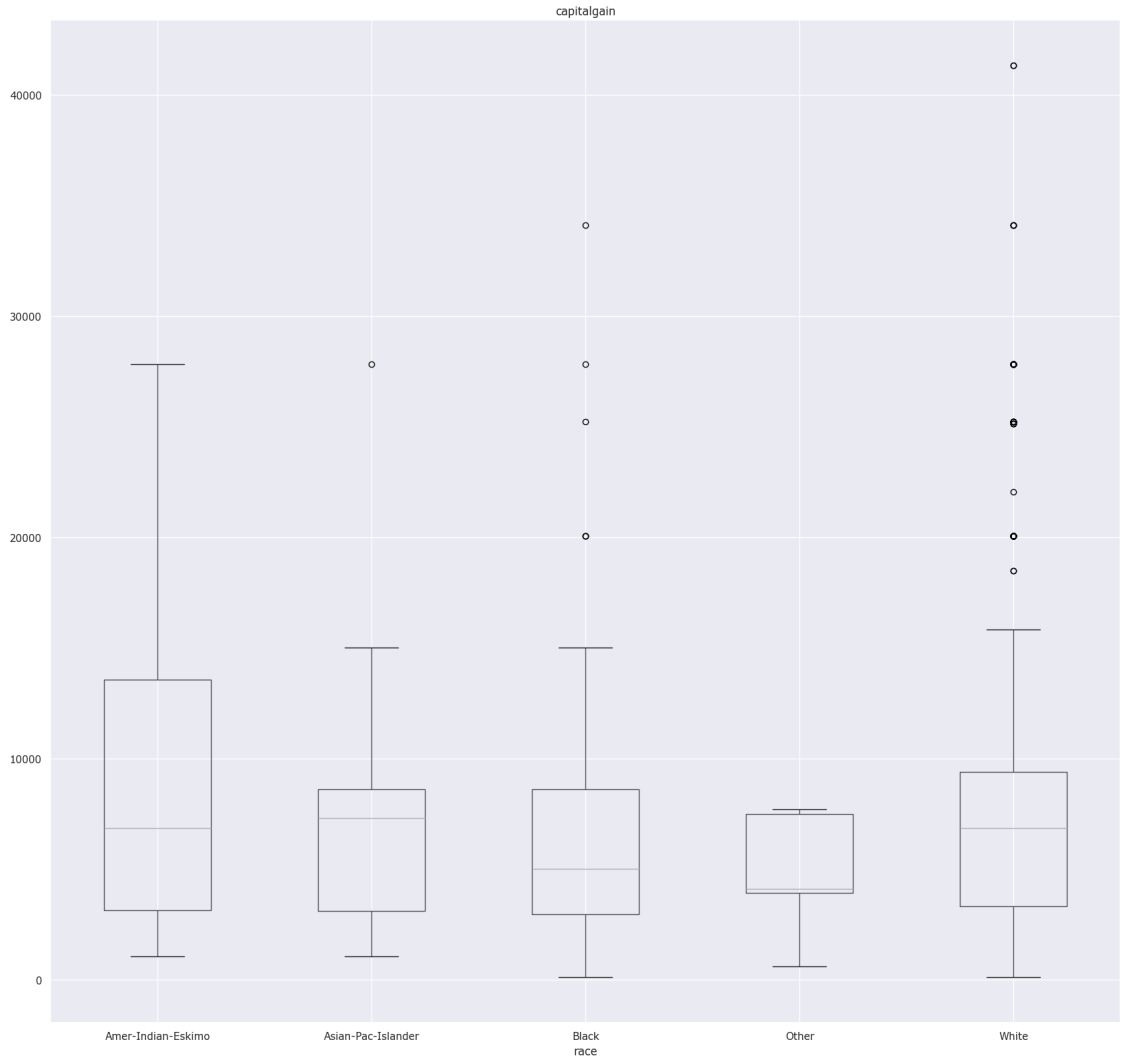
[ ]:
```
parallel_coordinates(df_pcp, "income", color = ['blue', 'green'])
```

[ ]:
```
df2 = df[['gender','Education', 'agerange', 'income', 'Education_number']]
#fig = px.parallel_categories(df2)
fig = px.parallel_categories(df2, dimensions=['gender', 'agerange', 'income'],
                color="Education_number", color_continuous_scale=px.colors.
 ↪sequential.Inferno,
                labels={'gender':'Gender', 'agerange':'Age Range', 'income':
 ↪'Income level'})
fig.show()
```

[60]:
```
df_box = df.query('capitalgain > 0 and capitalgain < 99999')

df_box.boxplot(column='capitalgain', by='race')
```

[60]:
```
<Axes: title={'center': 'capitalgain'}, xlabel='race'>
```


Boxplot grouped by race
capitalgain
40000
30000
20000
10000
0
Amer-Indian-Eskimo
Asian-Pac-Islander
Black
Other
White
race

[ ]: