Context Summary

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.

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.

Solution:

Head Function

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap
0	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
1	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
2	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
3	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
4	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

5 rows × 22 columns

Tail Function

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freesv
8187	16	12	3009	360	244	1.6	5.81	405250.0	85282.0	8.02	...	55.11	0.6	35.87	47.90	139.28	270.74	CPU_Bound	387	986
8188	4	0	1596	170	146	2.4	1.80	89489.0	41764.0	3.80	...	0.20	0.8	3.80	4.40	122.40	212.60	Not_CPU_Bound	263	1055
8189	16	5	3116	289	190	0.6	0.60	325948.0	52640.0	0.40	...	0.00	0.4	28.40	45.20	60.20	219.80	Not_CPU_Bound	400	969
8190	32	45	5180	254	179	1.2	1.20	62571.0	29505.0	1.40	...	18.04	0.4	23.05	24.25	93.19	202.81	CPU_Bound	141	1022
8191	2	0	985	55	46	1.6	4.80	111111.0	22256.0	0.00	...	0.00	0.2	3.40	6.20	91.80	110.00	CPU_Bound	659	1756

5 rows × 22 columns

Shape Function

```
no.of rows:  8192 no.of columns: 22
```

```
(8192, 22)
```

Duplicate Function

```
Number of duplicate rows = 0
```

```
0
```

Describe Function

	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

Info Function

```
<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
```

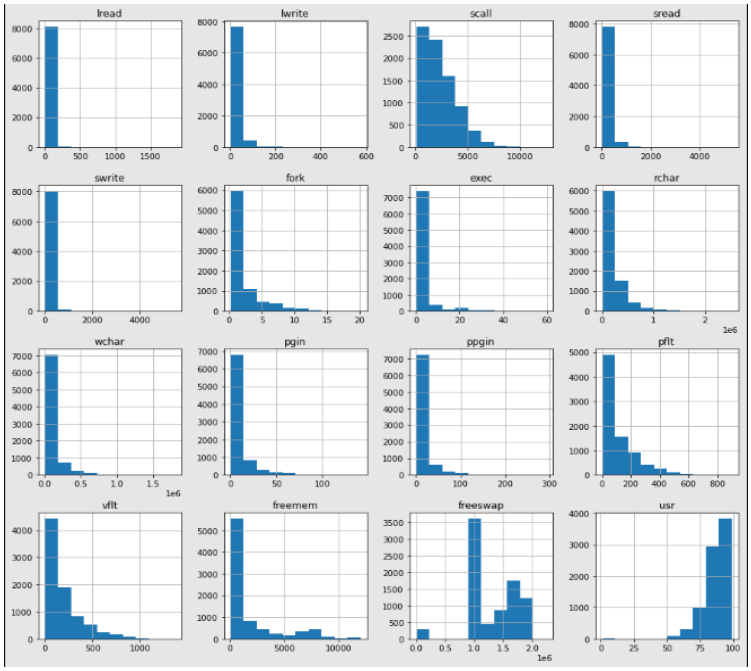
Null Function

```
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
```

Summary:

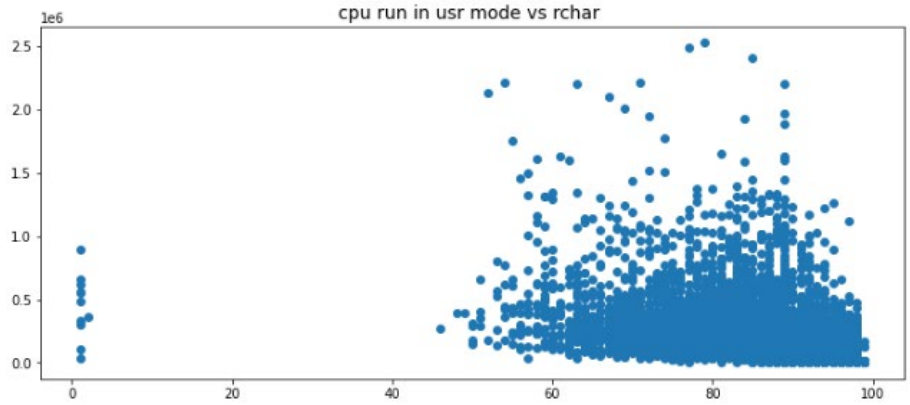
- Database administrator would be checking the size of the dataset using '.shape' which throw output for number of rows and number of columns and using '.info()' for checking the type of variables in the dataset.
- There are 8192 number of rows and 22 columns in the dataset.
- There are 8 in int64, 13 in float64 and 1 in object present in the dataset
- No duplicate records is present in the dataset
- Null value present in the dataset for rchar is having 104 rows & wchar is having 15 rows

Univariate Analysis – Visualization of bar plot

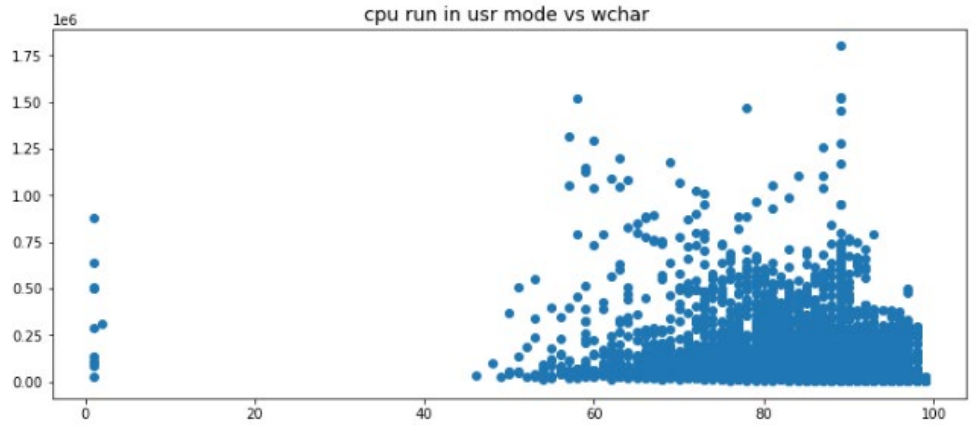


Bivariate Analysis

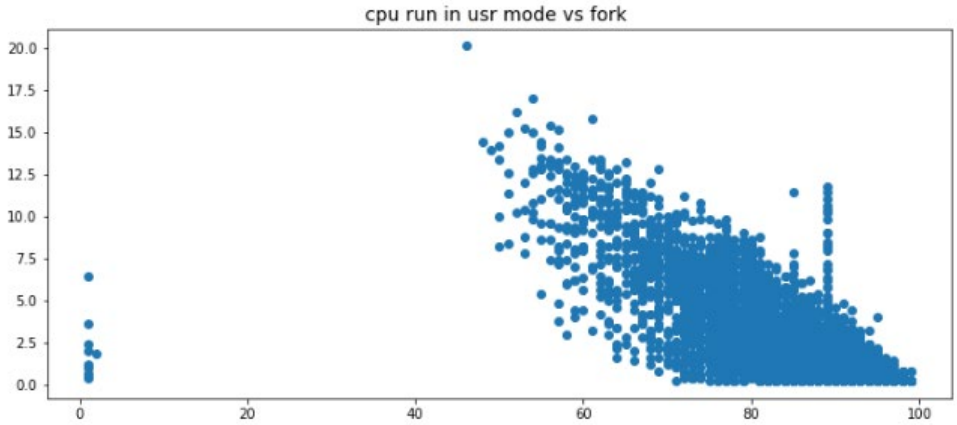
Visualization plot between CPU run in usr mode vs rchar using scatter plot



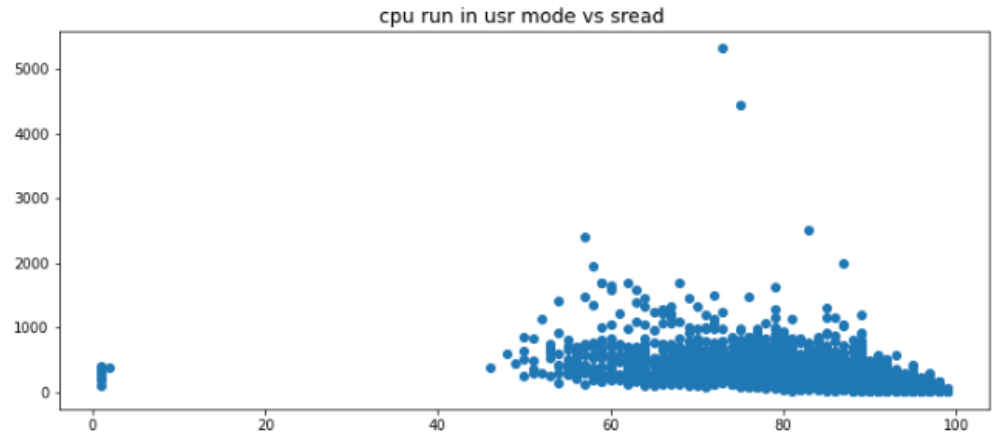
Visualization plot between CPU run in usr mode vs wchar using scatter plot



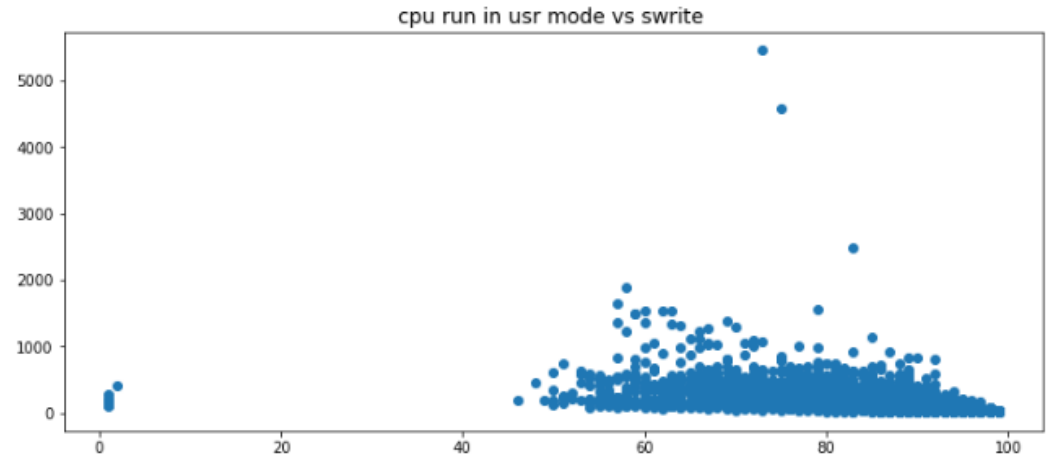
Visualization plot between CPU run in usr mode vs fork using scatter plot



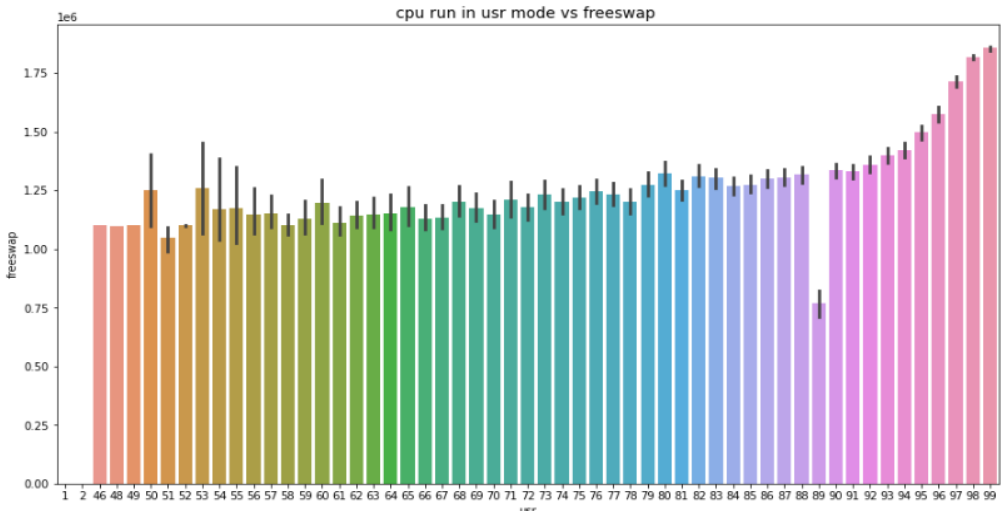
Visualization plot between CPU run in usr mode vs sread using scatter plot



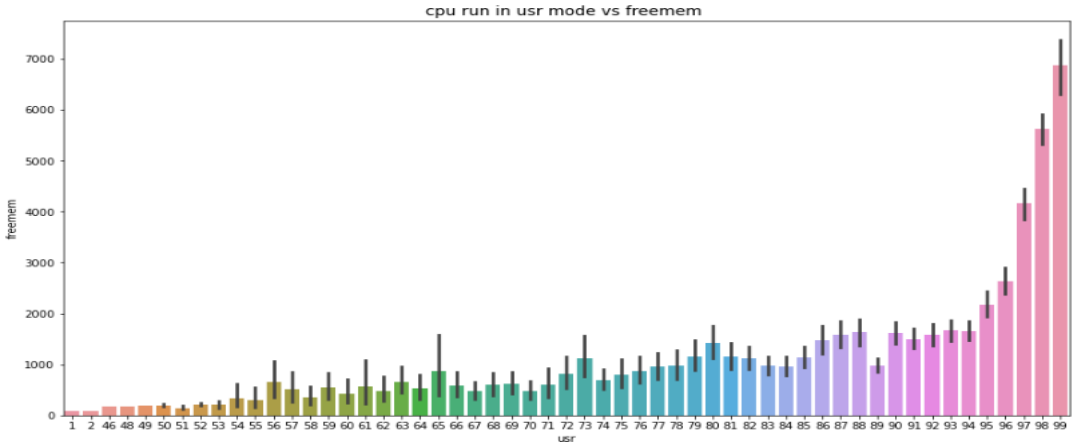
Visualization plot between CPU run in usr mode vs swrite using scatter plot



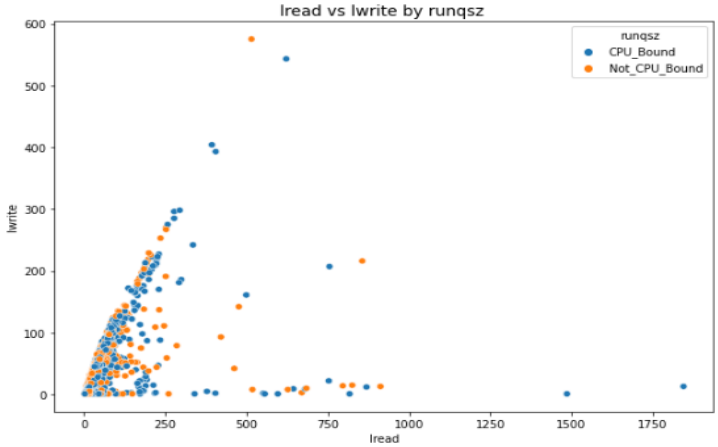
Visualization plot between CPU run in usr mode vs freeswap using bar plot



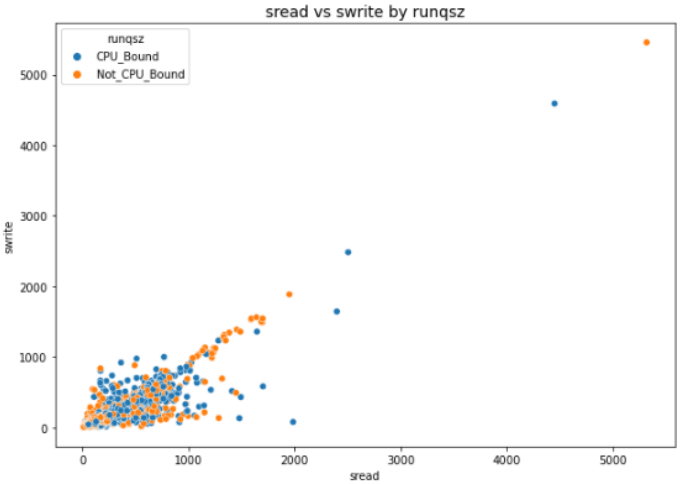
Visualization plot between CPU run in usr mode vs freemam using bar plot



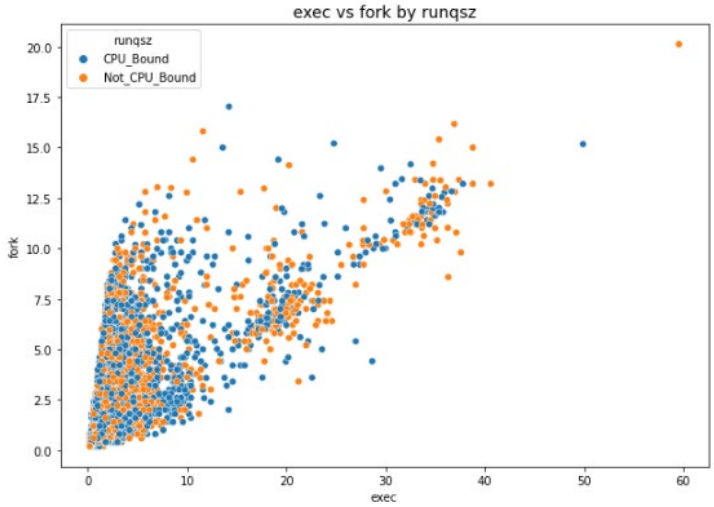
Visualization plot between CPU run in lread vs lwrite by runqsz using scatter plot



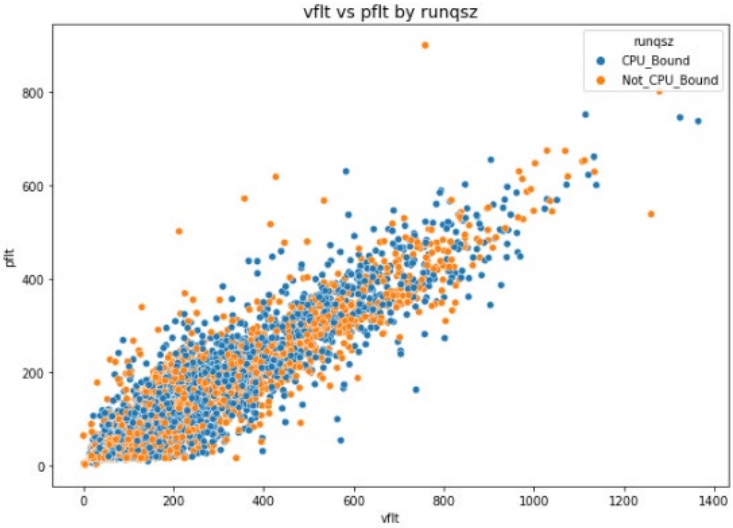
Visualization plot between CPU run in sread mode vs swrite by runqsz using scatter plot



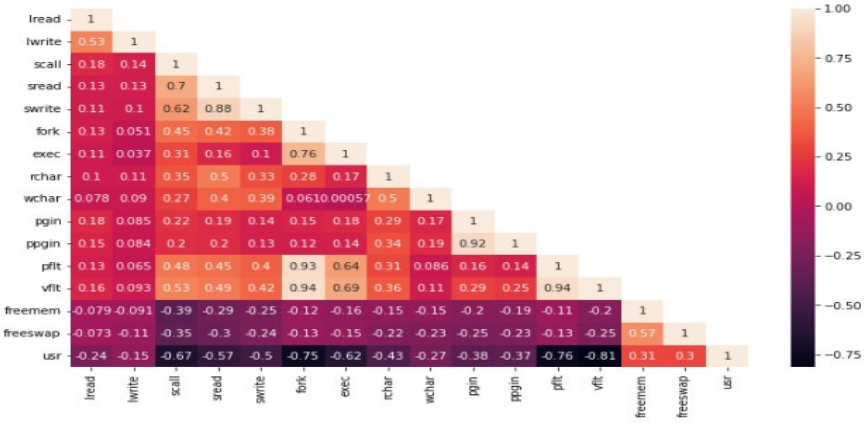
Visualization plot between CPU run in exec mode vs fork by runqsz using scatter plot



Visualization plot between CPU run in vflt mode vs pflt by runqsz using scatter plot



Multivariate Analysis – Visualization of Heat map for Multivariate Analysis



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.

Solution:

Replace the null value with median function

```
# Replacing blank value with Median
median_1 = ca['wchar'].median()
ca['wchar'] = ca['wchar'].replace(np.nan,median_1)
ca['wchar'].isnull().sum()
median_2 = ca['rchar'].median()
ca['rchar'] = ca['rchar'].replace(np.nan,median_2)
ca['rchar'].isnull().sum()
```

```
0
```

To check for the values which are equal to zero

```
* of records with 0 value in lread: 8.24%
* of records with 0 value in lwrite: 32.76%
* of records with 0 value in scall: 0.00%
* of records with 0 value in sread: 0.00%
* of records with 0 value in swrite: 0.00%
* of records with 0 value in fork: 0.26%
* of records with 0 value in exec: 0.26%
* of records with 0 value in rchar: 0.00%
* of records with 0 value in wchar: 0.00%
* of records with 0 value in pgout: 59.55%
* of records with 0 value in ppgout: 59.55%
* of records with 0 value in pgfree: 59.44%
* of records with 0 value in pgscan: 78.71%
* of records with 0 value in atch: 55.85%
* of records with 0 value in pgin: 14.89%
* of records with 0 value in ppgin: 14.89%
* of records with 0 value in pflt: 0.04%
* of records with 0 value in vflt: 0.00%
* of records with 0 value in runqsz: 0.00%
* of records with 0 value in freemem: 0.00%
* of records with 0 value in freeswap: 0.00%
* of records with 0 value in usr: 3.45%
```

Drop function is used to drop columns and check the dataset using head function

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
0	1	0	2147	79	68	0.2	0.2	40671.0	53995.0	1.6	2.6	16.00	26.40	CPU_Bound	4670	1730946	95
1	0	0	170	18	21	0.2	0.2	448.0	8385.0	0.0	0.0	15.63	16.83	Not_CPU_Bound	7278	1869002	97
2	15	3	2162	159	119	2.0	2.4	125473.5	31950.0	6.0	9.4	150.20	220.20	Not_CPU_Bound	702	1021237	87
3	0	0	160	12	16	0.2	0.2	125473.5	8670.0	0.2	0.2	15.60	16.80	Not_CPU_Bound	7248	1863704	98
4	5	1	330	39	38	0.4	0.4	125473.5	12185.0	1.0	1.2	37.80	47.60	Not_CPU_Bound	633	1760253	90

Replace minimum zero value by median function

```
# Replace the 0 value into median
ca.lread.replace(to_replace=0, value=ca.lread.median(),inplace=True)
ca.lwrite.replace(to_replace=0, value=ca.lwrite.median(),inplace=True)
ca.fork.replace(to_replace=0,value=ca.fork.median(),inplace=True)
ca.exec.replace(to_replace=0,value=ca.exec.median(),inplace=True)
ca.pgin.replace(to_replace=0,value=ca.pgin.median(),inplace=True)
ca.ppgin.replace(to_replace=0,value=ca.ppgin.median(),inplace=True)
ca.pflt.replace(to_replace=0,value=ca.pflt.median(),inplace=True)
ca.usr.replace(to_replace=0,value=ca.usr.median(),inplace=True)
```

Describe Function

	count	mean	std	min	25%	50%	75%	max
lread	8192.0	2.013647e+01	53.176752	1.00	3.00	7.0	20.000	1845.00
lwrite	8192.0	1.343384e+01	29.751410	1.00	1.00	1.0	10.000	575.00
scall	8192.0	2.306318e+03	1633.617322	109.00	1012.00	2051.5	3317.250	12493.00
sread	8192.0	2.104800e+02	198.980146	6.00	86.00	166.0	279.000	5318.00
swrite	8192.0	1.500582e+02	160.478980	7.00	63.00	117.0	185.000	5456.00
fork	8192.0	1.886605e+00	2.478264	0.19	0.40	0.8	2.200	20.12
exec	8192.0	2.795074e+00	5.211161	0.19	0.20	1.2	2.800	59.56
rchar	8192.0	1.964728e+05	238446.012054	278.00	34860.50	125473.5	265394.750	2526649.00
wchar	8192.0	9.581275e+04	140728.464118	1498.00	22977.75	46619.0	106037.000	1801623.00
pgin	8192.0	8.694952e+00	13.660319	0.19	1.60	2.8	9.765	141.20
ppgin	8192.0	1.295450e+01	22.006000	0.20	2.00	3.8	13.800	292.61
pflt	8192.0	1.098172e+02	114.403309	0.80	25.15	63.8	159.600	899.80
vflt	8192.0	1.853158e+02	191.000603	0.20	45.40	120.4	251.800	1365.00
freemem	8192.0	1.763456e+03	2482.104511	55.00	231.00	579.0	2002.250	12027.00
freeswap	8192.0	1.328126e+06	422019.426957	2.00	1042623.50	1289289.5	1730379.500	2243187.00
usr	8192.0	8.704346e+01	9.297604	1.00	83.00	89.0	94.000	99.00

Info function:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8192 entries, 0 to 8191
Data columns (total 17 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                 8192 non-null   float64
 8   wchar                 8192 non-null   float64
 9   pgin                  8192 non-null   float64
 10  ppgin                 8192 non-null   float64
 11  pflt                  8192 non-null   float64
 12  vflt                  8192 non-null   float64
 13  freemem               8192 non-null   int64  
 14  freeswap              8192 non-null   int64  
 15  usr                   8192 non-null   int64  
 16  runqsz_Not_CPU_Bound  8192 non-null   uint8  
dtypes: float64(8), int64(8), uint8(1)
memory usage: 1.0 MB
```

Duplicate Function: To check the number of duplicate records present in the dataset

```
Number of duplicate rows = 0
```

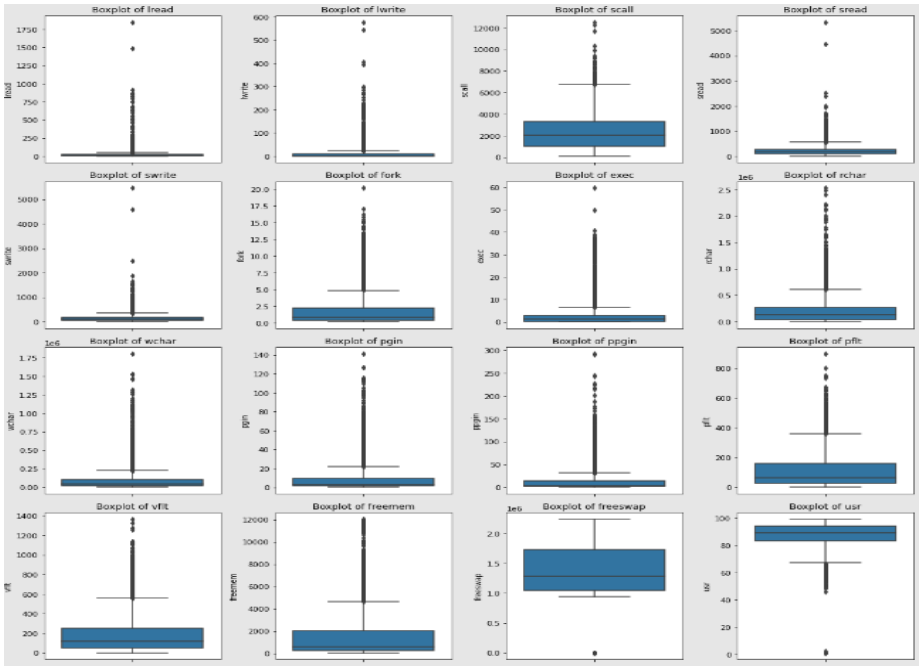
Shape function:

```
no.of rows:  8192 no.of columns: 17
```

Summary:

- Null value is replaced by the median value for rchar & wchar in the dataset.
- The variables columns pgout, ppgout, pgfree, pgscan and atch have at least 50% of the values is having 0.
- Yes we need to drop the columns is having more 50% of values is having 0. The variable columns is dropped from the dataset and columns details are pgout, ppgout, pgfree, pgscan and atch
- The remaining variables columns lread, lwrite, fork, exec, pgin, ppgin, usr & pflt are having minimum values of 0. Needs to clean up the columns
- Treat the columns is having minimum values of 0 with replace value using median function
- Database administrator would be checking the size of the dataset using '.shape' which throw output for number of rows and number of columns and using '.info()' for checking the type of variables in the dataset
- There are 8192 number of rows and 17 columns in the dataset.
- There are 8 in int64, 8 in float64 and 1 in object present in the dataset
- No duplicate records is present in the dataset and No Null value present in the dataset

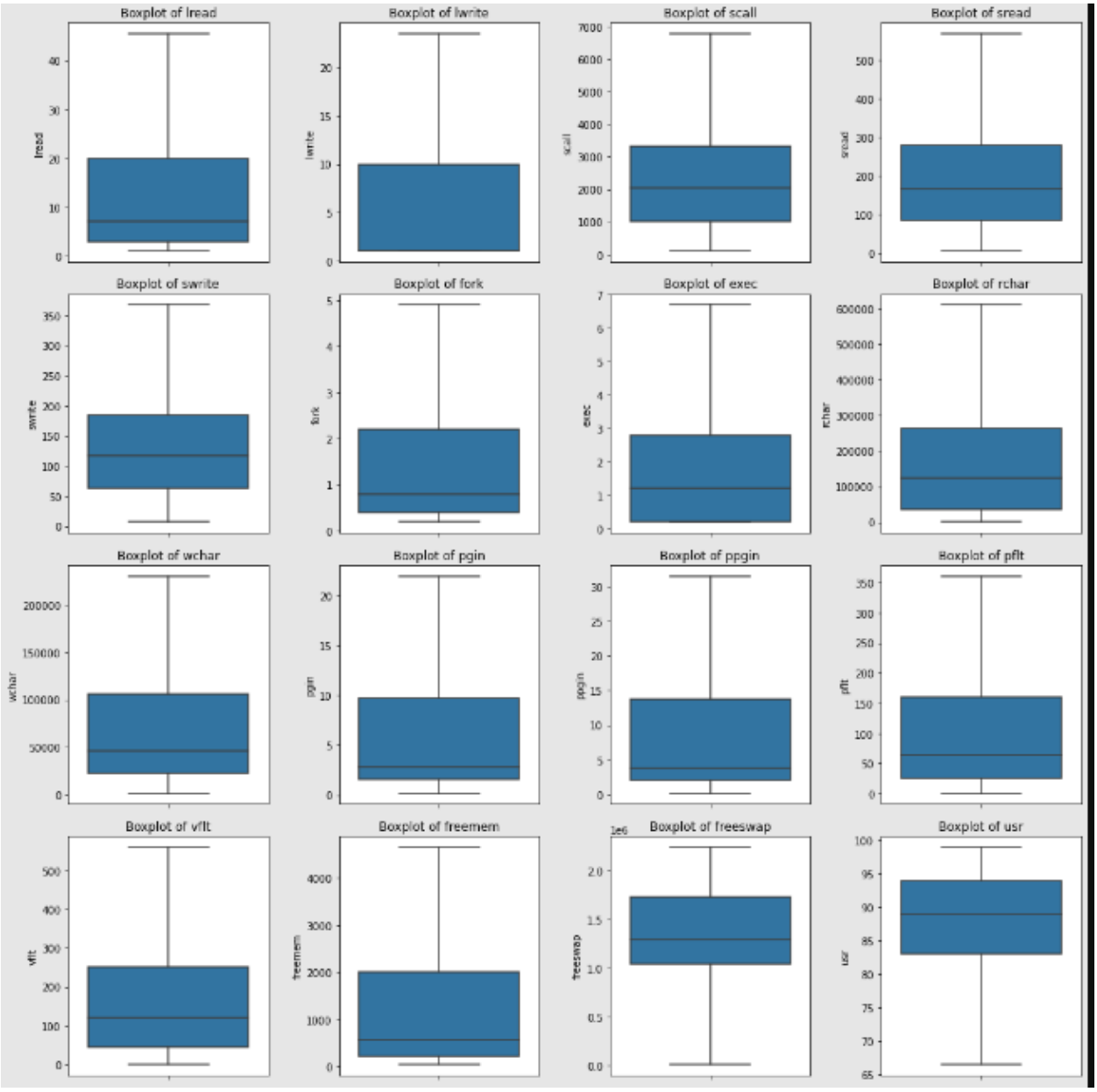
Visualization of the box plot to check the outlier present in the dataset



Summary:

1. Yes outliers have been detected in all items. These outliers value needs to be treated and there are several ways of treating them:
    - Drop the outlier value
    - Replace the outlier value using the IQR

Visualization of box plot after treating outliers



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.

Solution:

Linear Regression Model

```
LinearRegression()
```

Linear Regression using stats models (OLS)

OLS Regression Results

Dep. Variable:	usr	**R-squared:**	0.799
Model:	OLS	**Adj. R-squared:**	0.799
Method:	Least Squares	**F-statistic:**	1421.
Date:	Wed, 23 Nov 2022	**Prob (F-statistic):**	0.00
Time:	09:50:46	**Log-Likelihood:**	-16397.
No. Observations:	5734	**AIC:**	3.283e+04
Df Residuals:	5717	**BIC:**	3.294e+04
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	98.0406	0.288	340.806	0.000	97.477	98.605
lread	-0.0119	0.001	-9.628	0.000	-0.014	-0.009
lwrite	-0.0024	0.002	-1.009	0.313	-0.007	0.002
scall	-0.0014	5.26e-05	-26.695	0.000	-0.002	-0.001
sread	0.0006	0.001	0.901	0.367	-0.001	0.002
swrite	-0.0039	0.001	-5.035	0.000	-0.005	-0.002
fork	0.2044	0.094	2.185	0.029	0.021	0.388
exec	-0.3242	0.019	-17.451	0.000	-0.361	-0.288
rchar	-1.197e-06	3.23e-07	-3.709	0.000	-1.83e-06	-5.64e-07
wchar	-5.136e-06	4.91e-07	-10.453	0.000	-6.1e-06	-4.17e-06
pgin	-0.0139	0.011	-1.231	0.218	-0.036	0.008
ppgin	-0.0530	0.007	-7.685	0.000	-0.066	-0.039
pflt	-0.0176	0.002	-10.622	0.000	-0.021	-0.014
vflt	-0.0155	0.001	-12.458	0.000	-0.018	-0.013
freemem	0.0002	2.91e-05	7.319	0.000	0.000	0.000
freeswap	-3.669e-07	1.76e-07	-2.090	0.037	-7.11e-07	-2.27e-08
runqsz_Not_CPU_Bound	-0.1211	0.118	-1.025	0.306	-0.353	0.111

Omnibus:	9057.361	**Durbin-Watson:**	1.997
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	9167845.549
Skew:	-9.883	**Prob(JB):**	0.00
Kurtosis:	197.889	**Cond. No.**	7.39e+06

Intercept Model

```
The intercept for our model is 98.04063083065795
```

Coefficients for each of the independent attribute

```
The coefficient for lread is -0.011900566545768987
The coefficient for lwrite is -0.002382139356811826
The coefficient for scall is -0.001405192374101927
The coefficient for sread is 0.0006411282096700219
The coefficient for swrite is -0.003931752249884772
The coefficient for fork is 0.2043696725744861
The coefficient for exec is -0.32419117788998597
The coefficient for rchar is -1.1965134149888863e-06
The coefficient for wchar is -5.1359696811800335e-06
The coefficient for pgin is -0.013921553275936208
The coefficient for ppgin is -0.05295469120861771
The coefficient for pflt is -0.01764265064165738
The coefficient for vflt is -0.015474555520646915
The coefficient for freemem is 0.00021261163686064945
The coefficient for freeswap is -3.669252004797546e-07
The coefficient for runqsz_Not_CPU_Bound is -0.12111793824531514
```

Summary:

- Linear Regression Equation is (98.0406) * const + (-0.0119) * lread + (-0.0024) * lwrite + (-0.0014) * scall + (0.0006) * sread + (-0.0039) * swrite + (0.2044) * fork + (-0.3242) * exec + (-0.0) * rchar + (-0.0) * wchar + (-0.0139) * pgin + (-0.053) * ppgin + (-0.0176) * pflt + (-0.0155) * vflt + (0.0002) * freemem + (-0.0) * freeswap + (-0.1211) * runqsz_Not_CPU_Bound
- The R-squared value tells us that our model can explain 0.799% of the variance in the training set.
- The Adj. R-squared value tells us that our model can explain 0.799% of the variance in the training set.
- The RMSE training data having value of 4.2237 and RMSE test data having value of 4.202

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.

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.

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.

Solution:

Head Function

	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

Tail Function

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposu
1468	33.0	Tertiary	Tertiary	NaN	Scientology	Yes	2	Very High	Expos
1469	33.0	Tertiary	Tertiary	NaN	Scientology	No	1	Very High	Expos
1470	39.0	Secondary	Secondary	NaN	Scientology	Yes	1	Very High	Expos
1471	33.0	Secondary	Secondary	NaN	Scientology	Yes	2	Low	Expos
1472	17.0	Secondary	Secondary	1.0	Scientology	No	2	Very High	Expos

Shape Function

```
no.of rows:  1473 no.of columns: 10


(1473, 10)
```

Describe Function

	count	mean	std	min	25%	50%	75%	max
Wife_age	1402.0	32.606277	8.274927	16.0	26.0	32.0	39.0	49.0
No_of_children_born	1452.0	3.254132	2.365212	0.0	1.0	3.0	4.0	16.0
Husband_Occupation	1473.0	2.137814	0.864857	1.0	1.0	2.0	3.0	4.0

Info Function

```
<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
```

Check the is Null value function

```
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
```

Duplicate Function

```
Number of duplicate rows = 80
```

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposu
79	38.0	Tertiary	Tertiary	1.0	Scientology	Yes	1	Very High	Expos
167	26.0	Tertiary	Tertiary	1.0	Scientology	No	1	Very High	Expos
224	47.0	Tertiary	Tertiary	4.0	Scientology	No	1	Very High	Expos
270	30.0	Tertiary	Tertiary	2.0	Scientology	No	1	Very High	Expos
299	26.0	Tertiary	Tertiary	1.0	Scientology	No	1	Very High	Expos
...	...	...	...	...	...	...	...	...	
1367	44.0	Tertiary	Tertiary	5.0	Scientology	Yes	1	Very High	Expos
1387	NaN	Secondary	Tertiary	2.0	Scientology	Yes	2	Very High	Expos
1423	NaN	Tertiary	Tertiary	2.0	Non-Scientology	No	1	Very High	Expos
1440	NaN	Tertiary	Tertiary	1.0	Non-Scientology	Yes	2	Very High	Expos
1447	NaN	Tertiary	Tertiary	2.0	Non-Scientology	Yes	2	Very High	Expos

80 rows × 10 columns

Summary: Before treating the Null values and Duplicate values

- Database administrator would be checking the size of the dataset using '.shape' which throw output for number of rows and number of columns and using '.info()' for checking the type of variables in the dataset.
- There are 1473 number of rows and 10 columns in the dataset.
- There are 1 in int64, 2 in float64 and 7 in object present in the dataset
- Duplicate records is present in the dataset with 80
- Null value present in the dataset for wife_age is having 71 rows & No_of_children_born is having 21 rows

Duplicate Function: Drop the duplicate records in the dataset

```
Number of duplicate rows = 0
```

Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure

Shape Function

```
no.of rows:  1393 no.of columns: 10
```

Replace the null value with median

```
# Replace the null value with Median
cmd[['Wife_age', 'No_of_children_born']] = cmd[['Wife_age', 'No_of_children_born']].fillna(cmd[['Wife_age', 'No_of_children_born
```

Info Function:

```
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
```

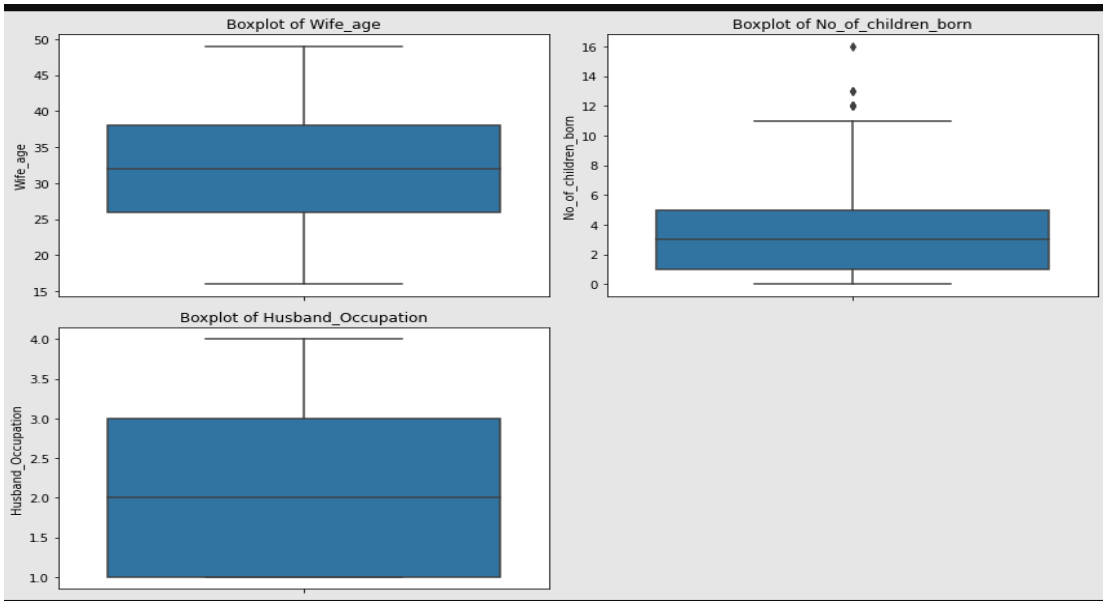
Describe Function

	count	mean	std	min	25%	50%	75%	max
Wife_age	1393.0	32.530510	8.088188	16.0	26.0	32.0	38.0	49.0
No_of_children_born	1393.0	3.286432	2.381791	0.0	1.0	3.0	5.0	16.0
Husband_Occupation	1393.0	2.174444	0.854590	1.0	1.0	2.0	3.0	4.0

Summary: After removing the duplicate values and replacing the null value with median

- Database administrator would be checking the size of the dataset using '.shape' which throw output for number of rows and number of columns and using '.info()' for checking the type of variables in the dataset.
- There are 1393 number of rows and 10 columns in the dataset.
- There are 1 in int64, 2 in float64 and 7 in object present in the dataset
- No Duplicate records is present in the dataset
- No Null value present in the dataset

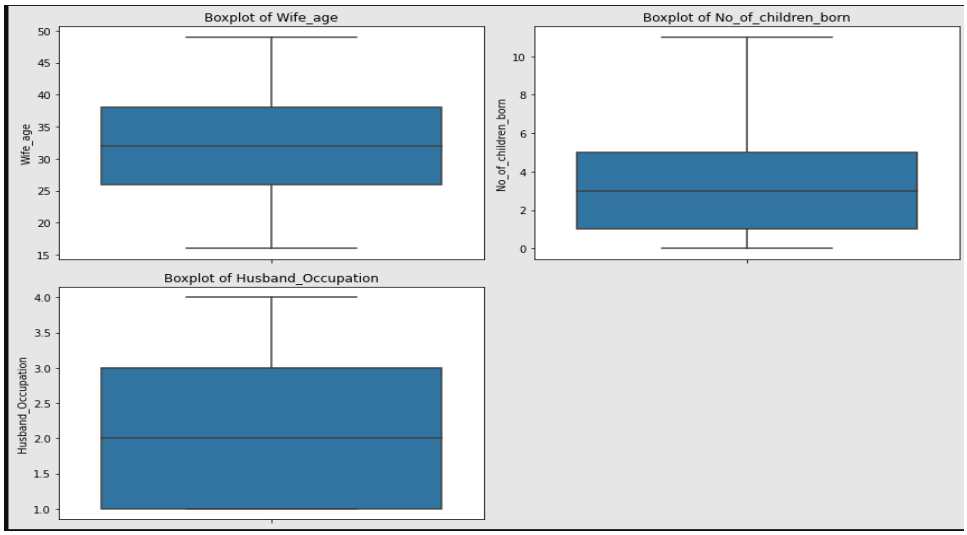
Visualization of box plot with outliers



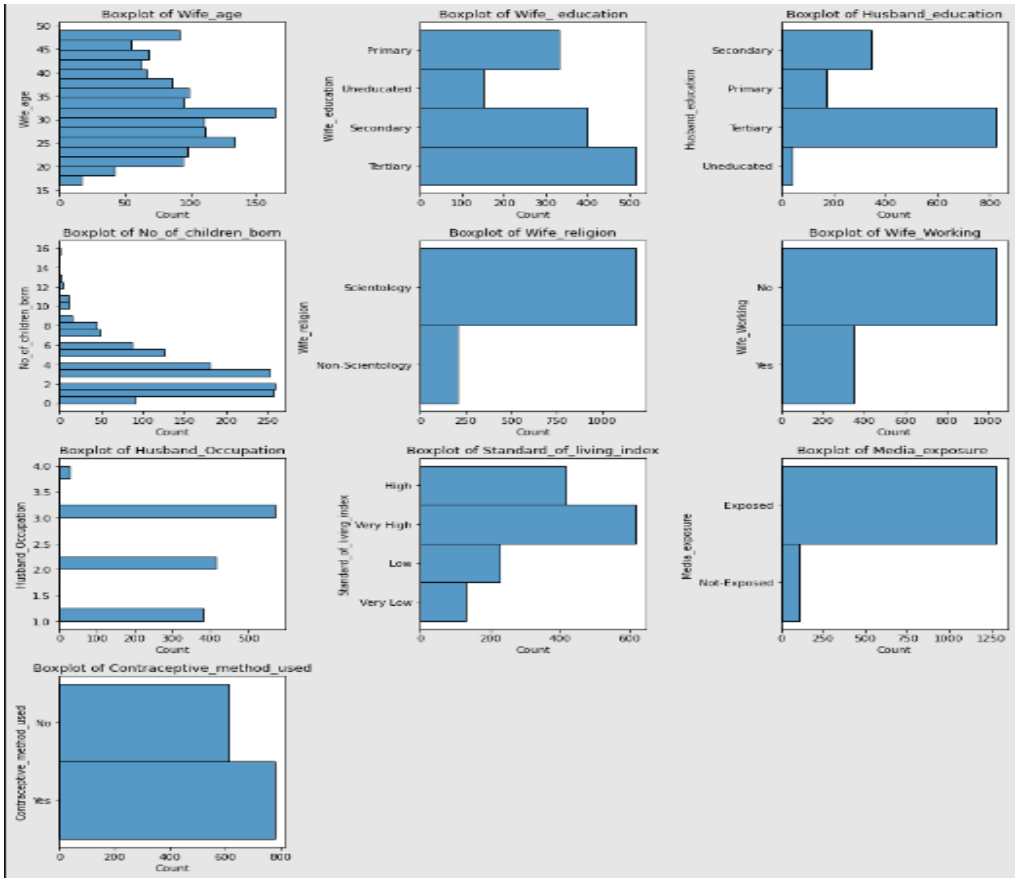
Summary:

- Yes outliers have been detected in all items. These outliers value needs to be treated and there are several ways of treating them:
    - Drop the outlier value
    - Replace the outlier value using the IQR

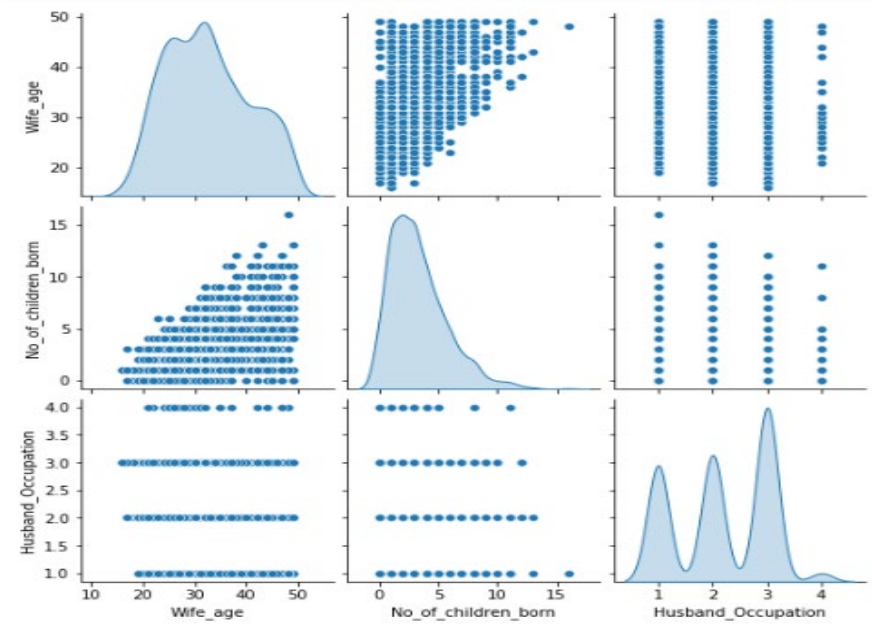
Visualization of box plot after outlier's treatment



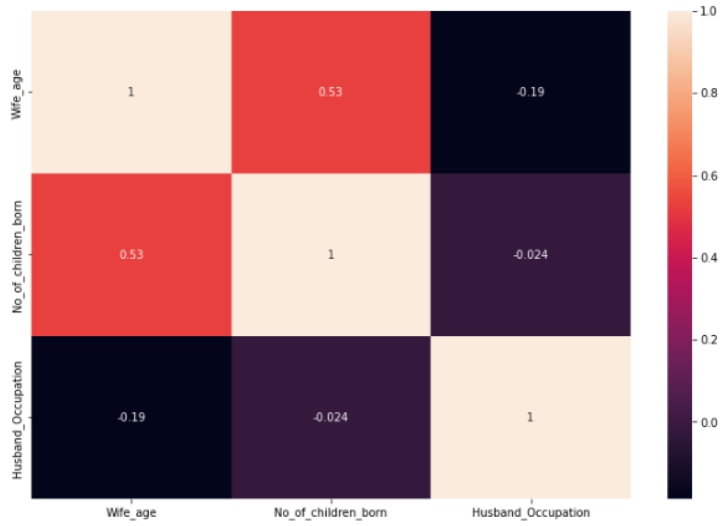
Visualization of the Univariate Analysis



Visualization of the Bivariate Analysis



Visualization of the Multivariate Analysis



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.

Solution:

Logistic Regression

Define the Label Encoder object

```
LabelEncoder()
```

For the created Label Encoder object for the target class

	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	1.0	2.0	3.0	1.0	0.0	2.0	2.0	1.0
1	45.0	0.0	2.0	10.0	1.0	0.0	3.0	3.0	1.0
2	43.0	1.0	2.0	7.0	1.0	0.0	3.0	3.0	1.0
3	42.0	2.0	1.0	9.0	1.0	0.0	3.0	2.0	1.0
4	36.0	2.0	2.0	8.0	1.0	0.0	3.0	1.0	1.0

Fit the Logistic Regression model

```
LogisticRegression(max_iter=400, n_jobs=2, penalty='none', solver='newton-cg',
                   verbose=True)
```

Generate the Predicted Classes and Probability

	0	1
0	0.269587	0.730413
1	0.626732	0.373268
2	0.331280	0.668720
3	0.366515	0.633485
4	0.305626	0.694374

Linear Discriminant Analysis (LDA)

Define the LDA Model

```
LinearDiscriminantAnalysis()
```

Generate Coefficients and intercept for the Linear Discriminant Function

```
array([-0.67133942])
```

Coefficients for the Linear Discriminant Function

```
array([[-7.10680959e-02,  5.60975886e-01,  6.93257323e-02,
         3.12163468e-01, -1.09593225e-15, -1.63112756e-01,
         1.44856377e-01,  3.50587041e-01, -0.00000000e+00]])
```

Linear Discriminant Function = -0.67133942 + (-7.10680959e-02 x Wife_age) + (5.60975886e-01 x Wife_education) + (6.93257323e-02xHusband_education) + (3.12163468e-01 x No_of_children_born) + (-1.09593225e-15 x Wife_religion) + (-1.63112756e-01 x Wife_Working) + (1.44856377e-01 x Husband_Occupation) + (3.50587041e-01 x Standard_of_living_index) + (-0.00000000e+00 xMedia_exposure)

CART

Define the CART Model

```
DecisionTreeClassifier()
```

Regularizing the Decision Tree

```
DecisionTreeClassifier(max_depth=7, min_samples_leaf=10, min_samples_split=30)
```

Importance of features in the tree building

	Imp
Wife_age	0.317299
Wife_ education	0.120375
Husband_education	0.045355
No_of_children_born	0.253368
Wife_religion	0.000000
Wife_Working	0.061082
Husband_Occupation	0.113701
Standard_of_living_index	0.088819
Media_exposure	0.000000

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. Generate Coefficients and intercept for the Linear Discriminant Function

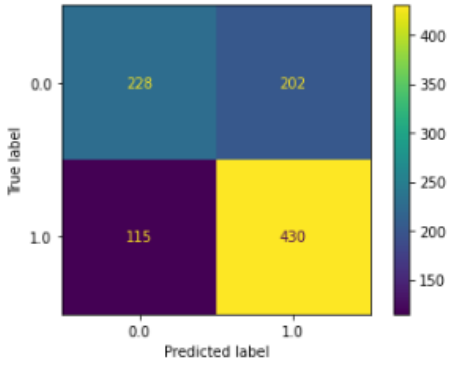
Solution:

Logistic Regression

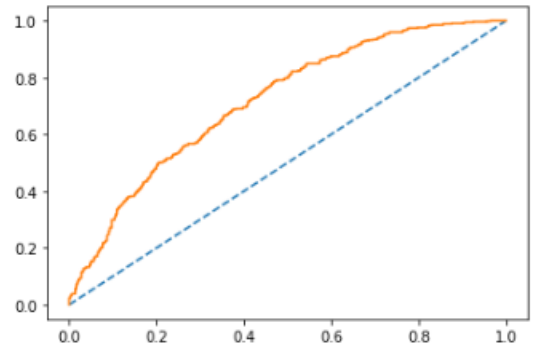
Confusion Matrix for Training Data

	precision	recall	f1-score	support
0.0	0.66	0.53	0.59	430
1.0	0.68	0.79	0.73	545
accuracy			0.67	975
macro avg	0.67	0.66	0.66	975
weighted avg	0.67	0.67	0.67	975

Visualization Confusion Matrix for Training Data Logistic Regression



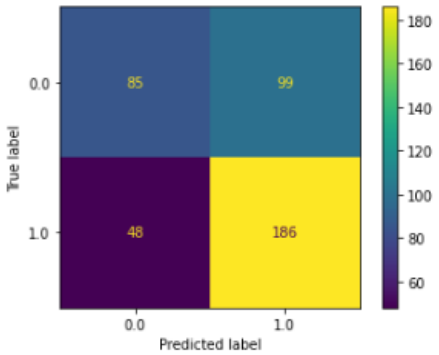
Visualization of AUC and ROC for the training data Logistic Regression



Confusion Matrix for Test Data

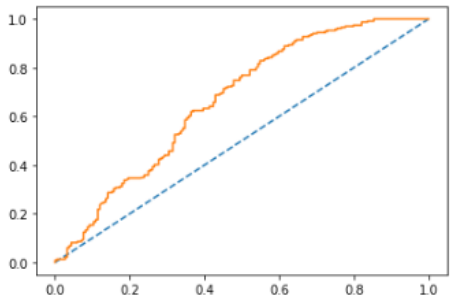
	precision	recall	f1-score	support
0.0	0.64	0.46	0.54	184
1.0	0.65	0.79	0.72	234
accuracy			0.65	418
macro avg	0.65	0.63	0.63	418
weighted avg	0.65	0.65	0.64	418

Visualization Confusion Matrix for Test Data Logistic Regression



Visualization of AUC and ROC for the test data Logistic Regression

AUC: 0.718



Summary:

1) The confusion matrix and classification report for testing data
    a. The precision is having 66% and recall is having 53% for Contraceptive_method_used label is 0
    b. The precision is having 68% and recall is having 79% for Contraceptive_method_used label is 1
    c. The overall accuracy is having 67%

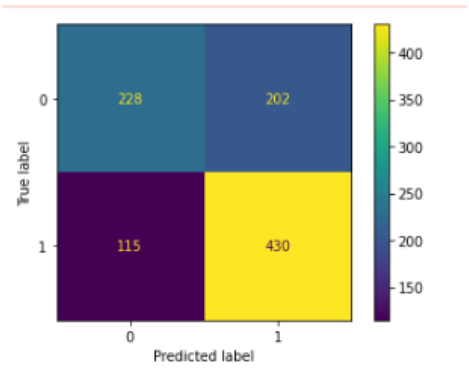
2) The confusion matrix and classification report for test data
    a. The precision is having 64% and recall is having 46% for Contraceptive_method_used label is 0
    b. The precision is having 65% and recall is having 79% for Contraceptive_method_used label is 1
    c. The overall accuracy is having 65%

LDA Model

Confusion Matrix for Training Data LDA Model

	precision	recall	f1-score	support
0	0.66	0.53	0.59	430
1	0.68	0.79	0.73	545
accuracy			0.67	975
macro avg	0.67	0.66	0.66	975
weighted avg	0.67	0.67	0.67	975

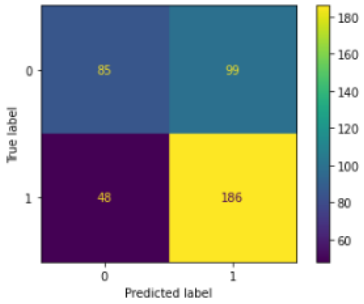
Visualization Confusion Matrix for Training Data LDA Model



Confusion Matrix for Test Data LDA Model

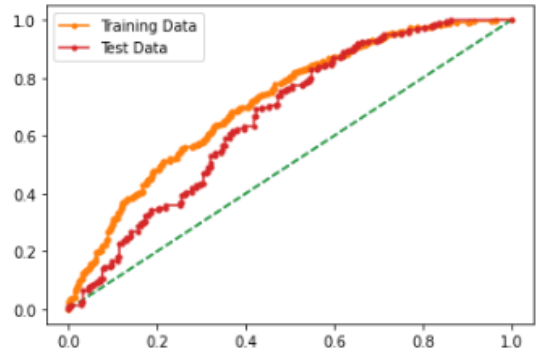
	precision	recall	f1-score	support
0	0.64	0.46	0.54	184
1	0.65	0.79	0.72	234
accuracy			0.65	418
macro avg	0.65	0.63	0.63	418
weighted avg	0.65	0.65	0.64	418

Visualization Confusion Matrix for Test Data LDA Model



Visualization of AUC and ROC for the training and test data LDA Model

```
AUC for the Training Data: 0.717
AUC for the Test Data: 0.665
```



Summary:

1) The confusion matrix and classification report for training data
    a. The precision is having 66% and recall is having 53% for Contraceptive_method_used label is 0
    b. The precision is having 68% and recall is having 79% for Contraceptive_method_used label is 1
    c. The overall accuracy is having 67%
2) The confusion matrix and classification report for test data
    a. The precision is having 64% and recall is having 46% for Contraceptive_method_used label is 0
    b. The precision is having 65% and recall is having 79% for Contraceptive_method_used label is 1
    c. The overall accuracy is having 65%

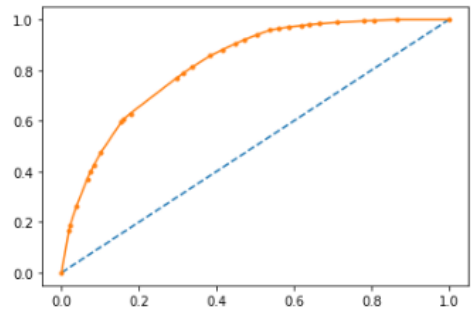
CART

Confusion Matrix for Training Data

	precision	recall	f1-score	support
0.0	0.77	0.62	0.68	422
1.0	0.75	0.86	0.80	553
accuracy			0.75	975
macro avg	0.76	0.74	0.74	975
weighted avg	0.75	0.75	0.75	975

Visualization of AUC and ROC for the training data CART

AUC: 0.824

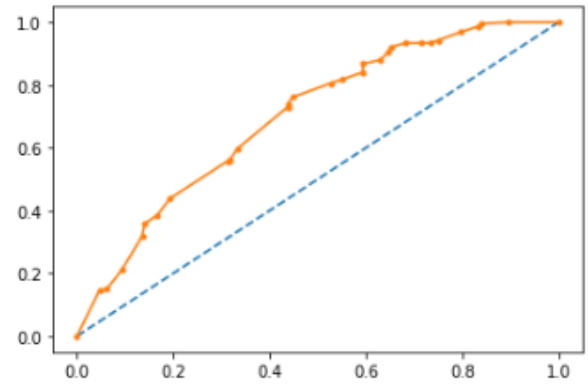


Confusion Matrix for Test Data

	precision	recall	f1-score	support
0.0	0.67	0.47	0.56	192
1.0	0.64	0.81	0.72	226
accuracy			0.65	418
macro avg	0.66	0.64	0.64	418
weighted avg	0.66	0.65	0.64	418

Visualization of AUC and ROC for the test data CART

AUC: 0.700



Summary:

1) The confusion matrix and classification report for training data
    a. The precision is having 77% and recall is having 62% for Contraceptive_method_used label is 0
    b. The precision is having 75% and recall is having 86% for Contraceptive_method_used label is 1
    c. The overall accuracy is having 75%
2) The confusion matrix and classification report for test data
    a. The precision is having 67% and recall is having 47% for Contraceptive_method_used label is 0
    b. The precision is having 64% and recall is having 81% for Contraceptive_method_used label is 1
    c. The overall accuracy is having 65%

Comment:

- Accuracy is having 65% for all three models for test data
- Accuracy is having 75% in CART Model and other 2 model is having 67% for training data

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