DA_assignment3

Group member: Huan Chen(huc48), Jiayu Chen(jic117), Chang Tian(cht97)

1.Import data from csv file into Pandas Dataframe

```
retention = pd.read_csv('Retention.csv')
retention.head()
```

	spend	apret	top10	rejr	tstsc	pacc	strat	salar
0	9855	52.50	15	29.474	65.063	36.887	12.0	60800
1	10527	64.25	36	22.309	71.063	30.970	12.8	63900
2	7904	37.75	26	25.853	60.750	41.985	20.3	57800
3	6601	57.00	23	11.296	67.188	40.289	17.0	51200
4	7251	62.00	17	22.635	56.250	46.780	18.1	48000

(a) generate descriptive statistics and plot histograms for the following three columns: apret, tstsc, and salar.

1). generate descriptive statistics:

```
print('The descriptive statistics of retention')
retention.describe(include='all')
```

```
The descriptive statistics of retention
```

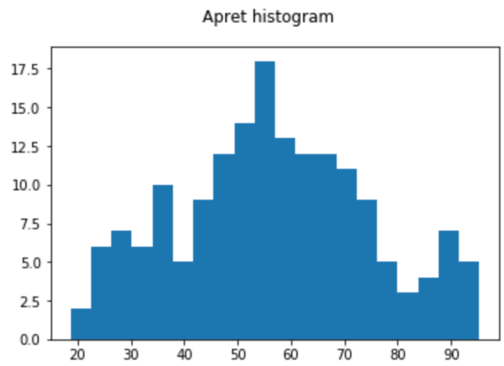
	spend	apret	top10	rejr	tstsc	pacc	strat	salar
count	170.000000	170.000000	170.000000	170.000000	170.000000	170.000000	170.000000	170.000000
mean	10974.511765	56.721076	38.458824	30.654218	66.164165	43.173106	16.086471	61357.647059
std	5500.065580	18.077097	23.406393	17.098104	6.975306	13.105195	4.006503	9802.786457
min	4125.000000	18.750000	8.000000	0.000000	48.125000	8.964000	7.200000	38640.000000
25%	7371.750000	45.374750	22.000000	19.171000	61.111000	33.903750	13.400000	54650.000000
50%	9265.000000	55.708500	30.000000	27.390500	64.781500	40.850500	16.000000	61150.000000
75%	12838.000000	68.687500	49.500000	36.807500	70.453250	51.773250	18.575000	67100.000000
max	35863.000000	95.250000	98.000000	84.067000	87.500000	76.253000	29.200000	87900.000000

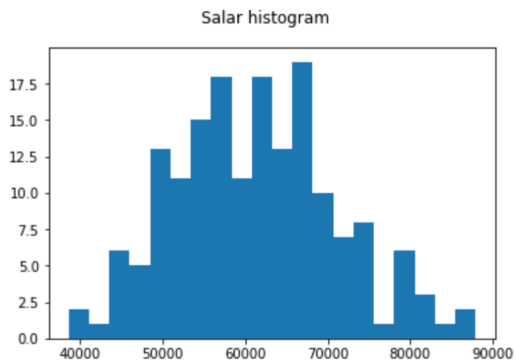
2). Plot histograms for apret, tstsc and salar

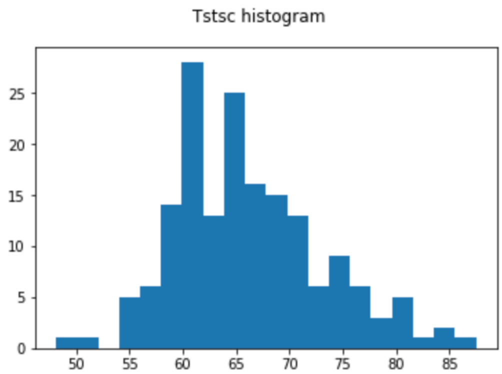
```
print('Plot histograms for the following three columns')
plt.hist(retention.apret, 20)
plt.suptitle('Apret histogram')
plt.show()

plt.hist(retention.salar, 20)
plt.suptitle('Salar histogram')
plt.show()

plt.hist(retention.tstsc, 20)
plt.suptitle('Tstsc histogram')
plt.show()
```

DA_assignment3

Group member: Huan Chen(huc48), Jiayu Chen(jic117), Chang Tian(cht97)







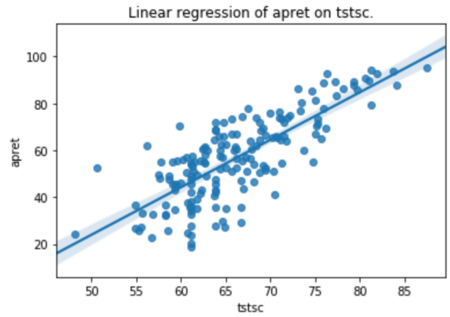
(b) perform linear regression of apret on tstsc and salar separately and then of apret on both tstsc and salar.

1). linear regression of apret on tstsc

DA_assignment3

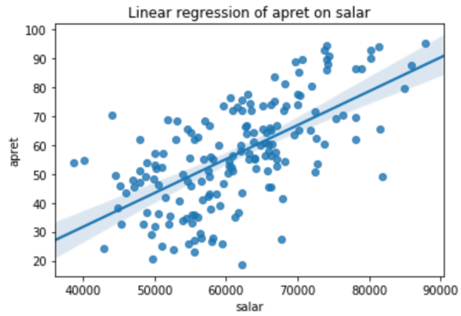
Group member: Huan Chen(huc48), Jiayu Chen(jic117), Chang Tian(cht97)

```
sns.regplot(x='tstsc', y='apret', data=retention[['apret','tstsc']])
result = plt.title('Linear regression of apret on tstsc.')
```



2). linear regression of apret on salar

```
sns.regplot(x='salar', y='apret', data=retention[['apret', 'salar']])
result = plt.title('Linear regression of apret on salar')
```



3). linear regression of apret on both tstsc and salar.

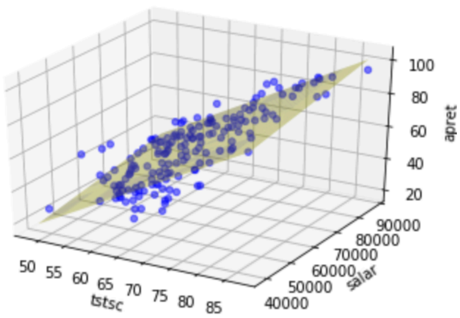
```
x1=pd.Series(retention['tstsc'])
x2=pd.Series(retention['salar'])
X=pd.concat([x1,x2],axis=1)
df2=pd.DataFrame(X,columns=['tstsc','salar'])

y=retention['apret']
df2['apret']=pd.Series(y)

model=smf.ols(formula='apret ~ tstsc + salar',data=df2)
results_formula=model.fit()
print(results_formula.summary())
```

DA_assignment3

Group member: Huan Chen(huc48), Jiayu Chen(jic117), Chang Tian(cht97)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  apret   R-squared:                       0.624
Model:                            OLS   Adj. R-squared:                  0.619
Method:                 Least Squares   F-statistic:                     138.4
Date:                Mon, 12 Feb 2018   Prob (F-statistic):           3.60e-36
Time:                        15:10:26   Log-Likelihood:                -649.73
No. Observations:                 170   AIC:                             1305.
Df Residuals:                     167   BIC:                             1315.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -75.9111      8.210     -9.246      0.000     -92.119     -59.703
tstsc          1.7375      0.176      9.868      0.000       1.390       2.085
salar          0.0003      0.000      2.298      0.023    4.06e-05       0.001
==============================================================================
Omnibus:                        1.657   Durbin-Watson:                   1.703
Prob(Omnibus):                  0.437   Jarque-Bera (JB):                1.589
Skew:                          -0.235   Prob(JB):                        0.452
Kurtosis:                       2.938   Cond. No.                     5.96e+05
==============================================================================
```

```
x_surf,y_surf =np.meshgrid(np.linspace(df2.tstsc.min(),df2.tstsc.max(),10),
                           np.linspace(df2.salar.min(),df2.salar.max(),1500))
onlyX= pd.DataFrame({'tstsc': x_surf.ravel(),'salar':y_surf.ravel()})
fittedY=results_formula.predict(exog=onlyX)

fig=plt.figure()
ax=fig.add_subplot(111,projection='3d')
ax.scatter(df2['tstsc'],df2['salar'],df2['apret'],c='blue',marker='o',alpha=0.5)
ax.plot_surface(x_surf,y_surf,fittedY.values.reshape(x_surf.shape),color='y',alpha=0.4)
ax.set_xlabel('tstsc')
ax.set_ylabel('salar')
ax.set_zlabel('apret')
plt.show()
```



(c) Test normality of each column

H0: make an assumption that all columns in retention table obey normal distribution.

alpha level is .05 for a one-tail test.

DA_assignment3

Group member: Huan Chen(huc48), Jiayu Chen(jic117), Chang Tian(cht97)

If p-value > alpha, we should accept H0 and make a conclusion that the column obey normal distribution.

If p-value <= alpha, we should reject H0 and make a conclusion that the column doesn't obey normal distribution.

```
import scipy.stats as stats
def check_norm(data, colum_name):
    alpha=.05
    p_value = stats.shapiro(data)[1]
    if p_value>alpha:
        print(colum_name, "obey normal distribution")
    else:
        print(colum_name, "does not obey normal distribution")

check_norm(retention.spend, 'spend')
check_norm(retention.apret, 'apret')
check_norm(retention.top10, 'top10')
check_norm(retention.rejr, 'rejr')
check_norm(retention.tstsc, 'tstsc')
check_norm(retention.pacc, 'pacc')
check_norm(retention.strat, 'strat')
check_norm(retention.salar, 'salar')
```

```
spend does not obey normal distribution
apret obey normal distribution
top10 does not obey normal distribution
rejr does not obey normal distribution
tstsc does not obey normal distribution
pacc does not obey normal distribution
strat obey normal distribution
salar obey normal distribution
```

d. Correlation & Coefficient

```
print('Correlation matrix using pearson method')
retention.corr(method='pearson')
```

```
Correlation matrix using pearson method
```

	spend	apret	top10	rejr	tstsc	pacc	strat	salar
spend	1.000000	0.601231	0.675656	0.633544	0.714910	-0.236730	-0.561755	0.711838
apret	0.601231	1.000000	0.642464	0.514958	0.782183	-0.302834	-0.458311	0.635852
top10	0.675656	0.642464	1.000000	0.643163	0.798807	-0.207505	-0.247857	0.637648
rejr	0.633544	0.514958	0.643163	1.000000	0.628601	-0.071521	-0.283617	0.606777
tstsc	0.714910	0.782183	0.798807	0.628601	1.000000	-0.164223	-0.465226	0.715472
pacc	-0.236730	-0.302834	-0.207505	-0.071521	-0.164223	1.000000	0.131858	-0.375240
strat	-0.561755	-0.458311	-0.247857	-0.283617	-0.465226	0.131858	1.000000	-0.347673
salar	0.711838	0.635852	0.637648	0.606777	0.715472	-0.375240	-0.347673	1.000000

DA_assignment3

Group member: Huan Chen(huc48), Jiayu Chen(jic117), Chang Tian(cht97)

As can be seen in table above, the correlation of salar and tstsc is close.

Then we want to find the coeffecient of these two variables.

Find the P-value of the TV feature and the 95% confidence interval of the corresponding coefficients.

```
import statsmodels.formula.api as smf
model=smf.ols('tstsc ~ salar', retention)
Fitting_results=model.fit()
print(Fitting_results.summary().tables[1])
print('p-values are:\n',Fitting_results.pvalues)
```

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     34.9268      2.383     14.657      0.000      30.222      39.631
salar          0.0005   3.84e-05     13.274      0.000       0.000       0.001
==============================================================================
p-values are:
 Intercept    7.358734e-32
salar        5.840126e-28
dtype: float64
```