```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv(r"C:\Users\ADITI DUNGYAN\Documents\Visual Studio 2022\
hrdataset.csv")

df

      Gender  Business Dependancies Calls           Type Billing
Rating  Age  \
0     Female         0           No   Yes  Month-to-month      No
Yes   18
1     Female         0           No   Yes  Month-to-month      No
Yes   19
2       Male         0           No   Yes  Month-to-month     Yes
No   22
3     Female         1           No   Yes  Month-to-month     Yes
Yes   21
4       Male         0           No   Yes  Month-to-month     Yes
Yes   23
...      ...       ...          ...   ...             ...     ...    .
..  ...
4995  Female         0           No   Yes  Month-to-month      No
No   72
4996    Male         0           No   Yes  Month-to-month     Yes
No   73
4997    Male         0           No   Yes  Month-to-month     Yes
No   74
4998    Male         1           No   Yes  Month-to-month     Yes
Yes   74
4999    Male         0          Yes   Yes        Two year     Yes
No   88

         Salary   Base_pay      Bonus  Unit_Price    Volume
openingbalance  \
0       5089.00   2035.600   254.4500    3.770000  21226600
3.75
1       5698.12   2279.248   284.9060    3.740000  10462800
3.85
2       5896.65   2358.660   294.8325    3.890000  18761000
4.23
3       6125.12   2450.048   306.2560    4.350000  66130600
4.26
4       6245.00   2498.000   312.2500    4.340000  26868200
4.79
...         ...        ...        ...         ...       ...
...
4995  180696.80  72278.720  9034.8400  629.511067   3927000
```

```
NaN
4996  185685.90  74274.360  9284.2950  627.841071   6031900
NaN
4997  192636.80  77054.720  9631.8400  625.860033   7949400
NaN
4998  195970.70  78388.280  9798.5350  629.510005   3908400
NaN
4999  199970.74  79988.296  9998.5370  627.839984   6003300
NaN

      closingbalance         low  Unit_Sales Total_Sales  Months  \
0           3.760000    3.650000       18.25        18.8       0
1           3.680000    3.650000       18.40       18.85       0
2           4.290000    3.720000       18.70        18.9       0
3           4.310000    3.830000       18.75          19       0
4           4.410000    4.080000       18.80       19.05       1
...              ...         ...         ...         ...     ...
4995      293.838840  310.955001      117.80                  72
4996      301.311314  309.610028      118.60                  72
4997      306.040009  303.483494      118.60                  72
4998      308.579987  312.432438      118.65                  72
4999      312.307316  311.081089      118.75                  72

                Education
0     High School or less
1     High School or less
2     High School or less
3     High School or less
4     High School or less
...                   ...
4995                   PG
4996                   PG
4997                   PG
4998                   PG
4999                   PG

[5000 rows x 20 columns]
```

```
df.shape
```

```
(5000, 20)
```

```
df.size
```

```
100000
```

```
df.describe()
```

```
          Business          Age         Salary      Base_pay
Bonus  \
count  5000.000000  5000.000000    5000.000000   4977.000000
```

```
5000.000000
mean      0.160000    51.865000   99821.928553  40046.187707
4991.096428
std       0.366643     8.560691   25376.961744  10135.686075
1268.848087
min       0.000000    18.000000    5089.000000   2035.600000
254.450000
25%       0.000000    47.000000   83890.338980  33720.552420
4194.516950
50%       0.000000    52.000000  100579.378500  40282.016040
5028.968925
75%       0.000000    57.000000  116912.092475  46792.232410
5845.604624
max       1.000000    88.000000  199970.740000  79988.296000
9998.537000

        Unit_Price        Volume  openingbalance  closingbalance
low  \
count  5000.000000  5.000000e+03     3524.000000     5000.000000
5000.000000
mean     51.258522  6.761260e+06       43.922020       43.577828
43.034129
std      52.244022  1.620476e+07       38.361497       37.148512
36.760641
min       1.440000  0.000000e+00        3.680000        3.680000
3.650000
25%      25.727500  1.283850e+06       22.098750       21.990000
21.718750
50%      39.205000  2.870600e+06       33.119999       33.340000
32.880001
75%      58.715000  6.247100e+06       51.421839       51.117500
50.415000
max     629.511067  3.208684e+08      313.903904      313.688694
312.432438

       Unit_Sales      Months
count  5000.00000  5000.00000
mean     64.84151    32.18480
std      30.13968    24.63673
min      18.25000     0.00000
25%      35.50000     8.00000
50%      70.50000    28.00000
75%      89.95000    55.00000
max     118.75000    72.00000

df.head(10)

   Gender  Business Dependancies Calls            Type Billing Rating
Age  \
0  Female         0           No   Yes  Month-to-month      No    Yes
```

```
18
1  Female          0            No   Yes  Month-to-month      No    Yes
19
2    Male          0            No   Yes  Month-to-month     Yes     No
22
3  Female          1            No   Yes  Month-to-month     Yes    Yes
21
4    Male          0            No   Yes  Month-to-month     Yes    Yes
23
5    Male          0            No   Yes        Two year     Yes     No
23
6    Male          0           Yes    No        Two year     Yes     No
23
7  Female          0            No   Yes        One year     Yes     No
24
8  Female          1            No   Yes  Month-to-month     Yes    Yes
24
9    Male          0            No   Yes  Month-to-month     Yes     No
43

       Salary     Base_pay       Bonus  Unit_Price    Volume
openingbalance  \
0  5089.00000  2035.600000  254.450000        3.77  21226600
3.7500
1  5698.12000  2279.248000  284.906000        3.74  10462800
3.8500
2  5896.65000  2358.660000  294.832500        3.89  18761000
4.2300
3  6125.12000  2450.048000  306.256000        4.35  66130600
4.2600
4  6245.00000  2498.000000  312.250000        4.34  26868200
4.7900
5  6444.23000  2577.692000  322.211500        4.37  29869600
5.8800
6  6455.50000  2582.200000  322.775000        4.42  25239200
6.0925
7  6458.35722  2583.342888  322.917861        4.44  28307500
6.1000
8  6529.23000  2611.692000  326.461500        4.45  24295600
6.1500
9  6682.33000  2672.932000  334.116500        4.41  17671600
6.2600

   closingbalance   low  Unit_Sales Total_Sales  Months
Education
0            3.760  3.65       18.25        18.8       0  High School
or less
1            3.680  3.65       18.40       18.85       0  High School
or less
```

```
2            4.290  3.72       18.70        18.9       0  High School
or less
3            4.310  3.83       18.75          19       0  High School
or less
4            4.410  4.08       18.80       19.05       1  High School
or less
5            5.040  4.13       18.80        19.1       1  High School
or less
6            5.590  4.15       18.80        19.1       1  High School
or less
7            5.670  4.21       18.80       19.15       1
Intermediate
8            6.170  4.27       18.85        19.2       1
Intermediate
9            6.095  4.22       18.85        19.2       1
Intermediate
```

```python
df.tail(10)
```

```
      Gender  Business Dependancies Calls            Type Billing
Rating  Age  \
4990    Male         0           No   Yes  Month-to-month      No
No   70
4991    Male         1           No   Yes        Two year      No
No   70
4992    Male         1           No   Yes        One year      No
No   71
4993    Male         0           No   Yes  Month-to-month     Yes
Yes   71
4994    Male         0           No   Yes  Month-to-month     Yes
No   71
4995  Female         0           No   Yes  Month-to-month      No
No   72
4996    Male         0           No   Yes  Month-to-month     Yes
No   73
4997    Male         0           No   Yes  Month-to-month     Yes
No   74
4998    Male         1           No   Yes  Month-to-month     Yes
Yes   74
4999    Male         0          Yes   Yes        Two year     Yes
No   88

           Salary     Base_pay        Bonus  Unit_Price   Volume  \
4990  168974.5280  61235.51239  8448.726400  312.500000   317200
4991  169149.7070  67659.88280  8457.485350  309.660004   443500
4992  170372.5473  68149.01893  8518.627365  312.700012   295300
4993  170639.5565  68255.82259  8531.977825  314.000000   294600
4994  175689.3000  70275.72000  8784.465000  625.861078  7987100
4995  180696.8000  72278.72000  9034.840000  629.511067  3927000
4996  185685.9000  74274.36000  9284.295000  627.841071  6031900
```

```
4997  192636.8000  77054.72000  9631.840000  625.860033  7949400
4998  195970.7000  78388.28000  9798.535000  629.510005  3908400
4999  199970.7400  79988.29600  9998.537000  627.839984  6003300

      openingbalance  closingbalance        low  Unit_Sales
Total_Sales  \
4990             NaN      223.960007  307.399994      116.85
8672.45
4991             NaN      219.080002  302.779999      117.15
8684.8
4992             NaN      238.089996  308.489990      117.20

4993             NaN      237.899994  309.420013      117.45

4994             NaN      238.470001  302.048370      117.60

4995             NaN      293.838840  310.955001      117.80

4996             NaN      301.311314  309.610028      118.60

4997             NaN      306.040009  303.483494      118.60

4998             NaN      308.579987  312.432438      118.65

4999             NaN      312.307316  311.081089      118.75


      Months Education
4990      72        PG
4991      72        PG
4992      72        PG
4993      72        PG
4994      72        PG
4995      72        PG
4996      72        PG
4997      72        PG
4998      72        PG
4999      72        PG
```

```
df=df.replace(r'^\s*$', float(np.nan), regex=True)

df.tail()
```

```
      Gender  Business Dependancies Calls            Type Billing
Rating  Age  \
4995  Female         0           No   Yes  Month-to-month      No
No   72
4996    Male         0           No   Yes  Month-to-month     Yes
No   73
4997    Male         0           No   Yes  Month-to-month     Yes
No   74
```

```
4998    Male         1           No   Yes  Month-to-month     Yes
Yes   74
4999    Male         0          Yes   Yes        Two year     Yes
No   88

         Salary   Base_pay     Bonus  Unit_Price   Volume
openingbalance  \
4995  180696.80  72278.720  9034.840  629.511067  3927000
NaN
4996  185685.90  74274.360  9284.295  627.841071  6031900
NaN
4997  192636.80  77054.720  9631.840  625.860033  7949400
NaN
4998  195970.70  78388.280  9798.535  629.510005  3908400
NaN
4999  199970.74  79988.296  9998.537  627.839984  6003300
NaN

      closingbalance         low  Unit_Sales Total_Sales  Months
Education
4995      293.838840  310.955001      117.80         NaN      72
PG
4996      301.311314  309.610028      118.60         NaN      72
PG
4997      306.040009  303.483494      118.60         NaN      72
PG
4998      308.579987  312.432438      118.65         NaN      72
PG
4999      312.307316  311.081089      118.75         NaN      72
PG

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 20 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Gender          5000 non-null   object
 1   Business        5000 non-null   int64
 2   Dependancies    5000 non-null   object
 3   Calls           5000 non-null   object
 4   Type            5000 non-null   object
 5   Billing         5000 non-null   object
 6   Rating          5000 non-null   object
 7   Age             5000 non-null   int64
 8   Salary          5000 non-null   float64
 9   Base_pay        4977 non-null   float64
 10  Bonus           5000 non-null   float64
 11  Unit_Price      5000 non-null   float64
```

```
 12  Volume          5000 non-null   int64
 13  openingbalance  3524 non-null   float64
 14  closingbalance  5000 non-null   float64
 15  low             5000 non-null   float64
 16  Unit_Sales      5000 non-null   float64
 17  Total_Sales     4984 non-null   object
 18  Months          5000 non-null   int64
 19  Education       5000 non-null   object
dtypes: float64(8), int64(4), object(8)
memory usage: 781.4+ KB
```

```
df['Total_Sales']=df['Total_Sales'].astype('float64')
```

```
df.tail()
```

```
      Gender  Business Dependancies Calls            Type Billing
Rating  Age  \
4995  Female         0           No   Yes  Month-to-month      No
No   72
4996    Male         0           No   Yes  Month-to-month     Yes
No   73
4997    Male         0           No   Yes  Month-to-month     Yes
No   74
4998    Male         1           No   Yes  Month-to-month     Yes
Yes   74
4999    Male         0          Yes   Yes        Two year     Yes
No   88

         Salary   Base_pay     Bonus  Unit_Price   Volume
openingbalance  \
4995  180696.80  72278.720  9034.840  629.511067  3927000
NaN
4996  185685.90  74274.360  9284.295  627.841071  6031900
NaN
4997  192636.80  77054.720  9631.840  625.860033  7949400
NaN
4998  195970.70  78388.280  9798.535  629.510005  3908400
NaN
4999  199970.74  79988.296  9998.537  627.839984  6003300
NaN

      closingbalance         low  Unit_Sales  Total_Sales  Months
Education
4995      293.838840  310.955001      117.80          NaN      72
PG
4996      301.311314  309.610028      118.60          NaN      72
PG
4997      306.040009  303.483494      118.60          NaN      72
PG
4998      308.579987  312.432438      118.65          NaN      72
```

```
PG
4999      312.307316  311.081089      118.75          NaN      72
PG
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 20 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Gender          5000 non-null   object 
 1   Business        5000 non-null   int64  
 2   Dependancies    5000 non-null   object 
 3   Calls           5000 non-null   object 
 4   Type            5000 non-null   object 
 5   Billing         5000 non-null   object 
 6   Rating          5000 non-null   object 
 7   Age             5000 non-null   int64  
 8   Salary          5000 non-null   float64
 9   Base_pay        4977 non-null   float64
 10  Bonus           5000 non-null   float64
 11  Unit_Price      5000 non-null   float64
 12  Volume          5000 non-null   int64  
 13  openingbalance  3524 non-null   float64
 14  closingbalance  5000 non-null   float64
 15  low             5000 non-null   float64
 16  Unit_Sales      5000 non-null   float64
 17  Total_Sales     4984 non-null   float64
 18  Months          5000 non-null   int64  
 19  Education       5000 non-null   object 
dtypes: float64(9), int64(4), object(7)
memory usage: 781.4+ KB
```

```
df['Total_Sales']=df['Total_Sales'].astype('float64')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 20 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Gender          5000 non-null   object 
 1   Business        5000 non-null   int64  
 2   Dependancies    5000 non-null   object 
 3   Calls           5000 non-null   object 
 4   Type            5000 non-null   object 
 5   Billing         5000 non-null   object 
 6   Rating          5000 non-null   object 
```

```
 7   Age              5000 non-null   int64
 8   Salary           5000 non-null   float64
 9   Base_pay         4977 non-null   float64
 10  Bonus            5000 non-null   float64
 11  Unit_Price       5000 non-null   float64
 12  Volume           5000 non-null   int64
 13  openingbalance   3524 non-null   float64
 14  closingbalance   5000 non-null   float64
 15  low              5000 non-null   float64
 16  Unit_Sales       5000 non-null   float64
 17  Total_Sales      4984 non-null   float64
 18  Months           5000 non-null   int64
 19  Education        5000 non-null   object
dtypes: float64(9), int64(4), object(7)
memory usage: 781.4+ KB
```

```
missing_values=df.isnull().sum()
missing_values
```

```
Gender               0
Business             0
Dependancies         0
Calls                0
Type                 0
Billing              0
Rating               0
Age                  0
Salary               0
Base_pay            23
Bonus                0
Unit_Price           0
Volume               0
openingbalance    1476
closingbalance       0
low                  0
Unit_Sales           0
Total_Sales         16
Months               0
Education            0
dtype: int64
```

```
df.isnull().sum() *100/len(df)
```

```
Gender             0.00
Business           0.00
Dependancies       0.00
Calls              0.00
Type               0.00
Billing            0.00
Rating             0.00
```

```
Age                  0.00
Salary               0.00
Base_pay             0.46
Bonus                0.00
Unit_Price           0.00
Volume               0.00
openingbalance      29.52
closingbalance       0.00
low                  0.00
Unit_Sales           0.00
Total_Sales          0.32
Months               0.00
Education            0.00
dtype: float64
```

```
from sklearn.impute import KNNImputer
x=df[['Base_pay', 'openingbalance', 'Total_Sales']]
imputer=KNNImputer(n_neighbors=2)
x=imputer.fit_transform(x)

df[['Base_pay', 'openingbalance', 'Total_Sales']]=pd.DataFrame(x,
         columns=['Base_pay', 'openingbalance', 'Total_Sales'])

df.isna().sum()
```

```
Gender              0
Business            0
Dependancies        0
Calls               0
Type                0
Billing             0
Rating              0
Age                 0
Salary              0
Base_pay            0
Bonus               0
Unit_Price          0
Volume              0
openingbalance      0
closingbalance      0
low                 0
Unit_Sales          0
Total_Sales         0
Months              0
Education           0
dtype: int64
```

```
x
```

```
array([[2.03560000e+03, 3.75000000e+00, 1.88000000e+01],
       [2.27924800e+03, 3.85000000e+00, 1.88500000e+01],
```

```
       [2.35866000e+03, 4.23000000e+00, 1.89000000e+01],
       ...,
       [7.70547200e+04, 3.09164993e+02, 8.31165000e+03],
       [7.83882800e+04, 3.09164993e+02, 8.31165000e+03],
       [7.99882960e+04, 3.09164993e+02, 8.31165000e+03]])
```

```
degree_wise=df.Education.value_counts()
degree_wise
```

```
Education
PG                       2979
Graduation               1980
Intermediate               27
High School or less        14
Name: count, dtype: int64
```

```
Gender_wise=df.Gender.value_counts()
Gender_wise
```

```
Gender
Male      2528
Female    2472
Name: count, dtype: int64
```

```
Rating_wise=df.Rating.value_counts()
Rating_wise
```

```
Rating
No     3682
Yes    1318
Name: count, dtype: int64
```

```
call_wise=df.Calls.value_counts()
call_wise
```

```
Calls
Yes    4539
No      461
Name: count, dtype: int64
```

```
Dependancies_wise=df.Dependancies.value_counts()
Dependancies_wise
```

```
Dependancies
No     3524
Yes    1476
Name: count, dtype: int64
```

```
Age_wise=df.Age.value_counts()
Age_wise
```

```
Age
50    256
53    254
55    248
54    245
51    244
     ...
88      2
80      1
82      1
85      1
79      1
Name: count, Length: 65, dtype: int64
```

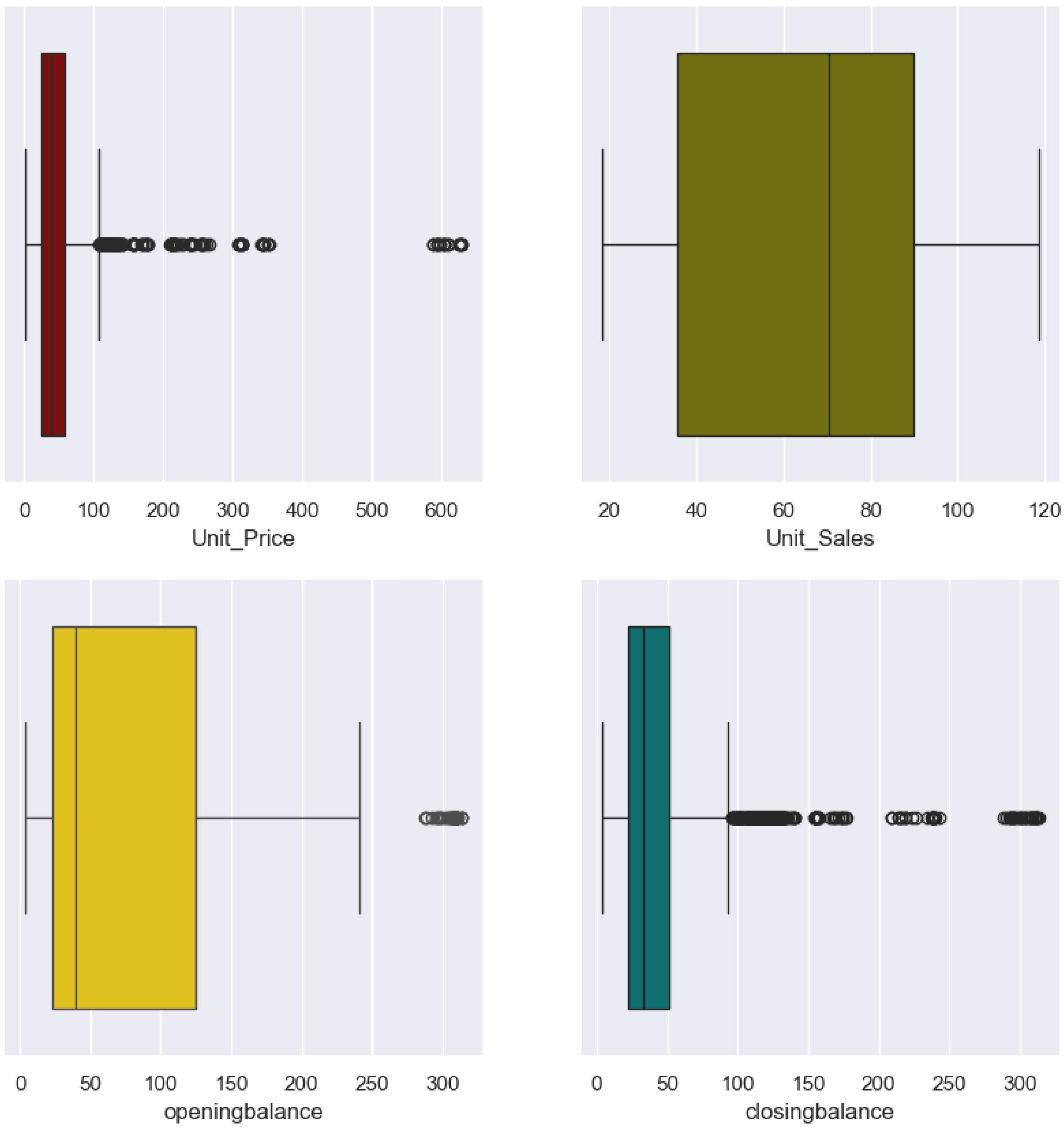
```
Type_wise=df.Type.value_counts()
Type_wise
```

```
Type
Month-to-month    2777
Two year          1195
One year          1028
Name: count, dtype: int64
```

```
# setting a grey background
sns.set(style="darkgrid")
# Creating subplots with 10*10 figure size
fig, axs = plt.subplots(2, 2, figsize=(10, 10))

# Ploting subplots with variables
sns.boxplot(data=df, x="Salary", color="Lime", ax=axs[0, 0])# boxplot to see Distribution of salary
sns.boxplot(data=df, x="Base_pay", color="olive", ax=axs[0, 1])# boxplot to see Distribution of base_pay
sns.boxplot(data=df, x="Bonus", color="gold", ax=axs[1, 0])# boxplot to see Distribution of bonus
sns.boxplot(data=df, x="Age", color="pink", ax=axs[1, 1])# boxplot to see Distribution of age
plt.show()
```

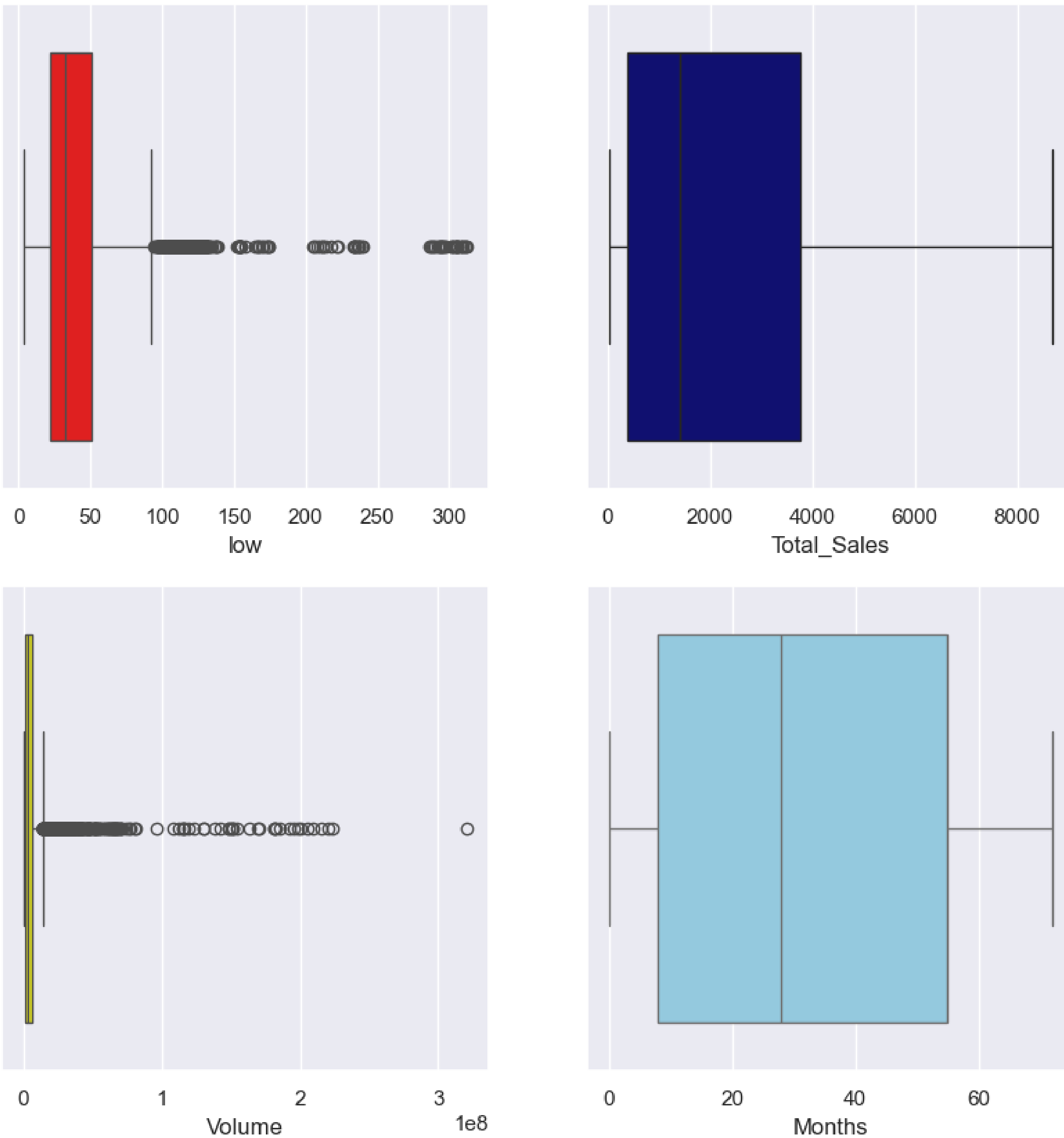

```
# setting a grey background
sns.set(style="darkgrid")
# Creating subplots with 10*10 figure size
fig, axs = plt.subplots(2, 2, figsize=(10, 10))

# Ploting subplots with all variables
sns.boxplot(data=df, x="Unit_Price", color="darkred", ax=axs[0, 0])#
boxplot to see outliers of unitprice
sns.boxplot(data=df, x="Unit_Sales", color="olive", ax=axs[0, 1])#
boxplot to see outliers of unit_sales
```

```
sns.boxplot(data=df, x="openingbalance", color="gold", ax=axs[1, 0])#
boxplot to see outliers of openingbalance
sns.boxplot(data=df, x="closingbalance", color="teal", ax=axs[1, 1])#
boxplot to see outliers of closingbalance

plt.show()
```




```
sns.set(style="darkgrid")
# Creating subplots with 10*10 figure size
fig, axs = plt.subplots(2, 2, figsize=(10, 10))
```

```python
# Ploting subplots with variables
sns.boxplot(data=df, x="low", color="red", ax=axs[0,0])# boxplot to
see outliers of low column
sns.boxplot(data=df, x="Total_Sales", color="Navy", ax=axs[0,1])#
boxplot to see outliers of Total_sales column
sns.boxplot(data=df, x="Volume", color="Yellow", ax=axs[1,0])# boxplot
to see outliers of volume column
sns.boxplot(data=df, x="Months", color="skyblue", ax=axs[1,1])#
boxplot to see outliers of months column
plt.show()
```

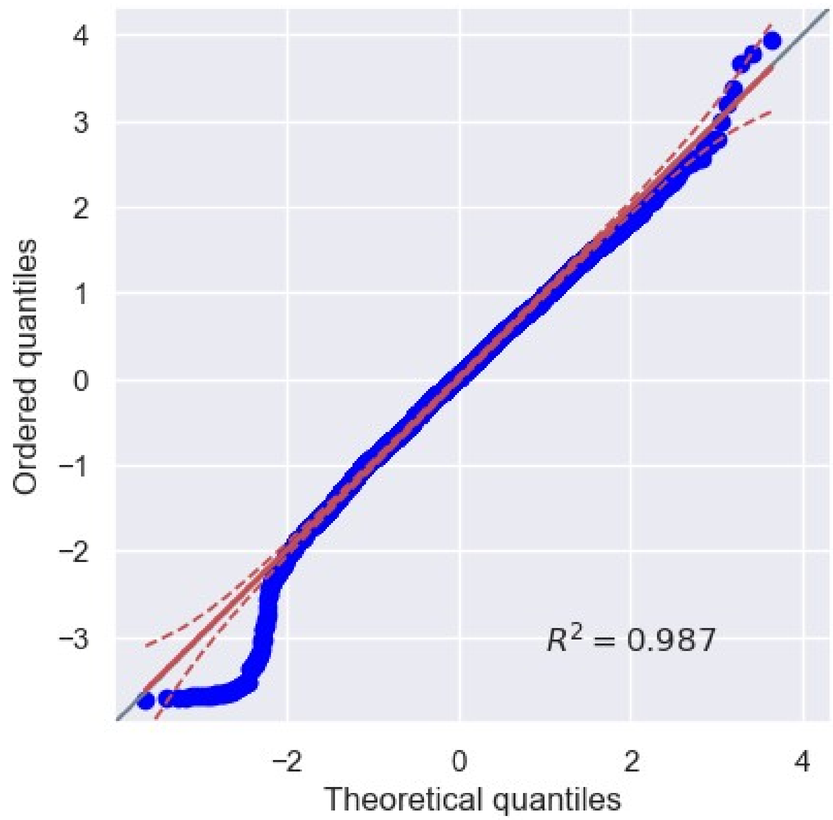



```
pip install pingouin

Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: pingouin in c:\users\aditi dungyan\
appdata\roaming\python\python313\site-packages (0.5.5)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\
lib\site-packages (from pingouin) (3.10.0)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\
site-packages (from pingouin) (2.1.3)
```

```
Requirement already satisfied: pandas>=1.5 in c:\programdata\
anaconda3\lib\site-packages (from pingouin) (2.2.3)
Requirement already satisfied: pandas-flavor in c:\users\aditi
dungyan\appdata\roaming\python\python313\site-packages (from pingouin)
(0.7.0)
Requirement already satisfied: scikit-learn>=1.2 in c:\programdata\
anaconda3\lib\site-packages (from pingouin) (1.6.1)
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\
site-packages (from pingouin) (1.15.3)
Requirement already satisfied: seaborn in c:\programdata\anaconda3\
lib\site-packages (from pingouin) (0.13.2)
Requirement already satisfied: statsmodels in c:\programdata\
anaconda3\lib\site-packages (from pingouin) (0.14.4)
Requirement already satisfied: tabulate in c:\programdata\anaconda3\
lib\site-packages (from pingouin) (0.9.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\
programdata\anaconda3\lib\site-packages (from pandas>=1.5->pingouin)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\
anaconda3\lib\site-packages (from pandas>=1.5->pingouin) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\programdata\
anaconda3\lib\site-packages (from pandas>=1.5->pingouin) (2025.2)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\
lib\site-packages (from python-dateutil>=2.8.2->pandas>=1.5->pingouin)
(1.17.0)
Requirement already satisfied: joblib>=1.2.0 in c:\programdata\
anaconda3\lib\site-packages (from scikit-learn>=1.2->pingouin) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\programdata\
anaconda3\lib\site-packages (from scikit-learn>=1.2->pingouin) (3.5.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\programdata\
anaconda3\lib\site-packages (from matplotlib->pingouin) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\programdata\
anaconda3\lib\site-packages (from matplotlib->pingouin) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\
anaconda3\lib\site-packages (from matplotlib->pingouin) (4.55.3)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\programdata\
anaconda3\lib\site-packages (from matplotlib->pingouin) (1.4.8)
Requirement already satisfied: packaging>=20.0 in c:\programdata\
anaconda3\lib\site-packages (from matplotlib->pingouin) (24.2)
Requirement already satisfied: pillow>=8 in c:\programdata\anaconda3\
lib\site-packages (from matplotlib->pingouin) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\programdata\
anaconda3\lib\site-packages (from matplotlib->pingouin) (3.2.0)
Requirement already satisfied: xarray in c:\programdata\anaconda3\lib\
site-packages (from pandas-flavor->pingouin) (2025.4.0)
Requirement already satisfied: patsy>=0.5.6 in c:\programdata\
anaconda3\lib\site-packages (from statsmodels->pingouin) (1.0.1)
Note: you may need to restart the kernel to use updated packages.
```

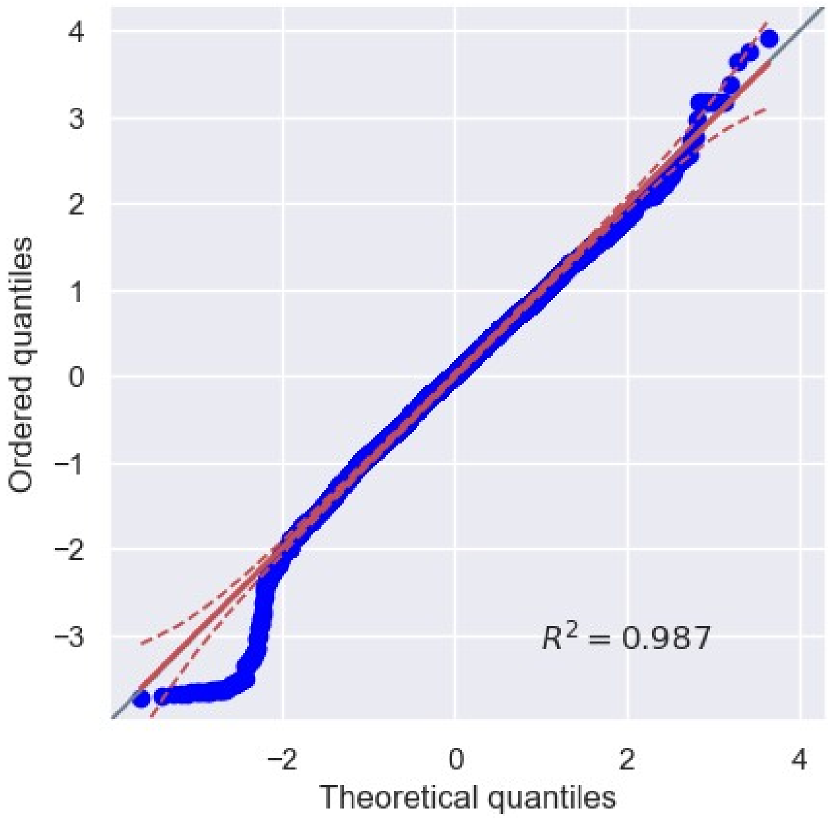
```
import pingouin as pg
pg.qqplot(df['Salary'],dist='norm')
```

```
<Axes: xlabel='Theoretical quantiles', ylabel='Ordered quantiles'>
```



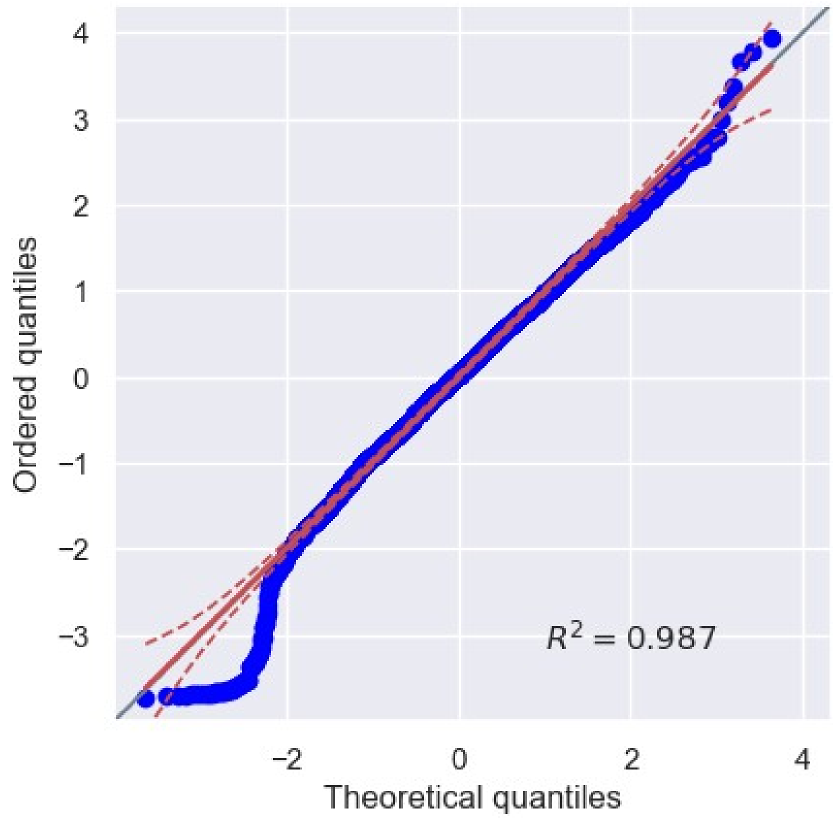

```
pg.qqplot(df['Base_pay'], dist='norm')
```

```
<Axes: xlabel='Theoretical quantiles', ylabel='Ordered quantiles'>
```




```
pg.qqplot(df['Bonus'],dist='norm')
```

```
<Axes: xlabel='Theoretical quantiles', ylabel='Ordered quantiles'>
```

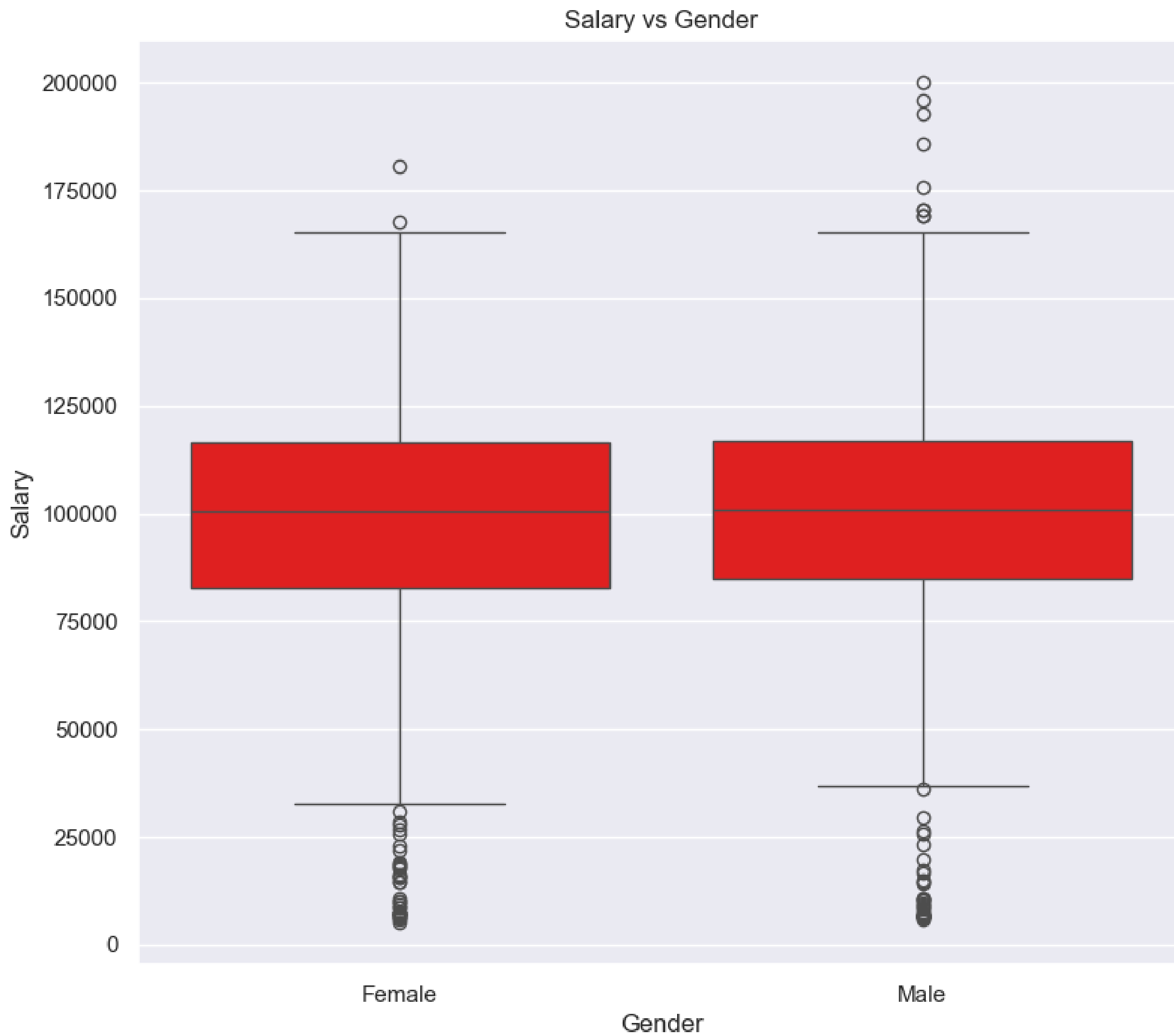



```
df.select_dtypes(include=['object']).columns.tolist()
```

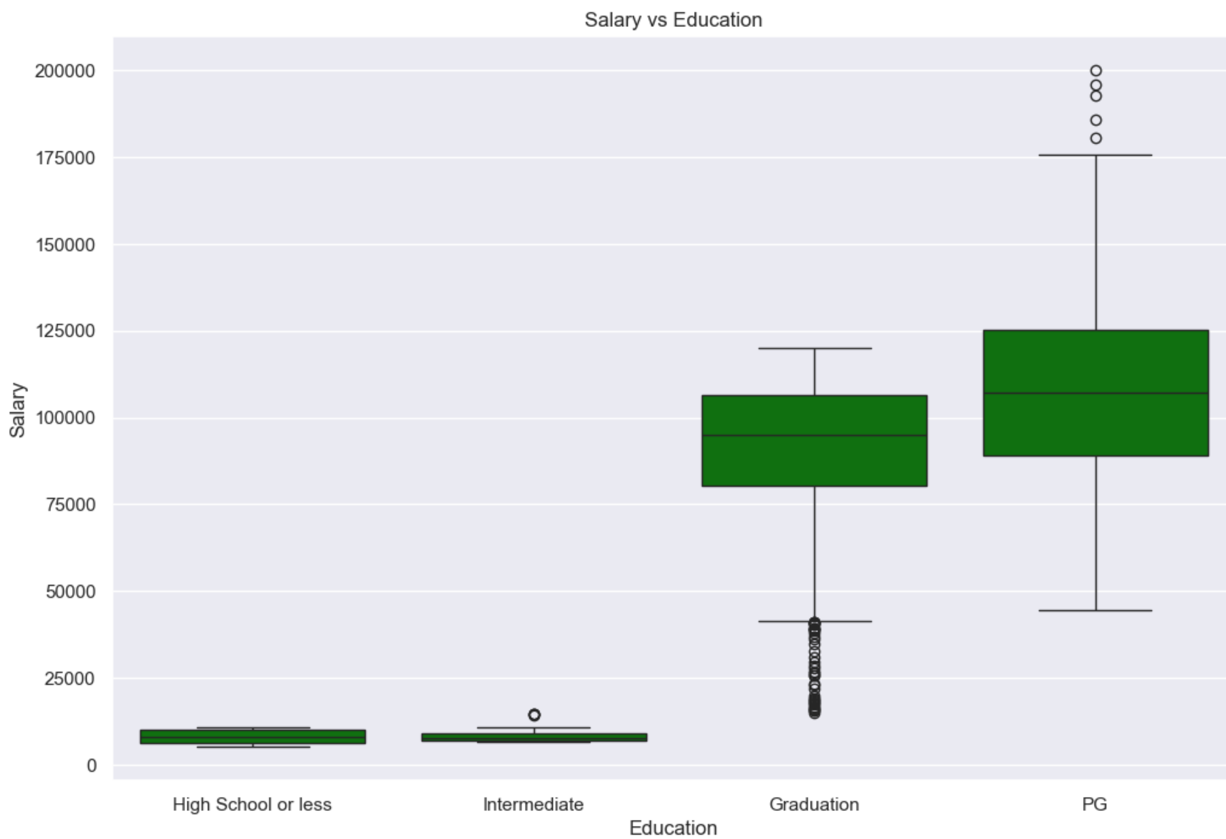
```
['Gender', 'Dependancies', 'Calls', 'Type', 'Billing', 'Rating',
'Education']
```

```
plt.figure(figsize=(9,8))
sns.boxplot(data=df,x="Gender", y="Salary",color="Red")
plt.title("Salary vs Gender")
```

```
Text(0.5, 1.0, 'Salary vs Gender')
```

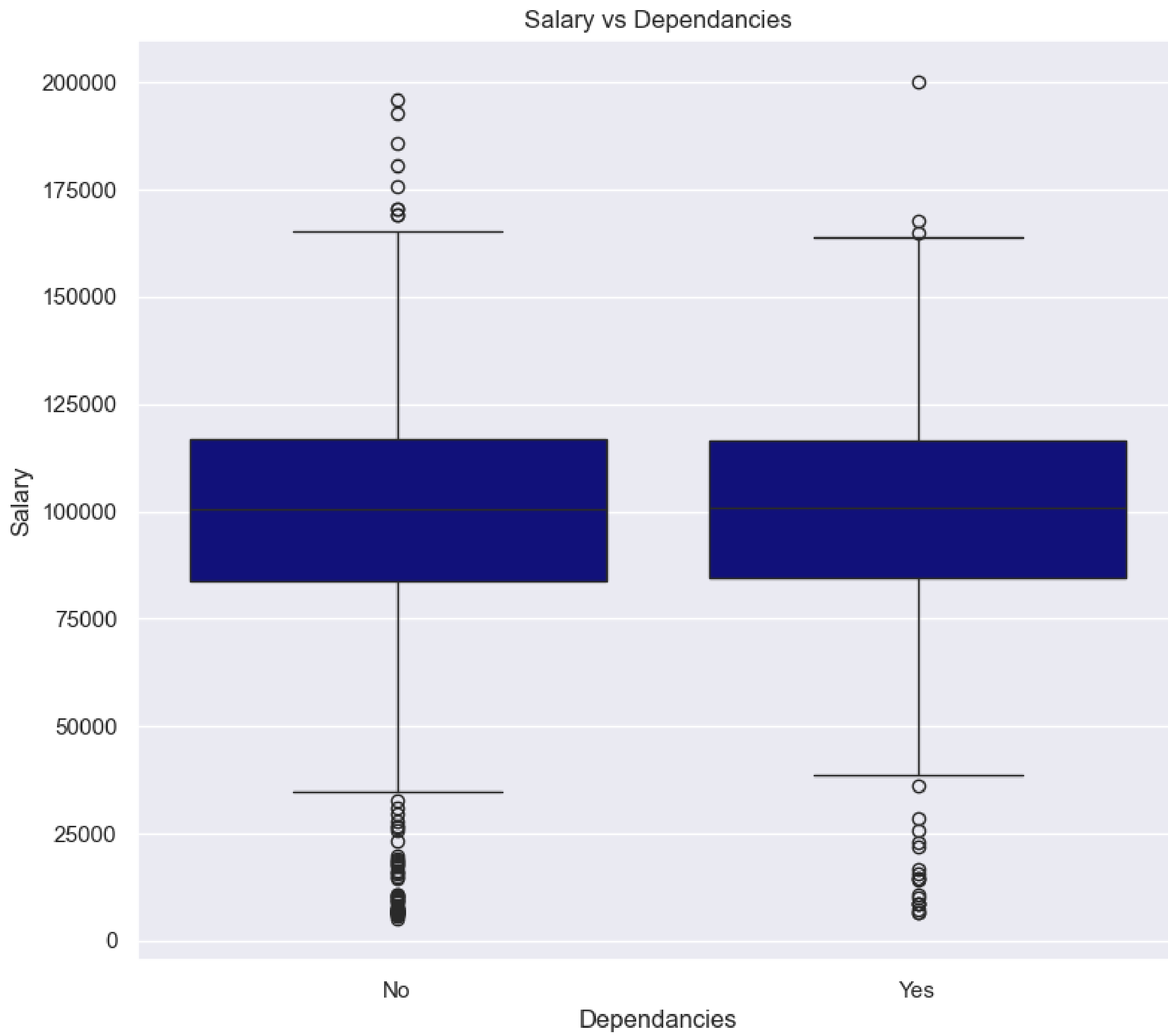



```
plt.figure(figsize=(12,8))
sns.boxplot(data=df,x="Education", y="Salary",color="green")
plt.title("Salary vs Education")

Text(0.5, 1.0, 'Salary vs Education')
```



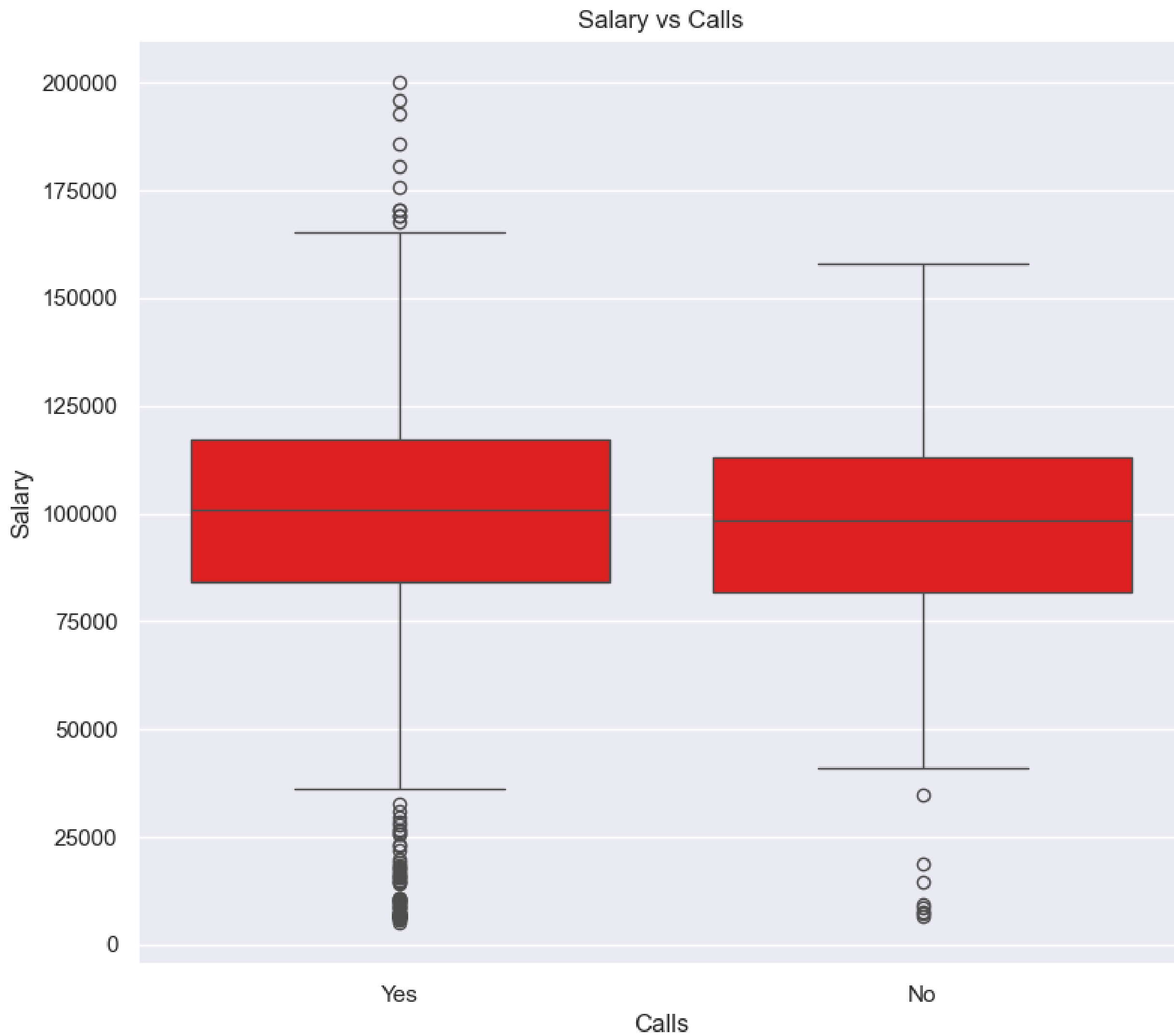

```
plt.figure(figsize=(9,8))
sns.boxplot(data=df,x="Dependancies", y="Salary",color="darkblue")
plt.title("Salary vs Dependancies")
```

```
Text(0.5, 1.0, 'Salary vs Dependancies')
```



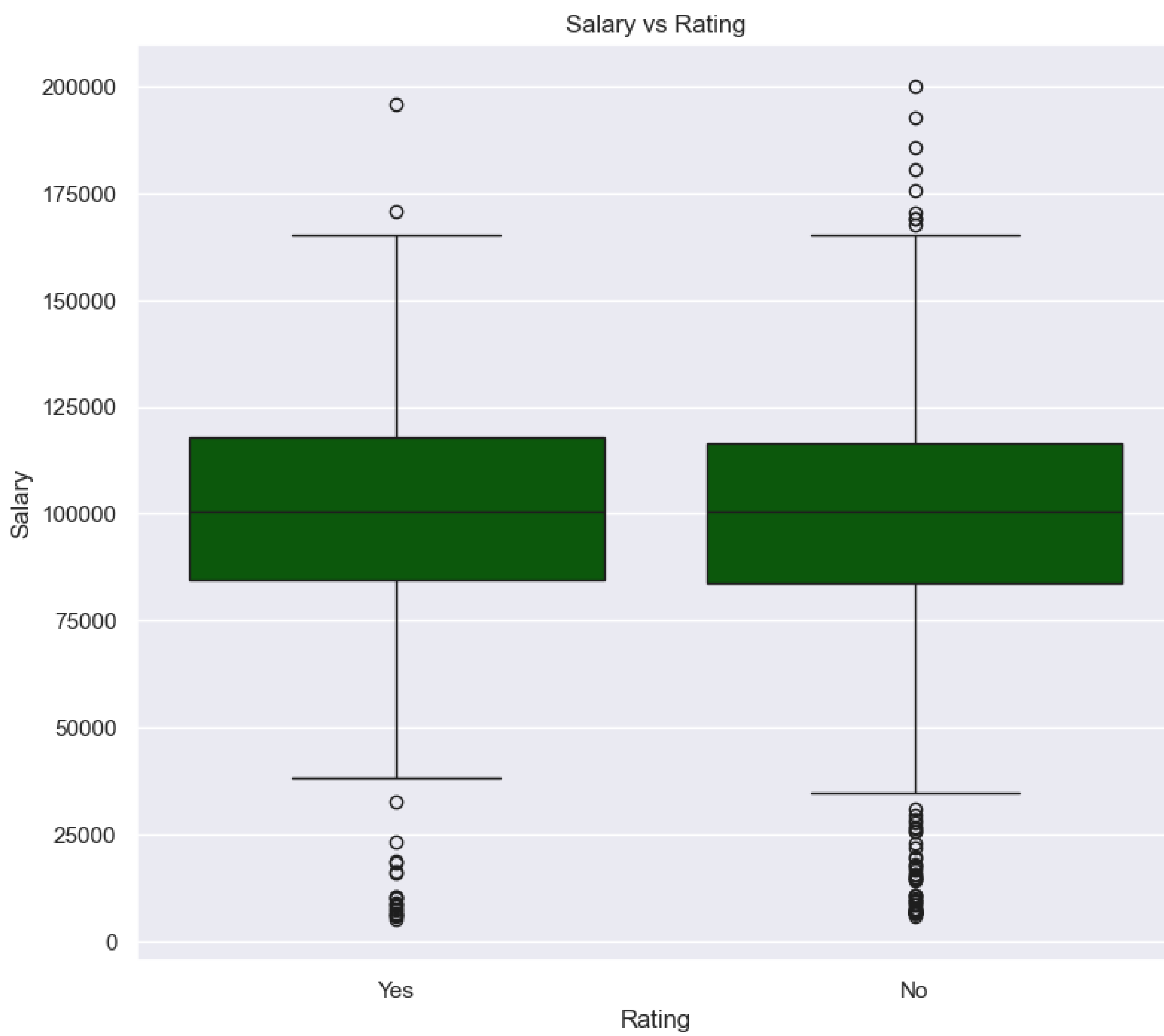

```
plt.figure(figsize=(9,8))
sns.boxplot(data=df,x="Calls", y="Salary",color="Red")
plt.title("Salary vs Calls")
```

```
Text(0.5, 1.0, 'Salary vs Calls')
```



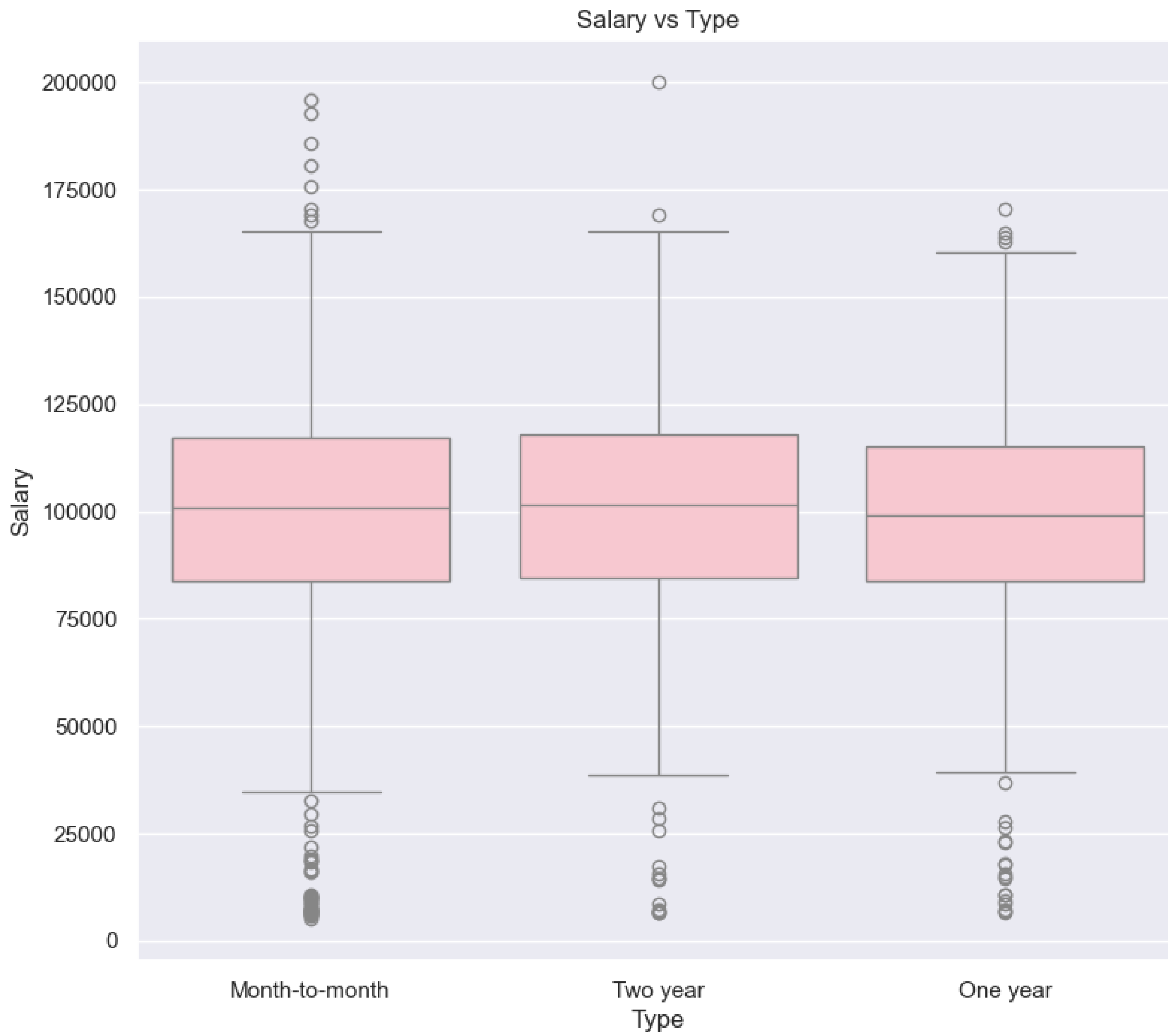

```
plt.figure(figsize=(9,8))
sns.boxplot(data=df,x="Rating", y="Salary",color="darkgreen")
plt.title("Salary vs Rating")
```

```
Text(0.5, 1.0, 'Salary vs Rating')
```

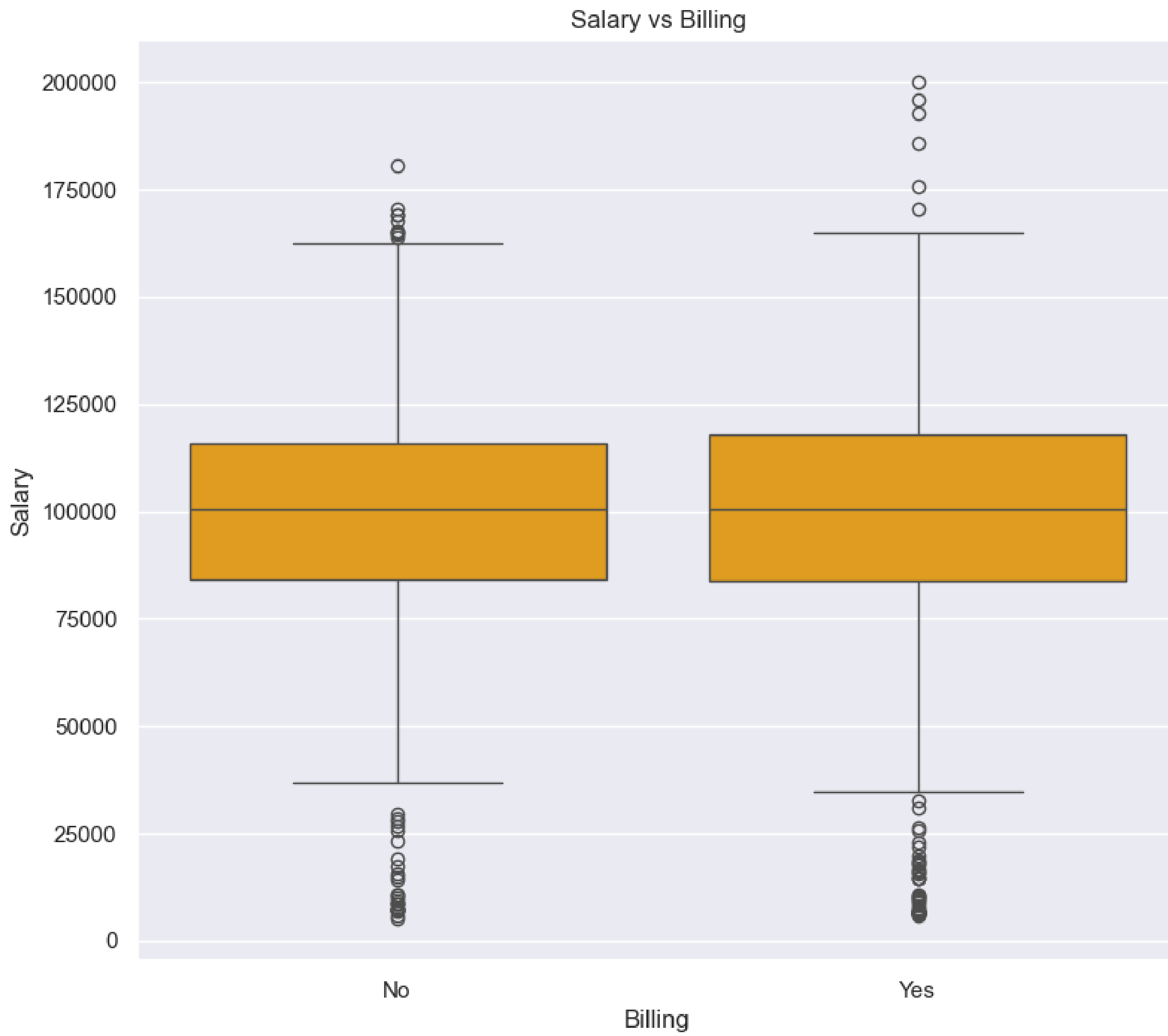



```
plt.figure(figsize=(9,8))
sns.boxplot(data=df,x="Type", y="Salary",color="pink")
plt.title("Salary vs Type")
```

```
Text(0.5, 1.0, 'Salary vs Type')
```



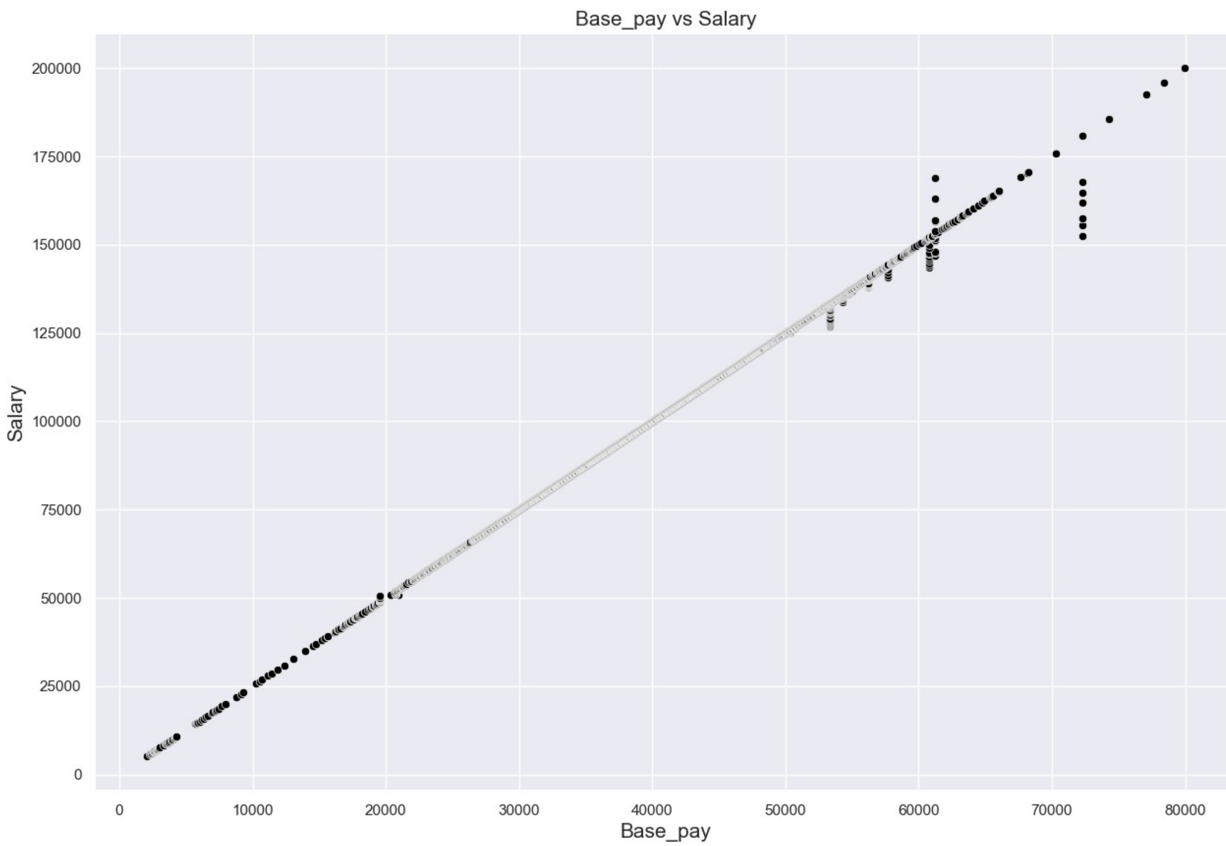

```
plt.figure(figsize=(9,8))
sns.boxplot(data=df,x="Billing", y="Salary",color="orange")
plt.title("Salary vs Billing")

Text(0.5, 1.0, 'Salary vs Billing')
```



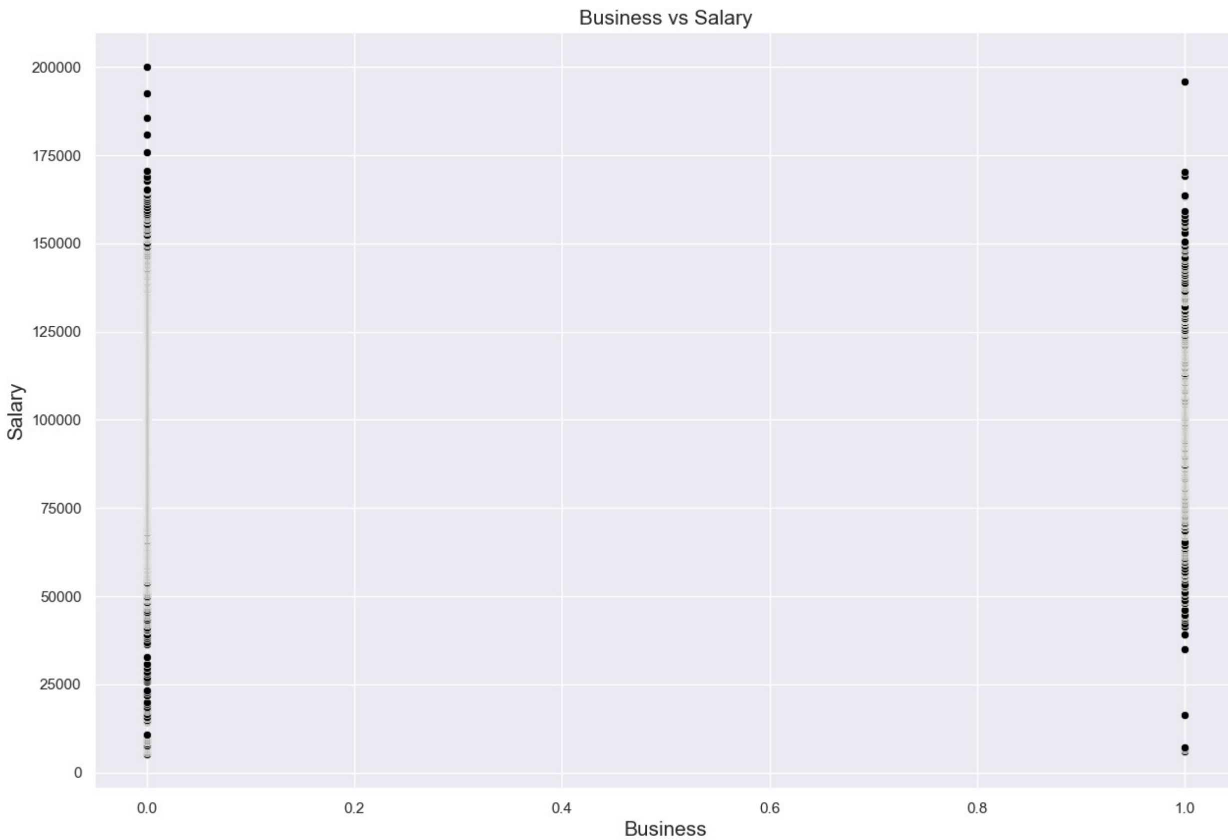

```
df.select_dtypes(include=['number']).columns.tolist()
```

```
['Business',
 'Age',
 'Salary',
 'Base_pay',
 'Bonus',
 'Unit_Price',
 'Volume',
 'openingbalance',
 'closingbalance',
 'low',
 'Unit_Sales',
 'Total_Sales',
 'Months']
```

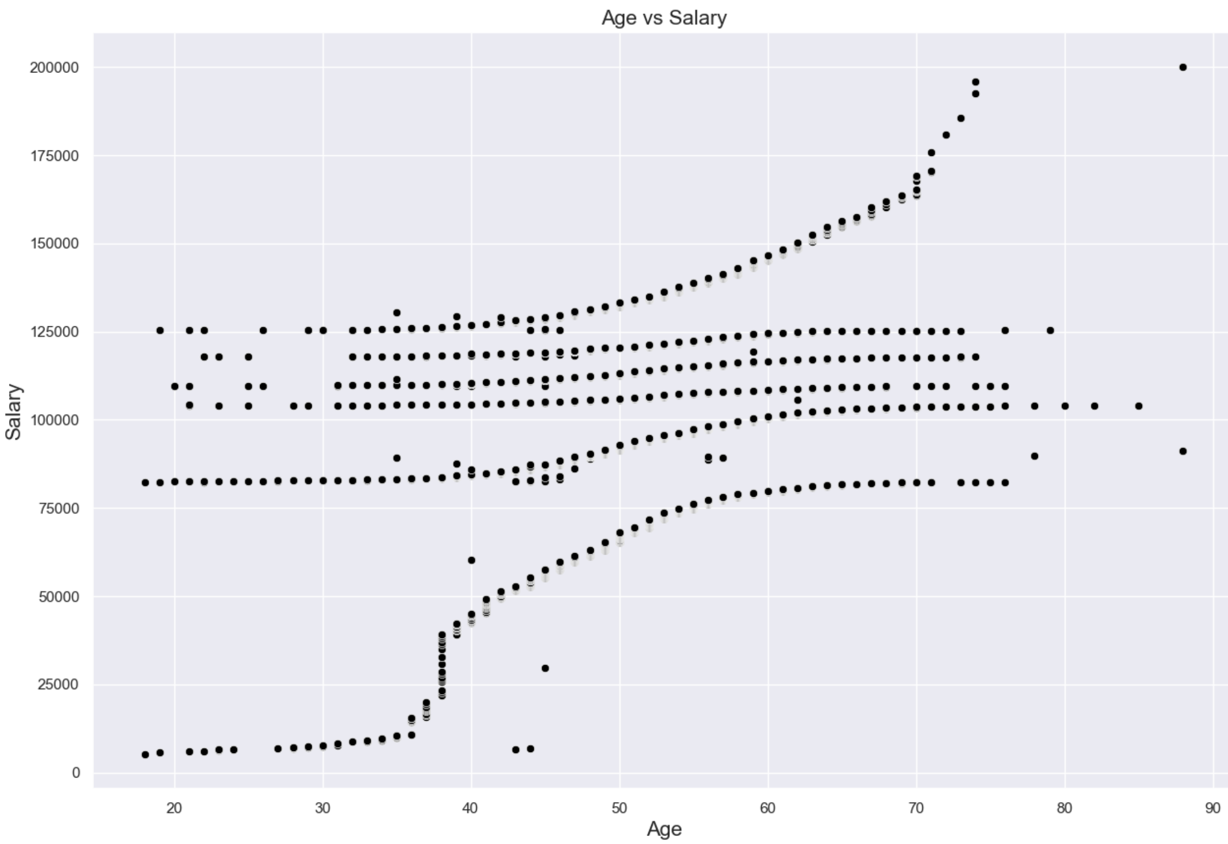
```
plt.figure(figsize=(15,10))
plt.xlabel('Base_pay',fontsize=15)
plt.ylabel('Salary',fontsize=15)
plt.title('Base_pay vs Salary',fontsize=15)

sns.scatterplot(x=df['Base_pay'],y=df['Salary'],color='black')
plt.show()
```



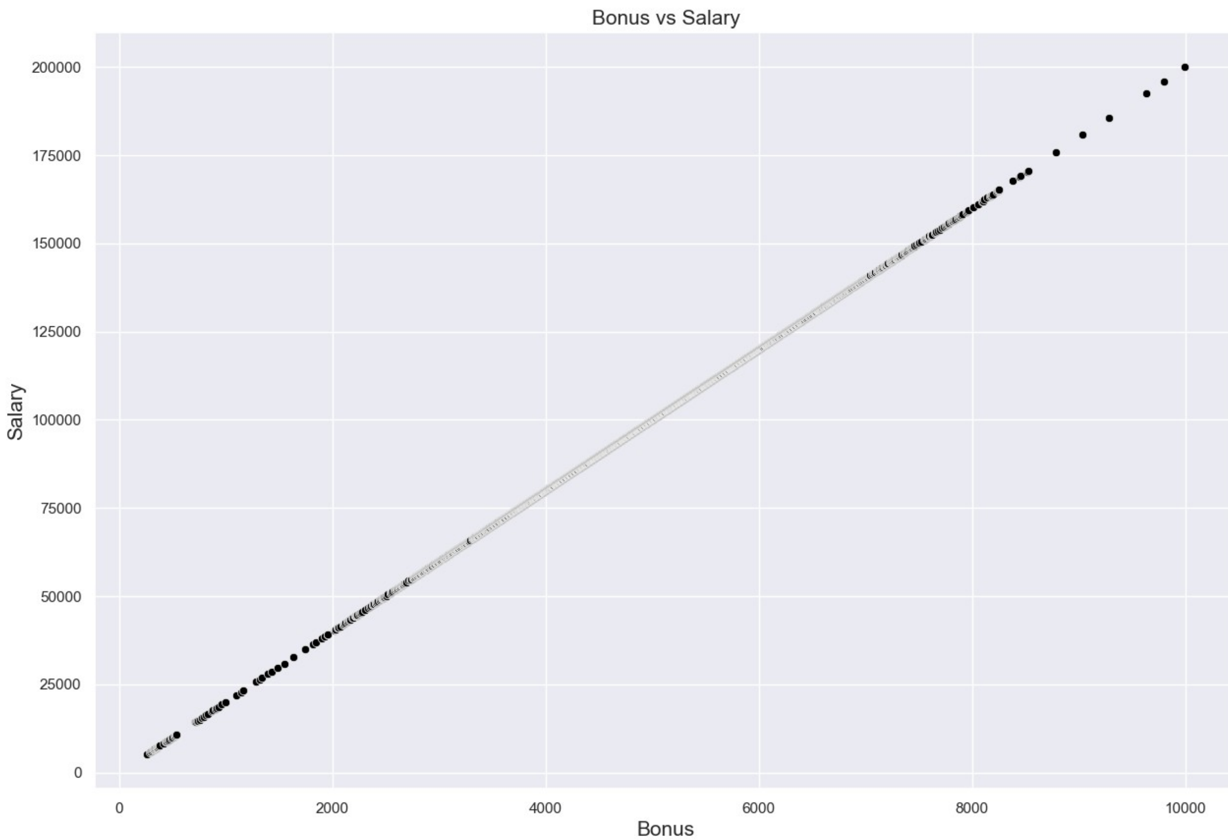

```
plt.figure(figsize=(15,10))
plt.xlabel('Business',fontsize=15)
plt.ylabel('Salary',fontsize=15)
plt.title('Business vs Salary',fontsize=15)

sns.scatterplot(x=df['Business'],y=df['Salary'],color='black')
plt.show()
```



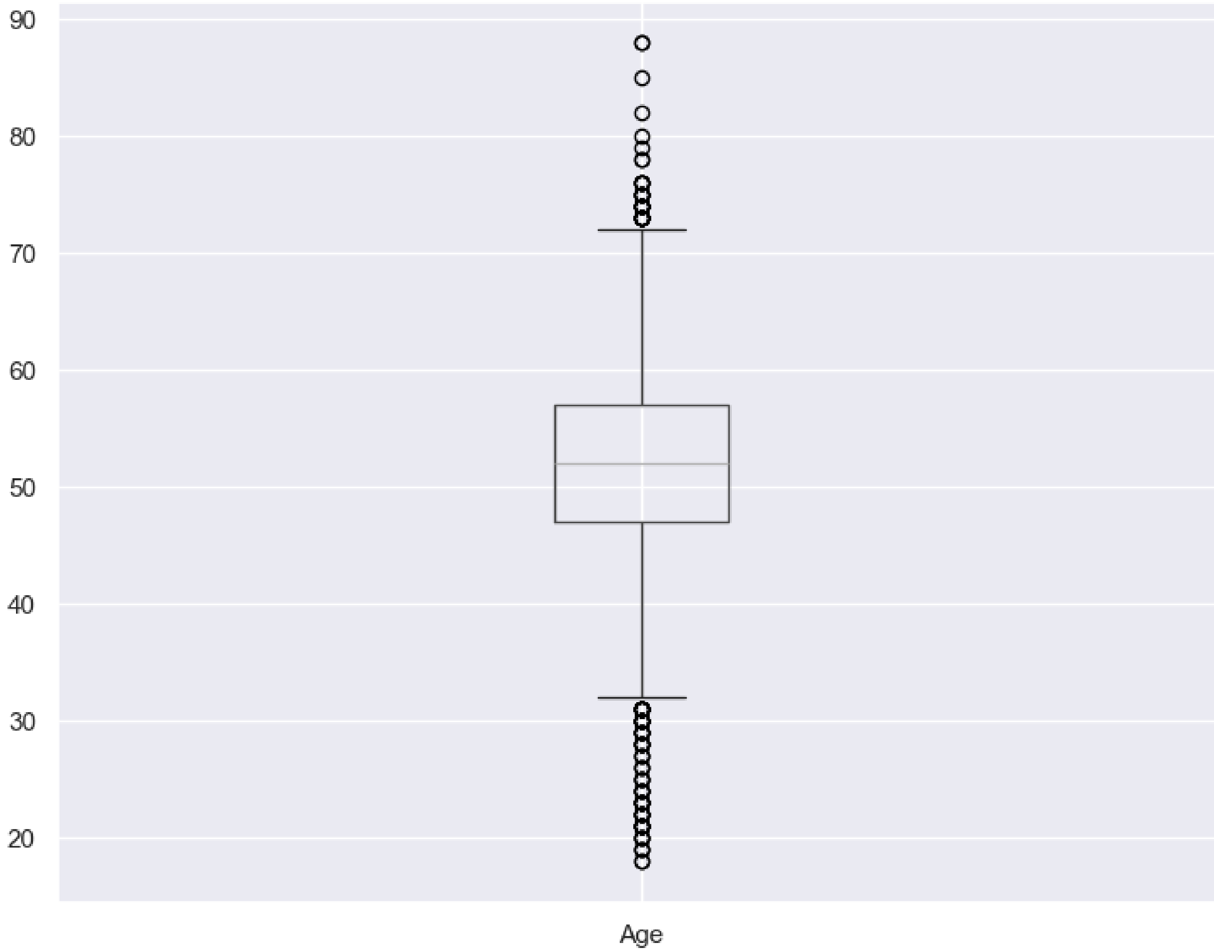

```
plt.figure(figsize=(15,10))
plt.xlabel('Age',fontsize=15)
plt.ylabel('Salary',fontsize=15)
plt.title('Age vs Salary',fontsize=15)

sns.scatterplot(x=df['Age'],y=df['Salary'],color='black')
plt.show()
```



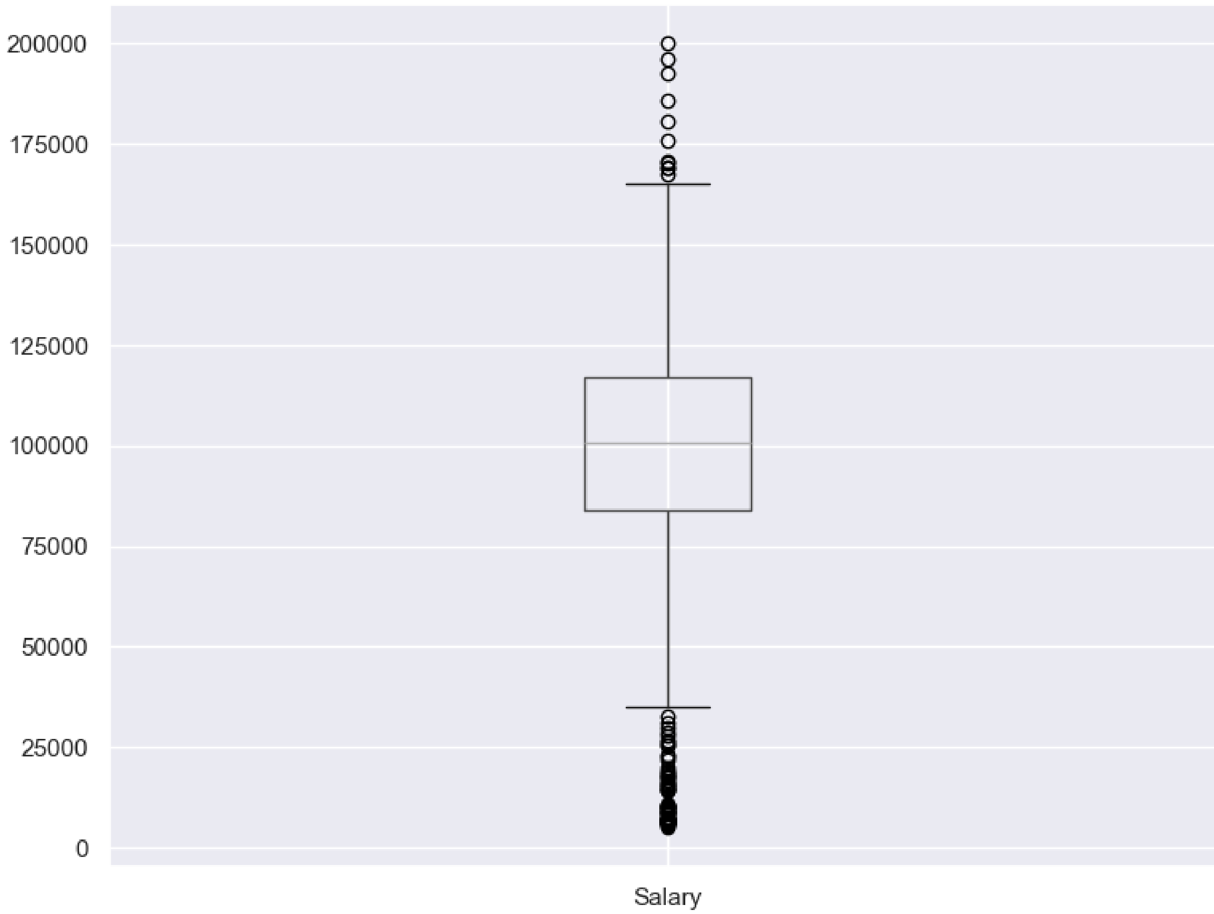

```
plt.figure(figsize=(15,10))
plt.xlabel('Bonus',fontsize=15)
plt.ylabel('Salary',fontsize=15)
plt.title('Bonus vs Salary',fontsize=15)

sns.scatterplot(x=df['Bonus'],y=df['Salary'],color='black')
plt.show()
```



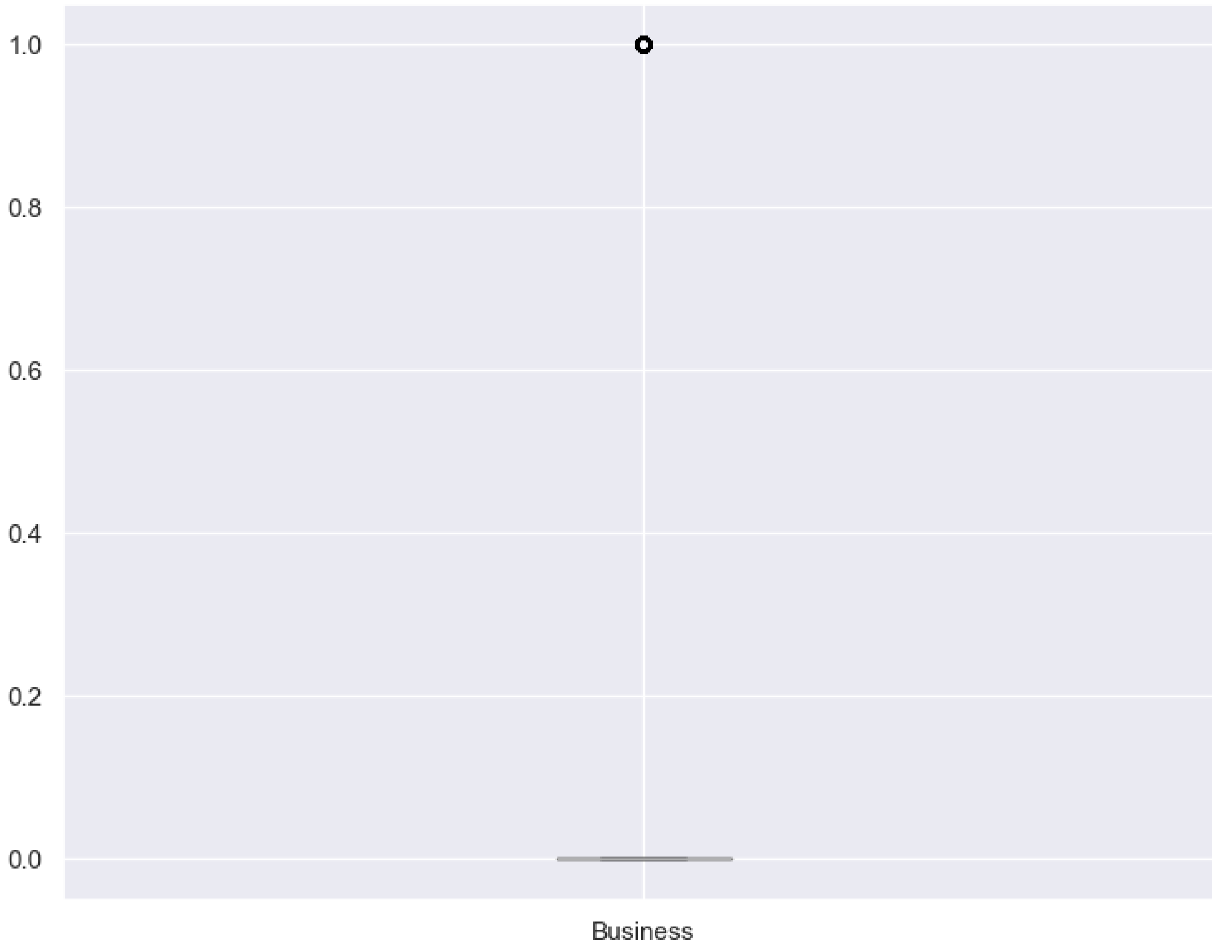

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Age')
```



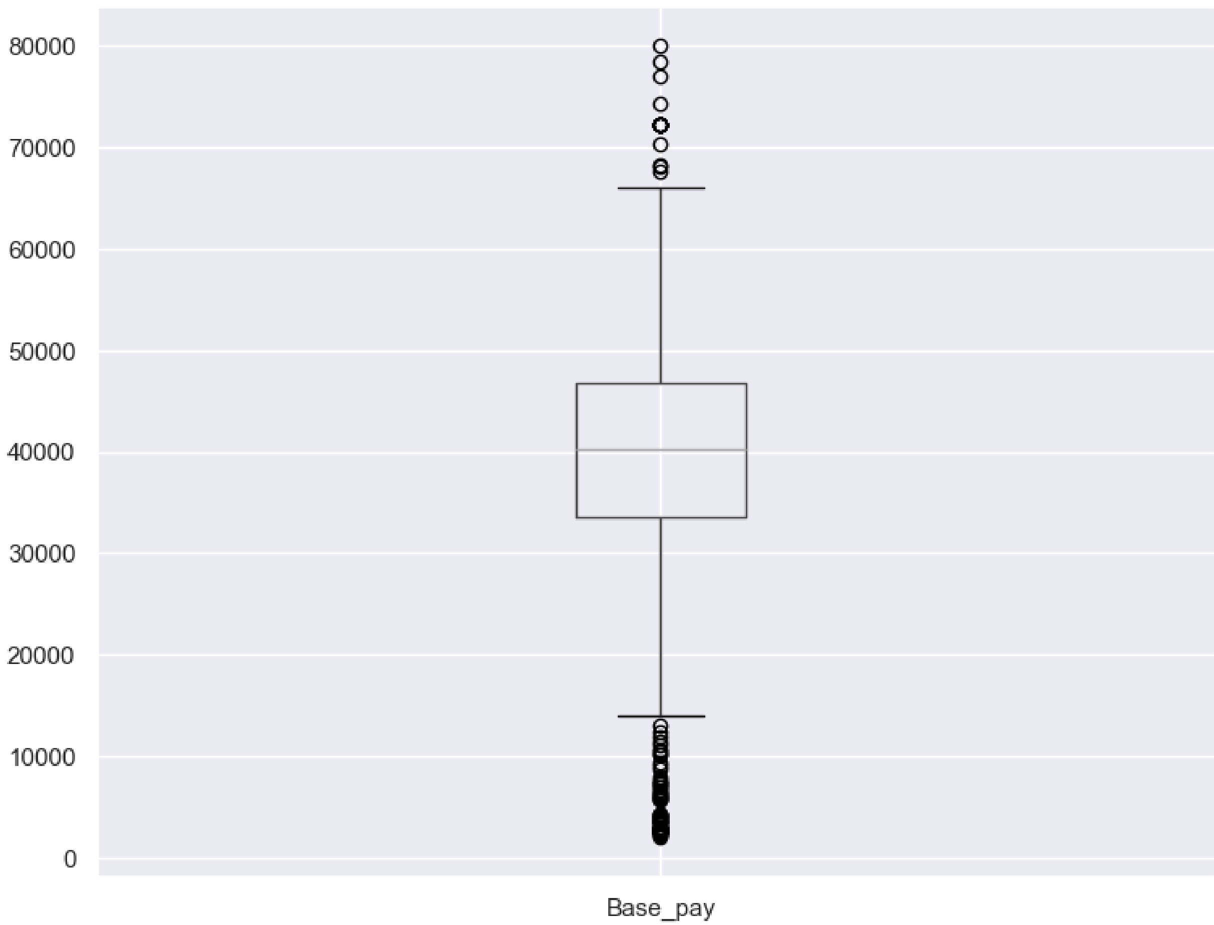

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Salary')
```



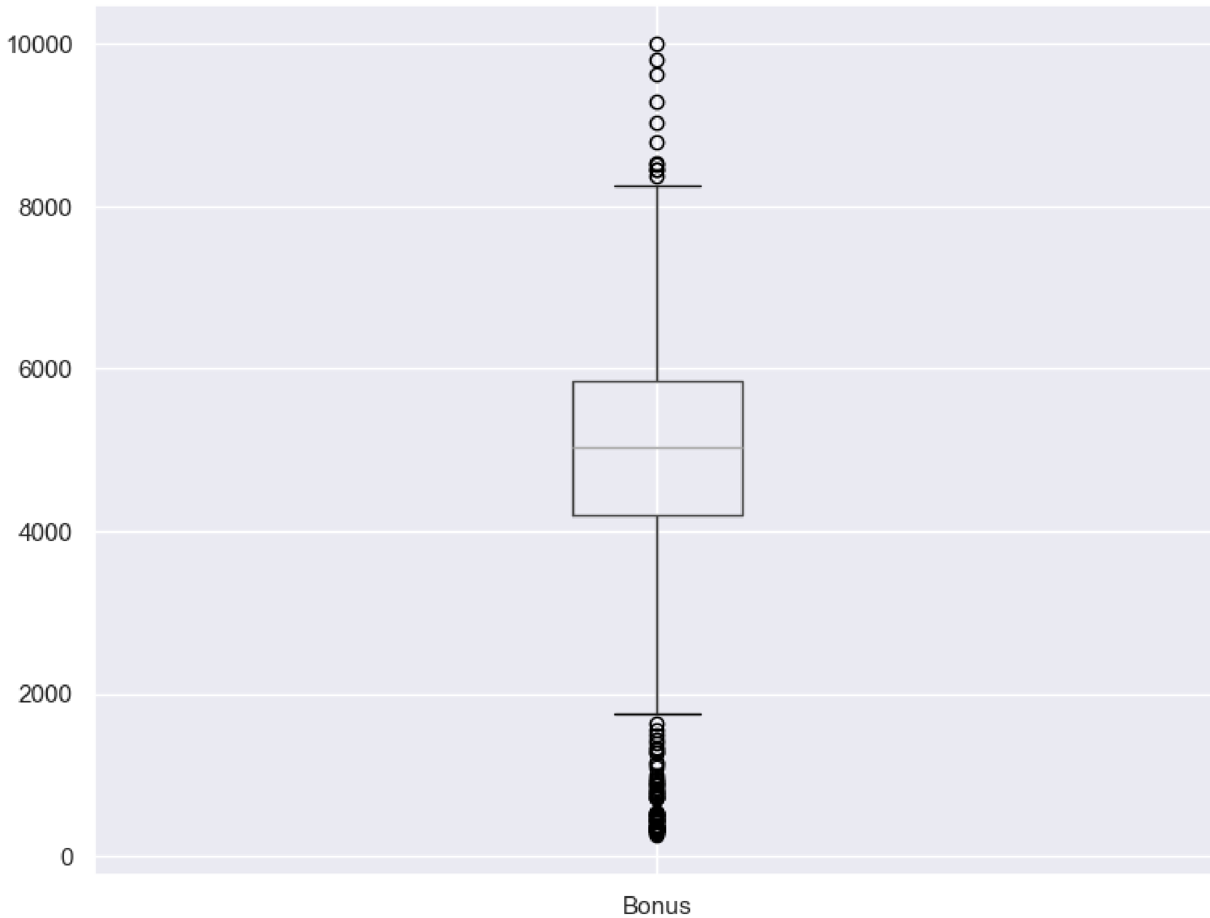

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Business')
```



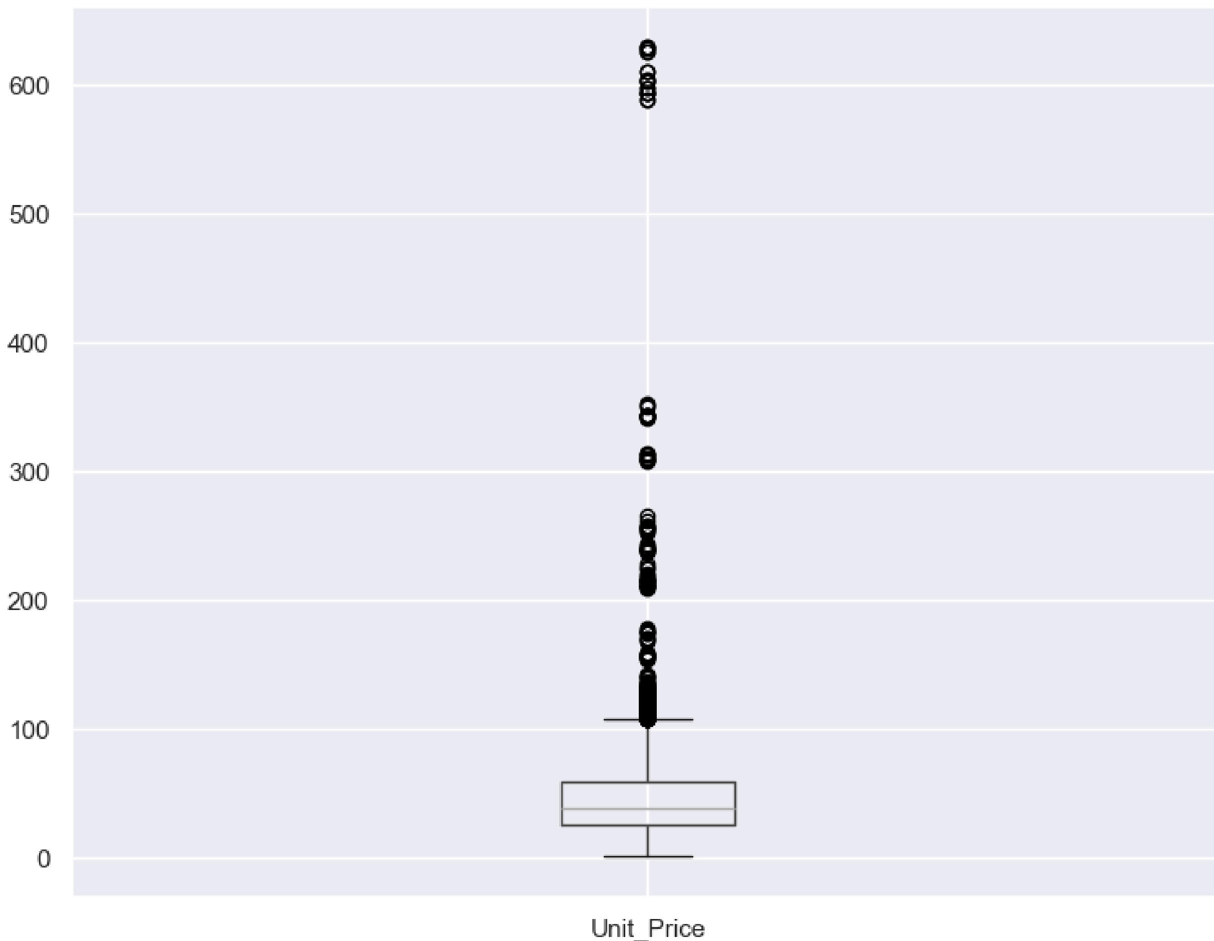

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Base_pay')
```



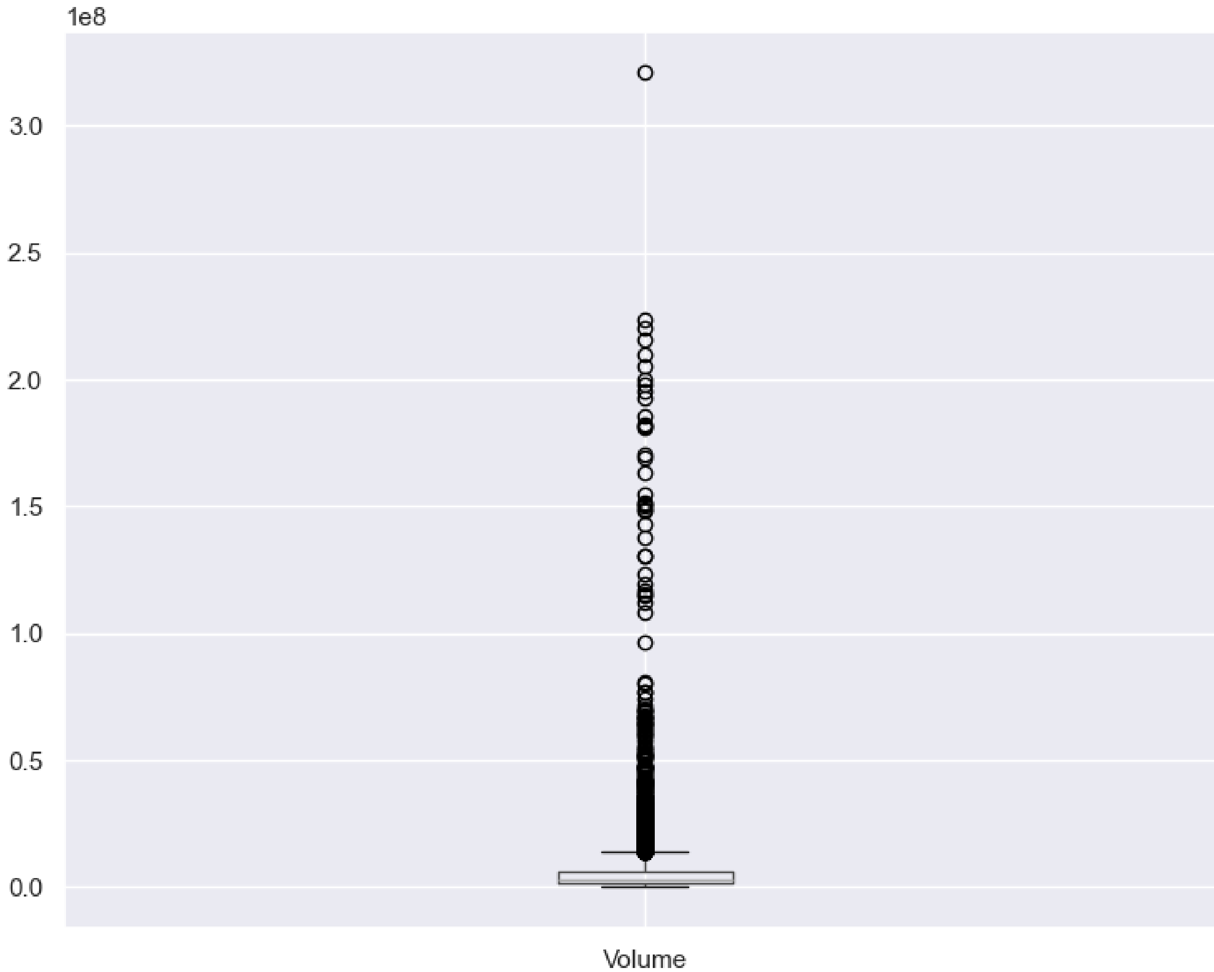

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Bonus')
```



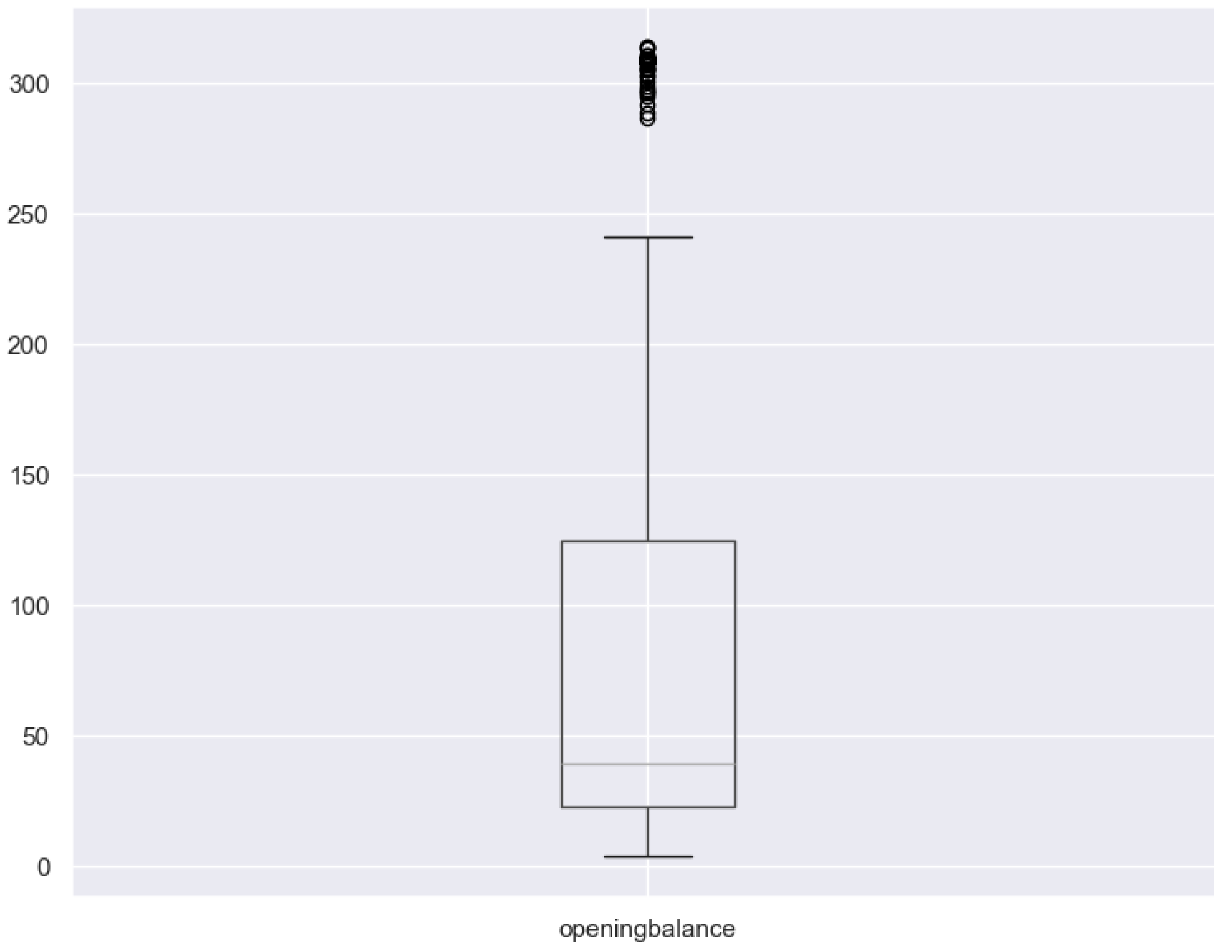

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Unit_Price')
```



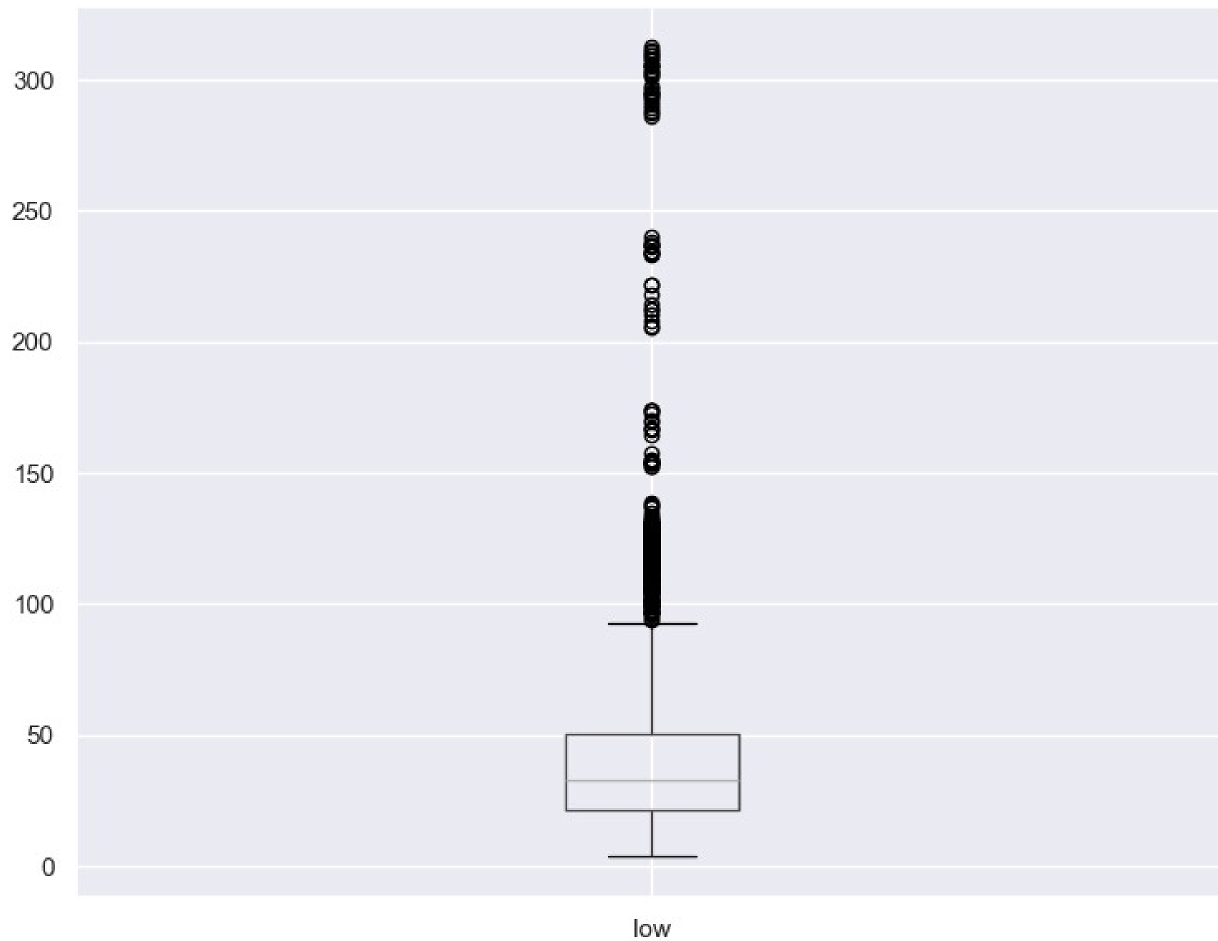

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Volume')
```



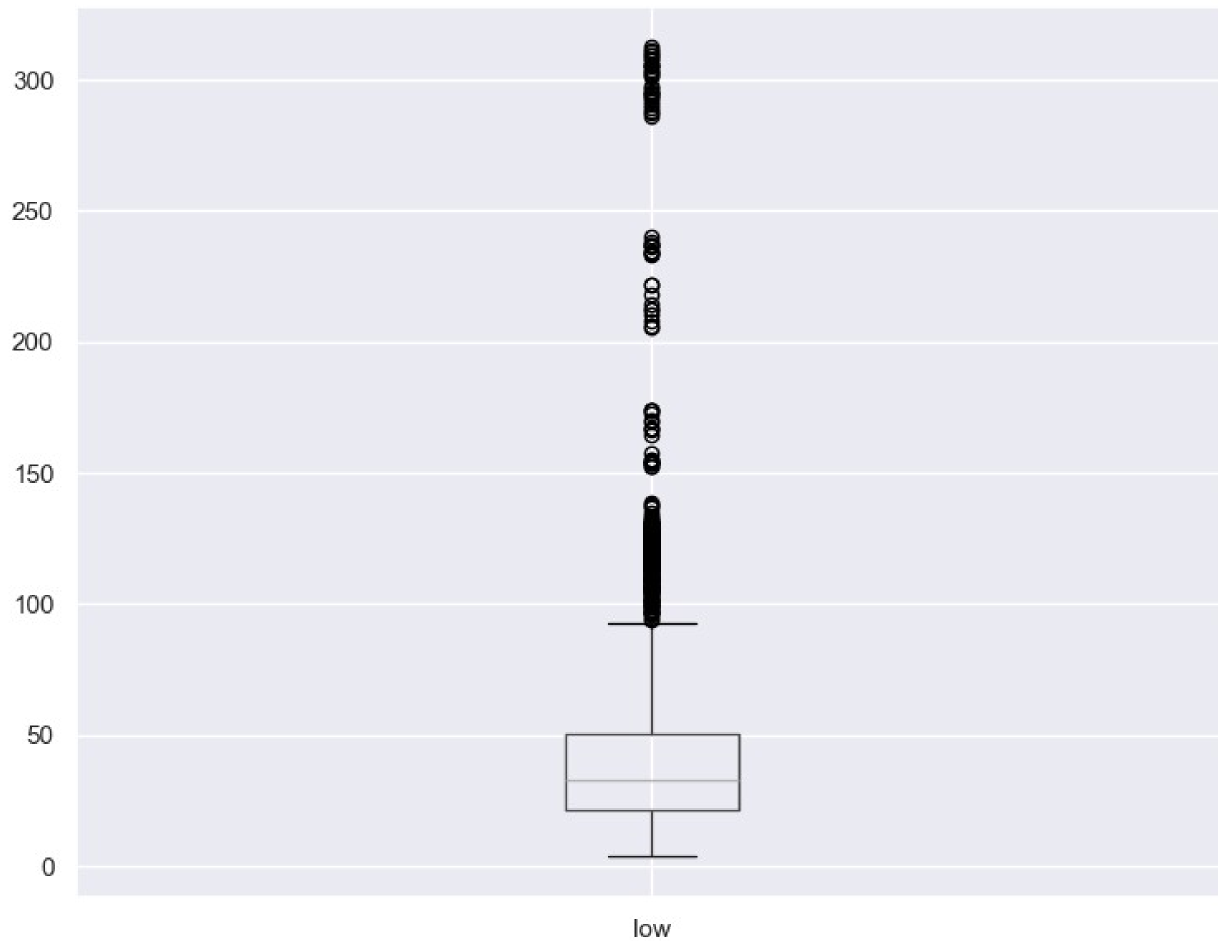

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='openingbalance')
```



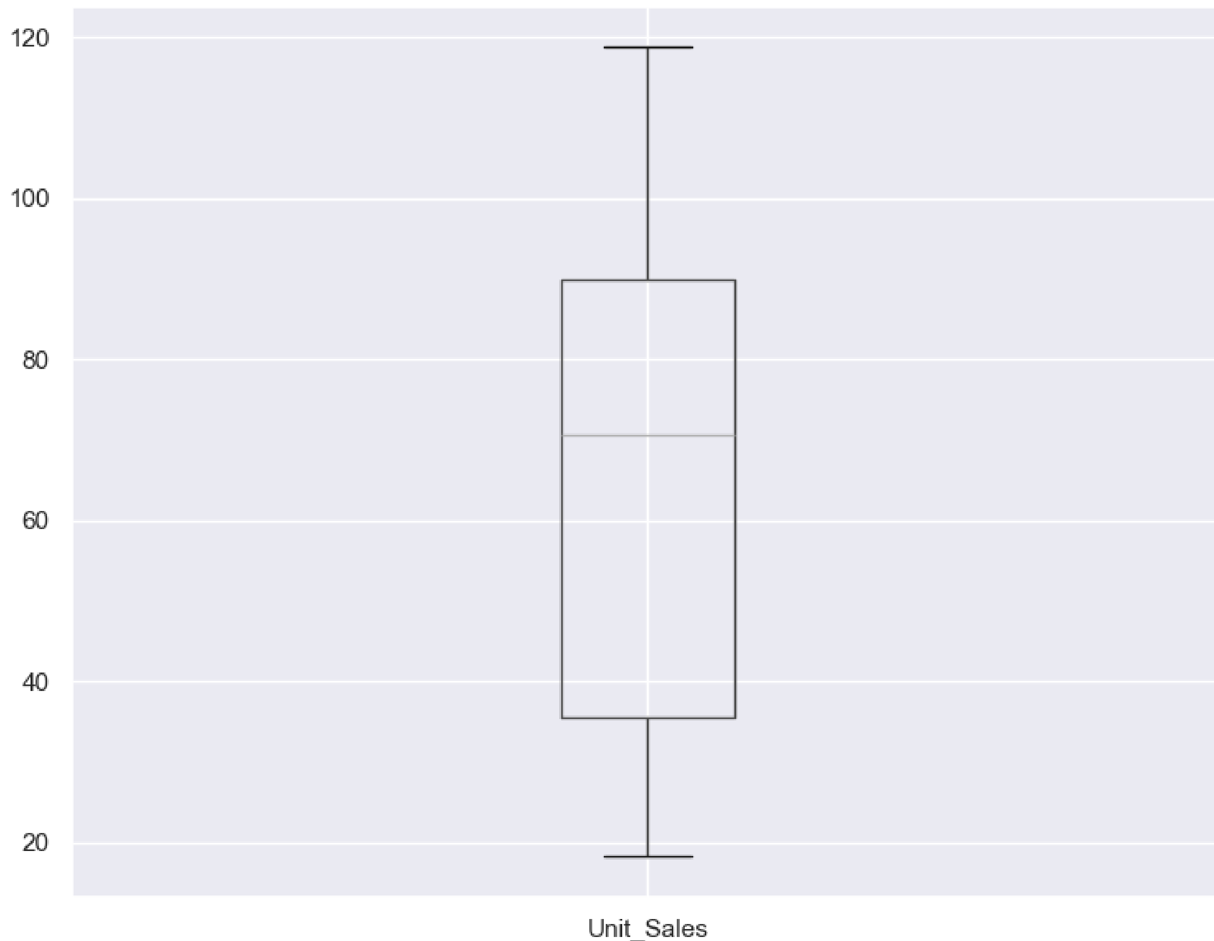

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='low')
```



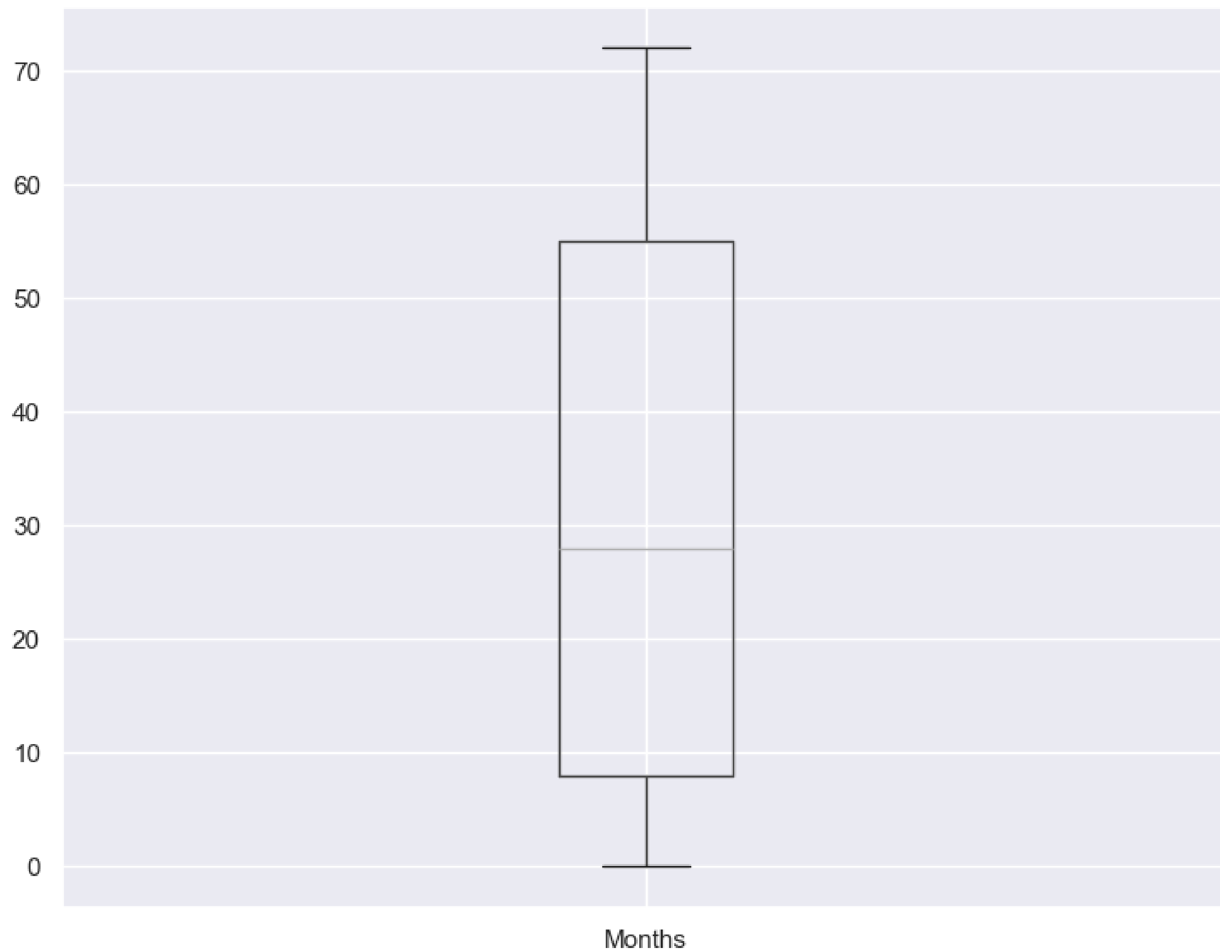

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='low')
```




```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Unit_Sales')
```




```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Months')
```



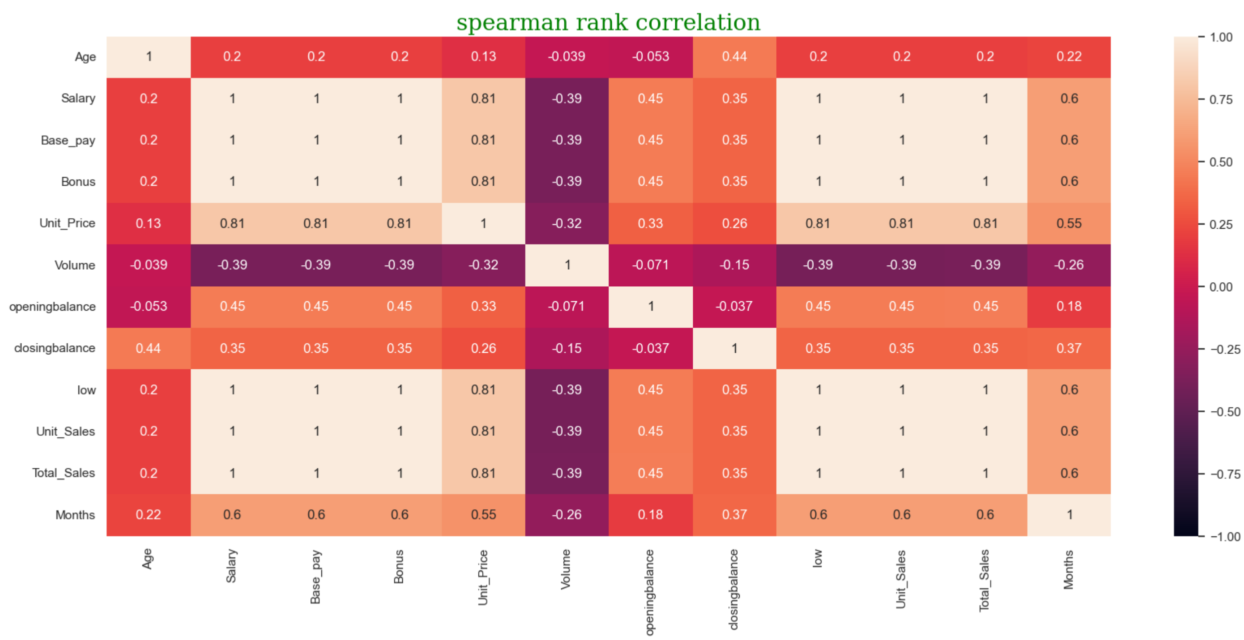

```
plt.figure(figsize=(9,7))
figure=df.boxplot(column='Total_Sales')
```


```
df3=df.drop(['Business'],axis=1)
df3.head()
```

```
   Gender Dependancies Calls            Type Billing Rating  Age
Salary  \
0  Female           No   Yes  Month-to-month      No    Yes   18
5089.00
1  Female           No   Yes  Month-to-month      No    Yes   19
5698.12
2    Male           No   Yes  Month-to-month     Yes     No   22
5896.65
3  Female           No   Yes  Month-to-month     Yes    Yes   21
6125.12
4    Male           No   Yes  Month-to-month     Yes    Yes   23
6245.00

   Base_pay     Bonus  Unit_Price    Volume  openingbalance
closingbalance  \
0  2035.600  254.4500        3.77  21226600            3.75
3.76
1  2279.248  284.9060        3.74  10462800            3.85
```

```
3.68
2  2358.660  294.8325        3.89  18761000            4.23
4.29
3  2450.048  306.2560        4.35  66130600            4.26
4.31
4  2498.000  312.2500        4.34  26868200            4.79
4.41

     low  Unit_Sales  Total_Sales  Months            Education
0  3.65       18.25        18.80       0  High School or less
1  3.65       18.40        18.85       0  High School or less
2  3.72       18.70        18.90       0  High School or less
3  3.83       18.75        19.00       0  High School or less
4  4.08       18.80        19.05       1  High School or less
```

```
df3_numeric=df3.select_dtypes(include='number')
df3_numeric.corr(method="spearman")
plt.figure(figsize=(20,8))
heatmap=sns.heatmap(df3_numeric.corr (method="spearman").round(3),
vmin=-1,vmax=1,annot=True)
font2={'family':'serif','color':'green','size':20}
plt.title("spearman rank correlation", font2)
plt.show()
```




```
# Looking to the percentage of correlation
df3_numeric = df3.select_dtypes(include='number')
df3_numeric.corr(method='spearman')*100
```

```
                       Age      Salary     Base_pay        Bonus
Unit_Price  \
```

```
Age             100.000000   20.228180   20.212073   20.228180
12.826637
Salary           20.228180  100.000000   99.997589  100.000000
81.171311
Base_pay         20.212073   99.997589  100.000000   99.997589
81.168113
Bonus            20.228180  100.000000   99.997589  100.000000
81.171311
Unit_Price       12.826637   81.171311   81.168113   81.171311
100.000000
Volume           -3.934477  -39.048779  -39.042656  -39.048779  -
32.422983
openingbalance   -5.291931   45.346909   45.285139   45.346909
33.340773
closingbalance   44.098882   34.649833   34.640860   34.649833
26.369923
low              20.203950   99.985870   99.983405   99.985870
81.241207
Unit_Sales       20.226808   99.999741   99.997324   99.999741
81.173482
Total_Sales      20.217334   99.996520   99.994102   99.996520
81.170467
Months           22.283665   60.379576   60.376626   60.379576
55.006199

                    Volume  openingbalance  closingbalance         low
\
Age              -3.934477       -5.291931       44.098882   20.203950

Salary          -39.048779       45.346909       34.649833   99.985870

Base_pay        -39.042656       45.285139       34.640860   99.983405

Bonus           -39.048779       45.346909       34.649833   99.985870

Unit_Price      -32.422983       33.340773       26.369923   81.241207

Volume          100.000000       -7.115316      -14.710785  -39.220924

openingbalance   -7.115316      100.000000       -3.721154   45.343163

closingbalance  -14.710785       -3.721154      100.000000   34.627752

low             -39.220924       45.343163       34.627752  100.000000

Unit_Sales      -39.050546       45.347171       34.648512   99.985599

Total_Sales     -39.048003       45.358275       34.642390   99.982649

Months          -26.207721       18.500185       36.592681   60.375227
```

```
                 Unit_Sales  Total_Sales      Months
Age               20.226808    20.217334   22.283665
Salary            99.999741    99.996520   60.379576
Base_pay          99.997324    99.994102   60.376626
Bonus             99.999741    99.996520   60.379576
Unit_Price        81.173482    81.170467   55.006199
Volume           -39.050546   -39.048003  -26.207721
openingbalance    45.347171    45.358275   18.500185
closingbalance    34.648512    34.642390   36.592681
low               99.985599    99.982649   60.375227
Unit_Sales       100.000000    99.996259   60.377897
Total_Sales       99.996259   100.000000   60.374805
Months            60.377897    60.374805  100.000000
```

```
# Dropping the non-required variables
dff =
df.drop(columns=['Gender','Business','Dependancies','Calls','Type','Bi
lling',

'Rating','Base_pay','Unit_Price','low','Unit_Sales',

'Total_Sales','openingbalance','closingbalance','Volume'])
dff.head()
```

```
   Age   Salary     Bonus  Months            Education
0   18  5089.00  254.4500       0  High School or less
1   19  5698.12  284.9060       0  High School or less
2   22  5896.65  294.8325       0  High School or less
3   21  6125.12  306.2560       0  High School or less
4   23  6245.00  312.2500       1  High School or less
```

```
dff['Education']=dff['Education'].map({'High School or less': 0,
                                       'Intermediate': 1,
                                       'Graduation': 2, 'PG': 3})
#assigning high school =0,Intermediate =1 and gradaution=2

dff.head()
```

```
   Age   Salary     Bonus  Months  Education
0   18  5089.00  254.4500       0          0
1   19  5698.12  284.9060       0          0
2   22  5896.65  294.8325       0          0
3   21  6125.12  306.2560       0          0
4   23  6245.00  312.2500       1          0
```

```
dff.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
```

```
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Age        5000 non-null   int64
 1   Salary     5000 non-null   float64
 2   Bonus      5000 non-null   float64
 3   Months     5000 non-null   int64
 4   Education  5000 non-null   int64
dtypes: float64(2), int64(3)
memory usage: 195.4 KB
```

```
Q1=dff.quantile(0.25).round(3)
Q3=dff.quantile(0.75).round(3)
IQR=Q3-Q1
print(IQR)
```

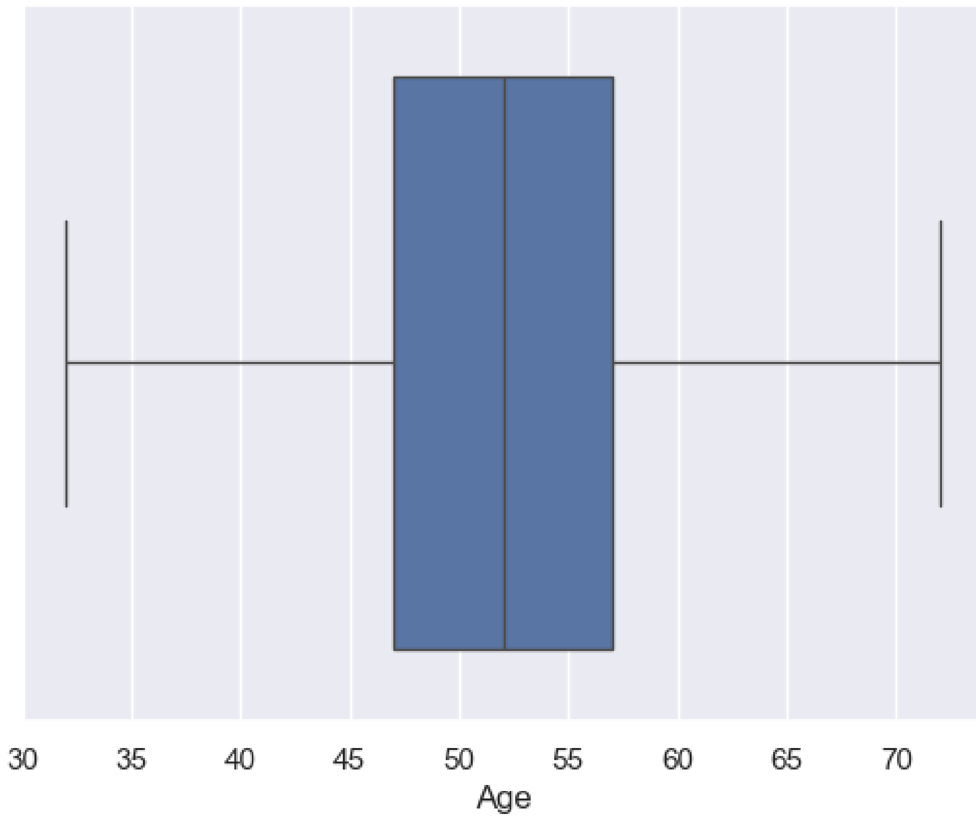
```
Age              10.000
Salary        33021.753
Bonus          1651.088
Months           47.000
Education         1.000
dtype: float64
```

```
dfout = dff[~((dff < (Q1 - 1.5 * IQR)) |(dff > (Q3 + 1.5 *
IQR))).any(axis=1)]
dfout.shape
```

```
(4833, 5)
```

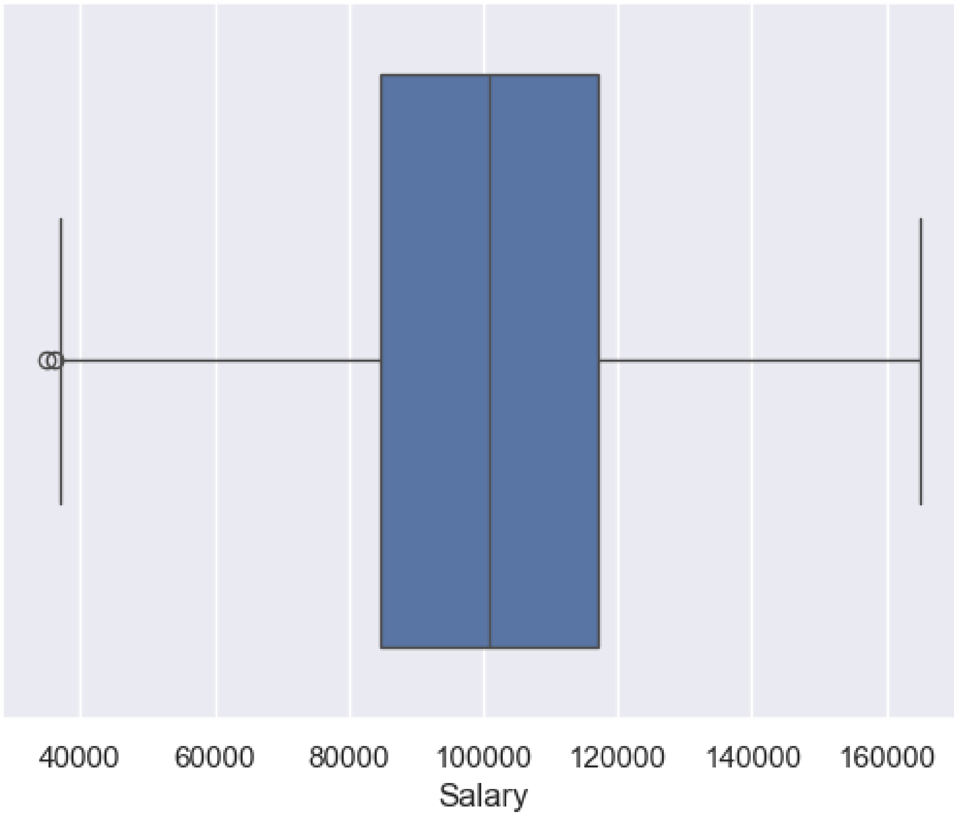
```
sns.boxplot(x=dfout['Age'])
```

```
<Axes: xlabel='Age'>
```



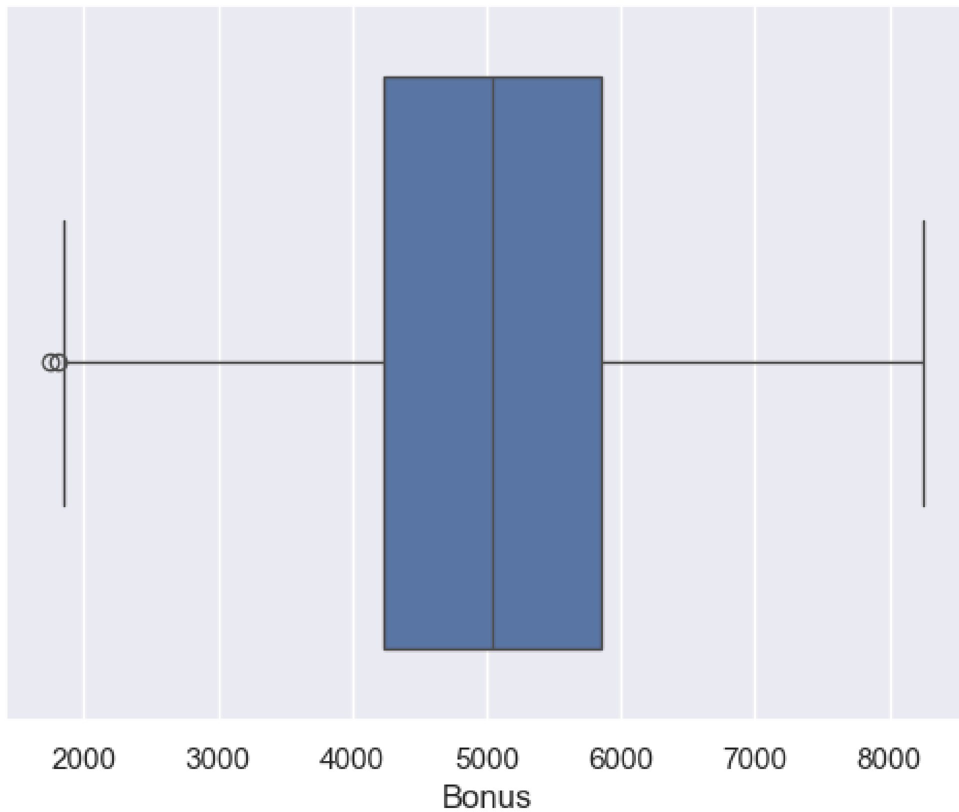

```
sns.boxplot(x=dfout['Salary'])
```

```
<Axes: xlabel='Salary'>
```



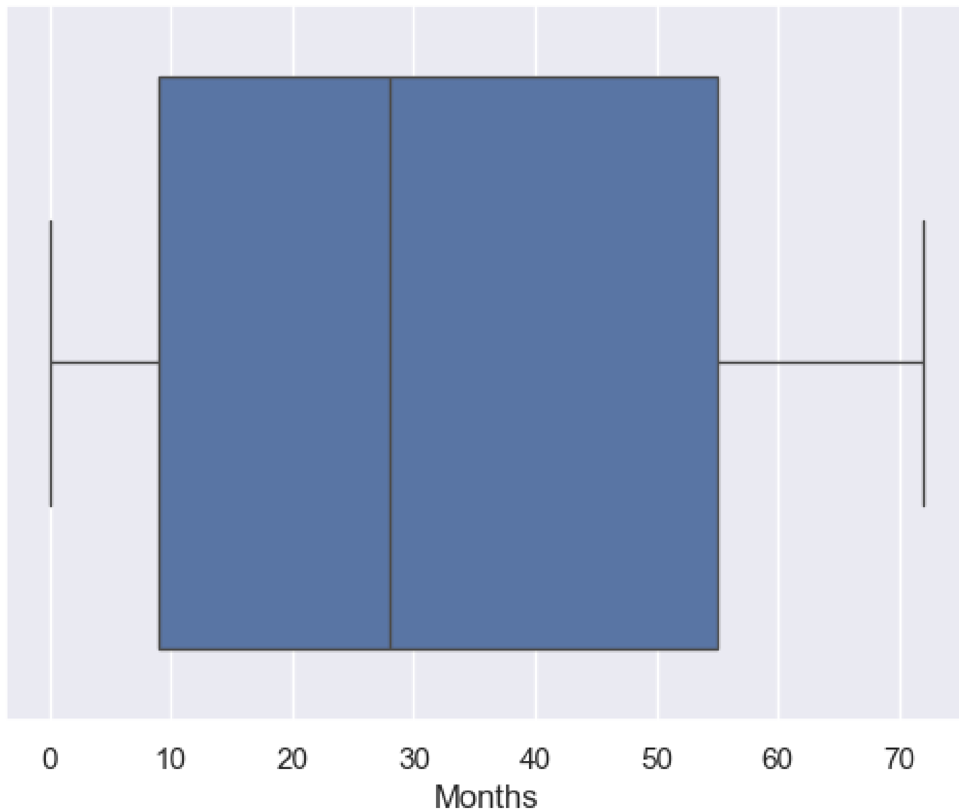

```
sns.boxplot(x=dfout['Bonus'])
```

```
<Axes: xlabel='Bonus'>
```



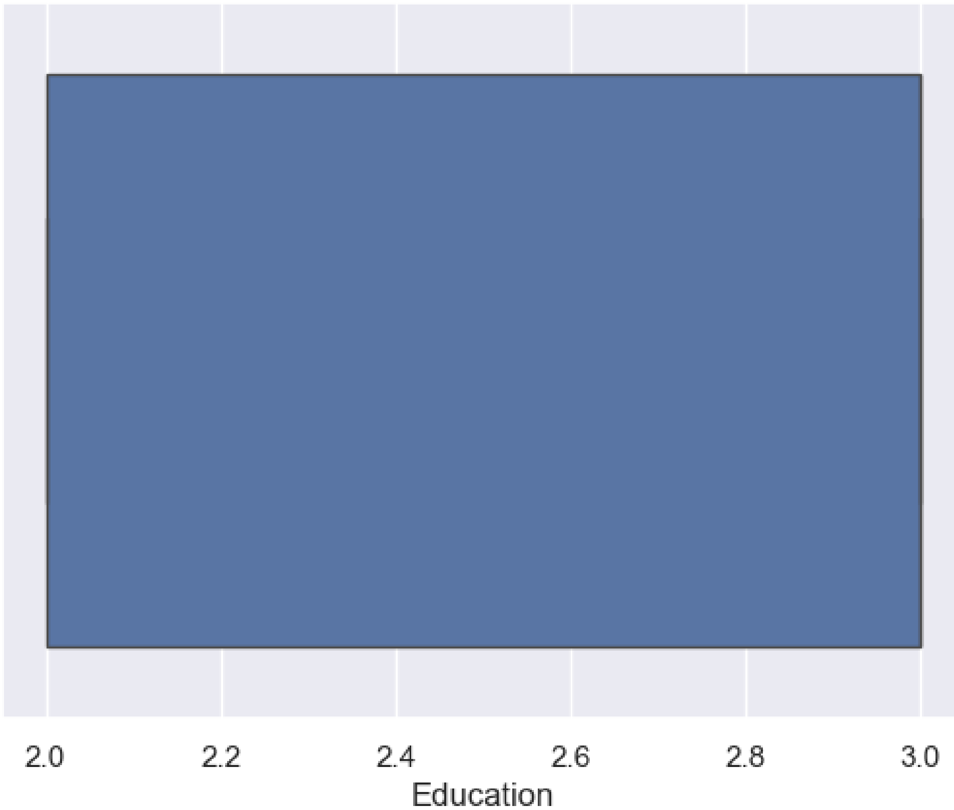

```
sns.boxplot(x=dfout['Months'])
```

```
<Axes: xlabel='Months'>
```




```
sns.boxplot(x=dfout['Education'])
```

```
<Axes: xlabel='Education'>
```




```
x=dff.drop(['Salary'],axis=1)
x
```

```
      Age      Bonus  Months  Education
0      18   254.4500       0          0
1      19   284.9060       0          0
2      22   294.8325       0          0
3      21   306.2560       0          0
4      23   312.2500       1          0
...   ...        ...     ...        ...
4995   72  9034.8400      72          3
4996   73  9284.2950      72          3
4997   74  9631.8400      72          3
4998   74  9798.5350      72          3
4999   88  9998.5370      72          3

[5000 rows x 4 columns]
```

```
y=dff['Salary']
y
```

```
0       5089.00
1       5698.12
2       5896.65
3       6125.12
```

```
4         6245.00
            ...
4995    180696.80
4996    185685.90
4997    192636.80
4998    195970.70
4999    199970.74
Name: Salary, Length: 5000, dtype: float64
```

```
from sklearn.model_selection import train_test_split
x_train, x_test,y_train,
y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

```
from sklearn.preprocessing import StandardScaler
SC=StandardScaler()
x_train=SC.fit_transform(x_train)
x_test=SC.transform(x_test)
```

```
y_train
```

```
2858    104938.82440
1559     88835.89005
1441     86903.28139
2179     96844.15570
1390     86096.55833
            ...
4931    151880.35890
3264    110244.51700
1653     90043.98773
2607    101985.79940
2732    103369.50500
Name: Salary, Length: 3500, dtype: float64
```

```
dff.head()
```

	Age	Salary	Bonus	Months	Education
0	18	5089.00	254.4500	0	0
1	19	5698.12	284.9060	0	0
2	22	5896.65	294.8325	0	0
3	21	6125.12	306.2560	0	0
4	23	6245.00	312.2500	1	0

```
x_test
```

```
array([[-0.33476566, -1.36317701, -0.81633125,  0.79882921],
       [-1.49312433,  0.7314342 ,  0.64158909, -1.14142217],
       [ 0.82359302,  1.71012811,  1.61353598,  0.79882921],
       ...,
       [ 1.40277236, -0.71199409,  1.24905589,  0.79882921],
       [-0.45060152,  0.23566474, -0.41135338,  0.79882921],
       [-1.02978086, -0.5458082 , -1.26180691,  0.79882921]])
```

```
y_test
```

```
398      64631.43297
3833    118137.14110
4836    143137.34650
4572    133346.50250
636      73049.74210
            ...     
4554    132723.77580
4807    141312.27900
1073     81265.54772
2906    105472.97890
1357     85510.67637
Name: Salary, Length: 1500, dtype: float64
```

```
# data normalization with sklearn
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
# fit scaler on training data
x_train = sc.fit_transform(x_train)

# transform testing data
x_test = sc.transform(x_test)
```

```
x_train
```

```
array([[-0.79810913,  0.21475396, -0.45185117, -1.14142217],
       [-0.56643739, -0.41563461, -1.26180691,  0.79882921],
       [-0.79810913, -0.49129128, -1.26180691,  0.79882921],
       ...,
       [-0.45060152, -0.36834069, -1.22130912, -1.14142217],
       [ 1.17110062,  0.09915061, -0.69483789, -1.14142217],
       [ 1.86611582,  0.15331913, -0.57334453, -1.14142217]])
```

```
x_test
```

```
array([[-0.33476566, -1.36317701, -0.81633125,  0.79882921],
       [-1.49312433,  0.7314342 ,  0.64158909, -1.14142217],
       [ 0.82359302,  1.71012811,  1.61353598,  0.79882921],
       ...,
       [ 1.40277236, -0.71199409,  1.24905589,  0.79882921],
       [-0.45060152,  0.23566474, -0.41135338,  0.79882921],
       [-1.02978086, -0.5458082 , -1.26180691,  0.79882921]])
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error

#Create and train the Linear Regression model
model_lr = LinearRegression()
model_lr.fit(x_train,y_train)
```

```python
# Make predictions
y_pred_lr = model_lr.predict(x_test)

# Calculate performance metrics
mae_lr = mean_absolute_error(y_test, y_pred_lr)
mse_lr = mean_squared_error(y_test, y_pred_lr)
rmse_lr = np.sqrt(mse_lr)
mape_lr = np.mean(np.abs((y_test - y_pred_lr) / np.abs(y_test)))
accuracy_lr = 100 * (1- mape_lr)

#Print the performance metrics for linear regression
print('Linear Regression Metrics:')
print('Mean Absolute Error (MAE):', mae_lr)
print('Mean Squared Error(MSE):',mse_lr)
print('Root Mean Squared Error(RMSE):',rmse_lr)
print('Mean Absolute Percentage Error (MAPE):',round(mape_lr * 100,2))
print('Accuracy:',round(accuracy_lr,2))
```

```
Linear Regression Metrics:
Mean Absolute Error (MAE): 2.9884760533605003e-06
Mean Squared Error(MSE): 1.4433781272304171e-11
Root Mean Squared Error(RMSE): 3.7991816582395965e-06
Mean Absolute Percentage Error (MAPE): 0.0
Accuracy: 100.0
```

```python
from sklearn.tree import DecisionTreeRegressor

#create and train the Decision Tree model
model = DecisionTreeRegressor()
model.fit(x_train,y_train)

#make predictions
y_pred=model.predict(x_test)

#calculate performance metrics
mae=mean_absolute_error(y_test,y_pred)
mse=mean_squared_error(y_test,y_pred)
rmse=np.sqrt(mse)
mape=np.mean(np.abs((y_test-y_pred)/np.abs(y_test)))
accuracy=100*(1-mape)

#print the performance metrics
print('Mean Absolute Error (MAE):',mae)
print('Mean Squared Error(MSE):',mse)
print('Root Mean Squared Error (RMSE):',rmse)
print('Mean Absolute Percentage Error (MAPE):',round(mape*100,2))
print('Accuracy:',round(accuracy,2))
```

```
Mean Absolute Error (MAE): 26.015430194666482
Mean Squared Error(MSE): 20819.817952338522
Root Mean Squared Error (RMSE): 144.29074104854587
```

```
Mean Absolute Percentage Error (MAPE): 0.05
Accuracy: 99.95
```

```python
from sklearn.ensemble import RandomForestRegressor

#create and train the Random Forest model
model_rf=RandomForestRegressor(n_estimators=100,random_state=42)
model_rf.fit(x_train,y_train)

#Make Predictions
y_pred_rf=model_rf.predict(x_test)

#Calculate Performance metrics
mae_rf=mean_absolute_error(y_test,y_pred_rf)
mse_rf=mean_squared_error(y_test,y_pred_rf)
rmse_rf=np.sqrt(mse_rf)
mape_rf=np.mean(np.abs((y_test-y_pred_rf)/np.abs(y_test)))
accuracy_rf = 100 * (1 - mape_rf)

#print the performance metrics random forest
print('Random Forest Metrics:')
print('Mean Absolute Error (MAE):',mae_rf)
print('Mean Squared Error(MSE):',mse)
print('Root Mean Squared Error (RMSE):',rmse_rf)
print('Mean Absolute Percentage Error (MAPE):',round(mape_rf*100,2))
print('Accuracy:',round(accuracy_rf,2))
```

```
Random Forest Metrics:
Mean Absolute Error (MAE): 21.23846068615708
Mean Squared Error(MSE): 20819.817952338522
Root Mean Squared Error (RMSE): 86.42775989747088
Mean Absolute Percentage Error (MAPE): 0.06
Accuracy: 99.94
```

```
pip install xgBoost
```

```
Defaulting to user installation because normal site-packages is not
writeableNote: you may need to restart the kernel to use updated
packages.

Collecting xgBoost
  Downloading xgboost-3.1.2-py3-none-win_amd64.whl.metadata (2.1 kB)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\
site-packages (from xgBoost) (2.1.3)
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\
site-packages (from xgBoost) (1.15.3)
Downloading xgboost-3.1.2-py3-none-win_amd64.whl (72.0 MB)
   ---------------------------------------- 0.0/72.0 MB ? eta -:--:--
   ---------------------------------------- 0.0/72.0 MB ? eta -:--:--
   ---------------------------------------- 0.3/72.0 MB ? eta -:--:--
   ---------------------------------------- 0.8/72.0 MB 1.8 MB/s eta
```

```
0:00:40
    --------------------------------------- 1.0/72.0 MB 1.4 MB/s eta
0:00:52
    --------------------------------------- 1.0/72.0 MB 1.4 MB/s eta
0:00:52
    --------------------------------------- 1.0/72.0 MB 1.4 MB/s eta
0:00:52
    --------------------------------------- 1.6/72.0 MB 1.1 MB/s eta
0:01:07
   - -------------------------------------- 1.8/72.0 MB 1.1 MB/s eta
0:01:06
   - -------------------------------------- 2.1/72.0 MB 1.1 MB/s eta
0:01:04
   - -------------------------------------- 2.1/72.0 MB 1.1 MB/s eta
0:01:04
   - -------------------------------------- 2.6/72.0 MB 1.1 MB/s eta
0:01:05
   - -------------------------------------- 2.6/72.0 MB 1.1 MB/s eta
0:01:05
   - -------------------------------------- 2.6/72.0 MB 1.1 MB/s eta
0:01:05
   - -------------------------------------- 2.9/72.0 MB 949.7 kB/s eta
0:01:13
   - -------------------------------------- 3.1/72.0 MB 954.1 kB/s eta
0:01:13
   -- ------------------------------------- 3.7/72.0 MB 1.0 MB/s eta
0:01:08
   -- ------------------------------------- 3.9/72.0 MB 1.0 MB/s eta
0:01:06
   -- ------------------------------------- 4.2/72.0 MB 1.0 MB/s eta
0:01:06
   -- ------------------------------------- 4.5/72.0 MB 1.0 MB/s eta
0:01:05
   -- ------------------------------------- 5.0/72.0 MB 1.1 MB/s eta
0:01:02
   -- ------------------------------------- 5.2/72.0 MB 1.1 MB/s eta
0:01:01
   --- ------------------------------------ 5.5/72.0 MB 1.1 MB/s eta
0:01:00
   --- ------------------------------------ 5.8/72.0 MB 1.1 MB/s eta
0:01:00
   --- ------------------------------------ 6.0/72.0 MB 1.1 MB/s eta
0:00:59
   --- ------------------------------------ 6.6/72.0 MB 1.1 MB/s eta
0:00:58
   --- ------------------------------------ 6.6/72.0 MB 1.1 MB/s eta
0:00:58
   --- ------------------------------------ 7.1/72.0 MB 1.2 MB/s eta
0:00:56
```

```
   ---- ----------------------------------- 7.3/72.0 MB 1.2 MB/s eta
0:00:55
   ---- ----------------------------------- 7.6/72.0 MB 1.2 MB/s eta
0:00:55
   ---- ----------------------------------- 7.9/72.0 MB 1.2 MB/s eta
0:00:56
   ---- ----------------------------------- 8.1/72.0 MB 1.2 MB/s eta
0:00:56
   ---- ----------------------------------- 8.1/72.0 MB 1.2 MB/s eta
0:00:56
   ---- ----------------------------------- 8.7/72.0 MB 1.1 MB/s eta
0:00:56
   ---- ----------------------------------- 8.9/72.0 MB 1.1 MB/s eta
0:00:55
   ----- ---------------------------------- 9.2/72.0 MB 1.2 MB/s eta
0:00:55
   ----- ---------------------------------- 9.4/72.0 MB 1.2 MB/s eta
0:00:55
   ----- ---------------------------------- 9.4/72.0 MB 1.2 MB/s eta
0:00:55
   ----- ---------------------------------- 9.7/72.0 MB 1.1 MB/s eta
0:00:55
   ----- ---------------------------------- 10.0/72.0 MB 1.1 MB/s eta
0:00:55
   ----- ---------------------------------- 10.2/72.0 MB 1.1 MB/s eta
0:00:56
   ----- ---------------------------------- 10.2/72.0 MB 1.1 MB/s eta
0:00:56
   ----- ---------------------------------- 10.5/72.0 MB 1.1 MB/s eta
0:00:57
   ----- ---------------------------------- 10.5/72.0 MB 1.1 MB/s eta
0:00:57
   ----- ---------------------------------- 10.7/72.0 MB 1.1 MB/s eta
0:00:57
   ------ --------------------------------- 11.0/72.0 MB 1.1 MB/s eta
0:00:57
   ------ --------------------------------- 11.3/72.0 MB 1.1 MB/s eta
0:00:57
   ------ --------------------------------- 11.8/72.0 MB 1.1 MB/s eta
0:00:56
   ------ --------------------------------- 11.8/72.0 MB 1.1 MB/s eta
0:00:56
   ------ --------------------------------- 12.1/72.0 MB 1.1 MB/s eta
0:00:56
   ------ --------------------------------- 12.3/72.0 MB 1.1 MB/s eta
0:00:56
   ------ --------------------------------- 12.3/72.0 MB 1.1 MB/s eta
0:00:56
   ------ --------------------------------- 12.6/72.0 MB 1.1 MB/s eta
```

```
0:00:56
   ------ --------------------------------- 12.6/72.0 MB 1.1 MB/s eta
0:00:56
   ------- -------------------------------- 12.8/72.0 MB 1.0 MB/s eta
0:00:57
   ------- -------------------------------- 13.1/72.0 MB 1.0 MB/s eta
0:00:57
   ------- -------------------------------- 13.4/72.0 MB 1.0 MB/s eta
0:00:57
   ------- -------------------------------- 13.6/72.0 MB 1.0 MB/s eta
0:00:57
   ------- -------------------------------- 13.9/72.0 MB 1.0 MB/s eta
0:00:57
   ------- -------------------------------- 14.2/72.0 MB 1.0 MB/s eta
0:00:56
   -------- ------------------------------- 14.4/72.0 MB 1.0 MB/s eta
0:00:56
   -------- ------------------------------- 14.7/72.0 MB 1.0 MB/s eta
0:00:55
   -------- ------------------------------- 14.9/72.0 MB 1.0 MB/s eta
0:00:55
   -------- ------------------------------- 15.2/72.0 MB 1.0 MB/s eta
0:00:55
   -------- ------------------------------- 15.5/72.0 MB 1.1 MB/s eta
0:00:54
   -------- ------------------------------- 15.7/72.0 MB 1.1 MB/s eta
0:00:54
   -------- ------------------------------- 16.0/72.0 MB 1.1 MB/s eta
0:00:53
   --------- ------------------------------ 16.3/72.0 MB 1.1 MB/s eta
0:00:53
   --------- ------------------------------ 16.5/72.0 MB 1.1 MB/s eta
0:00:53
   --------- ------------------------------ 16.8/72.0 MB 1.1 MB/s eta
0:00:53
   --------- ------------------------------ 17.0/72.0 MB 1.1 MB/s eta
0:00:52
   --------- ------------------------------ 17.3/72.0 MB 1.1 MB/s eta
0:00:52
   --------- ------------------------------ 17.8/72.0 MB 1.1 MB/s eta
0:00:51
   ---------- ----------------------------- 18.1/72.0 MB 1.1 MB/s eta
0:00:51
   ---------- ----------------------------- 18.4/72.0 MB 1.1 MB/s eta
0:00:50
   ---------- ----------------------------- 18.4/72.0 MB 1.1 MB/s eta
0:00:50
   ---------- ----------------------------- 18.6/72.0 MB 1.1 MB/s eta
0:00:51
```

```
   ---------- ----------------------------- 19.1/72.0 MB 1.1 MB/s eta
0:00:50
   ---------- ----------------------------- 19.7/72.0 MB 1.1 MB/s eta
0:00:48
   ----------- ---------------------------- 19.9/72.0 MB 1.1 MB/s eta
0:00:48
   ----------- ---------------------------- 20.2/72.0 MB 1.1 MB/s eta
0:00:48
   ----------- ---------------------------- 20.7/72.0 MB 1.1 MB/s eta
0:00:47
   ----------- ---------------------------- 21.2/72.0 MB 1.1 MB/s eta
0:00:46
   ----------- ---------------------------- 21.2/72.0 MB 1.1 MB/s eta
0:00:46
   ----------- ---------------------------- 21.2/72.0 MB 1.1 MB/s eta
0:00:46
   ----------- ---------------------------- 21.5/72.0 MB 1.1 MB/s eta
0:00:46
   ------------ --------------------------- 21.8/72.0 MB 1.1 MB/s eta
0:00:46
   ------------ --------------------------- 22.0/72.0 MB 1.1 MB/s eta
0:00:46
   ------------ --------------------------- 22.3/72.0 MB 1.1 MB/s eta
0:00:46
   ------------ --------------------------- 22.8/72.0 MB 1.1 MB/s eta
0:00:45
   ------------ --------------------------- 23.1/72.0 MB 1.1 MB/s eta
0:00:44
   ------------- -------------------------- 23.6/72.0 MB 1.1 MB/s eta
0:00:44
   ------------- -------------------------- 23.6/72.0 MB 1.1 MB/s eta
0:00:44
   ------------- -------------------------- 23.9/72.0 MB 1.1 MB/s eta
0:00:44
   ------------- -------------------------- 24.1/72.0 MB 1.1 MB/s eta
0:00:43
   ------------- -------------------------- 24.1/72.0 MB 1.1 MB/s eta
0:00:43
   ------------- -------------------------- 24.1/72.0 MB 1.1 MB/s eta
0:00:43
   ------------- -------------------------- 24.4/72.0 MB 1.1 MB/s eta
0:00:44
   ------------- -------------------------- 24.6/72.0 MB 1.1 MB/s eta
0:00:44
   ------------- -------------------------- 24.9/72.0 MB 1.1 MB/s eta
0:00:44
   ------------- -------------------------- 24.9/72.0 MB 1.1 MB/s eta
0:00:44
   ------------- -------------------------- 25.2/72.0 MB 1.1 MB/s eta
```

```
0:00:44
   ------------- -------------------------- 25.2/72.0 MB 1.1 MB/s eta
0:00:44
   -------------- ------------------------- 25.4/72.0 MB 1.1 MB/s eta
0:00:44
   -------------- ------------------------- 25.4/72.0 MB 1.1 MB/s eta
0:00:44
   -------------- ------------------------- 25.4/72.0 MB 1.1 MB/s eta
0:00:44
   -------------- ------------------------- 25.7/72.0 MB 1.1 MB/s eta
0:00:45
   -------------- ------------------------- 26.0/72.0 MB 1.0 MB/s eta
0:00:44
   -------------- ------------------------- 26.5/72.0 MB 1.1 MB/s eta
0:00:44
   -------------- ------------------------- 26.7/72.0 MB 1.1 MB/s eta
0:00:43
   --------------- ------------------------ 27.0/72.0 MB 1.1 MB/s eta
0:00:43
   --------------- ------------------------ 27.3/72.0 MB 1.1 MB/s eta
0:00:43
   --------------- ------------------------ 27.5/72.0 MB 1.1 MB/s eta
0:00:42
   --------------- ------------------------ 27.8/72.0 MB 1.1 MB/s eta
0:00:42
   --------------- ------------------------ 28.0/72.0 MB 1.1 MB/s eta
0:00:42
   --------------- ------------------------ 28.3/72.0 MB 1.1 MB/s eta
0:00:42
   --------------- ------------------------ 28.6/72.0 MB 1.1 MB/s eta
0:00:41
   ---------------- ----------------------- 28.8/72.0 MB 1.1 MB/s eta
0:00:41
   ---------------- ----------------------- 29.4/72.0 MB 1.1 MB/s eta
0:00:40
   ---------------- ----------------------- 29.4/72.0 MB 1.1 MB/s eta
0:00:40
   ---------------- ----------------------- 29.6/72.0 MB 1.1 MB/s eta
0:00:40
   ---------------- ----------------------- 29.9/72.0 MB 1.1 MB/s eta
0:00:40
   ---------------- ----------------------- 30.1/72.0 MB 1.1 MB/s eta
0:00:40
   ----------------- ---------------------- 30.7/72.0 MB 1.1 MB/s eta
0:00:39
   ----------------- ---------------------- 31.2/72.0 MB 1.1 MB/s eta
0:00:38
   ----------------- ---------------------- 31.5/72.0 MB 1.1 MB/s eta
0:00:38
```

```
   ----------------- ---------------------- 32.0/72.0 MB 1.1 MB/s eta
0:00:37
   ----------------- ---------------------- 32.2/72.0 MB 1.1 MB/s eta
0:00:37
   ------------------ --------------------- 32.5/72.0 MB 1.1 MB/s eta
0:00:37
   ------------------ --------------------- 33.0/72.0 MB 1.1 MB/s eta
0:00:36
   ------------------ --------------------- 33.6/72.0 MB 1.1 MB/s eta
0:00:36
   ------------------ --------------------- 33.8/72.0 MB 1.1 MB/s eta
0:00:35
   ------------------- -------------------- 34.3/72.0 MB 1.1 MB/s eta
0:00:34
   ------------------- -------------------- 34.9/72.0 MB 1.1 MB/s eta
0:00:33
   ------------------- -------------------- 35.1/72.0 MB 1.1 MB/s eta
0:00:33
   ------------------- -------------------- 35.7/72.0 MB 1.1 MB/s eta
0:00:33
   -------------------- ------------------- 36.2/72.0 MB 1.1 MB/s eta
0:00:32
   -------------------- ------------------- 36.4/72.0 MB 1.2 MB/s eta
0:00:31
   -------------------- ------------------- 37.2/72.0 MB 1.2 MB/s eta
0:00:30
   -------------------- ------------------- 37.5/72.0 MB 1.2 MB/s eta
0:00:30
   -------------------- ------------------- 37.7/72.0 MB 1.2 MB/s eta
0:00:30
   --------------------- ------------------ 38.0/72.0 MB 1.2 MB/s eta
0:00:30
   --------------------- ------------------ 38.3/72.0 MB 1.2 MB/s eta
0:00:29
   --------------------- ------------------ 38.5/72.0 MB 1.2 MB/s eta
0:00:29
   --------------------- ------------------ 38.8/72.0 MB 1.2 MB/s eta
0:00:29
   --------------------- ------------------ 39.1/72.0 MB 1.2 MB/s eta
0:00:29
   --------------------- ------------------ 39.3/72.0 MB 1.2 MB/s eta
0:00:28
   ---------------------- ----------------- 39.8/72.0 MB 1.2 MB/s eta
0:00:28
   ---------------------- ----------------- 39.8/72.0 MB 1.2 MB/s eta
0:00:28
   ---------------------- ----------------- 40.1/72.0 MB 1.2 MB/s eta
0:00:28
   ---------------------- ----------------- 40.1/72.0 MB 1.2 MB/s eta
```

```
0:00:28
   ---------------------- ----------------- 40.6/72.0 MB 1.1 MB/s eta
0:00:28
   ---------------------- ----------------- 41.2/72.0 MB 1.2 MB/s eta
0:00:27
   ----------------------- ---------------- 41.7/72.0 MB 1.2 MB/s eta
0:00:27
   ----------------------- ---------------- 41.9/72.0 MB 1.2 MB/s eta
0:00:26
   ----------------------- ---------------- 42.5/72.0 MB 1.2 MB/s eta
0:00:26
   ----------------------- ---------------- 42.7/72.0 MB 1.2 MB/s eta
0:00:26
   ------------------------ --------------- 43.3/72.0 MB 1.2 MB/s eta
0:00:25
   ------------------------ --------------- 43.8/72.0 MB 1.2 MB/s eta
0:00:24
   ------------------------ --------------- 44.3/72.0 MB 1.2 MB/s eta
0:00:24
   ------------------------ --------------- 44.8/72.0 MB 1.2 MB/s eta
0:00:23
   ------------------------- -------------- 45.1/72.0 MB 1.2 MB/s eta
0:00:23
   ------------------------- -------------- 45.4/72.0 MB 1.2 MB/s eta
0:00:23
   ------------------------- -------------- 45.9/72.0 MB 1.2 MB/s eta
0:00:22
   ------------------------- -------------- 46.1/72.0 MB 1.2 MB/s eta
0:00:22
   ------------------------- -------------- 46.4/72.0 MB 1.2 MB/s eta
0:00:22
   ------------------------- -------------- 46.7/72.0 MB 1.2 MB/s eta
0:00:21
   -------------------------- ------------- 46.9/72.0 MB 1.2 MB/s eta
0:00:21
   -------------------------- ------------- 47.2/72.0 MB 1.2 MB/s eta
0:00:21
   -------------------------- ------------- 47.7/72.0 MB 1.2 MB/s eta
0:00:20
   -------------------------- ------------- 48.0/72.0 MB 1.2 MB/s eta
0:00:20
   -------------------------- ------------- 48.0/72.0 MB 1.2 MB/s eta
0:00:20
   -------------------------- ------------- 48.0/72.0 MB 1.2 MB/s eta
0:00:20
   -------------------------- ------------- 48.2/72.0 MB 1.2 MB/s eta
0:00:20
   -------------------------- ------------- 48.2/72.0 MB 1.2 MB/s eta
0:00:20
```

```
   -------------------------- ------------- 48.5/72.0 MB 1.2 MB/s eta
0:00:20
   --------------------------- ------------ 48.8/72.0 MB 1.2 MB/s eta
0:00:20
   --------------------------- ------------ 49.0/72.0 MB 1.2 MB/s eta
0:00:20
   --------------------------- ------------ 49.3/72.0 MB 1.2 MB/s eta
0:00:19
   --------------------------- ------------ 49.3/72.0 MB 1.2 MB/s eta
0:00:19
   --------------------------- ------------ 49.5/72.0 MB 1.2 MB/s eta
0:00:19
   --------------------------- ------------ 49.8/72.0 MB 1.2 MB/s eta
0:00:19
   --------------------------- ------------ 50.3/72.0 MB 1.2 MB/s eta
0:00:19
   ---------------------------- ----------- 50.6/72.0 MB 1.2 MB/s eta
0:00:19
   ---------------------------- ----------- 51.1/72.0 MB 1.2 MB/s eta
0:00:19
   ---------------------------- ----------- 51.6/72.0 MB 1.2 MB/s eta
0:00:18
   ---------------------------- ----------- 52.2/72.0 MB 1.2 MB/s eta
0:00:18
   ----------------------------- ---------- 53.0/72.0 MB 1.2 MB/s eta
0:00:17
   ----------------------------- ---------- 53.2/72.0 MB 1.2 MB/s eta
0:00:17
   ----------------------------- ---------- 53.2/72.0 MB 1.2 MB/s eta
0:00:17
   ----------------------------- ---------- 53.5/72.0 MB 1.1 MB/s eta
0:00:17
   ------------------------------ --------- 54.0/72.0 MB 1.1 MB/s eta
0:00:16
   ------------------------------ --------- 54.3/72.0 MB 1.1 MB/s eta
0:00:16
   ------------------------------ --------- 54.5/72.0 MB 1.1 MB/s eta
0:00:16
   ------------------------------ --------- 54.8/72.0 MB 1.1 MB/s eta
0:00:16
   ------------------------------ --------- 55.1/72.0 MB 1.1 MB/s eta
0:00:16
   ------------------------------ --------- 55.3/72.0 MB 1.1 MB/s eta
0:00:16
   ------------------------------ --------- 55.3/72.0 MB 1.1 MB/s eta
0:00:16
   ------------------------------ --------- 55.6/72.0 MB 1.1 MB/s eta
0:00:15
   ------------------------------- -------- 55.8/72.0 MB 1.1 MB/s eta
```

```
0:00:15
   ------------------------------- -------- 56.4/72.0 MB 1.1 MB/s eta
0:00:15
   ------------------------------- -------- 56.6/72.0 MB 1.1 MB/s eta
0:00:15
   ------------------------------- -------- 56.9/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 56.9/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.1/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.1/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.1/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.1/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.1/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.1/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.1/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.4/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.4/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.4/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.4/72.0 MB 1.1 MB/s eta
0:00:14
   ------------------------------- -------- 57.4/72.0 MB 1.1 MB/s eta
0:00:14
   -------------------------------- ------- 57.7/72.0 MB 1.0 MB/s eta
0:00:15
   -------------------------------- ------- 57.7/72.0 MB 1.0 MB/s eta
0:00:15
   -------------------------------- ------- 57.9/72.0 MB 999.1 kB/s
eta 0:00:15
   -------------------------------- ------- 57.9/72.0 MB 999.1 kB/s
eta 0:00:15
   -------------------------------- ------- 58.2/72.0 MB 994.1 kB/s
eta 0:00:14
   -------------------------------- ------- 58.5/72.0 MB 992.0 kB/s
eta 0:00:14
   -------------------------------- ------- 58.7/72.0 MB 983.2 kB/s
eta 0:00:14
   -------------------------------- ------- 58.7/72.0 MB 983.2 kB/s
eta 0:00:14
```

```
   -------------------------------- ------- 59.0/72.0 MB 984.3 kB/s
eta 0:00:14
   -------------------------------- ------- 59.2/72.0 MB 979.1 kB/s
eta 0:00:14
   --------------------------------- ------ 59.5/72.0 MB 980.1 kB/s
eta 0:00:13
   --------------------------------- ------ 59.8/72.0 MB 971.6 kB/s
eta 0:00:13
   --------------------------------- ------ 60.0/72.0 MB 970.0 kB/s
eta 0:00:13
   --------------------------------- ------ 60.3/72.0 MB 963.0 kB/s
eta 0:00:13
   --------------------------------- ------ 60.3/72.0 MB 963.0 kB/s
eta 0:00:13
   --------------------------------- ------ 60.6/72.0 MB 949.3 kB/s
eta 0:00:13
   --------------------------------- ------ 60.8/72.0 MB 945.9 kB/s
eta 0:00:12
   --------------------------------- ------ 61.1/72.0 MB 945.5 kB/s
eta 0:00:12
   ---------------------------------- ----- 61.3/72.0 MB 944.0 kB/s
eta 0:00:12
   ---------------------------------- ----- 61.9/72.0 MB 930.5 kB/s
eta 0:00:11
   ---------------------------------- ----- 62.1/72.0 MB 928.7 kB/s
eta 0:00:11
   ---------------------------------- ----- 62.4/72.0 MB 921.5 kB/s
eta 0:00:11
   ---------------------------------- ----- 62.4/72.0 MB 921.5 kB/s
eta 0:00:11
   ---------------------------------- ----- 62.7/72.0 MB 900.7 kB/s
eta 0:00:11
   ---------------------------------- ----- 62.7/72.0 MB 900.7 kB/s
eta 0:00:11
   ---------------------------------- ----- 62.9/72.0 MB 879.0 kB/s
eta 0:00:11
   ---------------------------------- ----- 62.9/72.0 MB 879.0 kB/s
eta 0:00:11
   ----------------------------------- ---- 63.2/72.0 MB 858.0 kB/s
eta 0:00:11
   ----------------------------------- ---- 63.4/72.0 MB 854.8 kB/s
eta 0:00:10
   ----------------------------------- ---- 63.4/72.0 MB 854.8 kB/s
eta 0:00:10
   ----------------------------------- ---- 63.7/72.0 MB 844.1 kB/s
eta 0:00:10
   ----------------------------------- ---- 64.0/72.0 MB 841.2 kB/s
eta 0:00:10
   ----------------------------------- ---- 64.2/72.0 MB 843.0 kB/s
eta 0:00:10
```

```
   ----------------------------------- ---- 64.5/72.0 MB 843.4 kB/s
eta 0:00:09
   ----------------------------------- ---- 64.7/72.0 MB 845.1 kB/s
eta 0:00:09
   ------------------------------------ --- 65.3/72.0 MB 848.0 kB/s
eta 0:00:08
   ------------------------------------ --- 65.5/72.0 MB 849.9 kB/s
eta 0:00:08
   ------------------------------------ --- 65.8/72.0 MB 856.5 kB/s
eta 0:00:08
   ------------------------------------ --- 66.1/72.0 MB 859.8 kB/s
eta 0:00:07
   ------------------------------------- -- 66.6/72.0 MB 848.5 kB/s
eta 0:00:07
   ------------------------------------- -- 66.8/72.0 MB 839.4 kB/s
eta 0:00:07
   ------------------------------------- -- 67.1/72.0 MB 832.9 kB/s
eta 0:00:06
   ------------------------------------- -- 67.4/72.0 MB 832.2 kB/s
eta 0:00:06
   ------------------------------------- -- 67.6/72.0 MB 822.2 kB/s
eta 0:00:06
   ------------------------------------- -- 67.9/72.0 MB 821.4 kB/s
eta 0:00:05
   -------------------------------------- - 68.4/72.0 MB 805.8 kB/s
eta 0:00:05
   -------------------------------------- - 68.4/72.0 MB 805.8 kB/s
eta 0:00:05
   -------------------------------------- - 68.7/72.0 MB 789.7 kB/s
eta 0:00:05
   -------------------------------------- - 69.2/72.0 MB 796.3 kB/s
eta 0:00:04
   -------------------------------------- - 69.5/72.0 MB 798.7 kB/s
eta 0:00:04
   -------------------------------------- - 69.7/72.0 MB 791.8 kB/s
eta 0:00:03
   -------------------------------------- - 70.0/72.0 MB 795.5 kB/s
eta 0:00:03
   ---------------------------------------  70.3/72.0 MB 796.8 kB/s
eta 0:00:03
   ---------------------------------------  70.5/72.0 MB 797.5 kB/s
eta 0:00:02
   ---------------------------------------  70.5/72.0 MB 797.5 kB/s
eta 0:00:02
   ---------------------------------------  70.8/72.0 MB 795.9 kB/s
eta 0:00:02
   ---------------------------------------  70.8/72.0 MB 795.9 kB/s
eta 0:00:02
   ---------------------------------------  71.0/72.0 MB 790.3 kB/s
```

```
eta 0:00:02
   ---------------------------------------  71.3/72.0 MB 789.8 kB/s
eta 0:00:01
   ---------------------------------------  71.3/72.0 MB 789.8 kB/s
eta 0:00:01
   ---------------------------------------  71.6/72.0 MB 789.1 kB/s
eta 0:00:01
   ---------------------------------------  71.6/72.0 MB 789.1 kB/s
eta 0:00:01
   ---------------------------------------  71.8/72.0 MB 789.9 kB/s
eta 0:00:01
   ---------------------------------------  71.8/72.0 MB 789.9 kB/s
eta 0:00:01
   ---------------------------------------  71.8/72.0 MB 789.9 kB/s
eta 0:00:01
   ---------------------------------------  71.8/72.0 MB 789.9 kB/s
eta 0:00:01
   ---------------------------------------  71.8/72.0 MB 789.9 kB/s
eta 0:00:01
   ---------------------------------------  71.8/72.0 MB 789.9 kB/s
eta 0:00:01
   ---------------------------------------- 72.0/72.0 MB 759.4 kB/s
eta 0:00:00
Installing collected packages: xgBoost
Successfully installed xgBoost-3.1.2
```

```
import xgboost as xgb

#create and train the XGBOOST model
model_xgb=xgb.XGBRegressor()
model_xgb.fit(x_train,y_train)

#Make Predictions
y_pred_xgb=model_xgb.predict(x_test)

#calculate performance metrics
mae_xgb=mean_absolute_error(y_test,y_pred_xgb)
mse_xgb=mean_squared_error(y_test,y_pred_xgb)
rmse_xgb=np.sqrt(mse_xgb)
mape_xgb=np.mean(np.abs((y_test-y_pred_xgb)/np.abs(y_test)))
accuracy_xgb=100*(1-mape_xgb)

#print the performance metrics for XGBOOST
print('XGBoost metrics:')
print('Mean Absolute Error (MAE):',mae_xgb)
print('Mean Squared Error(MSE):',xgb)
print('Root Mean Squared Error (RMSE):',rmse_xgb)
print('Mean Absolute Percentage Error (MAPE):',round(mape_xgb*100,2))
print('Accuracy:',round(accuracy_xgb,2))
```

```
XGBoost metrics:
Mean Absolute Error (MAE): 129.57372342362487
Mean Squared Error(MSE): <module 'xgboost' from 'C:\\Users\\ADITI
DUNGYAN\\AppData\\Roaming\\Python\\Python313\\site-packages\\xgboost\\
__init__.py'>
Root Mean Squared Error (RMSE): 375.74948736971436
Mean Absolute Percentage Error (MAPE): 0.32
Accuracy: 99.68

 #cross validation for our model
from sklearn.model_selection import ShuffleSplit, cross_val_score
model=LinearRegression()
ssplit=ShuffleSplit(n_splits=10,test_size=0.30)
from sklearn.model_selection import cross_val_score
results=cross_val_score(model,x,y,cv=ssplit)
print(results)
print("\nMean Cross-validation Accuracy:",np.mean(results))

[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

Mean Cross-validation Accuracy: 1.0

# cross validation for our model
from sklearn.model_selection import ShuffleSplit, cross_val_score
model=DecisionTreeRegressor()
ssplit=ShuffleSplit(n_splits=10,test_size=0.30)
from sklearn.model_selection import cross_val_score
results=cross_val_score(model,x,y,cv=ssplit)
print(results)
print("\nMean Cross-validation Accuracy:",np.mean(results))

[0.99992803 0.999945   0.99997681 0.99959326 0.99996683 0.99987803
 0.999966   0.99996209 0.99995568 0.99993745]

Mean Cross-validation Accuracy: 0.9999109159881971

# Cross validation for our model
model=RandomForestRegressor(random_state=5)
ssplit=ShuffleSplit(n_splits=10,test_size=0.30)
from sklearn.model_selection import cross_val_score
results=cross_val_score(model,x,y,cv=ssplit)
print(results)
print("\nMean Cross-validation Accuracy:",np.mean(results))

[0.99997922 0.99999107 0.99989371 0.99998163 0.99992473 0.99982829
 0.99998882 0.9998989  0.99998798 0.9996799 ]

Mean Cross-validation Accuracy: 0.9999154243790827

model_xgb = xgb.XGBRegressor(objective='reg:squarederror',
random_state=42)
```

```
# Define ShuffleSplit cross-validation
ssplit = ShuffleSplit(n_splits=10, test_size=0.3)

# Perform cross-validation
results_xgb = cross_val_score(model_xgb, x, y, cv=ssplit)

# Print results for each fold
print("Cross-validation results for each fold:")
print(results_xgb)

# Print mean accuracy across all folds
print("\nMean Cross-validation Accuracy:", np.mean(results_xgb))
```

```
Cross-validation results for each fold:
[0.99981973 0.99956774 0.99898209 0.99958634 0.99947931 0.99989019
 0.99985813 0.99884491 0.99939327 0.99770881]

Mean Cross-validation Accuracy: 0.9993130515881153
```

```
import pickle
from xgboost import XGBRegressor

# Example: Train a model (XGBoost Regressor as an example)
model_xgb = XGBRegressor()
model_xgb.fit(x_train, y_train)

# Save the trained model to a file using pickle
with open('hr.pkl', 'wb') as file:
    pickle.dump(model_xgb, file)

print("Model saved successfully.")
```

```
Model saved successfully.
```

```
with open('schr.pkl', 'wb') as scaler_file:
    pickle.dump(sc, scaler_file)

#app

#streamlit run hrapp.py

# Define the content of the updated Streamlit script
streamlit_code = """
import streamlit as st
import pickle
import numpy as np

# Load the trained XGBoost model and scaler
with open('hr.pkl', 'rb') as model_file:
    model = pickle.load(model_file)
```

```
with open('schr.pkl', 'rb') as scaler_file:
    scaler = pickle.load(scaler_file)

# Create the web app
st.title('Salary Prediction App')

# Input fields
age = st.number_input('Age', min_value=0, max_value=120, value=30)
education = st.selectbox('Education Level', ['High School or less',
'Intermediate', 'Graduation', 'PG'])
experience_months = st.number_input('Months of Experience',
min_value=0, max_value=600, value=60)  # Assuming max experience is 50
years

# Convert education to numeric encoding
education_mapping = {'High School or less': 0, 'Intermediate': 1,
'Graduation': 2, 'PG': 3}
education_encoded = education_mapping[education]

# Prepare the feature vector
features = np.array([[age, education_encoded, experience_months]],
dtype=np.float64)

# Scale the features
features_scaled = scaler.transform(features)

# Predict salary
predicted_salary = model.predict(features_scaled)

# Display the result
st.write(f"Predicted Salary: ${predicted_salary[0]:,.2f}")
"""

# Specify the file path where the hrapp.py file will be saved
file_path = 'hrapptry.py'

# Write the content to the file
with open(file_path, 'w') as file:
    file.write(streamlit_code)

print(f"File '{file_path}' has been saved.")
```

```
File 'hrapptry.py' has been saved.
```

```
# Results:

Data analysis and interpretations

1.Data set was read.

2. Head and tail of the data set was visualized.
```

```
3. Data basic information was seen like mean,max,etc.

4. Shape of the data is 5000 row and 20 column.

5. Sum of null values were seen and null values were present in the
data set.

6. null values were successfully replaced with the help of KNN
imputer.

7. Total_sales has some empty spaces which was replaced with nan then
replaced by mean value.

8. various category vs there count were plotted to understand the
data.

9. Gender- There is not much difference between male and femlae
employee.

10.Companies has hired large number of employee which can be seen in
month column.

11.Most of the employee are seniors with mean age of 51.

12.Few data normalization checked,and they were not normal.It means
they have outliers.

13 Spearman co-relation was obtained.Spearmen correlation between
salary and Total_sales is 0.99.

13.a. It means it is Perfectly monotonically increasing relationship
and salary linearly increases with Total_sales.

13.b. We can depict that employee who will do high Total_sales will
get higher salary.Similary all features can be related in spearman
table.
```

```
14 Cateograical and dependent variable analyzed.


15. Both Male and female working in almost equal number in the
company.


16.We can depicts from this graph that both female and male drawing
almost equal salary(total number wise) from company.


17. Mean of the both person who has business or not is almost same.


18.Large number of employee have authority to call.


19. We can observe that employee with PG degree draws salary
approximately ranges from almost 50000 to 2 lac.
Employee with Graduation is drawing salary from approximately 25000 to
1.25 lac.
Salary of High School or less  and Intermediate is drawing salary less
than 25000.

20.We can observe that high number of unit sales ,higher will be the
salary.


21.We can see that higher number of sales,higher is the salary.


22.we can observe that company has hired high number of new employees.
In last four months company has hired.

almost 765 employees. Also company has 269 employees who are giving
there services from 72 months.


23. by scatter plot we can see the relation between numerical data and
salary.

24 In Data cleaning all nan values were already replaced with mean.



Data Cleaning and justification

25. Outliers were checked for all numberical data and it was found
that Many features has large number of outlers.

26. Outliers were removed using IQR method .
```

```
27. varification done weather outliers were removed by re-plotting the
box-plot and it was found that outliers were successfully removed.


 Feature Engineering

28. Data set has cateograical data which have to change ino numberical
data for machine learning.

29. Feature selection is a technique where we choose those features in
our data that contribute most to the target variable.

30. Here we can see that in scores business, calls, type, billing,
rating,gender,openingbalance,closing balance has less contribution to
target variable.
so we can drop them.
31. Using label-encoding all cateograical data was changed to
numerical data.

32. spearman Co-relation method was used to check co-relation between
two feautres.

33. Highly co-related datas have to be removed to make the model
stable.

34. Thresold was kept 0.80 and it was found that
Basic_pay,Bonus,Total_Sales and Unit_sales has the co-relation above
0.8.


 Model Building

35. Cateogrical data like
calls,rating,dependancies,billing,type,Business were dropped.

36. Highly co-related data were dropped.

37. Model were split into x_train,y_train,x_test,y_test.

38. Normalization using standardscalar done to scale down our target
vairable.


 Machine learning techniques

39.  ### Linear regression accuracy:-78.27


40. #### Decission Tree Regression :- 99.35
```

```
41. Accuracy came good but we should try  another techniques to get
better accuracy.

42. ### Random Forest Regression algorithm were used in the model
now.99.09

43. #### accuracy comes around

43 b. Xgboost:-99.24

44. Desired accuracy were achieved. And we can say that our model is
working correct.

45. ### Linear regression were tried and accuracy we got less
accuracy.


Step 8 :- Cross Validation

47.Cross-validation is a resampling method that uses different
portions of the data to test and  train a model on different
iterations.



Verdict :- After explotatotry analysis,data cleaning and model
building,various machine learning techniques were tried.
           In our model we got it is random forest regreesor giving
highest accuracy as compared to rest of all the models.

### Recommendations :

1. From the analysis done, it was observed that employee with high
experience giving high sales to the company.
   So company should hire experienced candidate more.

2. Employee performance was dominated by age,months(services to the
company), education.

3. So strategic approach to the efficient management and maximum
business gain to the  company can be
done by  hiring  the employee  who is experienced and has high
education like PG.

4. Salary component mostly affected by the age,months and
education.Also company total_sales are direcly related to salary.

5. So hr department can plan their approach by keeping above facts in
mind for hiring the candidate who can help them to get maximum
sales,which can benift the company most.
```