Introduction to Big Data

Lecture 10. Regression

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import seaborn as sns
```

Regression with Boston Housing Data

CRIM - per capita crime rate by town

ZN - proportion of residential land zoned for lots over 25,000 sq.ft.

INDUS - proportion of non-retail business acres per town.

CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)

NOX - nitric oxides concentration (parts per 10 million)

RM - average number of rooms per dwelling

AGE - proportion of owner-occupied units built prior to 1940

DIS - weighted distances to five Boston employment centres

RAD - index of accessibility to radial highways

TAX - full-value property-tax rate per $10,000

PTRATIO - pupil-teacher ratio by town

B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town

LSTAT - % lower status of the population MEDV - Median value of owner-occupied homes in $1000's

```
from statsmodels.formula.api import ols
```

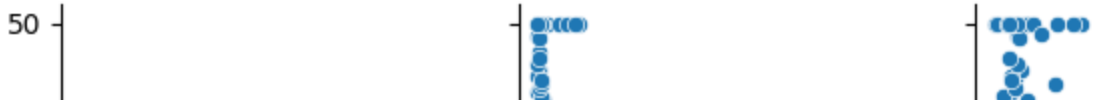
```
housing_df = pd.read_csv('/content/drive/MyDrive/[Lecture]/IntBigData/BigData_Python/10_Regress
housing_df
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	NaN	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	NaN	22.4
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0.0	0.573	6.030	NaN	2.5050	1	273	21.0	396.90	7.88	11.9

506 rows × 14 columns

```
sns.pairplot(housing_df[['MEDV','CRIM','LSTAT']])
```

```
<seaborn.axisgrid.PairGrid at 0x7c40cd3b3a60>
```



Simple Regression

```
statsOLSModel = ols('MEDV ~ CRIM', data=housing_df)
statsOLSModel
```

```
'MEDV ~ CRIM'
```

```
statsOLSModel_res = statsOLSModel.fit()
statsOLSModel_res
```

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7c40cadc4160>
```


```
print(statsOLSModel_res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.153
Model:                            OLS   Adj. R-squared:                  0.151
Method:                 Least Squares   F-statistic:                     87.54
Date:                Sun, 26 Nov 2023   Prob (F-statistic):           3.08e-19
Time:                        10:22:59   Log-Likelihood:                -1722.2
No. Observations:                 486   AIC:                             3448.
Df Residuals:                     484   BIC:                             3457.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     23.8792      0.412     57.978      0.000      23.070      24.689
CRIM          -0.4086      0.044     -9.356      0.000      -0.494      -0.323
==============================================================================
```

```
Omnibus:                      137.385   Durbin-Watson:                   0.764
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              296.868
Skew:                           1.505   Prob(JB):                     3.44e-65
Kurtosis:                       5.367   Cond. No.                         10.2
==============================================================================

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

```
statsOLSModel_res.params
```

```
Intercept    23.879229
CRIM         -0.408635
dtype: float64
```

Checking residual

```
beta0_hat = statsOLSModel_res.params[0]
beta1_hat = statsOLSModel_res.params[1]
```

```
MEDV_hat = beta0_hat + beta1_hat * housing_df.CRIM
MEDV_hat
```

```
0      23.876647
1      23.868070
2      23.868078
3      23.866002
4      23.851013
         ...
501    23.853637
502    23.860731
503    23.854401
504    23.834447
505    23.859856
Name: CRIM, Length: 506, dtype: float64
```

```
housing_df.MEDV - MEDV_hat
```

```
0       0.123353
1      -2.268070
2      10.831922
3       9.533998
4      12.348987
         ...
501    -1.453637
502    -3.260731
503     0.045599
504    -1.834447
505   -11.959856
Length: 506, dtype: float64
```

```
statsOLSModel_res.resid
```

```
0       0.123353
1      -2.268070
2      10.831922
3       9.533998
4      12.348987
         ...
501    -1.453637
502    -3.260731
503     0.045599
504    -1.834447
505   -11.959856
Length: 486, dtype: float64
```

Checking R-squared

```
housing_df.MEDV
```

```
0      24.0
1      21.6
2      34.7
```

```
3      33.4
4      36.2
       ...
501    22.4
502    20.6
503    23.9
504    22.0
505    11.9
Name: MEDV, Length: 506, dtype: float64
```

- Version 1. Why they are different?

```
y_mu = housing_df.MEDV.mean(axis=0)
y = housing_df.MEDV
y_hat = statsOLSModel_res.predict()
```

```
# 1 - sum(statsOLSModel_res.resid**2) / sum((y-y_mu)**2)
1 - sum((y-MEDV_hat).fillna(0)**2) / sum((y-y_mu)**2)
```

```
0.20290503505097468
```

```
statsOLSModel_res.rsquared
```

```
0.15316501089896506
```

- Version 2. Fixed one

```
print(len(y), len(y_hat))
```

```
506 486
```

```
housing_df.isna().any()
```

```
CRIM       True
ZN         True
INDUS      True
CHAS       True
NOX       False
RM        False
AGE        True
DIS       False
RAD       False
TAX       False
PTRATIO   False
B         False
LSTAT      True
MEDV      False
dtype: bool
```

```
y_mu = housing_df.MEDV.mean(axis=0)
y = housing_df.MEDV[housing_df.CRIM.isna()==False]
y_hat = statsOLSModel_res.predict()
print(len(y), len(y_hat))
```

```
486 486
```

```
1 - sum(statsOLSModel_res.resid**2) / sum((y-y_mu)**2)
1 - sum((y-MEDV_hat).fillna(0)**2) / sum((y-y_mu)**2)
```

```
0.15333667384313576
```

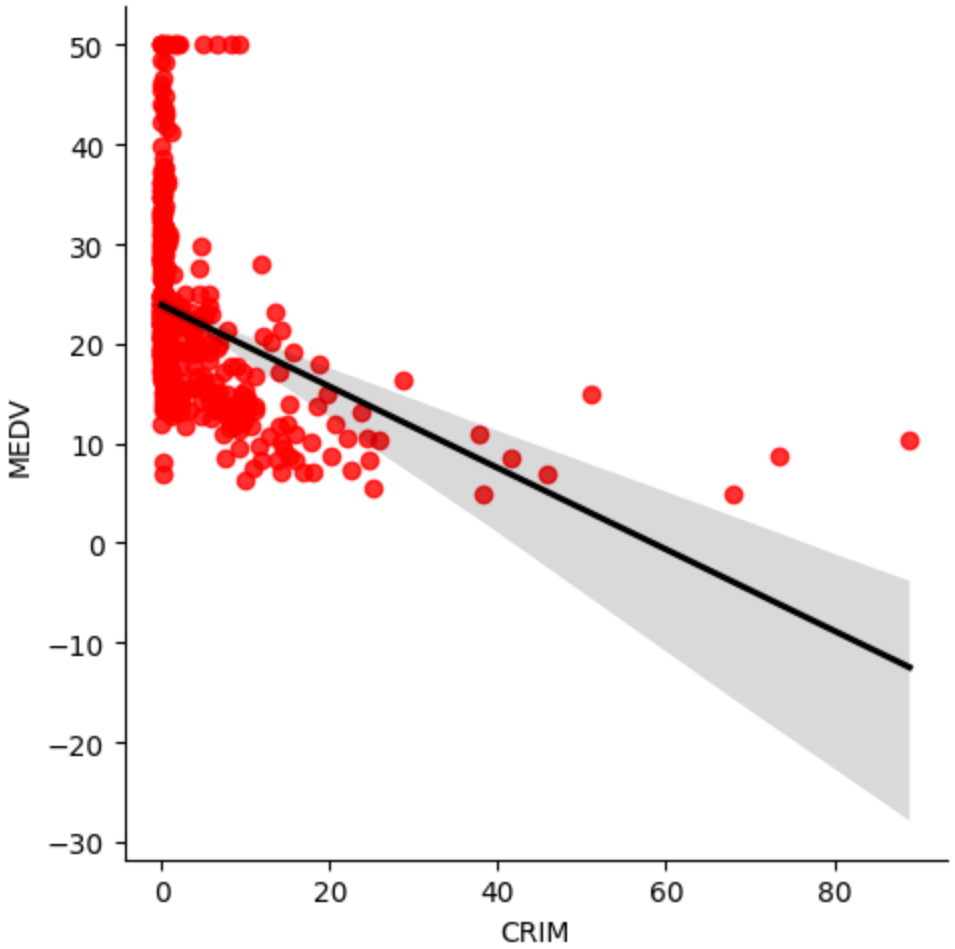
```
statsOLSModel_res.rsquared
```

```
0.15316501089896506
```

Visualization

```
sns.lmplot(x='CRIM',y='MEDV', data=housing_df,
           scatter_kws = {'color':'red'}, line_kws={'color':'black'})
```

```
<seaborn.axisgrid.FacetGrid at 0x7c40c8f46290>
```



Multiple Regression

```
statsOLSModel_all = ols('MEDV ~ CRIM+ZN+INDUS+CHAS+NOX+RM+AGE+DIS+RAD+TAX+PTRATIO+B
                        data=housing_df)
statsOLSModel_all
```

```
<statsmodels.regression.linear_model.OLS at 0x7c40c8f46170>
```

```
statsOLSModel_all_res = statsOLSModel_all.fit()
statsOLSModel_all_res
```

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7c40c8f475b0>
```

```
print(statsOLSModel_all_res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.767
Model:                            OLS   Adj. R-squared:                  0.759
Method:                 Least Squares   F-statistic:                     96.29
Date:                Sun, 26 Nov 2023   Prob (F-statistic):          1.75e-111
Time:                        11:17:16   Log-Likelihood:                -1143.4
No. Observations:                 394   AIC:                             2315.
Df Residuals:                     380   BIC:                             2370.
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     32.6801      5.681      5.752      0.000      21.509      43.851
CRIM          -0.0976      0.032     -3.007      0.003      -0.161      -0.034
ZN             0.0489      0.014      3.397      0.001       0.021       0.077
INDUS          0.0304      0.066      0.461      0.645      -0.099       0.160
CHAS           2.7694      0.925      2.993      0.003       0.950       4.588
NOX          -17.9690      4.243     -4.235      0.000     -26.311      -9.627
RM             4.2833      0.471      9.100      0.000       3.358       5.209
AGE           -0.0130      0.014     -0.898      0.370      -0.041       0.015
DIS           -1.4585      0.211     -6.912      0.000      -1.873      -1.044
RAD            0.2859      0.069      4.125      0.000       0.150       0.422
```

```
TAX          -0.0131      0.004     -3.324      0.001      -0.021      -0.005
PTRATIO      -0.9146      0.141     -6.506      0.000      -1.191      -0.638
B             0.0097      0.003      3.251      0.001       0.004       0.015
LSTAT        -0.4237      0.055     -7.700      0.000      -0.532      -0.315
==============================================================================
Omnibus:                      161.243   Durbin-Watson:                   1.247
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              904.814
Skew:                           1.657   Prob(JB):                    3.33e-197
Kurtosis:                       9.643   Cond. No.                     1.57e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.57e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
housing_df.corr()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO
CRIM	1.000000	-0.191178	0.401863	-0.054355	0.417130	-0.219150	0.354342	-0.374166	0.624765	0.580595	0.281110
ZN	-0.191178	1.000000	-0.531871	-0.037229	-0.513704	0.320800	-0.563801	0.656739	-0.310919	-0.312371	-0.414046

```
statsOLSModel_all.exog
```

```
array([[1.0000e+00, 6.3200e-03, 1.8000e+01, ..., 1.5300e+01, 3.9690e+02,
        4.9800e+00],
       [1.0000e+00, 2.7310e-02, 0.0000e+00, ..., 1.7800e+01, 3.9690e+02,
        9.1400e+00],
       [1.0000e+00, 2.7290e-02, 0.0000e+00, ..., 1.7800e+01, 3.9283e+02,
        4.0300e+00],
       ...,
       [1.0000e+00, 4.5270e-02, 0.0000e+00, ..., 2.1000e+01, 3.9690e+02,
        9.0800e+00],
       [1.0000e+00, 6.0760e-02, 0.0000e+00, ..., 2.1000e+01, 3.9690e+02,
        5.6400e+00],
       [1.0000e+00, 1.0959e-01, 0.0000e+00, ..., 2.1000e+01, 3.9345e+02,
        6.4800e+00]])
```

B	-0.381411	0.171303	-0.360532	0.051264	-0.380051	0.128069	-0.275303	0.291512	-0.444413	-0.441808	-0.177383

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
variance_inflation_factor(statsOLSModel_all.exog,1)
```

```
1.7414043701695407
```

```
lf = pd.DataFrame({'variable': column,
                   'VIF': variance_inflation_factor(statsOLSModel_all.exog, i)}
                  for i, column in enumerate(statsOLSModel_all.exog_names)
                  if column != 'Intercept')
lf
```

	variable	VIF
0	CRIM	1.741404
1	ZN	2.321843
2	INDUS	4.049690
3	CHAS	1.069182
4	NOX	4.495772
5	RM	2.107004
6	AGE	3.173844
7	DIS	3.827427
8	RAD	6.986683
9	TAX	8.651382
10	PTRATIO	1.810597
11	B	1.372310

```
all_vif_df.mean()
```

```
<ipython-input-88-53419d599982>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future ve
  all_vif_df.mean()
VIF    3.443344
dtype: float64
```

```
model_vif = ols('MEDV ~ CRIM+ZN+INDUS+CHAS+NOX+RM+AGE+DIS+PTRATIO+B+LSTAT', data=housing_df)

vif_df = pd.DataFrame({'variable': column,
                       'VIF': variance_inflation_factor(model_vif.exog, i)}
                      for i, column in enumerate(model_vif.exog_names)
                      if column != 'Intercept')
vif_df
```

	variable	VIF
0	CRIM	1.455569
1	ZN	2.155963
2	INDUS	3.090276
3	CHAS	1.048445
4	NOX	3.892387
5	RM	2.046956
6	AGE	3.126951
7	DIS	3.825071
8	PTRATIO	1.542539
9	B	1.346292
10	LSTAT	3.134480

```
vif_df.mean()
```

```
<ipython-input-90-c5cc5c826f24>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future ve
  vif_df.mean()
VIF    2.424085
dtype: float64
```

```
print(model_vif.fit().summary())
```

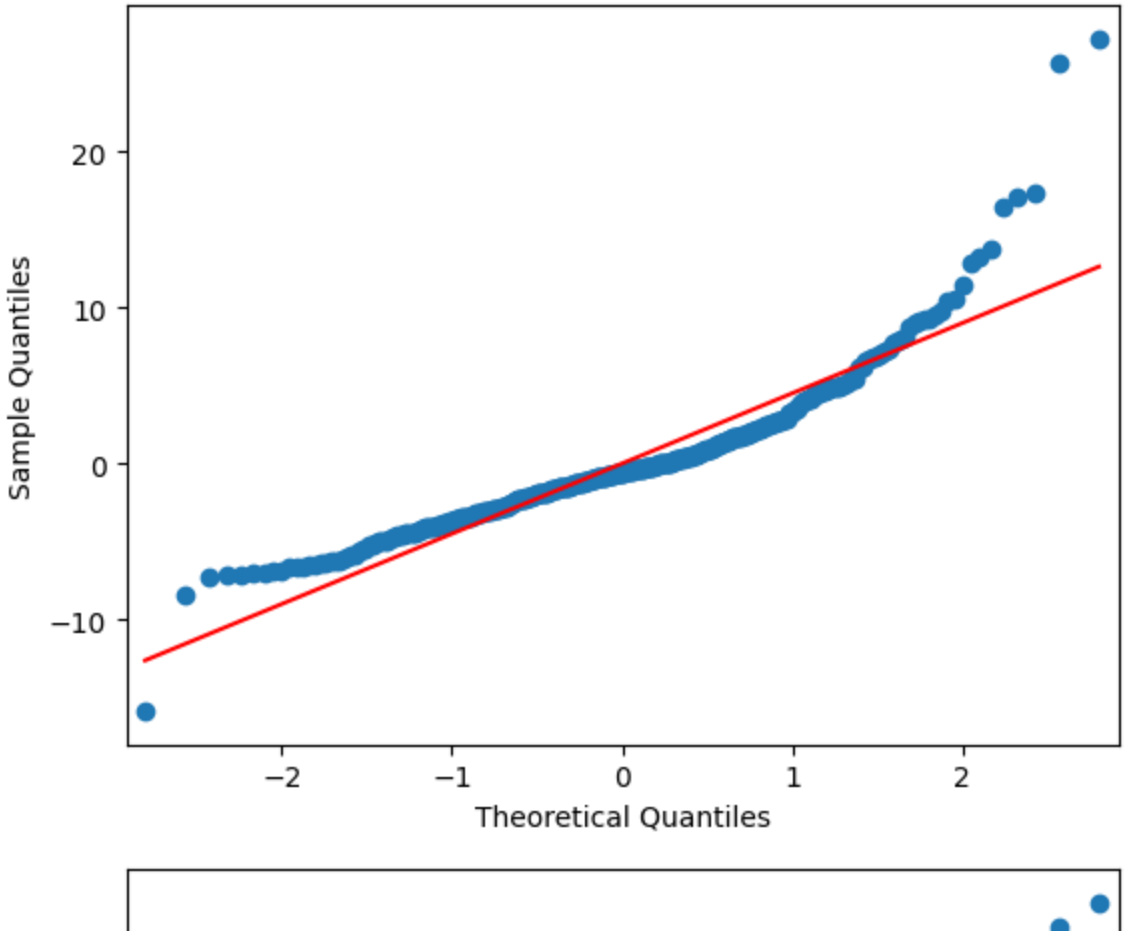
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.757
Model:                            OLS   Adj. R-squared:                  0.750
Method:                 Least Squares   F-statistic:                     108.0
Date:                Sun, 26 Nov 2023   Prob (F-statistic):          6.29e-110
Time:                        11:23:58   Log-Likelihood:                -1152.1
No. Observations:                 394   AIC:                             2328.
Df Residuals:                     382   BIC:                             2376.
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     26.0322      5.438      4.787      0.000      15.340      36.724
CRIM          -0.0626      0.030     -2.070      0.039      -0.122      -0.003
ZN             0.0415      0.014      2.937      0.004       0.014       0.069
INDUS         -0.0579      0.059     -0.986      0.325      -0.173       0.058
CHAS           3.2859      0.934      3.518      0.000       1.449       5.122
NOX          -14.5679      4.025     -3.619      0.000     -22.482      -6.654
RM             4.5579      0.473      9.636      0.000       3.628       5.488
AGE           -0.0184      0.015     -1.258      0.209      -0.047       0.010
DIS           -1.4391      0.215     -6.692      0.000      -1.862      -1.016
PTRATIO       -0.8162      0.132     -6.170      0.000      -1.076      -0.556
B              0.0088      0.003      2.935      0.004       0.003       0.015
LSTAT         -0.4190      0.056     -7.496      0.000      -0.529      -0.309
==============================================================================
Omnibus:                      165.399   Durbin-Watson:                   1.232
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              963.905
Skew:                           1.694   Prob(JB):                     4.91e-210
Kurtosis:                       9.873   Cond. No.                     9.76e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.76e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Normality issue

```
res_all_fix = ols('MEDV ~ CRIM+ZN+INDUS+CHAS+NOX+RM+AGE+DIS+PTRATIO+B+LSTAT',
                  data=housing_df).fit()
```

```
import statsmodels.api as sm
sm.qqplot(res_all_fix.resid, line = 's')
```


Sample Quantiles
20
10
0
−10
−2
−1
0
1
2
Theoretical Quantiles