Intro to MLR OLS in statmodels.api

In [1]:
```
import pandas as pd, numpy as np, statsmodels.api as sm

df=pd.read_csv("D:\\datasets\\insurance.csv")
df.head()
```

Out[1]:

	age	sex	bmi	children	smoker	region	expenses
0	19	female	27.9	0	yes	southwest	16884.92
1	18	male	33.8	1	no	southeast	1725.55
2	28	male	33.0	3	no	southeast	4449.46
3	33	male	22.7	0	no	northwest	21984.47
4	32	male	28.9	0	no	northwest	3866.86

In [2]:
```
label= 'expenses'

y= df.expenses
x= df[['age', 'bmi', 'children']].assign(const=1)

model= sm.OLS(y, x)
results= model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               expenses   R-squared:                       0.120
Model:                            OLS   Adj. R-squared:                  0.118
Method:                 Least Squares   F-statistic:                     60.74
Date:                Thu, 10 Aug 2023   Prob (F-statistic):           8.32e-37
Time:                        16:49:34   Log-Likelihood:                -14392.
No. Observations:                1338   AIC:                         2.879e+04
Df Residuals:                    1334   BIC:                         2.881e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
```

```
------------------------------------------------------------------------------
age          239.9626     22.288     10.766      0.000     196.239     283.686
bmi          332.5216     51.307      6.481      0.000     231.870     433.173
children     543.0436    258.230      2.103      0.036      36.462    1049.625
const      -6929.3145   1757.434     -3.943      0.000   -1.04e+04   -3481.678
==============================================================================
Omnibus:                      325.223   Durbin-Watson:                   2.012
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              602.850
Skew:                           1.520   Prob(JB):                    1.24e-131
Kurtosis:                       4.254   Cond. No.                        290.
==============================================================================

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

In [3]:

```
df['predictions']= results.fittedvalues
df
```

Out[3]:

	age	sex	bmi	children	smoker	region	expenses	predictions
0	19	female	27.9	0	yes	southwest	16884.92	6907.326136
1	18	male	33.8	1	no	southeast	1725.55	9172.284502
2	28	male	33.0	3	no	southeast	4449.46	12391.979997
3	33	male	22.7	0	no	northwest	21984.47	8537.689692
4	32	male	28.9	0	no	northwest	3866.86	10359.360926
...	...	...	...	...	...	...	...	...
1333	50	male	31.0	3	no	northwest	10600.55	17006.113044
1334	18	female	31.9	0	no	northeast	2205.98	7997.449896
1335	18	female	36.9	0	no	southeast	1629.83	9660.057790
1336	21	female	25.8	0	no	southwest	2007.95	6688.955930
1337	61	female	29.1	0	yes	northwest	29141.36	17384.779329

1338 rows × 8 columns

In [4]:

```
print(results.predict([33, 22.7, 0, 0]))
```

```
[15467.00414517]
```

In [5]:
```
print(results.predict([19, 27.9, 0, 1]))
```

```
[6907.32613573]
```

MLR with categorical values dummy codes

In [6]:
```
#df= pd.get_dummies(df, columns=['sex'], prefix='sex', drop_first=True)
#df= pd.get_dummies(df, columns=['smoker'], prefix='smoker', drop_first=True)
#df= pd.get_dummies(df, columns=['region'], prefix='region', drop_first=True)
#df.head()
```

In [7]:
```
for col in df:
    if not pd.api.types.is_numeric_dtype(df[col]):
        df=pd.get_dummies(df, columns=[col], prefix=col, drop_first=True)

df.head()
```

Out[7]:

	age	bmi	children	expenses	predictions	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	19	27.9	0	16884.92	6907.326136	0	1	0	0	1
1	18	33.8	1	1725.55	9172.284502	1	0	0	1	0
2	28	33.0	3	4449.46	12391.979997	1	0	0	1	0
3	33	22.7	0	21984.47	8537.689692	1	0	1	0	0
4	32	28.9	0	3866.86	10359.360926	1	0	1	0	0

In [8]:
```
x= df.drop(columns=[label]).assign(const=1)
results=sm.OLS(y, x).fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               expenses   R-squared:                       0.751
Model:                            OLS   Adj. R-squared:                  0.749
```

```
Method:                 Least Squares   F-statistic:                     500.9
Date:                Thu, 10 Aug 2023   Prob (F-statistic):               0.00
Time:                        16:49:34   Log-Likelihood:                -13548.
No. Observations:                1338   AIC:                         2.711e+04
Df Residuals:                    1329   BIC:                         2.716e+04
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
age               -154.8900     31.521     -4.914      0.000    -216.727     -93.053
bmi               -231.2527     30.546     -7.571      0.000    -291.177    -171.328
children          -456.0686    149.330     -3.054      0.002    -749.018    -163.120
predictions          1.7158      0.142     12.122      0.000       1.438       1.993
sex_male          -131.3520    332.935     -0.395      0.693    -784.488     521.784
smoker_yes        2.385e+04    413.139     57.723      0.000     2.3e+04    2.47e+04
region_northwest  -352.7901    476.261     -0.741      0.459   -1287.095     581.515
region_southeast -1035.5957    478.681     -2.163      0.031   -1974.648     -96.544
region_southwest  -959.3058    477.912     -2.007      0.045   -1896.850     -21.762
const              -52.2030     12.612     -4.139      0.000     -76.944     -27.462
==============================================================================
Omnibus:                      300.499   Durbin-Watson:                   2.088
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              719.382
Skew:                           1.212   Prob(JB):                     6.14e-157
Kurtosis:                       5.652   Cond. No.                     2.34e+19
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.75e-28. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

MLR OLS *standardization* normalization

In [9]:

```
x= df.drop(columns=[label, 'predictions']).assign(const=1)
results=sm.OLS(y, x).fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               expenses   R-squared:                       0.751
Model:                            OLS   Adj. R-squared:                  0.749
Method:                 Least Squares   F-statistic:                     500.9
```

```
Date:                Thu, 10 Aug 2023   Prob (F-statistic):               0.00
Time:                        16:49:34   Log-Likelihood:                -13548.
No. Observations:                1338   AIC:                         2.711e+04
Df Residuals:                    1329   BIC:                         2.716e+04
Df Model:                           8
Covariance Type:            nonrobust
================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
age                256.8392     11.899     21.586      0.000     233.497     280.181
bmi                339.2899     28.598     11.864      0.000     283.187     395.393
children           475.6889    137.800      3.452      0.001     205.360     746.017
sex_male          -131.3520    332.935     -0.395      0.693    -784.488     521.784
smoker_yes        2.385e+04    413.139     57.723      0.000     2.3e+04    2.47e+04
region_northwest  -352.7901    476.261     -0.741      0.459   -1287.095     581.515
region_southeast -1035.5957    478.681     -2.163      0.031   -1974.648     -96.544
region_southwest  -959.3058    477.912     -2.007      0.045   -1896.850     -21.762
const            -1.194e+04    987.811    -12.089      0.000   -1.39e+04      -1e+04
==============================================================================
Omnibus:                      300.499   Durbin-Watson:                   2.088
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              719.382
Skew:                           1.212   Prob(JB):                    6.14e-157
Kurtosis:                       5.652   Cond. No.                          311.
==============================================================================

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

In [10]:

```
from sklearn import preprocessing

df_zscore= pd.DataFrame(preprocessing.StandardScaler().fit_transform(df), columns=df.columns)
df_zscore.head()
```

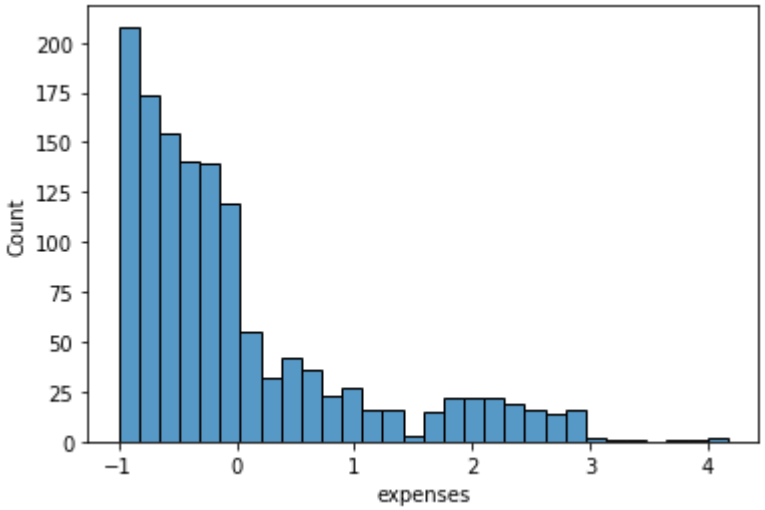
Out[10]:

	age	bmi	children	expenses	predictions	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	-1.438764	-0.453646	-0.908614	0.298583	-1.516295	-1.010519	1.970587	-0.566418	-0.611324	1.765481
1	-1.509965	0.514186	-0.078767	-0.953689	-0.976566	0.989591	-0.507463	-0.566418	1.635795	-0.566418
2	-0.797954	0.382954	1.580926	-0.728675	-0.209329	0.989591	-0.507463	-0.566418	1.635795	-0.566418
3	-0.441948	-1.306650	-0.908614	0.719843	-1.127787	0.989591	-0.507463	1.765481	-0.611324	-0.566418
4	-0.513149	-0.289606	-0.908614	-0.776802	-0.693692	0.989591	-0.507463	1.765481	-0.611324	-0.566418

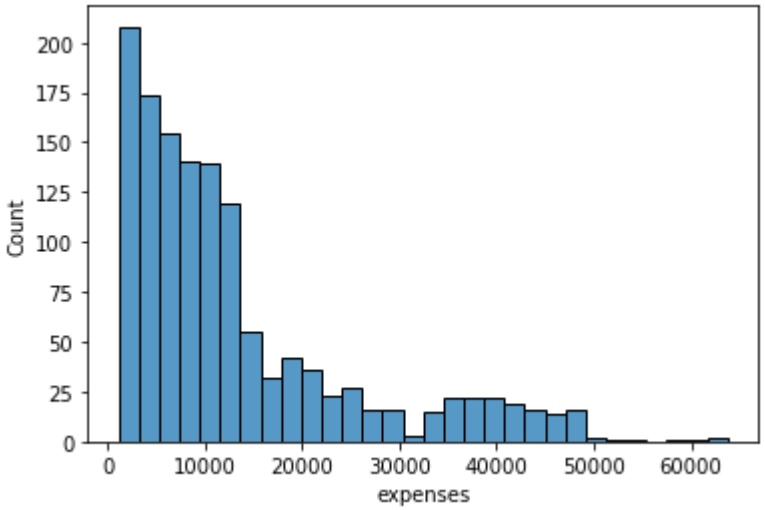
In [11]:
```
y= df_zscore.expenses
x=df_zscore.drop(columns=['predictions', 'expenses']).assign(const=1)

results=sm.OLS(y, x).fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               expenses   R-squared:                       0.751
Model:                            OLS   Adj. R-squared:                  0.749
Method:                 Least Squares   F-statistic:                     500.9
Date:                Thu, 10 Aug 2023   Prob (F-statistic):               0.00
Time:                        16:49:34   Log-Likelihood:                -968.62
No. Observations:                1338   AIC:                             1955.
Df Residuals:                    1329   BIC:                             2002.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
age                0.2980      0.014     21.586      0.000       0.271       0.325
bmi                0.1709      0.014     11.864      0.000       0.143       0.199
children           0.0474      0.014      3.452      0.001       0.020       0.074
sex_male          -0.0054      0.014     -0.395      0.693      -0.032       0.022
smoker_yes         0.7950      0.014     57.723      0.000       0.768       0.822
region_northwest  -0.0125      0.017     -0.741      0.459      -0.046       0.021
region_southeast  -0.0381      0.018     -2.163      0.031      -0.073      -0.004
region_southwest  -0.0340      0.017     -2.007      0.045      -0.067      -0.001
const           3.296e-17      0.014   2.41e-15      1.000      -0.027       0.027
==============================================================================
Omnibus:                      300.499   Durbin-Watson:                   2.088
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              719.382
Skew:                           1.212   Prob(JB):                    6.14e-157
Kurtosis:                       5.652   Cond. No.                         2.21
==============================================================================

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

In [12]:
```
import seaborn as sns

sns.histplot(y);
```



In [13]:
```
sns.histplot(df.expenses);
```



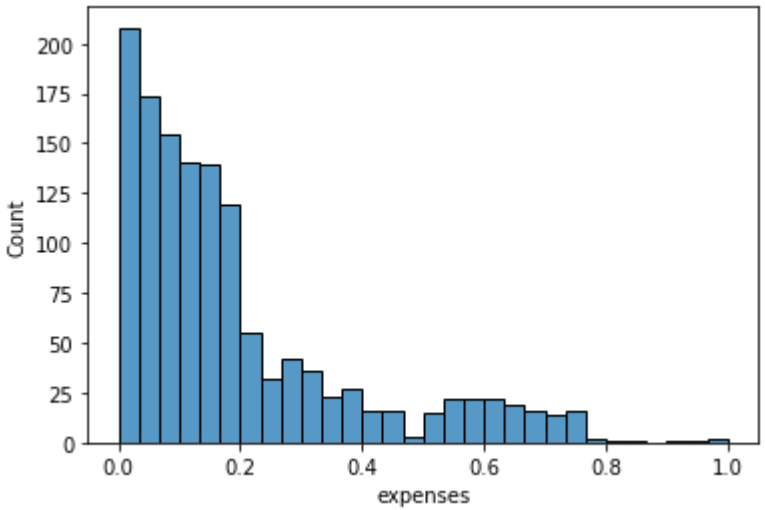
In [14]:
```
df_MinMax= pd.DataFrame(preprocessing.MinMaxScaler().fit_transform(df), columns=df.columns)
df_MinMax.head()
```

Out[14]:

	age	bmi	children	expenses	predictions	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	0.021739	0.320755	0.0	0.251611	0.198761	0.0	1.0	0.0	0.0	1.0

	age	bmi	children	expenses	predictions	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
1	0.000000	0.479784	0.2	0.009636	0.306026	1.0	0.0	0.0	1.0	0.0
2	0.217391	0.458221	0.6	0.053115	0.458506	1.0	0.0	0.0	1.0	0.0
3	0.326087	0.180593	0.0	0.333010	0.275973	1.0	0.0	1.0	0.0	0.0
4	0.304348	0.347709	0.0	0.043816	0.362244	1.0	0.0	1.0	0.0	0.0

In [15]:
```
sns.histplot(df_MinMax.expenses);
```



In [16]:
```
y= df_MinMax.expenses
x=df_MinMax.drop(columns=['predictions', 'expenses']).assign(const=1)

results=sm.OLS(y, x).fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               expenses   R-squared:                       0.751
Model:                            OLS   Adj. R-squared:                  0.749
Method:                 Least Squares   F-statistic:                     500.9
Date:                Thu, 10 Aug 2023   Prob (F-statistic):               0.00
Time:                        16:49:36   Log-Likelihood:                1230.9
No. Observations:                1338   AIC:                            -2444.
```

```
Df Residuals:                     1329   BIC:                            -2397.
Df Model:                            8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
age              0.1886      0.009     21.586      0.000       0.171       0.206
bmi              0.2009      0.017     11.864      0.000       0.168       0.234
children         0.0380      0.011      3.452      0.001       0.016       0.060
sex_male        -0.0021      0.005     -0.395      0.693      -0.013       0.008
smoker_yes       0.3807      0.007     57.723      0.000       0.368       0.394
region_northwest -0.0056     0.008     -0.741      0.459      -0.021       0.009
region_southeast -0.0165     0.008     -2.163      0.031      -0.032      -0.002
region_southwest -0.0153     0.008     -2.007      0.045      -0.030      -0.000
const            -0.0481     0.009     -5.137      0.000      -0.066      -0.030
==============================================================================
Omnibus:                      300.499   Durbin-Watson:                   2.088
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              719.382
Skew:                           1.212   Prob(JB):                    6.14e-157
Kurtosis:                       5.652   Cond. No.                         9.58
==============================================================================

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

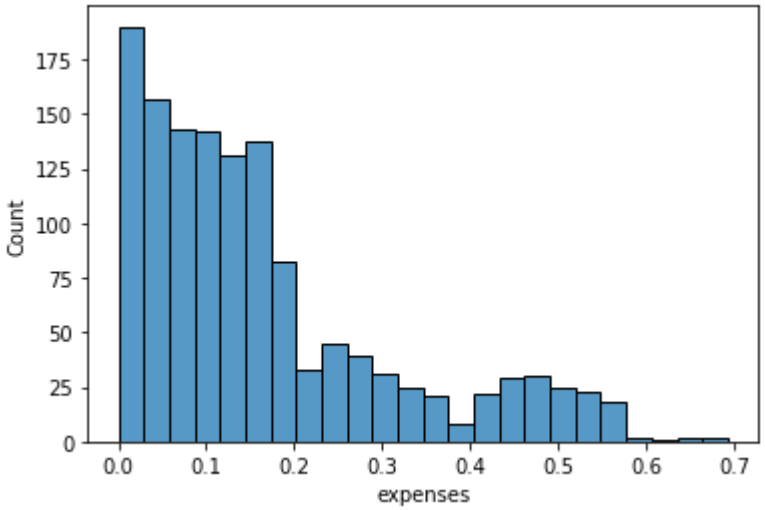
_MLR_OLS assumptions normality multicollinearity VIF

In [17]:

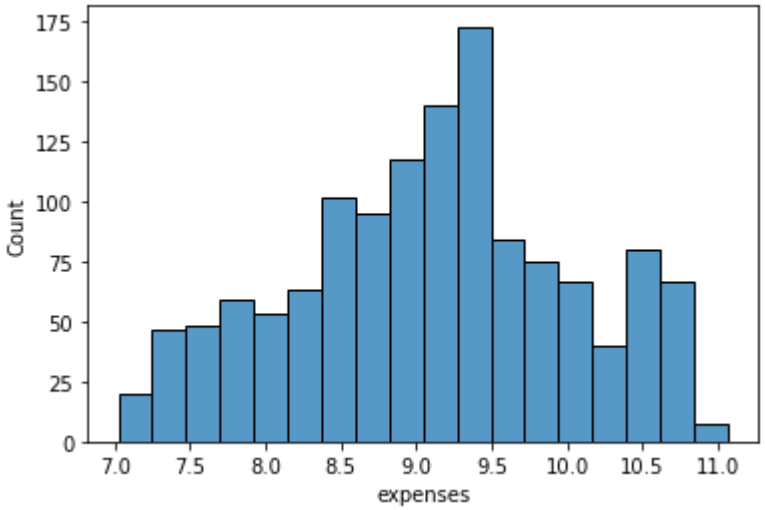
```
y= np.log1p(y)
sns.histplot(y)
```

Out[17]:

```
<AxesSubplot:xlabel='expenses', ylabel='Count'>
```



In [18]:
```
sns.histplot(np.log(df.expenses));
```



In [19]:
```
y= np.log(df.expenses)
x= df.drop(columns=['predictions', 'expenses']).assign(const=1)

print(sm.OLS(y, x).fit().summary())
```

```
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:               expenses   R-squared:                       0.768
Model:                            OLS   Adj. R-squared:                  0.767
Method:                 Least Squares   F-statistic:                     549.7
Date:                Thu, 10 Aug 2023   Prob (F-statistic):               0.00
Time:                        16:49:36   Log-Likelihood:                -808.54
No. Observations:                1338   AIC:                             1635.
Df Residuals:                    1329   BIC:                             1682.
Df Model:                           8
Covariance Type:            nonrobust
================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
age                0.0346      0.001     39.654      0.000       0.033       0.036
bmi                0.0134      0.002      6.377      0.000       0.009       0.017
children           0.1019      0.010     10.086      0.000       0.082       0.122
sex_male          -0.0754      0.024     -3.090      0.002      -0.123      -0.028
smoker_yes         1.5543      0.030     51.330      0.000       1.495       1.614
region_northwest  -0.0638      0.035     -1.827      0.068      -0.132       0.005
region_southeast  -0.1572      0.035     -4.480      0.000      -0.226      -0.088
region_southwest  -0.1289      0.035     -3.680      0.000      -0.198      -0.060
const              7.0308      0.072     97.111      0.000       6.889       7.173
==============================================================================
Omnibus:                      463.941   Durbin-Watson:                   2.046
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1674.108
Skew:                           1.679   Prob(JB):                         0.00
Kurtosis:                       7.331   Cond. No.                         311.
==============================================================================

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

_Multicollinearity

In [20]:

```
# VIF = variance inflation factor= 1/ (1-R^2)
def vif(df):
    import pandas as pd
    from sklearn.linear_model import LinearRegression

    #initialize dictionaries
    vif_dict, tolerance_dict = {}, {}

    #from input data for each exogenous veriable
```

```
    for col in df.drop(columns=['const']):
        y= df[col]
        x= df.drop(columns=[col])

        #extract r_squared from the fit

        r_squared = LinearRegression().fit(x, y).score(x, y)

        #calculate VIF
        if r_squared < 1: # Prevent division by zero runtime error
            vif = 1/(1- r_squared)
        else:
            vif = 100
        vif_dict[col] = vif

        #calculate tolerance
        tolerance = 1- r_squared
        tolerance_dict[col] = tolerance

        # generate the DataFrame to return
        df.output = pd.DataFrame({'VIF': vif_dict, 'Tolerance': tolerance_dict})

    return df_output.sort_values(by=['VIF'] , ascending=False)

VIF(x)
#10 > adequate
#5 >good
#3 >ideal
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\TAWABC~1\AppData\Local\Temp/ipykernel_12936/3568147519.py in <module>
     33     return df_output.sort_values(by=['VIF'] , ascending=False)
     34
---> 35 VIF(x)
     36 #10 > adequate
     37 #5 >good

NameError: name 'VIF' is not defined
```

In [ ]: