In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

Here i took a SAS data to find the exact difference for R-sq and Adjust R-sq

In [2]:
```
healthmpd.read_sas7bdat("C:\\Users\\anirv\\Downloads\\health.sas7bdat")
```

In [3]:
```
health.head()
```

Out[3]:

	X1	X2	X3	X4	X5	y6
0	64.9	78.0	284.0	9.1	109.0	28.0
1	70.3	68.0	433.0	8.7	144.0	29.0
2	60.8	70.0	739.0	7.2	113.0	27.0
3	72.5	25.0	250.0	2.5	34.0	23.0
4	76.7	74.0	477.0	8.3	206.0	21.0

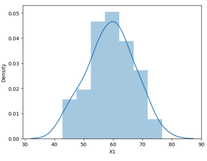
In [4]:
```
health.info() # There is no missing values
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53 entries, 0 to 52
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   X1      53 non-null     float64
 1   X2      53 non-null     float64
 2   X3      53 non-null     float64
 3   X4      53 non-null     float64
 4   X5      53 non-null     float64
 5   y6      53 non-null     float64
dtypes: float64(6)
memory usage: 2.6 KB
```

Here i want to know the variable is normally distributed or not

In [7]:
```
sns.distplot(health["X1"]) # it is normally distributed
```

Out[7]:
```
<AxesSubplot:xlabel="X1", ylabel="Density">
```



Our goal is minimising the sum of square of deviation [OLS-method]

In [11]:
```
import statsmodels.formula.api as sm

model=sm.ols(formula="X1~X2+X3+X4+X5+y6", data=health).fit()
model.summary()
```

Out[11]:

OLS Regression Results

Dep. Variable:	X1	**R-squared:**	0.953
Model:	OLS	**Adj. R-squared:**	0.948
Method:	Least Squares	**F-statistic:**	191.1
Date:	Sat, 04 Mar 2023	**Prob (F-statistic):**	1.06e-30
Time:	11:03:02	**Log-Likelihood:**	-102.74
No. Observations:	53	**AIC:**	217.5
Df Residuals:	47	**BIC:**	229.3
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	75.4490	2.619	28.814	0.000	70.181	80.717
X2	-0.1025	0.008	-12.644	0.000	-0.119	-0.086
X3	-0.0118	0.001	-11.498	0.000	-0.014	-0.010
X4	-1.0950	0.211	-5.202	0.000	-1.519	-0.672
X5	0.0994	0.006	17.698	0.000	0.088	0.111
y6	0.0503	0.092	0.548	0.586	-0.134	0.235

Omnibus:	4.263	**Durbin-Watson:**	2.116
Prob(Omnibus):	0.119	**Jarque-Bera (JB):**	2.609
Skew:	-0.337	**Prob(JB):**	0.271
Kurtosis:	2.148	**Cond. No.**	6.93e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.93e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

intercept+75.44 -0.1025*x2 -0.0118*x3 -1.095*x4 -0.0994*x5 -0.053*x6 # mathematical Equation# Here going to eliminate the x6 variable because there is no relation reason: if the value is less than 0.05 there is some relation -> if the value is more than 0.05 there is no relation

Once check the differnce between from the above and below table

1.--> From the above table the R squared value is 0.93 see the below table here i removed the one reduant variable the R squared doesnot changed 2.--> From the above table the Adj.R squared value is 0.948 see the below table here after removed the one reduant variable the Adj.R-squared is change the Adj.R-squared value is 0.949

In [14]:
```
import statsmodels.formula.api as sm

model=sm.ols(formula="X1~X2+X3+X4+X5", data=health).fit()
model.summary()
```

Out[14]:

OLS Regression Results

Dep. Variable:	X1	**R-squared:**	0.953
Model:	OLS	**Adj. R-squared:**	0.949
Method:	Least Squares	**F-statistic:**	242.4
Date:	Sat, 04 Mar 2023	**Prob (F-statistic):**	3.53e-31
Time:	11:14:26	**Log-Likelihood:**	-102.91
No. Observations:	53	**AIC:**	215.8
Df Residuals:	48	**BIC:**	225.7
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	76.5587	1.650	46.402	0.000	73.241	79.876
X2	-0.1029	0.008	-12.757	0.000	-0.119	-0.086
X3	-0.0116	0.001	-11.890	0.000	-0.014	-0.010
X4	-1.0878	0.209	-5.215	0.000	-1.507	-0.668
X5	0.0989	0.006	17.957	0.000	0.088	0.110

Omnibus:	5.230	**Durbin-Watson:**	2.162
Prob(Omnibus):	0.073	**Jarque-Bera (JB):**	2.977
Skew:	-0.351	**Prob(JB):**	0.233
Kurtosis:	2.091	**Cond. No.**	4.42e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.42e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Check the above two results the R-sq does not change the values it increases the values or constant it doesnot decrease when the variables increase or decrease

The R-sq is the proportion of total variation in Y explained by all X variables taken together (the model)

in simple linear regression we use R square Because there is only one independent variable

In [ ]:
```

```

In [15]:
```
health.head()
```

Out[15]:

	X1	X2	X3	X4	X5	y6
0	64.9	78.0	284.0	9.1	109.0	28.0
1	70.3	68.0	433.0	8.7	144.0	29.0
2	60.8	70.0	739.0	7.2	113.0	27.0
3	72.5	25.0	250.0	2.5	34.0	23.0
4	76.7	74.0	477.0	8.3	206.0	21.0

In [17]:
```
health["Predict"]=model.predict()
```

In [18]:
```
health.head()
```

Out[18]:

	X1	X2	X3	X4	X5	y6	Predict
0	64.9	78.0	284.0	9.1	109.0	28.0	66.191506
1	70.3	68.0	433.0	8.7	144.0	29.0	69.379553
2	60.8	70.0	739.0	7.2	113.0	27.0	64.190430
3	72.5	25.0	250.0	2.5	34.0	23.0	71.751975
4	76.7	74.0	477.0	8.3	206.0	21.0	74.824799

In [19]:
```
health["Error"]=health["X1"]-health["Predict"]
```

In [20]:
```
health.head()
```

Out[20]:

	X1	X2	X3	X4	X5	y6	Predict	Error
0	64.9	78.0	284.0	9.1	109.0	28.0	66.191506	-1.291507
1	70.3	68.0	433.0	8.7	144.0	29.0	69.379553	0.920447
2	60.8	70.0	739.0	7.2	113.0	27.0	64.190430	-3.390430
3	72.5	25.0	250.0	2.5	34.0	23.0	71.751975	0.748025
4	76.7	74.0	477.0	8.3	206.0	21.0	74.824799	1.875201

In [23]:
```
round(sum(health["Error"]),2)
```

Out[23]:
```
0.0
```

In [24]:
```
air=pd.read_csv("C:\\Users\\anirv\\Downloads\\data sets\\AirPassengers.csv")
```

In [25]:
```
air.head()
```

Out[25]:

	Week_num	Passengers	Promotion_Budget	Service_Quality_Score	Holiday_week	Delayed_Cancelled_flight_ind	Inter_metro_flight_ratio	Bad_Weather_Ind	Technical_Issues_Ind
0	1	37824	517356	4.00000	NO	NO	0.70	YES	YES
1	2	47936	646086	2.67466	NO	YES	0.80	YES	YES
2	3	42896	638330	3.29473	NO	NO	0.90	NO	NO
3	4	31792	506452	3.85684	NO	NO	0.40	NO	NO
4	5	38624	609658	3.90757	NO	NO	0.87	NO	YES

In [28]:
```
air1=air[["Passengers","Promotion_Budget","Service_Quality_Score","Inter_metro_flight_ratio"]].copy()
```

In [31]:
```
air1.head()
```

Out[31]:

	Passengers	Promotion_Budget	Service_Quality_Score	Inter_metro_flight_ratio
0	37824	517356	4.00000	0.70
1	47936	646086	2.67466	0.80
2	42896	638330	3.29473	0.90
3	31792	506452	3.85684	0.40
4	38624	609658	3.90757	0.87

In [38]:
```
import statsmodels.formula.api as sm1

model1=sm1.ols(formula="Passengers~Promotion_Budget+Service_Quality_Score+Inter_metro_flight_ratio",data=air1).fit()
model1.summary()
```

Out[38]:

OLS Regression Results

Dep. Variable:	Passengers	**R-squared:**	0.951
Model:	OLS	**Adj. R-squared:**	0.949
Method:	Least Squares	**F-statistic:**	495.6
Date:	Sat, 04 Mar 2023	**Prob (F-statistic):**	8.77e-50
Time:	11:35:09	**Log-Likelihood:**	-738.45
No. Observations:	80	**AIC:**	1485.
Df Residuals:	76	**BIC:**	1494.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.521e+04	3542.694	5.424	0.000	1.22e+04	2.83e+04
Promotion_Budget	0.0153	0.004	15.476	0.000	0.048	0.062
Service_Quality_Score	-2802.0708	530.382	-5.283	0.000	-3858.419	-1745.723
Inter_metro_flight_ratio	-2003.4508	2129.095	-0.941	0.350	-6243.912	2237.010

Omnibus:	6.902	**Durbin-Watson:**	2.312
Prob(Omnibus):	0.032	**Jarque-Bera (JB):**	2.759
Skew:	-0.051	**Prob(JB):**	0.252
Kurtosis:	2.096	**Cond. No.**	8.22e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.22e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

Here i removed the one variable

look in to here after remoed the one redunt variable Adjust- Rsq is increase

In [39]:
```
import statsmodels.formula.api as sm1

model1=sm1.ols(formula="Passengers~Promotion_Budget+Service_Quality_Score",data=air1).fit()
model1.summary()
```

Out[39]:

OLS Regression Results

Dep. Variable:	Passengers	**R-squared:**	0.951
Model:	OLS	**Adj. R-squared:**	0.950
Method:	Least Squares	**F-statistic:**	744.0
Date:	Sat, 04 Mar 2023	**Prob (F-statistic):**	4.38e-51
Time:	11:35:31	**Log-Likelihood:**	-738.91
No. Observations:	80	**AIC:**	1484.
Df Residuals:	77	**BIC:**	1491.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.853e+04	1454.796	12.740	0.000	1.56e+04	2.14e+04
Promotion_Budget	0.0144	0.003	16.063	0.000	0.048	0.061
Service_Quality_Score	-2807.3095	529.958	-5.297	0.000	-3862.592	-1752.028

Omnibus:	7.720	**Durbin-Watson:**	2.331
Prob(Omnibus):	0.021	**Jarque-Bera (JB):**	2.913
Skew:	-0.043	**Prob(JB):**	0.233
Kurtosis:	2.069	**Cond. No.**	7.97e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.97e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

Once check the differnce between from the above and below table

1.--> From the above table the R squared value is 0.951 see the below table here i removed the one reduant variable the R squared doesnot changed 2.--> From the above table the Adj.R squared value is 0.949 see the below table here after removed the one reduant variable the Adj.R-squared is change the Adj.R squared value is 0.950

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```