



Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3

Code

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

```
In [2]: diab = pd.read_csv('diabetes.csv')
diab
```

```
Out[2]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows x 9 columns

```
In [3]: diab_desc = diab.sort_values(by = ['Outcome'],ascending=False)
diab_desc.head(50)
```

```
Out[3]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1



Type here to search



11:10 AM
11-01-2021

```
In [3]: diab_desc = diab.sort_values(by = ['Outcome'],ascending=False)
        diab_desc.head(50)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
276	7	106	60	24	0	26.5	0.296	29	1
270	10	101	96	37	0	45.6	1.136	38	1
269	2	146	0	0	0	27.5	0.240	28	1
266	0	138	0	0	0	36.3	0.933	25	1
603	7	150	78	29	126	35.2	0.692	54	1
264	4	123	62	0	0	32.0	0.226	35	1
604	4	183	0	0	0	28.4	0.212	36	1
261	3	141	0	0	0	30.0	0.761	27	1
606	1	181	78	42	293	40.0	1.258	22	1
259	11	155	76	28	150	33.3	1.353	51	1
255	1	113	64	35	0	33.6	0.543	21	1
254	12	92	62	7	258	27.6	0.926	44	1
611	3	174	58	22	194	32.9	0.593	36	1
612	7	168	88	42	321	38.2	0.787	40	1
614	11	138	74	26	144	36.1	0.557	50	1
245	9	184	85	15	0	30.0	1.213	49	1
618	9	112	82	24	0	28.2	1.282	50	1
243	6	119	50	22	176	27.1	1.318	33	1
598	1	173	74	0	0	36.8	0.088	38	1
595	0	188	82	14	185	32.0	0.682	22	1
619	0	119	0	0	0	32.4	0.141	24	1

```
In [4]: diab_asce = diab.sort_values(by = ['Outcome'],ascending=True)
        diab_asce.head(50)
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
383	1	90	62	18	59	25.1	1.268	25	0
465	0	124	56	13	105	21.8	0.452	21	0
466	0	74	52	10	36	27.8	0.269	22	0
467	0	97	64	36	100	36.8	0.600	25	0
469	6	154	78	41	140	46.1	0.571	27	0
470	1	144	82	40	0	41.3	0.607	28	0
471	0	137	70	38	0	33.2	0.170	22	0
472	0	119	66	27	0	38.8	0.259	22	0
473	7	136	90	0	0	29.9	0.210	50	0
474	4	114	64	0	0	28.9	0.126	24	0
475	0	137	84	27	0	27.3	0.231	59	0
477	7	114	76	17	110	23.8	0.406	31	0
478	8	126	74	38	75	25.9	0.162	39	0
479	4	132	86	31	0	28.0	0.419	63	0
481	0	123	88	37	0	35.2	0.197	29	0
482	4	85	58	22	49	27.8	0.306	28	0
464	10	115	98	0	0	24.0	1.022	34	0
483	0	84	82	31	125	38.2	0.233	23	0
463	5	88	78	30	0	27.6	0.258	37	0
461	1	71	62	0	0	21.8	0.416	26	0
438	1	97	70	15	0	18.2	0.147	21	0
439	6	107	88	0	0	36.8	0.727	31	0

```
In [5]: diab50 = diab.head(50)
        diab50
```

Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1
10	4	110	92	0	0	37.6	0.191	30	0
11	10	168	74	0	0	38.0	0.537	34	1
12	10	139	80	0	0	27.1	1.441	57	0
13	1	189	60	23	846	30.1	0.398	59	1
14	5	166	72	19	175	25.8	0.587	51	1
15	7	100	0	0	0	30.0	0.484	32	1
16	0	118	84	47	230	45.8	0.551	31	1
17	7	107	74	0	0	29.6	0.254	31	1
18	1	103	30	38	83	43.3	0.183	33	0
19	1	115	70	30	96	34.6	0.529	32	1
20	3	126	88	41	235	39.3	0.704	27	0
21	8	99	84	0	0	35.4	0.388	50	0

1. Missing values

In [6]: `diab.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Pregnancies            768 non-null    int64
1   Glucose                768 non-null    int64
2   BloodPressure          768 non-null    int64
3   SkinThickness          768 non-null    int64
4   Insulin                768 non-null    int64
5   BMI                    768 non-null    float64
6   DiabetesPedigreeFunction 768 non-null    float64
7   Age                    768 non-null    int64
8   Outcome                768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

This tells us that there are no missing values. clear!

2. Description

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

out[7]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105409	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.780232	0.476951

2. Description

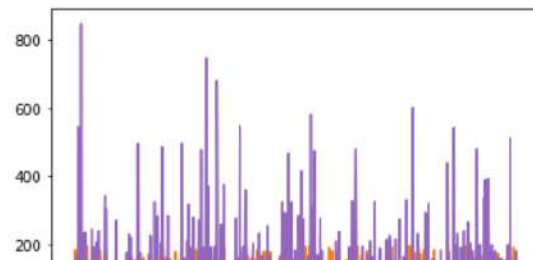
```
In [7]: diab.describe()
```

Out[7]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

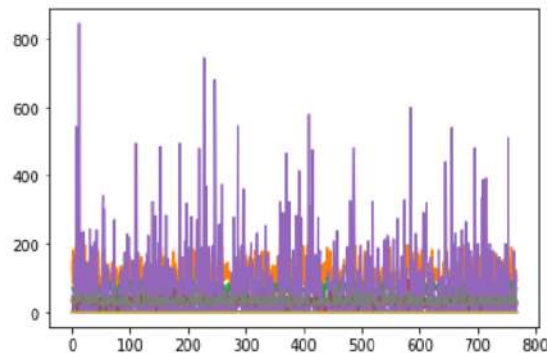
3. Plots

```
In [8]: plt.plot(diab)
plt.show()
```



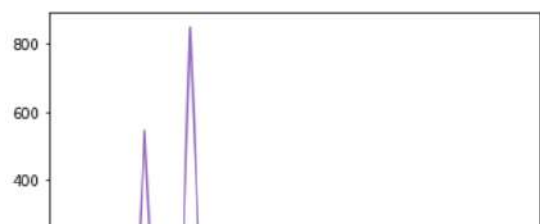
3. Plots

```
In [8]: plt.plot(diab)
plt.show()
```



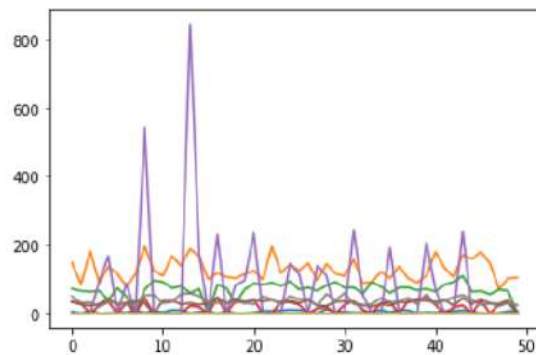
```
In [9]: diab50 = diab.head(50)
```

```
In [10]: plt.plot(diab50)
plt.show()
```



```
In [9]: diab50 = diab.head(50)
```

```
In [10]: plt.plot(diab50)
plt.show()
```



4. Bivariate analysis

A. Between parameters and the target.

```
In [11]: # pregnancies and outcome
plt.scatter(diab['Pregnancies'], diab['Outcome'])
plt.xticks([0,2,4,6,8,10,12,14,16,18])
plt.show()
```

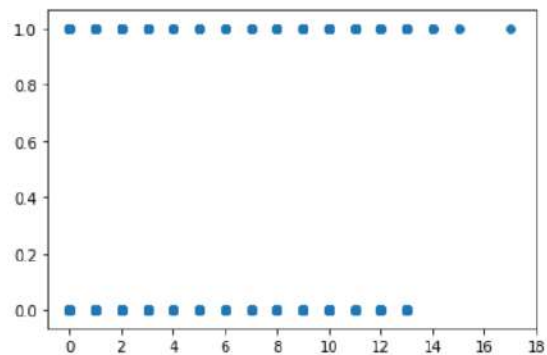




4. Bivariate analysis

A. Between parameters and the target.

```
In [11]: # pregnancies and outcome
plt.scatter(diab['Pregnancies'], diab['Outcome'])
plt.xticks([0,2,4,6,8,10,12,14,16,18])
plt.show()
```



From the plot, people who are pregnant more than 13 times are more prone to diabetes.

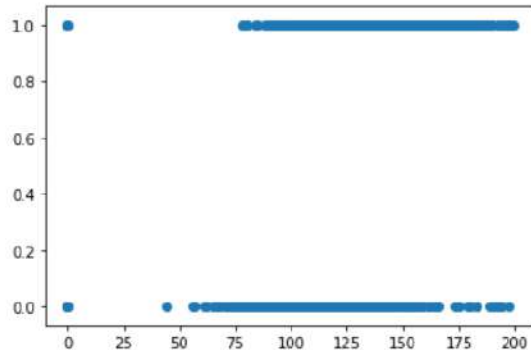
```
In [12]: # diab['Pregnancies'].isin([15])
```

```
In [13]: # glucose and outcome
plt.scatter(diab['Glucose'], diab['Outcome'])
plt.show()
```



```
In [12]: # diab['Pregnancies'].isin([15])

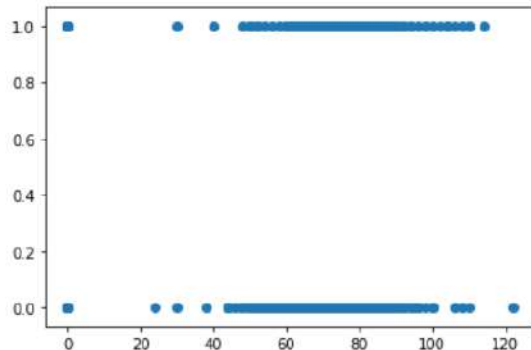
In [13]: # glucose and outcome
plt.scatter(diab['Glucose'], diab['Outcome'])
plt.show()
```



```
In [14]: # BloodPressure and outcome
plt.scatter(diab['BloodPressure'], diab['Outcome'])
plt.show()
```



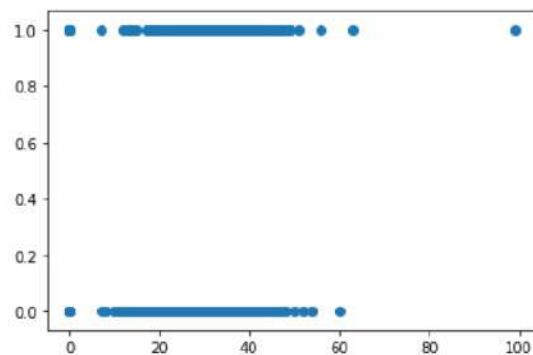
```
In [14]: # BloodPressure and outcome
plt.scatter(diab['BloodPressure'], diab['Outcome'])
plt.show()
```



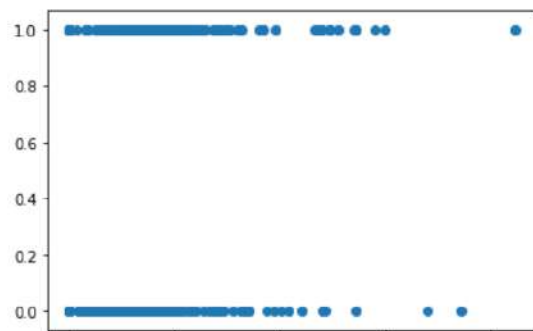
```
In [15]: # SkinThickness and outcome
plt.scatter(diab['SkinThickness'], diab['Outcome'])
plt.show()
```



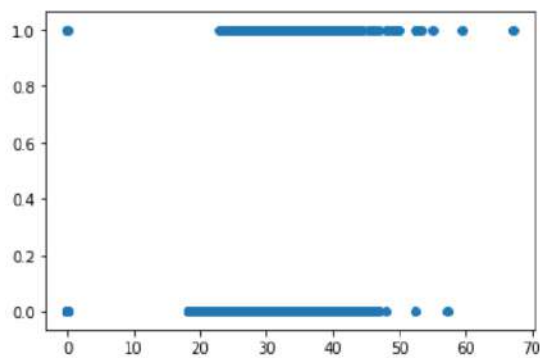
```
In [15]: # SkinThickness and outcome
plt.scatter(diab['SkinThickness'], diab['Outcome'])
plt.show()
```



```
In [16]: # Insulin and outcome
plt.scatter(diab['Insulin'], diab['Outcome'])
plt.show()
```



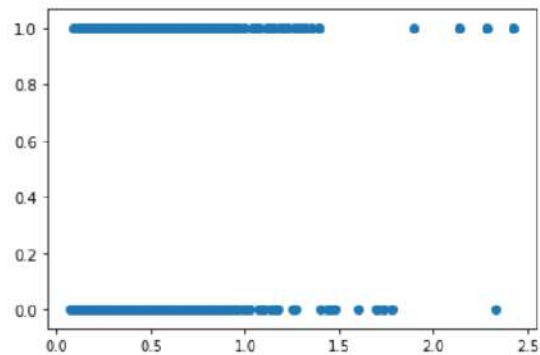
```
In [17]: # BMI and outcome
plt.scatter(diab['BMI'], diab['Outcome'])
plt.show()
```



```
In [18]: # Age and outcome
plt.scatter(diab['Age'], diab['Outcome'])
plt.show()
```

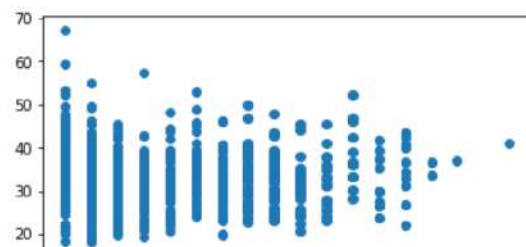



```
In [19]: # DPF and outcome
plt.scatter(diab['DiabetesPedigreeFunction'], diab['Outcome'])
plt.show()
```



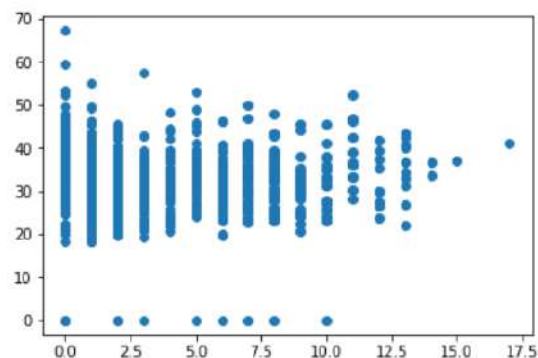
Bilateral analysis of various properties other than outcome.

```
In [20]: plt.scatter(diab['Pregnancies'], diab['BMI'])
plt.show()
```

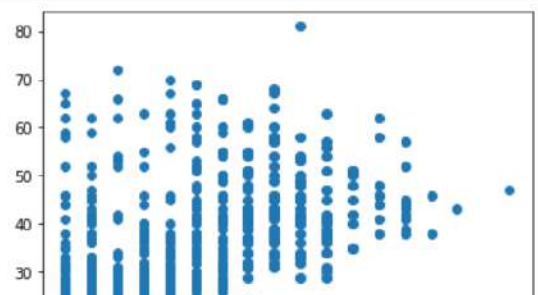


Bilateral analysis of various properties other than outcome.

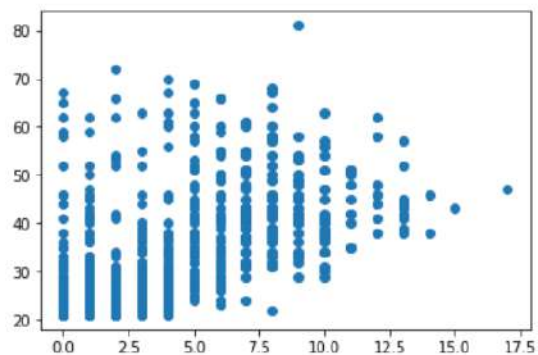
```
In [20]: plt.scatter(diab['Pregnancies'], diab['BMI'])  
plt.show()
```



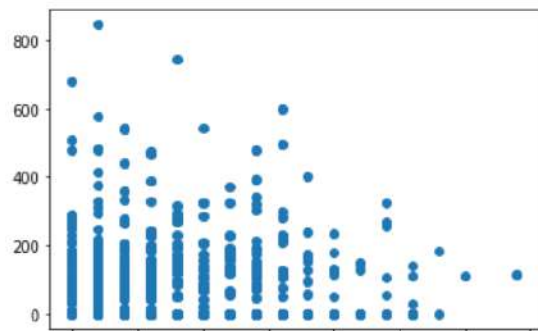
```
In [21]: plt.scatter(diab['Pregnancies'], diab['Age'])  
plt.show()
```



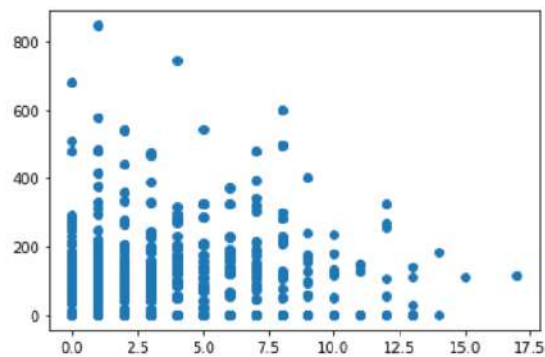
```
In [21]: plt.scatter(diab['Pregnancies'], diab['Age'])
plt.show()
```



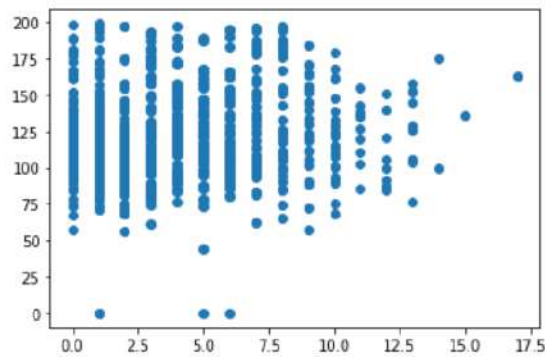
```
In [22]: plt.scatter(diab['Pregnancies'], diab['Insulin'])
plt.show()
```



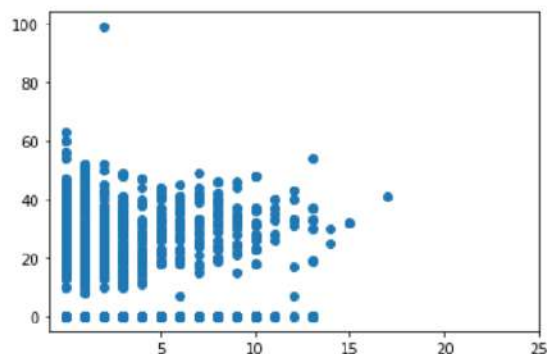
```
In [22]: plt.scatter(diab['Pregnancies'], diab['Insulin'])  
plt.show()
```



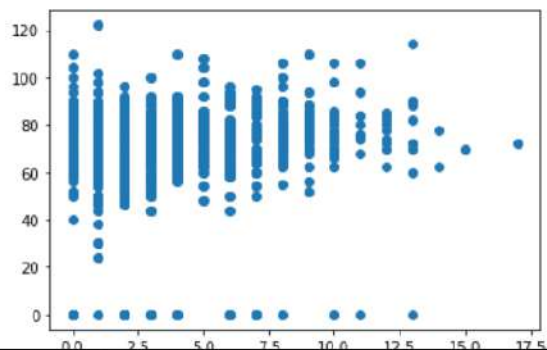
```
In [23]: plt.scatter(diab['Pregnancies'], diab['Glucose'])  
plt.show()
```



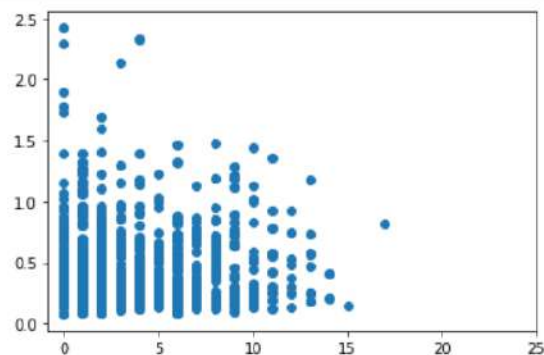
```
In [24]: plt.scatter(diab['Pregnancies'], diab['SkinThickness'])  
plt.xticks([5,10,15,20,25])  
plt.show()
```



```
In [25]: plt.scatter(diab['Pregnancies'], diab['BloodPressure'])  
plt.show()
```



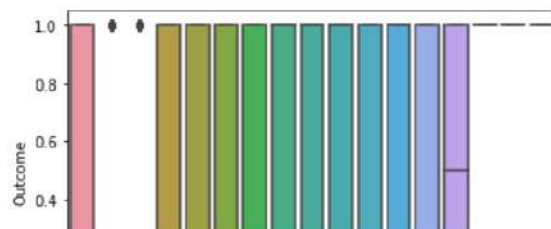

```
In [26]: plt.scatter(diab['Pregnancies'], diab['DiabetesPedigreeFunction'])
plt.xticks([0,5,10,15,20,25])
plt.show()
```



5.Box Plots

```
In [27]: sns.boxplot(x='Pregnancies',y='Outcome',data= diab)
```

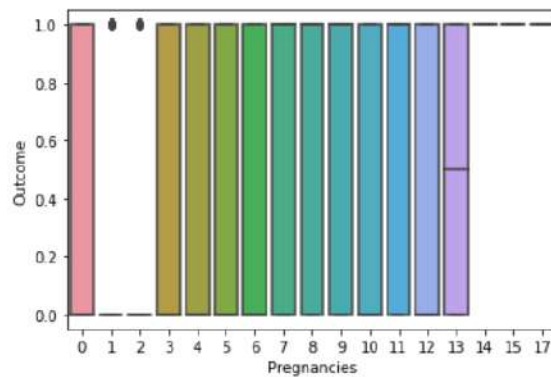
```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1be62950>
```



5.Box Plots

```
In [27]: sns.boxplot(x='Pregnancies',y='Outcome',data= diab)
```

```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1be62950>
```



```
In [28]: # list = np.arange(1,200,1)
# glucose = sns.boxplot(x='Glucose',y='Outcome',data= diab)
# glucose.set(xticklabels=List)
# glucose
```

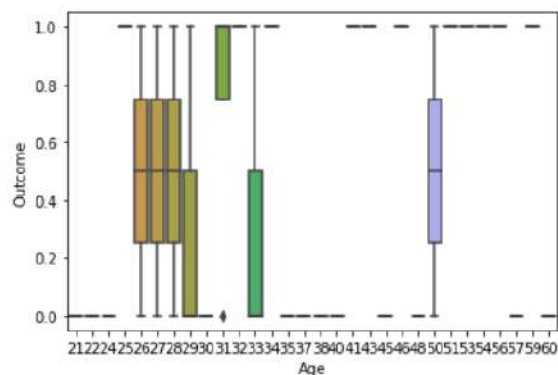
```
In [30]: sns.boxplot(x='Age',y='Outcome',data= diab50)
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1c549890>
```



```
In [30]: sns.boxplot(x='Age',y='Outcome',data= diab50)
```

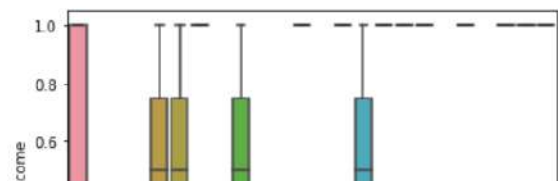
```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1c549890>
```



```
In [ ]: # sns.boxplot(x='BMI',y='Outcome',data= diab)
```

```
In [31]: list = np.arange(0,100,10)
st = sns.boxplot(x='SkinThickness',y='Outcome',data= diab50)
st.set(xticklabels=list)
st
```

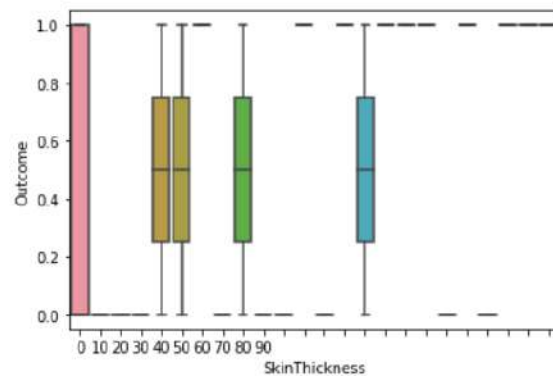
```
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1c147310>
```



```
In [ ]: # sns.boxplot(x='BMI',y='Outcome',data= diab)
```

```
In [31]: list = np.arange(0,100,10)
          st = sns.boxplot(x='SkinThickness',y='Outcome',data= diab50)
          st.set(xticklabels=list)
          st
```

```
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1c147310>
```



In []: