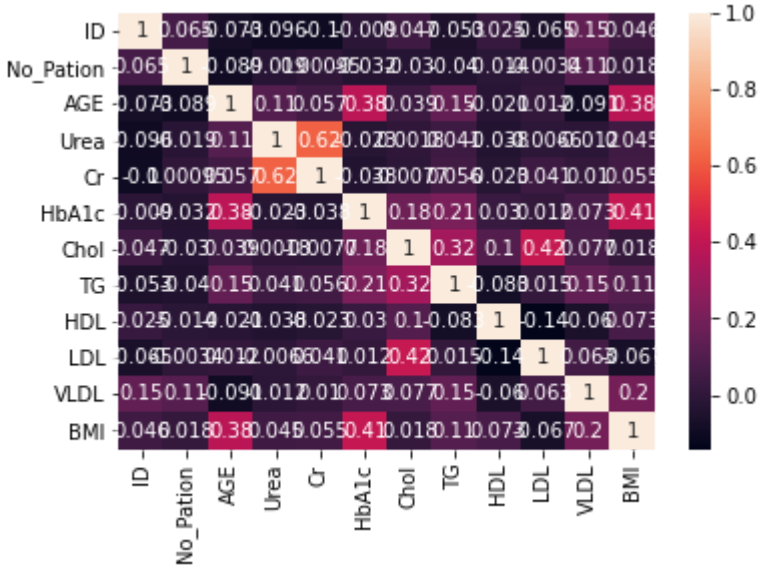
In [85]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('Datasets/diabetes_unclean.csv')
df.info()
```

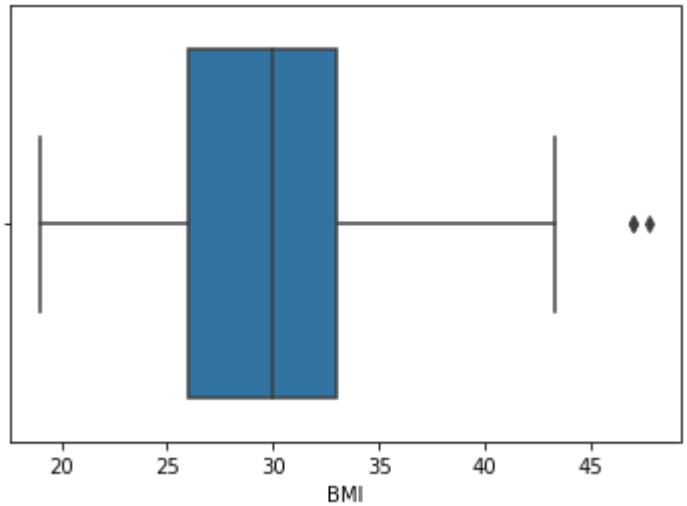
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1009 entries, 0 to 1008
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   ID         1009 non-null   int64  
 1   No_Pation  1009 non-null   int64  
 2   Gender     1009 non-null   object 
 3   AGE        1008 non-null   float64
 4   Urea       1008 non-null   float64
 5   Cr         1007 non-null   float64
 6   HbA1c      1006 non-null   float64
 7   Chol       1007 non-null   float64
 8   TG         1007 non-null   float64
 9   HDL        1008 non-null   float64
 10  LDL        1007 non-null   float64
 11  VLDL       1008 non-null   float64
 12  BMI        1009 non-null   float64
 13  CLASS      1009 non-null   object 
dtypes: float64(10), int64(2), object(2)
memory usage: 110.5+ KB
```

In [3]:
```
df=df.dropna()
```

In [4]:
```
sns.heatmap(data=df.corr(),annot=True)
plt.show()
```

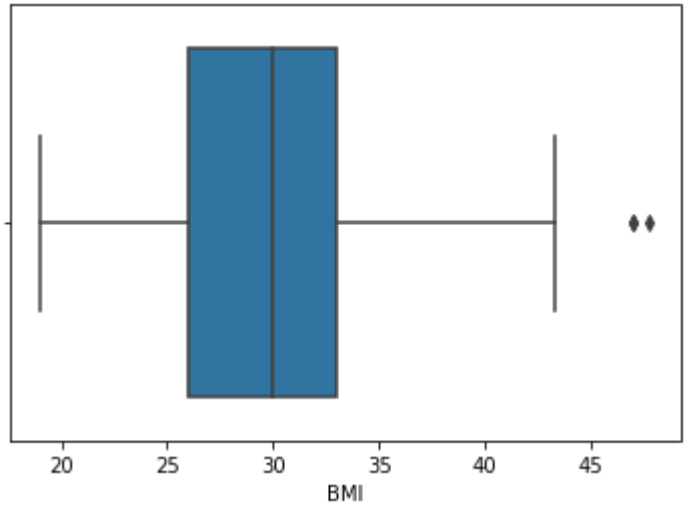


In [5]:
```
sns.boxplot(x=df.BMI)
plt.show()
```



In [6]:
```
#remove outlier in Cr column
q1=df['Cr'].quantile(0.25)
q3=df['Cr'].quantile(0.75)
iqr=q3-q1
ul=q3+(1.5)*iqr
ll=q1-(1.5)*iqr
df=df[(df['Cr']>=ll) & (df['Cr']<=ul)]
```

In [7]:
```
sns.boxplot(x=df.BMI)
plt.show()
```



In [9]:
```
df=pd.read_csv('Datasets/supermarket_sales.csv')
df.info()
df
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Invoice ID               1000 non-null   object 
 1   Branch                   1000 non-null   object 
 2   City                     1000 non-null   object 
 3   Customer type            1000 non-null   object 
 4   Gender                   1000 non-null   object 
 5   Product line             1000 non-null   object 
 6   Unit price               1000 non-null   float64
 7   Quantity                 1000 non-null   int64  
 8   Tax 5%                   1000 non-null   float64
 9   Total                    1000 non-null   float64
 10  Date                     1000 non-null   object 
 11  Time                     1000 non-null   object 
 12  Payment                  1000 non-null   object 
 13  cogs                     996 non-null    float64
 14  gross margin percentage  1000 non-null   float64
 15  gross income             1000 non-null   float64
 16  Rating                   995 non-null    float64
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB
```

Out[9]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	01-05-2019	13:08	Ewallet	522.83	4.761905
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	03-08-2019	10:29	Cash	76.40	4.761905
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	03-03-2019	13:23	Credit card	324.31	4.761905
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	20:33	Ewallet	465.76	4.761905
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	02-08-2019	10:37	Ewallet	604.17	4.761905
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1	2.0175	42.3675	1/29/2019	13:46	Ewallet	40.35	4.761905
996	303-96-2227	B	Mandalay	Normal	Female	Home and lifestyle	97.38	10	48.6900	1022.4900	03-02-2019	17:16	Ewallet	973.80	4.761905
997	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1	1.5920	33.4320	02-09-2019	13:22	Cash	31.84	4.761905
998	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1	3.2910	69.1110	2/22/2019	15:33	Cash	65.82	4.761905
999	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7	30.9190	649.2990	2/18/2019	13:28	Cash	618.38	4.761905

1000 rows × 17 columns

In [10]:
```
1 df.head(5)
```

Out[10]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	g inc
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	01-05-2019	13:08	Ewallet	522.83	4.761905	26.
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	03-08-2019	10:29	Cash	76.40	4.761905	3.
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	03-03-2019	13:23	Credit card	324.31	4.761905	16.
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	20:33	Ewallet	465.76	4.761905	23.
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	02-08-2019	10:37	Ewallet	604.17	4.761905	30.

In [11]:
```
1 df.info
```

Out[11]:
```
<bound method DataFrame.info of       Invoice ID Branch       City Customer type  Gender  \
0    750-67-8428      A     Yangon        Member  Female
1    226-31-3081      C  Naypyitaw        Normal  Female
2    631-41-3108      A     Yangon        Normal    Male
3    123-19-1176      A     Yangon        Member    Male
4    373-73-7910      A     Yangon        Normal    Male
..           ...    ...        ...           ...     ...
995  233-67-5758      C  Naypyitaw        Normal    Male
996  303-96-2227      B   Mandalay        Normal  Female
997  727-02-1313      A     Yangon        Member    Male
998  347-56-2442      A     Yangon        Normal    Male
999  849-09-3807      A     Yangon        Member  Female

               Product line  Unit price  Quantity   Tax 5%      Total  \
0         Health and beauty       74.69         7  26.1415   548.9715
1    Electronic accessories       15.28         5   3.8200    80.2200
2        Home and lifestyle       46.33         7  16.2155   340.5255
3         Health and beauty       58.22         8  23.2880   489.0480
4         Sports and travel       86.31         7  30.2085   634.3785
..                      ...         ...       ...      ...        ...
995       Health and beauty       40.35         1   2.0175    42.3675
996       Home and lifestyle       97.38        10  48.6900  1022.4900
997       Food and beverages       31.84         1   1.5920    33.4320
998       Home and lifestyle       65.82         1   3.2910    69.1110
999     Fashion accessories       88.34         7  30.9190   649.2990

           Date   Time      Payment    cogs  gross margin percentage  \
0    01-05-2019  13:08      Ewallet  522.83                 4.761905
1    03-08-2019  10:29         Cash   76.40                 4.761905
2    03-03-2019  13:23  Credit card  324.31                 4.761905
3     1/27/2019  20:33      Ewallet  465.76                 4.761905
4    02-08-2019  10:37      Ewallet  604.17                 4.761905
..          ...    ...          ...     ...                      ...
995   1/29/2019  13:46      Ewallet   40.35                 4.761905
996  03-02-2019  17:16      Ewallet  973.80                 4.761905
997  02-09-2019  13:22         Cash   31.84                 4.761905
998   2/22/2019  15:33         Cash   65.82                 4.761905
999   2/18/2019  13:28         Cash  618.38                 4.761905

     gross income  Rating
0         26.1415     9.1
```

```
1          3.8200     9.6
2         16.2155     7.4
3         23.2880     8.4
4         30.2085     5.3
..            ...     ...
995        2.0175     6.2
996       48.6900     4.4
997        1.5920     7.7
998        3.2910     4.1
999       30.9190     6.6

[1000 rows x 17 columns]>
```

In [16]:
```
1 df.isna().sum()
```

Out[16]:
```
Invoice ID                0
Branch                    0
City                      0
Customer type             0
Gender                    0
Product line              0
Unit price                0
Quantity                  0
Tax 5%                    0
Total                     0
Date                      0
Time                      0
Payment                   0
cogs                      4
gross margin percentage   0
gross income              0
Rating                    5
dtype: int64
```

In [17]:
```
1 df.describe(exclude=np.number)
```

Out[17]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Date	Time	Payment
count	1000	1000	1000	1000	1000	1000	1000	1000	1000
unique	1000	3	3	2	2	6	89	506	3
top	840-19-2096	A	Yangon	Member	Female	Fashion accessories	02-07-2019	14:42	Ewallet
freq	1	340	340	501	501	178	20	7	345

In [18]:
```
1 df['cogs']=df.cogs.fillna(df.cogs.mean())
```

In [19]:
```
1 df['Rating']=df.Rating.fillna(df.Rating.mean())
```

In [20]:
```
1 df.isna().sum()
```

Out[20]:
```
Invoice ID                0
Branch                    0
City                      0
Customer type             0
Gender                    0
Product line              0
Unit price                0
Quantity                  0
Tax 5%                    0
Total                     0
Date                      0
Time                      0
Payment                   0
cogs                      0
gross margin percentage   0
gross income              0
Rating                    0
dtype: int64
```

In [25]:
```
# normal customer no total # member customer no total
total = df.groupby('Customer type')['Total'].sum()
print(total)
```

```
Customer type
Member    164223.444
Normal    158743.305
Name: Total, dtype: float64
```

In [30]:
```
df[df['Customer type']=='Normal']['Total'].sum()

```

Out[30]: 158743.305

In [31]:
```
df[df['Customer type']=='Member']['Total'].sum()
```

Out[31]: 164223.44400000002

Paper

In [33]:
```
df=pd.read_csv("Datasets/train.csv")
df
```

Out[33]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [34]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [35]:
```
1 df.isna().sum()
```

Out[35]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [36]:
```
df=df.dropna()
```

In [37]:
```
df.isna().sum()
```

Out[37]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

In [40]:
```
df.describe(exclude=np.number)
```

Out[40]:

	Name	Sex	Ticket	Cabin	Embarked
count	183	183	183	183	183
unique	183	2	127	133	3
top	Bonnell, Miss. Elizabeth	male	113760	G6	S
freq	1	95	4	4	116

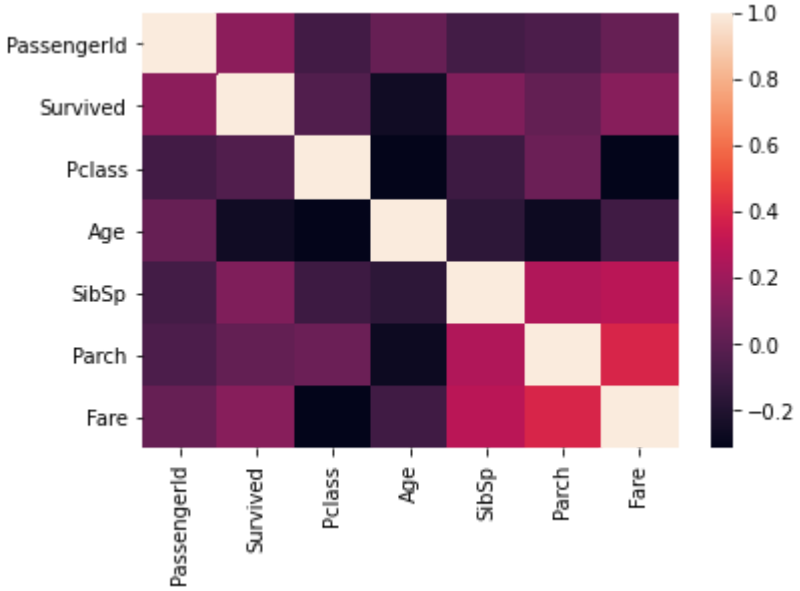
In [43]:
```
1 df.corr()
```

Out[43]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	0.148495	-0.089136	0.030933	-0.083488	-0.051454	0.029740
Survived	0.148495	1.000000	-0.034542	-0.254085	0.106346	0.023582	0.134241
Pclass	-0.089136	-0.034542	1.000000	-0.306514	-0.103592	0.047496	-0.315235
Age	0.030933	-0.254085	-0.306514	1.000000	-0.156162	-0.271271	-0.092424
SibSp	-0.083488	0.106346	-0.103592	-0.156162	1.000000	0.255346	0.286433
Parch	-0.051454	0.023582	0.047496	-0.271271	0.255346	1.000000	0.389740
Fare	0.029740	0.134241	-0.315235	-0.092424	0.286433	0.389740	1.000000

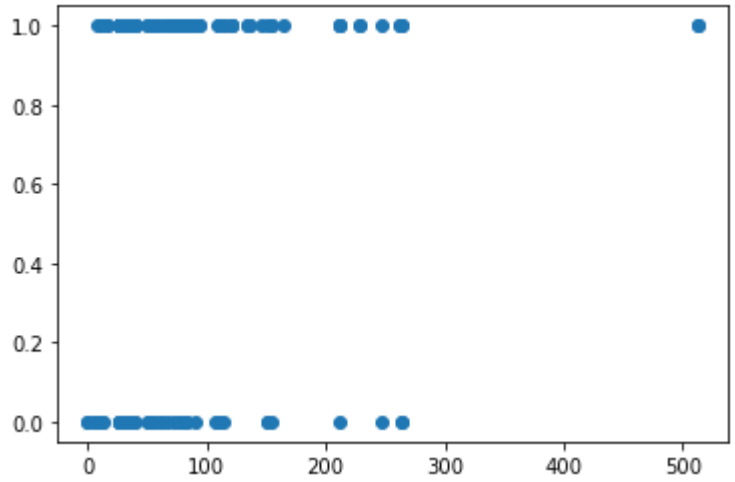
In [44]:
```
1 import seaborn as sns
```

In [45]:
```
1 sns.heatmap(data=df.corr())
2 plt.show()
```



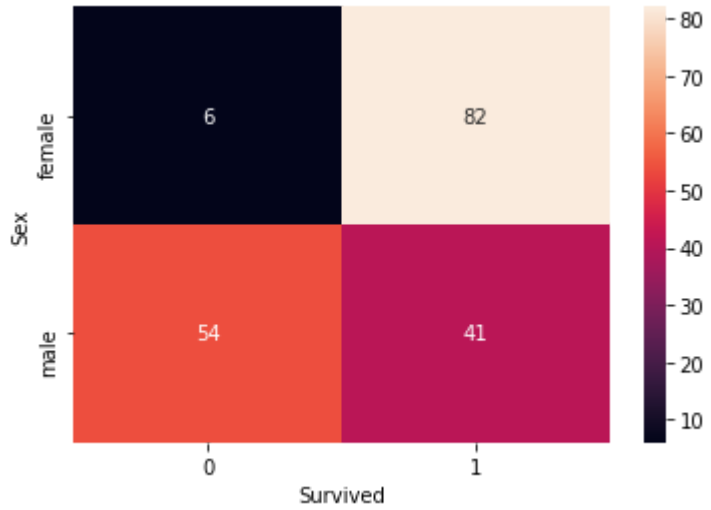
In [46]:
```
1 plt.scatter(df.Fare,df.Survived)
```

Out[46]: <matplotlib.collections.PathCollection at 0x1840834d1c0>



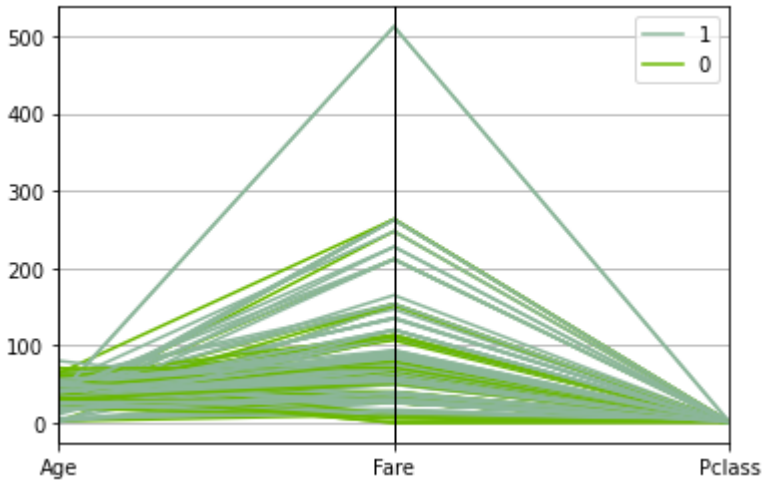
In [47]:
```
1 sns.heatmap(data=pd.crosstab(df.Sex,df.Survived),annot=True)
```

Out[47]: <AxesSubplot:xlabel='Survived', ylabel='Sex'>



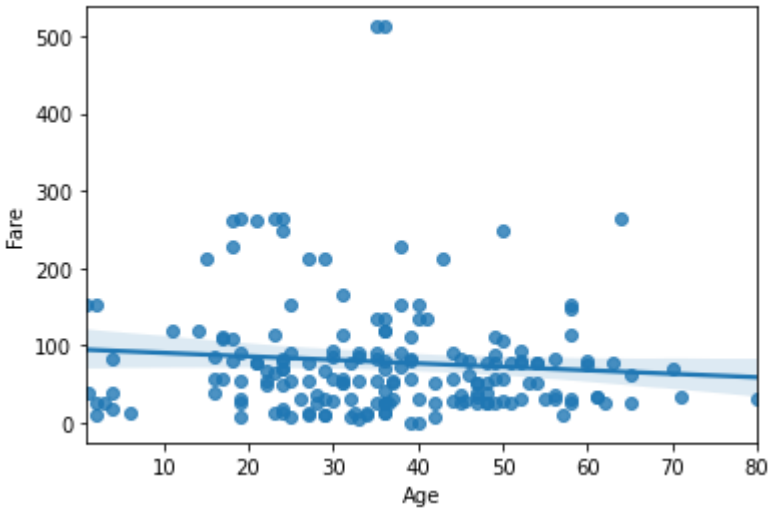
In [53]:
```
1 pd.plotting.parallel_coordinates(df,'Survived',cols=['Age','Fare','Pclass'])
```

Out[53]: <AxesSubplot:>



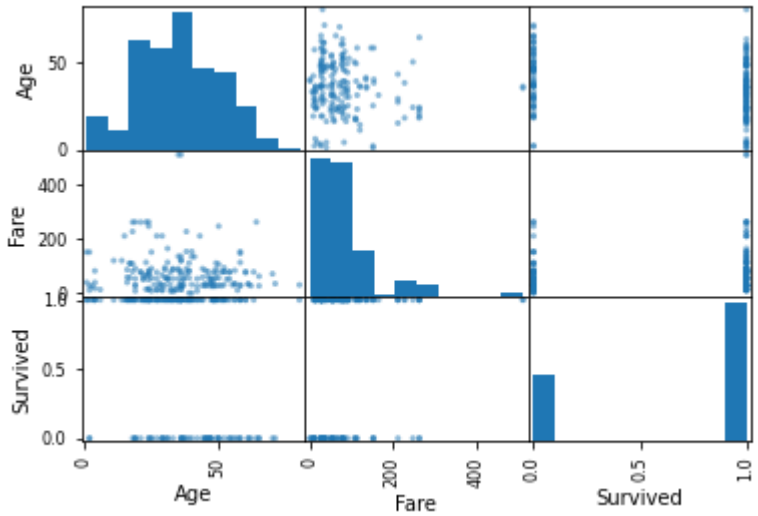
In [54]:
```
1 sns.regplot(x=df.Age,y=df.Fare)
```

Out[54]: <AxesSubplot:xlabel='Age', ylabel='Fare'>



In [55]:
```
1 pd.plotting.scatter_matrix(df[['Age','Fare','Survived']])
```

Out[55]:
```
array([[<AxesSubplot:xlabel='Age', ylabel='Age'>,
        <AxesSubplot:xlabel='Fare', ylabel='Age'>,
        <AxesSubplot:xlabel='Survived', ylabel='Age'>],
       [<AxesSubplot:xlabel='Age', ylabel='Fare'>,
        <AxesSubplot:xlabel='Fare', ylabel='Fare'>,
        <AxesSubplot:xlabel='Survived', ylabel='Fare'>],
       [<AxesSubplot:xlabel='Age', ylabel='Survived'>,
        <AxesSubplot:xlabel='Fare', ylabel='Survived'>,
        <AxesSubplot:xlabel='Survived', ylabel='Survived'>]], dtype=object)
```



In [69]:
```
1 pd.crosstab(df.Sex,df.Survived,margins=True,normalize=True)
```

Out[69]:

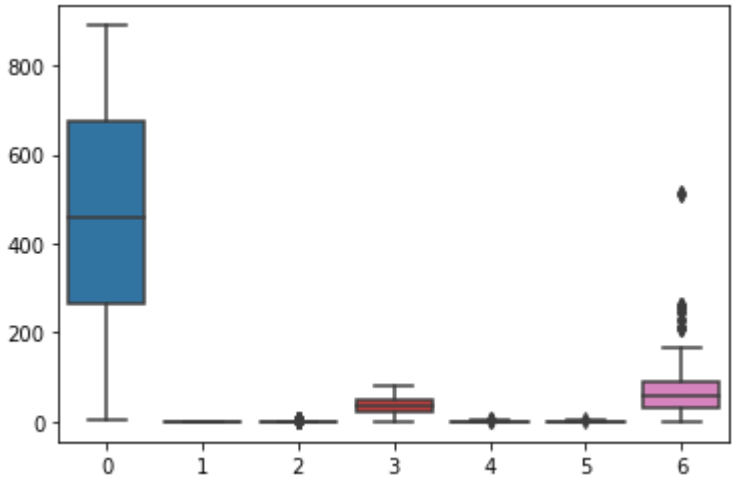
Survived	0	1	All
Sex			
female	0.032787	0.448087	0.480874
male	0.295082	0.224044	0.519126
All	0.327869	0.672131	1.000000

In [63]:
```
x=df[df.Sex=='Male'].shape[0]
y=df[(df.Sex=='Male')&(df.Survived==1)].shape[0]

```

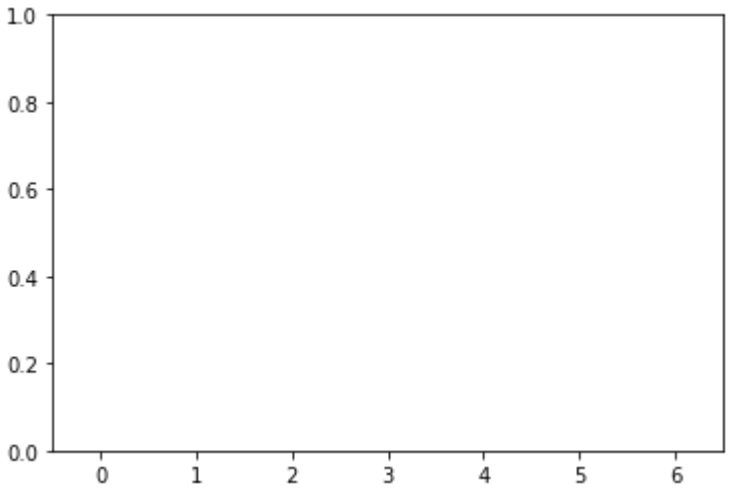
```
0
0
```

In [75]:
```
sns.boxplot(data=(df.PassengerId,df.Survived,df.Pclass,df.Age,df.SibSp,df.Parch,df.Fare))
plt.show()
```



In [81]:
```
q1=df.Fare.quantile(0.25)
q3=df.Fare.quantile(0.75)
iqr=q3-q1
ll=q1-1.5*iqr
ul=q3+1.5*iqr
df=df[(df.Fare<ul)&(df.Fare>ll)]
```

In [87]:
```
sns.boxplot(data=(df.PassengerId,df.Survived,df.Pclass,df.Age,df.SibSp,df.Parch,df.Fare))
plt.show()
```

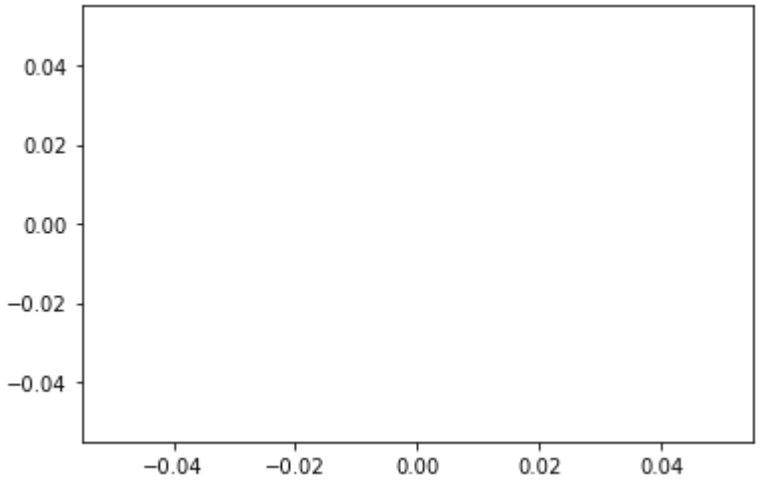


In [88]:
```
df[df.Pclass==1]['Fare'].sum()
```

Out[88]: 0.0

In [89]:
```
1 plt.scatter(df.Age,df.Fare)
```

Out[89]:
```
<matplotlib.collections.PathCollection at 0x18409b07b20>
```



In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```