In [37]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import *
from sklearn import linear_model
```

In [38]:
```
df=pd.read_csv('D:/shree/Salaries.csv')
```

In [39]:
```
df.head()
# top 5 rows
```

Out[39]:

	rank	discipline	yrs.since.phd	yrs.service	sex	salary
0	Prof	B	19	18	Male	139750
1	Prof	B	20	16	Male	173200
2	AssoProf	B	4	3	Male	79750
3	Prof	B	45	39	Male	115000
4	Prof	B	40	41	Male	141500

In [40]:
```
df.rename(columns={'yrs.since.phd':'yrs_since_phd','yrs.service':'yrs_service'},inplace=True)
# renaming particular column name we want to change
```

In [41]:
```
df.columns
# columns title name
```

Out[41]:
```
Index(['rank', 'discipline', 'yrs_since_phd', 'yrs_service', 'sex', 'salary'], dtype='object')
```

In [42]:
```
df.tail()
# bottom 5 rows
```

Out[42]:

	rank	discipline	yrs_since_phd	yrs_service	sex	salary
392	Prof	A	33	30	Male	103106
393	Prof	A	31	19	Male	150564
394	Prof	A	42	25	Male	101738
395	Prof	A	25	15	Male	95329
396	AssoProf	A	8	4	Male	81035

In [43]:
```
df.describe()
# description of data
```

Out[43]:

	yrs_since_phd	yrs_service	salary
count	397.000000	397.000000	397.000000
mean	22.314861	17.614610	113706.458438
std	12.887003	10.000024	30299.038695
min	1.000000	0.000000	57800.000000
25%	12.000000	7.000000	91000.000000
50%	21.000000	16.000000	107300.000000
75%	32.000000	27.000000	134185.000000
max	56.000000	60.000000	231545.000000

In [44]:
```
df.shape
# rows and columns
```

Out[44]:
```
(397, 6)
```

In [45]:
```
df.isnull()
# checking for null values
```

Out[45]:

	rank	discipline	yrs_since_phd	yrs_service	sex	salary
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	**...**	**...**	**...**	**...**	**...**	**...**
392	False	False	False	False	False	False
393	False	False	False	False	False	False
394	False	False	False	False	False	False
395	False	False	False	False	False	False
396	False	False	False	False	False	False

397 rows × 6 columns

In [46]:
```
df.isnull().sum()
# counting of null values
```

Out[46]:
```
rank            0
discipline      0
yrs_since_phd   0
yrs_service     0
sex             0
salary          0
dtype: int64
```

In [47]:
```
df.dtypes
# data types
```

Out[47]:
```
rank            object
discipline      object
yrs_since_phd    int64
yrs_service      int64
sex             object
salary           int64
dtype: object
```

In [48]:
```
column_values = df[['rank']].values. ravel()
unique_values = pd. unique(column_values)
unique_values

# checking for unique values

array(['Prof', 'AsstProf', 'AssocProf'], dtype=object)
```

In [49]:
```
column_values = df[['discipline']]. values. ravel()
unique_values = pd. unique(column_values)
unique_values

# checking for unique values

array(['B', 'A'], dtype=object)
```

In [50]:
```
column_values = df[['sex']]. values. ravel()
unique_values = pd. unique(column_values)
unique_values

# checking for unique values

array(['Male', 'Female'], dtype=object)
```

In [51]:
```
def tran_rank(x):
    if x == 'Prof':
        return 1
    if x == 'AsstProf':
        return 2
    if x == 'AssocProf':
        return 3
```

In [52]:
```
df['rank']=df['rank'].apply(tran_rank)
df
```

Out[52]:

	rank	discipline	yrs_since_phd	yrs_service	sex	salary
0	1	B	19	18	Male	139750
1	1	B	20	16	Male	173200
2	2	B	4	3	Male	79750
3	1	B	45	39	Male	115000
4	1	B	40	41	Male	141500
...	**...**	**...**	**...**	**...**	**...**	**...**
392	1	A	33	30	Male	103106
393	1	A	31	19	Male	150564
394	1	A	42	25	Male	101738
395	1	A	25	15	Male	95329
396	2	A	8	4	Male	81035

397 rows × 6 columns

In [53]:
```
def tran_discipline(x):
    if x == 'A':
        return 0
    if x == 'B':
        return 1
```

In [54]:
```
df['discipline']=df['discipline'].apply(tran_discipline)
df
```

Out[54]:

	rank	discipline	yrs_since_phd	yrs_service	sex	salary
0	1	1	19	18	Male	139750
1	1	1	20	16	Male	173200
2	2	1	4	3	Male	79750
3	1	1	45	39	Male	115000
4	1	1	40	41	Male	141500
...	**...**	**...**	**...**	**...**	**...**	**...**
392	1	0	33	30	Male	103106
393	1	0	31	19	Male	150564
394	1	0	42	25	Male	101738
395	1	0	25	15	Male	95329
396	2	0	8	4	Male	81035

397 rows × 6 columns

In [55]:
```
def tran_sex(x):
    if x == 'Male':
        return 0
    if x == 'Female':
        return 1
```

In [56]:
```
df['sex']=df['sex'].apply(tran_sex)
df
```

Out[56]:

	rank	discipline	yrs_since_phd	yrs_service	sex	salary
0	1	1	19	18	0	139750
1	1	1	20	16	0	173200
2	2	1	4	3	0	79750
3	1	1	45	39	0	115000
4	1	1	40	41	0	141500
...	**...**	**...**	**...**	**...**	**...**	**...**
392	1	0	33	30	0	103106
393	1	0	31	19	0	150564
394	1	0	42	25	0	101738
395	1	0	25	15	0	95329
396	2	0	8	4	0	81035

397 rows × 6 columns

In [57]:
```
df.head(10)
```

Out[57]:

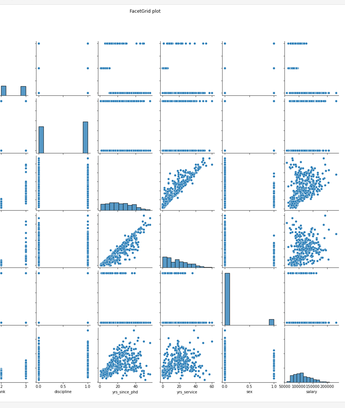
	rank	discipline	yrs_since_phd	yrs_service	sex	salary
0	1	1	19	18	0	139750
1	1	1	20	16	0	173200
2	2	1	4	3	0	79750
3	1	1	45	39	0	115000
4	1	1	40	41	0	141500
5	3	1	6	6	0	97000
6	1	1	30	23	0	175000
7	1	1	45	45	0	147765
8	1	1	21	20	0	119250
9	1	1	18	18	1	129000

In [58]:
```
df.corr()
# correlation
```

Out[58]:

	rank	discipline	yrs_since_phd	yrs_service	sex	salary
rank	1.000000	0.086266	-0.525500	-0.447499	0.132492	-0.522207
discipline	0.086266	1.000000	-0.218687	-0.164589	-0.003734	0.150084
yrs_since_phd	-0.525500	-0.218687	1.000000	0.909649	-0.140788	0.419231
yrs_service	-0.447499	-0.164589	0.909649	1.000000	-0.153740	0.334745
sex	0.132492	-0.003734	-0.140788	-0.153740	1.000000	-0.138610
salary	-0.522207	0.150084	0.419231	0.334745	-0.138610	1.000000

In [59]:
```
plot = sns.pairplot(df)
plot.fig.suptitle("FacetGrid plot", fontsize = 12)
plot.fig.subplots_adjust(top= 0.9)
```



In [ ]:
```
######
```

In [60]:
```
reg=linear_model.LinearRegression()
reg.fit(df[['rank','discipline','yrs_since_phd','yrs_service','sex']],df.salary)
```

Out[60]:
```
LinearRegression()
```

In [61]:
```
reg.coef_
```

Out[61]:
```
array([-15691.03042263,  15508.15775276,   1161.29315651,   -596.666992 ,
       -5230.62332834])
```

In [62]:
```
reg.intercept_
```

Out[62]:
```
113777.56787645792
```

In [ ]: