

```
In [1]: import pandas as pd
df = pd.read_csv(r"D:\log2.csv")
print("Original DataFrame:")
print(df)
```

Original DataFrame:

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	\
0	57222	53	54587	53	
1	56258	3389	56258	3389	
2	6881	50321	43265	50321	
3	50553	3389	50553	3389	
4	50002	443	45848	443	
...	...	...	...	...	...
65527	63691	80	13237	80	
65528	50964	80	13485	80	
65529	54871	445	0	0	
65530	54870	445	0	0	
65531	54867	445	0	0	

	Action	Bytes	Bytes Sent	Bytes Received	Packets	\
0	allow	177	94	83	2	
1	allow	4768	1600	3168	19	
2	allow	238	118	120	2	
3	allow	3327	1438	1889	15	
4	allow	25358	6778	18580	31	
...	...	...	...	...	...	...
65527	allow	314	192	122	6	
65528	allow	4680740	67312	4613428	4675	
65529	drop	70	70	0	1	
65530	drop	70	70	0	1	
65531	drop	70	70	0	1	

	Elapsed Time (sec)	pkts_sent	pkts_received
0	30	1	1
1	17	10	9
2	1199	1	1
3	17	8	7
4	16	13	18
...	...	...	...
65527	15	4	2
65528	77	985	3690
65529	0	1	0
65530	0	1	0
65531	0	1	0

[65532 rows x 12 columns]

```
In [3]: q1 = df['Destination Port'].quantile(0.25)
q3 = df['Destination Port'].quantile(0.75)
iqr = q3 - q1
lb = q1 - 1.5 * iqr
ub = q3 + 1.5 * iqr
outliers = df[(df['Destination Port'] < lb) | (df['Destination Port'] > ub)]
print("\nOutliers in 'Destination Port':")
print(outliers)

print("Original DataFrame:")
print(df)

q1 = df['Bytes Sent'].quantile(0.25)
q3 = df['Bytes Sent'].quantile(0.75)
iqr = q3 - q1
```

```
lb = q1 - 1.5 * iqr  
ub = q3 + 1.5 * iqr  
  
outliers = df[(df['Bytes Sent'] < lb) | (df['Bytes Sent'] > ub)]  
  
print("\nOutliers in 'Bytes Sent':")  
print(outliers)
```

Outliers in 'Destination Port':

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	\
2	6881	50321	43265	50321	
6	60513	47094	45469	47094	
8	52244	58774	2211	58774	
50	63842	45682	31353	45682	
62	60811	40010	33835	40010	
...	...	...	...	...	
65495	51688	47961	35098	47961	
65508	36226	60038	0	0	
65518	54013	37965	0	0	
65522	53314	64097	0	0	
65526	51710	43069	65147	43069	

	Action	Bytes	Bytes Sent	Bytes Received	Packets	\
2	allow	238	118	120	2	
6	allow	320	140	180	6	
8	allow	70	70	0	1	
50	allow	4687209	3850148	837061	4974	
62	allow	316	136	180	6	
...	...	...	...	...	...	
65495	allow	66	66	0	1	
65508	allow	66	66	0	1	
65518	deny	66	66	0	1	
65522	deny	66	66	0	1	
65526	allow	70	70	0	2	

	Elapsed Time (sec)	pkts_sent	pkts_received
2	1199	1	1
6	7	3	3
8	5	1	0
50	107	3004	1970
62	5	3	3
...	...	...	...
65495	5	1	0
65508	91	1	0
65518	0	1	0
65522	0	1	0
65526	8	2	0

[9043 rows x 12 columns]

Original DataFrame:

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	\
0	57222	53	54587	53	
1	56258	3389	56258	3389	
2	6881	50321	43265	50321	
3	50553	3389	50553	3389	
4	50002	443	45848	443	
...	...	...	...	...	
65527	63691	80	13237	80	
65528	50964	80	13485	80	
65529	54871	445	0	0	
65530	54870	445	0	0	
65531	54867	445	0	0	

	Action	Bytes	Bytes Sent	Bytes Received	Packets	\
0	allow	177	94	83	2	
1	allow	4768	1600	3168	19	
2	allow	238	118	120	2	
3	allow	3327	1438	1889	15	
4	allow	25358	6778	18580	31	
...	...	...	...	...	...	
65527	allow	314	192	122	6	
65528	allow	4680740	67312	4613428	4675	

65529	drop	70	70	0	1
65530	drop	70	70	0	1
65531	drop	70	70	0	1

	Elapsed Time (sec)	pkts_sent	pkts_received
0	30	1	1
1	17	10	9
2	1199	1	1
3	17	8	7
4	16	13	18
...	...	...	...
65527	15	4	2
65528	77	985	3690
65529	0	1	0
65530	0	1	0
65531	0	1	0

[65532 rows x 12 columns]

Outliers in 'Bytes Sent':

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	\
1	56258	3389	56258	3389	
3	50553	3389	50553	3389	
4	50002	443	45848	443	
5	51465	443	39975	443	
7	50049	443	21285	443	
...	...	...	...	...	...
65499	50343	80	49722	80	
65501	50438	3389	50438	3389	
65505	35608	443	62915	443	
65511	58574	443	3429	443	
65528	50964	80	13485	80	

	Action	Bytes	Bytes Sent	Bytes Received	Packets	\
1	allow	4768	1600	3168	19	
3	allow	3327	1438	1889	15	
4	allow	25358	6778	18580	31	
5	allow	3961	1595	2366	21	
7	allow	7912	3269	4643	23	
...	...	...	...	...	...	...
65499	allow	22233	10123	12110	37	
65501	allow	3429	1474	1955	16	
65505	allow	5776	1880	3896	19	
65511	allow	3447	788	2659	13	
65528	allow	4680740	67312	4613428	4675	

	Elapsed Time (sec)	pkts_sent	pkts_received
1	17	10	9
3	17	8	7
4	16	13	18
5	16	12	9
7	96	12	11
...	...	...	...
65499	28	16	21
65501	16	8	8
65505	272	11	8
65511	135	6	7
65528	77	985	3690

[14701 rows x 12 columns]

```
In [13]: import matplotlib.pyplot as plt
d_no = df[df['Bytes Sent'].notnull()]
d_with = df[df['Bytes Sent'].isnull()]
```

```

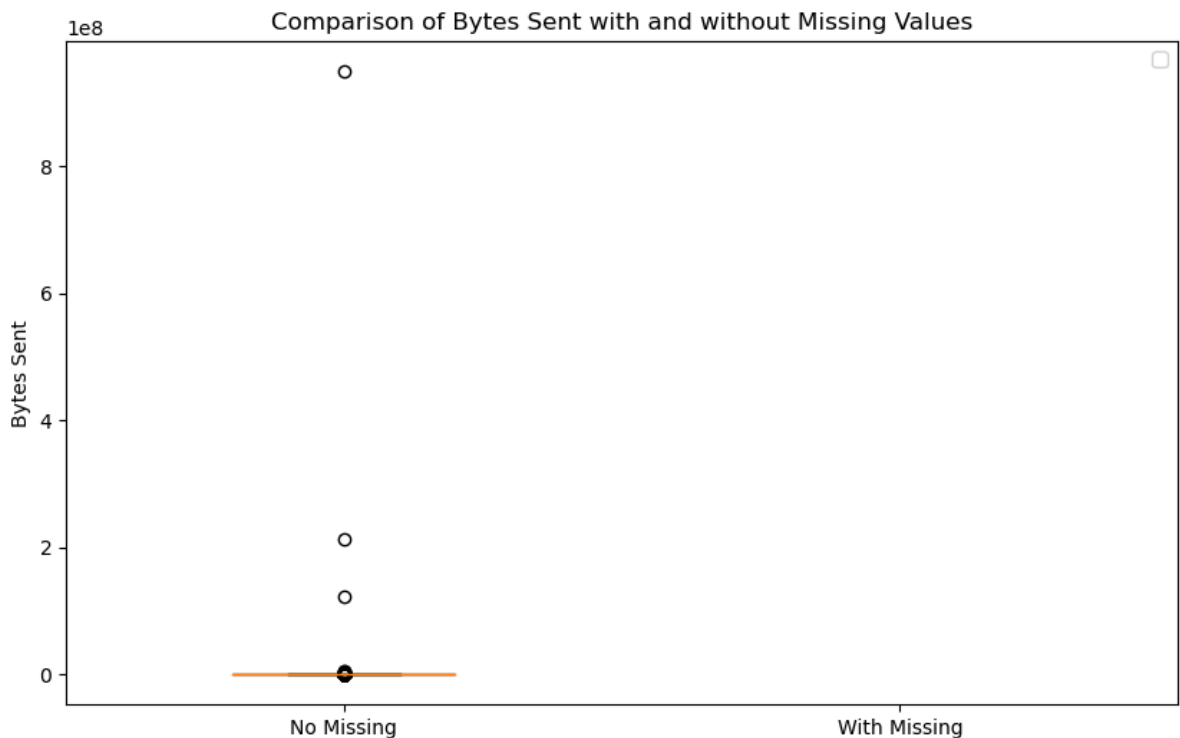
print("\nNumber of rows with missing 'Bytes Sent':", d_with.shape[0])
print("Number of rows without missing 'Bytes Sent':", d_no.shape[0])
plt.figure(figsize=(10, 6))
plt.boxplot(d_no['Bytes Sent'], positions=[1], widths=0.4, patch_artist=True,
            boxprops=dict(facecolor='lightblue'),)
plt.boxplot(d_with['Bytes Sent'], positions=[2], widths=0.4, patch_artist=True,
            boxprops=dict(facecolor='lightcoral'),)
plt.title('Comparison of Bytes Sent with and without Missing Values')
plt.xticks([1, 2], ['No Missing', 'With Missing'])
plt.ylabel('Bytes Sent')
plt.legend()
plt.show()

```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

Number of rows with missing 'Bytes Sent': 0

Number of rows without missing 'Bytes Sent': 65532



```

In [12]: d_no = df[df['Packets'].notnull()]
          d_with = df[df['Packets'].isnull()]

print("\nNumber of rows with missing 'Packets':", d_with.shape[0])
print("Number of rows without missing 'Packets':", d_no.shape[0])

plt.figure(figsize=(10, 6))

plt.boxplot(d_no['Packets'], positions=[1], widths=0.4, patch_artist=True,
            boxprops=dict(facecolor='lightblue'))

plt.boxplot(d_with['Packets'], positions=[2], widths=0.4, patch_artist=True,
            boxprops=dict(facecolor='lightcoral'))

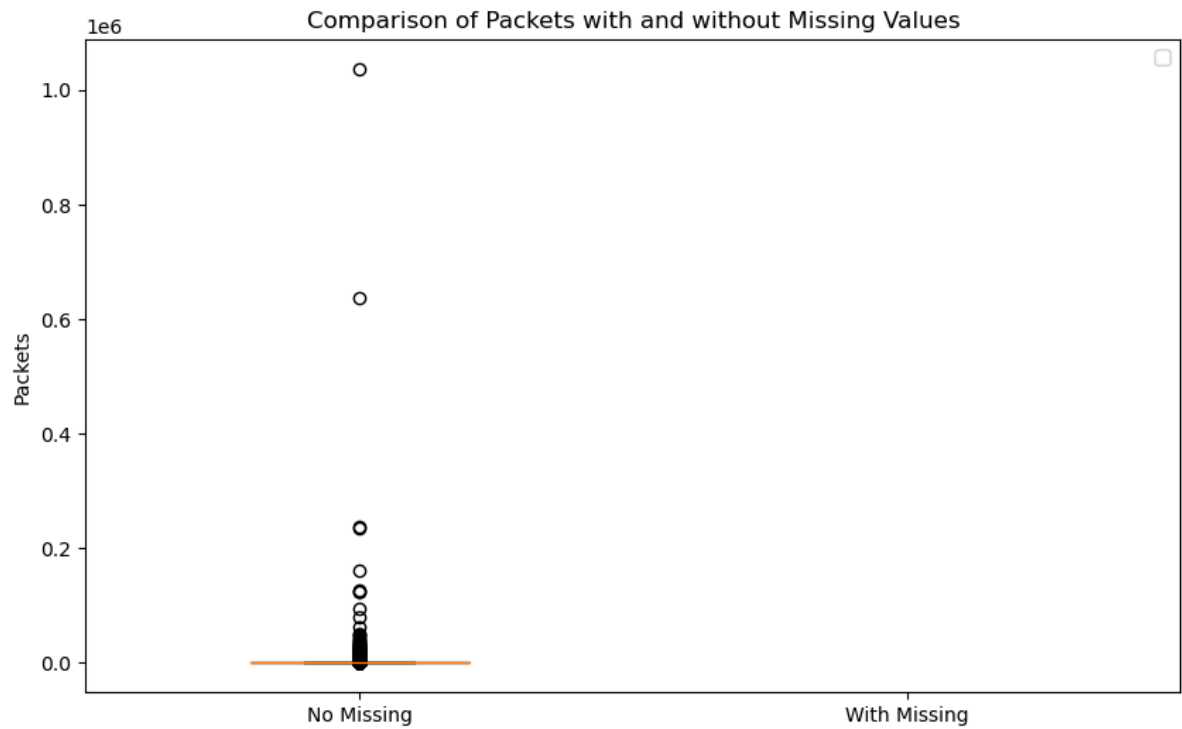
plt.title('Comparison of Packets with and without Missing Values')
plt.xticks([1, 2], ['No Missing', 'With Missing'])
plt.ylabel('Packets')
plt.legend()
plt.show()

```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

Number of rows with missing 'Packets': 0

Number of rows without missing 'Packets': 65532



```
In [16]: d_no = df[df['Elapsed Time (sec)'].notnull()]
d_with = df[df['Elapsed Time (sec)'].isnull()]

print("\nNumber of rows with missing 'Elapsed Time (sec)':", d_with.shape[0])
print("Number of rows without missing 'Elapsed Time (sec)':", d_no.shape[0])

plt.figure(figsize=(10, 6))

plt.boxplot(d_no['Elapsed Time (sec)'], positions=[1], widths=0.4, patch_artist=True,
            boxprops=dict(facecolor='lightblue'), )

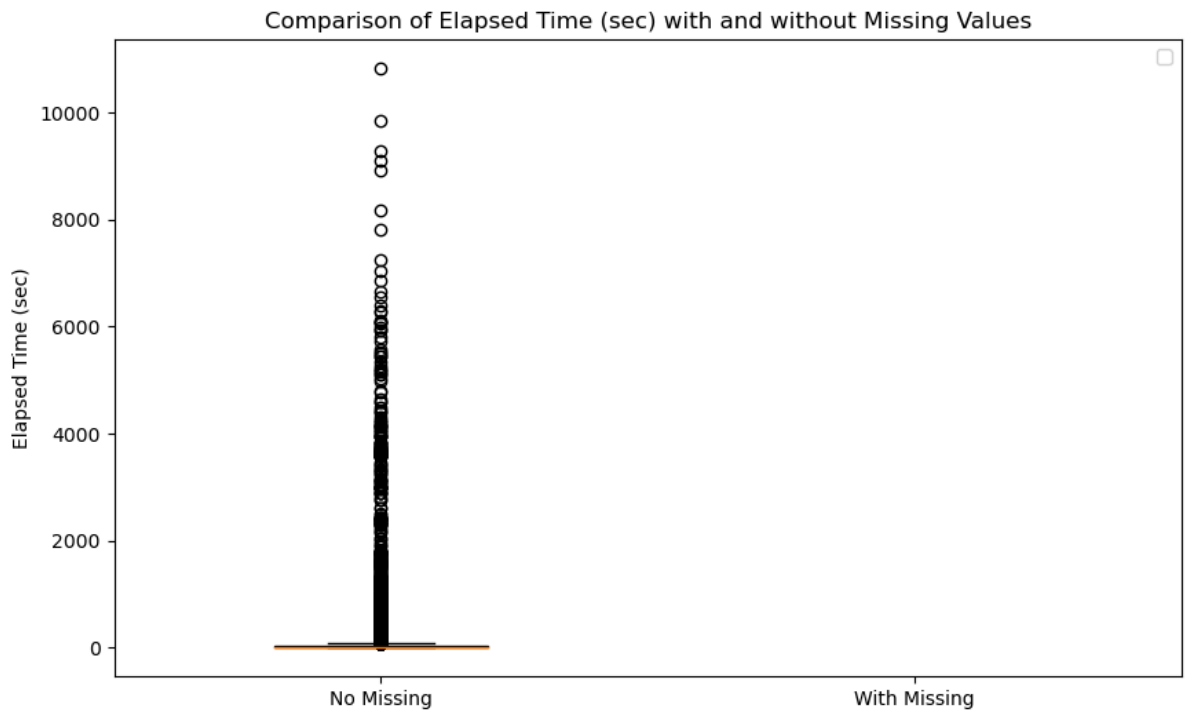
plt.boxplot(d_with['Elapsed Time (sec)'], positions=[2], widths=0.4, patch_artist=True,
            boxprops=dict(facecolor='lightcoral'),)

plt.title('Comparison of Elapsed Time (sec) with and without Missing Values')
plt.xticks([1, 2], ['No Missing', 'With Missing'])
plt.ylabel('Elapsed Time (sec)')
plt.legend()
plt.show()
```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

Number of rows with missing 'Elapsed Time (sec)': 0

Number of rows without missing 'Elapsed Time (sec)': 65532



```
In [19]: d_no = df[df['Elapsed Time (sec)'].notnull()]
d_with = df[df['Elapsed Time (sec)'].isnull()]
print("\nNumber of rows with missing 'Elapsed Time (sec)':", d_with.shape[0])
print("Number of rows without missing 'Elapsed Time (sec)':", d_no.shape[0])

plt.figure(figsize=(10, 6))

plt.boxplot(d_no['Elapsed Time (sec)'], positions=[1], widths=0.4, patch_artist=True,
            boxprops=dict(facecolor='lightblue'), )

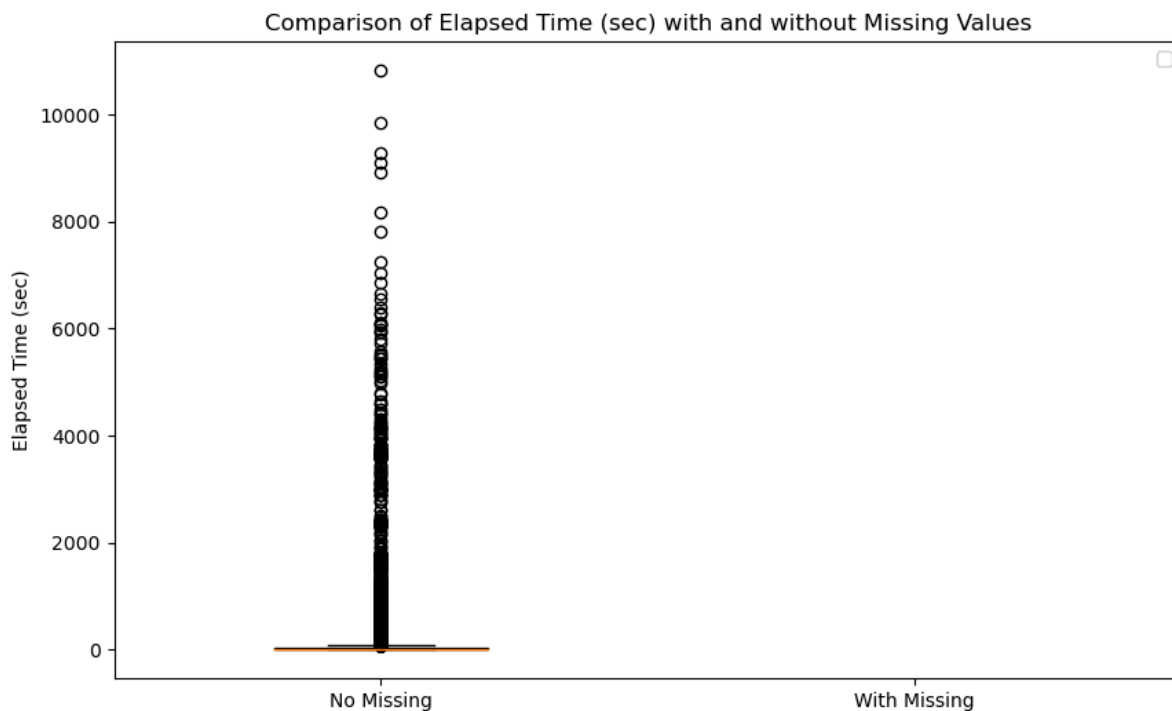
plt.boxplot(d_with['Elapsed Time (sec)'], positions=[2], widths=0.4, patch_artist=True,
            boxprops=dict(facecolor='lightcoral'), )

plt.title('Comparison of Elapsed Time (sec) with and without Missing Values')
plt.xticks([1, 2], ['No Missing', 'With Missing'])
plt.ylabel('Elapsed Time (sec)')
plt.legend()
plt.show()
```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

Number of rows with missing 'Elapsed Time (sec)': 0

Number of rows without missing 'Elapsed Time (sec)': 65532



In [25]: `import numpy as np`

```
c = 'Bytes Sent'
if df[c].isnull().sum() == 0:
    print(f"\nNo missing values in '{c}'. Creating null values...")
for i in range(5):
    df.at[i, c] = np.nan
```

No missing values in 'Bytes Sent'. Creating null values...

In [26]: `d = pd.read_csv(r"D:\log2.csv")`

In [27]: `print("\nDataFrame after creating null values:")`  
`print(d[c].isnull().sum())`  
`print(d)`



DataFrame after creating null values:

0	Source Port	Destination Port	NAT Source Port	NAT Destination Port	\
0	57222	53	54587	53	
1	56258	3389	56258	3389	
2	6881	50321	43265	50321	
3	50553	3389	50553	3389	
4	50002	443	45848	443	
...	...	...	...	...	...
65527	63691	80	13237	80	
65528	50964	80	13485	80	
65529	54871	445	0	0	
65530	54870	445	0	0	
65531	54867	445	0	0	

0	Action	Bytes	Bytes Sent	Bytes Received	Packets	\
0	allow	177	94	83	2	
1	allow	4768	1600	3168	19	
2	allow	238	118	120	2	
3	allow	3327	1438	1889	15	
4	allow	25358	6778	18580	31	
...	...	...	...	...	...	...
65527	allow	314	192	122	6	
65528	allow	4680740	67312	4613428	4675	
65529	drop	70	70	0	1	
65530	drop	70	70	0	1	
65531	drop	70	70	0	1	

0	Elapsed Time (sec)	pkts_sent	pkts_received
0	30	1	1
1	17	10	9
2	1199	1	1
3	17	8	7
4	16	13	18
...	...	...	...
65527	15	4	2
65528	77	985	3690
65529	0	1	0
65530	0	1	0
65531	0	1	0

[65532 rows x 12 columns]

```
In [28]: c = 'Bytes Received'
if d[c].isnull().sum() == 0:
    print(f"\nNo missing values in '{c}'. Creating null values...")
```

No missing values in 'Bytes Received'. Creating null values...

```
In [30]: for i in range(5):
    d.at[i, c] = np.nan
print("\nDataFrame after creating null values:")
print(d[c].isnull().sum())
print(d)
```

DataFrame after creating null values:

```
5
      Source Port  Destination Port  NAT Source Port  NAT Destination Port  \
0          57222           53          54587           53
1          56258           3389          56258           3389
2           6881          50321          43265          50321
3          50553           3389          50553           3389
4          50002           443          45848           443
...          ...          ...          ...          ...
65527        63691           80          13237           80
65528        50964           80          13485           80
65529        54871           445              0           0
65530        54870           445              0           0
65531        54867           445              0           0
```

```
      Action  Bytes  Bytes Sent  Bytes Received  Packets  \
0      allow    177          94             NaN         2
1      allow   4768         1600             NaN        19
2      allow    238          118             NaN         2
3      allow   3327         1438             NaN        15
4      allow  25358         6778             NaN        31
...      ...     ...          ...          ...     ...
65527  allow    314          192             122.0         6
65528  allow  4680740        67312        4613428.0       4675
65529  drop      70           70              0.0         1
65530  drop      70           70              0.0         1
65531  drop      70           70              0.0         1
```

```
      Elapsed Time (sec)  pkts_sent  pkts_received
0              30           1           1
1              17          10           9
2             1199           1           1
3              17           8           7
4              16          13          18
...           ...          ...          ...
65527           15           4           2
65528           77          985        3690
65529           0           1           0
65530           0           1           0
65531           0           1           0
```

[65532 rows x 12 columns]

```
In [31]: c = 'Packets'
if d[c].isnull().sum() == 0:
    print(f"\nNo missing values in '{c}'. Creating null values...")
for i in range(5):
    d.at[i, c] = np.nan
print("\nDataFrame after creating null values:")
```

No missing values in 'Packets'. Creating null values...

DataFrame after creating null values:

```
In [33]: print(d[c].isnull().sum())
print(d)
```

5	Source Port	Destination Port	NAT Source Port	NAT Destination Port	\
0	57222	53	54587	53	
1	56258	3389	56258	3389	
2	6881	50321	43265	50321	
3	50553	3389	50553	3389	
4	50002	443	45848	443	
...	...	...	...	...	
65527	63691	80	13237	80	
65528	50964	80	13485	80	
65529	54871	445	0	0	
65530	54870	445	0	0	
65531	54867	445	0	0	

	Action	Bytes	Bytes Sent	Bytes Received	Packets	\
0	allow	177	94	NaN	NaN	
1	allow	4768	1600	NaN	NaN	
2	allow	238	118	NaN	NaN	
3	allow	3327	1438	NaN	NaN	
4	allow	25358	6778	NaN	NaN	
...	...	...	...	...	...	
65527	allow	314	192	122.0	6.0	
65528	allow	4680740	67312	4613428.0	4675.0	
65529	drop	70	70	0.0	1.0	
65530	drop	70	70	0.0	1.0	
65531	drop	70	70	0.0	1.0	

	Elapsed Time (sec)	pkts_sent	pkts_received
0	30	1	1
1	17	10	9
2	1199	1	1
3	17	8	7
4	16	13	18
...	...	...	...
65527	15	4	2
65528	77	985	3690
65529	0	1	0
65530	0	1	0
65531	0	1	0

[65532 rows x 12 columns]

```
In [35]: from sklearn.impute import SimpleImputer
c = 'Packets'
imputer = SimpleImputer(strategy='mean')
print(f"\nMissing values in '{c}' before imputation:", d[c].isnull().sum())
d[c] = imputer.fit_transform(d[[c]])
print(f"\nMissing values in '{c}' after imputation:", d[c].isnull().sum())
```

Missing values in 'Packets' before imputation: 5

Missing values in 'Packets' after imputation: 0

```
In [36]: print(d.isnull().sum())
```

Source Port	0
Destination Port	0
NAT Source Port	0
NAT Destination Port	0
Action	0
Bytes	0
Bytes Sent	0
Bytes Received	5
Packets	0
Elapsed Time (sec)	0
pkts_sent	0
pkts_received	0
dtype:	int64

```
In [37]: d_deleted = d.dropna()
```

```
In [38]: print("\nShape of original DataFrame:", d.shape)
print("Shape of DataFrame after deletion:", d_deleted.shape)
```

```
Shape of original DataFrame: (65532, 12)
Shape of DataFrame after deletion: (65527, 12)
```

```
In [40]: c='Bytes Received'
imputer = SimpleImputer(strategy='mean')
d[c] = imputer.fit_transform(d[[c]])
print(f"\nMissing values in '{c}' after imputation:", d[c].isnull().sum())
```

```
Missing values in 'Bytes Received' after imputation: 0
```

```
In [41]: print(f"Missing values in '{c}':", d[c].isnull().sum())
```

```
Missing values in 'Bytes Received': 0
```

```
In [ ]:
```