Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Transformaciones

TC3004B.104 Inteligencia Artificial Avanzada para la Ciencia de Datos I

Profesores:

Ivan Mauricio Amaya Contreras

Blanca Rosa Ruiz Hernandez

Antonio Carlos Bento

Frumencio Olivas Alvarez

Hugo Terashima Marín

Julian Lawrence Gil Soares - A00832272

3 de Septiembre de 2023

In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, cal
l drive.mount("/content/drive", force_remount=True).
```

In [ ]:
```
#Lectura de archivo y selecion de variables para analizar

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import statsmodels.api as sm

data = pd.read_csv('/content/drive/MyDrive/Stats/mc-donalds-menu-1.csv')

data['Carbohydrates'] = np.where(data['Carbohydrates'] <= 0, 1e-6, data['Car
```

In [ ]:
```
#Box-Cox transformation
transformedData, lambdaValue = stats.boxcox(data['Carbohydrates'])

print("Lambda value:", lambdaValue)
```

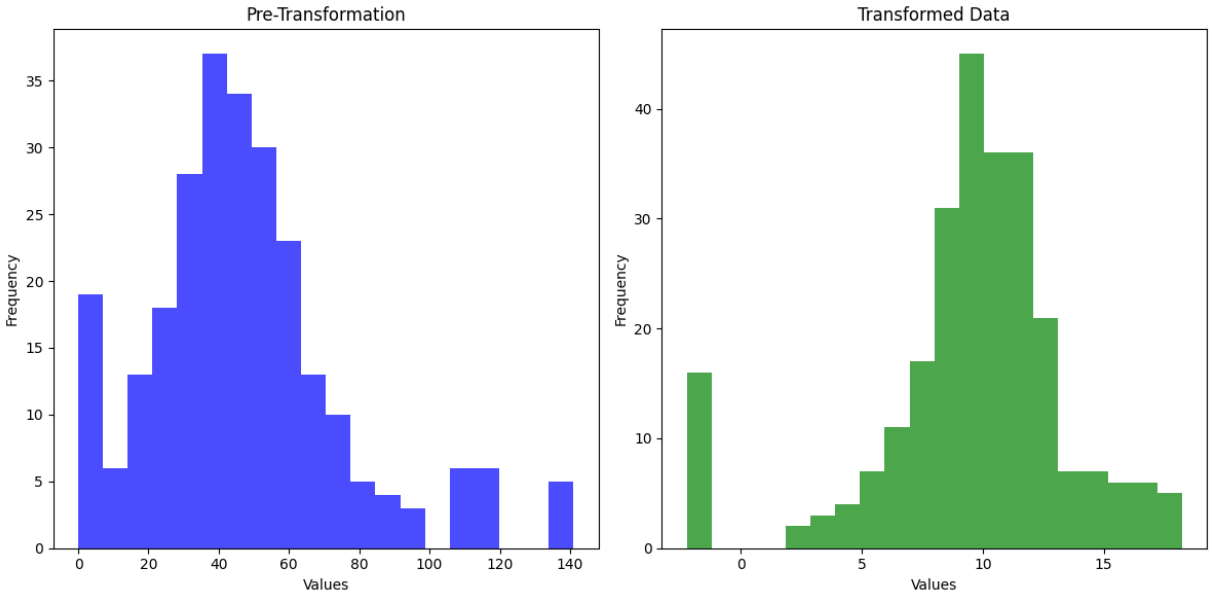
```
Lambda value: 0.4475286790617023
```

Ecuacion original: y(λ)=(y^λ-1)/λ ,log(y)

Ecuacion sustituido: y(.45)=(y^.45-1)/.45 ,log(y)

In [ ]:
```
originalData = data['Carbohydrates']
transformed_data, _ = stats.boxcox(originalData)

original = {
    'Minimum': np.min(originalData),
    'Maximum': np.max(originalData),
    'Mean': np.mean(originalData),
    'Median': np.median(originalData),
    'Q1': np.percentile(originalData, 25),
    'Q3': np.percentile(originalData, 75),
    'Skewness': stats.skew(originalData),
    'Kurtosis': stats.kurtosis(originalData)
}

transformed = {
    'Minimum': np.min(transformedData),
    'Maximum': np.max(transformedData),
    'Mean': np.mean(transformedData),
    'Median': np.median(transformedData),
    'Q1': np.percentile(transformedData, 25),
    'Q3': np.percentile(transformedData, 75),
    'Skewness': stats.skew(transformedData),
    'Kurtosis': stats.kurtosis(transformedData)
}
```

```
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.hist(originalData, bins=20, color='blue', alpha=0.7)
plt.title('Pre-Transformation')
plt.xlabel('Values')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
plt.hist(transformed_data, bins=20, color='green', alpha=0.7)
plt.title('Transformed Data')
plt.xlabel('Values')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()

alpha = 0.05

andersonOriginal = stats.anderson(originalData, dist='norm')
andersonTransformed = stats.anderson(transformed_data, dist='norm')

print("Original data:")
print("Statistic:", andersonOriginal.statistic)
print("Critical Values:", andersonOriginal.critical_values)
print("Significance Levels:", andersonOriginal.significance_level)

print("\nTransformed data:")
print("Statistic:", andersonTransformed.statistic)
print("Critical Values:", andersonTransformed.critical_values)
print("Significance Levels:", andersonTransformed.significance_level)

jb_original = sm.stats.jarque_bera(originalData)
jb_transformed = sm.stats.jarque_bera(transformed_data)

print("\nOriginal data:")
print("Statistic:", jb_original[0])
print("p-value:", jb_original[1])

print("\nTransformed data:")
print("Statistic:", jb_transformed[0])
print("p-value:", jb_transformed[1])
```




```
Original data:
Statistic: 4.140223823947167
Critical Values: [0.567 0.646 0.775 0.904 1.076]
Significance Levels: [15.  10.   5.   2.5  1. ]

Transformed data:
Statistic: 8.058843907744972
Critical Values: [0.567 0.646 0.775 0.904 1.076]
Significance Levels: [15.  10.   5.   2.5  1. ]

Original data:
Statistic: 55.646413397093596
p-value: 8.25153720093447e-13

Transformed data:
Statistic: 90.87952070835992
p-value: 1.844008077782418e-20
```

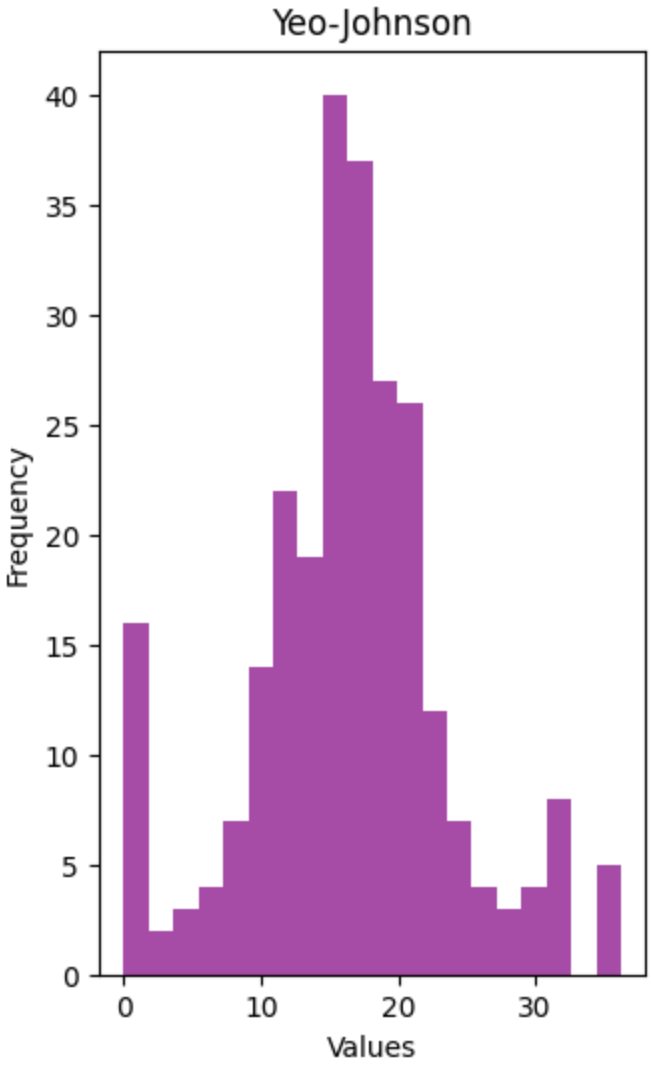
In [ ]:
```
#Yeo-Johnson
transformedDataYeo, lambdaValue = stats.yeojohnson(data['Carbohydrates'])

print("Lambda value:", lambdaValue)
```

```
Lambda value: 0.6442759156008031
```

In [ ]:
```
statsYeo = {
    'Minimum': np.min(transformedDataYeo),
    'Maximum': np.max(transformedDataYeo),
    'Mean': np.mean(transformedDataYeo),
    'Median': np.median(transformedDataYeo),
    'Q1': np.percentile(transformedDataYeo, 25),
    'Q3': np.percentile(transformedDataYeo, 75),
    'Skewness': stats.skew(transformedDataYeo),
    'Kurtosis': stats.kurtosis(transformedDataYeo)
}

plt.figure(figsize=(12, 6))
plt.subplot(1, 3, 3)
```

```
plt.hist(transformedDataYeo, bins=20, color='purple', alpha=0.7)
plt.title('Yeo-Johnson')
plt.xlabel('Values')
plt.ylabel('Frequency')

ad= stats.anderson(transformedDataYeo, dist='norm')
jb= sm.stats.jarque_bera(transformedDataYeo)

print("Yeo-Johnson:")
print(statsYeo)
print("\nAnderson-Darling:")
print("Statistic:", ad.statistic)
print("\nJarque-Bera:")
print("Statistic:", jb[0])
print("p-value:", jb[1])
```

```
Yeo-Johnson:
{'Minimum': 9.999998220186988e-07, 'Maximum': 36.25690965716253, 'Mean': 16.
46994535681572, 'Median': 16.480252389761144, 'Q1': 12.631144979080263, 'Q
3': 20.384661960100356, 'Skewness': -0.027311414900634615, 'Kurtosis': 0.653
6831903408977}

Anderson-Darling:
Statistic: 3.0002763911932107

Jarque-Bera:
Statistic: 4.661424807755655
p-value: 0.09722645790588215
```



In [ ]:
```
'''
Entre las dos transformaciones las que mejor resultados tuvo fue Yeo-Johnsor
en la transformacion Yeo-Johnson nuestro valor de P fue mayor asi nuestros c
que en la transformacion box-cox.
'''
```

Out[ ]:
```
'\nEntre las dos transformaciones las que mejor resultados tuvo fue Yeo-Joh
nson. Comparamos el valor de p\nen la transformacion Yeo-Johnson nuestro va
lor de P fue mayor asi nuestros datos terminaron mas normalizados \nque en
la transformacion box-cox.\n'
```