assignment_1

March 27, 2024

Data Mining and Machine Learning - Assignment 1

1 Question 1 - NOx Study

Modelling of *LNOx* concentration as function of other variables

[25]:

```
# Import of used libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import statsmodels.api as sm
```

[2]:

```
# Import of the dataset
q1_pd = pd.read_csv('NOxEmissions.csv')
q1_pd
```

[2]:

```
      rownames  julday      LNOx    LNOxEm    sqrtWS
0          193     373  4.457250  5.536489  0.856446
1          194     373  4.151827  5.513000  1.016612
2          195     373  3.834061  4.886994  1.095445
3          196     373  4.172848  5.138912  1.354068
4          197     373  4.322807  5.666518  1.204159
…          …      …        …        …         …
8083      8779     730  5.000585  6.730993  1.396424
8084      8780     730  4.669552  6.165086  1.466288
8085      8781     730  4.380776  5.855493  1.559808
8086      8782     730  4.284276  5.691445  1.449138
8087      8783     730  4.143928  5.505866  1.466288

[8088 rows x 5 columns]
```

1.1 (a) - Data Pre-processing

In the pre-processing we want to address data quality problems like Incorrect Data, Missing Values, duplicate data, outliers…

- **Missing data:** No missing data found in the dataset

- **Duplicates:** No duplicates were found.

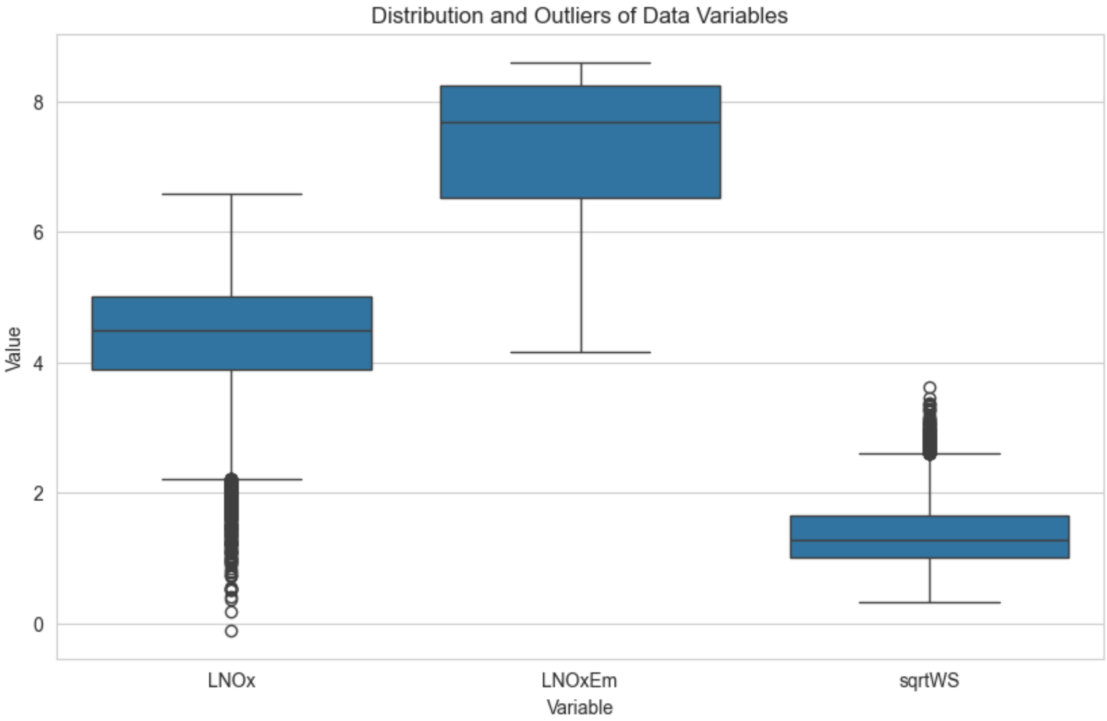
[15]:

```
# (a) - Pre-processing

# Check if missing/duplicated/Invalid data is present in the dataset

## Missing data
print(f"Number of missing data: {q1_pd.isnull().sum().sum()}")
## Duplicated data
print(f"Number of duplicated data: {q1_pd.duplicated().sum()}")

## Statistical Summary
print(f"===Statistical Summary===\n{q1_pd.describe()}")
```

```
Number of missing data: 0
Number of duplicated data: 0
===Statistical Summary===
          rownames       julday         LNOx       LNOxEm       sqrtWS
count  8088.000000  8088.000000  8088.000000  8088.000000  8088.000000
mean   4597.584570   556.078882     4.378691     7.338244     1.365253
std    2464.686179   102.706509     0.937389     1.016658     0.466280
min     193.000000   373.000000    -0.105361     4.157866     0.316228
25%    2507.750000   469.000000     3.891820     6.514982     1.016612
50%    4681.500000   560.000000     4.497028     7.692495     1.284523
75%    6709.250000   644.000000     5.012134     8.239159     1.648181
max    8783.000000   730.000000     6.576121     8.600040     3.624017
```

[18]:

```
# Check for outliers

melted_data = pd.melt(q1_pd, value_vars=['LNOx', 'LNOxEm', 'sqrtWS'],␣
 ↪var_name='Variable', value_name='Value')
sns.set_style("whitegrid")

plt.figure(figsize=(10, 6))
boxplot = sns.boxplot(x='Variable', y='Value', data=melted_data)
boxplot.set_title('Distribution and Outliers of Data Variables')
boxplot.set_ylabel('Value')
boxplot.set_xlabel('Variable')

plt.show()
```



1.2 (b) - Distribution of LNOx variable

To describe the distribution of the *LNOx* variable we are going to use descriptive statistics indicators along with diagrams for visualization.

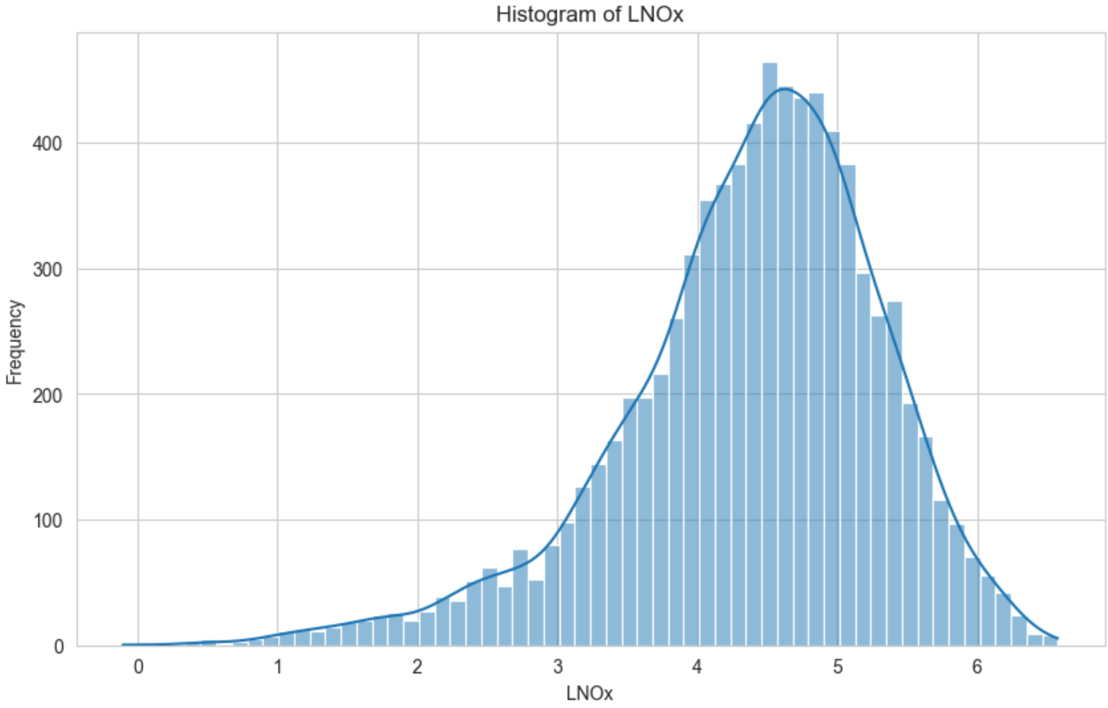
LNOx appears to have a normal distribution with a negative (left) skewness

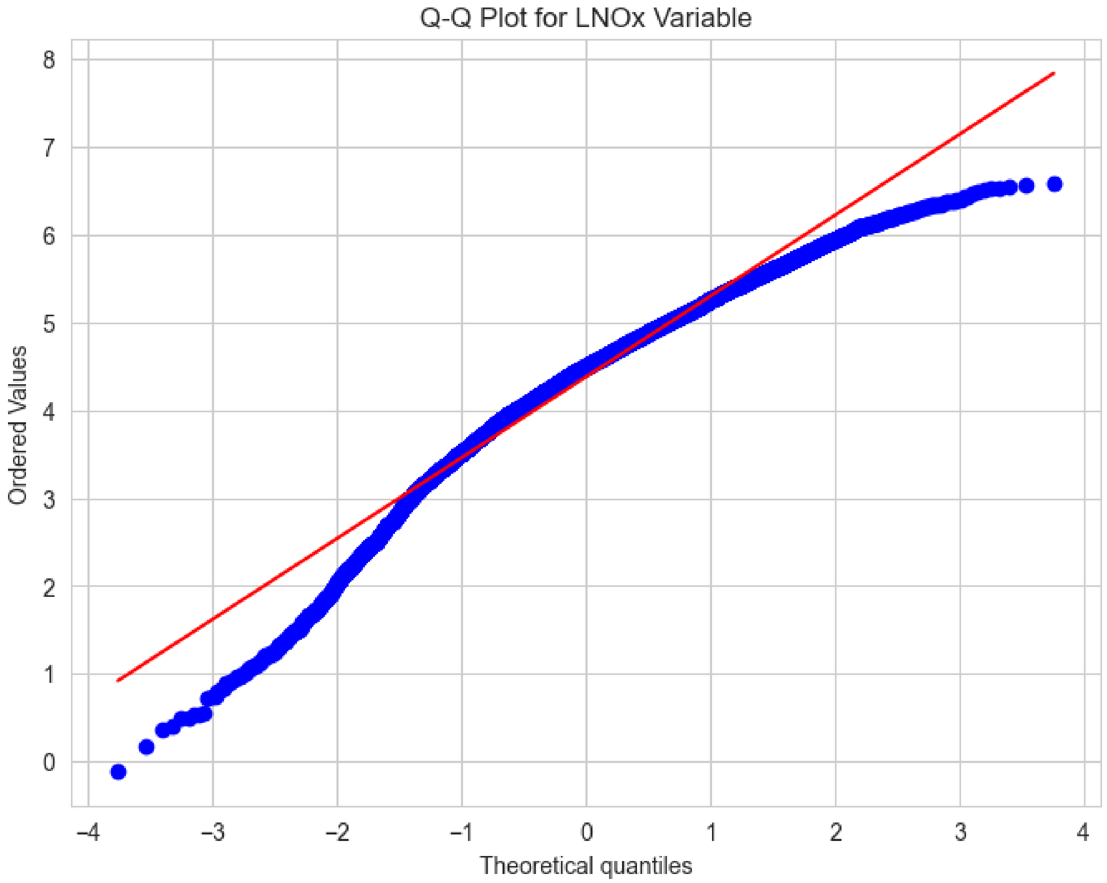
[24]:

```python
# (b) - LNOx distribution

lnox = q1_pd['LNOx']

## Descriptive Stats
range_lnox = lnox.max() - lnox.min()
print(f"Mean: {lnox.mean()}\nMedian: {lnox.median()}\nStandard Deviation: {lnox.std()}\nVariance: {lnox.var()}\nRange: {range_lnox}\nSkewness: {lnox.skew()}\nKurtosis: {lnox.kurt()}")

## Histogram plot
plt.figure(figsize=(10, 6))
sns.histplot(q1_pd['LNOx'], kde=True)
plt.title('Histogram of LNOx')
plt.xlabel('LNOx')
plt.ylabel('Frequency')
plt.show()
```

```python
# Q-Q plot
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111)
stats.probplot(q1_pd['LNOx'], dist="norm", plot=ax)
ax.set_title("Q-Q Plot for LNOx Variable")
plt.show()
```

```
Mean: 4.378690810185019
Median: 4.49702802736839
Standard Deviation: 0.937388582502527
Variance: 0.8786973546060968
Range: 6.681481834658996
Skewness: -0.8244320335510329
Kurtosis: 1.1307787937580986
```





1.3 (c) - Linear Model of LNOx

the $LNOx$ linear model is fitted below using a multiple linear regression, $LNOx$ is the dependent variable, $LNOxEm$ and $sqrtWS$ are the indipendent variables as requested by the question.

[27]:

```
# (c) - LNOx linear model

X = q1_pd[['LNOxEm', 'sqrtWS']]
X = sm.add_constant(X)
y = q1_pd['LNOx']
model = sm.OLS(y, X).fit()

# Print model summary
model.summary()
```

[27]:

Dep. Variable:	LNOx	**R-squared:**	0.663
Model:	OLS	**Adj. R-squared:**	0.663
Method:	Least Squares	**F-statistic:**	7952.
Date:	Wed, 27 Mar 2024	**Prob (F-statistic):**	0.00
Time:	12:29:35	**Log-Likelihood:**	-6554.7
No. Observations:	8088	**AIC:**	1.312e+04
Df Residuals:	8085	**BIC:**	1.314e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
const	1.0619	0.046	23.097	0.000	0.972	1.152
LNOxEm	0.6414	0.006	107.092	0.000	0.630	0.653
sqrtWS	-1.0182	0.013	-77.969	0.000	-1.044	-0.993

Omnibus:	28.937	**Durbin-Watson:**	0.497
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	30.943
Skew:	-0.115	**Prob(JB):**	1.91e-07
Kurtosis:	3.198	**Cond. No.**	58.3

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

2 Question 2 - Airbag study