Actividad t-student

Ernesto Reynoso Lizárraga A01639915

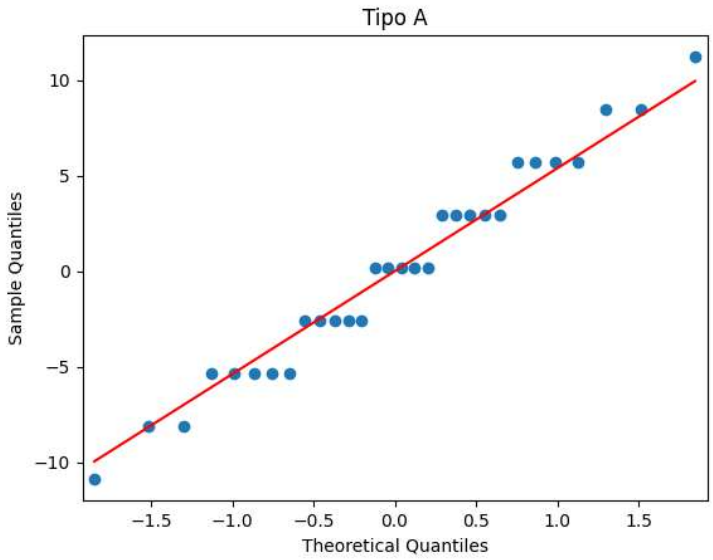
```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from scipy import stats
import math
```

```
df = pd.DataFrame({'A':[20, 25, 22, 23, 28, 26, 24, 21, 27, 25, 24, 22, 23, 26, 25, 23, 24, 22, 27, 26, 25, 24, 23, 22, 21, 26, 24, 25, 22, 23],
                   'B':[19, 18, 21, 20, 23, 22, 20, 19, 22, 21, 20, 19, 18, 23, 22, 21, 20, 19, 23, 22, 21, 20, 19, 18, 23, 22, 21, 20, 19, 18]})
```

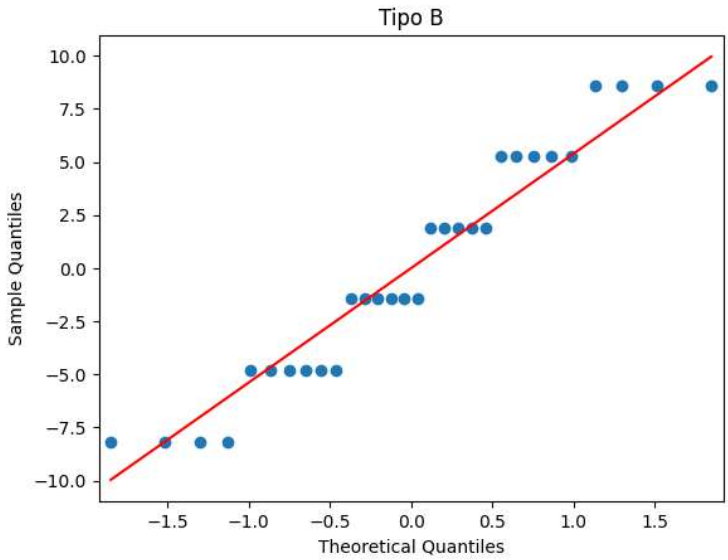
```
def estandar(data):
  mean = data.mean()
  std = data.std()/math.sqrt(df.shape[0])
  data_standar = (data-mean)/std
  return data_standar
```

```
df_estandarizada = df.apply(estandar)
```

```
QQ1 = sm.qqplot(df_estandarizada['A'], stats.norm, line='s')
plt.title('Tipo A')
plt.show()
```




```
QQ2 = sm.qqplot(df_estandarizada['B'], stats.norm, line='s')
plt.title('Tipo B')
plt.show()
```



```
stats.kstest(df_estandarizada['A'],'norm')
```

```
KstestResult(statistic=0.42839148113992026, pvalue=1.6245194483008648e-05, statistic_location=-2.5798716450471013, statistic_sign=1)
```

```
stats.kstest(df_estandarizada['B'],'norm')
```

```
KstestResult(statistic=0.4602396467922807, pvalue=2.465171784639058e-06, statistic_location=-1.453131044320842, statistic_sign=1)
```

En el test de Kolmogorov Smirnov se rechaza la hipotesis, por lo que los datos no son normales.

Intervalo de confianza

Nivel de confianza del 99%

los grados de libertad son 29 y 0.005, por lo que el valor critico de la distribucion es 2.756

```
n=df.shape[0]
df_mean = df.mean()
t = 2.756
```

Intevalo Tipo A

```
a_s = math.sqrt(sum((df['A'] - df_mean['A'])**2)/(n-1))
a_li = df_mean['A']  - t * (a_s/math.sqrt(n))
a_ls = df_mean['A']  + t * (a_s/math.sqrt(n))
print ("[",a_li,",",a_ls,"]")
print("Ancho:",a_ls - a_li)
```

```
[ 22.936281146025888 , 24.93038552064078 ]
Ancho: 1.994104374614892
```

Intervalo Tipo B

```
b_s = math.sqrt(sum((df['B'] - df_mean['B'])**2)/(n-1))
b_li = df_mean['B']  - t * (b_s/math.sqrt(n))
b_ls = df_mean['B']  + t * (b_s/math.sqrt(n))
print ("[",b_li,",",b_ls,"]")
print("Ancho:",b_ls - b_li)
```

```
[ 19.611475819976828 , 21.25519084668984 ]
Ancho: 1.6437150267130107
```