四. 多重线性回归

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
```

1. 读取数据

表4.5 2000-2016年财政收入及其影响因素数据

一般公共预算收入CZSR ｜国内生产总值GDP ｜ 税收总额SSZE｜ 工业增加值GYZJZ

```
data = pd.read_csv('chapter4.csv', encoding =  'UTF-8')
data
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	year	Y	X1	X2	X3
0	2000.0	13395.23	100280.1	12581.51	40259.7
1	2001.0	16386.04	110863.1	15301.38	43855.6
2	2002.0	18903.64	121717.4	17636.45	47776.6
3	2003.0	21715.25	137422.0	20017.31	55363.8
4	2004.0	26396.47	161840.2	24165.68	65776.8
5	2005.0	31649.29	187318.9	28778.54	77960.5
6	2006.0	38760.20	219438.5	34804.35	92238.4
7	2007.0	51321.78	270232.3	45621.97	111693.9
8	2008.0	61330.35	319515.5	54223.79	131727.6
9	2009.0	68518.30	349081.4	59521.59	138095.5
10	2010.0	83101.51	413030.3	73210.79	165126.4
11	2011.0	103874.43	489300.6	89738.39	195142.8
12	2012.0	117253.00	540367.4	100614.28	208905.6
13	2013.0	129209.64	595244.4	110530.70	222337.6
14	2014.0	140370.03	643974.0	119175.31	233856.4
15	2015.0	152269.23	689052.1	124922.20	236506.3
16	2016.0	159604.97	744127.2	130360.73	247860.1

提取因变量和自变量

```
X = data.iloc[:, 1:4]
Y = data.iloc[:, 1]
```

2. 判断多重共线性

ols法估计，R^2值高、F检验值高、且x1,x2，x3的t检验不显著

```
X1 = sm.add_constant(X) #加上一列常数1，这是回归模型中的常数项
reg = sm.OLS(Y, X1) #生成回归模型
model = reg.fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	Y	**R-squared:**	1.000
Model:	OLS	**Adj. R-squared:**	1.000
Method:	Least Squares	**F-statistic:**	1.456e+29
Date:	Sat, 18 Dec 2021	**Prob (F-statistic):**	1.15e-185
Time:	00:18:00	**Log-Likelihood:**	350.26
No. Observations:	17	**AIC:**	-692.5
Df Residuals:	13	**BIC:**	-689.2
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-1.455e-11	5.4e-10	-0.027	0.979	-1.18e-09	1.15e-09
Y	1.0000	5.99e-14	1.67e+13	0.000	1.000	1.000
X1	0	1.06e-14	0	1.000	-2.29e-14	2.29e-14
X2	1.776e-15	4.97e-14	0.036	0.972	-1.06e-13	1.09e-13

Omnibus:	3.722	**Durbin-Watson:**	0.013
Prob(Omnibus):	0.156	**Jarque-Bera (JB):**	1.719
Skew:	0.452	**Prob(JB):**	0.423
Kurtosis:	1.731	**Cond. No.**	3.09e+06

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.09e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

相关系数，对数据进行标准化处理（z-score标准化），可见有共线性

```
X = (X - X.mean())/np.std(X)
Y = (Y - Y.mean())/np.std(Y)
X.corr()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Y	X1	X2
Y	1.000000	0.999445	0.999310
X1	0.999445	1.000000	0.998801
X2	0.999310	0.998801	1.000000

分割数据

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,train_size=0.7, random_state=1)
```

3. 消除多重共线性（PCA法）

对模型进行训练，返回降维后数据

```
pca = PCA(n_components='mle')
pca.fit(X_train)
X_train = pca.transform(X_train)
Y_train= (Y_train - Y_train.mean())/np.std(Y)
X_train
```

```
array([[-1.88902515],
       [-2.26694608],
       [ 2.06530374],
       [-2.36830047],
       [ 2.71543885],
       [ 2.40543657],
       [-0.39972553],
       [-0.63710762],
       [ 1.25748604],
       [ 0.81213111],
       [-1.69469147]])
```

4. 重建线性回归

使用返回后的数据用线性回归模型建模，ols回归后R^2为0.933，p值小，说明模型拟合效果好

```
import statsmodels.api as sm
ols = sm.OLS(Y_train, X_train).fit()
ols.summary()
```

OLS Regression Results

Dep. Variable:	Y	**R-squared (uncentered):**	1.000
Model:	OLS	**Adj. R-squared (uncentered):**	1.000
Method:	Least Squares	**F-statistic:**	4.111e+04
Date:	Sat, 18 Dec 2021	**Prob (F-statistic):**	2.09e-19
Time:	00:18:29	**Log-Likelihood:**	29.443
No. Observations:	11	**AIC:**	-56.89
Df Residuals:	10	**BIC:**	-56.49
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.5787	0.003	202.762	0.000	0.572	0.585

Omnibus:	0.907	**Durbin-Watson:**	2.121
Prob(Omnibus):	0.636	**Jarque-Bera (JB):**	0.658
Skew:	0.522	**Prob(JB):**	0.720
Kurtosis:	2.412	**Cond. No.**	1.00

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
pca.explained_variance_ratio_
```

```
array([0.99941241])
```

```
pca.get_params()
```

```
{'copy': True,
 'iterated_power': 'auto',
 'n_components': 'mle',
 'random_state': None,
 'svd_solver': 'auto',
 'tol': 0.0,
 'whiten': False}
```

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train,Y_train)
lr.score(X_train, Y_train)
# X_test = data.iloc[64:, 0:4]
# y_test = data.iloc[64:, 5]
```

```
0.9997568238590593
```

5. 测试集验证

```
X_test = (X_test - X_test.mean())/np.std(X_test)
X_test = pca.transform(X_test)
X_test
```

```
array([[-1.89266847],
       [ 2.90668477],
       [-0.5355235 ],
       [-2.03143245],
       [-1.09116977],
       [ 0.92391669]])
```

预测值

```
y_pred = lr.predict(X_test)
y_pred
```

```
array([-1.09524387,  1.68203187, -0.30989518, -1.17554341, -0.63143494,
        0.53464942])
```

真实值

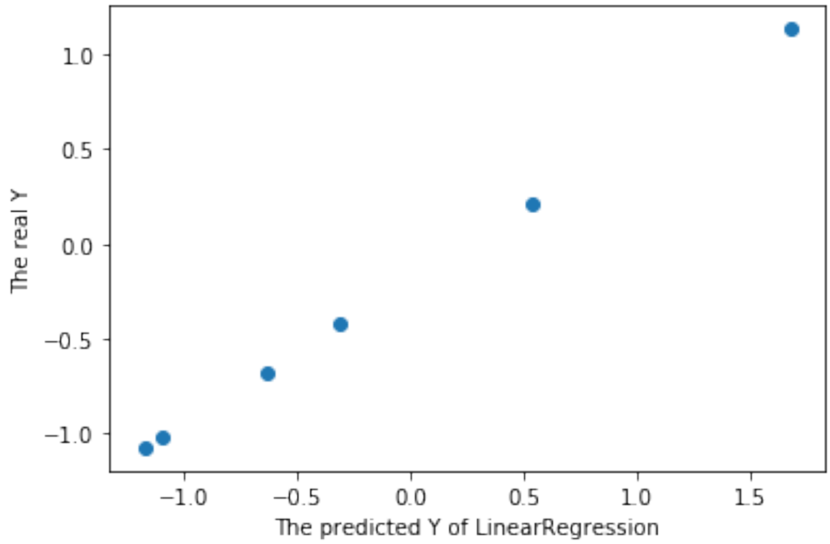
```
Y_test
```

```
3    -1.017676
13    1.132521
7    -0.425460
2    -1.073916
6    -0.676728
10    0.210226
Name: Y, dtype: float64
```

比较真实值与预测值

```
plt.scatter(y_pred, Y_test)
plt.xlabel('The predicted Y of LinearRegression')
plt.ylabel('The real Y')
```

```
Text(0, 0.5, 'The real Y')
```




```
olsr = sm.OLS(y_pred, Y_test).fit()
olsr.summary()
```

```
/Users/zcl271828/opt/anaconda3/lib/python3.7/site-
packages/statsmodels/stats/stattools.py:71: ValueWarning: omni_normtest is not valid
with less than 8 observations; 6 samples were given.
  "samples were given." % int(n), ValueWarning)
```

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	0.954
Model:	OLS	**Adj. R-squared (uncentered):**	0.945
Method:	Least Squares	**F-statistic:**	103.5
Date:	Sat, 18 Dec 2021	**Prob (F-statistic):**	0.000158
Time:	00:18:51	**Log-Likelihood:**	0.62337
No. Observations:	6	**AIC:**	0.7533
Df Residuals:	5	**BIC:**	0.5450
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Y	1.1923	0.117	10.172	0.000	0.891	1.494

Omnibus:	nan	**Durbin-Watson:**	0.312
Prob(Omnibus):	nan	**Jarque-Bera (JB):**	0.561
Skew:	0.363	**Prob(JB):**	0.756
Kurtosis:	1.691	**Cond. No.**	1.00

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

R^2值为0.954，说明在测试集上回归非常完备，也说明PCA方法较好地消除了多重共线性