INDEX

LIST OF TABLES

LIST OF FIGURES

Problem 1: Linear Regression

The comp-activ databases is a collection of a computer systems activity measures . The data was collected from a Sun Sparcstation 20/712 with 128 Mbytes of memory running in a multi-user university department. Users would typically be doing a large variety of tasks ranging from accessing the internet, editing files or running very cpu-bound programs.

As you are a budding data scientist you thought to find out a linear equation to build a model to predict 'usr'(Portion of time (%) that cpus run in user mode) and to find out how each attribute affects the system to be in 'usr' mode using a list of system attributes.

DATA DICTIONARY:

System measures used:

lread - Reads (transfers per second) between system memory and user memory

lwrite - writes (transfers per second) between system memory and user memory

scall - Number of system calls of all types per second

sread - Number of system read calls per second .

swrite - Number of system write calls per second .

fork - Number of system fork calls per second.

exec - Number of system exec calls per second.

rchar - Number of characters transferred per second by system read calls

wchar - Number of characters transfreed per second by system write calls

pgout - Number of page out requests per second

ppgout - Number of pages, paged out per second

pgfree - Number of pages per second placed on the free list.

pgscan - Number of pages checked if they can be freed per second

atch - Number of page attaches (satisfying a page fault by reclaiming a page in memory) per second

pgin - Number of page-in requests per second

ppgin - Number of pages paged in per second

pflt - Number of page faults caused by protection errors (copy-on-writes).

vflt - Number of page faults caused by address translation .

runqsz - Process run queue size (The number of kernel threads in memory that are waiting for a CPU

to run.

Typically, this value should be less than 2. Consistently higher values mean that the system might be CPU-bound.)

freemem - Number of memory pages available to user processes

freeswap - Number of disk blocks available for page swapping.

---

usr - Portion of time (%) that cpus run in user mode

1.1 Read the data and do exploratory data analysis. Describe the data briefly. (Check the Data types, shape, EDA, 5 point summary). Perform Univariate, Bivariate Analysis, Multivariate Analysis.

Import some libraries like Numpy, Pandas, Seaborn, Matplotlib, Linear Regression machine learning like LinearRegression, Ridge, Lasso, LinearDiscriminantAnalysis, confusion_matrix, StandardScaler, train_test_split etc. After that, load our data set, compactiv.xlsx, and use the head() function to view the Top 10 data and the tail() function to view the bottom 10 data. Using the shape function, we can determine that there are 8192 rows and 22 columns. Find out the characteristics of the columns using the info() method. The datatypes for the float64(13), int64(8), and object(1) columns are present. There are some null values, but there aren't any duplicate values.

- **head()** it given by default top five data

lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
1	0	2147	79	68	0.2	0.2	40671.0	53995.0	0.0	...	0.0	0.0	1.6	2.6	16.00	26.40	CPU_Bound	4670	1730946	95
0	0	170	18	21	0.2	0.2	448.0	8385.0	0.0	...	0.0	0.0	0.0	0.0	15.63	16.83	Not_CPU_Bound	7278	1869002	97
15	3	2162	159	119	2.0	2.4	NaN	31950.0	0.0	...	0.0	1.2	6.0	9.4	150.20	220.20	Not_CPU_Bound	702	1021237	87
0	0	160	12	16	0.2	0.2	NaN	8670.0	0.0	...	0.0	0.0	0.2	0.2	15.60	16.80	Not_CPU_Bound	7248	1863704	98
5	1	330	39	38	0.4	0.4	NaN	12185.0	0.0	...	0.0	0.0	1.0	1.2	37.80	47.60	Not_CPU_Bound	633	1760253	90
0	0	1201	65	61	0.4	0.4	NaN	58703.0	0.0	...	0.0	0.0	0.0	0.0	28.40	34.40	Not_CPU_Bound	6854	1877461	96
1	0	5744	168	190	0.2	0.2	NaN	189975.0	6.0	...	0.0	4.4	0.6	0.6	27.40	28.60	Not_CPU_Bound	312	1013458	89
21	18	2799	291	211	0.6	0.4	NaN	259868.0	2.6	...	0.0	0.0	1.0	1.0	35.40	71.00	CPU_Bound	87	13	0
0	0	264	42	33	0.2	0.2	NaN	10116.0	0.0	...	0.0	0.0	0.4	0.8	15.63	18.44	Not_CPU_Bound	1374	1749756	98
0	0	188	13	24	0.2	0.2	NaN	6777.0	0.0	...	0.0	0.0	0.0	0.0	15.60	16.80	Not_CPU_Bound	5310	1859912	98

Fig 1 : Top ten data of dataset comp_activ

- **tail()** it given by default bottom five data

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	fre
8182	10	0	5975	692	612	2.20	2.00	505718.0	475455.0	5.99	...	122.55	0.40	24.15	72.85	95.21	230.54	Not_CPU_Bound	126	1(
8183	4	3	1509	501	139	0.60	1.00	312242.0	255308.0	0.00	...	0.00	0.00	3.20	3.20	52.60	119.60	CPU_Bound	1051	1(
8184	8	2	4784	416	240	1.20	1.60	442876.0	110453.0	7.82	...	129.06	0.40	35.27	48.70	106.81	329.86	CPU_Bound	205	1(
8185	13	0	4279	235	176	12.22	34.47	289309.0	30701.0	0.40	...	7.01	8.82	4.21	4.21	397.60	751.30	Not_CPU_Bound	184	1(
8186	0	0	300	56	46	0.20	0.20	1995.0	18052.0	0.80	...	0.00	0.00	0.00	0.00	21.00	18.00	Not_CPU_Bound	272	1(
8187	16	12	3009	360	244	1.60	5.81	405250.0	85282.0	8.02	...	55.11	0.60	35.87	47.90	139.28	270.74	CPU_Bound	387	9
8188	4	0	1596	170	146	2.40	1.80	89489.0	41764.0	3.80	...	0.20	0.80	3.80	4.40	122.40	212.60	Not_CPU_Bound	263	1(
8189	16	5	3116	289	190	0.60	0.60	325948.0	52640.0	0.40	...	0.00	0.40	28.40	45.20	60.20	219.80	Not_CPU_Bound	400	9
8190	32	45	5180	254	179	1.20	1.20	62571.0	29505.0	1.40	...	18.04	0.40	23.05	24.25	93.19	202.81	CPU_Bound	141	1(
8191	2	0	985	55	46	1.60	4.80	111111.0	22256.0	0.00	...	0.00	0.20	3.40	6.20	91.80	110.00	CPU_Bound	659	1(

Fig: 2. Bottom five data of dataset comp_activ

- **shape** it tells numbers of rows and columns in given dataset.

```
(8192, 22)
```

Fig 3 : 8190 rows & 22 columns

- **info()** it tells a concise summary of a DataFrame

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8192 entries, 0 to 8191
Data columns (total 22 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   lread     8192 non-null   int64
 1   lwrite    8192 non-null   int64
 2   scall     8192 non-null   int64
 3   sread     8192 non-null   int64
 4   swrite    8192 non-null   int64
 5   fork      8192 non-null   float64
 6   exec      8192 non-null   float64
 7   rchar     8088 non-null   float64
 8   wchar     8177 non-null   float64
 9   pgout     8192 non-null   float64
 10  ppgout    8192 non-null   float64
 11  pgfree    8192 non-null   float64
 12  pgscan    8192 non-null   float64
 13  atch      8192 non-null   float64
 14  pgin      8192 non-null   float64
 15  ppgin     8192 non-null   float64
 16  pflt      8192 non-null   float64
 17  vflt      8192 non-null   float64
 18  runqsz    8192 non-null   object
 19  freemem   8192 non-null   int64
 20  freeswap  8192 non-null   int64
 21  usr       8192 non-null   int64
dtypes: float64(13), int64(8), object(1)
memory usage: 1.4+ MB
```

Fig 4 : Information about the comp_activ structure and content.

- No Duplicates Values.
- rchar and wchar have Null Values.

```
lread           0
lwrite          0
scall           0
sread           0
swrite          0
fork            0
exec            0
rchar         104
wchar          15
pgout           0
ppgout          0
pgfree          0
pgscan          0
atch            0
pgin            0
ppgin           0
pflt            0
vflt            0
runqsz          0
freemem         0
freeswap        0
usr             0
dtype: int64
```

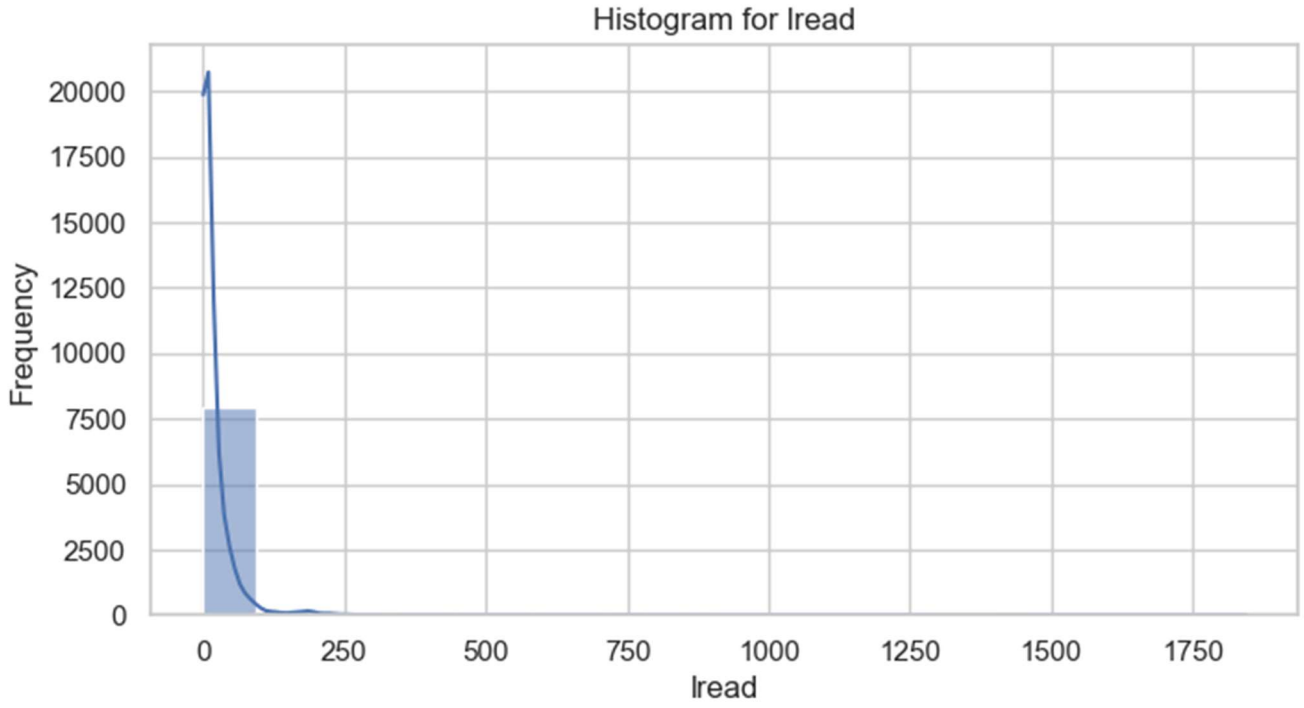
Fig 5 : Null values of comp_activ.

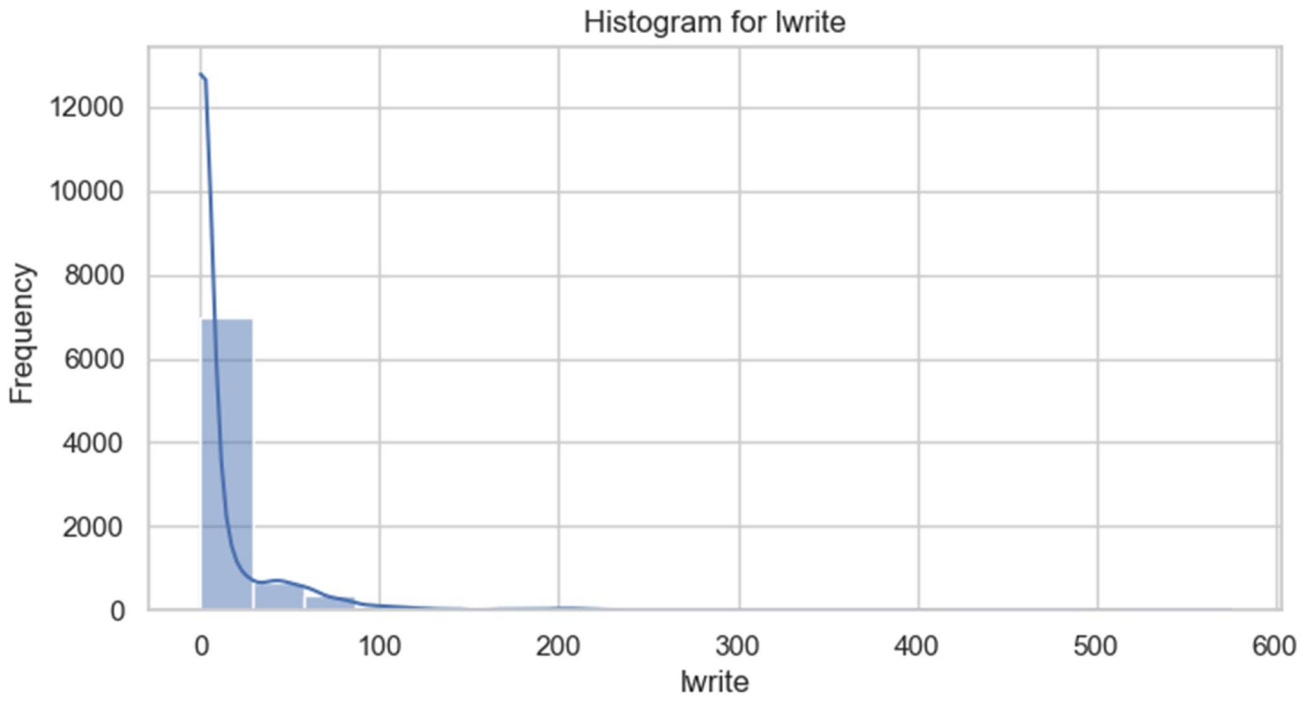
- **Describe()** it tells summary of the central tendency, dispersion, and shape of the distribution of the data.

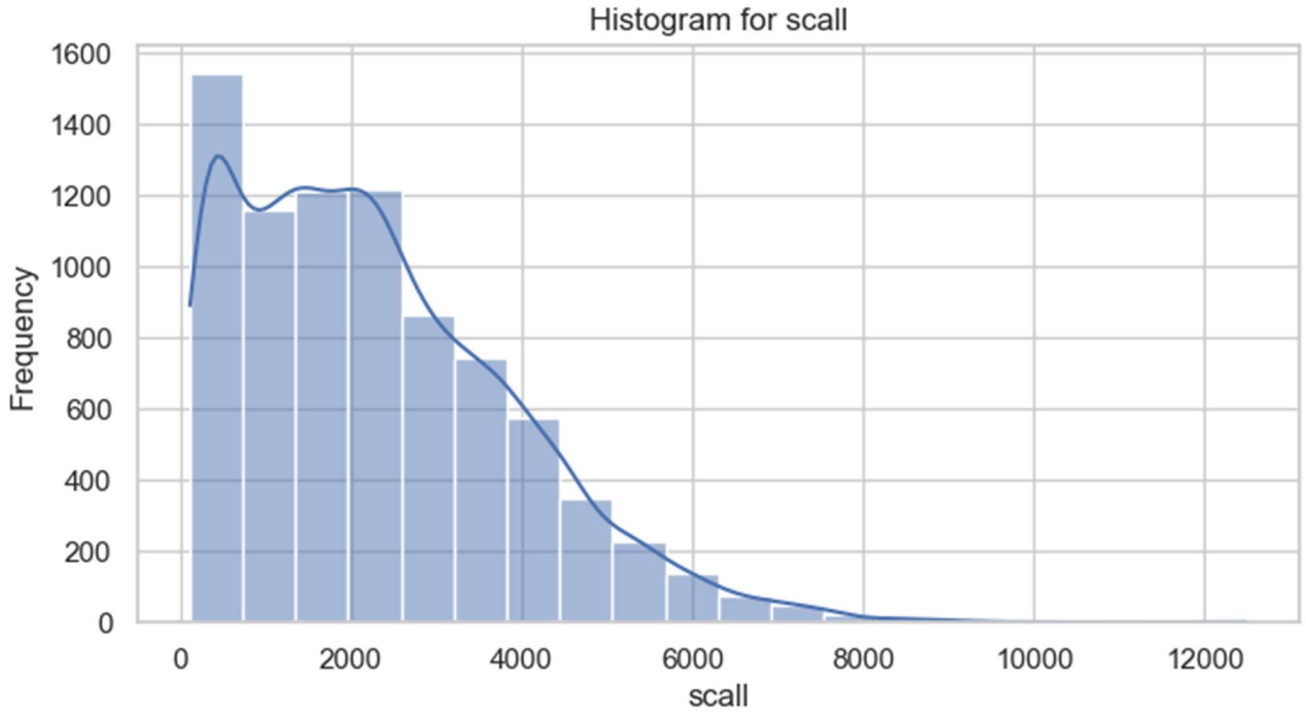
	count	mean	std	min	25%	50%	75%	max
lread	8192.0	1.955969e+01	53.353799	0.0	2.0	7.0	20.000	1845.00
lwrite	8192.0	1.310620e+01	29.891726	0.0	0.0	1.0	10.000	575.00
scall	8192.0	2.306318e+03	1633.617322	109.0	1012.0	2051.5	3317.250	12493.00
sread	8192.0	2.104800e+02	198.980146	6.0	86.0	166.0	279.000	5318.00
swrite	8192.0	1.500582e+02	160.478980	7.0	63.0	117.0	185.000	5456.00
fork	8192.0	1.884554e+00	2.479493	0.0	0.4	0.8	2.200	20.12
exec	8192.0	2.791998e+00	5.212456	0.0	0.2	1.2	2.800	59.56
rchar	8088.0	1.973857e+05	239837.493526	278.0	34091.5	125473.5	267828.750	2526649.00
wchar	8177.0	9.590299e+04	140841.707911	1498.0	22916.0	46619.0	106101.000	1801623.00
pgout	8192.0	2.285317e+00	5.307038	0.0	0.0	0.0	2.400	81.44
ppgout	8192.0	5.977229e+00	15.214590	0.0	0.0	0.0	4.200	184.20
pgfree	8192.0	1.191971e+01	32.363520	0.0	0.0	0.0	5.000	523.00
pgscan	8192.0	2.152685e+01	71.141340	0.0	0.0	0.0	0.000	1237.00
atch	8192.0	1.127505e+00	5.708347	0.0	0.0	0.0	0.600	211.58
pgin	8192.0	8.277960e+00	13.874978	0.0	0.6	2.8	9.765	141.20
ppgin	8192.0	1.238859e+01	22.281318	0.0	0.6	3.8	13.800	292.61
pflt	8192.0	1.097938e+02	114.419221	0.0	25.0	63.8	159.600	899.80
vflt	8192.0	1.853158e+02	191.000603	0.2	45.4	120.4	251.800	1365.00
freemem	8192.0	1.763456e+03	2482.104511	55.0	231.0	579.0	2002.250	12027.00
freeswap	8192.0	1.328126e+06	422019.426957	2.0	1042623.5	1289289.5	1730379.500	2243187.00
usr	8192.0	8.396887e+01	18.401905	0.0	81.0	89.0	94.000	99.00

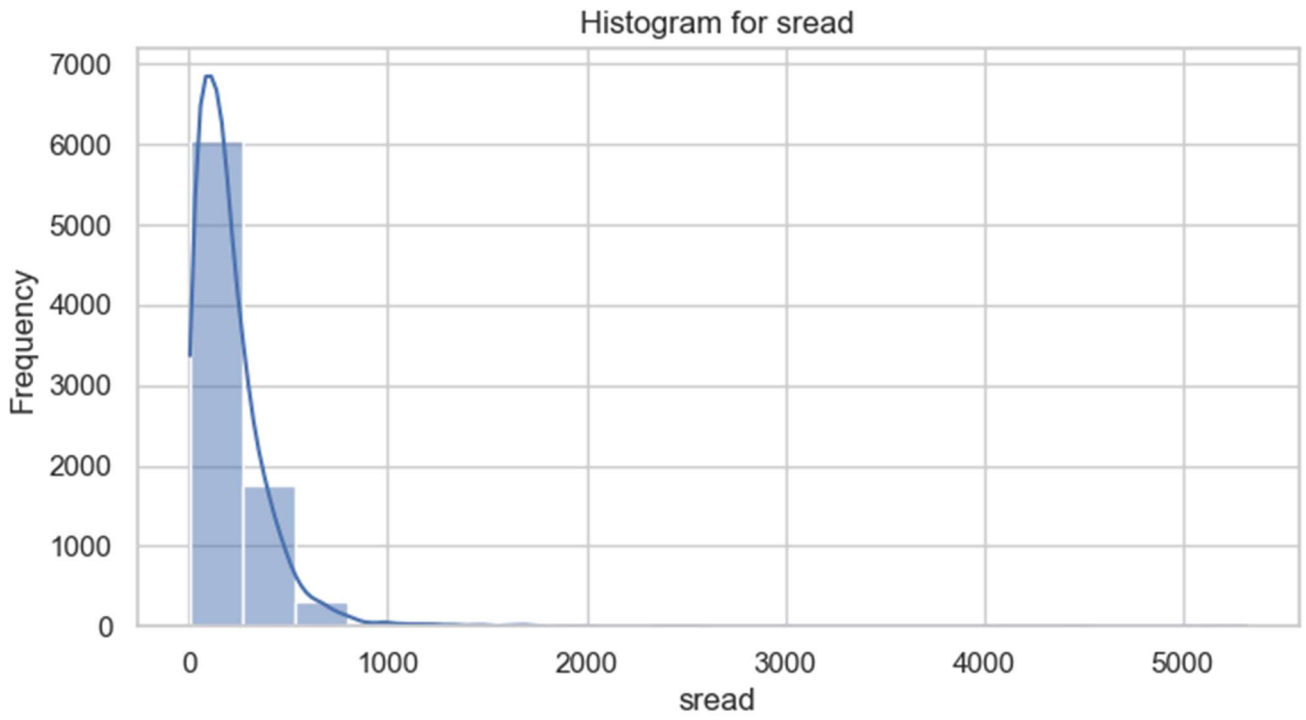
Fig 6 : Descriptive statistics of comp_activ

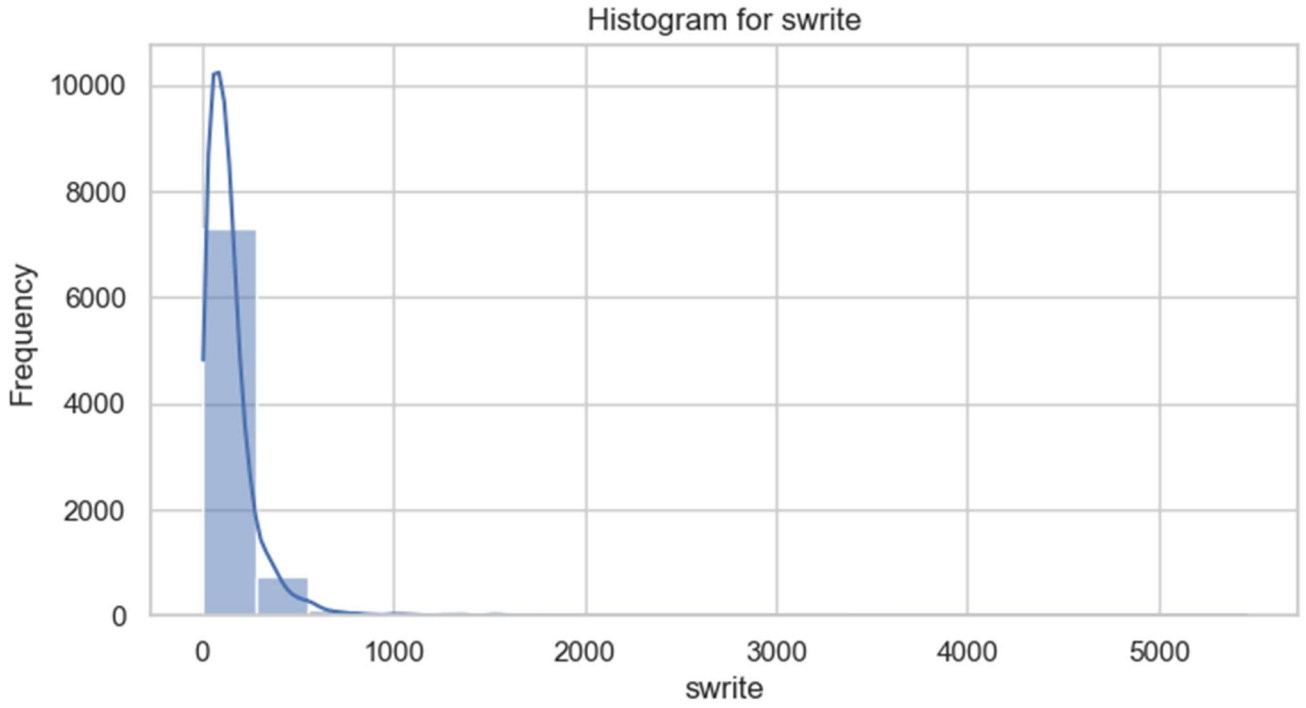
- **Histograms for numerical columns**

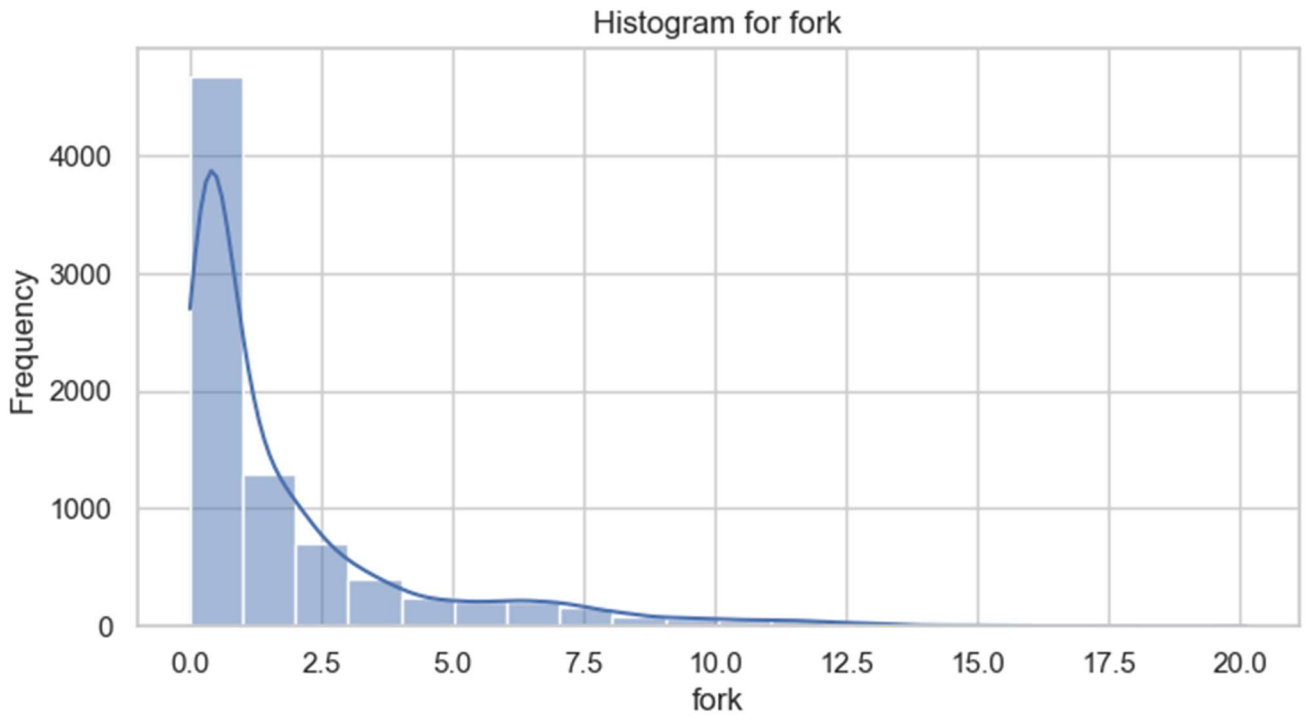


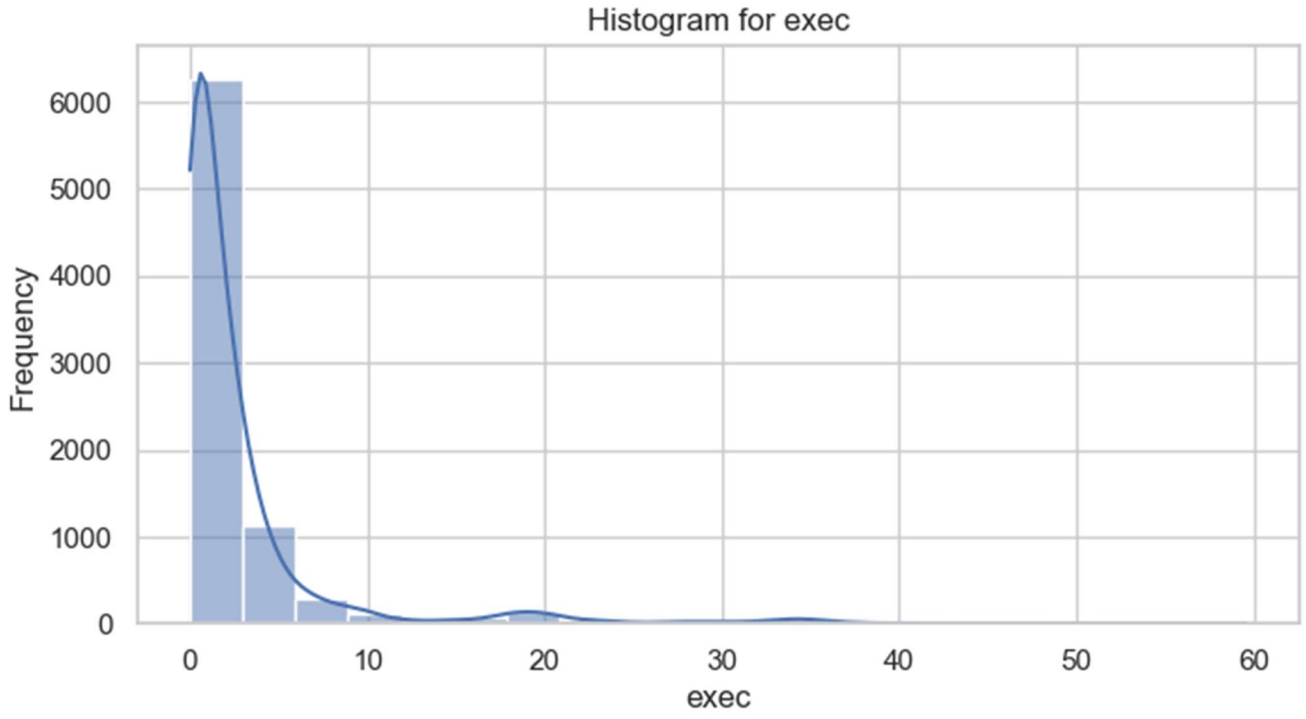

Histogram for lwrite
Frequency
12000
10000
8000
6000
4000
2000
0
0
100
200
300
400
500
600
lwrite

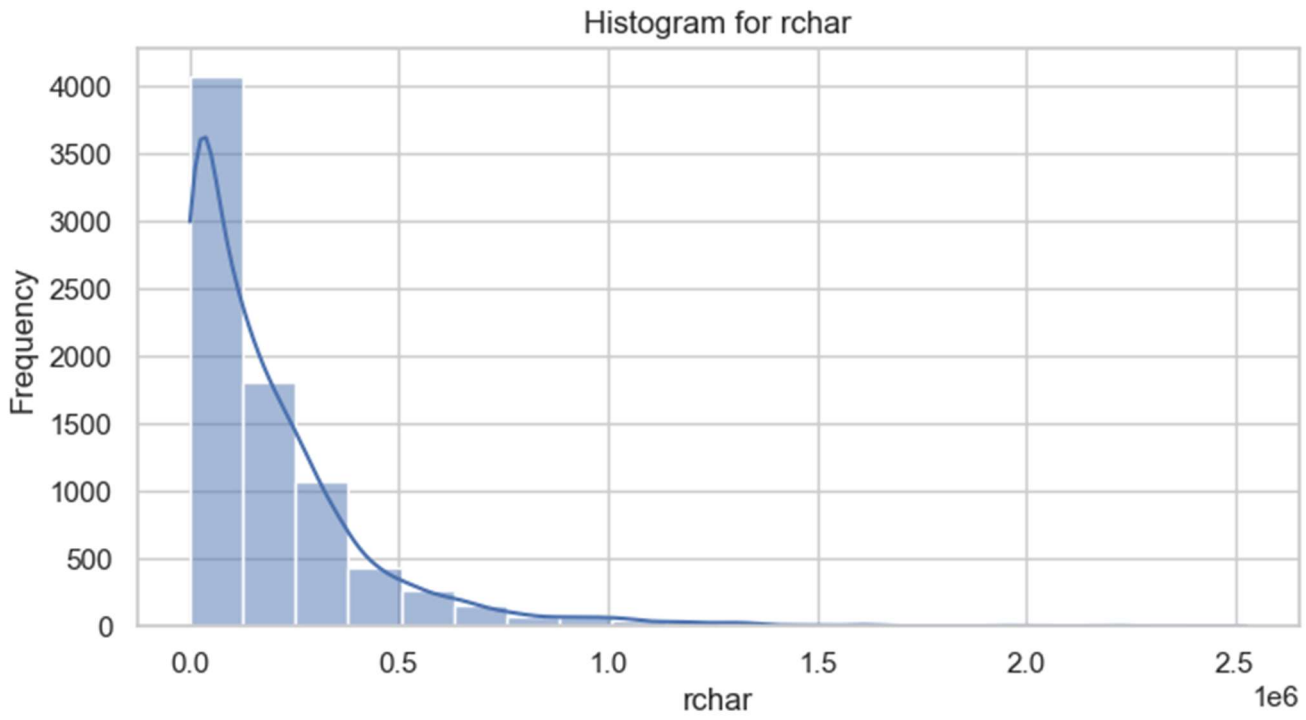

Histogram for scall
Frequency
1600
1400
1200
1000
800
600
400
200
0
0
2000
4000
6000
8000
10000
12000
scall

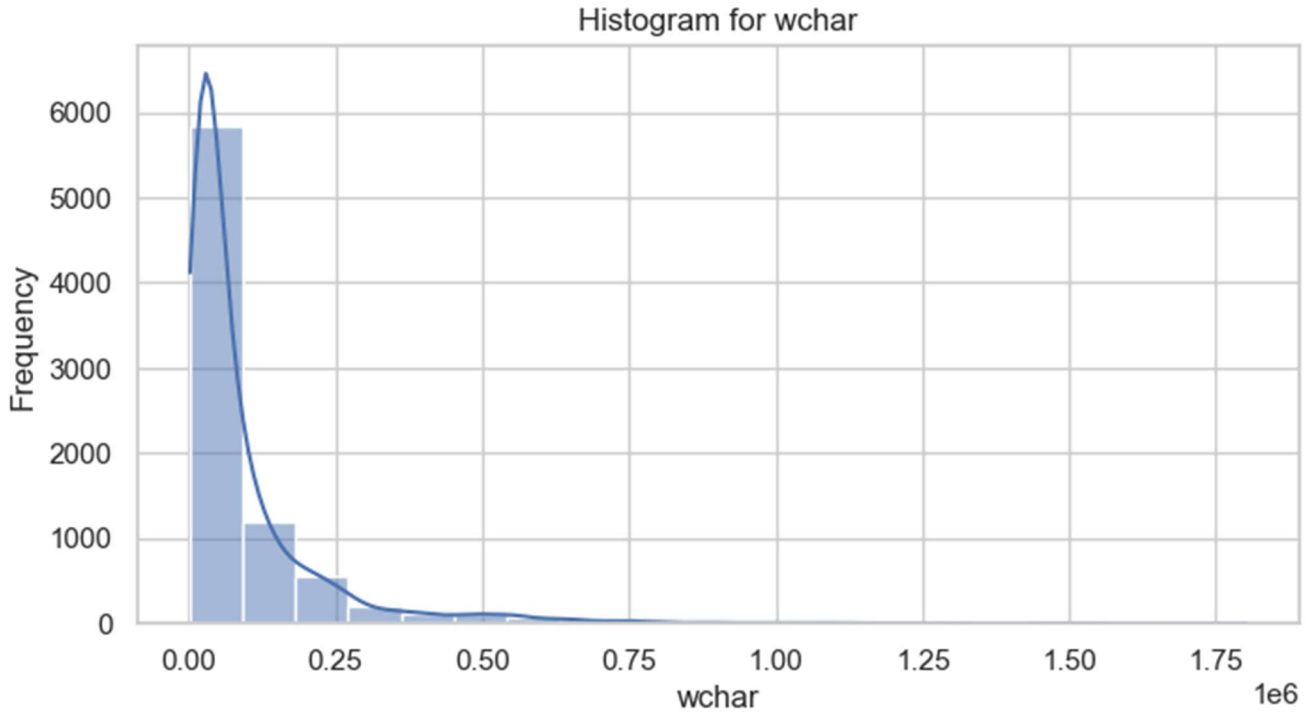

Histogram for sread
Frequency
7000
6000
5000
4000
3000
2000
1000
0
0
1000
2000
3000
4000
5000
sread

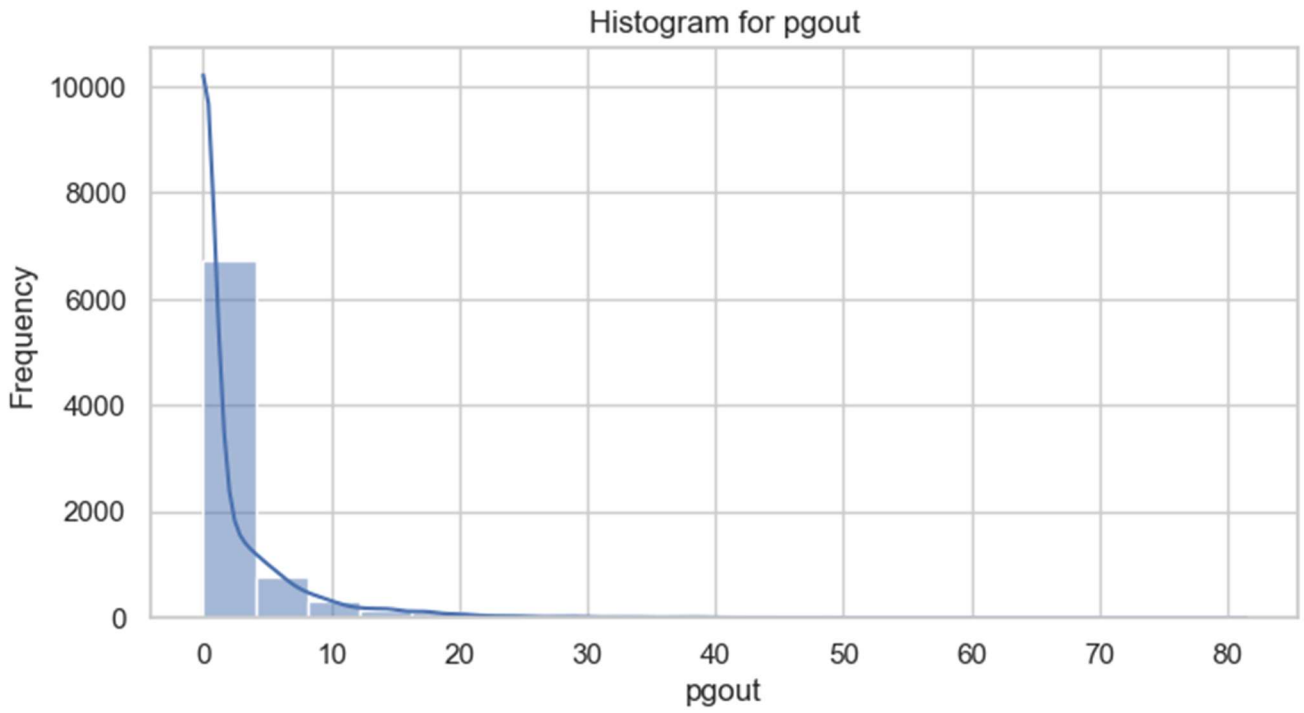

Histogram for swrite
Frequency
10000
8000
6000
4000
2000
0
0
1000
2000
3000
4000
5000
swrite

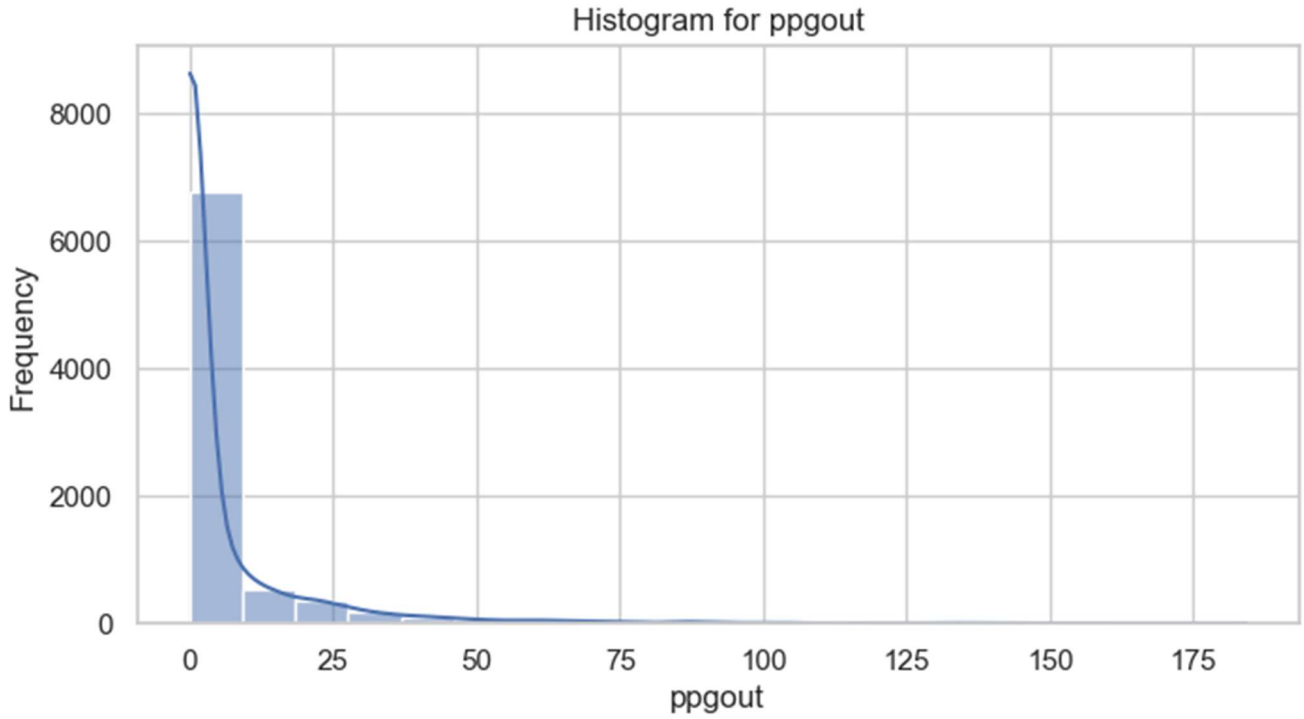

Histogram for fork
Frequency
4000
3000
2000
1000
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
fork

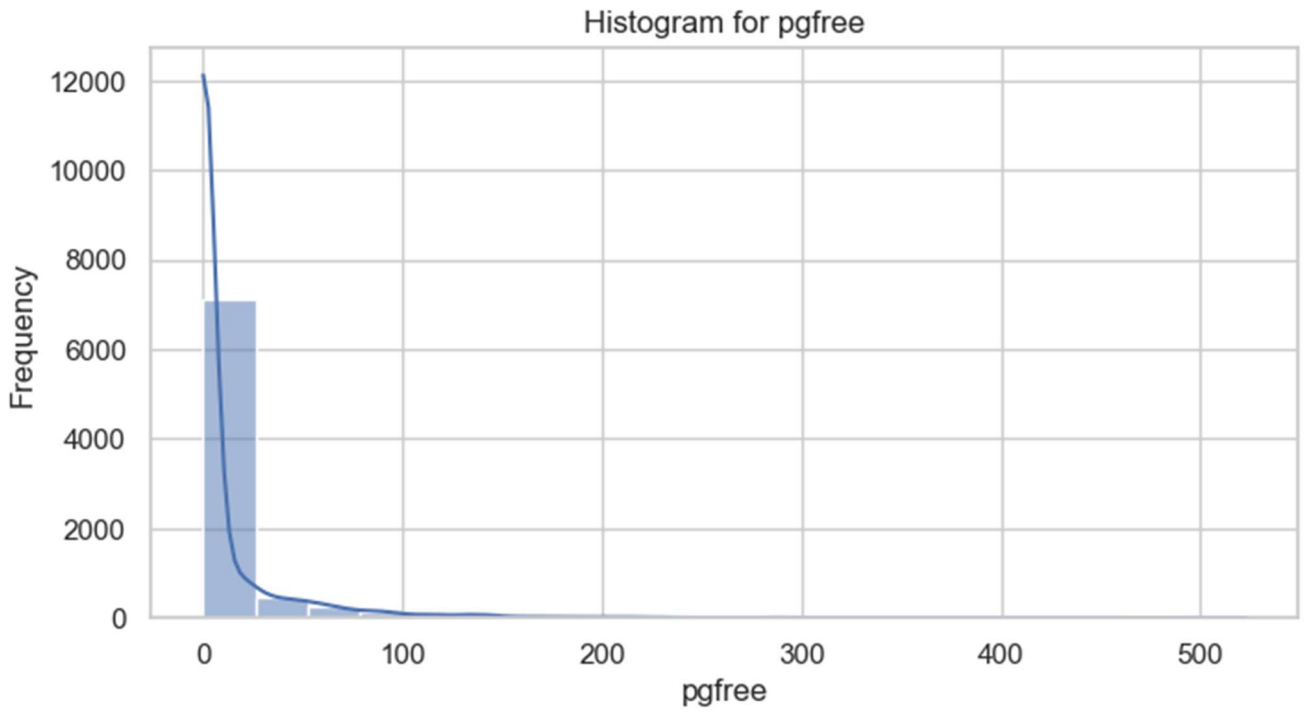

Histogram for exec
Frequency
6000
5000
4000
3000
2000
1000
0
0
10
20
30
40
50
60
exec

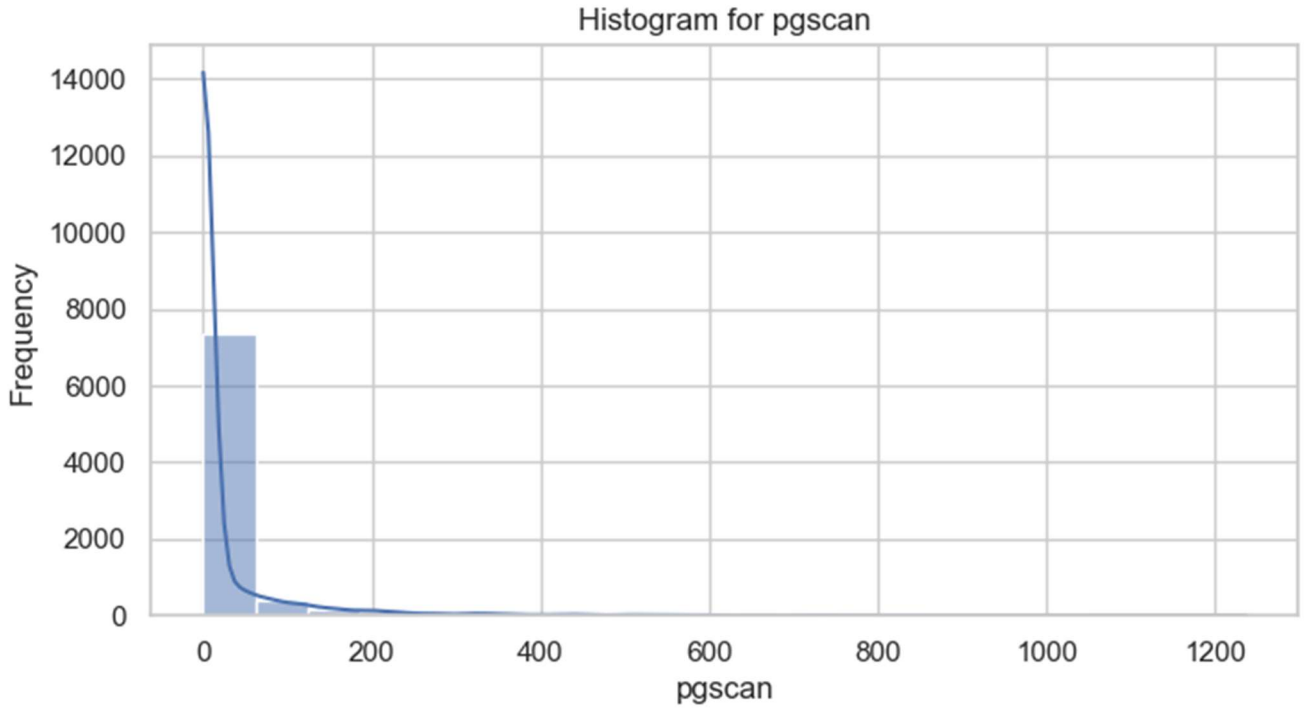

Histogram for rchar
Frequency
4000
3500
3000
2500
2000
1500
1000
500
0
0.0
0.5
1.0
1.5
2.0
2.5
rchar
1e6

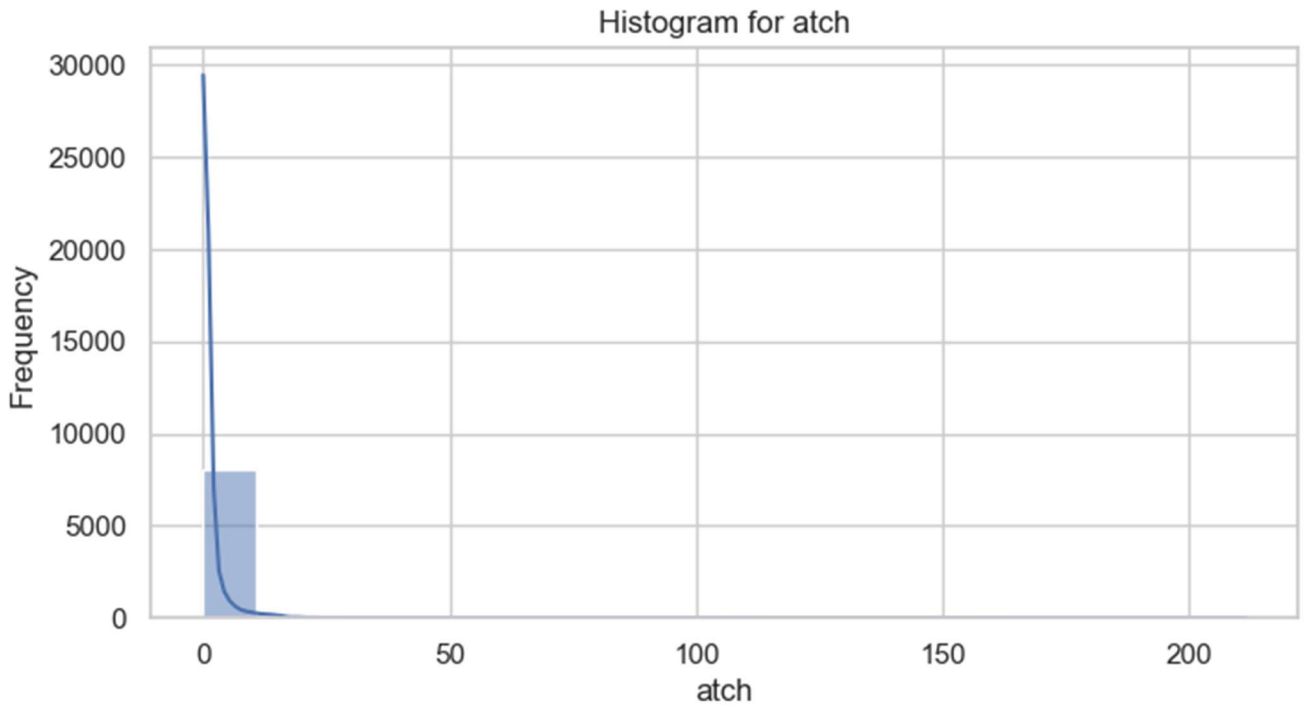

Histogram for wchar
Frequency
6000
5000
4000
3000
2000
1000
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
wchar
1e6

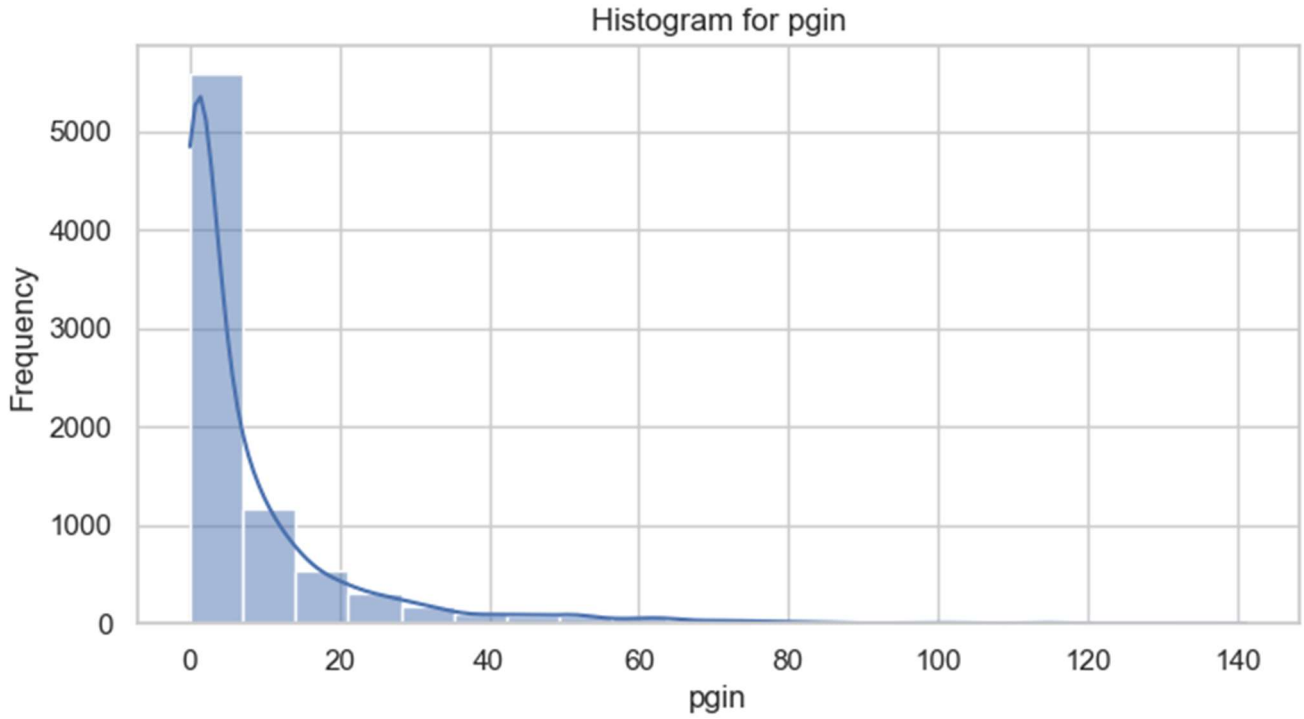

Histogram for pgout
Frequency
10000
8000
6000
4000
2000
0
0
10
20
30
40
50
60
70
80
pgout

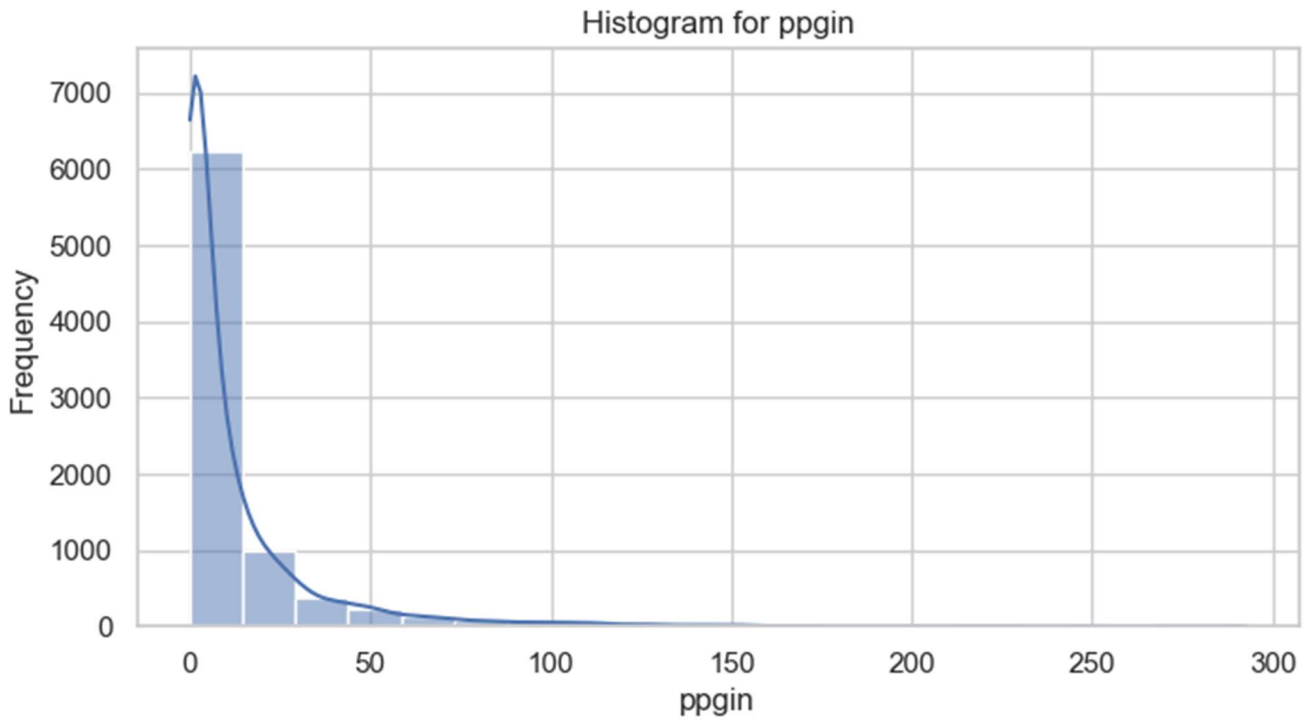

Histogram for ppgout
Frequency
8000
6000
4000
2000
0
0
25
50
75
100
125
150
175
ppgout

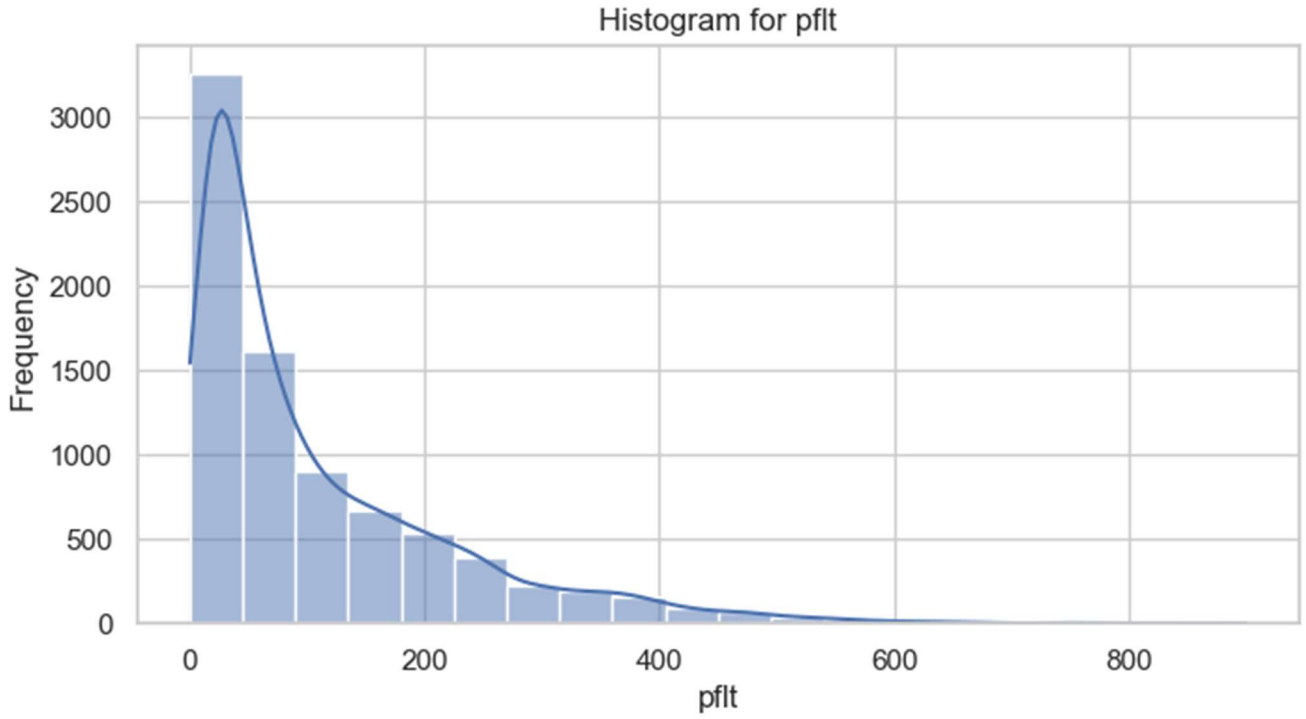

Histogram for pgfree
Frequency
12000
10000
8000
6000
4000
2000
0
0
100
200
300
400
500
pgfree

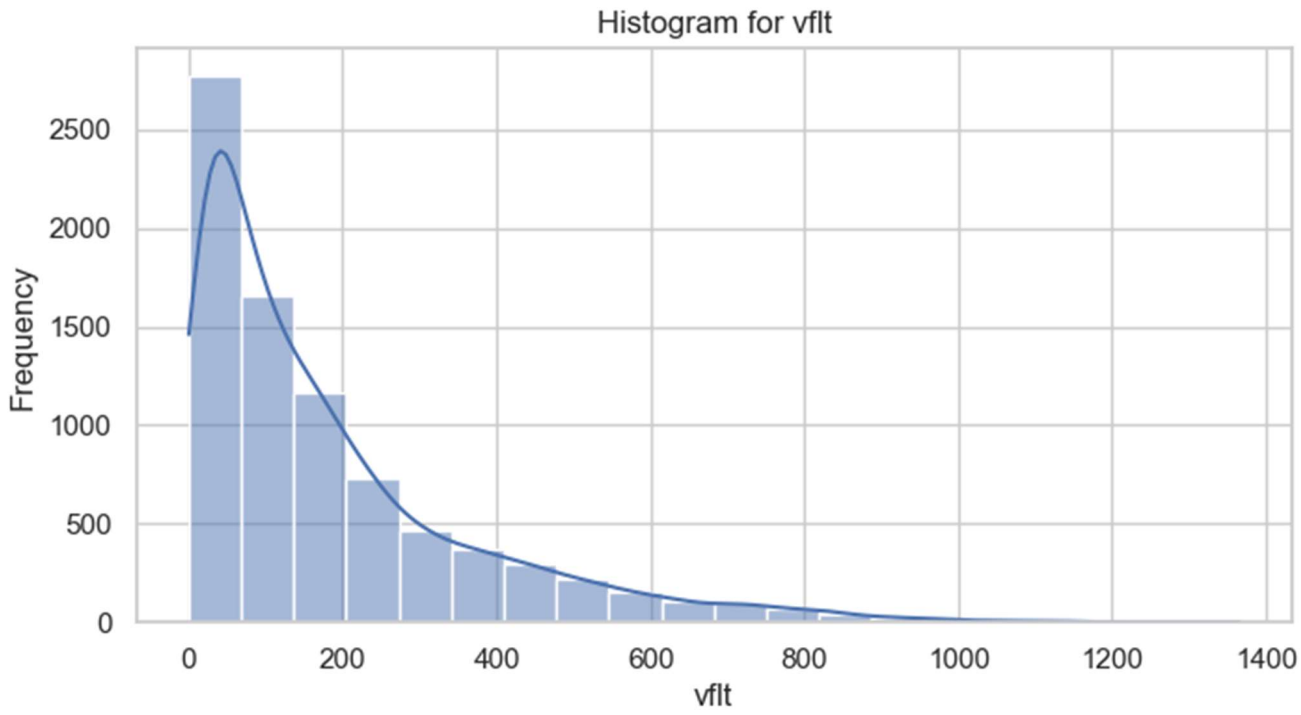

Histogram for pgscan
Frequency
14000
12000
10000
8000
6000
4000
2000
0
0
200
400
600
800
1000
1200
pgscan

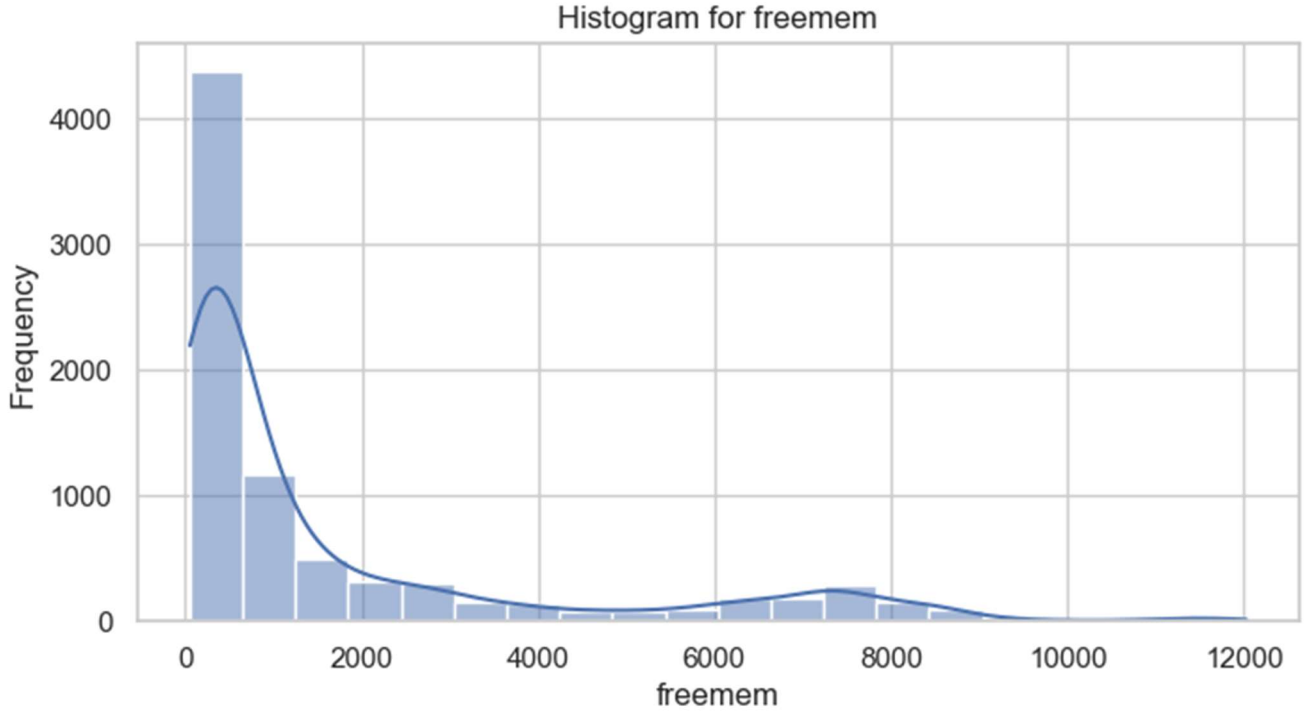

Histogram for atch
Frequency
30000
25000
20000
15000
10000
5000
0
0
50
100
150
200
atch


Histogram for pgin
Frequency
5000
4000
3000
2000
1000
0
0
20
40
60
80
100
120
140
pgin


Histogram for ppgin
Frequency
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300
ppgin


Histogram for pflt
Frequency
3000
2500
2000
1500
1000
500
0
0
200
400
600
800
pflt


Histogram for vflt
Frequency
2500
2000
1500
1000
500
0
0
200
400
600
800
1000
1200
1400
vflt


Histogram for freemem
Frequency
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000
freemem

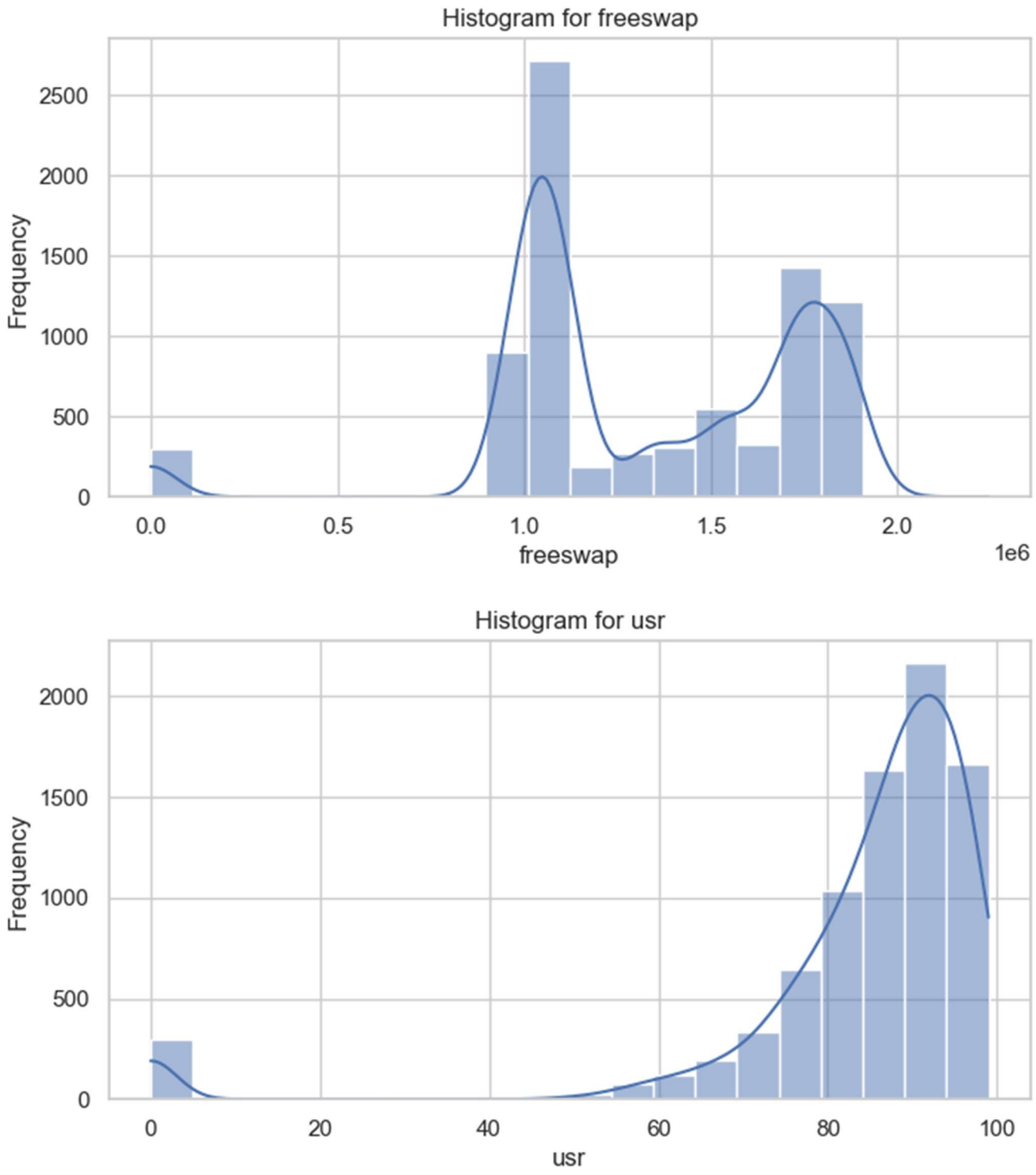
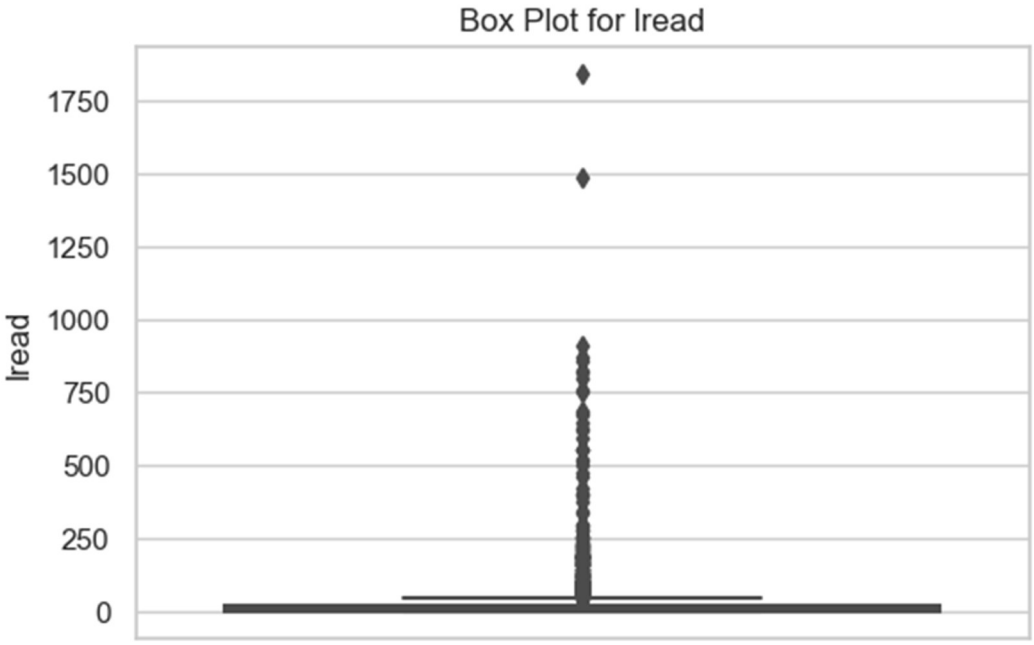
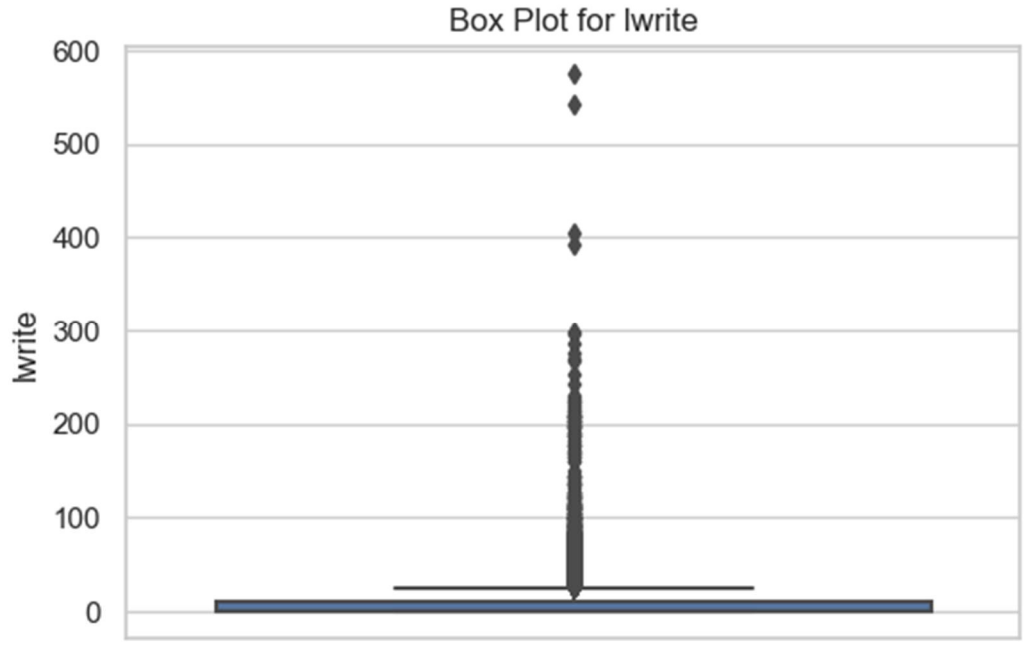
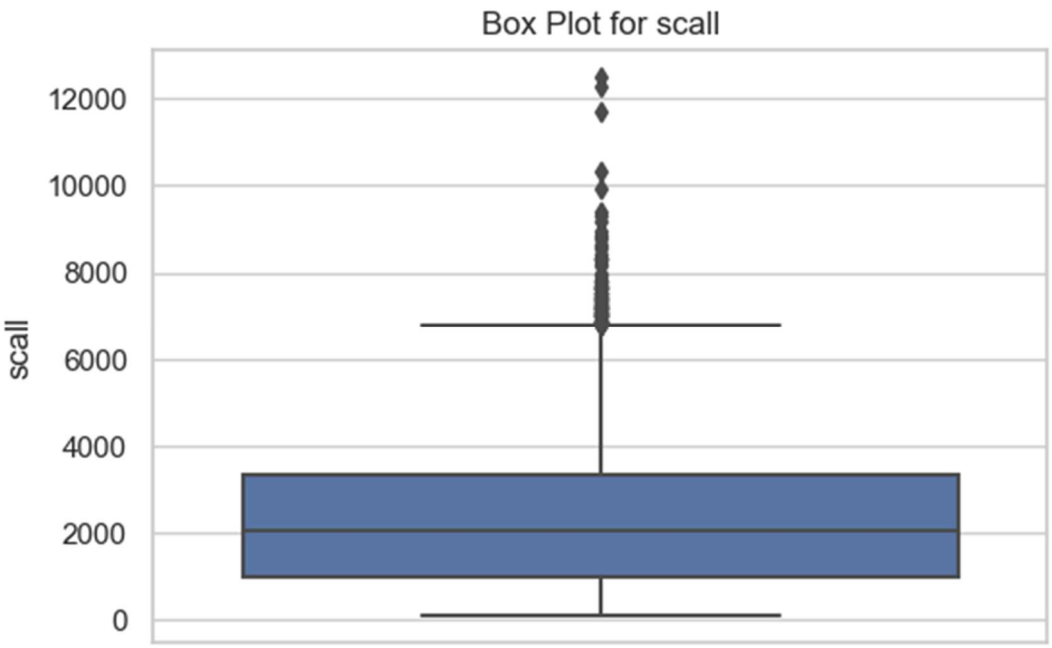


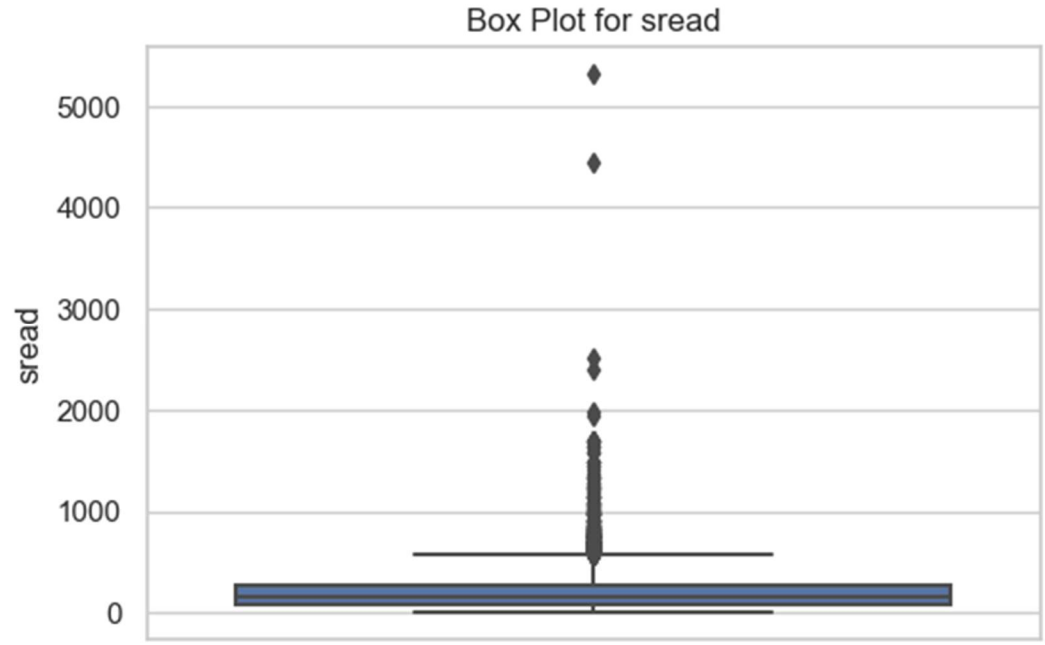
Fig 7: Histograms for numerical columns (21) of comp_activ

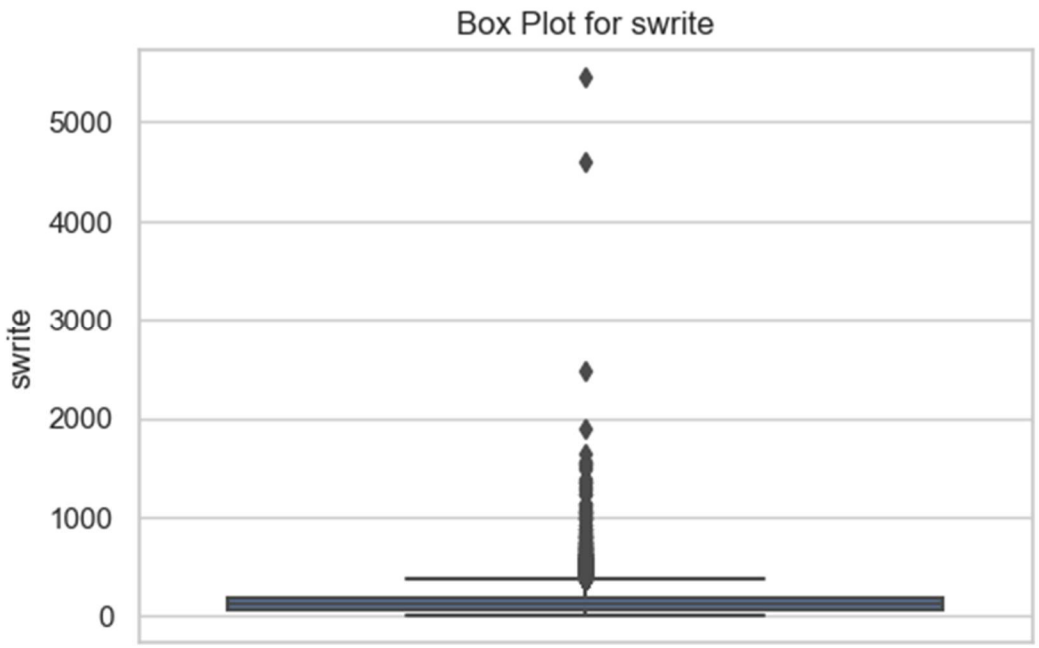
- **Box plots for numerical columns**

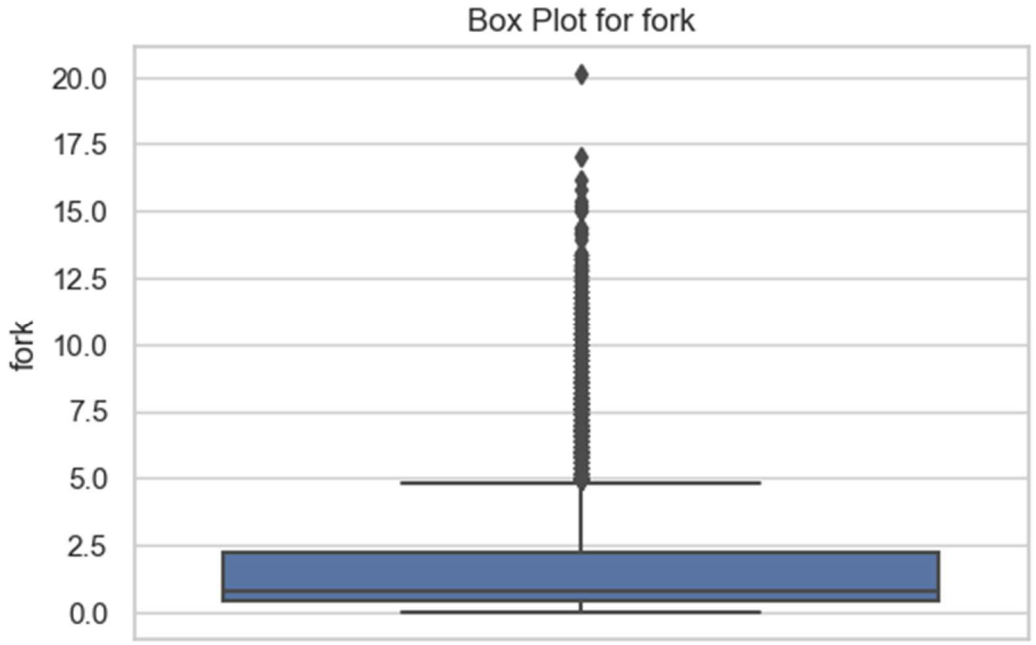


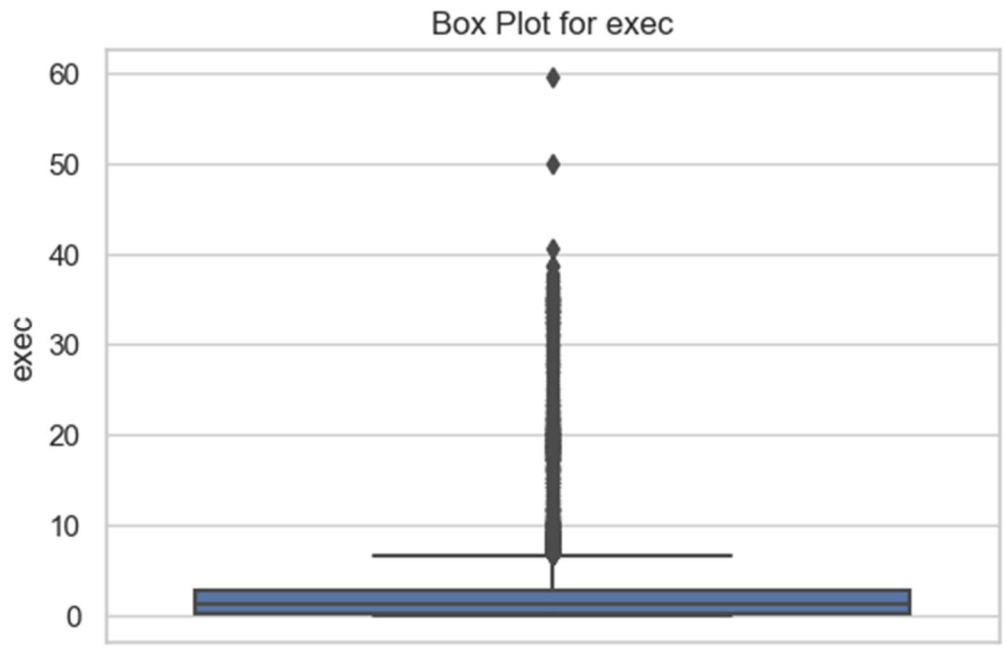


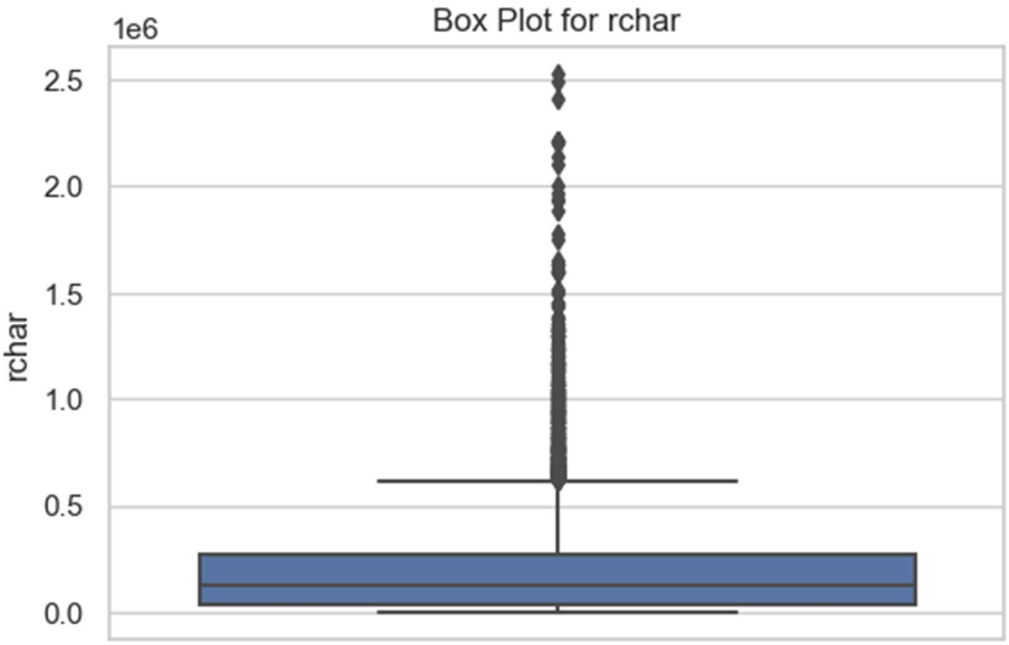

Box Plot for scall
scall
12000
10000
8000
6000
4000
2000
0

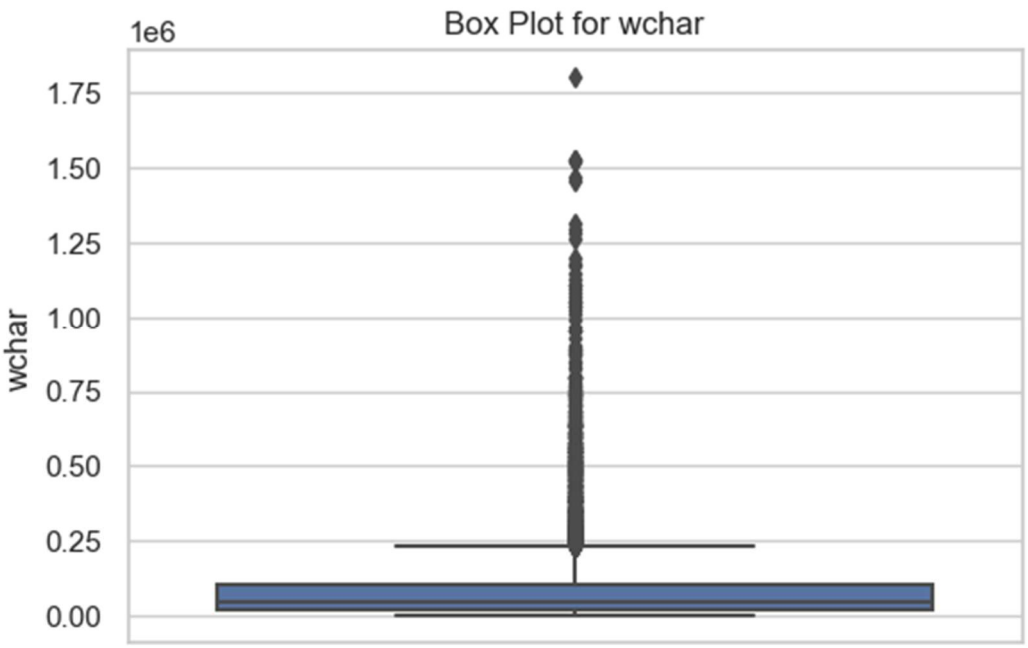

Box Plot for sread
sread
5000
4000
3000
2000
1000
0

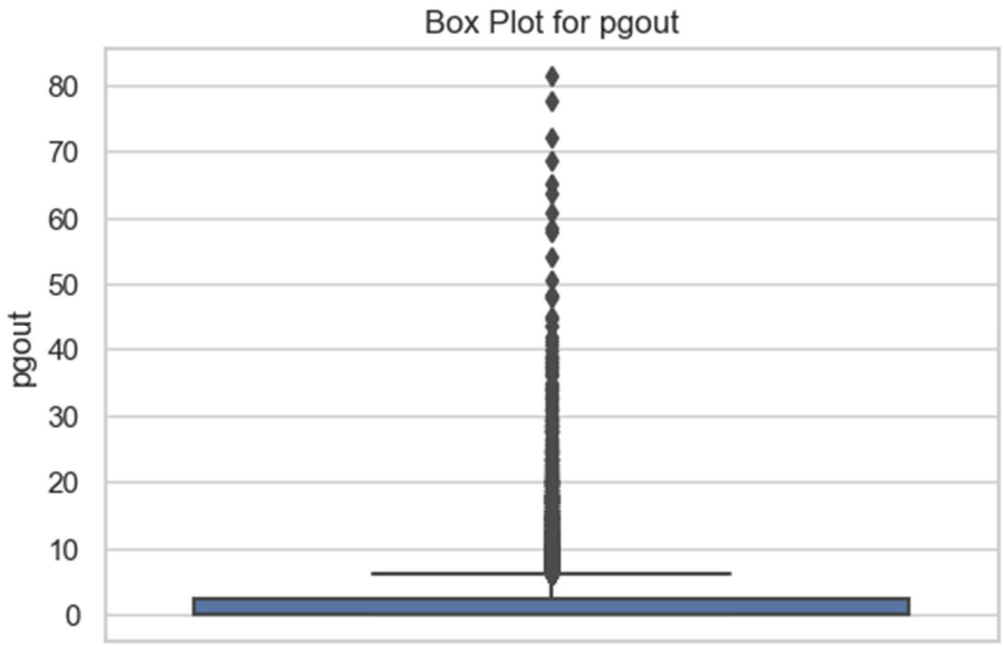

Box Plot for swrite
swrite
5000
4000
3000
2000
1000
0

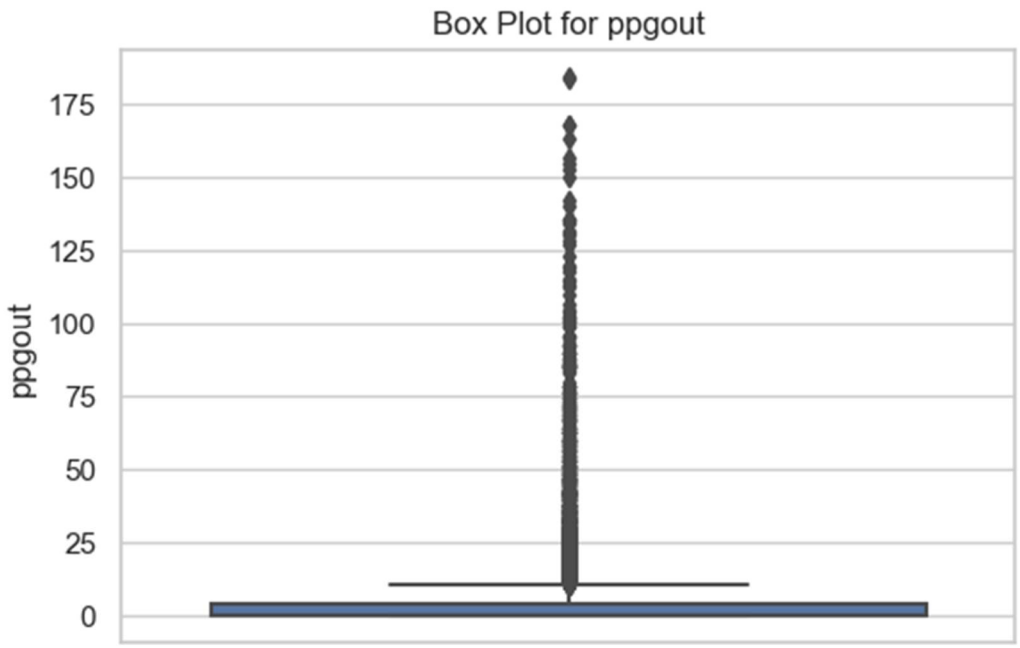

Box Plot for fork
fork
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0

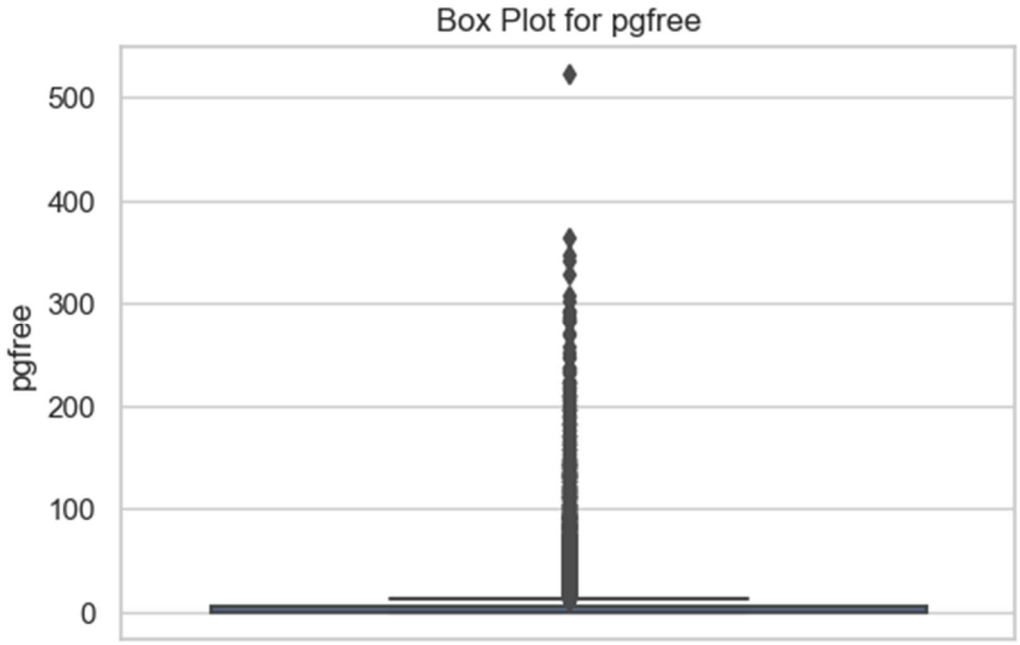

Box Plot for exec
exec
60
50
40
30
20
10
0

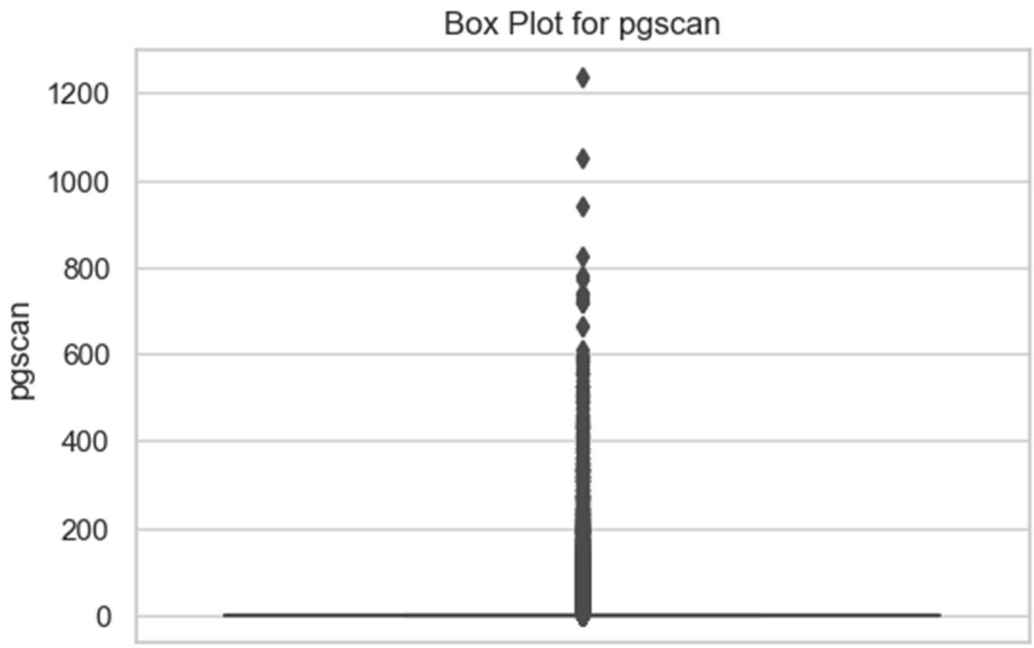

Box Plot for rchar
1e6
rchar
2.5
2.0
1.5
1.0
0.5
0.0

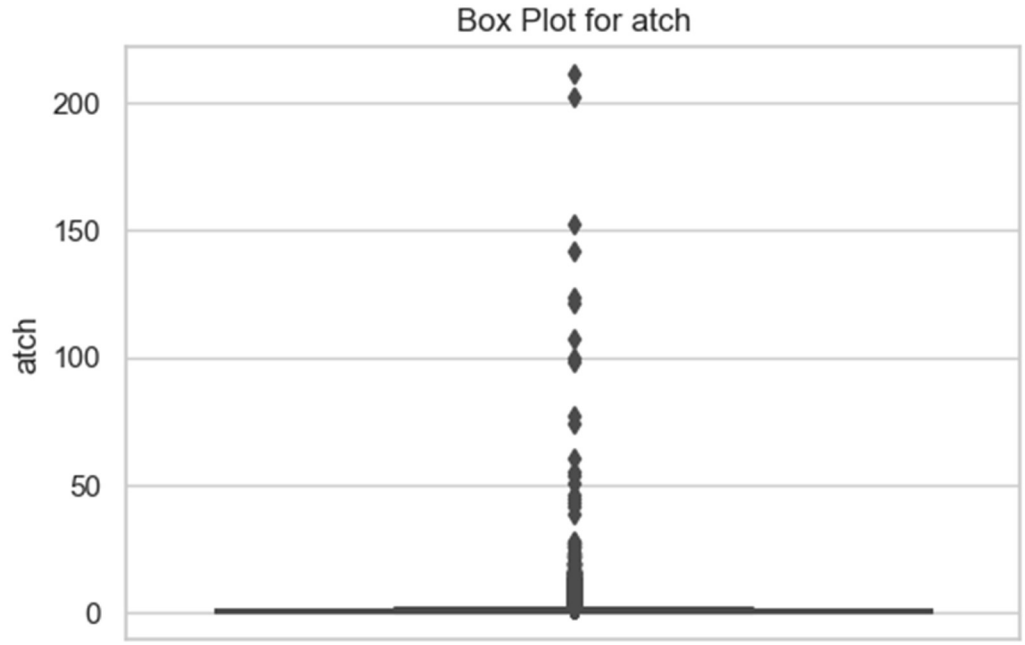

Box Plot for wchar
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
wchar

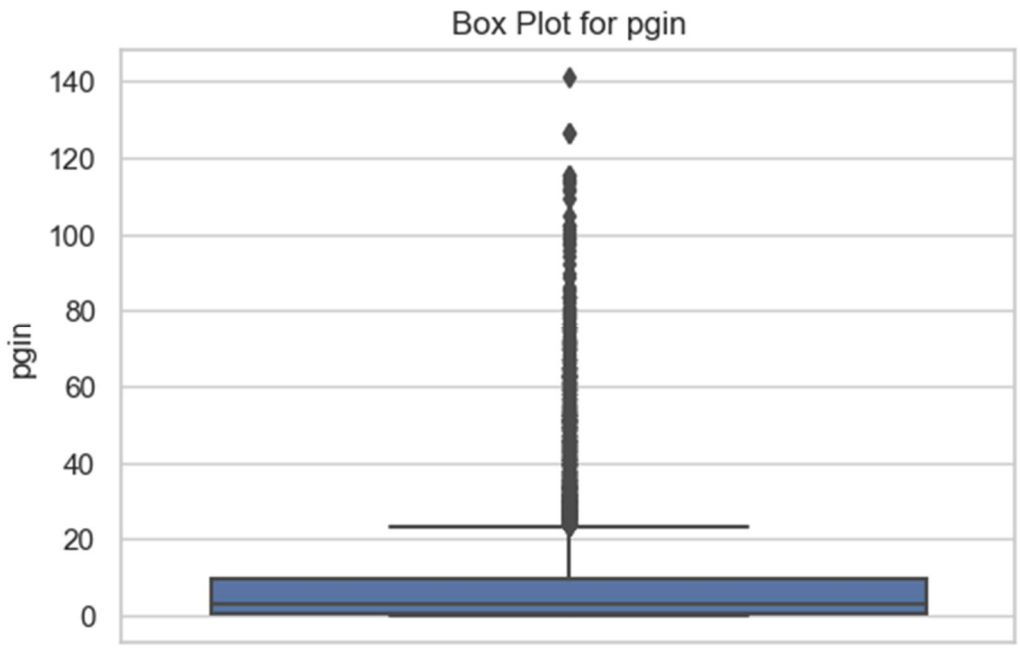

Box Plot for pgout
80
70
60
50
40
30
20
10
0
pgout

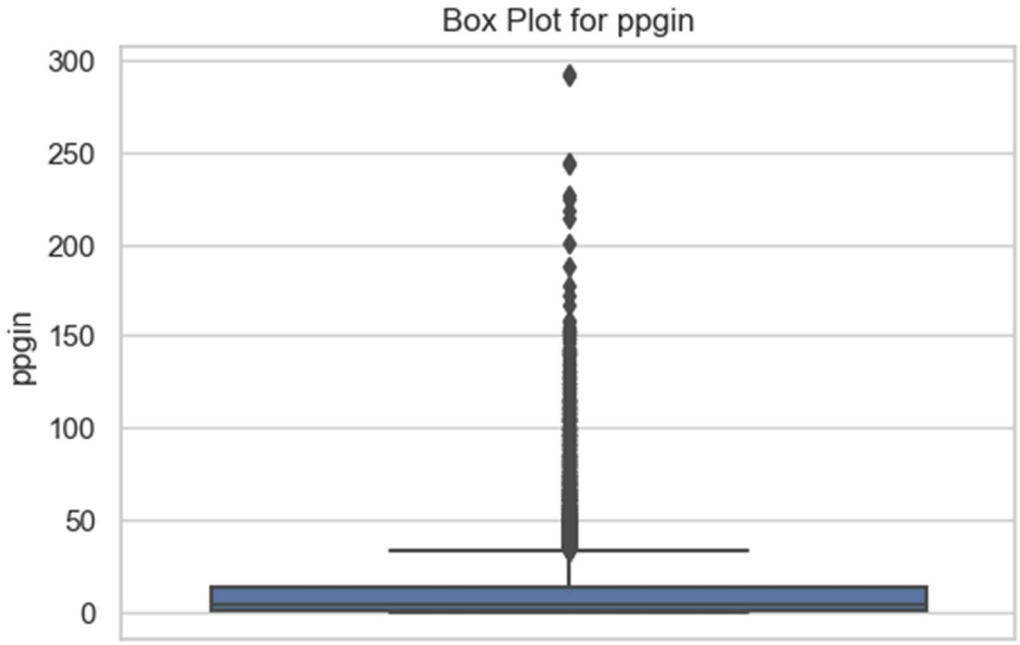

Box Plot for ppgout
ppgout
175
150
125
100
75
50
25
0

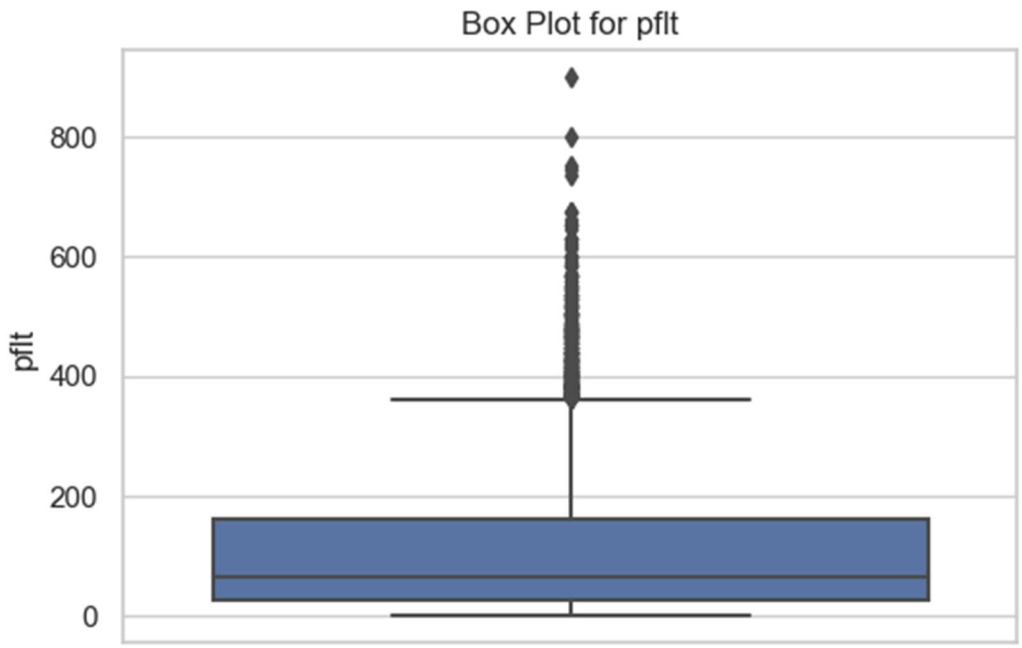

Box Plot for pgfree
pgfree
500
400
300
200
100
0

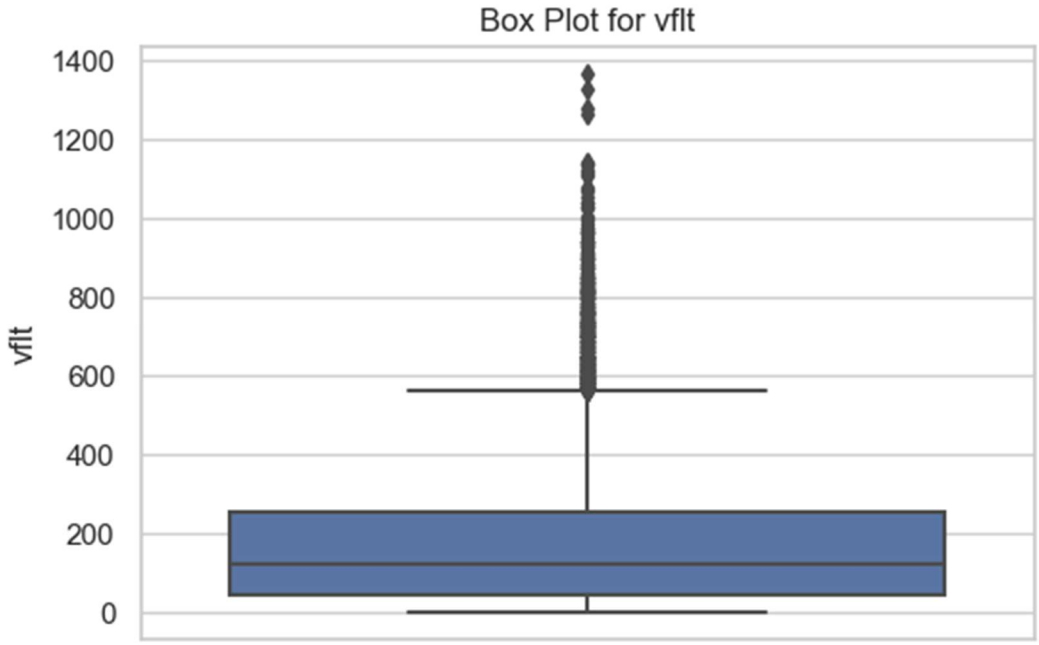

Box Plot for pgscan
pgscan
1200
1000
800
600
400
200
0

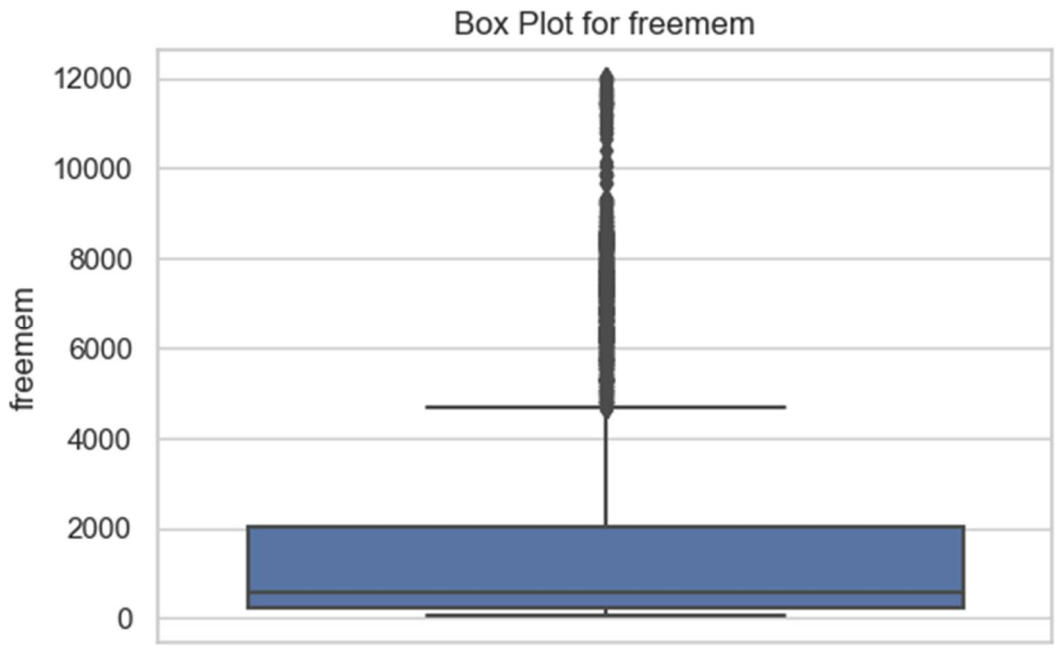

Box Plot for atch
atch
200
150
100
50
0

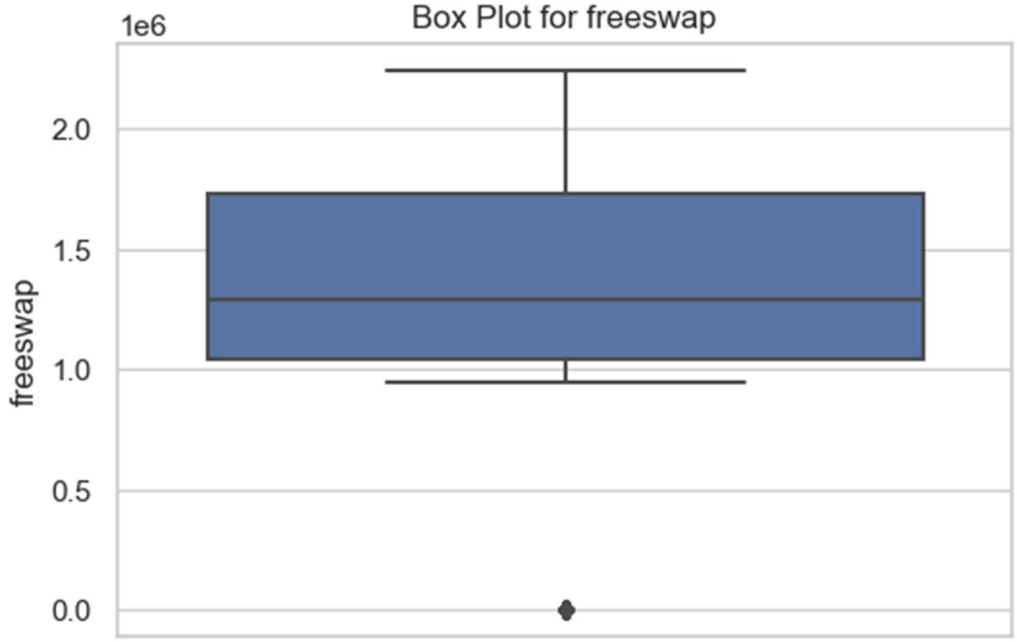

Box Plot for pgin
pgin
140
120
100
80
60
40
20
0


Box Plot for ppgin
ppgin
300
250
200
150
100
50
0


Box Plot for pflt
pflt
800
600
400
200
0


Box Plot for vflt
vflt
1400
1200
1000
800
600
400
200
0


Box Plot for freemem
freemem
12000
10000
8000
6000
4000
2000
0


Box Plot for freeswap
1e6
freeswap
2.0
1.5
1.0
0.5
0.0

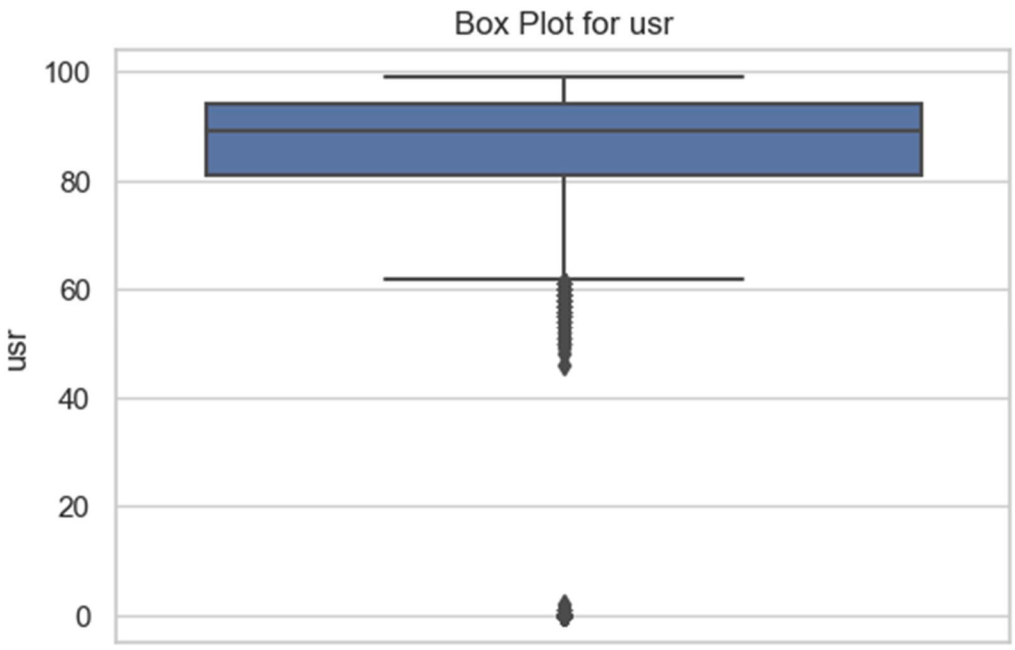


Fig 8: Box plots for numerical columns (21) of comp_activ

➢ **Correlation matrix heatmap (for all numerical columns)**

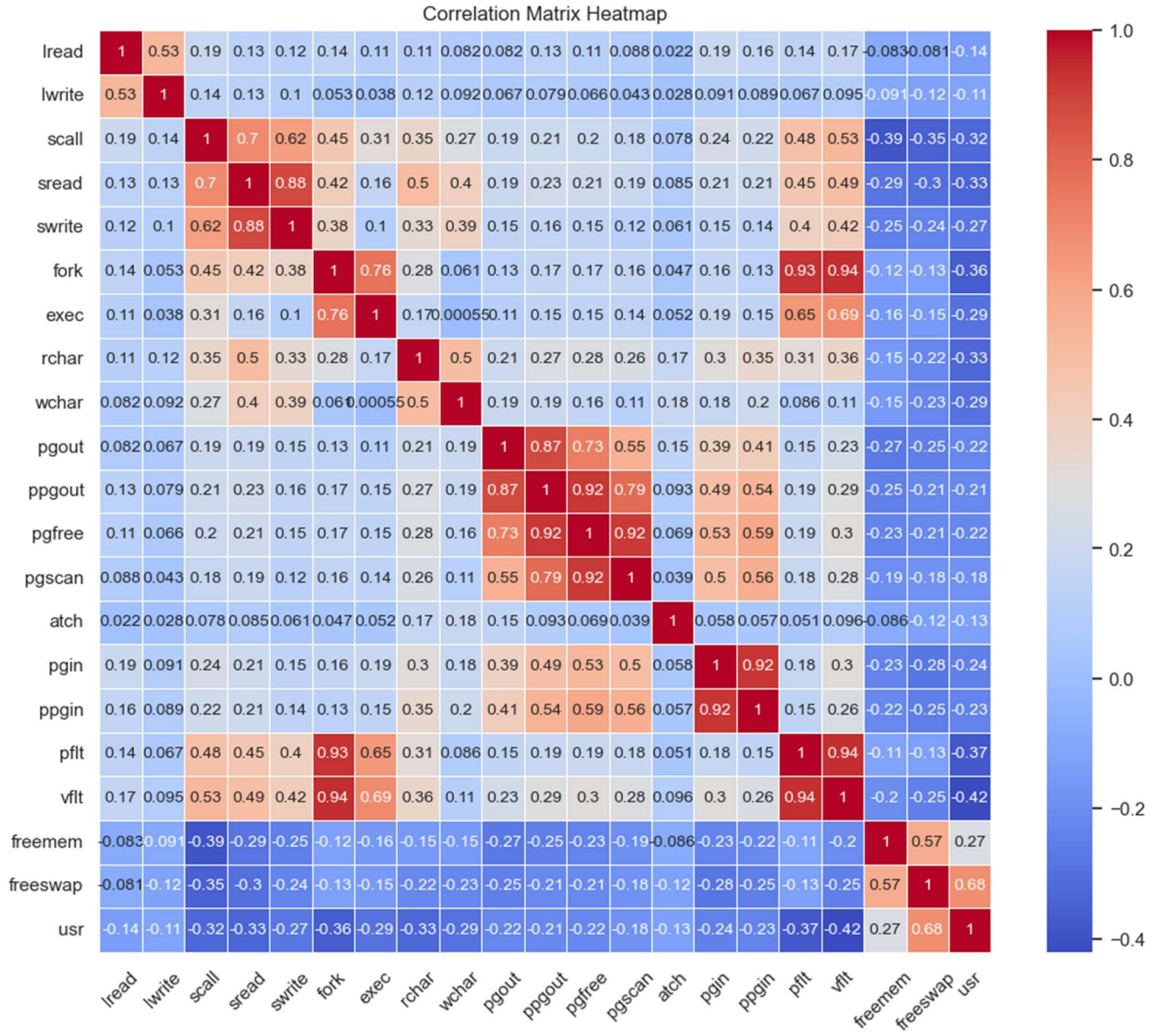


Fig 9: Correlation matrix heat map for all numerical columns of comp_activ

- **Pair plot**

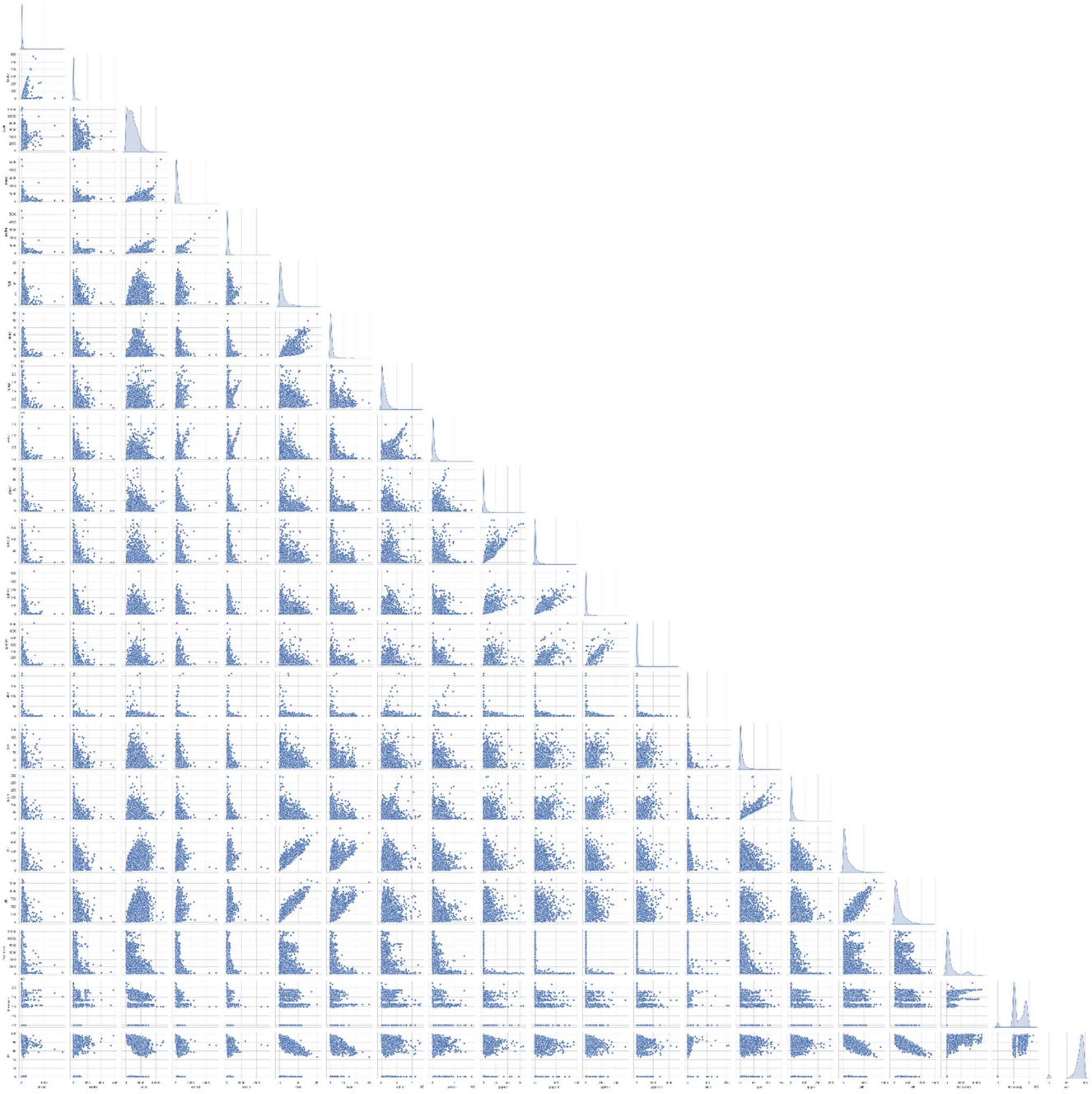


Fig 10 : Pair plot of comp_activ

- **Perform Univariate Analysis for Categorical Data**

	Category	Frequency
0	Not_CPU_Bound	4331
1	CPU_Bound	3861

Table 1 : frequency distribution runqsz categories of comp_activ

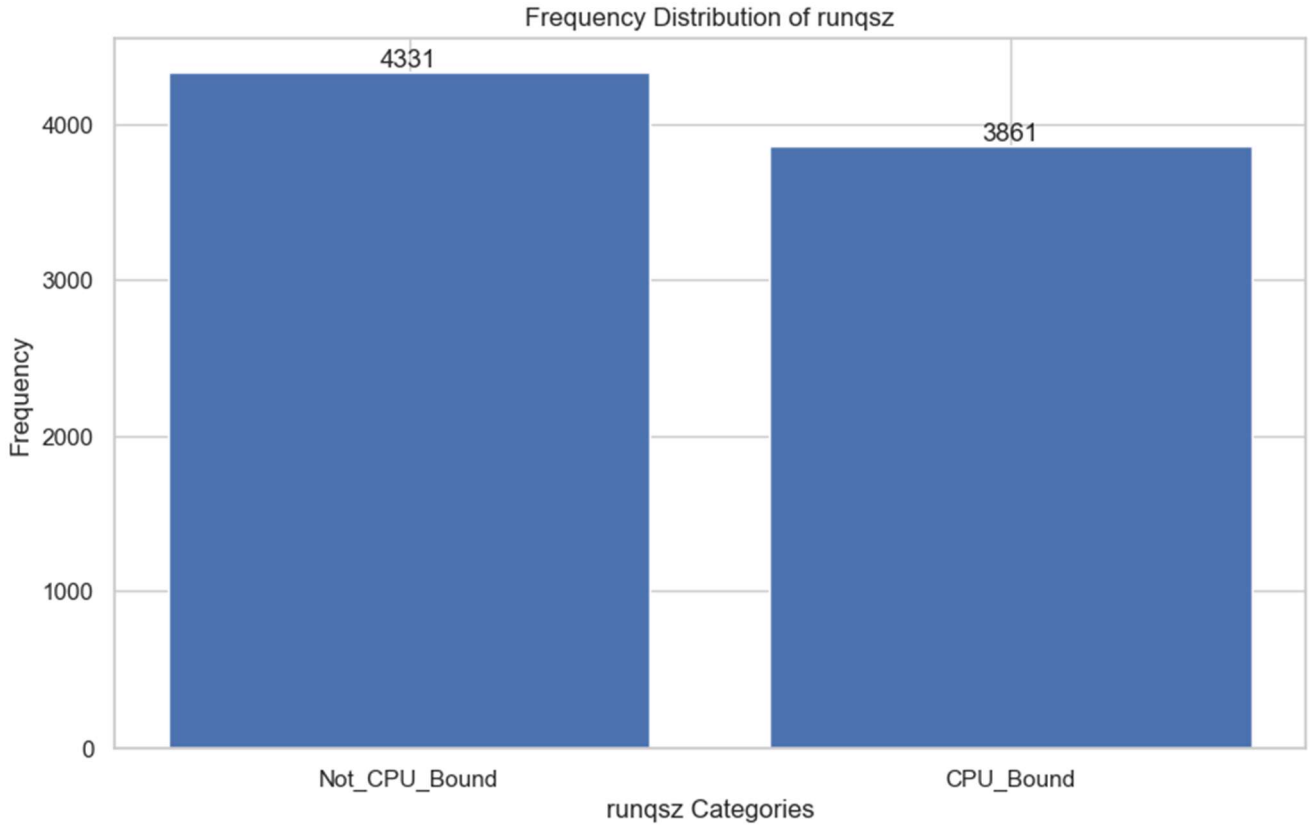


Fig 11 : Bar plot for runqsz of comp_activ

Insights:

1. In the data set, there are 8192 rows and 22 columns that consist of categorical and numerical values (float64(13), int64(8), and object(1)).
2. Two columns rchar and wchar have null values.
3. There are no duplicate values.
4. Most of the numerical columns have outliers.
5. From the histogram, we can say all the features have a left-skewed distribution. The usr' feature has a right skewed distribution.
6. 'usr' is the target variable, and all other variables are predector variables.
7. Bivariate and multivariate analysis indicates that there is a strong positive correlation between the target variable usr and the predictor variables freemem and freeswap.
8. From the analysis, we can say we have a total process run queue size of 3861 as CPU_Bound and 4331 as Not CPU_Bound.

1.2 Impute null values if present, also check for the values which are equal to zero. Do they have any meaning or do we need to change them or drop them? Check for the possibility of creating new features if required. Also check for outliers and duplicates if there.

- **Check Null-value** In the data set, there are null values for 'rchar' and 'wchar. As it is a continuous variable, the mean value can be imputed.

lread	0
lwrite	0
scall	0
sread	0
swrite	0
fork	0
exec	0
rchar	104
wchar	15
pgout	0
ppgout	0
pgfree	0
pgscan	0
atch	0
pgin	0
ppgin	0
pflt	0
vflt	0
runqsz	0
freemem	0
freeswap	0
usr	0
dtype: int64	

lread	0
lwrite	0
scall	0
sread	0
swrite	0
fork	0
exec	0
rchar	0
wchar	0
pgout	0
ppgout	0
pgfree	0
pgscan	0
atch	0
pgin	0
ppgin	0
pflt	0
vflt	0
runqsz	0
freemem	0
freeswap	0
usr	0
dtype: int64	

Fig 12 : Null-value of comp_activ

- **Check for the values which are equal to zero** We can keep the 0s in our dataset for further analysis, as we have some features in our dataset that can be 0s if the system stays idle.

```
lread         675
lwrite       2684
scall           0
sread           0
swrite          0
fork           21
exec           21
rchar           0
wchar           0
pgout        4878
ppgout       4878
pgfree       4869
pgscan       6448
atch         4575
pgin         1220
ppgin        1220
pflt            3
vflt            0
runqsz          0
freemem         0
freeswap        0
usr           283
dtype: int64
```

Fig 13 : Zero -value of comp_activ

➢ Box Plot Before Outliers Treatment

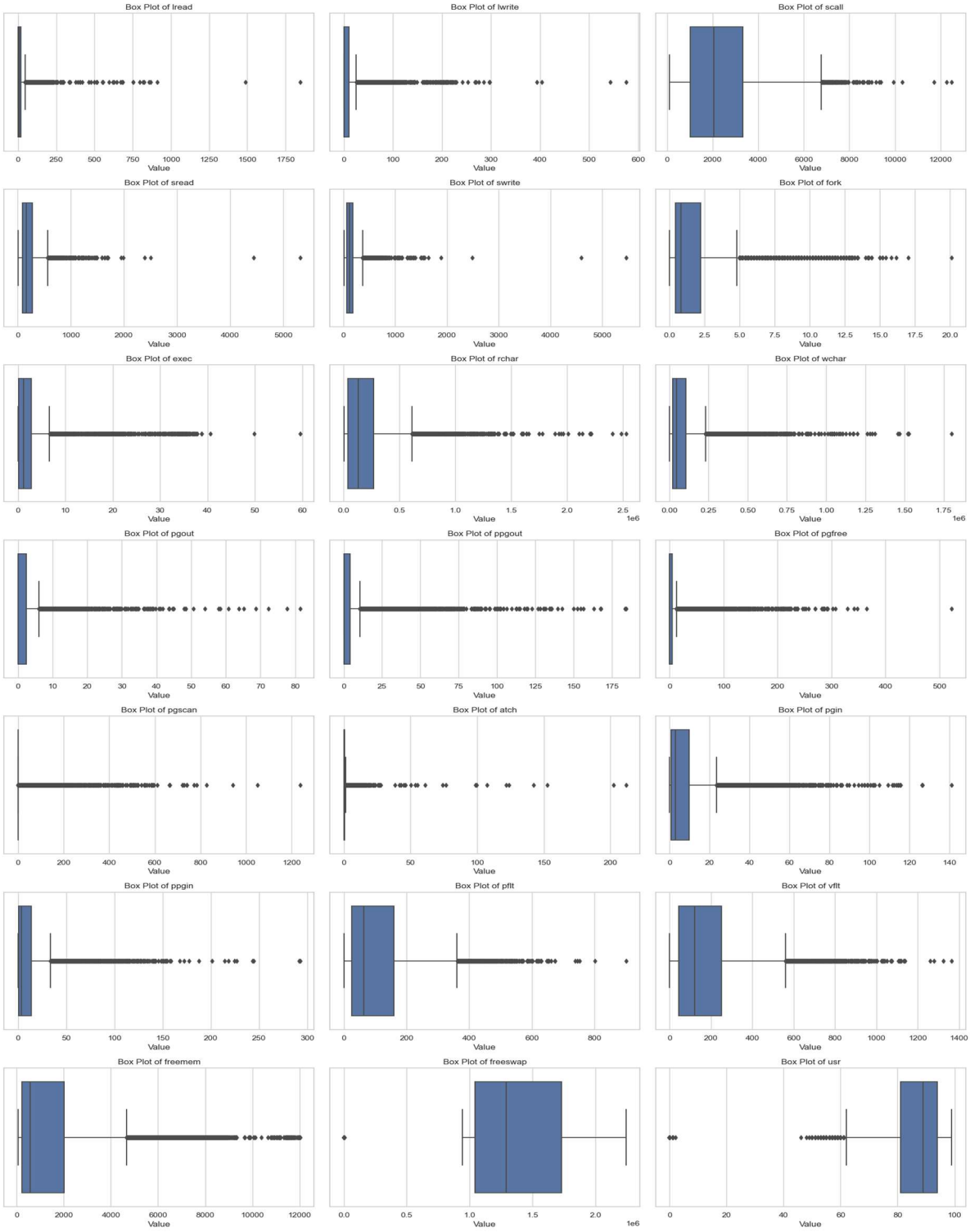


Fig 14 : Box Plot Before Outliers Treatment of comp_activ

➢ Box Plot After Outliers Treatment

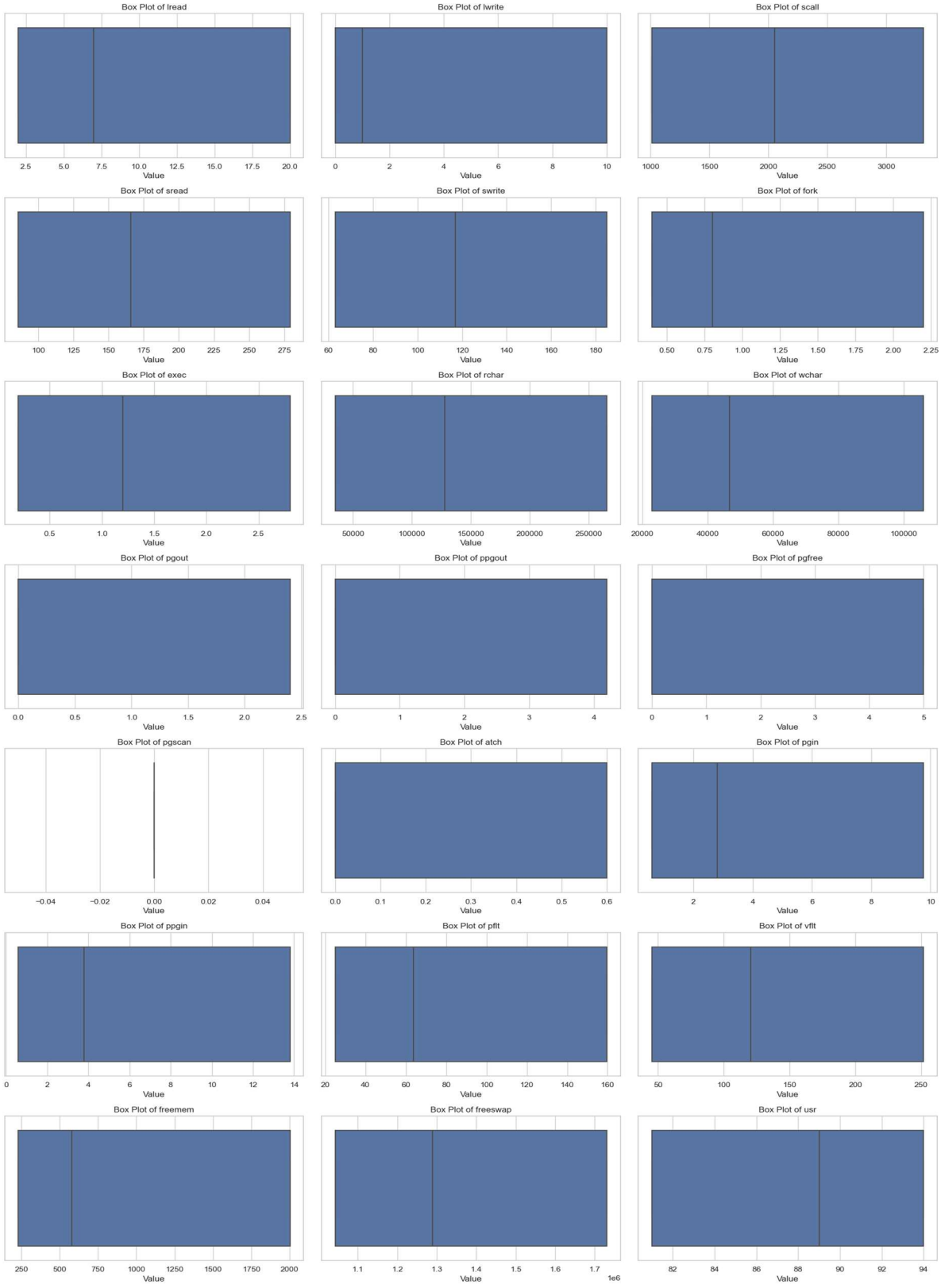


Fig 15 : Box Plot After Outliers Treatment of comp_activ

- **Unique values for categorical variables**

Unique values in 'runqsz'	
Not_CPU_Bound	4331
CPU_Bound	3861

Table 2 : Unique values in 'runqsz' column of comp_activ

- **Converting categorical to dummy variables**

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	usr	run
0	2.0	0.0	2147.0	86.0	68.0	0.4	0.2	40671.000000	53995.00	0.0	...	0.0	0.0	1.6	2.6	25.0	45.4	2002.25	1730379.5	94.0	
1	2.0	0.0	1012.0	86.0	63.0	0.4	0.2	34860.500000	22977.75	0.0	...	0.0	0.0	0.6	0.6	25.0	45.4	2002.25	1730379.5	94.0	
2	15.0	3.0	2162.0	159.0	119.0	2.0	2.4	197385.728363	31950.00	0.0	...	0.0	0.6	6.0	9.4	150.2	220.2	702.00	1042623.5	87.0	
3	2.0	0.0	1012.0	86.0	63.0	0.4	0.2	197385.728363	22977.75	0.0	...	0.0	0.0	0.6	0.6	25.0	45.4	2002.25	1730379.5	94.0	
4	5.0	1.0	1012.0	86.0	63.0	0.4	0.4	197385.728363	22977.75	0.0	...	0.0	0.0	1.0	1.2	37.8	47.6	633.00	1730379.5	90.0	

Fig 16 : categorical to dummy variables top five data of dataset comp_activ

Insights:

1. I found null values in the rchar and wchar fields.
2. We used the median value from the data set to impute the null values.
3. The majority of the continuous fields had outliers, which we handled using the IQR method.
4. In this situation, scaling the data is not essential because either way we would arrive at an equivalent result. The normal equation, for instance, has a closed-form solution that may be used to determine the ideal parameter values for a linear regression model. There is no stepwise optimization procedure in our solution that uses that equation; therefore, feature scaling is not required.
5. It's not necessary to remove entries with 0 values since they.

1.3 Encode the data (having string values) for Modelling. Split the data into train and test (70:30). Apply Linear regression using scikit learn. Perform checks for significant variables using appropriate method from statsmodel. Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare. Compare these models and select the best one with appropriate reasoning.

- **Split the data into train and test (70:30)**

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	\
1310	20.0	10.0	3317.25	279.0	185.0	0.8	0.8	155004.00	106037.00	
7365	15.0	3.0	1203.00	86.0	63.0	1.6	1.8	163076.00	33674.00	
2284	20.0	10.0	3317.25	279.0	185.0	2.2	2.8	265394.75	106037.00	
7076	2.0	0.0	2585.00	203.0	145.0	0.6	0.6	265394.75	106037.00	
3114	2.0	1.0	1827.00	86.0	88.0	0.4	0.2	34860.50	22977.75	

	pgout	...	pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	\
1310	0.0	...	0.0	0.0	0.6	0.600	0.6	48.8	134.00	249.00	
7365	0.0	...	0.0	0.0	0.0	0.600	0.6	127.8	199.40	2002.25	
2284	2.4	...	5.0	0.0	0.6	9.765	13.8	159.6	251.80	236.00	
7076	1.0	...	1.0	0.0	0.6	9.765	13.8	49.9	194.39	451.00	
3114	0.0	...	0.0	0.0	0.0	0.600	0.6	25.0	45.40	689.00	

	freeswap	runqsz_Not_CPU_Bound
1310	1383946.0	0
7365	1542915.0	0
2284	1042623.5	0
7076	1057294.0	1
3114	1730379.5	1

[5 rows x 21 columns]

Fig 17 : X_Train data of dataset comp_activ

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	\
5670	14.0	7.0	1495.0	197.0	169.0	0.8	1.0	34860.50	24435.0	
5369	10.0	8.0	3158.0	279.0	172.0	0.6	2.2	265394.75	106037.0	
2111	2.0	0.0	1012.0	117.0	113.0	1.8	0.6	59903.00	24550.0	
6659	20.0	10.0	3283.0	134.0	125.0	0.4	0.4	34860.50	23626.0	
5227	12.0	2.0	2357.0	113.0	96.0	2.2	2.8	55137.00	36291.0	

	pgout	...	pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	\
5670	2.4	...	5.0	0.0	0.6	2.000	2.00	63.07	106.79	231.0	
5369	0.0	...	0.0	0.0	0.0	9.765	13.80	46.00	79.20	510.0	
2111	0.6	...	5.0	0.0	0.0	0.600	0.60	96.00	135.60	231.0	
6659	2.4	...	5.0	0.0	0.6	1.800	2.20	36.40	56.20	461.0	
5227	0.0	...	0.0	0.0	0.0	8.380	12.18	159.60	251.80	530.0	

	freeswap	runqsz_Not_CPU_Bound
5670	1042623.5	0
5369	1042623.5	0
2111	1730379.5	1
6659	1129531.0	1
5227	1077027.0	1

Fig 18 : X_Test data of dataset comp_activ

- **Fit Linear Model**

OLS Regression Results

Dep. Variable:	usr	R-squared (uncentered):	0.982
Model:	OLS	Adj. R-squared (uncentered):	0.982
Method:	Least Squares	F-statistic:	1.588e+04
Date:	Sun, 03 Sep 2023	Prob (F-statistic):	0.00
Time:	16:59:27	Log-Likelihood:	-22244.
No. Observations:	5734	AIC:	4.453e+04
Df Residuals:	5714	BIC:	4.466e+04
Df Model:	20		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
lread	-0.1137	0.045	-2.519	0.012	-0.202	-0.025
lwrite	0.4539	0.073	6.251	0.000	0.312	0.596
scall	0.0028	0.000	9.876	0.000	0.002	0.003
sread	0.0153	0.005	3.184	0.001	0.006	0.025
swrite	0.0598	0.007	8.288	0.000	0.046	0.074
fork	-5.3890	0.681	-7.917	0.000	-6.723	-4.055
exec	1.4121	0.269	5.252	0.000	0.885	1.939
rchar	-6.391e-06	2.33e-06	-2.742	0.006	-1.1e-05	-1.82e-06
wchar	1.338e-05	5.7e-06	2.348	0.019	2.21e-06	2.45e-05
pgout	2.3852	0.721	3.308	0.001	0.972	3.799
ppgout	-2.0694	0.620	-3.336	0.001	-3.285	-0.853
pgfree	0.8804	0.394	2.234	0.026	0.108	1.653
pgscan	-3.148e-16	2.36e-16	-1.335	0.182	-7.77e-16	1.48e-16
atch	6.3307	0.852	7.428	0.000	4.660	8.001
pgin	1.9601	0.185	10.575	0.000	1.597	2.323
ppgin	-0.9193	0.129	-7.142	0.000	-1.172	-0.667
pflt	-0.0374	0.009	-4.102	0.000	-0.055	-0.020
vflt	0.0312	0.005	5.778	0.000	0.021	0.042
freemem	-0.0004	0.000	-1.331	0.183	-0.001	0.000
freeswap	4.651e-05	3.93e-07	118.411	0.000	4.57e-05	4.73e-05
runqsz_Not_CPU_Bound	10.6739	0.310	34.411	0.000	10.066	11.282

Omnibus:	188.164	Durbin-Watson:	1.940
Prob(Omnibus):	0.000	Jarque-Bera (JB):	206.373
Skew:	0.462	Prob(JB):	1.54e-45
Kurtosis:	3.096	Cond. No.	2.00e+22

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The smallest eigenvalue is 2.81e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Fig 19 : OLS Regression Results of dataset comp_activ

The regression results provide valuable insights into the relationship between the dependent variable 'usr' and the independent variables. Let's interpret the key findings:

R-squared (uncentered) and Adj. R-squared (uncentered):

The value of adjusted R-squared, values accounts for the number of predictors in the model, and also around 0.982, which indicates that the predictive power of the model is robust.

F-statistic:

The value of the F-statistic is 1.588e+04, and the associated p-value so close to zero (0.00). which indicates that in the dependent variable ‘usr’ there is at least one of the independent variables in the model is statistically significant in explaining the variance.

Individual Coefficients:

The coefficients for each independent variable represent the effect of that variable on 'usr' while holding other variables constant. For example, 'scall' has a positive coefficient of approximately 0.0028, suggesting that an increase in 'scall' is associated with an increase in 'usr.' 'fork' has a negative coefficient of approximately -5.3890, indicating that an increase in 'fork' is associated with a decrease in 'usr.' Other coefficients have similar interpretations.

P-values:

The p-values associated with each coefficient test the null hypothesis that the corresponding variable has no effect on 'usr.' Variables with p-values close to zero are considered statistically significant predictors. Multicollinearity: The notes section mentions that the model does not contain a constant term and highlights the possibility of strong multicollinearity or other numerical problems due to the large condition number (2.81e-29). It's essential to investigate multicollinearity further, as high multicollinearity can affect the stability and interpretability of the coefficients.

How to check for Multicollinearity

There are different ways of detecting (or testing) multicollinearity. One such way is Variation Inflation Factor.

Variance Inflation factor: Variance inflation factors measure the inflation in the variances of the regression coefficients estimates due to collinearities that exist among the predictors. It is a measure of how much the variance of the estimated regression coefficient βk is "inflated" by the existence of correlation among the predictor variables in the model.

General Rule of Thumb:

If VIF is 1, then there is no correlation among the k th predictor and the remaining predictor variables, and hence, the variance of βk is not inflated at all.

If VIF exceeds 5, we say there is moderate VIF, and if it is 10 or exceeding 10, it shows signs of high multi-collinearity.

The purpose of the analysis should dictate which threshold to use.

- **Checking Multicollinearity using Variance Inflation Factor (VIF)**

```
VIF values:

lread                   12.510466
lwrite                   6.918800
scall                   18.398697
sread                   35.959872
swrite                  37.045346
fork                    36.989257
exec                     9.227474
rchar                    6.629488
wchar                    6.174430
pgout                   37.054833
ppgout                  81.422159
pgfree                  46.294661
pgscan                        NaN
atch                     3.456503
pgin                    48.162294
ppgin                   46.184467
pflt                    34.527944
vflt                    32.280749
freemem                  6.259508
freeswap                12.409349
runqsz_Not_CPU_Bound     2.120644
dtype: float64
```

Fig 20 : VIF values of dataset comp_activ

Insights:

- The 'pgscan' variable has a VIF value of NaN, indicating perfect multicollinearity. This means that 'pgscan' can be perfectly predicted by other variables in the model, making it impossible to estimate its unique contribution to the dependent variable.
- Variables with VIF values significantly above 10 are considered to have high multicollinearity. In your case, 'sread,' 'swrite,' 'fork,' 'pgout,' 'ppgout,' 'pgfree,' 'pgin,' 'pflt,' 'vflt,' and 'freeswap' all have VIF values well above this threshold.
- 'ppgout' stands out with an exceptionally high VIF value of 81.42, indicating extremely high multicollinearity with other variables.
- Variables with VIF values close to 1 have low multicollinearity. In your case, 'atch' and 'runqsz_Not_CPU_Bound' have VIF values around or below 3, suggesting relatively low multicollinearity with other variables.
- To treat multicollinearity, we will have to drop one or more of the correlated features (cyclinders, displacement, horsepower and weight).
- We will drop the variable that has the least impact on the adjusted R-squared of the model.

- **Fit a simple linear model**

```
The coefficient for lread is -0.04118737280640683
The coefficient for lwrite is 0.005754585369277813
The coefficient for scall is -0.000884558947615138
The coefficient for sread is -0.004198342018953773
The coefficient for swrite is -0.010462027641040586
The coefficient for fork is -0.24025999623680555
The coefficient for exec is -0.15027290222463954
The coefficient for rchar is -4.469856223624041e-06
The coefficient for wchar is -5.103636147348358e-06
The coefficient for pgout is -0.1718826706061481
The coefficient for ppgout is -0.06674110161520845
The coefficient for pgfree is 0.009538097172911349
The coefficient for pgscan is -1.1102230246251565e-16
The coefficient for atch is 0.8029892541500525
The coefficient for pgin is -0.04169407370855619
The coefficient for ppgin is -0.07773998823058044
The coefficient for pflt is -0.01648471347750056
The coefficient for vflt is -0.01569978940678018
The coefficient for freemem is 0.0005138228242103541
The coefficient for freeswap is 5.984829259327423e-08
The coefficient for runqsz_Not_CPU_Bound is 0.2190579991478833
```

Fig 21 : The coefficient for columns of dataset comp_activ

Train R-squared	0.7682181016301031
Test R-squared	0.7551759667986745
Train RMSE	2.5016544758943864
Test RMSE	2.5387794553843652

Table 3 : Evaluation metrics of comp_activ

Insights:

The model explains about 75% of the variance in the test set, which is a reasonable level of performance for a linear regression model.

The RMSE values give an overview of the mean prediction error. The test RMSE is slightly higher than the train RMSE, which is to be expected since the model generally fits the training data better. However, both RMSE values are relatively low, indicating that the model's predictions are reasonably accurate.

The adjusted R-squared value takes into account the number of predictors in the model. In this case, the adjusted R-squared is close to the regular R-squared, indicating that the addition of predictors is not significantly impacting the model's performance.

1.4 Inference: Basis on these predictions, what are the business insights and recommendations. Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

Data Overview:

- The dataset contains 8192 rows and 22 columns.

- Data types include float64, int64, and one object column.
- Two columns, 'rchar' and 'wchar,' have null values.
- No duplicate values are present.
- Most numerical columns exhibit outliers.
- The data distribution appears left-skewed for most features, with 'usr' being right-skewed.

Data Pre-processing:

- Null values in 'rchar' and 'wchar' are imputed using the median value from the dataset.
- Outliers in continuous fields are treated using the IQR (Interquartile Range) method.
- Scaling the data is considered unnecessary for this project because certain equations used for modeling (e.g., normal equations for linear regression) do not require feature scaling.

Multicollinearity Analysis:

- Perfect multicollinearity is observed for the 'pgscan' variable, indicating it can be predicted perfectly by other variables.
- Several variables have high VIF (Variance Inflation Factor) values, suggesting significant multicollinearity. Notably, 'ppgout' stands out with an exceptionally high VIF value.
- Variables with VIF values close to 1 indicate low multicollinearity.

Handling Multicollinearity:

- To address multicollinearity, one or more correlated features will be dropped from the model.
- The variable to be dropped will be chosen based on its impact on the adjusted R-squared of the model.

Model Evaluation:

- The model explains approximately 75% of the variance in the test set, which is considered reasonable for a linear regression model.
- Root Mean Square Error (RMSE) values indicate that the model's predictions are reasonably accurate.
- The adjusted R-squared, which considers the number of predictors, is close to the regular R-squared, suggesting that adding more predictors does not significantly impact model performance.

Business Interpretation and Actionable Insights:

- Data Quality and Pre-processing: Addressing missing values and outliers is crucial for accurate modelling. The choice of imputation methods and outlier handling should be documented and validated.
- Feature Selection: Careful consideration of feature selection is required to address multicollinearity. Removing highly correlated features can improve model interpretability and prevent overfitting.
- Model Performance: The model's performance is reasonable, but there is always room for improvement. Further fine-tuning and exploring alternative algorithms or features may yield better results.
- Interpretable Models: Consider using interpretable models (if not already) to provide more actionable insights and explanations for stakeholders.
- Continual Monitoring: Monitor model performance over time, as the dataset may change, and model drift can occur. Regular updates and retraining may be necessary.

Problem 2: Logistic Regression, LDA and CART

You are a statistician at the Republic of Indonesia Ministry of Health and you are provided with a data of 1473 females collected from a Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of the survey.

The problem is to predict do/don't they use a contraceptive method of choice based on their demographic and socio-economic characteristics.

DATA DICTIONARY:

1. Wife's age (numerical)
2. Wife's education (categorical) 1=uneducated, 2, 3, 4=tertiary
3. Husband's education (categorical) 1=uneducated, 2, 3, 4=tertiary
4. Number of children ever born (numerical)
5. Wife's religion (binary) Non-Scientology, Scientology
6. Wife's now working? (binary) Yes, No
7. Husband's occupation (categorical) 1, 2, 3, 4(random)
8. Standard-of-living index (categorical) 1=verlow, 2, 3, 4=high
9. Media exposure (binary) Good, Not good
10. Contraceptive method used (class attribute) No,Yes

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, check for duplicates and outliers and write an inference on it. Perform Univariate and Bivariate Analysis and Multivariate Analysis.

Import some libraries like Numpy, Pandas, Seaborn, Matplotlib, Linear Regression machine learning like LinearRegression, Ridge, Lasso, LinearDiscriminantAnalysis, confusion_matrix, StandardScaler, train_test_split etc. After that, load our data set, compactiv.xlsx, and use the head() function to view the Top 10 data and the tail() function to view the bottom 10 data. Using the shape function, we can determine that there are 1473 rows and 10 columns. Find out the characteristics of the columns using the info() method. The datatypes for the float64(2), int64(1), and object(7) columns are present. There are some null values, but there aren't any duplicate values.

- **head()** it given by default top five data

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure
0	24.0	Primary	Secondary	3.0	Scientology	No	2	High	Exposed
1	45.0	Uneducated	Secondary	10.0	Scientology	No	3	Very High	Exposed
2	43.0	Primary	Secondary	7.0	Scientology	No	3	Very High	Exposed
3	42.0	Secondary	Primary	9.0	Scientology	No	3	High	Exposed
4	36.0	Secondary	Secondary	8.0	Scientology	No	3	Low	Exposed
5	19.0	Tertiary	Tertiary	0.0	Scientology	No	3	High	Exposed
6	38.0	Primary	Secondary	6.0	Scientology	No	3	Low	Exposed
7	21.0	Secondary	Secondary	1.0	Scientology	Yes	3	Low	Exposed
8	27.0	Primary	Secondary	3.0	Scientology	No	3	Very High	Exposed
9	45.0	Uneducated	Uneducated	8.0	Scientology	No	2	Low	Not-Exposed

Fig 22 : Top ten data of dataset Contraceptive

- **tail()** it given by default bottom five data

ısband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure	Contraceptive_method_used
Secondary	NaN	Scientology	No	3	Very High	Exposed	Yes
Primary	NaN	Scientology	No	2	Very High	Exposed	Yes
Tertiary	NaN	Scientology	No	1	High	Exposed	Yes
Tertiary	NaN	Scientology	No	2	Very High	Exposed	Yes
Tertiary	NaN	Scientology	No	1	Very High	Exposed	Yes
Tertiary	NaN	Scientology	Yes	2	Very High	Exposed	Yes
Tertiary	NaN	Scientology	No	1	Very High	Exposed	Yes
Secondary	NaN	Scientology	Yes	1	Very High	Exposed	Yes
Secondary	NaN	Scientology	Yes	2	Low	Exposed	Yes
Secondary	1.0	Scientology	No	2	Very High	Exposed	Yes

Fig 23 : Bottom ten data of dataset Contraceptive

- **shape** it tells numbers of rows and columns in given dataset.

```
(1473, 10)
```

Fig 24 : 1473 rows & 10 columns of Contraceptive

- **info()** it tells a concise summary of a DataFrame

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1473 entries, 0 to 1472
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Wife_age                  1402 non-null   float64
 1   Wife_ education           1473 non-null   object
 2   Husband_education         1473 non-null   object
 3   No_of_children_born       1452 non-null   float64
 4   Wife_religion             1473 non-null   object
 5   Wife_Working              1473 non-null   object
 6   Husband_Occupation        1473 non-null   int64
 7   Standard_of_living_index  1473 non-null   object
 8   Media_exposure            1473 non-null   object
 9   Contraceptive_method_used 1473 non-null   object
dtypes: float64(2), int64(1), object(7)
memory usage: 115.2+ KB
```

Fig 25 : Information about the Contraceptive structure and content.

- **Describe()** it tells summary of the central tendency, dispersion, and shape of the distribution of the data.

	Wife_age	No_of_children_born	Husband_Occupation
count	1402.000000	1452.000000	1473.000000
mean	32.606277	3.254132	2.137814
std	8.274927	2.365212	0.864857
min	16.000000	0.000000	1.000000
25%	26.000000	1.000000	1.000000
50%	32.000000	3.000000	2.000000
75%	39.000000	4.000000	3.000000
max	49.000000	16.000000	4.000000

Fig 26 : Descriptive statistics of Contraceptive

- **80 Duplicates Values.**

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_expos
79	38.000000	Tertiary	Tertiary	1.0	Scientology	Yes	1	Very High	Expo
167	26.000000	Tertiary	Tertiary	1.0	Scientology	No	1	Very High	Expo
224	47.000000	Tertiary	Tertiary	4.0	Scientology	No	1	Very High	Expo
270	30.000000	Tertiary	Tertiary	2.0	Scientology	No	1	Very High	Expo
299	26.000000	Tertiary	Tertiary	1.0	Scientology	No	1	Very High	Expo
...	...	...	...	...	...	...	...	...	
1367	44.000000	Tertiary	Tertiary	5.0	Scientology	Yes	1	Very High	Expo
1387	32.606277	Secondary	Tertiary	2.0	Scientology	Yes	2	Very High	Expo
1423	32.606277	Tertiary	Tertiary	2.0	Non-Scientology	No	1	Very High	Expo
1440	32.606277	Tertiary	Tertiary	1.0	Non-Scientology	Yes	2	Very High	Expo
1447	32.606277	Tertiary	Tertiary	2.0	Non-Scientology	Yes	2	Very High	Expo

80 rows × 10 columns

Fig 27 : Descriptive statistics of Contraceptive

- Wife_age and No_od_children_born have Null Values.

```
Wife_age                    71
Wife_ education              0
Husband_education            0
No_of_children_born         21
Wife_religion                0
Wife_Working                 0
Husband_Occupation           0
Standard_of_living_index     0
Media_exposure               0
Contraceptive_method_used    0
dtype: int64
```

```
Wife_age                    0
Wife_ education              0
Husband_education            0
No_of_children_born          0
Wife_religion                0
Wife_Working                 0
Husband_Occupation           0
Standard_of_living_index     0
Media_exposure               0
Contraceptive_method_used    0
dtype: int64
```

Fig 28 : Null values of Contraceptive

- **Before Outliers Treatment**

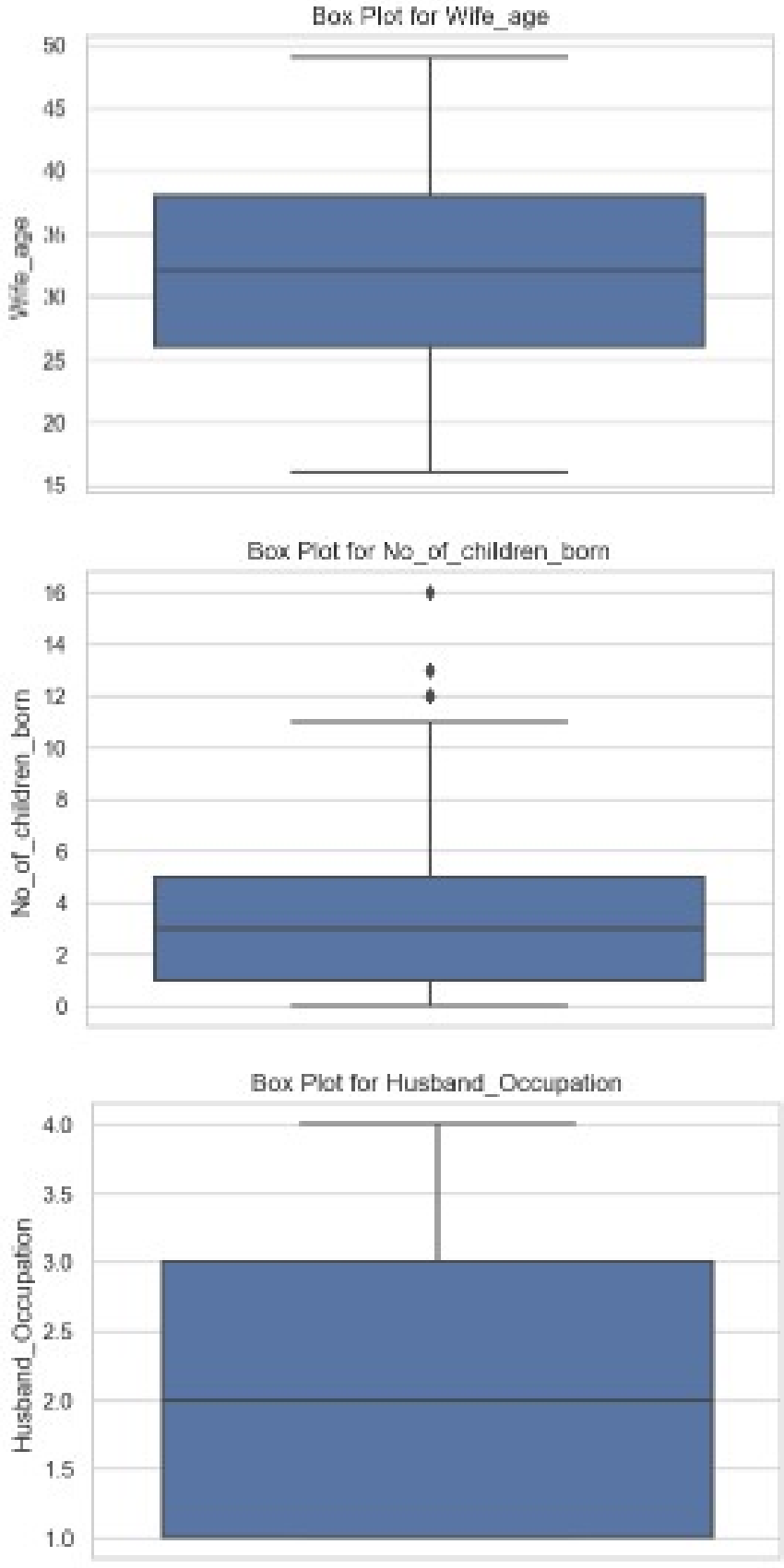


Fig 29 : Before Outliers Treatment Box plots for numeric columns of Contraceptive

- **After Outliers Treatment**

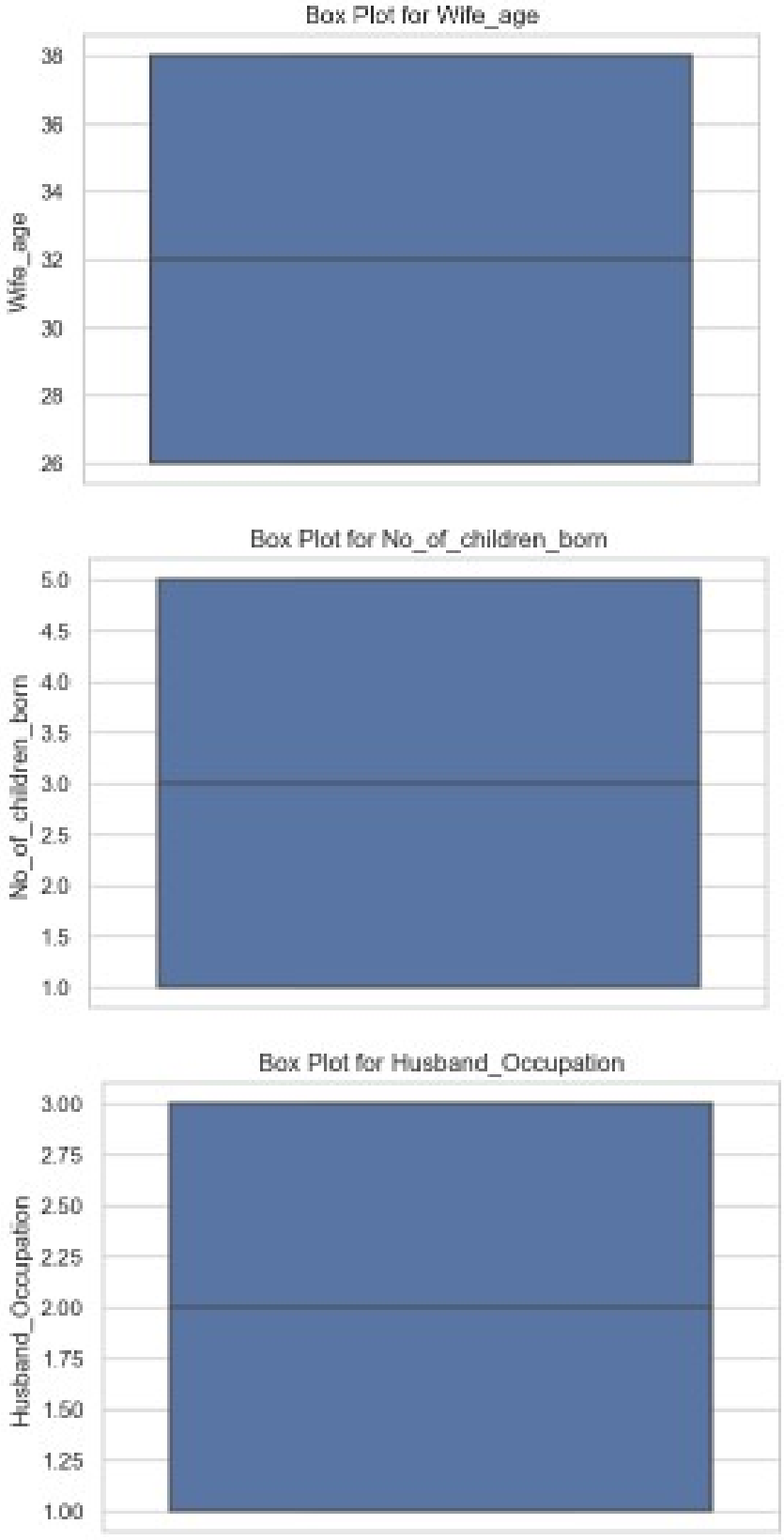


Fig 30 : After Outliers Treatment Box plots for numeric columns of Contraceptive

- **Histograms for Numerical Columns**

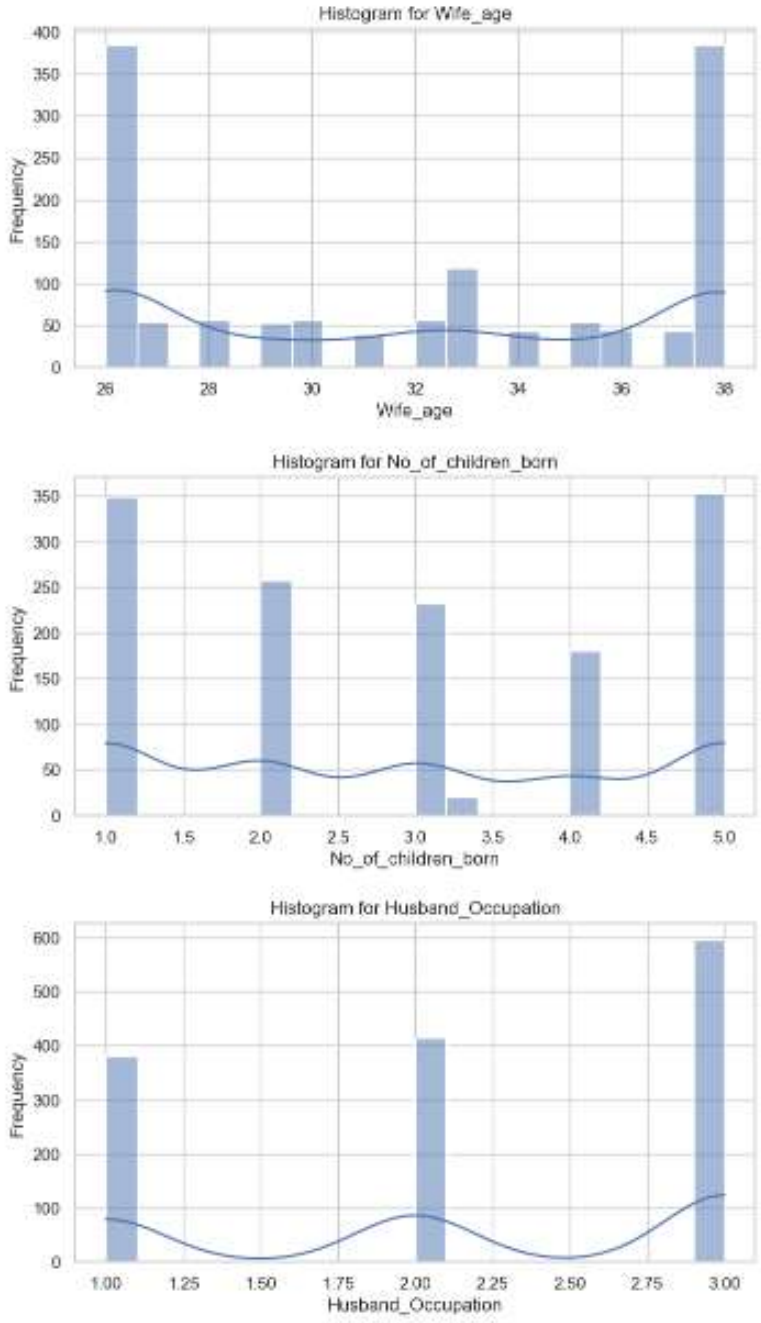


Fig 31 : Histogram for numeric columns of Contraceptive

➢ **Pair plot**

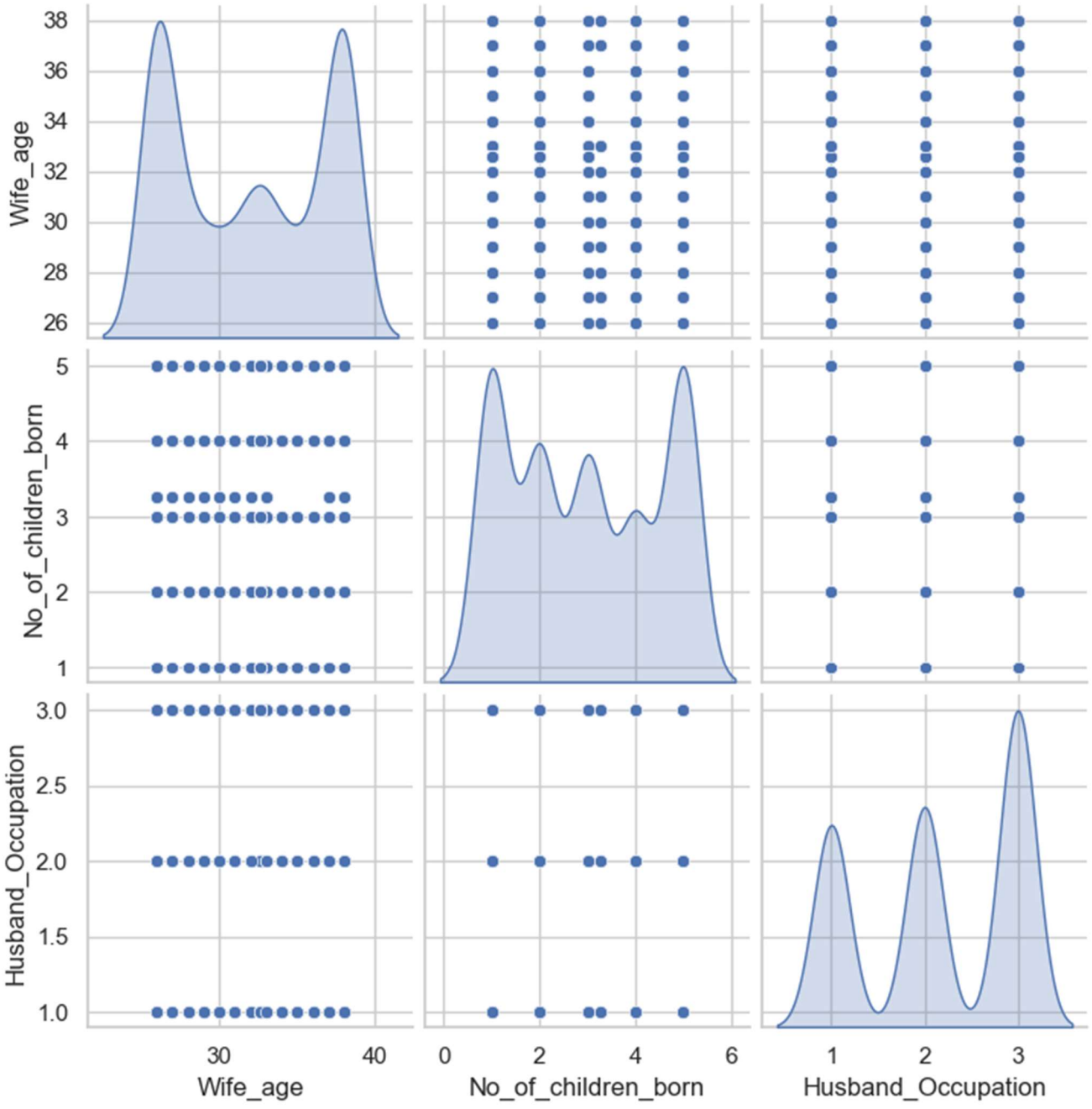


Fig 32 : Pair plot for numeric columns of Contraceptive

➢ **Correlation matrix heatmap**

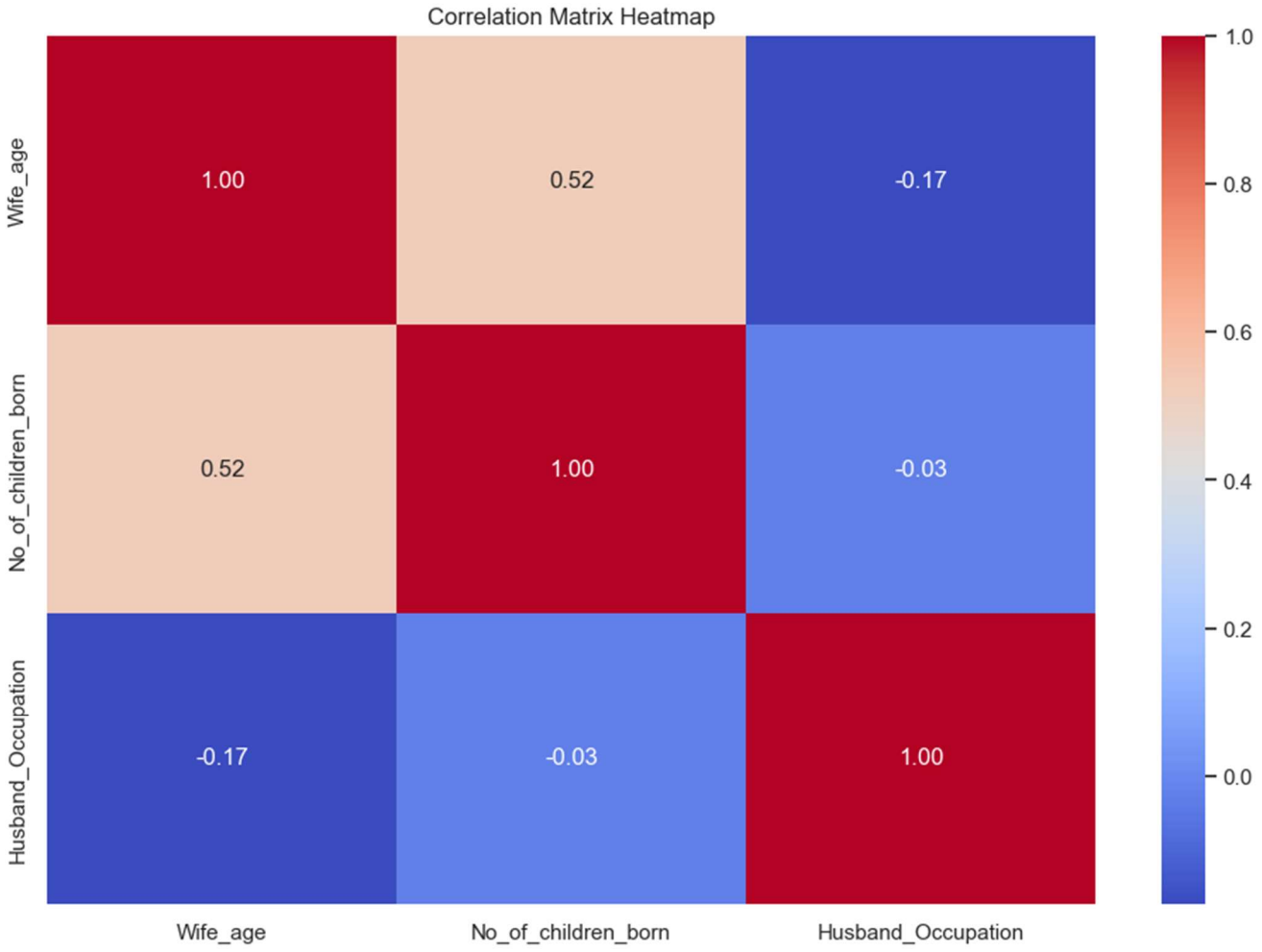
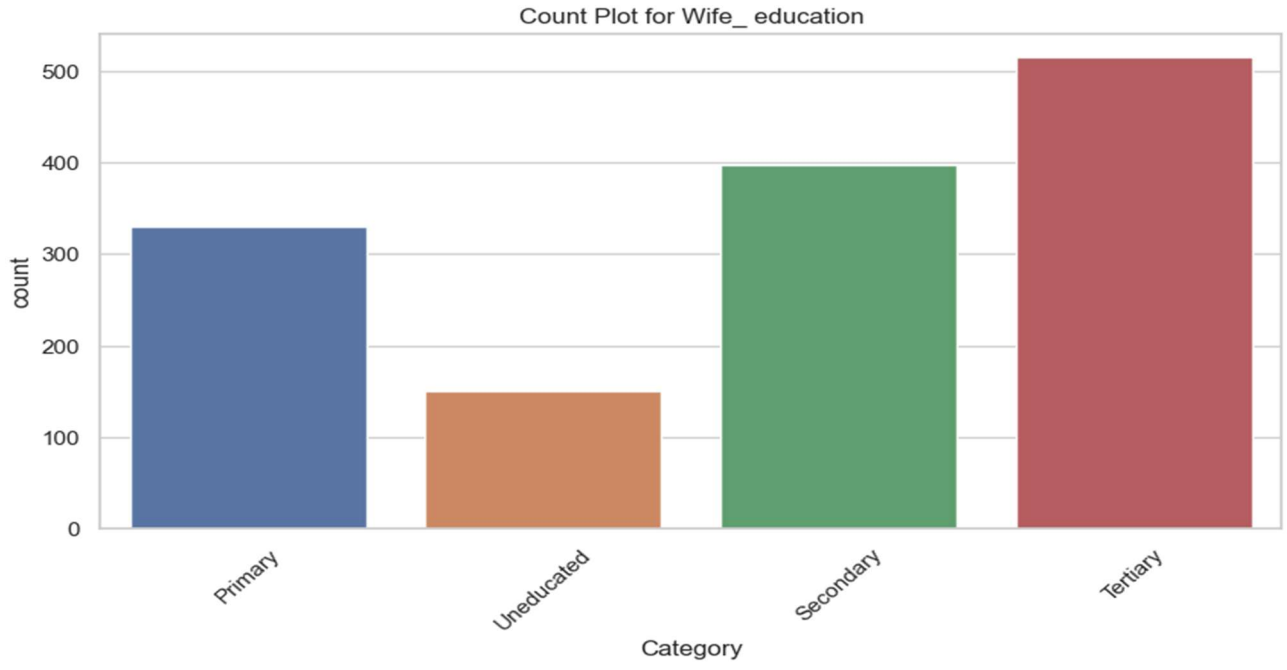
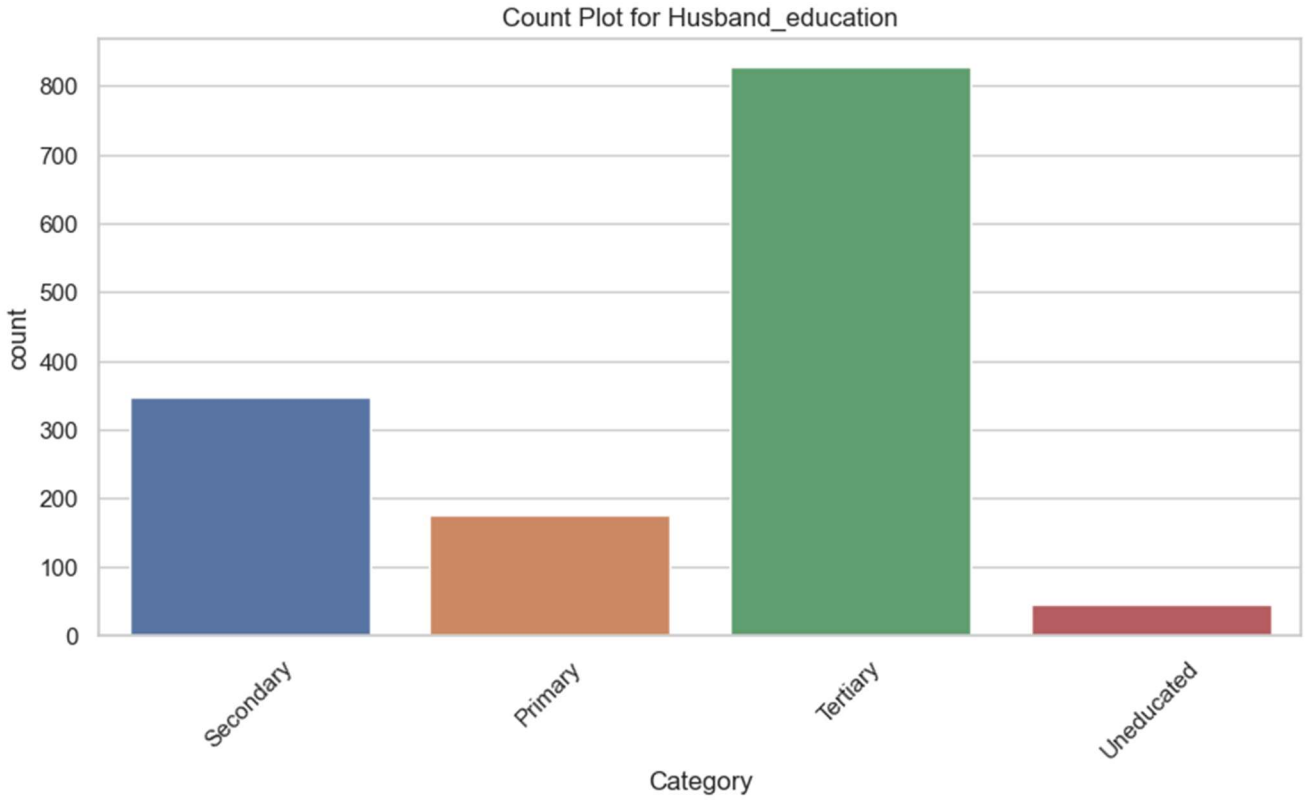
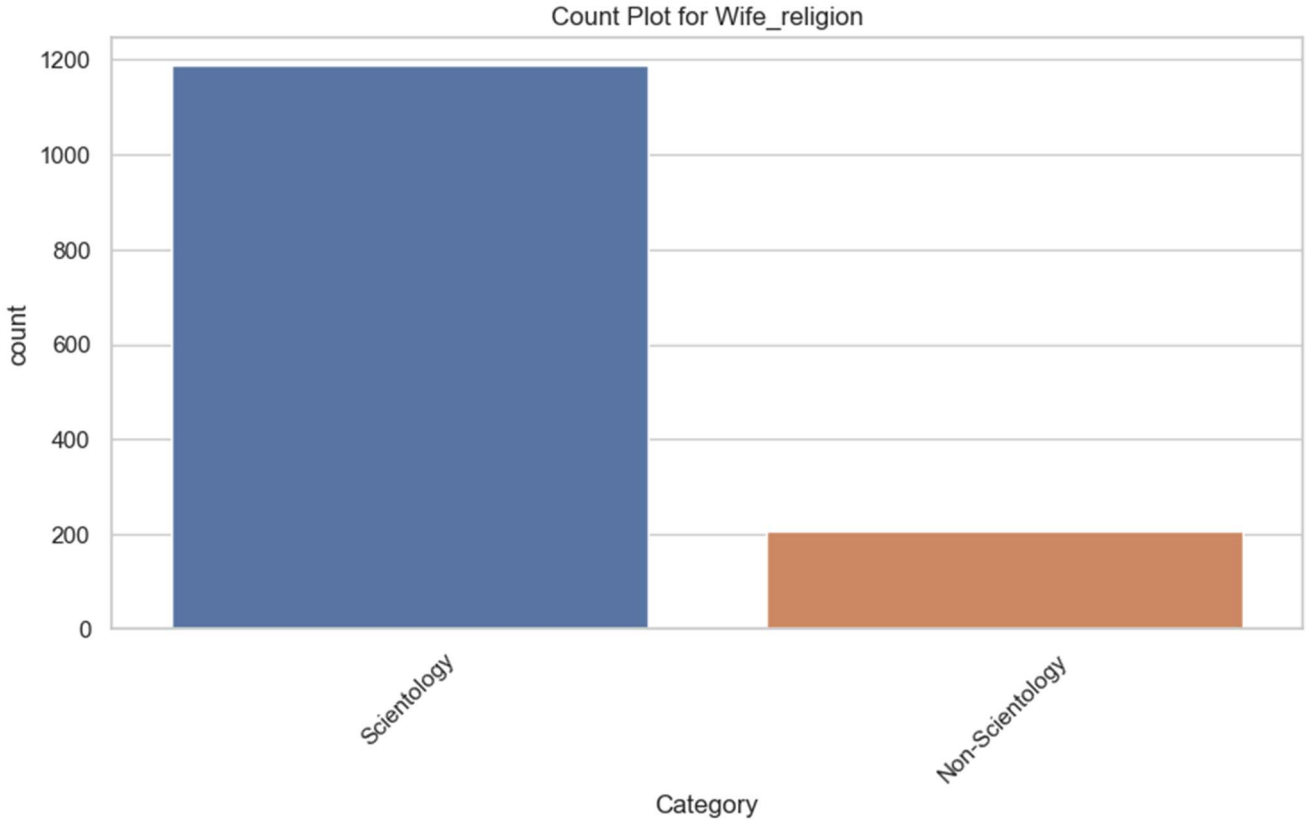


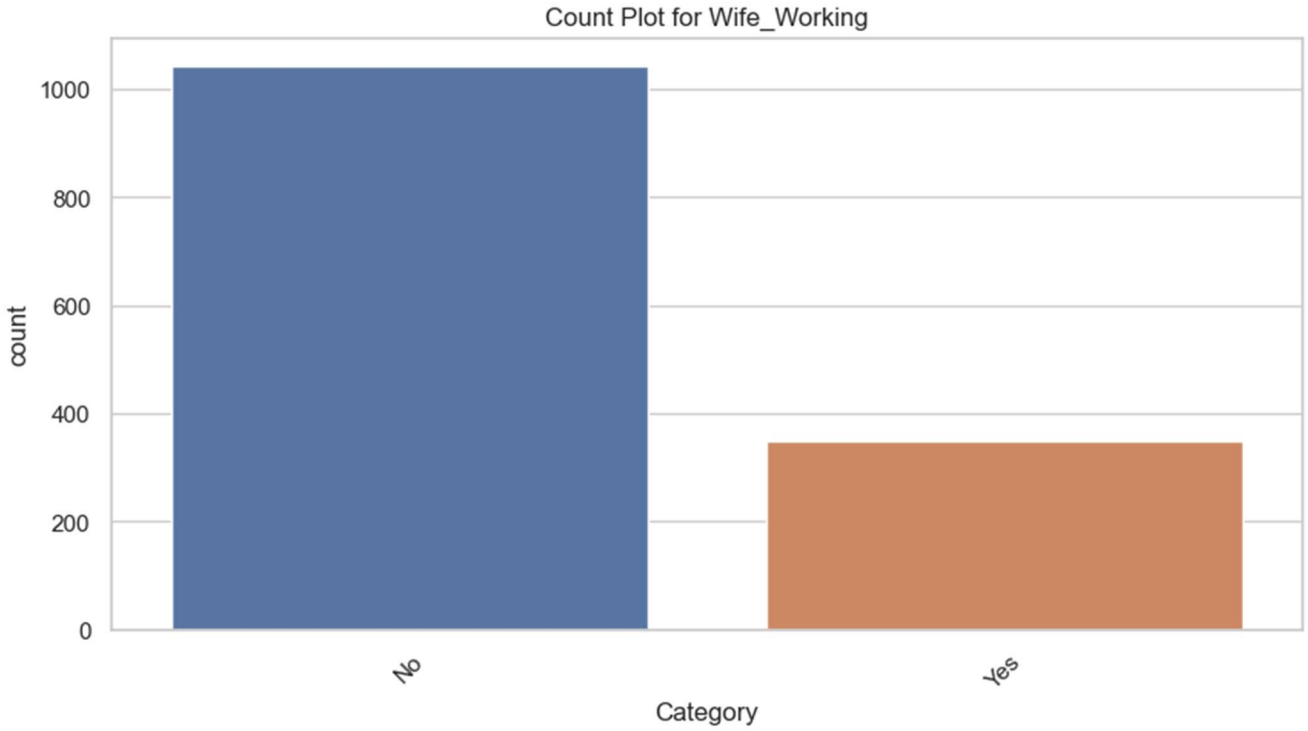
Fig 33 : Heat Map for numeric columns of Contraceptive

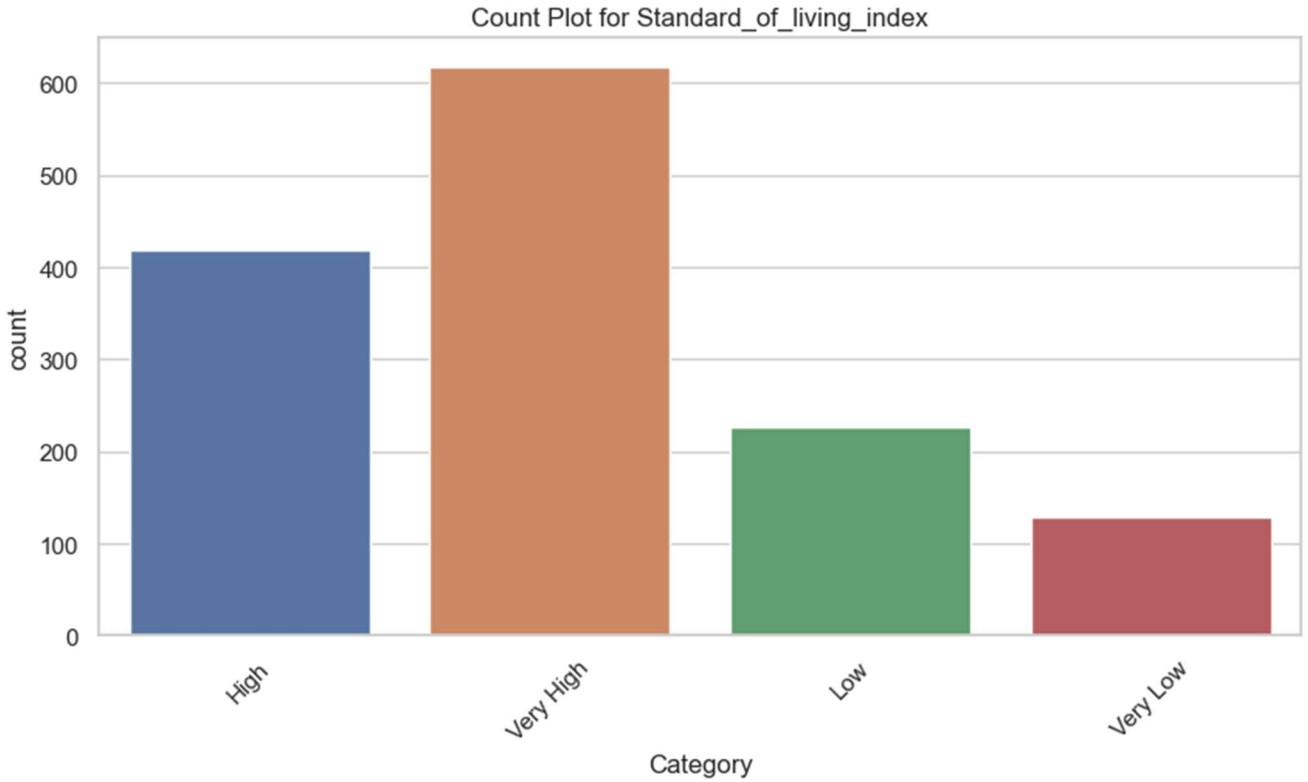
➢ **Count plots for each categorical column**




Count Plot for Husband_education
count
800
700
600
500
400
300
200
100
0
Secondary
Primary
Tertiary
Uneducated
Category


Count Plot for Wife_religion
count
1200
1000
800
600
400
200
0
Scientology
Non-Scientology
Category


Count Plot for Wife_Working
count
1000
800
600
400
200
0
No
Yes
Category


Count Plot for Standard_of_living_index
count
600
500
400
300
200
100
0
High
Very High
Low
Very Low
Category

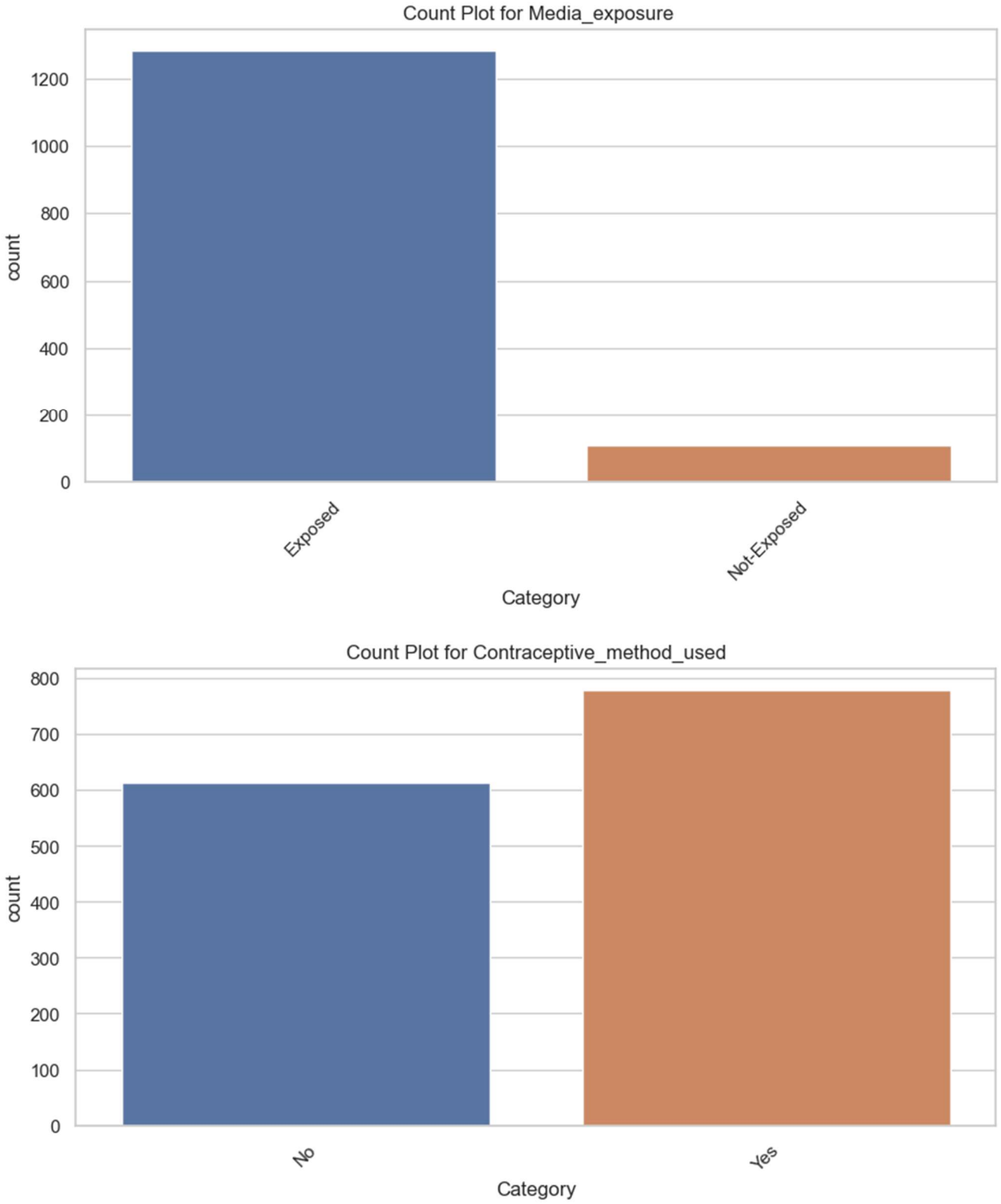
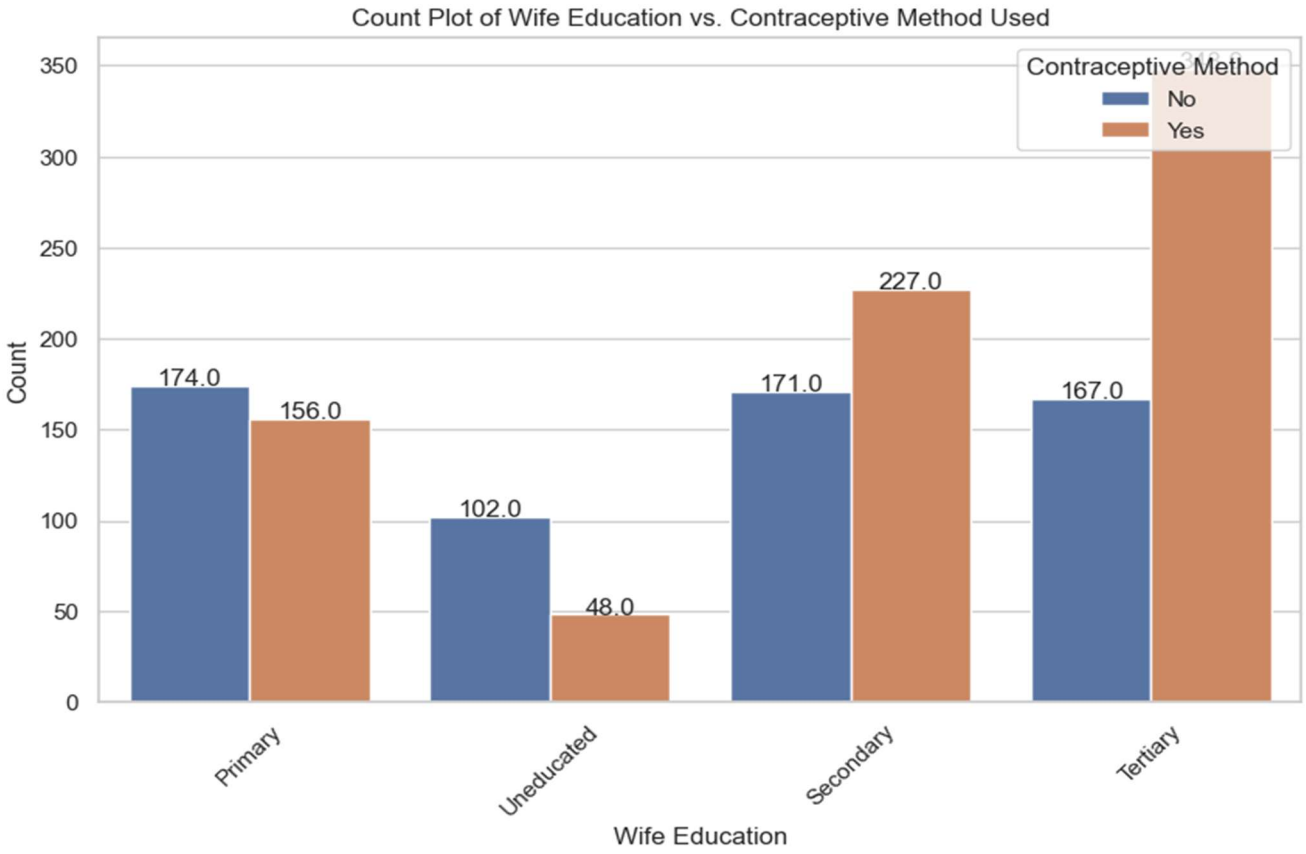
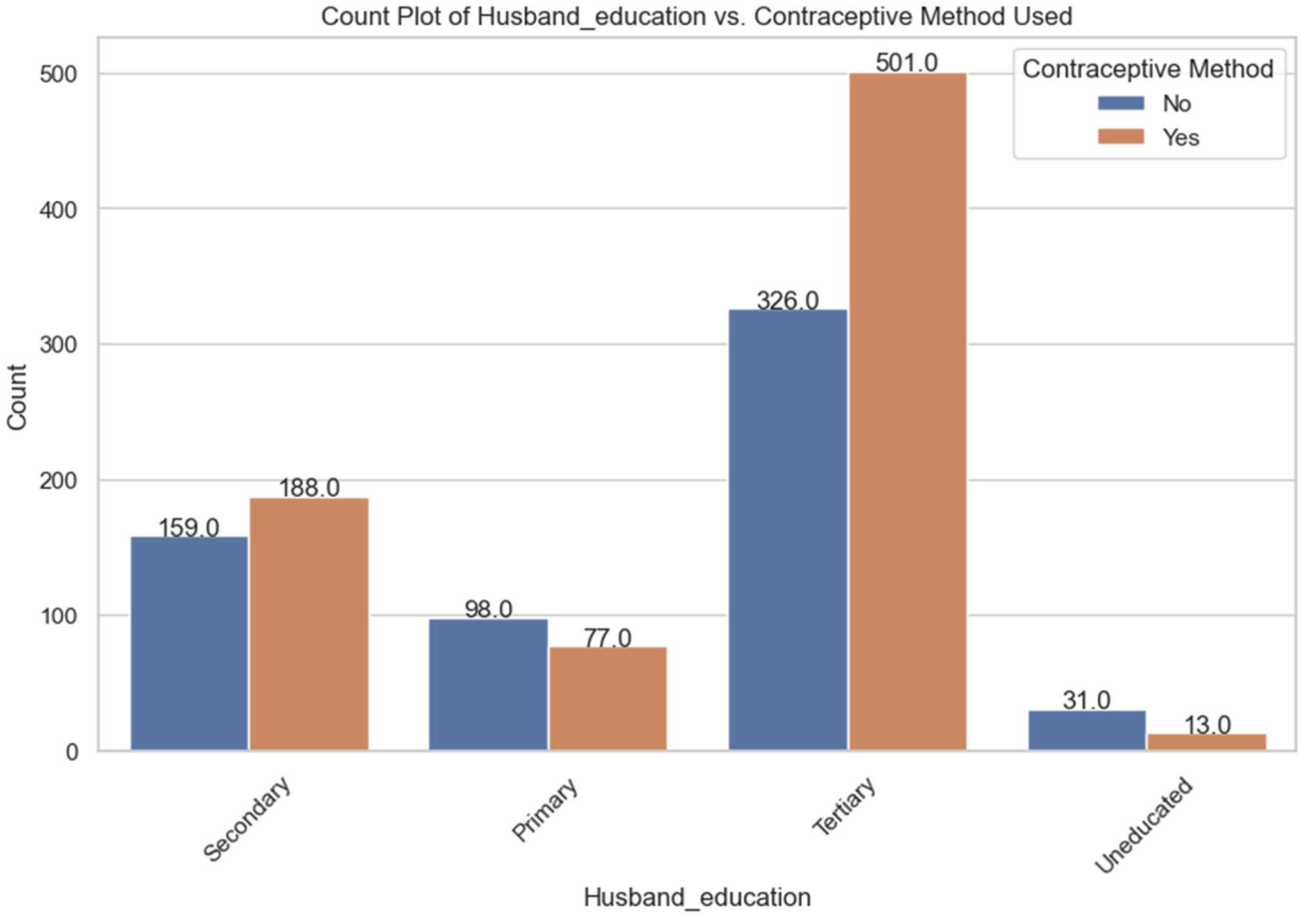
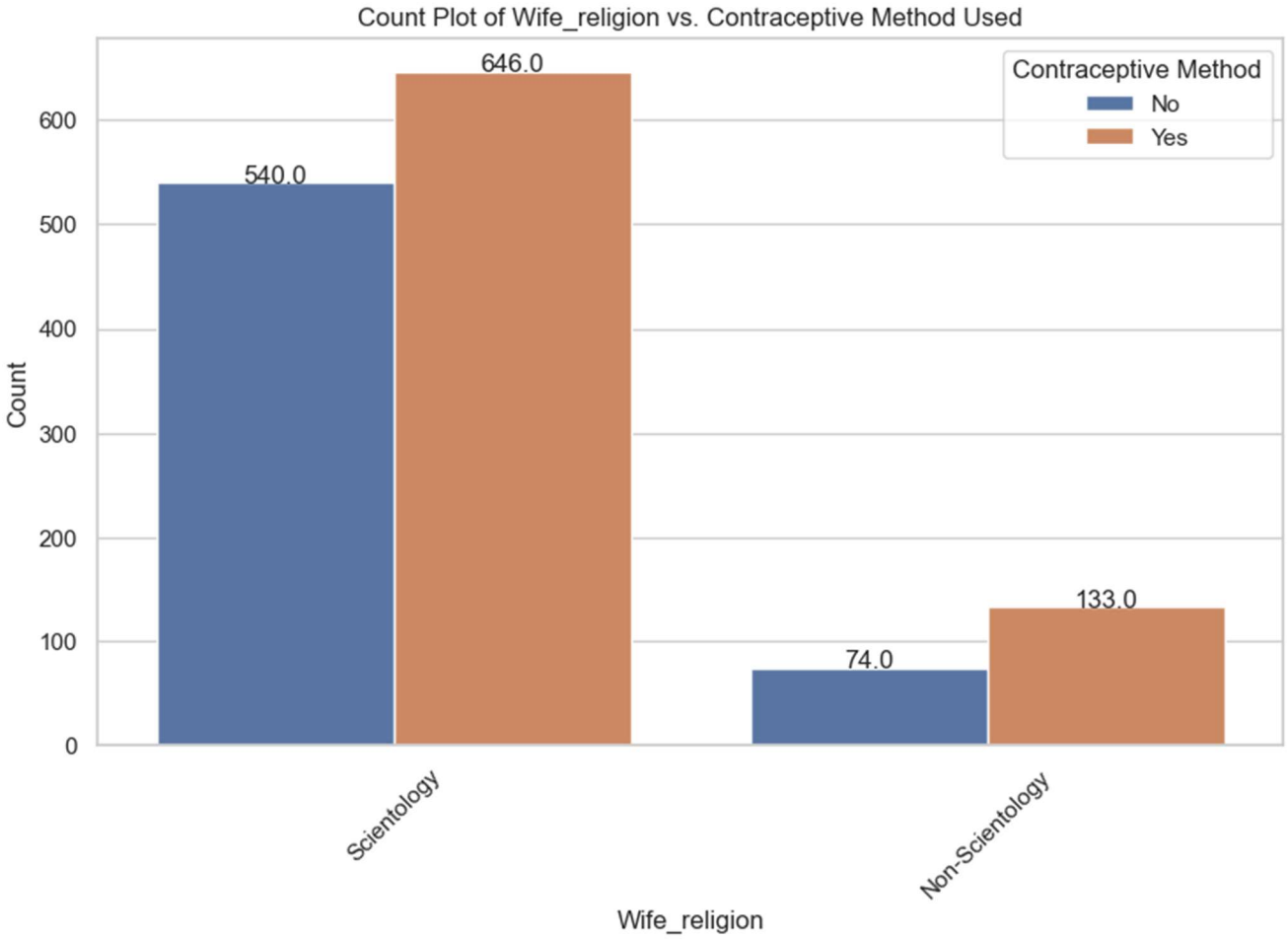
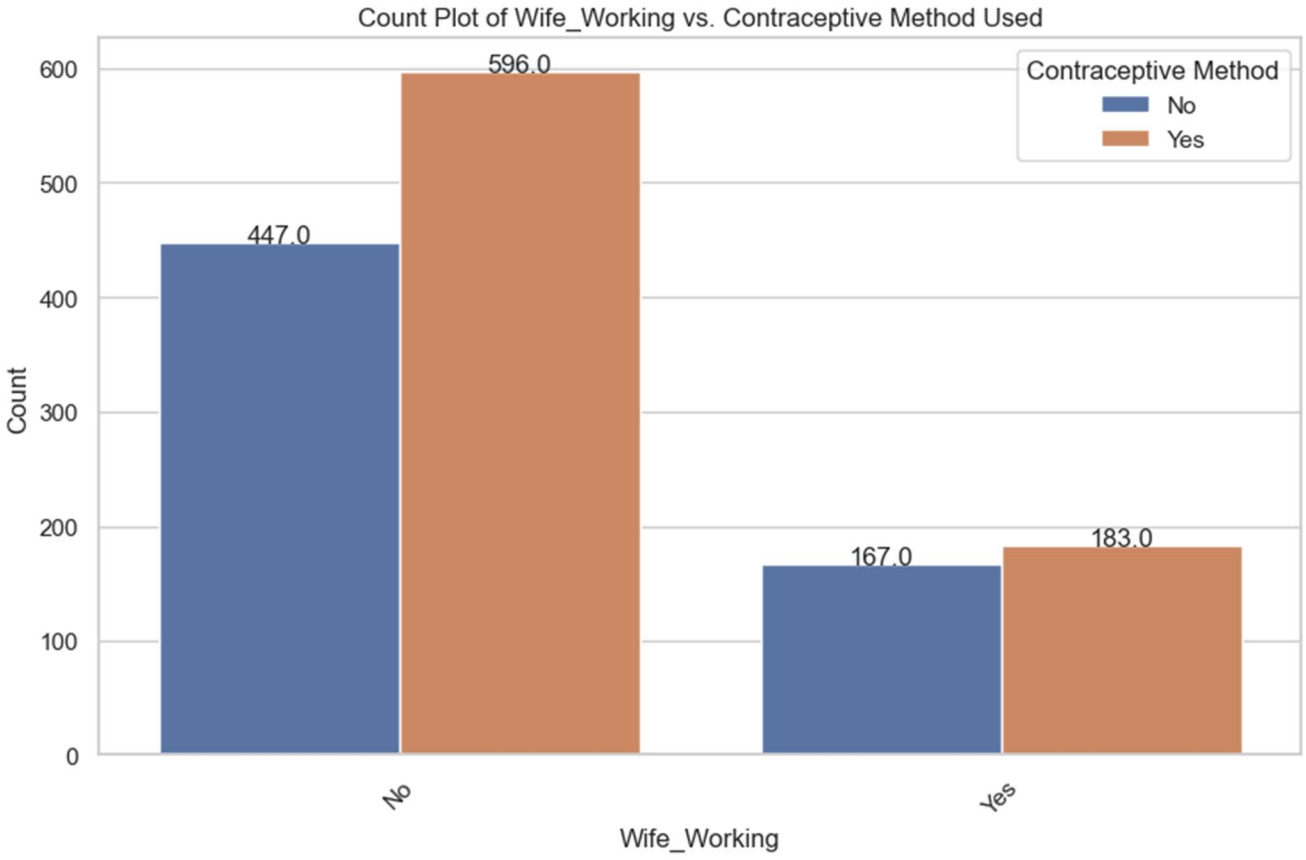


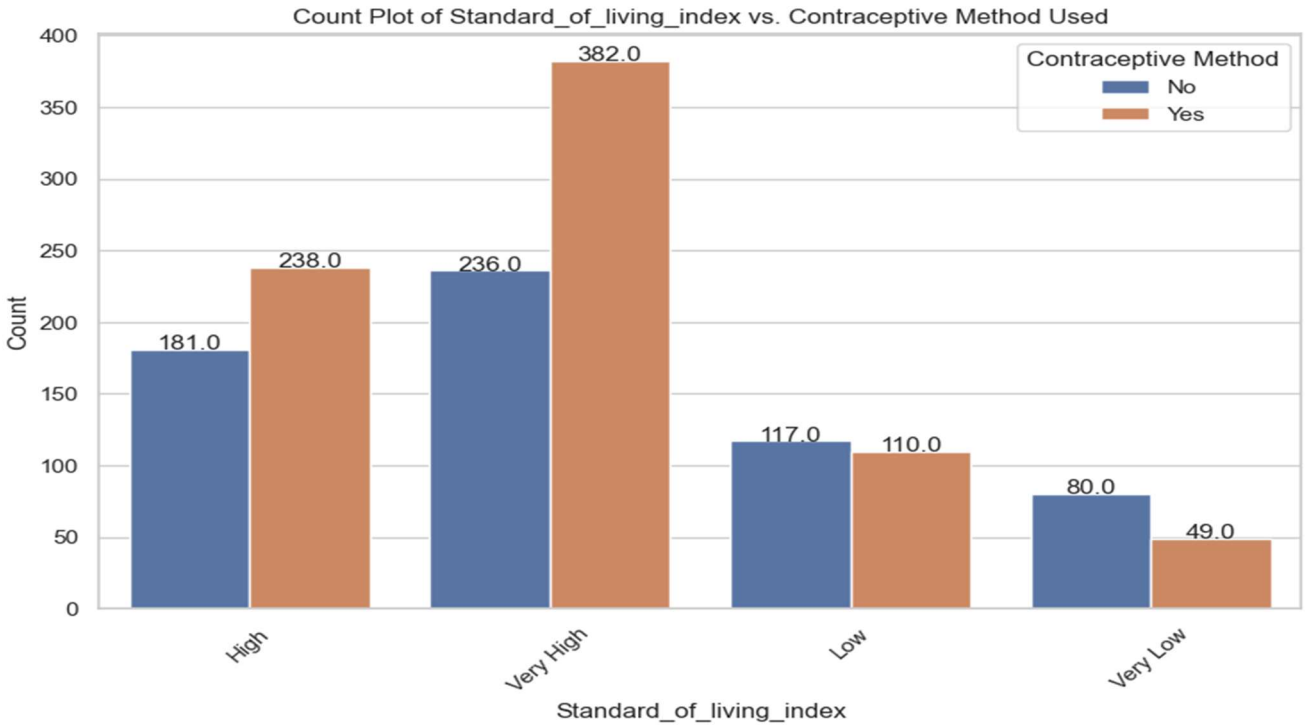
Fig 34 : Count plot for categorical column of Contraceptive

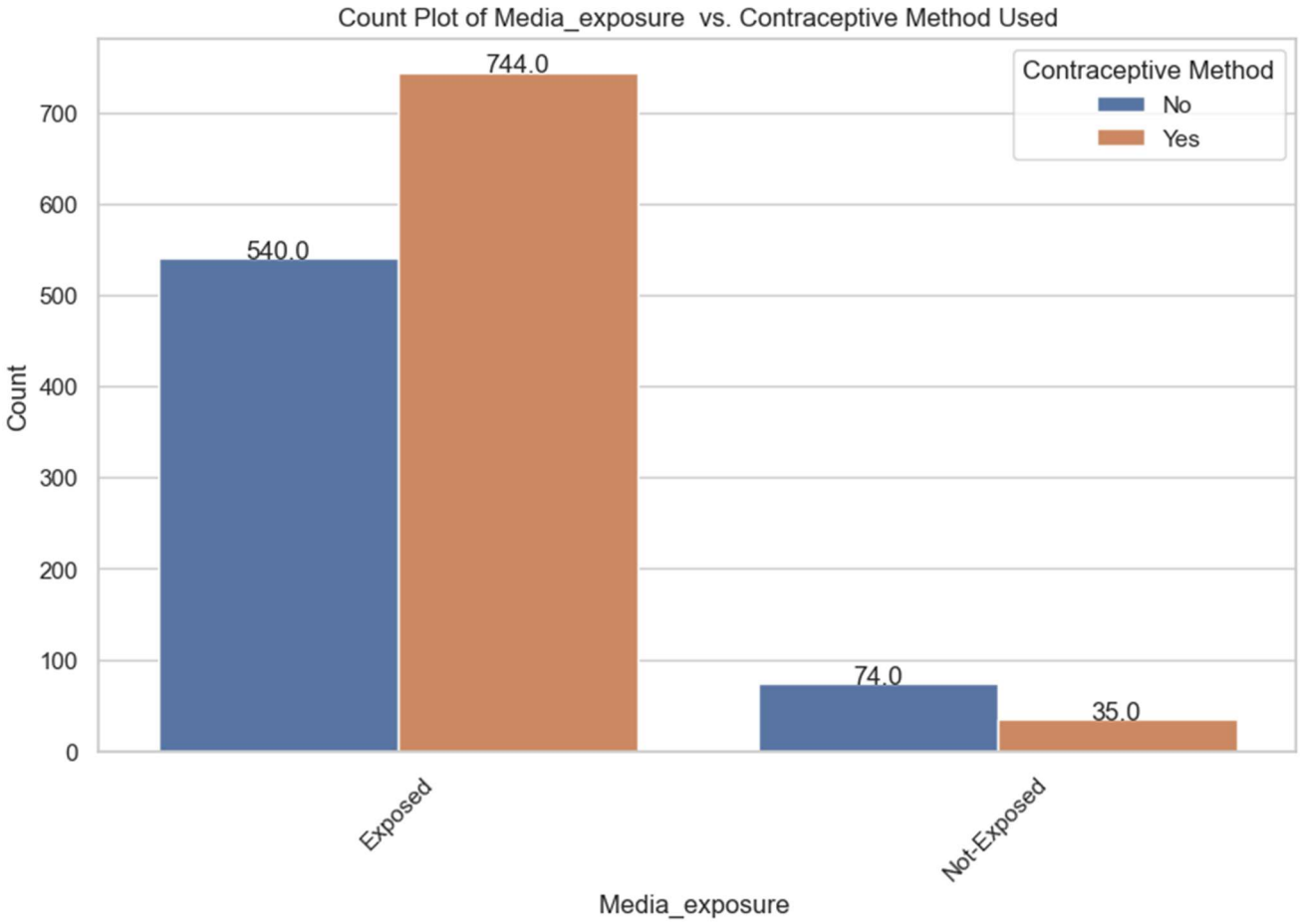

Count Plot of Wife Education vs. Contraceptive Method Used
Contraceptive Method
No
Yes
Count
Wife Education
Primary
Uneducated
Secondary
Tertiary
174.0
156.0
102.0
48.0
171.0
227.0
167.0

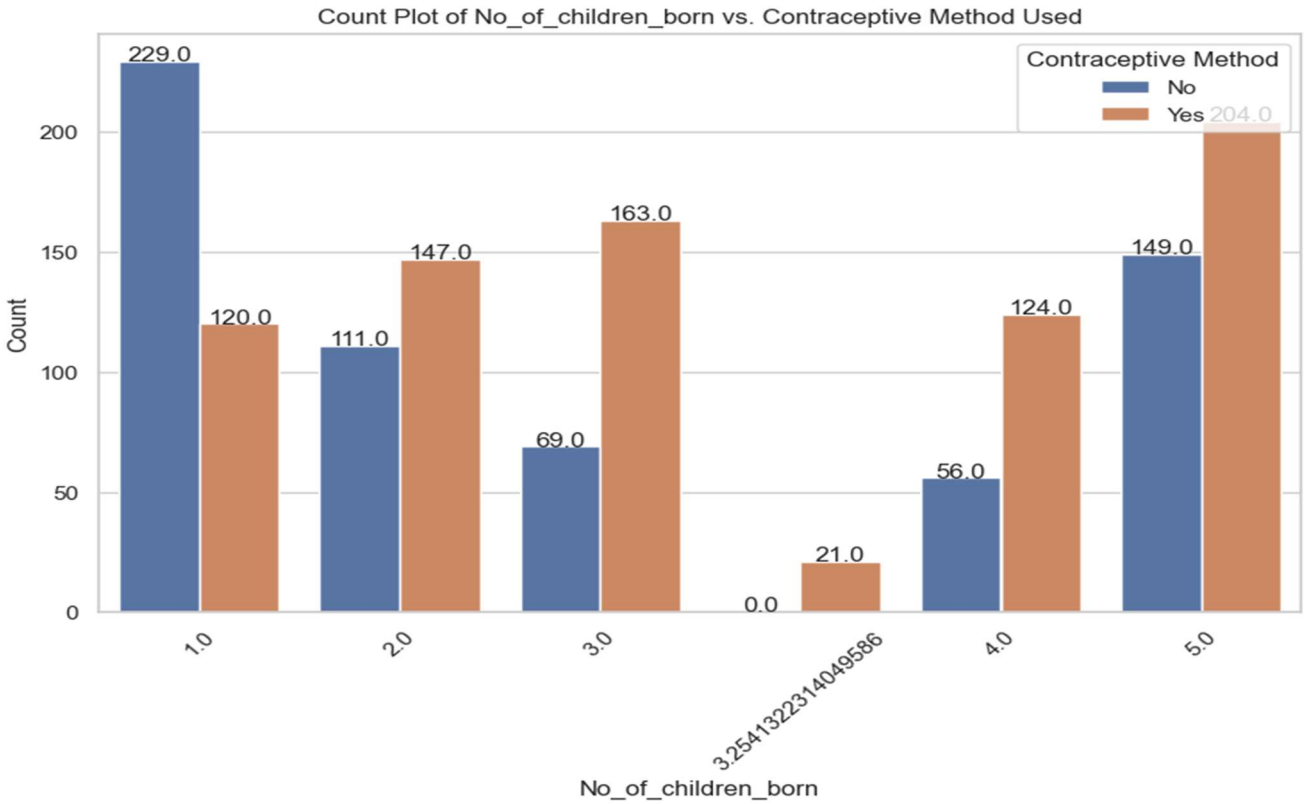

Count Plot of Husband_education vs. Contraceptive Method Used
Contraceptive Method
No
Yes
Count
Husband_education
Secondary
Primary
Tertiary
Uneducated
159.0
188.0
98.0
77.0
326.0
501.0
31.0
13.0

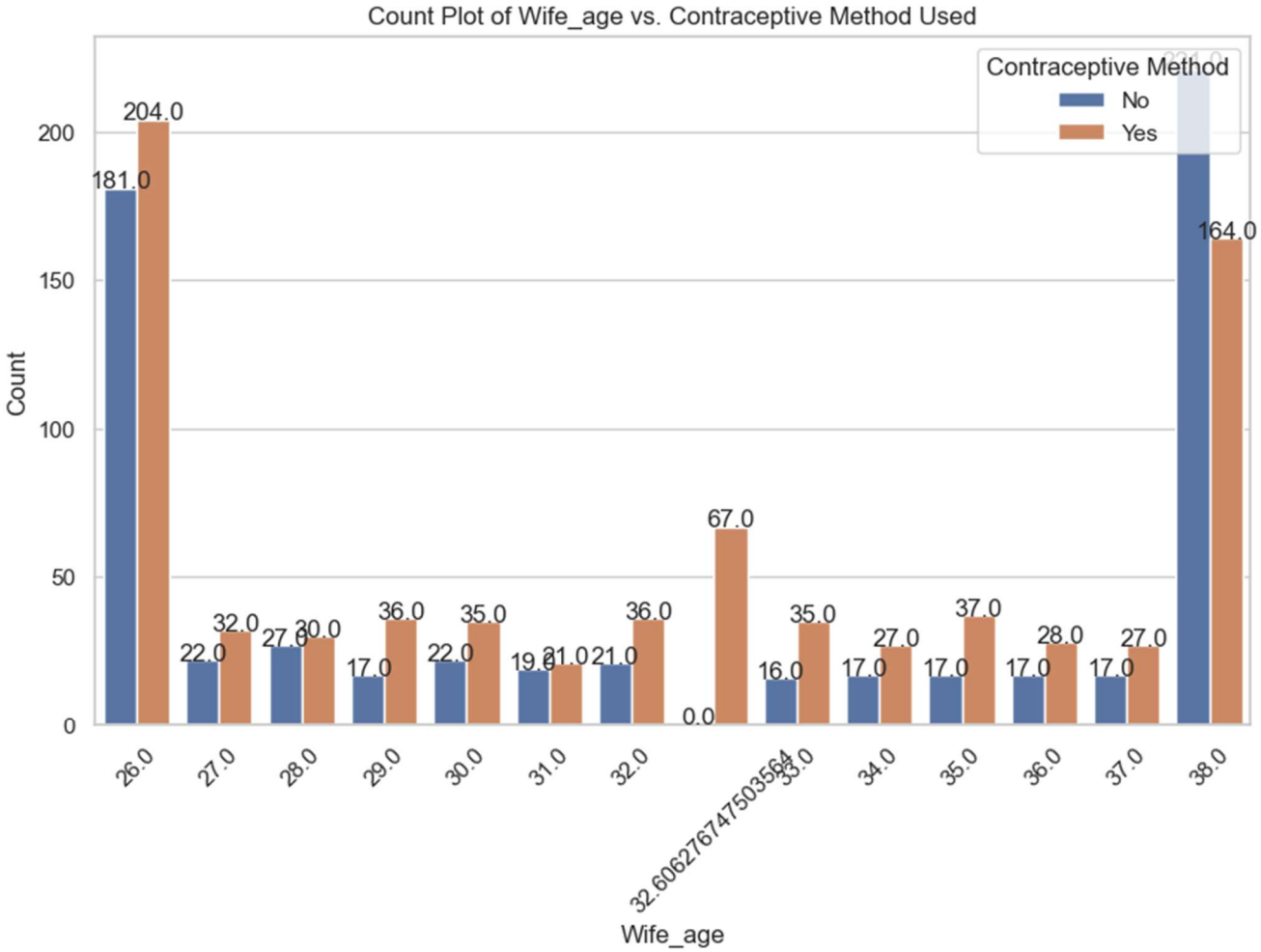

Count Plot of Wife_religion vs. Contraceptive Method Used
Contraceptive Method
No
Yes
646.0
540.0
133.0
74.0
Count
0
100
200
300
400
500
600
Scientology
Non-Scientology
Wife_religion


Count Plot of Wife_Working vs. Contraceptive Method Used
Contraceptive Method
No
Yes
596.0
447.0
183.0
167.0
Count
0
100
200
300
400
500
600
No
Yes
Wife_Working


Count Plot of Standard_of_living_index vs. Contraceptive Method Used
Contraceptive Method
No
Yes
Count
400
350
300
250
200
150
100
50
0
181.0
238.0
236.0
382.0
117.0
110.0
80.0
49.0
High
Very High
Low
Very Low
Standard_of_living_index


Count Plot of Media_exposure vs. Contraceptive Method Used
Contraceptive Method
No
Yes
Count
700
600
500
400
300
200
100
0
540.0
744.0
74.0
35.0
Exposed
Not-Exposed
Media_exposure


Count Plot of No_of_children_born vs. Contraceptive Method Used
Contraceptive Method
No
Yes
Count
No_of_children_born
229.0
120.0
111.0
147.0
69.0
163.0
0.0
21.0
56.0
124.0
149.0
204.0
1.0
2.0
3.0
3.2541322314049586
4.0
5.0


Count Plot of Wife_age vs. Contraceptive Method Used
Contraceptive Method
No
Yes
Count
Wife_age
181.0
204.0
22.0
32.0
27.0
30.0
17.0
36.0
22.0
35.0
19.0
21.0
21.0
36.0
0.0
67.0
16.0
35.0
17.0
27.0
17.0
37.0
17.0
28.0
17.0
27.0
164.0
26.0
27.0
28.0
29.0
30.0
31.0
32.0
32.606276747503564
33.0
34.0
35.0
36.0
37.0
38.0

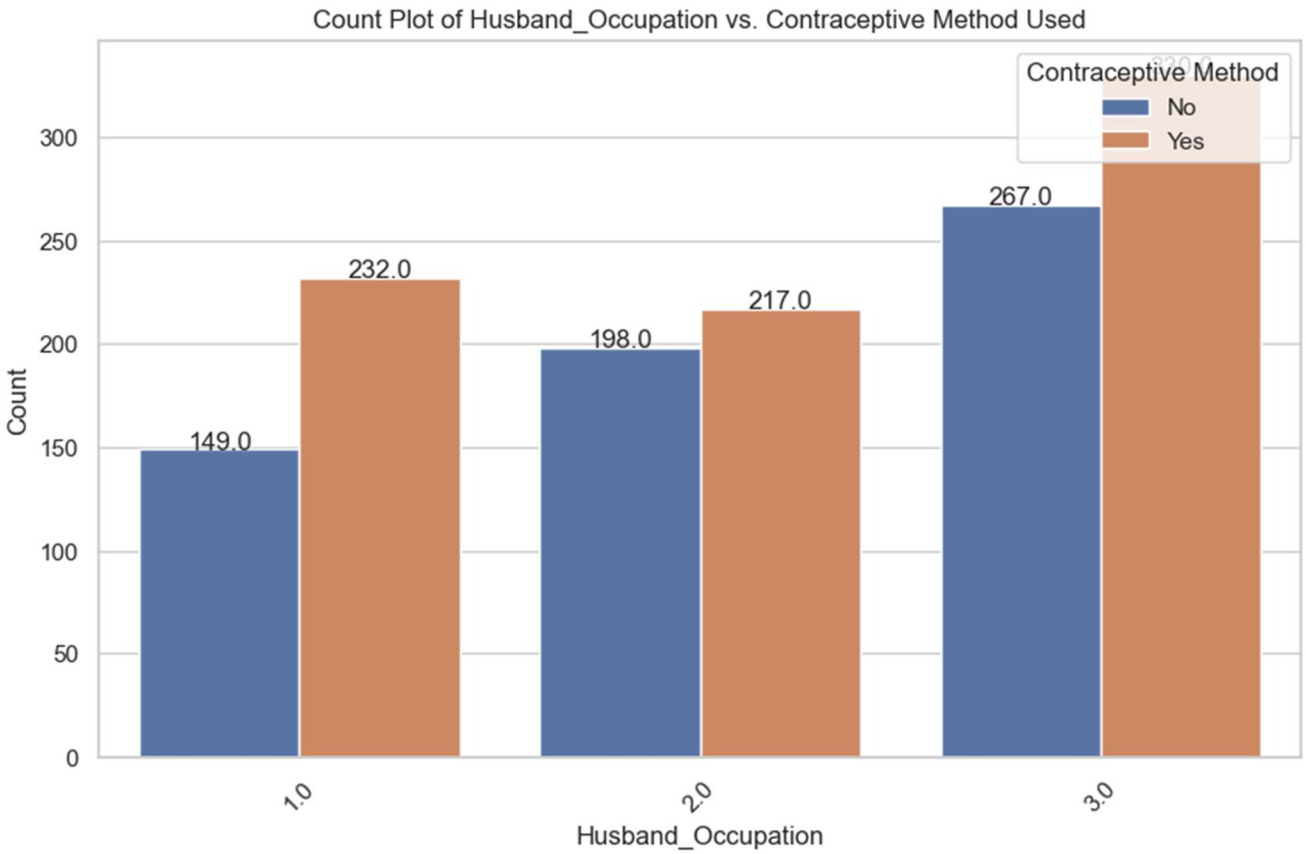


Fig 35 : Count plot for comparison of columns of Contraceptive

Insights :

1. Data consists of both categorical and numerical values.
2. There are total of 1473 rows and 10 columns in the dataset. Out of 22, 7 columns are of object type,1 columns of integer type and remaining 2 are of foat type data.
3. 'contraceptive used' is the target variable and all other are predector variables.
4. Looking into the fields in the univariate analysis, we see outliers is present only in the field number_of_children.
5. Looking in to the boxplot between target variable contraceptive method used and the no_of_children_born, we see that , No_of_children_born is high in the case of use of contracepton used.
6. Bivariate and multivariate analysis indicates that there is strong positive correlation between the fields wife_age and no_of_children_born.
7. We also notice that there are 80 duplicates records in the given data set and has been removed.
8. Null values identified has been imputed with mean.

2.2 Do not scale the data. Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Logistic Regression and LDA (linear discriminant analysis) and CART.

- **Encode Categorical Variables**

Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure	Co
26.0	0	1	3.0	1	0	2.0	0	0	
38.0	3	1	5.0	1	0	3.0	2	0	
38.0	0	1	5.0	1	0	3.0	2	0	
38.0	1	0	5.0	1	0	3.0	0	0	
36.0	1	1	5.0	1	0	3.0	1	0	

Fig 36 : Top 5 Encoded data of dataset Contraceptive

- **Split the Data into Train and Test Sets (70:30)**
- **Apply Logistic Regression**

```
Evaluate Logistic Regression
lr_accuracy =  0.65311004784689
lr_report
              precision    recall  f1-score   support

           0       0.67      0.43      0.52       186
           1       0.65      0.83      0.73       232

    accuracy                           0.65       418
   macro avg       0.66      0.63      0.63       418
weighted avg       0.66      0.65      0.64       418

lr_confusion
 [[ 80 106]
 [ 39 193]]
```

Fig 37 : Evaluate Logistic Regression Encoded data of dataset Contraceptive

Insights:

- The model shows better performance in predicting class 1 (positive class) with a higher precision, recall, and F1-score compared to class 0 (negative class).
- The model is better at identifying actual class 1 instances (higher recall) but is less precise in its predictions for class 1.
- For class 0, the model has moderate precision but struggles with recall, meaning it incorrectly classifies a significant portion of actual class 0 instances.
- The overall model accuracy is 65.31%, which indicates that it is better than random guessing, but there is room for improvement, especially in balancing the precision and recall for both classes.

- Depending on the specific problem and its requirements, you may need to fine-tune the model or consider other machine learning algorithms to improve its performance, especially if classifying both classes accurately is crucial.

➢ **Apply Linear Discriminant Analysis (LDA)**

```
Linear Discriminant Analysis (LDA)
lda_accuracy =  0.6555023923444976
lda_report
              precision    recall  f1-score   support

           0       0.68      0.43      0.53       186
           1       0.65      0.84      0.73       232

    accuracy                           0.66       418
   macro avg       0.66      0.63      0.63       418
weighted avg       0.66      0.66      0.64       418

lda_confusion
 [[ 80 106]
 [ 38 194]]
```

Fig 38 : Evaluate Linear Discriminant Analysis (LDA) Encoded data of dataset Contraceptive

Insights:

- The LDA model shows performance similar to the logistic regression model, with better performance in predicting class 1 (positive class) and lower performance in predicting class 0 (negative class).
- Similar to logistic regression, the model is better at identifying actual class 1 instances (higher recall) but is less precise in its predictions for class 1.
- For class 0, the model has moderate precision but struggles with recall, meaning it incorrectly classifies a significant portion of actual class 0 instances.
- The overall model accuracy is 65.55%, which indicates that it is better than random guessing, but there is room for improvement, especially in balancing the precision and recall for both classes.
- Depending on the specific problem and its requirements, you may need to fine-tune the model or consider other machine learning algorithms to improve its performance, especially if classifying both classes accurately is crucial. Additionally, you can try techniques like feature engineering or adjusting the model's hyperparameters to improve its performance further.

- **Apply Classification and Regression Trees (CART, Decision Tree)**

```
Evaluate CART
cart_accuracy =  0.5980861244019139
cart_report
              precision    recall  f1-score   support

           0       0.54      0.59      0.57       186
           1       0.65      0.60      0.62       232

    accuracy                           0.60       418
   macro avg       0.60      0.60      0.60       418
weighted avg       0.60      0.60      0.60       418

cart_confusion
 [[110  76]
 [ 92 140]]
```

Fig 39 : Evaluate Classification and Regression Tree (CART) Encoded data of dataset Contraceptive

Insights:

- The CART model shows an overall accuracy of 59.81%, which is lower than the logistic regression and LDA models previously evaluated.
- The model's performance is somewhat balanced between precision and recall for both classes, with class 1 (positive class) having slightly higher precision, while class 0 (negative class) has slightly higher recall.
- The F1-scores for both classes are moderate, indicating a reasonable balance between precision and recall.
- The confusion matrix shows that the model correctly predicts a significant portion of class 0 instances but struggles with class 1 instances, as indicated by the higher number of false negatives for class 1.
- While the CART model performs reasonably well, there may be room for improvement, and you could consider experimenting with different tree depths, pruning strategies, or other decision tree variants to fine-tune its performance.
- Depending on the specific problem and its requirements, you might also want to compare the CART model's performance with other machine learning algorithms to choose the one that best suits your needs.

2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model Final Model: Compare Both the models and write inference which model is best/optimized.

- **Plot ROC curve and calculate ROC AUC for Logistic Regression**

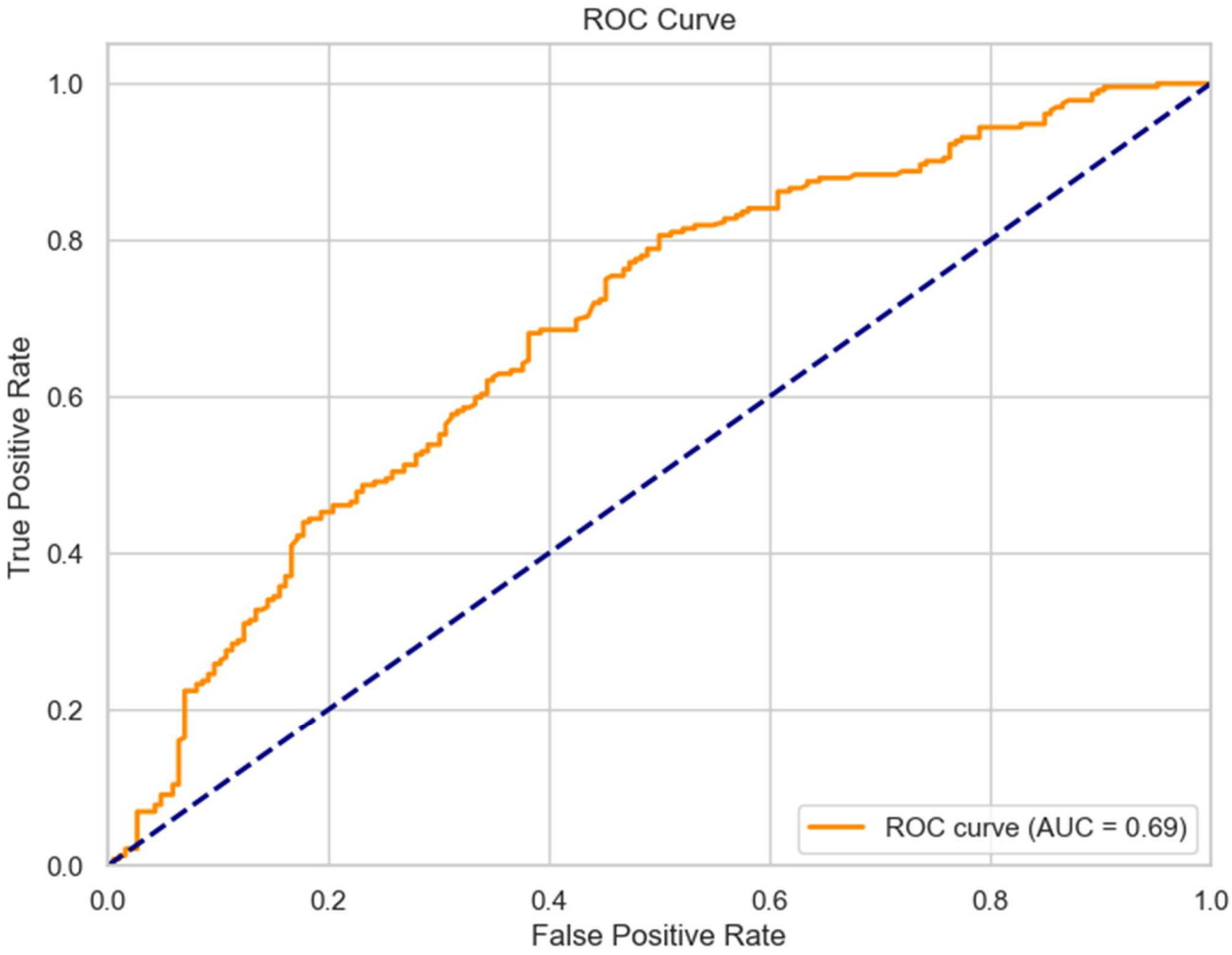


Fig 40 : ROC AUC for Logistic Regression

➢ **Plot ROC curve and calculate ROC AUC for LDA**

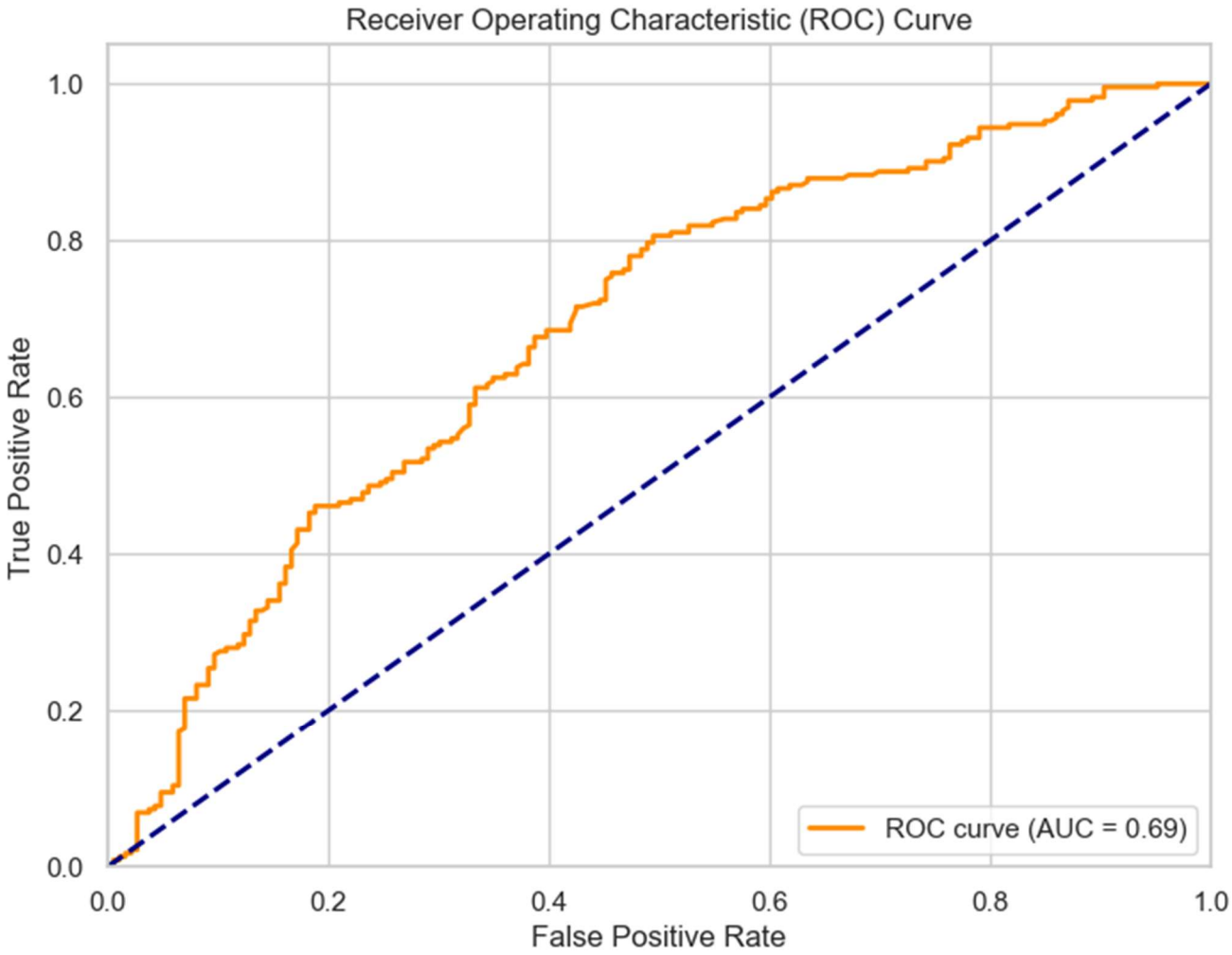


Fig 41 : ROC AUC for LDA

Insights:

1. Accuracy: Both models have similar accuracy, with LDA slightly outperforming LR by a very small margin.
2. Precision and Recall: Both models have similar precision and recall values for class 0 and class 1. There are no significant differences in these metrics between the two models.
3. F1-Score: F1-scores are also quite similar for both models in both classes. Both models have a higher F1-score for class 1 compared to class 0.
4. ROC AUC: The ROC AUC values for both models are very close, with LDA having a slightly higher value. However, the difference is minimal.

Conclusion:

Based on the provided evaluation metrics, Logistic Regression (LR) and Linear Discriminant Analysis (LDA) perform very similarly on your dataset. Both models have almost identical accuracy, precision, recall, F1-scores, and ROC AUC scores.

2.4 Inference: Basis on these predictions, what are the business insights and recommendations. Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

Data Understanding:

- The dataset contains both categorical and numerical values.
- There are 1473 rows and 10 columns in the dataset.
- The target variable is 'contraceptive used,' and the other columns are predictor variables.
- Out of 10 columns, 7 are of object type, 1 is of integer type, and 2 are of float type.
- Outliers are detected only in the 'number_of_children' field.
- There is a strong positive correlation between 'wife_age' and 'number_of_children_born.'
- Duplicate records (80 in total) have been identified and removed.
- Null values have been imputed with the mean.

Model Evaluation (Logistic Regression):

- The model performs better in predicting class 1 (positive class) with higher precision, recall, and F1-score compared to class 0 (negative class).
- It excels at identifying actual class 1 instances (higher recall) but is less precise in its predictions for class 1.
- For class 0, the model has moderate precision but struggles with recall.
- The overall model accuracy is 65.31%, indicating it's better than random guessing but with room for improvement.

Model Evaluation (Linear Discriminant Analysis - LDA):

- The LDA model shows performance similar to logistic regression.
- Like logistic regression, it excels in predicting class 1 with higher recall but lower precision.
- For class 0, it has moderate precision but struggles with recall.
- The overall model accuracy is 65.55%, slightly better than logistic regression.

Model Evaluation (CART - Classification and Regression Trees):

- The CART model achieves an accuracy of 59.81%, lower than both logistic regression and LDA.
- It shows a more balanced performance between precision and recall for both classes.

- The F1-scores for both classes are moderate.
- The model correctly predicts a significant portion of class 0 instances but struggles with class 1.

Model Comparison (Logistic Regression vs. LDA):

- Both LR and LDA have similar accuracy, precision, recall, F1-scores, and ROC AUC scores.
- There are no significant differences in these metrics between the two models.

Conclusion:

- Logistic Regression and Linear Discriminant Analysis perform similarly on the dataset, with nearly identical metrics.
- The models have reasonable but not outstanding performance, and there is room for improvement.
- Depending on the specific problem requirements, further model fine-tuning, feature engineering, or exploring other algorithms - might be necessary for better results.