Chapter 1

April 19, 2025

1 Hands-On Data Preprocessing in Python

Learn how to effectively prepare data for successful data analytics

`AUTHOR: Dr. Roy Jafari`

1.0.1 Chapter 1: Review of the core modules NumPy, Pandas, and Matplotlib

Hello World! This is a Markdown chunk!

[2]:
```python
print('Hello World! This is Code Chunk!')
```

```
Hello World! This is Code Chunk!
```

[4]:
```python
import numpy as np
```

[5]:
```python
lst_nums = [2,5,7,11,13,17,23,31,37,41,43,47]
np.mean(lst_nums)
```

[5]:
```
np.float64(23.083333333333332)
```

[6]:
```python
lst_nums = [2,5,7,11,13,17,23,31,37,41,43,47]
ary_nums = np.array(lst_nums)
ary_nums.mean()
```

[6]:
```
np.float64(23.083333333333332)
```

[7]:
```python
type(lst_nums)
```

[7]:
```
list
```

[8]:
```python
type(ary_nums)
```

[8]:
```
numpy.ndarray
```

[9]:
```python
np.arange(15)
```

[9]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

[10]:
```python
np.arange(5,15)
```

[10]: array([5, 6, 7, 8, 9, 10, 11, 12, 13, 14])

[11]:
```
np.arange(-7.1,7)
```

[11]:
```
array([-7.1, -6.1, -5.1, -4.1, -3.1, -2.1, -1.1, -0.1,  0.9,  1.9,  2.9,
        3.9,  4.9,  5.9,  6.9])
```

[12]:
```
np.zeros([4,5])
```

[12]:
```
array([[0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.]])
```

[13]:
```
np.ones(7)
```

[13]: array([1., 1., 1., 1., 1., 1., 1.])

Example #1

The following is the grade data of ten students. Create a code using NumPy that calcuate and report their grade average.

```
Names = ['Jevon', 'Dawn', 'Kayleigh', 'Jadene', 'Kennedy', 'Kaydee', 'Ansh', 'Flynn', 'Kier',
Math_grades = [80, 50, 60, 70, 60, 100, 70, 70, 60, 70]
Science_grades = [90, 80, 50, 50, 60, 50, 90, 70, 80, 80]
History_grades = [60, 90, 50, 90, 100, 100, 100, 100, 90, 70]
```

[14]:
```
Names = ['Jevon', 'Dawn', 'Kayleigh', 'Jadene', 'Kennedy', 'Kaydee',
         'Ansh', 'Flynn', 'Kier', 'Clarence']
Math_grades = [80, 50, 60, 70, 60, 100, 70, 70, 60, 70]
Science_grades = [90, 80, 50, 50, 60, 50, 90, 70, 80, 80]
History_grades = [60, 90, 50, 90, 100, 100, 100, 100, 90, 70]
```

[15]:
```
Average_grades = np.zeros(10)
print(Average_grades)

for i, name in enumerate(Names):
    Average_grades[i] = np.mean([Math_grades[i],Science_grades[i],
                                 History_grades[i]])

print(Average_grades)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[76.66666667 73.33333333 53.33333333 70.         73.33333333 83.33333333
 86.66666667 80.         76.66666667 73.33333333]
```

[16]:
```
# better-looking report

for i, name in enumerate(Names):
```

```
    print("Average for {} : {}".format(name,Average_grades[i]))
```

```
Average for Jevon : 76.66666666666667
Average for Dawn : 73.33333333333333
Average for Kayleigh : 53.333333333333336
Average for Jadene : 70.0
Average for Kennedy : 73.33333333333333
Average for Kaydee : 83.33333333333333
Average for Ansh : 86.66666666666667
Average for Flynn : 80.0
Average for Kier : 76.66666666666667
Average for Clarence : 73.33333333333333
```

[17]:
```
np.linspace(0,1,21)
```

[17]:
```
array([0.  , 0.05, 0.1 , 0.15, 0.2 , 0.25, 0.3 , 0.35, 0.4 , 0.45, 0.5 ,
       0.55, 0.6 , 0.65, 0.7 , 0.75, 0.8 , 0.85, 0.9 , 0.95, 1.  ])
```

[18]:
```
np.linspace(10,1000,100)
```

[18]:
```
array([  10.,   20.,   30.,   40.,   50.,   60.,   70.,   80.,   90.,
        100.,  110.,  120.,  130.,  140.,  150.,  160.,  170.,  180.,
        190.,  200.,  210.,  220.,  230.,  240.,  250.,  260.,  270.,
        280.,  290.,  300.,  310.,  320.,  330.,  340.,  350.,  360.,
        370.,  380.,  390.,  400.,  410.,  420.,  430.,  440.,  450.,
        460.,  470.,  480.,  490.,  500.,  510.,  520.,  530.,  540.,
        550.,  560.,  570.,  580.,  590.,  600.,  610.,  620.,  630.,
        640.,  650.,  660.,  670.,  680.,  690.,  700.,  710.,  720.,
        730.,  740.,  750.,  760.,  770.,  780.,  790.,  800.,  810.,
        820.,  830.,  840.,  850.,  860.,  870.,  880.,  890.,  900.,
        910.,  920.,  930.,  940.,  950.,  960.,  970.,  980.,  990.,
       1000.])
```

Example 2

We are interested in finding the value(s) that holds the following mathematical statement.

`x^2-5x+6=0`

And imagine that we don't know that the statement can be simplified easily to find either -1 or +1 will hold the statement.

`x^2-5x+6=(x-2)(x-3)`

so we would like to use NumPy to try out any values between -100 and 100 and see what the answer is.

[20]:
```
Candidates = np.linspace(-1000,1000,2001)
print(Candidates)

for candidate in Candidates:
```

```
    if(candidate**2 - 5*candidate +6 ==0):
        print("Just found a possible answer: {}".format(candidate))
```

```
[-1000.  -999.  -998. …   998.   999.  1000.]
Just found a possible answer: 2.0
Just found a possible answer: 3.0
```

2 Adult Dataset

"Census Income" dataset.

Number of Instances: 48842 Number of Attributes: 14 Date Donated: 1996-05-01 Missing Values?: Yes

2.0.1 Attributes:

Number of Attributes: 6 continuous, 8 nominal attributes

- age: continuous.
- workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- fnlwgt: continuous.
- education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- education-num: continuous.
- marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- sex: Female, Male.
- capital-gain: continuous.
- capital-loss: continuous.
- hours-per-week: continuous.
- native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
- class: >50K, <=50K

[22]:
```
import pandas as pd

adult_df = pd.read_csv('adult.csv')
adult_df.head()
```

[22]:

```
   age         workclass  fnlwgt  education  education-num  \
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

       marital-status         occupation   relationship   race     sex  \
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
```

[23]:

```python
type(adult_df.age)
```

[23]:

```
pandas.core.series.Series
```

[24]:

```python
type(adult_df)
```

[24]:

```
pandas.core.frame.DataFrame
```

[27]:

```python
adult_df.loc[0]
```

[27]:

```
age                           39
workclass              State-gov
fnlwgt                     77516
education              Bachelors
education-num                 13
marital-status     Never-married
occupation          Adm-clerical
relationship       Not-in-family
race                       White
sex                         Male
capitalGain                 2174
capitalLoss                    0
hoursPerWeek                  40
nativeCountry      United-States
income                     <=50K
Name: 0, dtype: object
```

[25]:
```python
adult_df.loc[0].index
```

[25]:
```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education-num',
       'marital-status', 'occupation', 'relationship', 'race', 'sex',
       'capitalGain', 'capitalLoss', 'hoursPerWeek', 'nativeCountry',
       'income'],
      dtype='object')
```

[26]:
```python
adult_df.age.index
```

[26]:
```
RangeIndex(start=0, stop=32561, step=1)
```

[28]:
```python
adult_df.set_index(np.arange(10000,42561),inplace=True)
```

[29]:
```python
adult_df.set_index(np.arange(10000,42561))
```

[29]:
```
       age         workclass  fnlwgt   education  education-num  \
10000   39         State-gov   77516   Bachelors             13
10001   50  Self-emp-not-inc   83311   Bachelors             13
10002   38           Private  215646     HS-grad              9
10003   53           Private  234721        11th              7
10004   28           Private  338409   Bachelors             13
…     …               …       …          …            …
42556   27           Private  257302  Assoc-acdm             12
42557   40           Private  154374     HS-grad              9
42558   58           Private  151910     HS-grad              9
42559   22           Private  201490     HS-grad              9
42560   52      Self-emp-inc  287927     HS-grad              9

           marital-status         occupation   relationship   race     sex  \
10000       Never-married       Adm-clerical  Not-in-family  White    Male
10001  Married-civ-spouse    Exec-managerial        Husband  White    Male
10002            Divorced  Handlers-cleaners  Not-in-family  White    Male
10003  Married-civ-spouse  Handlers-cleaners        Husband  Black    Male
10004  Married-civ-spouse     Prof-specialty           Wife  Black  Female
…                     …                  …            …    …     …
42556  Married-civ-spouse       Tech-support           Wife  White  Female
42557  Married-civ-spouse  Machine-op-inspct        Husband  White    Male
42558             Widowed       Adm-clerical      Unmarried  White  Female
42559       Never-married       Adm-clerical      Own-child  White    Male
42560  Married-civ-spouse    Exec-managerial           Wife  White  Female

       capitalGain  capitalLoss  hoursPerWeek  nativeCountry income
10000         2174            0            40  United-States  <=50K
10001            0            0            13  United-States  <=50K
10002            0            0            40  United-States  <=50K
10003            0            0            40  United-States  <=50K
```

```
10004              0              0             40            Cuba  <=50K
…             …           …            …            …    …
42556              0              0             38   United-States  <=50K
42557              0              0             40   United-States   >50K
42558              0              0             40   United-States  <=50K
42559              0              0             20   United-States  <=50K
42560          15024              0             40   United-States   >50K

[32561 rows x 15 columns]
```

[32]:
```python
# adult_df.education-num # Error
adult_df["education-num"]
```

[32]:
```
10000    13
10001    13
10002     9
10003     7
10004    13
         ..
42556    12
42557     9
42558     9
42559     9
42560     9
Name: education-num, Length: 32561, dtype: int64
```

[33]:
```python
adult_df.iloc[2].loc['education']
```

[33]: 'HS-grad'

[34]:
```python
adult_df.education.loc[10002]
```

[34]: 'HS-grad'

[35]:
```python
adult_df['education'].iloc[2]
```

[35]: 'HS-grad'

[36]:
```python
adult_df.at[10002,'education']
```

[36]: 'HS-grad'

[37]:
```python
row_series = adult_df.loc[10002]
print(row_series.loc['education'])
print(row_series.iloc[3])
print(row_series['education'])
print(row_series.education)
```

HS-grad
HS-grad
HS-grad
HS-grad

[38]:
```
columns_series = adult_df.education
print(columns_series.loc[10002])
print(columns_series.iloc[2])
print(columns_series[10002])
# print(row_series.10002)  This will give syntax error!
```

```
HS-grad
HS-grad
HS-grad
```

3 Slicing

[39]:
```
my_array = np.array([[2,3,5,7],[11,13,17,19],
                     [23,29,31,37,], [41,43,47,49]])
my_array
```

[39]:
```
array([[ 2,  3,  5,  7],
       [11, 13, 17, 19],
       [23, 29, 31, 37],
       [41, 43, 47, 49]])
```

[40]:
```
my_array[1,1]
```

[40]:
```
np.int64(13)
```

[41]:
```
my_array[1,:]
```

[41]:
```
array([11, 13, 17, 19])
```

[42]:
```
my_array[:,1]
```

[42]:
```
array([ 3, 13, 29, 43])
```

[43]:
```
my_array
```

[43]:
```
array([[ 2,  3,  5,  7],
       [11, 13, 17, 19],
       [23, 29, 31, 37],
       [41, 43, 47, 49]])
```

[44]:
```
my_array[1:3,:]
```

[44]:
```
array([[11, 13, 17, 19],
       [23, 29, 31, 37]])
```

[45]:
```
my_array[1:3,0:2]
```

[45]:
```
array([[11, 13],
       [23, 29]])
```

[46]:
```
my_array[1:3,[0,2]]
```

[46]:
```
array([[11, 17],
       [23, 31]])
```

[47]:
```
adult_df.loc[:,'education':'occupation']
```

[47]:
```
        education  education-num      marital-status         occupation
10000   Bachelors             13       Never-married       Adm-clerical
10001   Bachelors             13  Married-civ-spouse    Exec-managerial
10002     HS-grad              9            Divorced  Handlers-cleaners
10003        11th              7  Married-civ-spouse  Handlers-cleaners
10004   Bachelors             13  Married-civ-spouse     Prof-specialty
…             …              …                   …                  …
42556  Assoc-acdm             12  Married-civ-spouse       Tech-support
42557     HS-grad              9  Married-civ-spouse  Machine-op-inspct
42558     HS-grad              9             Widowed       Adm-clerical
42559     HS-grad              9       Never-married       Adm-clerical
42560     HS-grad              9  Married-civ-spouse    Exec-managerial

[32561 rows x 4 columns]
```

[48]:
```
adult_df.sort_values('education-num').reset_index().iloc[1:32561:3617]
```

[48]:
```
       index  age  workclass  fnlwgt     education  education-num  \
1      42432   36    Private  208068     Preschool              1
3618   14676   17    Private  262511          11th              7
7235   17294   53    Private  141388       HS-grad              9
10852  11766   24    Private  228424       HS-grad              9
14469  25218   30    Private  144064       HS-grad              9
18086  31147   39    Private  348521  Some-college             10
21703  25115   35    Private  257042  Some-college             10
25320  25231   30  State-gov  193380     Bachelors             13
28937  40173   24    Private  330571     Bachelors             13
32554  18280   55  Local-gov   37869     Doctorate             16

           marital-status         occupation   relationship   race    sex  \
1                Divorced      Other-service  Not-in-family  Other   Male
3618        Never-married              Sales      Own-child  White   Male
7235   Married-civ-spouse   Transport-moving        Husband  White   Male
```

```
10852       Never-married  Handlers-cleaners  Other-relative  Black    Male
14469  Married-civ-spouse    Exec-managerial         Husband  White    Male
18086  Married-civ-spouse    Farming-fishing         Husband  White    Male
21703  Married-civ-spouse     Prof-specialty         Husband  White    Male
25320       Never-married     Prof-specialty  Other-relative  White    Male
28937  Married-civ-spouse     Prof-specialty            Wife  White  Female
32554       Never-married     Prof-specialty   Not-in-family  White  Female

       capitalGain  capitalLoss  hoursPerWeek  nativeCountry income
1                0            0            72         Mexico  <=50K
3618             0            0            20  United-States  <=50K
7235             0            0            40  United-States   >50K
10852            0            0            40  United-States  <=50K
14469            0            0            62  United-States  <=50K
18086            0         2415            99  United-States   >50K
21703            0            0            50  United-States  <=50K
25320            0            0            35  United-States  <=50K
28937            0            0            40  United-States  <=50K
32554            0            0            60  United-States  <=50K
```

4 Boolean Masking

[49]:
```python
twopowers_sr = pd.Series([1,2,4,8,16,32,64,128,256,512,1024])
BM = [False,False,False,True,False,False,False,True,True,True,True]
twopowers_sr[BM]
```

[49]:
```
3        8
7      128
8      256
9      512
10    1024
dtype: int64
```

[50]:
```python
twopowers_sr >=500
```

[50]:
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9      True
10     True
```

```
dtype: bool
```

[51]:
```
BM = twopowers_sr >=500
twopowers_sr[BM]
```

[51]:
```
9      512
10    1024
dtype: int64
```

[52]:
```
twopowers_sr[twopowers_sr >=500]
```

[52]:
```
9      512
10    1024
dtype: int64
```

[53]:
```
BM = adult_df.education == 'Preschool'
print('Mean: {}'.format(np.mean(adult_df[BM].age)))
print('Median: {}'.format(np.median(adult_df[BM].age)))
```

```
Mean: 42.76470588235294
Median: 41.0
```

[54]:
```
BM1 = adult_df['education-num'] > 10
BM2 = adult_df['education-num'] < 10

print('More than 10 years of education - Capital Gain: {}'
      .format(np.mean(adult_df[BM1].capitalGain)))
print('Less than 10 years of education - Capital Gain: {}'
      .format(np.mean(adult_df[BM2].capitalGain)))
```

```
More than 10 years of education - Capital Gain: 2230.9397109166985
Less than 10 years of education - Capital Gain: 492.25532059102613
```

5 Get to know a dataset

[55]:
```
adult_df.shape
```

[55]:
```
(32561, 15)
```

[56]:
```
adult_df.columns
```

[56]:
```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education-num',
       'marital-status', 'occupation', 'relationship', 'race', 'sex',
       'capitalGain', 'capitalLoss', 'hoursPerWeek', 'nativeCountry',
       'income'],
      dtype='object')
```

[57]:
```python
adult_df.columns = ['age', 'workclass', 'fnlwgt', 'education',
                    'education_num', 'marital_status', 'occupation',
                    'relationship', 'race', 'sex', 'capitalGain',
                    'capitalLoss', 'hoursPerWeek', 'nativeCountry',
                    'income']
```

[59]:
```python
adult_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 32561 entries, 10000 to 42560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       30725 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education_num   32561 non-null  int64
 5   marital_status  32561 non-null  object
 6   occupation      30718 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capitalGain     32561 non-null  int64
 11  capitalLoss     32561 non-null  int64
 12  hoursPerWeek    32561 non-null  int64
 13  nativeCountry   31978 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.0+ MB
```

[58]:
```python
adult_df.describe()
```

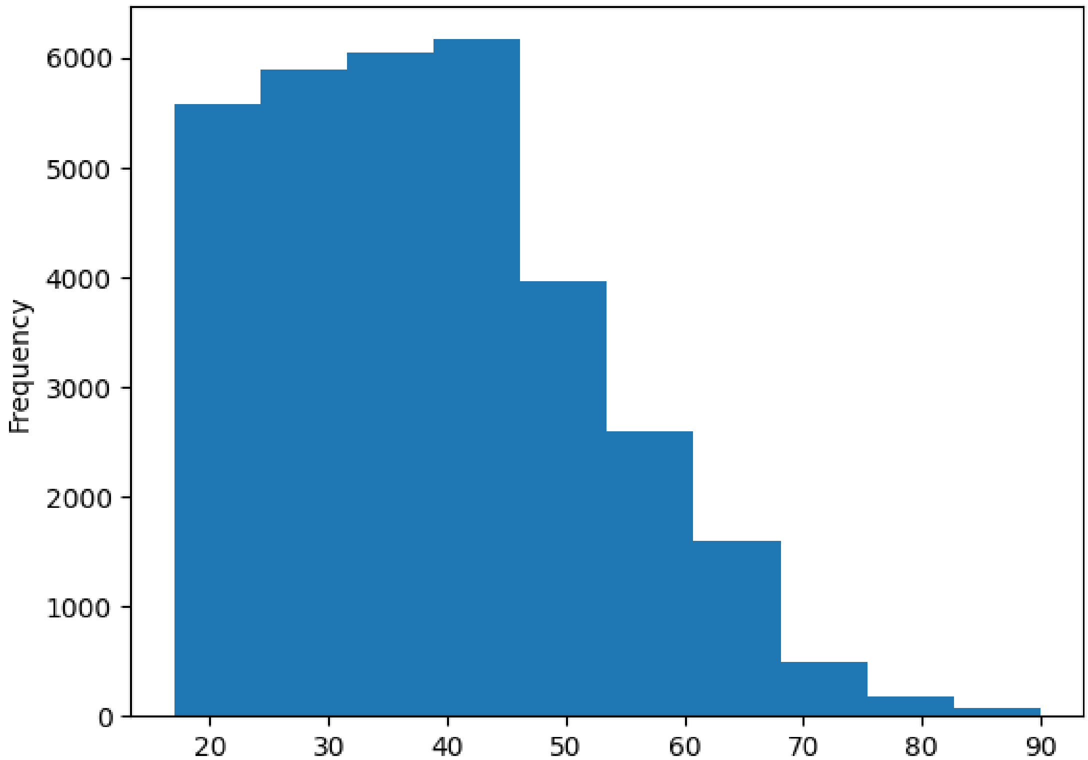
[58]:
```
                age        fnlwgt  education_num   capitalGain   capitalLoss  \
count  32561.000000  3.256100e+04   32561.000000  32561.000000  32561.000000
mean      38.581647  1.897784e+05      10.080679   1077.648844     87.303830
std       13.640433  1.055500e+05       2.572720   7385.292085    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

       hoursPerWeek
count  32561.000000
mean      40.437456
std       12.347429
min        1.000000
```

```
25%        40.000000
50%        40.000000
75%        45.000000
max        99.000000
```

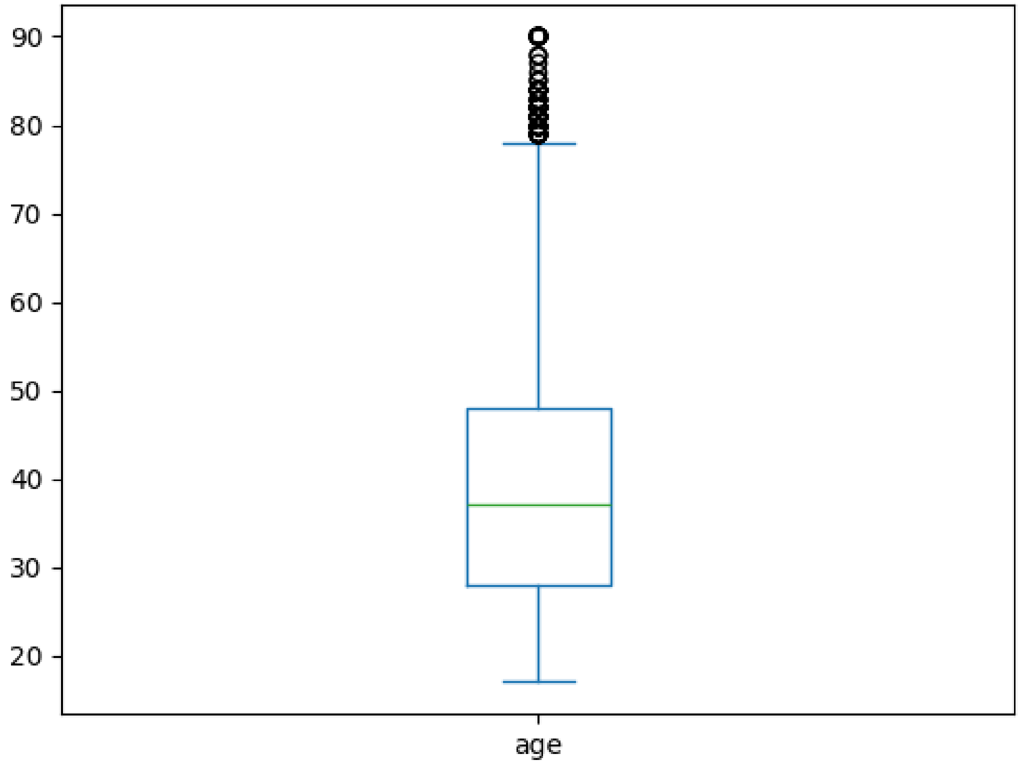
[63]:
```
adult_df.age.plot.hist()
```

[63]: <Axes: ylabel='Frequency'>



[64]:
```
adult_df.age.plot.box()
```

[64]: <Axes: >



[65]:
```
adult_df.relationship.unique()
```

[65]:
```
array(['Not-in-family', 'Husband', 'Wife', 'Own-child', 'Unmarried',
       'Other-relative'], dtype=object)
```

[66]:
```
adult_df.relationship.value_counts()
```

[66]:
```
relationship
Husband           13193
Not-in-family      8305
Own-child          5068
Unmarried          3446
Wife               1568
Other-relative      981
Name: count, dtype: int64
```

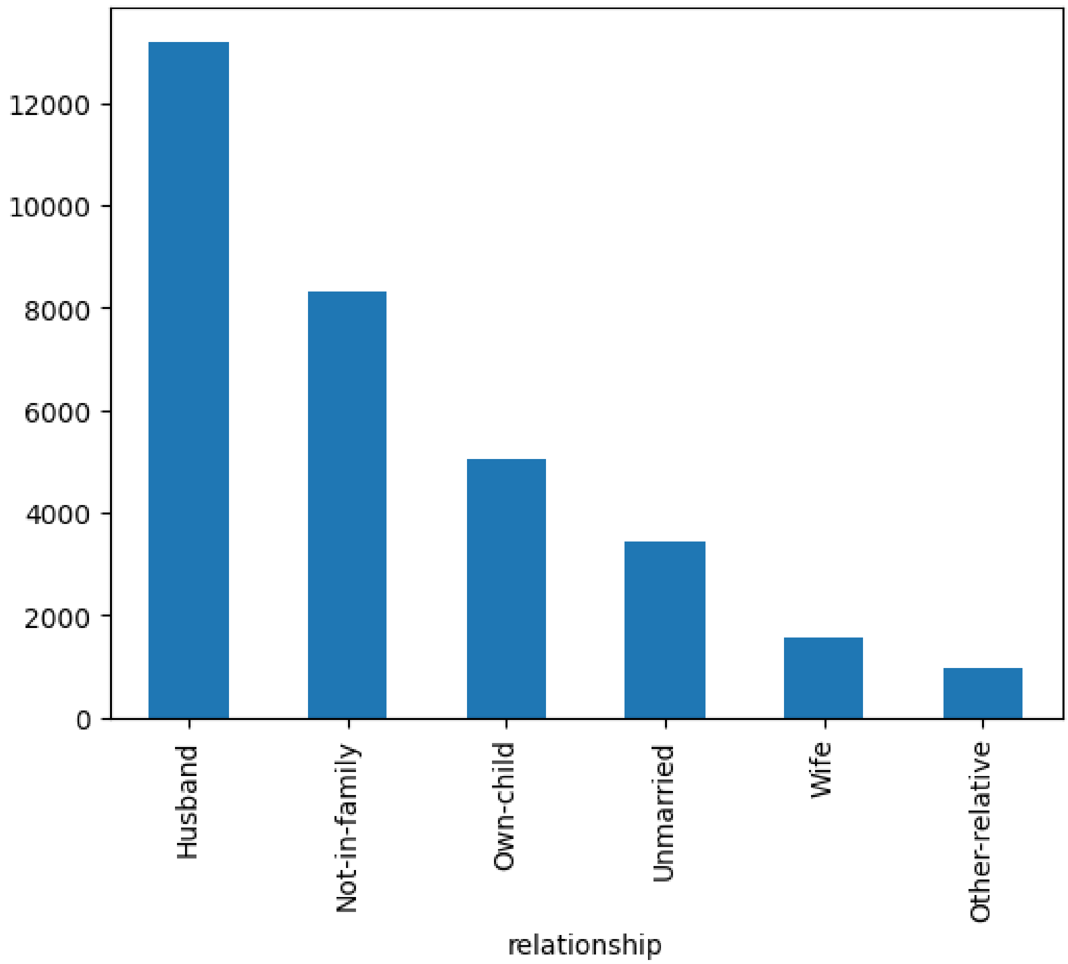
[67]:
```
adult_df.relationship.value_counts(normalize=True)
```

[67]:
```
relationship
Husband           0.405178
Not-in-family     0.255060
Own-child         0.155646
```

```
Unmarried         0.105832
Wife              0.048156
Other-relative    0.030128
Name: proportion, dtype: float64
```

[68]:
```
adult_df.relationship.value_counts().plot.bar()
```

[68]: <Axes: xlabel='relationship'>



6 Appy a function

[71]:
```
def MultiplyBy2(n):
    return n*2

adult_df.age.apply(MultiplyBy2)
```

[71]:
```
10000     78
10001    100
10002     76
10003    106
10004     56
        …
42556     54
42557     80
42558    116
42559     44
42560    104
Name: age, Length: 32561, dtype: int64
```

6.0.1 Applying a Function - Analytic Example 1

Divide every value in column fnlwgt by the sum of all its values.

[72]:
```python
total_fnlwgt = adult_df.fnlwgt.sum()

def CalculatePercentage(v):
    return v/total_fnlwgt*100

adult_df.fnlwgt = adult_df.fnlwgt.apply(CalculatePercentage)
adult_df
```

[72]:
```
       age          workclass    fnlwgt   education  education_num  \
10000   39          State-gov  0.001254   Bachelors             13
10001   50   Self-emp-not-inc  0.001348   Bachelors             13
10002   38            Private  0.003490     HS-grad              9
10003   53            Private  0.003798        11th              7
10004   28            Private  0.005476   Bachelors             13
…     …                   …         …           …             …
42556   27            Private  0.004164  Assoc-acdm             12
42557   40            Private  0.002498     HS-grad              9
42558   58            Private  0.002458     HS-grad              9
42559   22            Private  0.003261     HS-grad              9
42560   52       Self-emp-inc  0.004659     HS-grad              9

           marital_status         occupation   relationship   race     sex  \
10000       Never-married       Adm-clerical  Not-in-family  White    Male
10001  Married-civ-spouse    Exec-managerial        Husband  White    Male
10002            Divorced  Handlers-cleaners  Not-in-family  White    Male
10003  Married-civ-spouse  Handlers-cleaners        Husband  Black    Male
10004  Married-civ-spouse     Prof-specialty           Wife  Black  Female
…                       …                  …            …      …       …
42556  Married-civ-spouse       Tech-support           Wife  White  Female
42557  Married-civ-spouse  Machine-op-inspct        Husband  White    Male
42558             Widowed       Adm-clerical      Unmarried  White  Female
```

```
42559       Never-married       Adm-clerical       Own-child  White    Male
42560  Married-civ-spouse    Exec-managerial            Wife  White  Female

       capitalGain  capitalLoss  hoursPerWeek  nativeCountry income
10000         2174            0            40  United-States  <=50K
10001            0            0            13  United-States  <=50K
10002            0            0            40  United-States  <=50K
10003            0            0            40  United-States  <=50K
10004            0            0            40           Cuba  <=50K
…              …            …            …             …    …
42556            0            0            38  United-States  <=50K
42557            0            0            40  United-States   >50K
42558            0            0            40  United-States  <=50K
42559            0            0            20  United-States  <=50K
42560        15024            0            40  United-States   >50K

[32561 rows x 15 columns]
```

[73]:
```python
total_fnlwgt = adult_df.fnlwgt.sum()

adult_df.fnlwgt = adult_df.fnlwgt.apply(lambda v: v/total_fnlwgt*100)
adult_df
```

[73]:
```
       age         workclass    fnlwgt   education  education_num  \
10000   39         State-gov  0.001254   Bachelors             13
10001   50  Self-emp-not-inc  0.001348   Bachelors             13
10002   38           Private  0.003490     HS-grad              9
10003   53           Private  0.003798        11th              7
10004   28           Private  0.005476   Bachelors             13
…      …               …       …         …            …
42556   27           Private  0.004164  Assoc-acdm             12
42557   40           Private  0.002498     HS-grad              9
42558   58           Private  0.002458     HS-grad              9
42559   22           Private  0.003261     HS-grad              9
42560   52      Self-emp-inc  0.004659     HS-grad              9

           marital_status         occupation   relationship   race     sex  \
10000       Never-married       Adm-clerical  Not-in-family  White    Male
10001  Married-civ-spouse    Exec-managerial        Husband  White    Male
10002            Divorced  Handlers-cleaners  Not-in-family  White    Male
10003  Married-civ-spouse  Handlers-cleaners        Husband  Black    Male
10004  Married-civ-spouse     Prof-specialty           Wife  Black  Female
…                     …                  …            …      …     …
42556  Married-civ-spouse       Tech-support           Wife  White  Female
42557  Married-civ-spouse  Machine-op-inspct        Husband  White    Male
42558             Widowed       Adm-clerical      Unmarried  White  Female
42559       Never-married       Adm-clerical      Own-child  White    Male
```

```
42560  Married-civ-spouse    Exec-managerial          Wife  White  Female

       capitalGain  capitalLoss  hoursPerWeek  nativeCountry income
10000         2174            0            40  United-States  <=50K
10001            0            0            13  United-States  <=50K
10002            0            0            40  United-States  <=50K
10003            0            0            40  United-States  <=50K
10004            0            0            40           Cuba  <=50K
…              …            …            …              …      …
42556            0            0            38  United-States  <=50K
42557            0            0            40  United-States   >50K
42558            0            0            40  United-States  <=50K
42559            0            0            20  United-States  <=50K
42560        15024            0            40  United-States   >50K

[32561 rows x 15 columns]
```

[74]:
```python
def CalcLifeNoEd(row):
    return row.age - row.education_num

adult_df.apply(CalcLifeNoEd,axis=1)
```

[74]:
```
10000    26
10001    37
10002    29
10003    46
10004    15
         ..
42556    15
42557    31
42558    49
42559    13
42560    43
Length: 32561, dtype: int64
```

[75]:
```python
adult_df.apply(lambda r: r.age-r.education_num,axis=1)
```

[75]:
```
10000    26
10001    37
10002    29
10003    46
10004    15
         ..
42556    15
42557    31
42558    49
42559    13
```

```
42560    43
Length: 32561, dtype: int64
```

[76]:
```
adult_df['lifeNoEd'] = adult_df.apply(
    lambda r: r.age-r.education_num,axis=1)

adult_df['capitalNet'] = adult_df.apply(
    lambda r: r.capitalGain - r.capitalLoss,axis=1)

adult_df[['education_num','lifeNoEd','capitalNet']].corr()
```

[76]:

	education_num	lifeNoEd	capitalNet
education_num	1.000000	-0.150452	0.117891
lifeNoEd	-0.150452	1.000000	0.051490
capitalNet	0.117891	0.051490	1.000000

7 Groupby

[77]:
```
adult_df.groupby('marital_status').size()
```

[77]:
```
marital_status
Divorced                  4443
Married-AF-spouse           23
Married-civ-spouse       14976
Married-spouse-absent      418
Never-married            10683
Separated                 1025
Widowed                    993
dtype: int64
```

[81]:
```
adult_df.groupby(['marital_status', 'sex']).size()
```

[81]:
```
marital_status         sex
Divorced               Female     2672
                       Male       1771
Married-AF-spouse      Female       14
                       Male          9
Married-civ-spouse     Female     1657
                       Male      13319
Married-spouse-absent  Female      205
                       Male        213
Never-married          Female     4767
                       Male       5916
Separated              Female      631
                       Male        394
Widowed                Female      825
                       Male        168
```

```
dtype: int64
```

[82]:
```python
adult_df.groupby(['marital_status','sex']).age.median()
```

[82]:
```
marital_status         sex
Divorced               Female    43.0
                       Male      42.0
Married-AF-spouse      Female    31.0
                       Male      29.0
Married-civ-spouse     Female    38.0
                       Male      43.0
Married-spouse-absent  Female    39.0
                       Male      41.0
Never-married          Female    25.0
                       Male      25.0
Separated              Female    39.0
                       Male      38.0
Widowed                Female    60.0
                       Male      62.5
Name: age, dtype: float64
```

[83]:
```python
adult_df.groupby(['race','sex']).capitalNet.mean()
```

[83]:
```
race                sex
Amer-Indian-Eskimo  Female     530.142857
                    Male       628.864583
Asian-Pac-Islander  Female     727.583815
                    Male      1707.440115
Black               Female     471.142765
                    Male       627.268324
Other               Female     218.385321
                    Male      1314.438272
White               Female     508.219857
                    Male      1266.413112
Name: capitalNet, dtype: float64
```

[84]:
```python
adult_df.groupby(['race', 'sex', 'income']).fnlwgt.mean()
```

[84]:
```
race                sex     income
Amer-Indian-Eskimo  Female  <=50K     0.001764
                            >50K      0.002395
                    Male    <=50K     0.002046
                            >50K      0.001954
Asian-Pac-Islander  Female  <=50K     0.002398
                            >50K      0.002305
                    Male    <=50K     0.002652
                            >50K      0.002762
```

```
Black                Female  <=50K     0.003454
                             >50K      0.003331
                     Male    <=50K     0.003922
                             >50K      0.003971
Other                Female  <=50K     0.002803
                             >50K      0.002593
                     Male    <=50K     0.003478
                             >50K      0.003310
White                Female  <=50K     0.002969
                             >50K      0.002978
                     Male    <=50K     0.003074
                             >50K      0.003025
Name: fnlwgt, dtype: float64
```

[85]:
```python
grb_result =adult_df.groupby(['race','sex']).capitalNet.mean()

print(grb_result.index)
```

```
MultiIndex([('Amer-Indian-Eskimo', 'Female'),
            ('Amer-Indian-Eskimo',   'Male'),
            ('Asian-Pac-Islander', 'Female'),
            ('Asian-Pac-Islander',   'Male'),
            (             'Black', 'Female'),
            (             'Black',   'Male'),
            (             'Other', 'Female'),
            (             'Other',   'Male'),
            (             'White', 'Female'),
            (             'White',   'Male')],
           names=['race', 'sex'])
```

[86]:
```python
grb_result =adult_df.groupby(['race','sex']).capitalNet.mean()
grb_result
```

[86]:
```
race                sex
Amer-Indian-Eskimo  Female     530.142857
                    Male       628.864583
Asian-Pac-Islander  Female     727.583815
                    Male      1707.440115
Black               Female     471.142765
                    Male       627.268324
Other               Female     218.385321
                    Male      1314.438272
White               Female     508.219857
                    Male      1266.413112
Name: capitalNet, dtype: float64
```

[87]:
```python
grb_result.unstack()
```

[87]:
```
sex                       Female         Male
race
Amer-Indian-Eskimo    530.142857   628.864583
Asian-Pac-Islander    727.583815  1707.440115
Black                 471.142765   627.268324
Other                 218.385321  1314.438272
White                 508.219857  1266.413112
```

[88]:
```
mlt_seris =adult_df.groupby(['race','sex','income']).fnlwgt.mean()
mlt_seris
```

[88]:
```
race                sex     income
Amer-Indian-Eskimo  Female  <=50K     0.001764
                            >50K      0.002395
                    Male    <=50K     0.002046
                            >50K      0.001954
Asian-Pac-Islander  Female  <=50K     0.002398
                            >50K      0.002305
                    Male    <=50K     0.002652
                            >50K      0.002762
Black               Female  <=50K     0.003454
                            >50K      0.003331
                    Male    <=50K     0.003922
                            >50K      0.003971
Other               Female  <=50K     0.002803
                            >50K      0.002593
                    Male    <=50K     0.003478
                            >50K      0.003310
White               Female  <=50K     0.002969
                            >50K      0.002978
                    Male    <=50K     0.003074
                            >50K      0.003025
Name: fnlwgt, dtype: float64
```

[89]:
```
mlt_seris.unstack()
```

[89]:
```
income                        <=50K      >50K
race               sex
Amer-Indian-Eskimo Female  0.001764  0.002395
                   Male    0.002046  0.001954
Asian-Pac-Islander Female  0.002398  0.002305
                   Male    0.002652  0.002762
Black              Female  0.003454  0.003331
                   Male    0.003922  0.003971
Other              Female  0.002803  0.002593
                   Male    0.003478  0.003310
White              Female  0.002969  0.002978
```

```
                   Male    0.003074  0.003025
```

[90]:
```python
mlt_seris.unstack().unstack()
```

[90]:
```
income                    <=50K                >50K
sex                      Female      Male    Female      Male
race
Amer-Indian-Eskimo     0.001764  0.002046  0.002395  0.001954
Asian-Pac-Islander     0.002398  0.002652  0.002305  0.002762
Black                  0.003454  0.003922  0.003331  0.003971
Other                  0.002803  0.003478  0.002593  0.003310
White                  0.002969  0.003074  0.002978  0.003025
```

[91]:
```python
mlt_df= mlt_seris.unstack().unstack()
mlt_df.columns
```

[91]:
```
MultiIndex([('<=50K', 'Female'),
            ('<=50K',   'Male'),
            ( '>50K', 'Female'),
            ( '>50K',   'Male')],
           names=['income', 'sex'])
```

[92]:
```python
mlt_df.stack()
```

```
/tmp/ipykernel_2332935/2163858474.py:1: FutureWarning: The previous
implementation of stack is deprecated and will be removed in a future version of
pandas. See the What's New notes for pandas 2.1.0 for details. Specify
future_stack=True to adopt the new implementation and silence this warning.
  mlt_df.stack()
```

[92]:
```
income                              <=50K      >50K
race                 sex
Amer-Indian-Eskimo   Female      0.001764  0.002395
                     Male        0.002046  0.001954
Asian-Pac-Islander   Female      0.002398  0.002305
                     Male        0.002652  0.002762
Black                Female      0.003454  0.003331
                     Male        0.003922  0.003971
Other                Female      0.002803  0.002593
                     Male        0.003478  0.003310
White                Female      0.002969  0.002978
                     Male        0.003074  0.003025
```

[93]:
```python
mlt_df.stack().stack()
```

```
/tmp/ipykernel_2332935/2517862891.py:1: FutureWarning: The previous
implementation of stack is deprecated and will be removed in a future version of
pandas. See the What's New notes for pandas 2.1.0 for details. Specify
```

```
future_stack=True to adopt the new implementation and silence this warning.
  mlt_df.stack().stack()
```

[93]:
```
race                sex     income
Amer-Indian-Eskimo  Female  <=50K     0.001764
                            >50K      0.002395
                    Male    <=50K     0.002046
                            >50K      0.001954
Asian-Pac-Islander  Female  <=50K     0.002398
                            >50K      0.002305
                    Male    <=50K     0.002652
                            >50K      0.002762
Black               Female  <=50K     0.003454
                            >50K      0.003331
                    Male    <=50K     0.003922
                            >50K      0.003971
Other               Female  <=50K     0.002803
                            >50K      0.002593
                    Male    <=50K     0.003478
                            >50K      0.003310
White               Female  <=50K     0.002969
                            >50K      0.002978
                    Male    <=50K     0.003074
                            >50K      0.003025
dtype: float64
```

8 Pivot & Melt

[94]:
```
wide_df = pd.read_csv('wide.csv')
wide_df
```

[94]:

	ReadingDateTime	NO	NO2	NOX	PM10	PM2.5
0	01/01/2017 00:00	3.5	30.8	36.2	35.7	31.0
1	01/01/2017 01:00	3.6	31.5	37.0	28.5	31.0
2	01/01/2017 02:00	2.2	27.3	30.7	22.7	31.0

[95]:
```
wide_df.melt(id_vars='ReadingDateTime',
             value_vars=['NO','NO2','NOX','PM10','PM2.5'],
             var_name='Species',
             value_name='Value')
```

[95]:

	ReadingDateTime	Species	Value
0	01/01/2017 00:00	NO	3.5
1	01/01/2017 01:00	NO	3.6
2	01/01/2017 02:00	NO	2.2
3	01/01/2017 00:00	NO2	30.8
4	01/01/2017 01:00	NO2	31.5

```
5   01/01/2017 02:00     NO2   27.3
6   01/01/2017 00:00     NOX   36.2
7   01/01/2017 01:00     NOX   37.0
8   01/01/2017 02:00     NOX   30.7
9   01/01/2017 00:00    PM10   35.7
10  01/01/2017 01:00    PM10   28.5
11  01/01/2017 02:00    PM10   22.7
12  01/01/2017 00:00   PM2.5   31.0
13  01/01/2017 01:00   PM2.5   31.0
14  01/01/2017 02:00   PM2.5   31.0
```

[96]:

```
long_df = pd.read_csv('long.csv')
long_df
```

[96]:

```
     ReadingDateTime Species  Value
0   01/01/2017 00:00      NO    3.5
1   01/01/2017 01:00      NO    3.6
2   01/01/2017 02:00      NO    2.2
3   01/01/2017 00:00     NO2   30.8
4   01/01/2017 01:00     NO2   31.5
5   01/01/2017 02:00     NO2   27.3
6   01/01/2017 00:00     NOX   36.2
7   01/01/2017 01:00     NOX   37.0
8   01/01/2017 02:00     NOX   30.7
9   01/01/2017 00:00    PM10   35.7
10  01/01/2017 01:00    PM10   28.5
11  01/01/2017 02:00    PM10   22.7
12  01/01/2017 00:00   PM2.5   31.0
13  01/01/2017 01:00   PM2.5   31.0
14  01/01/2017 02:00   PM2.5   31.0
```

[97]:

```
long_df.pivot(index='ReadingDateTime',
              columns='Species',
              values='Value')
```

[97]:

```
Species            NO   NO2   NOX  PM10  PM2.5
ReadingDateTime
01/01/2017 00:00  3.5  30.8  36.2  35.7   31.0
01/01/2017 01:00  3.6  31.5  37.0  28.5   31.0
01/01/2017 02:00  2.2  27.3  30.7  22.7   31.0
```