Experiment 4: Regression Analysis

Name: Aryan Dali

UID: 2019120016

Class: TE EXTC

The given data set is a climate change dataset with the amount of various gases and substances in air and the temperature of the air in every year. We have to find out the correaltion of all the variables and build linear regression models with the data

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score
```

```
#importing the data and understanding it
data = pd.read_csv("/content/climate_change (1).csv")

data.head(10)
```

	Year	Month	MEI	CO2	CH4	N2O	CFC-11	CFC-12	TSI	Aerosols	T
0	1983	5	2.556	345.96	1638.59	303.677	191.324	350.113	1366.1024	0.0863	0.
1	1983	6	2.167	345.52	1633.71	303.746	192.057	351.848	1366.1208	0.0794	0.
2	1983	7	1.741	344.15	1633.22	303.795	192.818	353.725	1366.2850	0.0731	0.
3	1983	8	1.130	342.25	1631.35	303.839	193.602	355.633	1366.4202	0.0673	0.
4	1983	9	0.428	340.17	1648.40	303.901	194.392	357.465	1366.2335	0.0619	0.
5	1983	10	0.002	340.30	1663.79	303.970	195.171	359.174	1366.0589	0.0569	0.
6	1983	11	-0.176	341.53	1658.23	304.032	195.921	360.758	1366.1072	0.0524	0.
7	1983	12	-0.176	343.07	1654.31	304.082	196.609	362.174	1366.0607	0.0486	0.
8	1984	1	-0.339	344.05	1658.98	304.130	197.219	363.359	1365.4261	0.0451	0.
9	1984	2	-0.565	344.77	1656.48	304.194	197.759	364.296	1365.6618	0.0416	0.

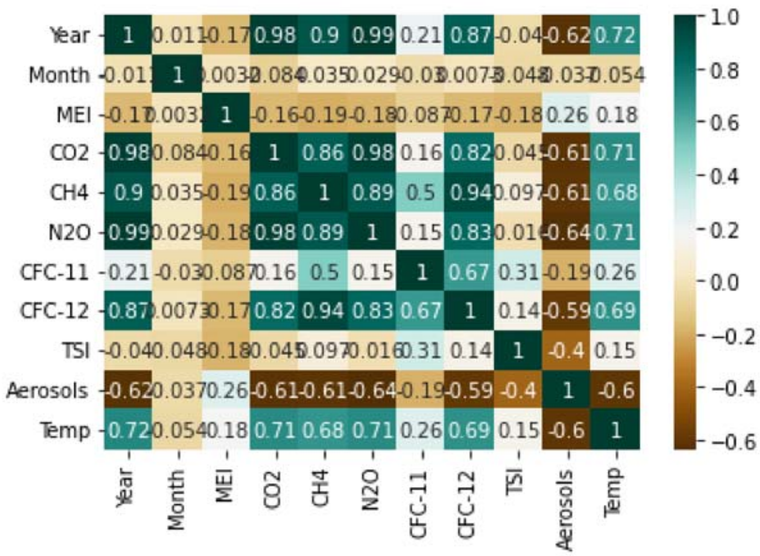
Data Cleaning

```
#finding inter quartile range (IQR) to remove outliers
Q1 = data.quantile(0.25)
Q3 = data.quantile(0.75)
IQR = Q3 - Q1
#print(IQR)
data = data[~((data < (Q1 - 1.5 * IQR)) |(data > (Q3 + 1.5 * IQR))).any(axis=1)]
```

Correlation heaat map of all he variables in the dataset

```
c = data.corr()
sns.heatmap(c, cmap = 'BrBG', annot = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f028d063650>
```



From the above correlation map we can see that CO2, CH4,N20 and CFC-12 are highly corellated with the Temperature.

```
data.columns
```

```
Index(['Year', 'Month', 'MEI', 'CO2', 'CH4', 'N2O', 'CFC-11', 'CFC-12', 'TSI',
       'Aerosols', 'Temp'],
      dtype='object')
```

Model Building with all variables

```
import statsmodels.api as sm
x = data[['MEI', 'CO2', 'CH4', 'N2O', 'CFC-11', 'CFC-12', 'TSI',
       'Aerosols']]
y = data[['Temp']]
```

```
X2 = sm.add_constant(x)
est = sm.OLS(y, X2)
est2 = est.fit()
print(est2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   Temp   R-squared:                       0.703
Model:                            OLS   Adj. R-squared:                  0.692
Method:                 Least Squares   F-statistic:                     69.11
Date:                Wed, 18 May 2022   Prob (F-statistic):           2.36e-57
Time:                        05:28:25   Log-Likelihood:                 251.36
No. Observations:                 243   AIC:                            -484.7
Df Residuals:                     234   BIC:                            -453.3
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -60.8378     23.736     -2.563      0.011    -107.600     -14.075
MEI            0.0665      0.007      9.650      0.000       0.053       0.080
CO2            0.0033      0.002      1.389      0.166      -0.001       0.008
CH4           -0.0005      0.001     -0.895      0.372      -0.002       0.001
N2O           -0.0033      0.010     -0.319      0.750      -0.023       0.017
CFC-11        -0.0032      0.002     -1.319      0.188      -0.008       0.002
CFC-12         0.0027      0.001      2.173      0.031       0.000       0.005
TSI            0.0449      0.018      2.532      0.012       0.010       0.080
Aerosols      -8.2339      2.042     -4.032      0.000     -12.257      -4.211
==============================================================================
Omnibus:                        3.269   Durbin-Watson:                   1.015
Prob(Omnibus):                  0.195   Jarque-Bera (JB):                2.996
Skew:                           0.194   Prob(JB):                        0.224
Kurtosis:                       3.381   Cond. No.                     9.94e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specif
[2] The condition number is large, 9.94e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/tsatools.py:117: FutureWarning:
  x = pd.concat(x[::order], 1)
```

The variables that have a P-Value less than 0.05 are CFC-12, TSI, Aerosols and MEI

```
df1 = data[data.iloc[:,0]<=2006]
df1.head()
```

	Year	Month	MEI	CO2	CH4	N2O	CFC-11	CFC-12	TSI	Aerosols
29	1985	10	-0.140	343.08	1681.56	305.395	215.327	390.676	1365.5269	0.0101
30	1985	11	-0.050	344.40	1680.68	305.530	216.282	392.714	1365.6289	0.0097

```
df1.shape
```

```
(219, 11)
```

```
df2 = data[data.iloc[:,0]>2006]
df2.head()
```

	Year	Month	MEI	CO2	CH4	N2O	CFC-11	CFC-12	TSI	Aerosols
284	2007	1	0.974	382.93	1799.66	320.561	248.372	539.206	1365.7173	0.0054
285	2007	2	0.510	383.81	1803.08	320.571	248.264	538.973	1365.7145	0.0051
286	2007	3	0.074	384.56	1803.10	320.548	247.997	538.811	1365.7544	0.0045
287	2007	4	-0.049	386.40	1802.11	320.518	247.574	538.586	1365.7228	0.0045
288	2007	5	0.183	386.58	1795.65	320.445	247.224	538.130	1365.6932	0.0041

```
df2.shape
```

```
(24, 11)
```

```
x = data[['MEI', 'CO2', 'CH4', 'N2O', 'CFC-11', 'CFC-12', 'TSI', 'Aerosols']]
y = data['Temp']
```

```
x_train = df1[['MEI','CO2', 'CH4', 'N2O', 'CFC-11', 'CFC-12', 'TSI', 'Aerosols']]
y_train = df1['Temp']
x_test = df2[['MEI', 'CO2', 'CH4', 'N2O', 'CFC-11',  'CFC-12', 'TSI', 'Aerosols']]
y_test = df2['Temp']
# 'CO2', 'CH4', 'N2O', 'CFC-11',
```

```
X2_train = sm.add_constant(x_train)
est_train = sm.OLS(y_train, X2_train)
est2 = est_train.fit()
print(est2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   Temp   R-squared:                       0.722
Model:                            OLS   Adj. R-squared:                  0.711
Method:                 Least Squares   F-statistic:                     68.15
Date:                Wed, 18 May 2022   Prob (F-statistic):           3.37e-54
```

```
Time:                        05:28:25   Log-Likelihood:                 229.49
No. Observations:                 219   AIC:                            -441.0
Df Residuals:                     210   BIC:                            -410.5
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -51.0320     24.469     -2.086      0.038     -99.268      -2.796
MEI            0.0622      0.007      8.508      0.000       0.048       0.077
CO2            0.0050      0.002      1.995      0.047    5.82e-05       0.010
CH4           -0.0004      0.001     -0.689      0.491      -0.001       0.001
N2O            0.0018      0.012      0.156      0.876      -0.021       0.025
CFC-11        -0.0011      0.003     -0.406      0.685      -0.007       0.004
CFC-12         0.0014      0.001      0.940      0.348      -0.002       0.004
TSI            0.0360      0.019      1.931      0.055      -0.001       0.073
Aerosols      -8.4359      2.024     -4.167      0.000     -12.427      -4.445
==============================================================================
Omnibus:                        6.330   Durbin-Watson:                   0.994
Prob(Omnibus):                  0.042   Jarque-Bera (JB):                6.027
Skew:                           0.363   Prob(JB):                       0.0491
Kurtosis:                       3.366   Cond. No.                     9.82e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specif
[2] The condition number is large, 9.82e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/tsatools.py:117: FutureWarning:
  x = pd.concat(x[::order], 1)
```

```
from sklearn.linear_model import LinearRegression
mlr = LinearRegression()
mlr.fit(x_train, y_train)
```

```
LinearRegression()
```

```
print("Intercept: ", mlr.intercept_)
print("Coefficients:")
list(zip(x, mlr.coef_))
```

```
Intercept:  -51.031969159858036
Coefficients:
[('MEI', 0.06223569777302381),
 ('CO2', 0.0049606987940408465),
 ('CH4', -0.00038810727802363575),
 ('N2O', 0.0018262419311547184),
 ('CFC-11', -0.0011344993284555694),
 ('CFC-12', 0.0014013277032073551),
 ('TSI', 0.03604734063953119),
 ('Aerosols', -8.435947559286046)]
```

```
#Prediction of test set
y_pred_mlr= mlr.predict(x_test)
#Predicted values
print("Prediction for test set: {}".format(y_pred_mlr))
```

```
Prediction for test set: [0.47395865 0.4503635  0.43347525 0.43430288 0.45407379 0.4255
 0.4256269  0.40086034 0.34299562 0.34452906 0.34506256 0.3546125
 0.37297068 0.35481019 0.34472978 0.3962589  0.43969896 0.46681556
 0.45350739 0.42400764 0.38408517 0.36584435 0.38171572 0.3874683 ]
```

```
#Actual value and the predicted value
mlr_diff = pd.DataFrame({'Actual value': y_test, 'Predicted value': y_pred_mlr})
mlr_diff.head()
```

	Actual value	Predicted value
284	0.601	0.473959
285	0.498	0.450364
286	0.435	0.433475
287	0.466	0.434303
288	0.372	0.454074

```
from sklearn import metrics
meanAbErr = metrics.mean_absolute_error(y_test, y_pred_mlr)
meanSqErr = metrics.mean_squared_error(y_test, y_pred_mlr)
rootMeanSqErr = np.sqrt(metrics.mean_squared_error(y_test, y_pred_mlr))
print('R squared: {:.2f}'.format(mlr.score(x,y)*100))
print('Mean Absolute Error:', meanAbErr)
print('Mean Square Error:', meanSqErr)
print('Root Mean Square Error:', rootMeanSqErr)
```

```
R squared: 69.69
Mean Absolute Error: 0.07881358260266709
Mean Square Error: 0.01062595645613692
Root Mean Square Error: 0.10308228002977486
```

The Rsquared value for the model is is 69.69

Correlation of N2O and CFC-11 with all other variables

```
r = np.corrcoef(df1['N2O'], df1['MEI'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: -0.06177124908370091
```

```
r = np.corrcoef(df1['N2O'], df1['CO2'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.9749926361365618
```

```
r = np.corrcoef(df1['N2O'], df1['CH4'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.8903124993701752
```

```
r = np.corrcoef(df1['N2O'], df1['N2O'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 1.0
```

```
r = np.corrcoef(df1['N2O'], df1['CFC-11'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.3273891567210462
```

```
r = np.corrcoef(df1['N2O'], df1['CFC-12'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.8645200386566597
```

```
r = np.corrcoef(df1['N2O'], df1['TSI'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.15958759016062263
```

```
r = np.corrcoef(df1['N2O'], df1['Aerosols'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: -0.6609381809526801
```

N2O is correlated with CO2, CH4, CFC-12 since their correlation coefficient is greater than 0.7

```
r = np.corrcoef(df1['CFC-11'], df1['MEI'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: -0.16610201966431865
```

```
r = np.corrcoef(df1['CFC-11'], df1['CO2'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.34154482148525867
```

```
r = np.corrcoef(df1['CFC-11'], df1['CH4'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.627913787898016
```

```
r = np.corrcoef(df1['CFC-11'], df1['N2O'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.3273891567210462
```

```
r = np.corrcoef(df1['CFC-11'], df1['CFC-12'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.7493680735584559
```

```
r = np.corrcoef(df1['CFC-11'], df1['TSI'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: 0.2571940874708528
```

```
r = np.corrcoef(df1['CFC-11'], df1['Aerosols'])
print("The correlation coefficient is: " + str(r[1][0]))
```

```
The correlation coefficient is: -0.23054354377641395
```

CFC-11 is corelated with CFC-12 since their correlation coefficient is greater than 0.7

Question.

Current scientific opinion is that nitrous oxide and CFC-11 are greenhouse gases: gases that are able to trap heat from the sun and contribute to the heating of the Earth. However, the regression coefficients of both the N2O and CFC-11 variables are negative, indicating that increasing atmospheric concentrations of either of these two compounds is associated with lower global temperatures.

Which of the following is the simplest correct explanation for this contradiction?

Answer

III. All of the gas concentration variables reflect human development - N2O and CFC.11 are correlated with other variables in the data set.

Training the Model with N20, Mei, TSI and Aerosols only

```
x_train_1 = df1[['MEI', 'N2O', 'TSI', 'Aerosols']]
y_train_1 = df1['Temp']
x_test_1 = df2[['MEI', 'N2O', 'TSI', 'Aerosols']]
y_test_1 = df2['Temp']
x_1 = data[['MEI', 'N2O', 'TSI', 'Aerosols']]
y_1 = data[['Temp']]
```

```
X2_train_1 = sm.add_constant(x_train_1)
est_train_1 = sm.OLS(y_train_1, X2_train_1)
est2_1 = est_train_1.fit()
print(est2_1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   Temp   R-squared:                       0.706
Model:                            OLS   Adj. R-squared:                  0.701
Method:                 Least Squares   F-statistic:                     128.7
Date:                Wed, 18 May 2022   Prob (F-statistic):           8.92e-56
Time:                        05:52:21   Log-Likelihood:                 223.51
No. Observations:                 219   AIC:                            -437.0
Df Residuals:                     214   BIC:                            -420.1
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -53.1366     23.146     -2.296      0.023     -98.759      -7.514
MEI            0.0606      0.007      8.212      0.000       0.046       0.075
N2O            0.0217      0.002     12.113      0.000       0.018       0.025
TSI            0.0342      0.017      2.028      0.044       0.001       0.067
Aerosols      -8.3714      1.995     -4.197      0.000     -12.303      -4.439
==============================================================================
Omnibus:                        9.543   Durbin-Watson:                   0.940
Prob(Omnibus):                  0.008   Jarque-Bera (JB):               10.250
Skew:                           0.407   Prob(JB):                      0.00595
Kurtosis:                       3.678   Cond. No.                     5.45e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specif
[2] The condition number is large, 5.45e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/tsatools.py:117: FutureWarning:
  x = pd.concat(x[::order], 1)
```

```
from sklearn.linear_model import LinearRegression
mlr = LinearRegression()
```

```
mlr.fit(x_train_1, y_train_1)
```

```
LinearRegression()
```

```
print("Intercept: ", mlr.intercept_)
print("Coefficients:")
list(zip(x_1, mlr.coef_))
```

```
Intercept:  -53.13657186298024
Coefficients:
[('MEI', 0.06057100564913947),
 ('N2O', 0.0217419646685703),
 ('TSI', 0.03415981688806212),
 ('Aerosols', -8.37144911641541)]
```

The coeffcient of N2O in this model is = 0.0217

In the previous model the coefficient of N2O was = 0.001826

So the coefficient has increased in this model suggesting that N2O has a greater influence on this model than the previous one.

```
#Prediction of test set
y_pred_mlr_1= mlr.predict(x_test_1)
#Predicted values
print("Prediction for test set: {}".format(y_pred_mlr_1))
```

```
Prediction for test set: [0.4994973  0.47402556 0.45350238 0.44432044 0.4591232  0.4270
 0.4311838  0.42411848 0.38135484 0.38933851 0.38846345 0.39489663
 0.4093001  0.38835418 0.3726421  0.41817244 0.45693594 0.48525866
 0.47434819 0.45574824 0.43001199 0.42539002 0.43910982 0.44125798]
```

```
#Actual value and the predicted value
mlr_diff_1 = pd.DataFrame({'Actual value': y_test_1, 'Predicted value': y_pred_mlr_1})
mlr_diff_1.head()
```

	Actual value	Predicted value
284	0.601	0.499497
285	0.498	0.474026
286	0.435	0.453502
287	0.466	0.444320
288	0.372	0.459123

```
from sklearn import metrics
meanAbErr = metrics.mean_absolute_error(y_test_1, y_pred_mlr_1)
```

```
meanAbErr = metrics.mean_absolute_error(y_test_1, y_pred_mlr_1)
meanSqErr = metrics.mean_squared_error(y_test_1, y_pred_mlr_1)
rootMeanSqErr = np.sqrt(metrics.mean_squared_error(y_test_1, y_pred_mlr_1))
print('R squared: {:.2f}'.format(mlr.score(x_1,y_1)*100))
print('Mean Absolute Error:', meanAbErr)
print('Mean Square Error:', meanSqErr)
print('Root Mean Square Error:', rootMeanSqErr)
```

```
R squared: 67.00
Mean Absolute Error: 0.09020686837753002
Mean Square Error: 0.013717618459981736
Root Mean Square Error: 0.11712223725655917
```

The R-sqaured value of the new model is 67.00 which has been decreased from the previous model which was 69.69

Conclusion

1. The variables that have a P-Value less than 0.05 are CFC-12, TSI, Aerosols and MEI
2. The R-squared value for the model with alll the variables is 69.69
3. N20 and CFC-11 were correlated with all the variables and we found out that:

   a. N2O is correlated with CO2, CH4, CFC-12 since their correlation coefficient is greater than 0.7

   b. CFC-11 is corelated with CFC-12 since their correlation coefficient is greater than 0.7

   Therefore the statement [III. All of the gas concentration variables reflect human development - N2O and CFC.11 are correlated with other variables in the data set.]

   Is true
4. Now a model with only N20, Mei, TSI and Aerosols is trained.

   a. The coeffcient of N2O in this model is = 0.0217

   In the previous model the coefficient of N2O was = 0.001826

   So the coefficient has increased in this model suggesting that N2O has a greater influence on this model than the previous one.

   b. The R-sqaured value of the new model is 67.00 which has been decreased from the previous model which was 69.69. So the first model is a better model for our varibles than our previous models.

0s completed at 11:23 AM