Problem 1

In [3]:
```
import numpy as np
import pandas as pd
```

In [28]:
```
csv_data = pd.read_csv('train.csv')
csv_data.head()
```

Out[28]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilitie
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPu
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPu
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPu
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPu
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPu

5 rows × 81 columns

In [17]:
```
print("Data type : ", type(csv_data))
print("Data dims : ", csv_data.shape)
```

```
Data type :  <class 'pandas.core.frame.DataFrame'>
Data dims :  (1460, 81)
```

In [29]:
```
print(csv_data.dtypes)
```

```
Id                int64
MSSubClass        int64
MSZoning         object
LotFrontage     float64
LotArea           int64
                 ...
MoSold            int64
YrSold            int64
SaleType         object
SaleCondition    object
SalePrice         int64
Length: 81, dtype: object
```

.info() method prints out a summary of the dataset

In [33]:
```
print(csv_data.info)
```

```
<bound method DataFrame.info of         Id  MSSubClass MSZoning  LotFrontage  LotAre
a Street Alley LotShape  \
0        1          60       RL         65.0     8450   Pave   NaN      Reg
1        2          20       RL         80.0     9600   Pave   NaN      Reg
2        3          60       RL         68.0    11250   Pave   NaN      IR1
3        4          70       RL         60.0     9550   Pave   NaN      IR1
4        5          60       RL         84.0    14260   Pave   NaN      IR1
...    ...         ...      ...          ...      ...    ...   ...      ...
1455  1456          60       RL         62.0     7917   Pave   NaN      Reg
```

```
1456  1457          20       RL         85.0    13175   Pave   NaN      Reg
1457  1458          70       RL         66.0     9042   Pave   NaN      Reg
1458  1459          20       RL         68.0     9717   Pave   NaN      Reg
1459  1460          20       RL         75.0     9937   Pave   NaN      Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature MiscVal  \
0            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
1            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
2            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
3            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
4            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
...          ...       ...  ...      ...    ...    ...         ...     ...
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN       0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN       0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed    2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN       0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN       0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
0         2   2008        WD         Normal     208500
1         5   2007        WD         Normal     181500
2         9   2008        WD         Normal     223500
3         2   2006        WD        Abnorml     140000
4        12   2008        WD         Normal     250000
...     ...    ...       ...            ...        ...
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[1460 rows x 81 columns]>
```

In [31]:

```
csv_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
 12  Neighborhood   1460 non-null   object
 13  Condition1     1460 non-null   object
 14  Condition2     1460 non-null   object
 15  BldgType       1460 non-null   object
 16  HouseStyle     1460 non-null   object
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
```

```
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrType     1452 non-null   object
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object
 28  ExterCond      1460 non-null   object
 29  Foundation     1460 non-null   object
 30  BsmtQual       1423 non-null   object
 31  BsmtCond       1423 non-null   object
 32  BsmtExposure   1422 non-null   object
 33  BsmtFinType1   1423 non-null   object
 34  BsmtFinSF1     1460 non-null   int64
 35  BsmtFinType2   1422 non-null   object
 36  BsmtFinSF2     1460 non-null   int64
 37  BsmtUnfSF      1460 non-null   int64
 38  TotalBsmtSF    1460 non-null   int64
 39  Heating        1460 non-null   object
 40  HeatingQC      1460 non-null   object
 41  CentralAir     1460 non-null   object
 42  Electrical     1459 non-null   object
 43  1stFlrSF       1460 non-null   int64
 44  2ndFlrSF       1460 non-null   int64
 45  LowQualFinSF   1460 non-null   int64
 46  GrLivArea      1460 non-null   int64
 47  BsmtFullBath   1460 non-null   int64
 48  BsmtHalfBath   1460 non-null   int64
 49  FullBath       1460 non-null   int64
 50  HalfBath       1460 non-null   int64
 51  BedroomAbvGr   1460 non-null   int64
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

.describe() gives a description of the statistic of the dataset

In [32]:

```
csv_data.describe()
```

Out[32]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuil
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.00000(
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.26780{
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.20290⁴
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.00000(
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.00000(
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.00000(
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.00000(
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.00000(

8 rows × 38 columns

Problem 2

In [11]:
```
html_data = pd.read_html('https://en.wikipedia.org/wiki/2016_Summer_Olympics_medal_t
```

In [12]:
```
print("Data type : ", type(html_data))
print("HTML tables : ", len(html_data))
```

```
Data type :  <class 'list'>
HTML tables :  7
```

Main medal table

In [73]:
```
html_data[2]
```

Out[73]:

	Rank	NOC	Gold	Silver	Bronze	Total
0	1	United States	46	37	38	121
1	2	Great Britain	27	23	17	67
2	3	China	26	18	26	70
3	4	Russia	19	17	20	56
4	5	Germany	17	10	15	42
...	...	...	...	...	...	...
82	78	Nigeria	0	0	1	1
83	78	Portugal	0	0	1	1
84	78	Trinidad and Tobago	0	0	1	1
85	78	United Arab Emirates	0	0	1	1
86	Totals (86 NOCs)	Totals (86 NOCs)	307	307	359	973

87 rows × 6 columns

In [79]:
```
maintable = pd.DataFrame(html_data[2])
print("Data type : ", type(maintable))
print("Data dims : ", maintable.size)
maintable.head()
```

```
Data type :  <class 'pandas.core.frame.DataFrame'>
Data dims :  522
```

Out[79]:

	Rank	NOC	Gold	Silver	Bronze	Total
0	1	United States	46	37	38	121
1	2	Great Britain	27	23	17	67
2	3	China	26	18	26	70
3	4	Russia	19	17	20	56
4	5	Germany	17	10	15	42

In [80]:
```
top20 = pd.DataFrame(html_data[2].head(20))
print("Data type : ", type(top20))
print("Data dims : ", top20.size)
top20.head(20)
```

```
Data type :  <class 'pandas.core.frame.DataFrame'>
Data dims :  120
```

Out[80]:

	Rank	NOC	Gold	Silver	Bronze	Total
0	1	United States	46	37	38	121
1	2	Great Britain	27	23	17	67
2	3	China	26	18	26	70
3	4	Russia	19	17	20	56
4	5	Germany	17	10	15	42
5	6	Japan	12	8	21	41
6	7	France	10	18	14	42
7	8	South Korea	9	3	9	21
8	9	Italy	8	12	8	28
9	10	Australia	8	11	10	29
10	11	Netherlands	8	7	4	19
11	12	Hungary	8	3	4	15
12	13	Brazil*	7	6	6	19
13	14	Spain	7	4	6	17
14	15	Kenya	6	6	1	13
15	16	Jamaica	6	3	2	11
16	17	Croatia	5	3	2	10
17	18	Cuba	5	2	4	11
18	19	New Zealand	4	9	5	18
19	20	Canada	4	3	15	22

In [94]:
```
top20.describe()
```

Out[94]:

	Gold	Silver	Bronze	Total
count	20.000000	20.000000	20.00000	20.000000
mean	12.100000	10.150000	11.35000	33.600000
std	10.452398	8.797577	9.51052	27.545942
min	4.000000	2.000000	1.00000	10.000000
25%	6.000000	3.000000	4.00000	16.500000
50%	8.000000	7.500000	8.50000	21.500000
75%	13.250000	13.250000	15.50000	42.000000
max	46.000000	37.000000	38.00000	121.000000

Bonus Problem A

In [87]:
```
adultdata = pd.read_table('adult.data', sep = ",", header = 0)
```

In [88]:
```
adultdata.head()
```

Out[88]:

	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40
0	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13
1	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40
2	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40
3	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40
4	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40

In [89]:
```
adultdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32560 entries, 0 to 32559
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   39              32560 non-null  int64 
 1    State-gov      32560 non-null  object
```

```
 2    77516          32560 non-null  int64
 3    Bachelors      32560 non-null  object
 4    13             32560 non-null  int64
 5    Never-married  32560 non-null  object
 6    Adm-clerical   32560 non-null  object
 7    Not-in-family  32560 non-null  object
 8    White          32560 non-null  object
 9    Male           32560 non-null  object
 10   2174           32560 non-null  int64
 11   0              32560 non-null  int64
 12   40             32560 non-null  int64
 13   United-States  32560 non-null  object
 14   <=50K          32560 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

In [90]:
```
print("Data type : ", type(adultdata))
print("Data dims : ", adultdata.shape)
```

```
Data type :  <class 'pandas.core.frame.DataFrame'>
Data dims :  (32560, 15)
```

In [91]:
```
adultdata.describe()
```

Out[91]:

	39	77516	13	2174	0	40
count	32560.000000	3.256000e+04	32560.000000	32560.000000	32560.000000	32560.000000
mean	38.581634	1.897818e+05	10.080590	1077.615172	87.306511	40.437469
std	13.640642	1.055498e+05	2.572709	7385.402999	402.966116	12.347618
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178315e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783630e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370545e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

Bonus Problem B

In [4]:
```
years = ['2000','2004','2008','2012','2016']
d = {}
top = {}
for i in years:
    html_data = pd.read_html('https://en.wikipedia.org/wiki/'+i+'_Summer_Olympics_me
    d[i]= html_data[2]
    top[i]= html_data[2].head(20)
```

In [7]:
```
print(2016, d['2016'].head(30))
```

```
2016    Rank            NOC  Gold  Silver  Bronze  Total
0     1  United States    46      37      38    121
1     2  Great Britain    27      23      17     67
2     3          China    26      18      26     70
3     4         Russia    19      17      20     56
4     5        Germany    17      10      15     42
```

5	6	Japan	12	8	21	41
6	7	France	10	18	14	42
7	8	South Korea	9	3	9	21
8	9	Italy	8	12	8	28
9	10	Australia	8	11	10	29
10	11	Netherlands	8	7	4	19
11	12	Hungary	8	3	4	15
12	13	Brazil*	7	6	6	19
13	14	Spain	7	4	6	17
14	15	Kenya	6	6	1	13
15	16	Jamaica	6	3	2	11
16	17	Croatia	5	3	2	10
17	18	Cuba	5	2	4	11
18	19	New Zealand	4	9	5	18
19	20	Canada	4	3	15	22
20	21	Uzbekistan	4	2	7	13
21	22	Kazakhstan	3	5	10	18
22	23	Colombia	3	2	3	8
23	24	Switzerland	3	2	2	7
24	25	Iran	3	1	4	8
25	26	Greece	3	1	2	6
26	27	Argentina	3	1	0	4
27	28	Denmark	2	6	7	15
28	29	Sweden	2	6	3	11
29	30	South Africa	2	6	2	10

In [102…

```
print(2016, top['2016'])
```

2016	Rank		NOC	Gold	Silver	Bronze	Total
0	1	United States	46	37	38	121	
1	2	Great Britain	27	23	17	67	
2	3	China	26	18	26	70	
3	4	Russia	19	17	20	56	
4	5	Germany	17	10	15	42	
5	6	Japan	12	8	21	41	
6	7	France	10	18	14	42	
7	8	South Korea	9	3	9	21	
8	9	Italy	8	12	8	28	
9	10	Australia	8	11	10	29	
10	11	Netherlands	8	7	4	19	
11	12	Hungary	8	3	4	15	
12	13	Brazil*	7	6	6	19	
13	14	Spain	7	4	6	17	
14	15	Kenya	6	6	1	13	
15	16	Jamaica	6	3	2	11	
16	17	Croatia	5	3	2	10	
17	18	Cuba	5	2	4	11	
18	19	New Zealand	4	9	5	18	
19	20	Canada	4	3	15	22	

In [ ]: