

## 1) Read The data

```
In [34]: import pandas as pd
df = pd.read_excel("DoctorVisits (2).csv")
df.info()
```

```
Out[34]: Unnamed: 0 visits gender age income illness reduced health private freepoor freerepat nchronic lchronic
0      1      1      1 female 0.19 0.55 1      4      1 yes no no no no no
1      2      3      1 female 0.19 0.45 1      2      1 yes no no no no no
2      3      3      1 male 0.19 0.90 3      0      0 no no no no no no
3      4      4      1 male 0.19 0.15 1      0      0 no no no no no no
4      5      5      1 male 0.19 0.45 2      5      1 no no no no yes no
```

## Display the information of data frame

```
In [9]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5190 entries, 0 to 5189
Data columns (total 13 columns):
 #   Column      Non-Null count  dtype
 --  --
 0   Unnamed: 0  5190 non-null    int64
 1   visits      5190 non-null    int64
 2   gender      5190 non-null    object
 3   age         5190 non-null    float64
 4   income      5190 non-null    float64
 5   illness     5190 non-null    int64
 6   reduced     5190 non-null    int64
 7   health      5190 non-null    int64
 8   private     5190 non-null    object
 9   freepoor    5190 non-null    object
10  freerepat   5190 non-null    object
11 nchronic    5190 non-null    object
12 lchronic    5190 non-null    object
dtypes: float64(2), int64(5), object(6)
memory usage: 527.2+ KB
```

## Describing the data in terms of their datatypes

```
In [18]: df.describe()
```

```
Out[18]: Unnamed: 0 visits age income illness reduced health
count 5190.000000  5190.000000  5190.000000  5190.000000  5190.000000  5190.000000
mean 2595.500000    0.301734    0.406385    0.583160    1.431965    0.861850
std 1498.368279    0.798134    0.204782    0.369907    1.384152    2.887628
min 1498.368279    0.000000    0.190000    0.000000    0.000000    0.000000
25% 1298.250000    0.000000    0.220000    0.250000    0.000000    0.000000
50% 2595.500000    0.000000    0.320000    0.550000    1.000000    0.000000
75% 3952.750000    0.000000    0.620000    0.900000    2.000000    0.000000
max 5190.000000    9.000000    0.720000    1.500000    5.000000    12.000000
```

## 2) Cleaning the Data

```
In [12]: df.dropna(axis=1)
```

```
Out[12]: Unnamed: 0 visits gender age income illness reduced health private freepoor freerepat nchronic lchronic
0      1      1      1 female 0.19 0.55 1      4      1 yes no no no no no
1      2      3      1 female 0.19 0.45 1      2      1 yes no no no no no
2      3      3      1 male 0.19 0.90 3      0      0 no no no no no no
3      4      4      1 male 0.19 0.15 1      0      0 no no no no no no
4      5      5      1 male 0.19 0.45 2      5      1 no no no no yes no
...
5185  5186    0 female 0.22 0.55 0      0      0 no no no no no no
5186  5187    0 male 0.27 1.30 0      0      1 no no no no no no
5187  5188    0 female 0.37 0.25 1      0      1 no no yes no no no
5188  5189    0 female 0.52 0.65 0      0      0 no no no no no no
5189  5190    0 male 0.72 0.25 0      0      0 no no yes no no no
5190 rows x 13 columns
```

## Filling the data which are missing

```
In [15]: df.ffill(axis=0)
```

```
Out[15]: Unnamed: 0 visits gender age income illness reduced health private freepoor freerepat nchronic lchronic
0      1      1      1 female 0.19 0.55 1      4      1 yes no no no no no
1      2      2      1 female 0.19 0.45 1      2      1 yes no no no no no
2      3      3      1 male 0.19 0.90 3      0      0 no no no no no no
3      4      4      1 male 0.19 0.15 1      0      0 no no no no no no
4      5      5      1 male 0.19 0.45 2      5      1 no no no no yes no
...
5185  5186    0 female 0.22 0.55 0      0      0 no no no no no no
5186  5187    0 male 0.27 1.30 0      0      1 no no no no no no
5187  5188    0 female 0.37 0.25 1      0      1 no no yes no no no
5188  5189    0 female 0.52 0.65 0      0      0 no no no no no no
5189  5190    0 male 0.72 0.25 0      0      0 no no yes no no no
5190 rows x 13 columns
```

```
In [16]: df.bfill()
```

```
Out[16]: Unnamed: 0 visits gender age income illness reduced health private freepoor freerepat nchronic lchronic
0      1      1      1 female 0.19 0.55 1      4      1 yes no no no no no
1      2      2      1 female 0.19 0.45 1      2      1 yes no no no no no
2      3      3      1 male 0.19 0.90 3      0      0 no no no no no no
3      4      4      1 male 0.19 0.15 1      0      0 no no no no no no
4      5      5      1 male 0.19 0.45 2      5      1 no no no no yes no
...
5185  5186    0 female 0.22 0.55 0      0      0 no no no no no no
5186  5187    0 male 0.27 1.30 0      0      1 no no no no no no
5187  5188    0 female 0.37 0.25 1      0      1 no no yes no no no
5188  5189    0 female 0.52 0.65 0      0      0 no no no no no no
5189  5190    0 male 0.72 0.25 0      0      0 no no yes no no no
5190 rows x 13 columns
```

## Dropping the Duplicates from the dataset

```
In [28]: df.drop_duplicates(subset=['health'])
```

```
Out[28]: Unnamed: 0 visits gender age income illness reduced health private freepoor freerepat nchronic lchronic
0      1      1      1 female 0.19 0.55 1      4      1 yes no no no no no
2      3      1      1 male 0.19 0.90 3      0      0 no no no no no no
5      6      1      1 female 0.19 0.35 5      1      9 no no no no yes no
6      7      1      1 female 0.19 0.85 4      0      2 no no no no no no
7      8      1      1 female 0.19 0.15 3      0      6 no no no no no no
8      9      1      1 female 0.19 0.65 2      0      5 yes no no no no no
15     16      1      1 male 0.19 0.55 2      0      7 no no no no no no
24     25      1      1 female 0.19 0.25 2     14     11 no no yes no yes yes
49     50      1      1 male 0.19 0.15 1      0      4 no yes no no no no
74     75      1      1 male 0.19 0.06 2      0     12 no no no no no no
83     84      1      1 female 0.19 0.35 5     14      3 no no no no no no
129    130      1      1 female 0.22 0.25 1      0     10 yes no no yes no no
192    193      1      1 male 0.22 0.65 4      0      8 no no no no yes no
```

```
In [21]: df.drop_duplicates(subset=['freepoor'])
```

```
Out[21]: Unnamed: 0 visits gender age income illness reduced health private freepoor freerepat nchronic lchronic
0      1      1      1 female 0.19 0.55 1      4      1 yes no no no no no
25     25      1      1 female 0.19 0.15 1      2      6 no yes no yes no no
```

## Observing the Features and Printing the features of data

```
In [24]: df.shape
```

```
Out[24]: (5190, 13)
```

```
In [25]: df.isna().sum()
```

```
Out[25]: Unnamed: 0      0
visits      0
gender      0
age         0
income      0
illness     0
reduced     0
health      0
private     0
freepoor    0
freerepat   0
nchronic    0
lchronic    0
dtype: int64
```

```
In [26]: df.income.isna().sum()
```

```
Out[26]: 0
```

```
In [27]: df.nunique()
```

```
Out[27]: Unnamed: 0      5190
visits      10
gender       2
age         12
income      14
illness      6
reduced     15
health      13
private      2
freepoor     2
freerepat    2
nchronic     2
lchronic     2
dtype: int64
```

```
In [28]: print(df.illness.value_counts())
```

```
0      1638
1      1554
2       945
3       842
4       274
5        235
Name: illness, dtype: int64
```

```
In [29]: df.illness.value_counts()
```

```
Out[29]: 0      1638
1      1554
2       945
3       842
4       274
5        235
Name: illness, dtype: int64
```

```
In [30]: df.freerepat.value_counts()
```

```
Out[30]: no      4999
yes      1991
Name: freerepat, dtype: int64
```

```
In [31]: df.columns
```

```
Out[31]: Index(['Unnamed: 0', 'visits', 'gender', 'age', 'income', 'illness', 'reduced',
'health', 'private', 'freepoor', 'freerepat', 'nchronic', 'lchronic'],
dtype='object')
```

## Analyzing the variables

```
In [4]: import pandas as pd
df = pd.read_excel("DoctorVisits (2).csv")
df.head()
```

```
Out[4]: array(['no', 'yes'], dtype=object)
```

```
In [6]: import pandas as pd
df = pd.read_excel("DoctorVisits (2).csv")
df.head()
```

```
Out[6]: array([1, 3, 2, 5, 4, 0], dtype=int64)
```

```
In [28]: df.gender.unique()
```

```
Out[28]: array(['female', 'male'], dtype=object)
```

```
In [21]: df.visits.unique()
```

```
Out[21]: array([1, 2, 3, 4, 0, 5, 7, 6, 9, 0], dtype=int64)
```

```
In [22]: df.income.unique()
```

```
Out[22]: array([0.55, 0.45, 0.9 , 0.15, 0.35, 0.65, 0.25, 0. , 0.06, 1.1 , 0.75,
0.61, 1.3 , 1.5 ])
```

```
In [23]: df.reduced.unique()
```

```
Out[23]: array([4, 2, 0, 5, 1, 13, 7, 3, 14, 6, 8, 9, 10, 12, 11],
dtype=int64)
```

```
In [24]: df.illness.unique()
```

```
Out[24]: array([1, 3, 2, 5, 4, 0], dtype=int64)
```

```
In [33]: df.age.unique()
```

```
Out[33]: array([0.19, 0.22, 0.27, 0.32, 0.37, 0.42, 0.47, 0.52, 0.57, 0.62, 0.67,
0.72])
```

```
In [34]: df.freepoor.unique()
```

```
Out[34]: array(['no', 'yes'], dtype=object)
```

```
In [35]: df.freerepat.unique()
```

```
Out[35]: array(['no', 'yes'], dtype=object)
```

```
In [36]: df.health.unique()
```

```
Out[37]: array([1, 0, 9, 2, 6, 5, 7, 11, 4, 12, 3, 10, 8], dtype=int64)
```

## 3) Observations

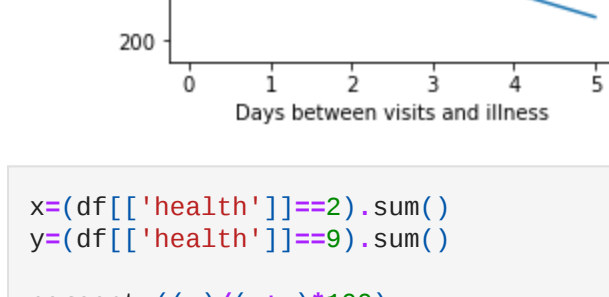
```
In [36]: df.hist(figsize=(20,44))
```



```
In [53]: import matplotlib.pyplot as plt
reduced_df=df[df[['visits','illness']].groupby('illness').count()]
```

```
In [52]: reduced_df.plot(kind='line',figsize=(4,9))
plt.title("Time taken to reduce the disease between visits and illness")
plt.xlabel("Days between visits and illness")
plt.ylabel("No.of visits ")
```

```
Out[52]: Text(0, 0.5, 'No:of visits ')
Time taken to Reduce the disease between visits and illness
```



```
In [41]: x=(df[['health']==2].sum())
y=(df[['health']==9].sum())
percent=((x)/(x+y))*100
percent
```

```
Out[41]: health      6.694561
dtype: float64
```

```
In [18]: df.head()
```

```
Out[18]: Unnamed: 0 visits gender age income illness reduced health private freepoor freerepat nchronic lchronic
0      1      1      1 female 0.19 0.55 1      4      1 yes no no no no no
1      2      3      1 female 0.19 0.45 1      2      1 yes no no no no no
2      3      3      1 male 0.19 0.90 3      0      0 no no no no no no
3      4      4      1 male 0.19 0.15 1      0      0 no no no no no no
4      5      5      1 male 0.19 0.45 2      5      1 no no no no yes no
```

```
In [ ]: import pandas as pd
df = pd.read_excel("DoctorVisits (2).csv")
df.head()
```

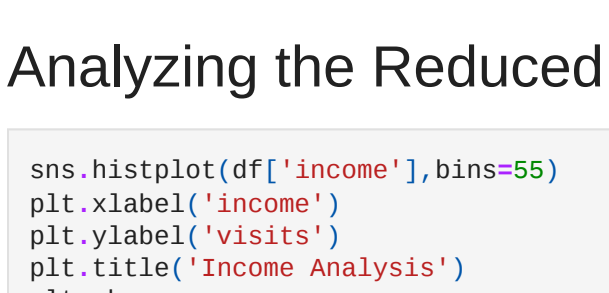
```
In [54]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [56]: df = pd.read_excel("DoctorVisits (2).csv")
df.head(10)
```

```
Out[56]: Unnamed: 0 visits gender age income illness reduced health private freepoor freerepat nchronic lchronic
0      1      1      1 female 0.19 0.55 1      4      1 yes no no no no no
1      2      3      1 female 0.19 0.45 1      2      1 yes no no no no no
2      3      3      1 male 0.19 0.90 3      0      0 no no no no no no
3      4      4      1 male 0.19 0.15 1      0      0 no no no no no no
4      5      5      1 male 0.19 0.45 2      5      1 no no no no yes no
5      6      1      1 female 0.19 0.35 5      1      9 no no no no yes no
6      7      1      1 female 0.19 0.55 4      0      2 no no no no no no
7      8      1      1 female 0.19 0.15 3      0      6 no no no no no no
8      9      1      1 female 0.19 0.65 2      0      5 yes no no no no no
9     10      1      1 male 0.19 0.15 1      0      0 yes no no no no no
```

## Analyzing the data in terms of Patients Visit

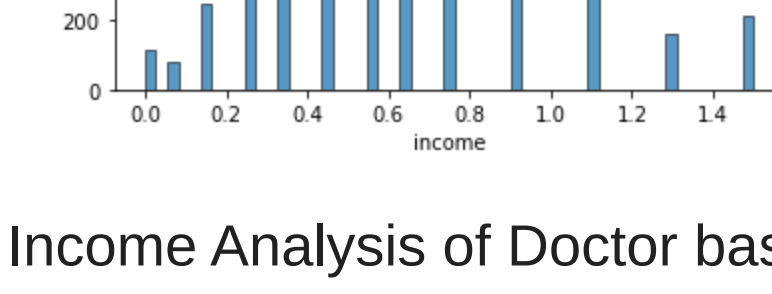
```
In [62]: labels=['visits','health','illness','private','nchronic']
size=(50,30,40,20,10)
plt.pie(sizes,labels=labels,autopct='%1.1f%%')
plt.title('Patient Visiting Analysis')
plt.show()
```



## Analyzing the Reduced illness of Patients based upon visit

```
In [64]: sns.histplot(df['income'],bins=55)
plt.xlabel('income')
sns.barplot(x=reduced_counts.index,y=reduced_counts.values)
plt.xlabel('reduced')
plt.ylabel('visit')
plt.title('Income Recovery Analysis')
plt.show()
```

```
Out[64]: <function matplotlib.pyplot.show(close=None, block=None)>
```



## Income Analysis of Doctor based upon visiting data

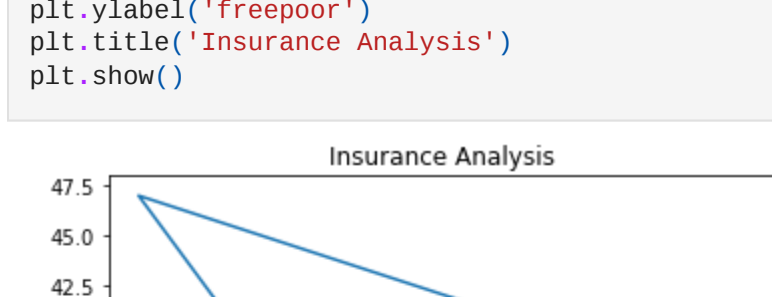
```
In [63]: reduced_counts=df[['reduced']].value_counts()
sns.barplot(x=reduced_counts.index,y=reduced_counts.values)
plt.xlabel('reduced')
plt.ylabel('visit')
plt.title('Patient Recovery Analysis')
plt.show()
```

```
Out[63]: <function matplotlib.pyplot.show(close=None, block=None)>
```



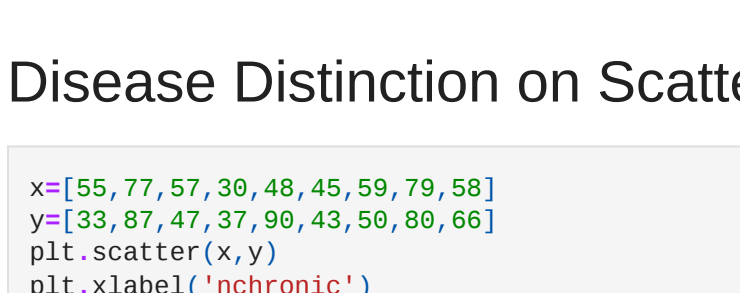
## Insurance Analysis of patients from government

```
In [66]: x=[25,37,5,38]
y=[33,27,47,37]
plt.scatter(x,y)
plt.xlabel('private')
plt.ylabel('freepoor')
plt.title('Insurance Analysis')
plt.show()
```



## Disease Distinction on Scatter Plot

```
In [18]: x=[55,57,30,48,45,59,79,58]
y=[33,67,47,37,60,43,50,60,66]
plt.scatter(x,y)
plt.xlabel('nchronic')
plt.ylabel('lchronic')
plt.title('Disease Analysis')
plt.show()
```



```
In [ ]:
```