# Assignment 3 Descriptive Statistics Adult data

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
df = pd.read_csv("data.csv")
df
```

Out[2]:

| | age | workclass | fnlwgt | education | educational-num | marital-status | occupation | relationship | race | gender | cap |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 25 | Private | 226802 | 11th | 7 | Never-married | Machine-op-inspct | Own-child | Black | Male | |
| **1** | 38 | Private | 89814 | HS-grad | 9 | Married-civ-spouse | Farming-fishing | Husband | White | Male | |
| **2** | 28 | Local-gov | 336951 | Assoc-acdm | 12 | Married-civ-spouse | Protective-serv | Husband | White | Male | |
| **3** | 44 | Private | 160323 | Some-college | 10 | Married-civ-spouse | Machine-op-inspct | Husband | Black | Male | |
| **4** | 18 | ? | 103497 | Some-college | 10 | Never-married | ? | Own-child | White | Female | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **48837** | 27 | Private | 257302 | Assoc-acdm | 12 | Married-civ-spouse | Tech-support | Wife | White | Female | |
| **48838** | 40 | Private | 154374 | HS-grad | 9 | Married-civ-spouse | Machine-op-inspct | Husband | White | Male | |
| **48839** | 58 | Private | 151910 | HS-grad | 9 | Widowed | Adm-clerical | Unmarried | White | Female | |
| **48840** | 22 | Private | 201490 | HS-grad | 9 | Never-married | Adm-clerical | Own-child | White | Male | |
| **48841** | 52 | Self-emp-inc | 287927 | HS-grad | 9 | Married-civ-spouse | Exec-managerial | Wife | White | Female | 1 |

48842 rows × 15 columns

In [3]:
```
df.columns
```

Out[3]:
```
Index(['age', 'workclass', 'fnlwgt', 'education', 'educational-num',
       'marital-status', 'occupation', 'relationship', 'race', 'gender',
       'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
       'income'],
      dtype='object')
```

In [4]:
```
df.shape
```

Out[4]:
```
(48842, 15)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             48842 non-null  int64 
 1   workclass       48842 non-null  object
 2   fnlwgt          48842 non-null  int64 
 3   education       48842 non-null  object
 4   educational-num 48842 non-null  int64 
 5   marital-status  48842 non-null  object
 6   occupation      48842 non-null  object
 7   relationship    48842 non-null  object
 8   race            48842 non-null  object
 9   gender          48842 non-null  object
 10  capital-gain    48842 non-null  int64 
 11  capital-loss    48842 non-null  int64 
 12  hours-per-week  48842 non-null  int64 
 13  native-country  48842 non-null  object
 14  income          48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
age               0
workclass         0
fnlwgt            0
education         0
educational-num   0
marital-status    0
occupation        0
relationship      0
race              0
gender            0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
```

In [7]:
```
df.describe()
```

Out[7]:

| | age | fnlwgt | educational-num | capital-gain | capital-loss | hours-per-week |
|---|---|---|---|---|---|---|
| **count** | 48842.000000 | 4.884200e+04 | 48842.000000 | 48842.000000 | 48842.000000 | 48842.000000 |
| **mean** | 38.643585 | 1.896641e+05 | 10.078089 | 1079.067626 | 87.502314 | 40.422382 |
| **std** | 13.710510 | 1.056040e+05 | 2.570973 | 7452.019058 | 403.004552 | 12.391444 |
| **min** | 17.000000 | 1.228500e+04 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| **25%** | 28.000000 | 1.175505e+05 | 9.000000 | 0.000000 | 0.000000 | 40.000000 |
| **50%** | 37.000000 | 1.781445e+05 | 10.000000 | 0.000000 | 0.000000 | 40.000000 |
| **75%** | 48.000000 | 2.376420e+05 | 12.000000 | 0.000000 | 0.000000 | 45.000000 |
| **max** | 90.000000 | 1.490400e+06 | 16.000000 | 99999.000000 | 4356.000000 | 99.000000 |

In [8]:
```
df.age.min()
```

Out[8]:
```
17
```

In [9]:
```
df['age'].max()
```

Out[9]:
```
90
```

In [10]:
```
df.age.mean()
```

Out[10]:
```
38.64358543876172
```

In [11]:
```
df.age.std()
```

Out[11]:
```
13.710509934443557
```

In [12]:
```
df.age.median()
```

Out[12]:
```
37.0
```

In [13]:
```
df['income'].unique()
```

Out[13]:
```
array(['<=50K', '>50K'], dtype=object)
```

In [14]:
```
df['income'].nunique()   ##count the unique
```

Out[14]:
```
2
```

In [15]:
```
df.groupby(['income','age']).count()
```

Out[15]:

| income | age | workclass | fnlwgt | education | educational-num | marital-status | occupation | relationship | race | gender | capi g |
|---|---|---|---|---|---|---|---|---|---|---|---|
| <=50K | 17 | 595 | 595 | 595 | 595 | 595 | 595 | 595 | 595 | 595 | |
| | 18 | 862 | 862 | 862 | 862 | 862 | 862 | 862 | 862 | 862 | |
| | 19 | 1050 | 1050 | 1050 | 1050 | 1050 | 1050 | 1050 | 1050 | 1050 | 1 |
| | 20 | 1112 | 1112 | 1112 | 1112 | 1112 | 1112 | 1112 | 1112 | 1112 | 1 |
| | 21 | 1090 | 1090 | 1090 | 1090 | 1090 | 1090 | 1090 | 1090 | 1090 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| >50K | 83 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| | 84 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | 85 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | 88 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | 90 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | |

142 rows × 13 columns

In [16]:
```
## total 142 rows for group
```

# Group By

In [17]:
```
df.groupby(['income','age']).min()
```

Out[17]:

| income | age | workclass | fnlwgt | education | educational- | marital- | occupation | relationship | race | gender |
|---|---|---|---|---|---|---|---|---|---|---|

| | | | | | num | status | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **income** | **age** | | | | | | | | | |
| **<=50K** | **17** | ? | 19752 | 10th | 3 | Married-civ-spouse | ? | Husband | Amer-Indian-Eskimo | Female |
| | **18** | ? | 20057 | 10th | 3 | Divorced | ? | Husband | Amer-Indian-Eskimo | Female |
| | **19** | ? | 20469 | 10th | 1 | Divorced | ? | Husband | Amer-Indian-Eskimo | Female |
| | **20** | ? | 19410 | 10th | 1 | Divorced | ? | Husband | Amer-Indian-Eskimo | Female |
| | **21** | ? | 20728 | 10th | 1 | Divorced | ? | Husband | Amer-Indian-Eskimo | Female |
| **...** | **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **>50K** | **83** | Self-emp-inc | 153183 | 10th | 6 | Married-civ-spouse | Exec-managerial | Husband | White | Male |
| | **84** | Self-emp-inc | 172907 | Some-college | 10 | Married-civ-spouse | Sales | Husband | White | Male |
| | **85** | Self-emp-inc | 155981 | Bachelors | 13 | Widowed | Exec-managerial | Not-in-family | White | Male |
| | **88** | Self-emp-not-inc | 263569 | 11th | 7 | Married-civ-spouse | Farming-fishing | Husband | White | Male |
| | **90** | ? | 46786 | Assoc-acdm | 9 | Married-civ-spouse | ? | Husband | Black | Female |

142 rows × 13 columns

In [18]:
```
df.groupby(['income','age']).max()
```

Out[18]:

| | | workclass | fnlwgt | education | educational-num | marital-status | occupation | relationship | race | gender |
|---|---|---|---|---|---|---|---|---|---|---|
| **income** | **age** | | | | | | | | | |
| **<=50K** | **17** | State-gov | 806316 | Some-college | 10 | Widowed | Transport-moving | Unmarried | White | Male |
| | **18** | State-gov | 761006 | Some-college | 14 | Widowed | Transport-moving | Wife | White | Male |
| | **19** | Without-pay | 1047822 | Some-college | 13 | Separated | Transport-moving | Wife | White | Male |
| | **20** | State-gov | 745817 | Some-college | 14 | Separated | Transport-moving | Wife | White | Male |
| | **21** | Without-pay | 811615 | Some-college | 14 | Widowed | Transport-moving | Wife | White | Male |
| **...** | **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **>50K** | **83** | Self-emp-inc | 240150 | Bachelors | 13 | Married-civ- | Farming-fishing | Husband | White | Male |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | spouse | | | | |
| **84** | Self-emp-inc | 172907 | Some-college | 10 | Married-civ-spouse | Sales | Husband | White | Male |
| **85** | Self-emp-inc | 155981 | Bachelors | 13 | Widowed | Exec-managerial | Not-in-family | White | Male |
| **88** | Self-emp-not-inc | 263569 | 11th | 7 | Married-civ-spouse | Farming-fishing | Husband | White | Male |
| **90** | Self-emp-not-inc | 313986 | Prof-school | 15 | Never-married | Sales | Wife | White | Male |

142 rows × 13 columns

In [35]:
```
# df.groupby(['income','age']).mean()
```

In [22]:
```
df.groupby("income")['age'].count()  ## age specified
```

Out[22]:
```
income
<=50K    37155
>50K     11687
Name: age, dtype: int64
```

In [24]:
```
df.groupby("income").count()  ## for all
```

Out[24]:

| income | age | workclass | fnlwgt | education | educational-num | marital-status | occupation | relationship | race | gender |
|---|---|---|---|---|---|---|---|---|---|---|
| **<=50K** | 37155 | 37155 | 37155 | 37155 | 37155 | 37155 | 37155 | 37155 | 37155 | 37155 |
| **>50K** | 11687 | 11687 | 11687 | 11687 | 11687 | 11687 | 11687 | 11687 | 11687 | 11687 |

In [25]:
```
df.groupby("income")['age'].min()
```

Out[25]:
```
income
<=50K    17
>50K     19
Name: age, dtype: int64
```

In [26]:
```
df.groupby("income")['age'].max()
```

Out[26]:
```
income
<=50K    90
>50K     90
Name: age, dtype: int64
```

In [27]:
```
df.groupby("income")['age'].mean()
```

Out[27]:
```
income
<=50K    36.872184
>50K     44.275178
Name: age, dtype: float64
```

In [28]:
```
df.groupby("income")['age'].median()
```

Out[28]:
```
income
<=50K    34.0
>50K     43.0
Name: age, dtype: float64
```

In [29]:
```
df.groupby("income")['age'].std()
```

Out[29]:
```
income
<=50K    14.104118
>50K     10.558983
Name: age, dtype: float64
```

In [30]:
```
df.groupby(["income","age"])['hours-per-week'].min()
```

Out[30]:
```
income  age
<=50K   17      4
        18      2
        19      2
        20      2
        21      1
               ..
>50K    83     50
        84     35
        85     40
        88     40
        90     15
Name: hours-per-week, Length: 142, dtype: int64
```

In [39]:
```
#summary statistics of age grouped by gender
```

In [36]:
```
df.groupby("gender")["age"].describe()
```

Out[36]:

| gender | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Female** | 16192.0 | 36.927989 | 14.137423 | 17.0 | 25.0 | 35.0 | 46.0 | 90.0 |
| **Male** | 32650.0 | 39.494395 | 13.412850 | 17.0 | 29.0 | 38.0 | 48.0 | 90.0 |

In [37]:
```
df.groupby("marital-status")["age"].mean()
```

Out[37]:
```
marital-status
Divorced                43.159204
Married-AF-spouse       31.945946
Married-civ-spouse      43.353724
Married-spouse-absent   40.613057
Never-married           28.128064
Separated               39.725490
Widowed                 59.377470
Name: age, dtype: float64
```

In [38]:
```
 df.groupby("marital-status")["age"].median()
```

Out[38]:
```
marital-status
Divorced                42.0
Married-AF-spouse       30.0
Married-civ-spouse      42.0
Married-spouse-absent   40.0
Never-married           25.0
Separated               39.0
Widowed                 60.0
Name: age, dtype: float64
```

In [40]:
```
#grouping can be done on multiple columns
# summary statistics of age grouped by gender & marital-status
df.groupby(["gender","marital-status"])["age"].std()
```

Out[40]:
```
gender  marital-status
Female  Divorced                10.794868
        Married-AF-spouse       12.342744
```

```
        Married-civ-spouse       11.402805
        Married-spouse-absent    13.019854
        Never-married            10.231671
        Separated                10.757639
        Widowed                  11.657268
Male    Divorced                 10.161659
        Married-AF-spouse         6.336522
        Married-civ-spouse       12.080786
        Married-spouse-absent    12.631023
        Never-married             9.717602
        Separated                10.811704
        Widowed                  14.216489
Name: age, dtype: float64
```

In [41]:
```
#Count number of records by category
#The value_counts() method counts the number of records for each category in a column.
df["marital-status"].value_counts()
```

Out[41]:
```
marital-status
Married-civ-spouse       22379
Never-married            16117
Divorced                  6633
Separated                 1530
Widowed                   1518
Married-spouse-absent      628
Married-AF-spouse           37
Name: count, dtype: int64
```

# Using User Defined functions:

In [45]:
```
income_less_than_50 = df[df["income"]=="<=50K"]
print("Less than 50K",income_less_than_50.head())
income_greater_than_50 = df[df["income"]==">50K"]
print("Greater than 50K",income_greater_than_50.head())
```

```
Less than 50K    age workclass  fnlwgt     education  educational-num      marital-statu
s
0   25   Private  226802          11th               7       Never-married  \
1   38   Private   89814       HS-grad               9  Married-civ-spouse
4   18         ?  103497  Some-college              10       Never-married
5   34   Private  198693          10th               6       Never-married
6   29         ?  227026       HS-grad               9       Never-married

          occupation   relationship   race  gender  capital-gain
0  Machine-op-inspct      Own-child  Black    Male             0  \
1    Farming-fishing        Husband  White    Male             0
4                  ?      Own-child  White  Female             0
5      Other-service  Not-in-family  White    Male             0
6                  ?      Unmarried  Black    Male             0

   capital-loss  hours-per-week native-country income
0             0              40  United-States  <=50K
1             0              50  United-States  <=50K
4             0              30  United-States  <=50K
5             0              30  United-States  <=50K
6             0              40  United-States  <=50K
Greater than 50K     age         workclass  fnlwgt     education  educational-num
2    28         Local-gov  336951    Assoc-acdm              12  \
3    44           Private  160323  Some-college              10
7    63  Self-emp-not-inc  104626   Prof-school              15
10   65           Private  184454       HS-grad               9
14   48           Private  279724       HS-grad               9

        marital-status         occupation relationship   race gender
```

```
2   Married-civ-spouse    Protective-serv      Husband  White   Male  \
3   Married-civ-spouse  Machine-op-inspct      Husband  Black   Male
7   Married-civ-spouse     Prof-specialty      Husband  White   Male
10  Married-civ-spouse  Machine-op-inspct      Husband  White   Male
14  Married-civ-spouse  Machine-op-inspct      Husband  White   Male

    capital-gain  capital-loss  hours-per-week native-country income
2              0             0              40  United-States   >50K
3           7688             0              40  United-States   >50K
7           3103             0              32  United-States   >50K
10          6418             0              40  United-States   >50K
14          3103             0              48  United-States   >50K
```

In [46]:

```
def display_statistics(income_data,income_class):
    column = ["age","fnlwgt","educational-num","capital-gain","capital-loss","hours-per-
    print("Statistics for Income - ",income_class)
    print("-------------------------------------------------------")
    print("Mean:")
    print(income_data[column].mean())
    print("\n")

    print("-------------------------------------------------------")
    print("Median:")
    print(income_data[column].median())
    print("\n")

    print("-------------------------------------------------------")
    print("Standard Deviation:")
    print(income_data[column].std())
    print("\n")

    print("25% Percentile:")
    print(income_data[column].quantile(0.25))
    print("\n")

    print("75% Percentile:")
    print(income_data[column].quantile(0.75))
    print("\n")

    print("Minimum:")
    print(income_data[column].min())
    print("\n")

    print("Maximum:")
    print(income_data[column].max())


display_statistics(income_less_than_50,"<=50K")
print("\n")

display_statistics(income_greater_than_50,">50K")
```

```
Statistics for Income -  <=50K
-------------------------------------------------------
Mean:
age                   36.872184
fnlwgt             190039.565523
educational-num        9.598493
capital-gain         147.010308
capital-loss          54.151931
hours-per-week        38.840048
dtype: float64


-------------------------------------------------------
Median:
```

```
age                   34.0
fnlwgt             178811.0
educational-num         9.0
capital-gain            0.0
capital-loss            0.0
hours-per-week         40.0
dtype: float64


--------------------------------------------------------
Standard Deviation:
age                   14.104118
fnlwgt             106577.604741
educational-num         2.437673
capital-gain          936.753678
capital-loss          313.320005
hours-per-week         12.356849
dtype: float64


25% Percentile:
age                   25.0
fnlwgt             117210.0
educational-num         9.0
capital-gain            0.0
capital-loss            0.0
hours-per-week         35.0
Name: 0.25, dtype: float64


75% Percentile:
age                   46.0
fnlwgt             238917.0
educational-num        10.0
capital-gain            0.0
capital-loss            0.0
hours-per-week         40.0
Name: 0.75, dtype: float64


Minimum:
age                   17
fnlwgt             12285
educational-num        1
capital-gain           0
capital-loss           0
hours-per-week         1
dtype: int64


Maximum:
age                   90
fnlwgt             1490400
educational-num         16
capital-gain         41310
capital-loss          4356
hours-per-week          99
dtype: int64


Statistics for Income -  >50K
--------------------------------------------------------
Mean:
age                   44.275178
fnlwgt             188470.574570
educational-num        11.602807
```

```
capital-gain         4042.239497
capital-loss          193.528964
hours-per-week         45.452896
dtype: float64


------------------------------------------------------
Median:
age                    43.0
fnlwgt             176729.0
educational-num        12.0
capital-gain            0.0
capital-loss            0.0
hours-per-week         40.0
dtype: float64


------------------------------------------------------
Standard Deviation:
age                    10.558983
fnlwgt             102442.731958
educational-num         2.382624
capital-gain        14756.771034
capital-loss          593.211612
hours-per-week         11.091176
dtype: float64


25% Percentile:
age                    36.0
fnlwgt             118942.5
educational-num        10.0
capital-gain            0.0
capital-loss            0.0
hours-per-week         40.0
Name: 0.25, dtype: float64


75% Percentile:
age                    51.0
fnlwgt             233505.0
educational-num        13.0
capital-gain            0.0
capital-loss            0.0
hours-per-week         50.0
Name: 0.75, dtype: float64


Minimum:
age                   19
fnlwgt             13769
educational-num        1
capital-gain           0
capital-loss           0
hours-per-week         1
dtype: int64


Maximum:
age                    90
fnlwgt             1226583
educational-num         16
capital-gain         99999
capital-loss          3683
hours-per-week          99
dtype: int64
```

In [47]:
```
def calculate_mean(data):
    if len(data)==0:
        return 0
    m = sum(data)/len(data)
    return m

def calculate_std(data,mean):
    if len(data)<=1:
        return 0
    difference_squared = sum((x-mean)**2 for x in data)
    ans = (difference_squared/(len(data)-1))**0.5
    return ans

def calculate_percentile(data,percentile):
    sorted_data = sorted(data)
    index = int(percentile*len(data))
    percentile_result = sorted_data[index]
    return percentile_result

def display_stats(income_data,income_class):
    column = ["age","fnlwgt","educational-num","capital-gain","capital-loss","hours-per-
    print(f"\n***************Statistics for {income_class}*******************")

    # Mean
    mean_values = [calculate_mean(income_data[col]) for col in column]
    print("Mean: ")
    print(pd.Series(mean_values, index=column))

    # Standard Deviation
    std_values = [calculate_std(income_data[col],mean_values[i]) for i, col in enumerate
    print("\nStandard Deviation")
    print(pd.Series(std_values, index=column))

    # Percentile
    percentiles = [0.25, 0.75]
    for percentile_value in percentiles:
        percentile_values = [calculate_percentile(income_data[col], percentile_value) fo
        print(f"\n{int(percentile_value * 100)}th Percentile : ")
        print(pd.Series(percentile_values, index=column))
```

In [49]:
```
display_stats(income_less_than_50, '<= 50K')
display_stats(income_less_than_50, '>50K')
```

```
***************Statistics for <= 50K*******************
Mean:
age                   36.872184
fnlwgt             190039.565523
educational-num        9.598493
capital-gain         147.010308
capital-loss          54.151931
hours-per-week        38.840048
dtype: float64

Standard Deviation
age                   14.104118
fnlwgt             106577.604741
educational-num        2.437673
capital-gain         936.753678
capital-loss         313.320005
hours-per-week        12.356849
dtype: float64

25th Percentile :
age                   25
```

```
fnlwgt             117210
educational-num         9
capital-gain            0
capital-loss            0
hours-per-week         35
dtype: int64

75th Percentile :
age                    46
fnlwgt             238917
educational-num        10
capital-gain            0
capital-loss            0
hours-per-week         40
dtype: int64

***************Statistics for >50K*********************
Mean:
age                    36.872184
fnlwgt             190039.565523
educational-num         9.598493
capital-gain          147.010308
capital-loss           54.151931
hours-per-week         38.840048
dtype: float64

Standard Deviation
age                    14.104118
fnlwgt             106577.604741
educational-num         2.437673
capital-gain          936.753678
capital-loss          313.320005
hours-per-week         12.356849
dtype: float64

25th Percentile :
age                    25
fnlwgt             117210
educational-num         9
capital-gain            0
capital-loss            0
hours-per-week         35
dtype: int64

75th Percentile :
age                    46
fnlwgt             238917
educational-num        10
capital-gain            0
capital-loss            0
hours-per-week         40
dtype: int64
```

In [ ]:

In [ ]: