DATA WAREHOUSE AND DATA MINING

LAB DIGITAL ASSIGNMENT-2

NAME:HRITHIK HEM SUNDAR.B

REGNO:19MID0021

Create Dataset:

1) Create a csv file with at least 20 entries on the following attributes Name ◊ Nominal Age ◊ 1…100 Experience ◊ 1…15

2) Remove values of Experience and Fill with “NA”.

3) Replace few entries of Age with >100 values.

Dataset:

	Empid	Name	Designation	Salary	Experience	Age
1	1	abc	Manager	20000	7	25
2	2	def	Supervisior	19000	NA	34
3	3	ghi	Clerk	10000	6	100
4	4	jkl	Labour	2000	2	56
5	5	mno	Supervisior	18000	6	34
6	6	pqr	Manager	25000	NA	23
7	7	stu	Supervisior	18000	10	56
8	8	vwx	Manager	20000	7	100
9	9	yza	Clerk	15000	5	43
10	10	bcd	Clerk	15000	5	34
11	11	efg	Manager	23000	NA	23
12	12	hij	Clerk	12000	4	100
13	13	klm	Labour	4000	4	35
14	14	nop	Supervisior	20000	NA	56
15	15	qrs	Manager	20000	7	43
16	16	tuv	Labour	2000	2	56
17	17	wxy	Clerk	12000	4	101
18	18	zab	Manager	20000	7	32
19	19	cde	Labour	2000	1	56
20	20	fgh	Supervisior	21000	NA	78
21	21	ijk	Manager	22000	8	54
22	22	lmn	Labour	2000	2	32
23	23	opq	Manager	23000	10	102
24	24	rst	Supervisior	20000	7	32
25	25	uvw	Labour	2000	2	45
26	26	xyz	Clerk	17000	NA	37
27	27	zyx	Labour	3000	3	56
28	28	wvu	Supervisior	15000	5	25
29	29	tsr	Labour	3000	NA	54
30	30	qpo	Clerk	10000	3	43

4) Print the dataset using R

```
> df=read.csv("E:/dataset.csv")
> print(df)
   Empid Name Designation Salary Experience Age
1      1  abc     Manager  20000          7  25
2      2  def Supervisior  19000         NA  34
3      3  ghi       Clerk  10000          6 100
4      4  jkl      Labour   2000          2  56
5      5  mno Supervisior  18000          6  34
6      6  pqr     Manager  25000         NA  23
7      7  stu Supervisior  18000         10  56
8      8  vwx     Manager  20000          7 100
9      9  yza       Clerk  15000          5  43
10    10  bcd       Clerk  15000          5  34
11    11  efg     Manager  23000         NA  23
12    12  hij       Clerk  12000          4 100
13    13  klm      Labour   4000          4  35
14    14  nop Supervisior  20000         NA  56
15    15  qrs     Manager  20000          7  43
16    16  tuv      Labour   2000          2  56
17    17  wxy       Clerk  12000          4 101
18    18  zab     Manager  20000          7  32
19    19  cde      Labour   2000          1  56
20    20  fgh Supervisior  21000         NA  78
21    21  ijk     Manager  22000          8  54
22    22  lmn      Labour   2000          2  32
23    23  opq     Manager  23000         10 102
24    24  rst Supervisior  20000          7  32
25    25  uvw      Labour   2000          2  45
26    26  xyz       Clerk  17000         NA  37
27    27  zyx      Labour   3000          3  56
28    28  wvu Supervisior  15000          5  25
29    29  tsr      Labour   3000         NA  54
30    30  qpo       Clerk  10000          3  43
```

Write R code for the following: Missing data

a) **Display the content of the dataset by removing the missing value tuples.**

```
> na.omit(df)
   Empid Name Designation Salary Experience Age
1      1  abc     Manager  20000          7  25
3      3  ghi       Clerk  10000          6 100
4      4  jkl      Labour   2000          2  56
5      5  mno Supervisior  18000          6  34
7      7  stu Supervisior  18000         10  56
8      8  vwx     Manager  20000          7 100
9      9  yza       Clerk  15000          5  43
10    10  bcd       Clerk  15000          5  34
12    12  hij       Clerk  12000          4 100
13    13  klm      Labour   4000          4  35
15    15  qrs     Manager  20000          7  43
16    16  tuv      Labour   2000          2  56
17    17  wxy       Clerk  12000          4 101
18    18  zab     Manager  20000          7  32
19    19  cde      Labour   2000          1  56
21    21  ijk     Manager  22000          8  54
22    22  lmn      Labour   2000          2  32
23    23  opq     Manager  23000         10 102
24    24  rst Supervisior  20000          7  32
25    25  uvw      Labour   2000          2  45
27    27  zyx      Labour   3000          3  56
28    28  wvu Supervisior  15000          5  25
30    30  qpo       Clerk  10000          3  43
```

b) Calculate the mean of the attribute without considering the missing values.

```
> mean(df[,5], na.rm = TRUE)
[1] 5.086957
> mean(df[,c("Experience")], na.rm = TRUE)
[1] 5.086957
```

c) Display the content of the dataset by replacing the „NA" with mean of the attribute.

```
> experience=na.omit(df[,5])
> exp_mean= mean(experience)
> df[is.na(df)] <-exp_mean
> df
   Empid Name Designation Salary Experience Age
1      1  abc     Manager  20000   7.000000  25
2      2  def Supervisior  19000   5.086957  34
3      3  ghi       Clerk  10000   6.000000 100
4      4  jkl      Labour   2000   2.000000  56
5      5  mno Supervisior  18000   6.000000  34
6      6  pqr     Manager  25000   5.086957  23
7      7  stu Supervisior  18000  10.000000  56
8      8  vwx     Manager  20000   7.000000 100
9      9  yza       Clerk  15000   5.000000  43
10    10  bcd       Clerk  15000   5.000000  34
11    11  efg     Manager  23000   5.086957  23
12    12  hij       Clerk  12000   4.000000 100
13    13  klm      Labour   4000   4.000000  35
14    14  nop Supervisior  20000   5.086957  56
15    15  qrs     Manager  20000   7.000000  43
16    16  tuv      Labour   2000   2.000000  56
17    17  wxy       Clerk  12000   4.000000 101
18    18  zab     Manager  20000   7.000000  32
19    19  cde      Labour   2000   1.000000  56
20    20  fgh Supervisior  21000   5.086957  78
21    21  ijk     Manager  22000   8.000000  54
22    22  lmn      Labour   2000   2.000000  32
23    23  opq     Manager  23000  10.000000 102
24    24  rst Supervisior  20000   7.000000  32
25    25  uvw      Labour   2000   2.000000  45
26    26  xyz       Clerk  17000   5.086957  37
27    27  zyx      Labour   3000   3.000000  56
28    28  wvu Supervisior  15000   5.000000  25
29    29  tsr      Labour   3000   5.086957  54
30    30  qpo       Clerk  10000   3.000000  43
```

Outlier detection
Method 1:
d) Replace the age value above 100 with “NA”.

```
> df$Age[df$Age>100] = NA
> df
   Empid Name Designation Salary Experience Age
1      1  abc     Manager  20000   7.000000  25
2      2  def Supervisior  19000   5.086957  34
3      3  ghi       Clerk  10000   6.000000 100
4      4  jkl      Labour   2000   2.000000  56
5      5  mno Supervisior  18000   6.000000  34
6      6  pqr     Manager  25000   5.086957  23
7      7  stu Supervisior  18000  10.000000  56
8      8  vwx     Manager  20000   7.000000 100
9      9  yza       Clerk  15000   5.000000  43
10    10  bcd       Clerk  15000   5.000000  34
11    11  efg     Manager  23000   5.086957  23
12    12  hij       Clerk  12000   4.000000 100
13    13  klm      Labour   4000   4.000000  35
14    14  nop Supervisior  20000   5.086957  56
15    15  qrs     Manager  20000   7.000000  43
16    16  tuv      Labour   2000   2.000000  56
17    17  wxy       Clerk  12000   4.000000  NA
18    18  zab     Manager  20000   7.000000  32
19    19  cde      Labour   2000   1.000000  56
20    20  fgh Supervisior  21000   5.086957  78
21    21  ijk     Manager  22000   8.000000  54
22    22  lmn      Labour   2000   2.000000  32
23    23  opq     Manager  23000  10.000000  NA
24    24  rst Supervisior  20000   7.000000  32
25    25  uvw      Labour   2000   2.000000  45
26    26  xyz       Clerk  17000   5.086957  37
27    27  zyx      Labour   3000   3.000000  56
28    28  wvu Supervisior  15000   5.000000  25
29    29  tsr      Labour   3000   5.086957  54
30    30  qpo       Clerk  10000   3.000000  43
```

d) Print the current dataset.

```
> print(df)
   Empid Name Designation Salary Experience Age
1      1  abc     Manager  20000   7.000000  25
2      2  def Supervisior  19000   5.086957  34
3      3  ghi       Clerk  10000   6.000000 100
4      4  jkl      Labour   2000   2.000000  56
5      5  mno Supervisior  18000   6.000000  34
6      6  pqr     Manager  25000   5.086957  23
7      7  stu Supervisior  18000  10.000000  56
8      8  vwx     Manager  20000   7.000000 100
9      9  yza       Clerk  15000   5.000000  43
10    10  bcd       Clerk  15000   5.000000  34
11    11  efg     Manager  23000   5.086957  23
12    12  hij       Clerk  12000   4.000000 100
13    13  klm      Labour   4000   4.000000  35
14    14  nop Supervisior  20000   5.086957  56
15    15  qrs     Manager  20000   7.000000  43
16    16  tuv      Labour   2000   2.000000  56
17    17  wxy       Clerk  12000   4.000000  NA
18    18  zab     Manager  20000   7.000000  32
19    19  cde      Labour   2000   1.000000  56
20    20  fgh Supervisior  21000   5.086957  78
21    21  ijk     Manager  22000   8.000000  54
22    22  lmn      Labour   2000   2.000000  32
23    23  opq     Manager  23000  10.000000  NA
24    24  rst Supervisior  20000   7.000000  32
25    25  uvw      Labour   2000   2.000000  45
26    26  xyz       Clerk  17000   5.086957  37
27    27  zyx      Labour   3000   3.000000  56
28    28  wvu Supervisior  15000   5.000000  25
29    29  tsr      Labour   3000   5.086957  54
30    30  qpo       Clerk  10000   3.000000  43
```

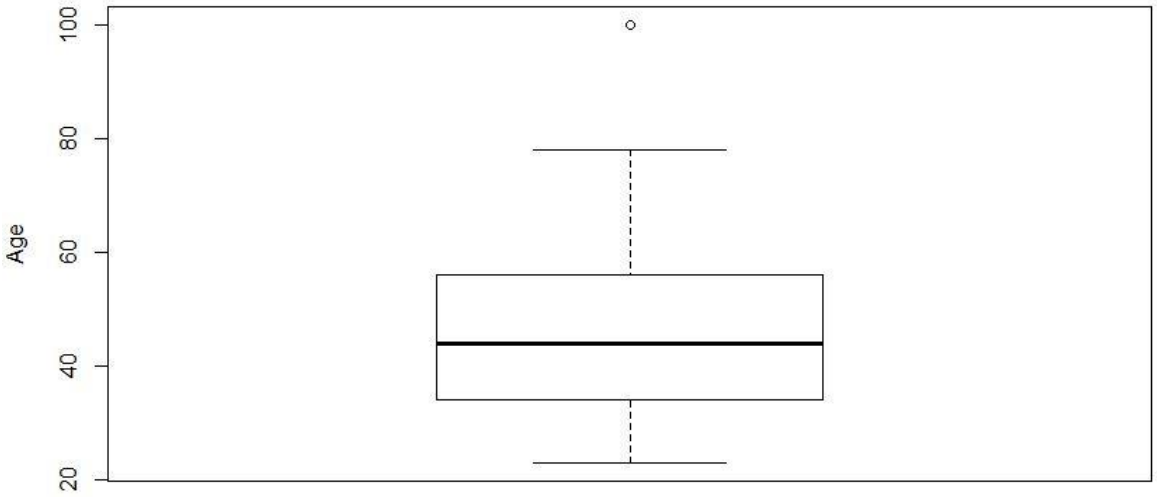
e) Replace the “NA” in age with Mean of age.

```
> means=mean(df[,c("Age")], na.rm = TRUE)
> means
[1] 48.64286
> df[is.na(df)] <-means
> df
   Empid Name Designation Salary Experience       Age
1      1  abc     Manager  20000   7.000000  25.00000
2      2  def Supervisior  19000   5.086957  34.00000
3      3  ghi       Clerk  10000   6.000000 100.00000
4      4  jkl      Labour   2000   2.000000  56.00000
5      5  mno Supervisior  18000   6.000000  34.00000
6      6  pqr     Manager  25000   5.086957  23.00000
7      7  stu Supervisior  18000  10.000000  56.00000
8      8  vwx     Manager  20000   7.000000 100.00000
9      9  yza       Clerk  15000   5.000000  43.00000
10    10  bcd       Clerk  15000   5.000000  34.00000
11    11  efg     Manager  23000   5.086957  23.00000
12    12  hij       Clerk  12000   4.000000 100.00000
13    13  klm      Labour   4000   4.000000  35.00000
14    14  nop Supervisior  20000   5.086957  56.00000
15    15  qrs     Manager  20000   7.000000  43.00000
16    16  tuv      Labour   2000   2.000000  56.00000
17    17  wxy       Clerk  12000   4.000000  48.64286
18    18  zab     Manager  20000   7.000000  32.00000
19    19  cde      Labour   2000   1.000000  56.00000
20    20  fgh Supervisior  21000   5.086957  78.00000
21    21  ijk     Manager  22000   8.000000  54.00000
22    22  lmn      Labour   2000   2.000000  32.00000
23    23  opq     Manager  23000  10.000000  48.64286
24    24  rst Supervisior  20000   7.000000  32.00000
25    25  uvw      Labour   2000   2.000000  45.00000
26    26  xyz       Clerk  17000   5.086957  37.00000
27    27  zyx      Labour   3000   3.000000  56.00000
28    28  wvu Supervisior  15000   5.000000  25.00000
29    29  tsr      Labour   3000   5.086957  54.00000
30    30  qpo       Clerk  10000   3.000000  43.00000
```

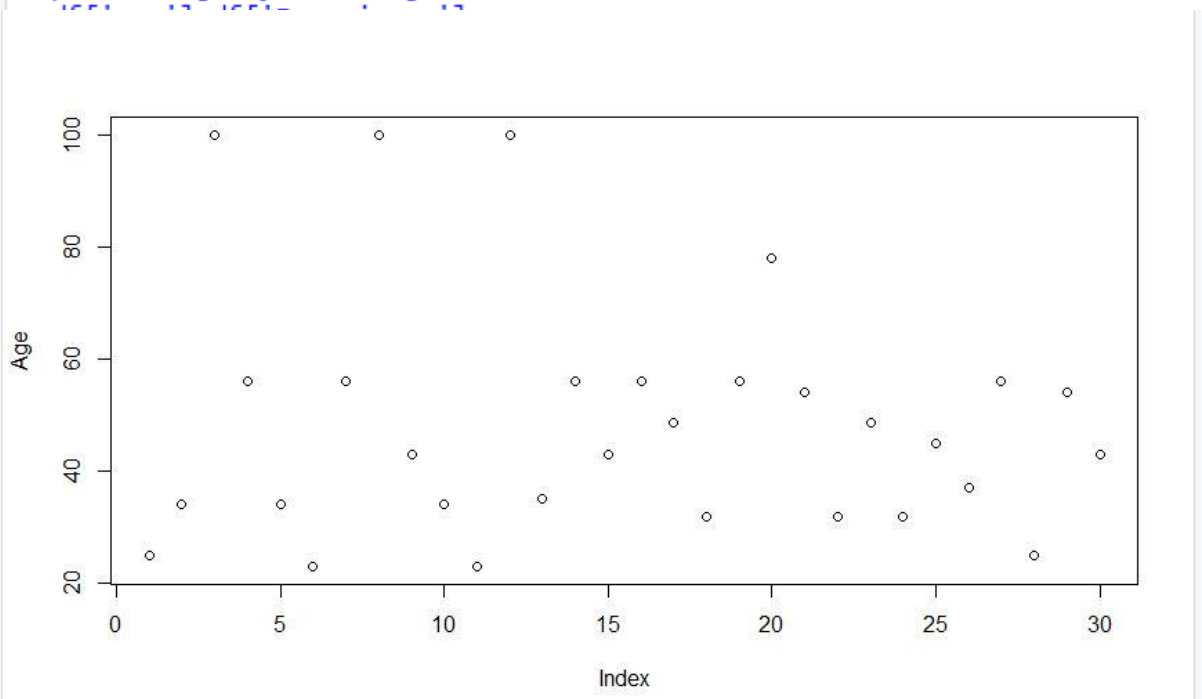
Method 2:

Use the same dataset and change the value of age to lie outside the range of 100. Apply box plot a nd scatter plot to find the outliers.

```
> boxplot(df$Age, ylab = "Age")
```




```
> plot(df$Age, ylab = "Age")
```



Redundancy check

Repeat the experience column as new attribute Exp. Find the correlation between age, Exp & experience.

```
> df['exp']=df['Experience']
> cor(df[, c('exp','Experience', 'Age')])
                  exp  Experience         Age
exp        1.000000000 1.000000000 0.007320676
Experience 1.000000000 1.000000000 0.007320676
Age        0.007320676 0.007320676 1.000000000
>
```

ASSOCIATION ANALYSIS IN PYTHON

In [3]:
```
import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori, association_rules
import matplotlib.pyplot as plt
```

In [6]:
```
df = pd.read_csv("E:/retail_dataset.csv", sep=',')
df.head(10)
```

Out[6]:

	0	1	2	3	4	5	6
0	Bread	Wine	Eggs	Meat	Cheese	Pencil	Diaper
1	Bread	Cheese	Meat	Diaper	Wine	Milk	Pencil
2	Cheese	Meat	Eggs	Milk	Wine	NaN	NaN
3	Cheese	Meat	Eggs	Milk	Wine	NaN	NaN
4	Meat	Pencil	Wine	NaN	NaN	NaN	NaN
5	Eggs	Bread	Wine	Pencil	Milk	Diaper	Bagel
6	Wine	Pencil	Eggs	Cheese	NaN	NaN	NaN
7	Bagel	Bread	Milk	Pencil	Diaper	NaN	NaN
8	Bread	Diaper	Cheese	Milk	Wine	Eggs	NaN
9	Bagel	Wine	Diaper	Meat	Pencil	Eggs	Cheese

In [40]:
```
items=set(x for x in items if pd.isnull(x)==False)
print(items)
```

```
{'Diaper', 'Wine', 'Bread', 'Eggs', 'Pencil', 'Cheese', 'Bagel', 'Meat', 'Milk'}
```

In [44]:
```
itemset = set(items)
encoded_vals = []
for index, row in df.iterrows():
    rowset = set(row)
    labels = {}
    uncommons = list(itemset - rowset)
    commons = list(itemset.intersection(rowset))
    for uc in uncommons:
        labels[uc] = 0
    for com in commons:
        labels[com] = 1
    encoded_vals.append(labels)
encoded_vals[0]
ohe_df = pd.DataFrame(encoded_vals)
```

In [45]:
```
freq_items = apriori(ohe_df, min_support=0.2, use_colnames=True, verbose=1)
freq_items.head(7)
```

```
Processing 147 combinations | Sampling itemset size 3
```

Out[45]:

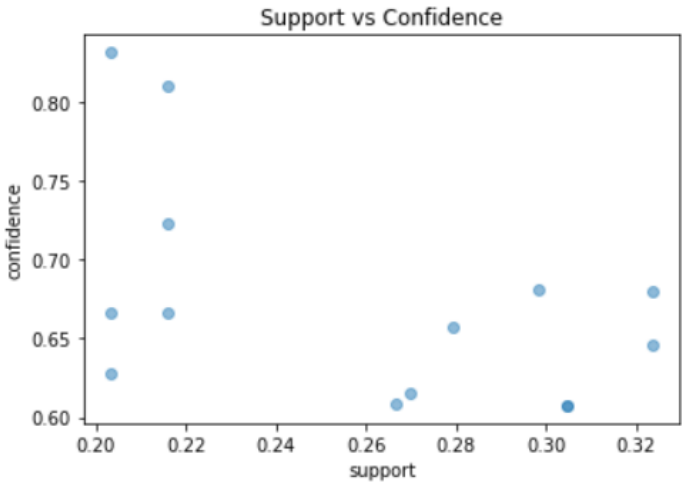
	support	itemsets
0	0.425397	(Bagel)
1	0.501587	(Milk)
2	0.406349	(Diaper)
3	0.438095	(Wine)
4	0.504762	(Bread)
5	0.438095	(Eggs)
6	0.361905	(Pencil)

In [46]:
```
rules = association_rules(freq_items, metric="confidence", min_threshold=0.6)
rules.head()
```

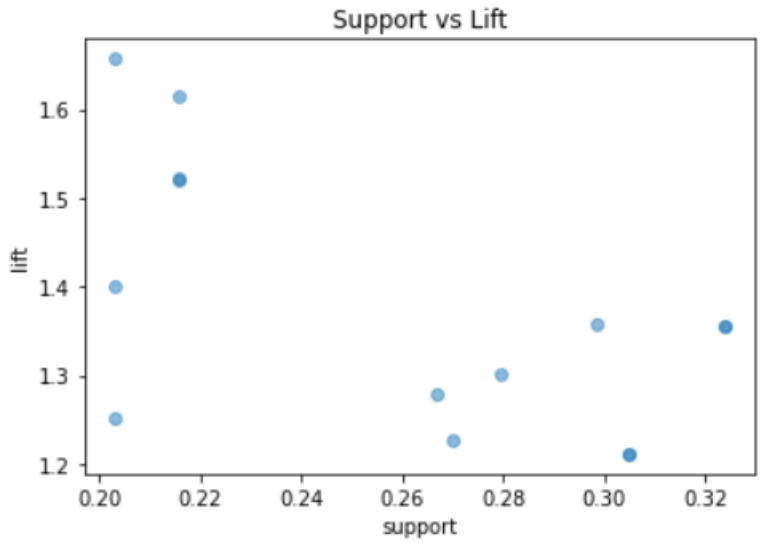
Out[46]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Bagel)	(Bread)	0.425397	0.504762	0.279365	0.656716	1.301042	0.064641	1.442650
1	(Cheese)	(Milk)	0.501587	0.501587	0.304762	0.607595	1.211344	0.053172	1.270148
2	(Milk)	(Cheese)	0.501587	0.501587	0.304762	0.607595	1.211344	0.053172	1.270148
3	(Wine)	(Cheese)	0.438095	0.501587	0.269841	0.615942	1.227986	0.050098	1.297754
4	(Eggs)	(Cheese)	0.438095	0.501587	0.298413	0.681159	1.358008	0.078670	1.563203

In [47]:
```
plt.scatter(rules['support'], rules['confidence'], alpha=0.5)
plt.xlabel('support')
plt.ylabel('confidence')
plt.title('Support vs Confidence')
plt.show()
```



In [49]:
```
plt.scatter(rules['support'], rules['lift'], alpha=0.5)
plt.xlabel('support')
plt.ylabel('lift')
plt.title('Support vs Lift')
plt.show()
```



In [51]:
```
fit = np.polyfit(rules['lift'], rules['confidence'], 1)
fit_fn = np.poly1d(fit)
plt.plot(rules['lift'], rules['confidence'], 'yo', rules['lift'],
 fit_fn(rules['lift']))
```

Out[51]:
```
[<matplotlib.lines.Line2D at 0x17d974ae370>,
 <matplotlib.lines.Line2D at 0x17d974ae310>]
```