Installing Libraries

```
%pip install matplotlib

Requirement already satisfied: matplotlib in c:\users\aakas\appdata\
local\programs\python\python312\lib\site-packages (3.8.3)Note: you may
need to restart the kernel to use updated packages.

Requirement already satisfied: contourpy>=1.0.1 in c:\users\aakas\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\aakas\appdata\
local\programs\python\python312\lib\site-packages (from matplotlib)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\aakas\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (4.49.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\aakas\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (1.4.5)
Requirement already satisfied: numpy<2,>=1.21 in c:\users\aakas\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\aakas\
appdata\roaming\python\python312\site-packages (from matplotlib)
(23.2)
Requirement already satisfied: pillow>=8 in c:\users\aakas\appdata\
local\programs\python\python312\lib\site-packages (from matplotlib)
(10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\aakas\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\aakas\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\aakas\appdata\
roaming\python\python312\site-packages (from python-dateutil>=2.7-
>matplotlib) (1.16.0)
```

Importing Libraries

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt

C:\Users\aakas\AppData\Local\Temp\ipykernel_10312\3311980270.py:1:
DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major
release of pandas (pandas 3.0),
```

```
(to allow more performant data types, such as the Arrow string type,
and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at
https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd

virus = pd.read_csv("assignment3-part2.csv")

virus.replace("?", np.nan, inplace=True)

virus.isna().sum()

Unnamed: 0    0
ANTI_A        0
ANTI_B        0
V1            0
dtype: int64
```

Binning

```
num_bin = 5

label = ["A", "B", "C","D","E"]

virus["E-W-Partition"] = pd.cut(virus["V1"], bins=num_bin,
labels=label)
virus["E-F-Partition"] = pd.qcut(virus["V1"], q = num_bin,
labels=label)

virus.iloc[:]
```

	Unnamed: 0	ANTI_A	ANTI_B	V1	E-W-Partition	E-F-Partition
0	0	2.77	98.1	6.0	A	A
1	1	9.79	16.8	6.7	A	A
2	2	9.29	13.6	7.5	A	A
3	3	3.41	1.6	7.6	A	A
4	4	9.83	55.3	8.4	A	A
5	5	5.76	94.4	9.3	A	A
6	6	2.73	67.2	9.4	A	A
7	7	0.17	41.9	9.6	A	A
8	8	7.50	22.7	10.6	A	B
9	9	0.21	39.6	11.2	A	B
10	10	0.66	61.6	11.3	A	B
11	11	2.84	78.9	11.5	A	B
12	12	7.90	15.5	11.5	A	B
13	13	2.00	93.3	11.9	A	B
14	14	5.07	87.9	13.0	A	B
15	15	2.14	85.0	13.8	A	B

16	16	9.94	16.4	14.1	A	C
17	17	7.21	20.6	14.9	A	C
18	18	4.41	4.8	15.2	A	C
19	19	8.70	48.3	15.8	A	C
20	20	3.05	86.3	16.5	A	C
21	21	9.34	52.6	17.7	A	C
22	22	7.36	13.0	21.4	A	C
23	23	7.79	61.4	55.5	C	C
24	24	4.24	53.3	60.0	D	D
25	25	0.63	11.3	61.4	D	D
26	26	1.43	26.2	62.1	D	D
27	27	1.44	32.9	64.9	D	D
28	28	7.34	83.4	64.9	D	D
29	29	1.74	12.7	69.0	D	D
30	30	6.12	35.4	70.1	D	D
31	31	7.59	76.7	73.4	D	D
32	32	7.31	58.4	74.9	E	E
33	33	7.17	86.9	76.8	E	E
34	34	2.99	20.6	77.0	E	E
35	35	2.84	33.6	77.9	E	E
36	36	0.67	9.2	80.1	E	E
37	37	8.62	78.9	81.2	E	E
38	38	7.44	98.8	83.8	E	E
39	39	7.21	88.0	90.7	E	E

Dividing Dataset with V1 less than and greater than 40

```
threshold = 40

virus_low = virus[virus["V1"] < threshold]
virus_high = virus[virus["V1"] > threshold]

virus_low.describe()
```

	Unnamed: 0	ANTI_A	ANTI_B	V1
count	23.00000	23.000000	23.000000	23.000000
mean	11.00000	5.307826	48.495652	11.952174
std	6.78233	3.363449	32.804302	3.830840
min	0.00000	0.170000	1.600000	6.000000
25%	5.50000	2.750000	16.600000	9.350000
50%	11.00000	5.070000	48.300000	11.500000
75%	16.50000	8.300000	81.950000	14.500000
max	22.00000	9.940000	98.100000	21.400000

```
virus_high.describe()
```

	Unnamed: 0	ANTI_A	ANTI_B	V1
count	17.000000	17.000000	17.000000	17.000000
mean	31.000000	4.857059	51.041176	71.982353
std	5.049752	2.937230	30.416526	9.593112

```
min     23.000000   0.630000   9.200000  55.500000
25%     27.000000   1.740000  26.200000  64.900000
50%     31.000000   6.120000  53.300000  73.400000
75%     35.000000   7.340000  78.900000  77.900000
max     39.000000   8.620000  98.800000  90.700000
```

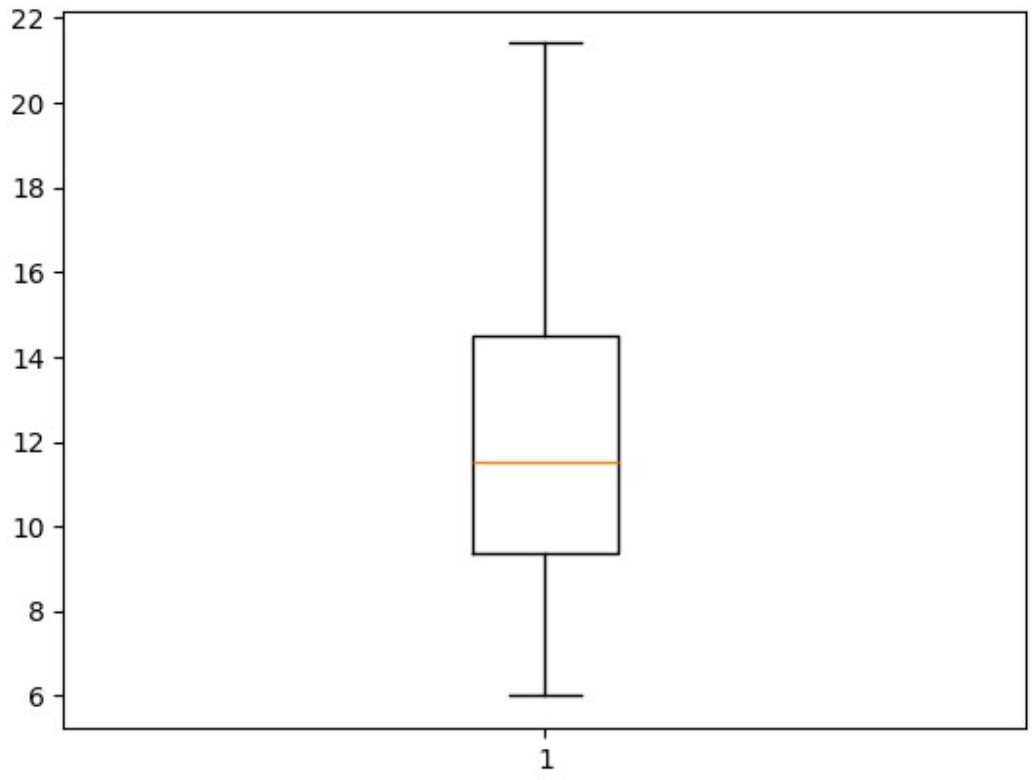
Plotting Boxplot from V1 for both low and high groups

```
fig = plt.figure(figsize= (10,5))
```

```
<Figure size 1000x500 with 0 Axes>
```

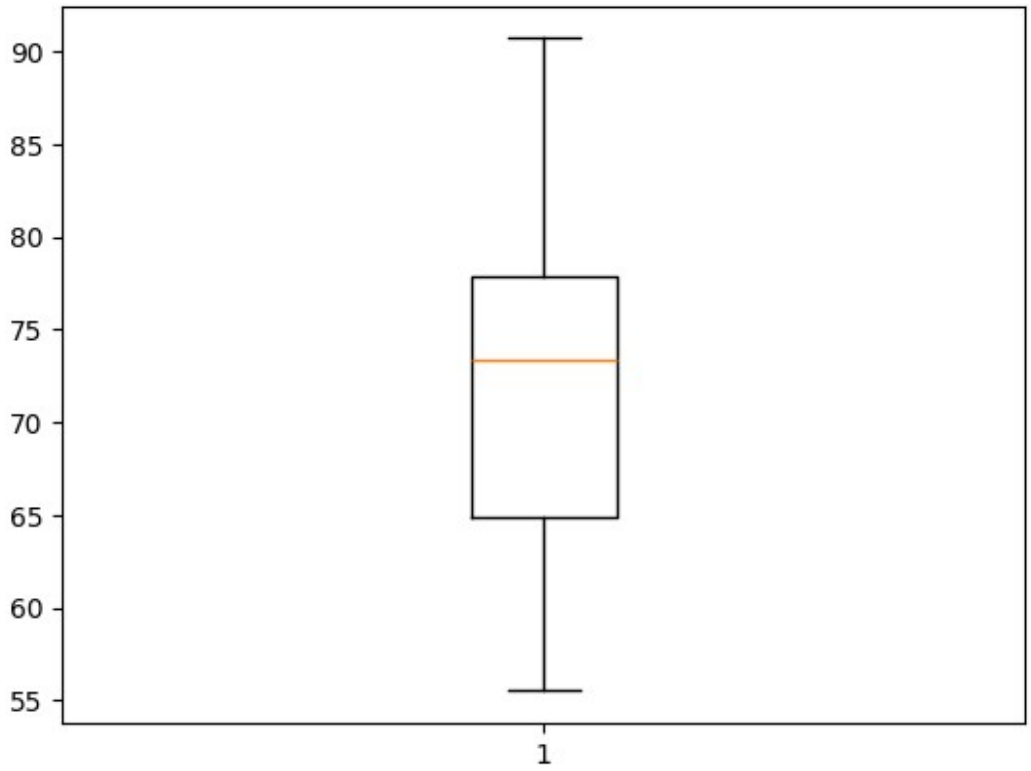
```
plt.boxplot(virus_low["V1"])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1593850e540>,
  <matplotlib.lines.Line2D at 0x159385cc440>],
 'caps': [<matplotlib.lines.Line2D at 0x159385cc740>,
  <matplotlib.lines.Line2D at 0x159385cca70>],
 'boxes': [<matplotlib.lines.Line2D at 0x1593858ff80>],
 'medians': [<matplotlib.lines.Line2D at 0x159385ccd40>],
 'fliers': [<matplotlib.lines.Line2D at 0x159385cd040>],
 'means': []}
```




```
plt.boxplot(virus_high["V1"])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1593a8901a0>,
  <matplotlib.lines.Line2D at 0x1593a8904d0>],
 'caps': [<matplotlib.lines.Line2D at 0x1593a8907d0>,
  <matplotlib.lines.Line2D at 0x1593a890ad0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1593a84bef0>],
 'medians': [<matplotlib.lines.Line2D at 0x1593a890dd0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1593a891040>],
 'means': []}
```




```
virus_low["ANTI_A"] = (virus_low["ANTI_A"] -
virus_low["ANTI_A"].min())/(virus_low["ANTI_A"].max() -
virus_low["ANTI_A"].min())
virus_low["ANTI_B"] = (virus_low["ANTI_B"] -
virus_low["ANTI_B"].min())/(virus_low["ANTI_B"].max() -
virus_low["ANTI_B"].min())
virus_low
```

```
C:\Users\aakas\AppData\Local\Temp\ipykernel_10312\3142245604.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
```

```
  virus_low["ANTI_A"] = (virus_low["ANTI_A"] -
virus_low["ANTI_A"].min())/(virus_low["ANTI_A"].max() -
virus_low["ANTI_A"].min())
C:\Users\aakas\AppData\Local\Temp\ipykernel_10312\3142245604.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  virus_low["ANTI_B"] = (virus_low["ANTI_B"] -
virus_low["ANTI_B"].min())/(virus_low["ANTI_B"].max() -
virus_low["ANTI_B"].min())
```

	Unnamed: 0	ANTI_A	ANTI_B	V1	E-W-Partition	E-F-Partition
0	0	0.266121	1.000000	6.0	A	A
1	1	0.984647	0.157513	6.7	A	A
2	2	0.933470	0.124352	7.5	A	A
3	3	0.331627	0.000000	7.6	A	A
4	4	0.988741	0.556477	8.4	A	A
5	5	0.572160	0.961658	9.3	A	A
6	6	0.262027	0.679793	9.4	A	A
7	7	0.000000	0.417617	9.6	A	A
8	8	0.750256	0.218653	10.6	A	B
9	9	0.004094	0.393782	11.2	A	B
10	10	0.050154	0.621762	11.3	A	B
11	11	0.273286	0.801036	11.5	A	B
12	12	0.791198	0.144041	11.5	A	B
13	13	0.187308	0.950259	11.9	A	B
14	14	0.501535	0.894301	13.0	A	B
15	15	0.201638	0.864249	13.8	A	B
16	16	1.000000	0.153368	14.1	A	C
17	17	0.720573	0.196891	14.9	A	C
18	18	0.433982	0.033161	15.2	A	C
19	19	0.873081	0.483938	15.8	A	C
20	20	0.294780	0.877720	16.5	A	C
21	21	0.938588	0.528497	17.7	A	C
22	22	0.735926	0.118135	21.4	A	C

Min-Max Normalization

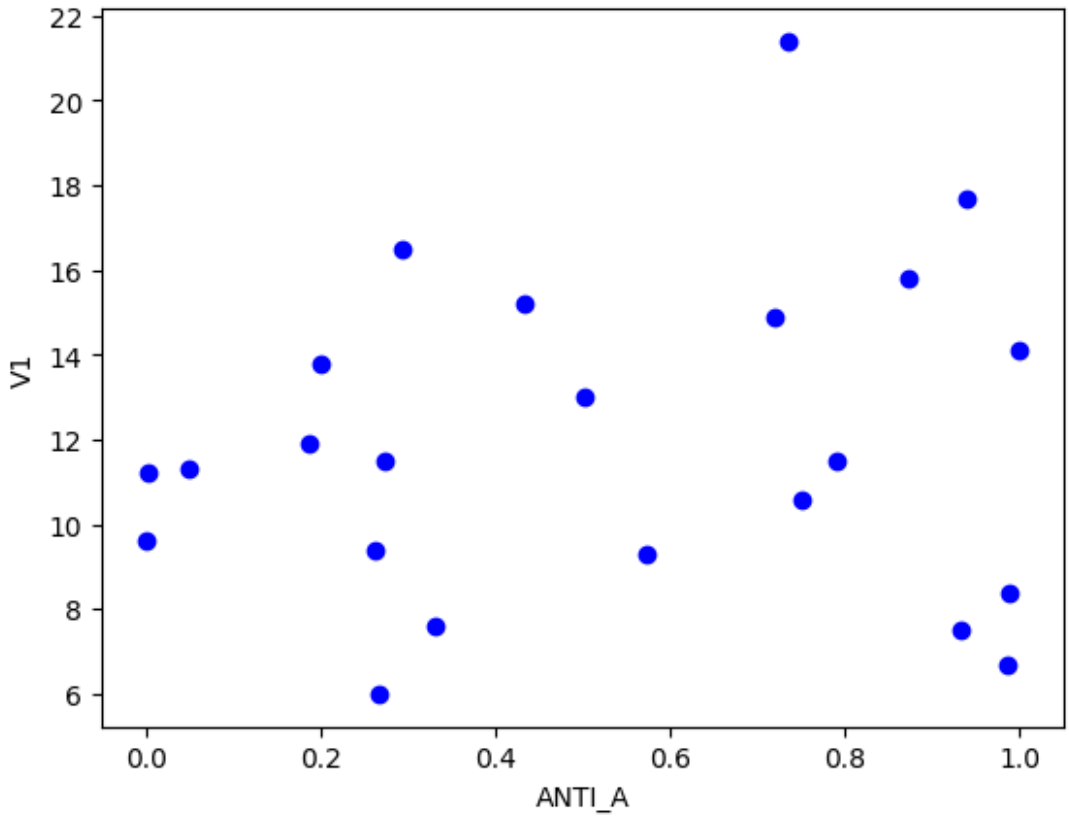
```
virus_high["ANTI_A"] = (virus_high["ANTI_A"] -
virus_high["ANTI_A"].min())/(virus_high["ANTI_A"].max() -
virus_high["ANTI_A"].min())
virus_high["ANTI_B"] = (virus_high["ANTI_B"] -
virus_high["ANTI_B"].min())/(virus_high["ANTI_B"].max() -
virus_high["ANTI_B"].min())
virus_high
```

```
C:\Users\aakas\AppData\Local\Temp\ipykernel_10312\34499113.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  virus_high["ANTI_A"] = (virus_high["ANTI_A"] -
virus_high["ANTI_A"].min())/(virus_high["ANTI_A"].max() -
virus_high["ANTI_A"].min())
C:\Users\aakas\AppData\Local\Temp\ipykernel_10312\34499113.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  virus_high["ANTI_B"] = (virus_high["ANTI_B"] -
virus_high["ANTI_B"].min())/(virus_high["ANTI_B"].max() -
virus_high["ANTI_B"].min())
```

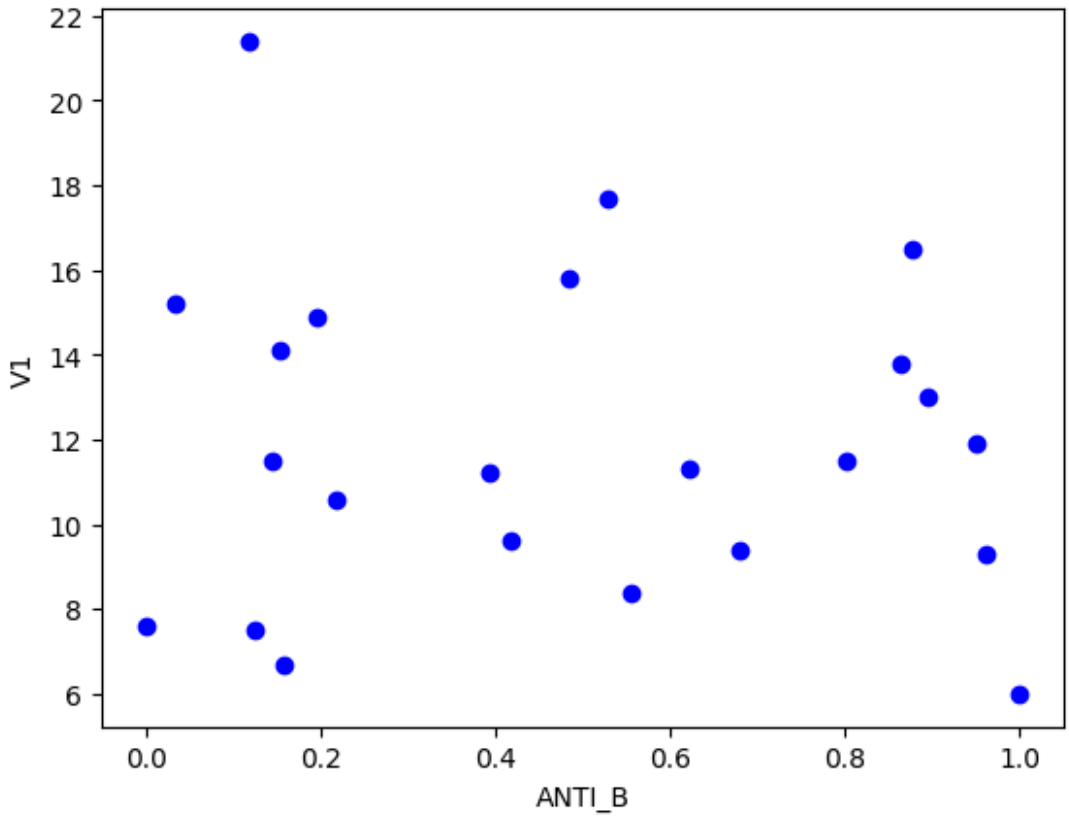
	Unnamed: 0	ANTI_A	ANTI_B	V1	E-W-Partition	E-F-Partition
23	23	0.896120	0.582589	55.5	C	C
24	24	0.451815	0.492187	60.0	D	D
25	25	0.000000	0.023438	61.4	D	D
26	26	0.100125	0.189732	62.1	D	D
27	27	0.101377	0.264509	64.9	D	D
28	28	0.839800	0.828125	64.9	D	D
29	29	0.138924	0.039062	69.0	D	D
30	30	0.687109	0.292411	70.1	D	D
31	31	0.871089	0.753348	73.4	D	D
32	32	0.836045	0.549107	74.9	E	E
33	33	0.818523	0.867188	76.8	E	E
34	34	0.295369	0.127232	77.0	E	E
35	35	0.276596	0.272321	77.9	E	E
36	36	0.005006	0.000000	80.1	E	E
37	37	1.000000	0.777902	81.2	E	E
38	38	0.852315	1.000000	83.8	E	E
39	39	0.823529	0.879464	90.7	E	E

Plotting Scatterplot

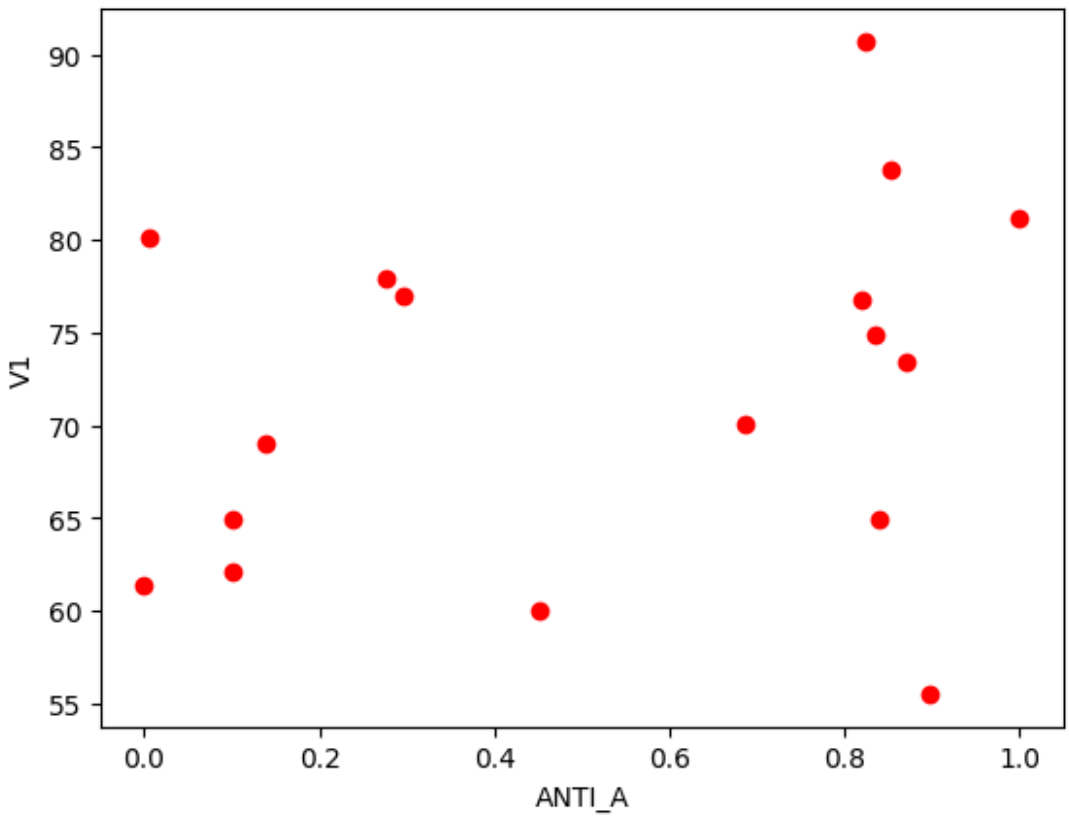
```
plt.scatter(virus_low["ANTI_A"], virus_low["V1"], c="blue")
plt.xlabel("ANTI_A")
plt.ylabel("V1")
plt.show()
```



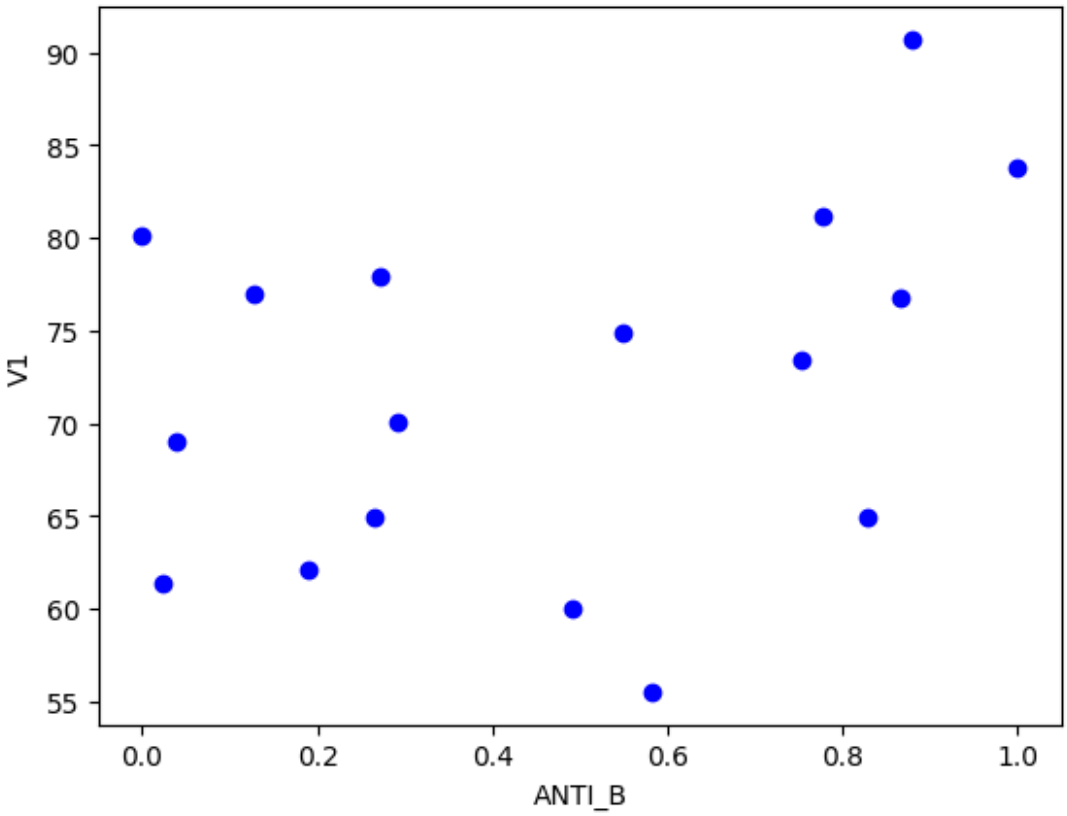

```
plt.scatter(virus_low["ANTI_B"], virus_low["V1"], c="blue")
plt.xlabel("ANTI_B")
plt.ylabel("V1")
plt.show()
```




```
plt.scatter(virus_high["ANTI_A"], virus_high["V1"], c="red")
plt.xlabel("ANTI_A")
plt.ylabel("V1")
plt.show()
```




```
plt.scatter(virus_high["ANTI_B"], virus_high["V1"], c="blue")
plt.xlabel("ANTI_B")
plt.ylabel("V1")
plt.show()
```



Corelation between ANTI_A and ANTI_B

```
virus_low["ANTI_A"].corr(virus_low["ANTI_B"])
```

```
-0.44962747254599306
```

```
virus_high["ANTI_A"].corr(virus_high["ANTI_B"])
```

```
0.8938242533758364
```

For low values, we must use only one of the medicines i.e ANTI_A or ANTI_B as the corelation is negative but for high values, we can use both ANTI_A and ANTI_B together as the corelation is positive