

EXEMPLO 1 (ESTUDO SÍNCRONO E ASSÍNCRONO)

```
In [6]: import numpy as np
import statistics
#import scipy as scp

sync = np.array([94. , 84.9, 82.6, 69.5, 80.1, 79.6, 81.4, 77.8, 81.7, 78.8, 73.2,
                87.9, 87.9, 93.5, 82.3, 79.3, 78.3, 71.6, 88.6, 74.6, 74.1, 80.6])
asynchr = np.array([77.1, 71.7, 91. , 72.2, 74.8, 85.1, 67.6, 69.9, 75.3, 71.7, 65.7, 72.6, 71
```

```
In [7]: sync.std()
```

```
Out[7]: 6.383736114287583
```

```
In [8]: sync.ptp()
```

```
Out[8]: 24.5
```

```
In [9]: print(sync.mean())
print(asynchr.mean())
```

```
81.01363636363634
74.60000000000001
```

```
In [10]: # TESTE DE NORMALIDADE!

# VERIFICAR SE OS DADOS SEGUEM UMA DISTRIBUIÇÃO NORMAL!

# teste de hipótese de Shapiro-Wilk

# H0:Os dados têm distribuição normal;
# H1:Os dados não têm distribuição normal.

# Conclusões:
# Se p-valor (valor de prova) <= 0.05, rejeita H0 (H0 é falso) e assume H1 (H1 é verdadeiro);
# Se p-valor (valor de prova) > 0.05 não deve rejeitar H0 (H0 é verdadeiro).

# 0.05 = 5% é a cauda do intervalo de confiança de 95%.

from scipy.stats import shapiro
from scipy import stats

def check_normality(data):
    test_stat_normality, p_value_normality=stats.shapiro(data)
    print("p value:%.4f" % p_value_normality)
    if p_value_normality <0.05:
        print("Reject null hypothesis >> The data is not normally distributed")
    else:
        print("Fail to reject null hypothesis >> The data is normally distributed")
```

```
In [11]: check_normality(sync)
check_normality(asynchr)
```

```
p value:0.6556
Fail to reject null hypothesis >> The data is normally distributed
p value:0.0803
Fail to reject null hypothesis >> The data is normally distributed
```

```
In [32]: print(sync.mean())
print(statistics.median(sync))
```

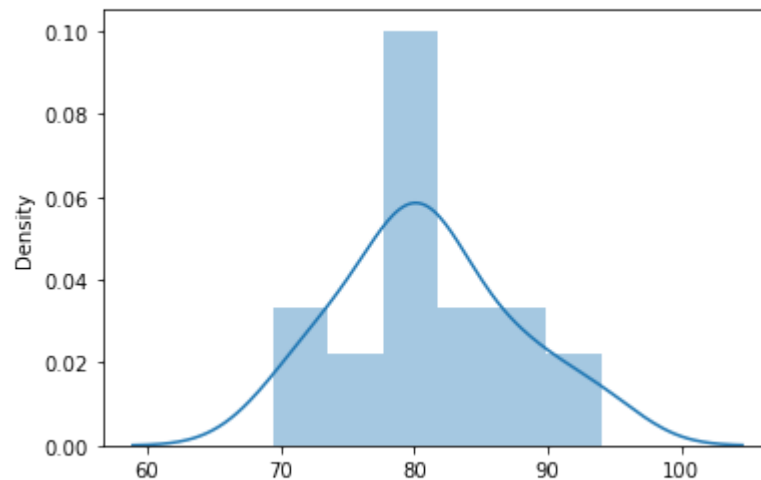
81.013636363634
80.35

```
In [33]: import seaborn as sns
import matplotlib.pyplot as plt

sns.distplot(sync)
plt.show()
```

C:\Users\Lucas\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

warnings.warn(msg, FutureWarning)



```
In [34]: # TESTANDO A VARIABILIDADE DOS DADOS

# Aplicando o teste de homogeneidade de Levene

# H0:As variâncias são iguais;
# H1:As variâncias são diferentes.

# Conclusões:
# Se pvalue <= 0.05, rejeita H0 e assume H1;
# Se pvalue > 0.05 não deve rejeitar H0.

from scipy.stats import levene

test_stat_var, p_value_var= stats.levene(sync, asyncr)
print("p value:%.4f" % p_value_var)
if p_value_var <0.05:
    print("Reject null hypothesis >> The variances of the samples are different.")
else:
    print("Fail to reject null hypothesis >> The variances of the samples are same.")

p value:0.8149
Fail to reject null hypothesis >> The variances of the samples are same.
```

```
In [35]: # PARÂMETROS:
# PRÉ-REQUISITOS
# DUAS AMOSTRAS NÃO PAREADAS
# SÃO NORMALMENTE DISTRIBUIDAS
# MESMA VARIÂNCIA

# APLICANDO T-TEST (TESTE PARAMÉTRICO)

# Hipóteses
# H0: as médias são iguais
# H1: as médias são diferentes

# Conclusões:
# Se pvalue <= 0.05, rejeita H0 e assume H1;
# Se pvalue > 0.05 não deve rejeitar H0.
```

```
ttest,p_value = stats.ttest_ind(sync,asyncr)
print("p value:%.8f" % p_value)
print("since the hypothesis is one sided >> use p_value/2 >> p_value_one_sided:%.4f" %(p_valu
if p_value/2 <0.05:
    print("Reject null hypothesis")
else:
    print("Fail to reject null hypothesis")
```

p value:0.00753598
 since the hypothesis is one sided >> use p_value/2 >> p_value_one_sided:0.0038
 Reject null hypothesis

In [9]: *# Conclusão: As médias das notas da população dos alunos que estudaram de forma síncrona e as
 # a média dos alunos com estudo síncrono é maior que a dos alunos com estudo assíncrono.*

EXEMPLO 2 (ALIMENTAÇÃO COM LEITE MATERNO, FÓRMULA OU OS DOIS)

```
In [44]: only_breast=np.array([794.1, 716.9, 993. , 724.7, 760.9, 908.2, 659.3 , 690.8, 768.7, 717.3 ,
    714.1, 810.3, 583.5, 679.9, 865.1])

only_formula=np.array([ 898.8, 881.2, 940.2, 966.2, 957.5, 1061.7, 1046.2, 980.4, 895.6, 919.
    796.3, 859.6, 871.1 , 1047.5, 919.1 , 1160.5, 996.9])

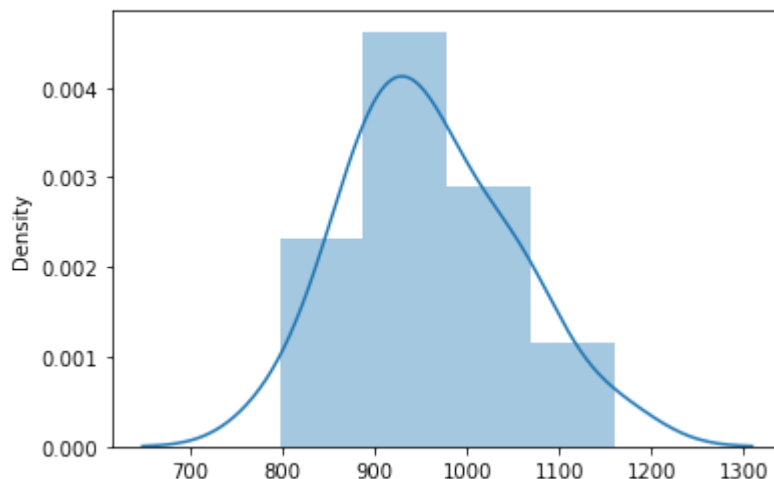
both=np.array([976.4, 656.4, 861.2, 706.8, 718.5, 717.1, 759.8, 894.6, 867.6, 805.6, 765.4, 8
    740. , 799.4, 790.3, 795.2 , 823.6, 818.7, 926.8, 791.7, 948.3])
```

```
In [45]: check_normality(only_breast)
check_normality(only_formula)
check_normality(both)
```

p value:0.4694
 Fail to reject null hypothesis >> The data is normally distributed
 p value:0.8879
 Fail to reject null hypothesis >> The data is normally distributed
 p value:0.7973
 Fail to reject null hypothesis >> The data is normally distributed

```
In [46]: sns.distplot(only_formula)
plt.show()
```

C:\Users\Lucas\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `dis
 tplot` is a deprecated function and will be removed in a future version. Please adapt your co
 de to use either `displot` (a figure-level function with similar flexibility) or `histplot`
 (an axes-level function for histograms).
 warnings.warn(msg, FutureWarning)



```
In [47]: stat, pvalue levene= stats.levene(only_breast,only_formula,both)
print("p value:%.4f" % pvalue levene)
```

```

if pvalue levene < 0.05:
    print("Reject null hypothesis >> The variances of the samples are different.")
else:
    print("Fail to reject null hypothesis >> The variances of the samples are same.")

```

p value:0.7673

Fail to reject null hypothesis >> The variances of the samples are same.

```

In [48]: # PARÂMETROS:
# PRÉ-REQUISITOS
# MAIS DE DUAS AMOSTRAS NÃO PAREADAS
# SÃO NORMALMENTE DISTRIBUIDAS
# MESMA VARIÂNCIA

# APLICANDO ANOVA - ANÁLISE DE VARIÂNCIA (TESTE PARAMÉTRICO)

#H0: as médias são iguais;
#H1: pelo menos uma das médias é diferente.

# Conclusões:
# Se pvalue <= 0.05, rejeita H0 e assume H1;
# Se pvalue > 0.05 não deve rejeitar H0.

F, p_value = stats.f_oneway(only_breast, only_formula, both)
print("p value: %.6f" % p_value)
if p_value < 0.05:
    print("Reject null hypothesis")
else:
    print("Fail to reject null hypothesis")

```

p value:0.000000

Reject null hypothesis

```

In [15]: #Conclusão: Pelo menos uma das médias da população é diferente.

```

```

In [16]: # PARÂMETROS:
# DUAS AMOSTRAS NÃO PAREADAS
# SÃO NORMALMENTE DISTRIBUIDAS
# MESMA VARIÂNCIA

# APLICANDO T-TEST

#H0: as médias são iguais
#H1: as médias são diferentes

# Conclusões:
# Se pvalue <= 0.05, rejeita H0 e assume H1;
# Se pvalue > 0.05 não deve rejeitar H0.

import scikit_posthocs as sp
posthoc_df = sp.posthoc_ttest([only_breast, only_formula, both], equal_var=True, p_adjust="bonf")

group_names = ["only breast", "only formula", "both"]
posthoc_df.columns = group_names
posthoc_df.index = group_names
posthoc_df.style.applymap(lambda x: "background-color:violet" if x < 0.05 else "background-color:white")

```

```

Out[16]:

```

	only breast	only formula	both
only breast	1.000000	0.000000	0.129454
only formula	0.000000	1.000000	0.000004
both	0.129454	0.000004	1.000000

```

In [17]: print(only_breast.mean())
print(only_formula.mean())
print(both.mean())

```

749.8235294117648
959.2157894736845
809.9521739130436

```
In [18]: # only_formula e only_breast p-valor = 0, as médias das populações são diferentes;  
# only_formula e both p-valor = 0, as médias das populações são diferentes;  
# only_breast e both p-valor = 0.129, as médias das populações são iguais;  
  
#Conclusão 1: alimentar os bebês com Leite materno ou intercalando Leite materno e fórmula te  
#Conclusão 2: o aumento médio de peso dos bebês que são alimentados com Leite artificial (fór  
#alimentados com Leite materno ou com os dois intercalados.
```

EXEMPLO 3 (HORAS EXTRAS POR EQUIPES DE DESENVOLVIMENTO DE SOFTWARE E EQUIPE DE TESTE)

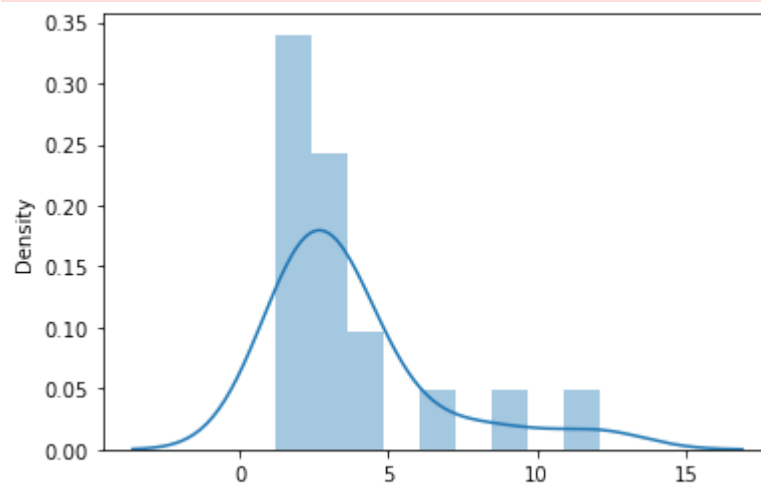
```
In [20]: test_team=np.array([6.2, 7.1, 1.5, 2.3, 2, 1.5, 6.1, 2.4, 2.3, 12.4, 1.8, 5.3, 3.1, 9.4, 2.3  
developer_team=np.array([2.3, 2.1, 1.4, 2.0, 8.7, 2.2, 3.1, 4.2, 3.6, 2.5, 3.1, 6.2, 12.1, 3.
```

```
In [21]: check_normality(test_team)  
check_normality(developer_team)
```

p value:0.0046
Reject null hypothesis >> The data is not normally distributed
p value:0.0005
Reject null hypothesis >> The data is not normally distributed

```
In [22]: sns.distplot(developer_team)  
plt.show()
```

C:\Users\Lucas\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)



```
In [23]: stat, pvalue_levene= stats.levene(test_team, developer_team)  
print("p value:%.4f" % pvalue_levene)  
if pvalue_levene < 0.05:  
    print("Reject null hypothesis >> The variances of the samples are different.")  
else:  
    print("Fail to reject null hypothesis >> The variances of the samples are same.")
```

p value:0.5410
Fail to reject null hypothesis >> The variances of the samples are same.

```
In [24]: # PARÂMETROS:  
# PRÉ-REQUISITOS  
# DUAS AMOSTRAS NÃO PAREADAS
```

```
# MESMA VARIÂNCIA
```

```
# CONSTATAÇÃO
```

```
# NÃO SÃO NORMALMENTE DISTRIBUIDOS (NORMALIDADE NÃO INFLUENCIA NO TESTE)
```

```
# APLICANDO MANN-WHITNEY U TEST (NÃO PARAMÉTRICO)
```

```
# Hipóteses
```

```
#H0: as médias são iguais
```

```
#H1: as médias são diferentes
```

```
# Conclusões:
```

```
# Se pvalue <= 0.05, rejeita H0 e assume H1;
```

```
# Se pvalue > 0.05 não deve rejeitar H0.
```

```
ttest,pvalue = stats.mannwhitneyu(test_team,developer_team, alternative="two-sided")
```

```
print("p-value:%.4f" % pvalue)
```

```
if pvalue <0.05:
```

```
    print("Reject null hypothesis")
```

```
else:
```

```
    print("Fail to recejt null hypothesis")
```

```
p-value:0.8226
```

```
Fail to recejt null hypothesis
```

```
In [26]: print(developer_team.mean())
```

```
print(test_team.mean())
```

```
3.7764705882352945
```

```
4.264705882352941
```

```
In [ ]: # Conclusão: As médias das horas extras trabalhadas por semana da população de teste é igual  
# população de desenvolvimento.
```

EXEMPLO 4 (CONSUMIDORES ATRAIDOS PELAS PLATAFORMAS YOUTUBE, INSTAGRAM E FACEBOOK POR DIA)

```
In [37]: youtube = np.array([1913, 1879, 1939, 2146, 2040, 2127, 2122, 2156, 2036, 1974, 1956, 2146, 2
```

```
instagram = np.array([2305., 2355., 2203., 2231., 2185., 2420., 2386., 2410., 2340., 2349., 2
```

```
facebook = np.array([2133., 2522., 2124., 2551., 2293., 2367., 2460., 2311., 2178., 2113., 20
```

```
In [38]: check_normality(youtube)
```

```
check_normality(instagram)
```

```
check_normality(facebook)
```

```
p value:0.0285
```

```
Reject null hypothesis >> The data is not normally distributed
```

```
p value:0.4156
```

```
Fail to reject null hypothesis >> The data is normally distributed
```

```
p value:0.1716
```

```
Fail to reject null hypothesis >> The data is normally distributed
```

```
In [39]: stat, pvalue levene= stats.levene(youtube, instagram, facebook)
```

```
print("p value:%.4f" % pvalue levene)
```

```
if pvalue levene <0.05:
```

```
    print("Reject null hypothesis >> The variances of the samples are different.")
```

```
else:
```

```
    print("Fail to reject null hypothesis >> The variances of the samples are same.")
```

p value:0.0012
Reject null hypothesis >> The variances of the samples are different.

```
In [40]: # PARÂMETROS:
# PRÉ-REQUISITOS
# MAIS DE DUAS AMOSTRAS NÃO PAREADAS

# CONSTATAÇÃO
# NÃO SÃO NORMALMENTE DISTRIBUIDOS (NORMALIDADE NÃO INFLUENCIA NO TESTE)
# VARIÂNCIAS DIFERENTES (VARIABILIDADE NÃO INFLUENCIA NO TESTE)

# APLICANDO KRUSKAL-WALLIS ANOVA (NÃO PARAMÉTRICO)

#H0: as médias são iguais
#H1: pelo menos uma das médias é diferente

# Conclusões:
# Se pvalue <= 0.05, rejeita H0 e assume H1;
# Se pvalue > 0.05 não deve rejeitar H0.

F, p_value = stats.kruskal(youtube, instagram, facebook)
print("p value:%.6f" % p_value)
if p_value <0.05:
    print("Reject null hypothesis")
else:
    print("Fail to reject null hypothesis")

p value:0.000015
Reject null hypothesis
```

```
In [ ]: # Conclusão: Existe significância suficiente para afirmar que pelo menos uma das médias da po
```

```
In [41]: # APLICANDO MANN-WHITNEY U TEST

#H0: as médias são iguais
#H1: as médias são diferentes

# Conclusões:
# Se pvalue <= 0.05, rejeita H0 e assume H1;
# Se pvalue > 0.05 não deve rejeitar H0.

posthoc_df = sp.posthoc_mannwhitney([youtube,instagram, facebook], p_adjust = 'bonferroni')
group_names= ["youtube", "instagram","facebook"]
posthoc_df.columns= group_names
posthoc_df.index= group_names
posthoc_df.style.applymap(lambda x: "background-color:violet" if x<0.05 else "background-colo
```

```
Out[41]:
```

	youtube	instagram	facebook
youtube	1.000000	0.000010	0.002337
instagram	0.000010	1.000000	1.000000
facebook	0.002337	1.000000	1.000000

```
In [42]: print(youtube.mean())
print(instagram.mean())
print(facebook.mean())

2043.5333333333333
2307.5333333333333
2295.4
```

```
In [43]: # Nível de confiança 95%
# Conclusão 1: Podemos afirmar que a população do facebook e Instagram tem a mesma média de c
# Conclusão 2: Podemos afirmar que a população do Youtube tem média de consumidores atraídos
# do facebook e do instagram.
```

EXEMPLO 5 (NÍVEL DE COLESTEROL ANTES DA DIETA E DEPOIS DA DIETA)

```
In [51]: before_diet=np.array([224, 235, 223, 253, 253, 224, 244, 225, 259, 220, 242, 240, 239, 229, 2
after_diet=np.array([198, 195, 213, 190, 246, 206, 225, 199, 214, 210, 188, 205, 200, 220, 19
```

```
In [52]: check_normality(before_diet)
check_normality(after_diet)

p value:0.1635
Fail to reject null hypothesis >> The data is normally distributed
p value:0.1003
Fail to reject null hypothesis >> The data is normally distributed
```

```
In [53]: # T-TEST PARA AMOSTRAS PAREADAS (PARAMÉTRICO)

# PRÉ-REQUISITO
# DISTRIBUIÇÃO NORMAL

#H0: as médias são iguais
#H1: as médias são diferentes

# Conclusões:
# Se pvalue <= 0.05, rejeita H0 e assume H1;
# Se pvalue > 0.05 não deve rejeitar H0.

test_stat, p_value_paired = stats.ttest_rel(before_diet,after_diet)
print("p value:%.6f" % p_value_paired , "one tailed p value:%.6f" %(p_value_paired/2))
if p_value_paired <0.05:
    print("Reject null hypothesis")
else:
    print("Fail to reject null hypothesis")

p value:0.000008 one tailed p value:0.000004
Reject null hypothesis
```

```
In [ ]: # Conclusão: Existe significância suficiente para afirmar que a média de colesterol na popul
# da dieta são diferentes, e podemos observar que a média de colesterol na população após a d
```

EXEMPLO 6 (SCORE DE DESEMPENHO DE EMPRESAS)

```
In [54]: piedpiper=np.array([4.57, 4.55, 5.47, 4.67, 5.41, 5.55, 5.53, 5.63, 3.86, 3.97, 5.44, 3.93, 5
endframe = np.array([4.27, 3.93, 4.01, 4.07, 3.87, 4. , 4. , 3.72, 4.16, 4.1 , 3.9 , 3.97, 4.
```

```
In [55]: check_normality(piedpiper)
check_normality(endframe)

p value:0.0304
Reject null hypothesis >> The data is not normally distributed
p value:0.9587
Fail to reject null hypothesis >> The data is normally distributed
```

```
In [56]: # APLICANDO WILCOXON SIGNED RANK (NÃO PARAMÉTRICO) PARA AMOSTRAS PAREADAS

# Hipóteses
#H0: as médias são iguais
#H1: as médias são diferentes

# Conclusões:
# Se pvalue <= 0.05, rejeita H0 e assume H1;
# Se pvalue > 0.05 não deve rejeitar H0.
```



```
test,pvalue = stats.wilcoxon(endframe,piedpiper) ##alternative default two sided
print("p-value:%.6f" %pvalue, ">> one_tailed_pval:%.6f" %(pvalue/2))

test,one_sided_pvalue = stats.wilcoxon(endframe,piedpiper, alternative="less")
print("one sided pvalue:%.6f" %(one_sided_pvalue))
if pvalue <0.05:
    print("Reject null hypothesis")
else:
    print("Fail to recejt null hypothesis")
```

```
p-value:0.000214 >> one_tailed_pval:0.000107
one sided pvalue:0.000107
Reject null hypothesis
```

```
In [ ]: # Conclusão: Existe significância suficiente para afirmar que a média do score na população
# são diferentes, e podemos observar que a média do score da empresa piedpiper é maior que o
```