

Tugas Besar Probabilitas dan Statistika IF2220

Contributors

1. Ditra Rizqa Amadia (13521019)
2. Kenny Benaya Nathan (13521023)

Setup

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import numpy as np
```

```
data = pd.read_csv("./dataset/anggur.csv")
```

Soal 1

Deskripsi Statistika (Descriptive Statistics) yang terdiri dari mean, median, modus, standar deviasi, variansi, range, nilai minimum, maksimum, kuartil, IQR, skewness dan kurtosis

- **Mean** atau **rataan** adalah indikator statistik yang dapat digunakan untuk mengukur rata-rata dari sebuah data.
- **Median** adalah nilai di tengah deret nilai yang disusun dengan data urut dari terkecil sampai terbesar.
- **Modus** adalah nilai data yang paling sering muncul atau nilai data yang memiliki frekuensi paling besar.
- **Standar deviasi** adalah nilai untuk menentukan seberapa dekat data dari suatu sampel statistik dengan mean data tersebut.
- **Variansi** didefinisikan sebagai nilai yang mendeskripsikan seberapa besar data tersebar dari nilai reratanya.
- **Range** adalah perbedaan antara nilai tertinggi dengan nilai terendah pada data.
- **Nilai minimum** adalah nilai terendah pada data.
- **Nilai maksimum** adalah nilai tertinggi pada data.
- **Kuartil** adalah nilai yang membagi data yang berurutan menjadi empat bagian yang sama banyak.
- **Interquartil** atau **IQR** adalah jangkauan kuartil sebagai selisih antara Q1 dengan Q3.
- **Skewness** atau **kemiringan** adalah ukuran simetri dari suatu fungsi distribusi peluang.
- **Kurtosis** atau **keruncingan** adalah ukuran lancip dari suatu fungsi distribusi peluang.

```
# Initiate data schema
```

```
desc = {}
desc[""] = []
```

```

desc["Mean"] = []
desc["Median"] = []
desc["Modus"] = []
desc["Std"] = []
desc["Variance"] = []
desc["Range"] = []
desc["Min"] = []
desc["Max"] = []
desc["Q1"] = []
desc["Q2"] = []
desc["Q3"] = []
desc["IQR"] = []
desc["Skewness"] = []
desc["Kurtosis"] = []

```

Insert data

```

for column in data:
    desc[""].append(column)
    desc["Mean"].append(data[column].mean())
    desc["Median"].append(data[column].median())
    desc["Modus"].append(data[column].mode("index")[0])
    desc["Std"].append(data[column].std())
    desc["Variance"].append(data[column].var())
    desc["Range"].append(data[column].max() - data[column].min())
    desc["Min"].append(data[column].min())
    desc["Max"].append(data[column].max())
    desc["Q1"].append(data[column].quantile(0.25))
    desc["Q2"].append(data[column].quantile(0.5))
    desc["Q3"].append(data[column].quantile(0.75))
    desc["IQR"].append(data[column].quantile(0.75) -
data[column].quantile(0.25))
    desc["Skewness"].append((3 * (data[column].mean() -
data[column].median()))/data[column].std())
    desc["Kurtosis"].append(data[column].kurtosis())

```

Show schema

```

descDataFrame = pd.DataFrame(data=desc)
descDataFrame

```

		Mean	Median	Modus	Std	
0	fixed acidity	7.152530	7.150000	6.540000	1.201598	\
1	volatile acidity	0.520839	0.524850	0.554600	0.095848	
2	citric acid	0.270517	0.272200	0.301900	0.049098	
3	residual sugar	2.567104	2.519430	0.032555	0.987915	
4	chlorides	0.081195	0.082167	0.015122	0.020111	
5	free sulfur dioxide	14.907679	14.860346	0.194679	4.888100	
6	total sulfur dioxide	40.290150	40.190000	35.200000	9.965767	
7	density	0.995925	0.996000	0.995900	0.002020	
8	pH	3.303610	3.300000	3.340000	0.104875	
9	sulphates	0.598390	0.595000	0.590000	0.100819	

10	alcohol	10.592280	10.610000	9.860000	1.510706
11	quality	7.958000	8.000000	8.000000	0.902802

	Variance	Range	Min	Max	Q1	Q2
0	1.443837	8.170000	3.320000	11.490000	6.377500	7.150000
1	0.009187	0.665200	0.139900	0.805100	0.456100	0.524850
2	0.002411	0.292900	0.116700	0.409600	0.237800	0.272200
3	0.975977	5.518200	0.032555	5.550755	1.896330	2.519430
4	0.000404	0.125635	0.015122	0.140758	0.066574	0.082167
5	23.893519	27.267847	0.194679	27.462525	11.426717	14.860346
6	99.316519	66.810000	3.150000	69.960000	33.785000	40.190000
7	0.000004	0.013800	0.988800	1.002600	0.994600	0.996000
8	0.010999	0.740000	2.970000	3.710000	3.230000	3.300000
9	0.010164	0.670000	0.290000	0.960000	0.530000	0.595000
10	2.282233	8.990000	6.030000	15.020000	9.560000	10.610000
11	0.815051	5.000000	5.000000	10.000000	7.000000	8.000000

	Q3	IQR	Skewness	Kurtosis
0	8.000000	1.622500	0.006317	-0.019292
1	0.585375	0.129275	-0.125558	0.161853
2	0.302325	0.064525	-0.102834	-0.104679
3	3.220873	1.324544	0.144770	-0.042980
4	0.095312	0.028738	-0.144960	-0.246508
5	18.313098	6.886381	0.029050	-0.364964
6	47.022500	13.237500	0.030148	0.063950
7	0.997200	0.002600	-0.110931	0.016366
8	3.370000	0.140000	0.103265	0.080910
9	0.670000	0.140000	0.100874	0.064819
10	11.622500	2.062500	-0.035189	-0.131732
11	9.000000	2.000000	-0.139566	0.108291

Soal 2

Visualisasi plot distribusi, dalam bentuk histogram dan boxplot

Fixed Acidity

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
fixedAcidity = data[['fixed acidity']]
```

Create figure

```
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))
```

Histogram

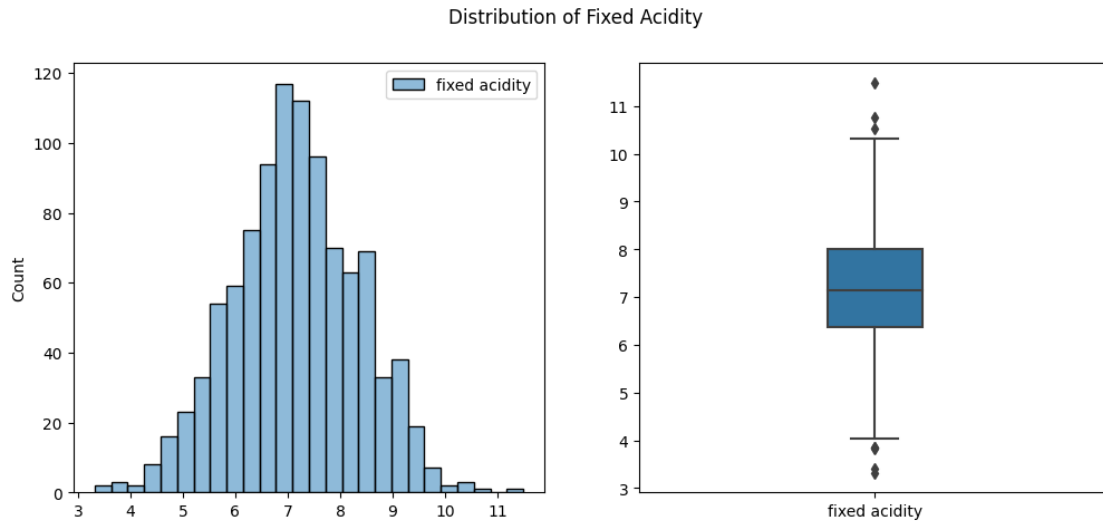
```
sns.histplot(fixedAcidity, ax=axs[0])
```

Boxplot

```
sns.boxplot(fixedAcidity, ax=axs[1], orient='v', width=0.2)
```

```
plt.suptitle("Distribution of Fixed Acidity")
plt.show

<function matplotlib.pyplot.show(close=None, block=None)>
```



Volatile Acidity

Pada histogram, distribusi terlihat membentuk puncak namun tidak simetris. Dari boxplot terlihat terdapat 2 outlier. Sehingga distribusi tidak bersifat normal secara grafis.

```
volatileAcidity = data[['volatile acidity']]

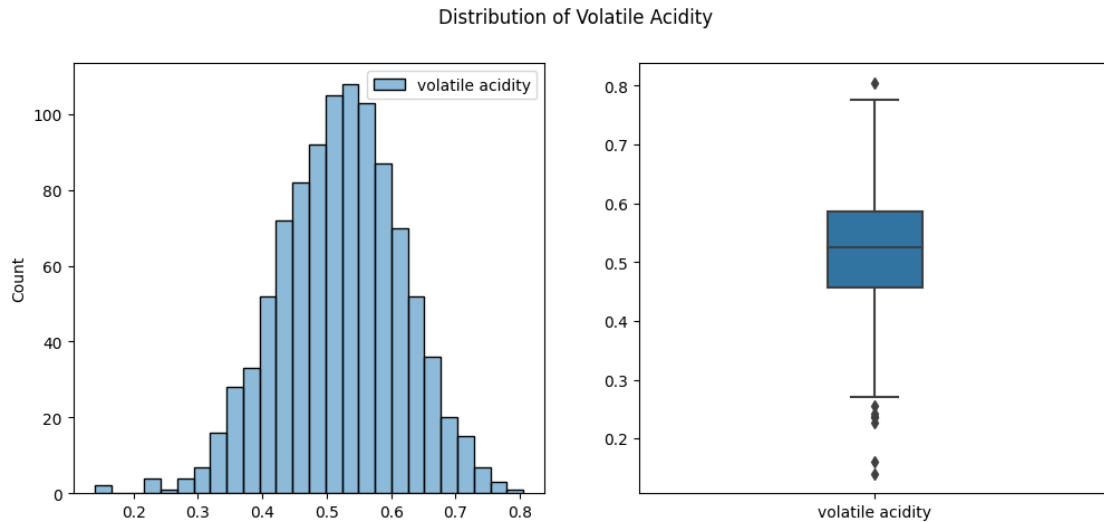
# Create figure
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))

# Histogram
sns.histplot(volatileAcidity, ax=axs[0])

# Boxplot
sns.boxplot(volatileAcidity, ax=axs[1], orient='v', width=0.2)

plt.suptitle("Distribution of Volatile Acidity")
plt.show

<function matplotlib.pyplot.show(close=None, block=None)>
```



Citric Acid

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
citricAcid = data[['citric acid']]

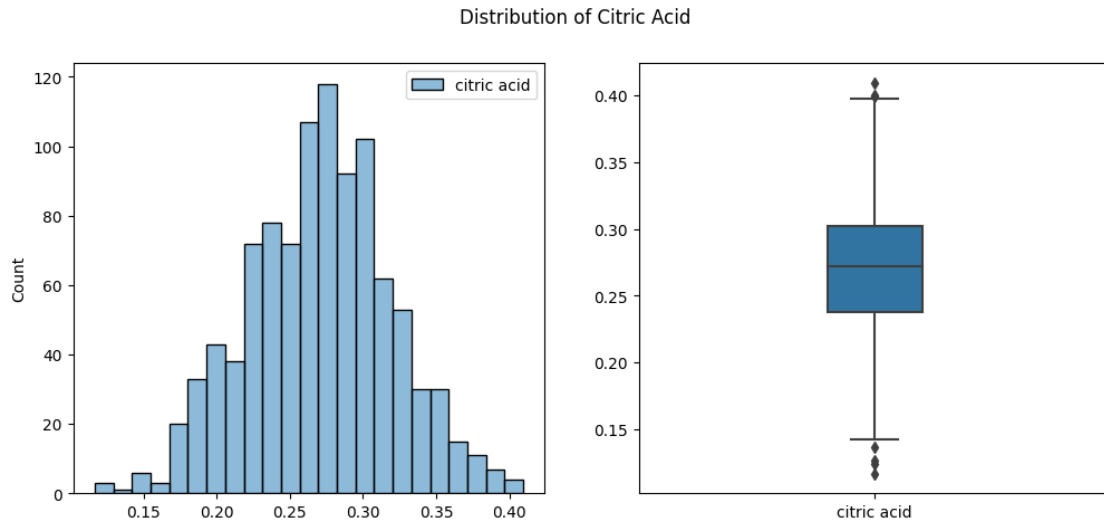
# Create figure
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))

# Histogram
sns.histplot(citricAcid, ax=axs[0])

# Boxplot
sns.boxplot(citricAcid, ax=axs[1], orient='v', width=0.2)

plt.suptitle("Distribution of Citric Acid")
plt.show

<function matplotlib.pyplot.show(close=None, block=None)>
```



Residual Sugar

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
residualSugar = data[['residual sugar']]
```

```
# Create figure
```

```
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# Histogram
```

```
sns.histplot(residualSugar, ax=axs[0])
```

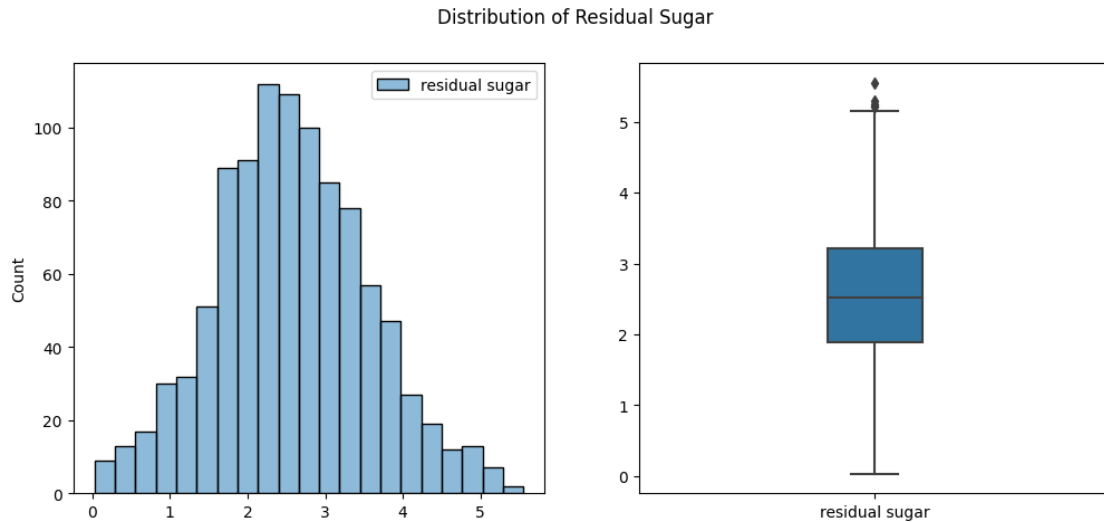
```
# Boxplot
```

```
sns.boxplot(residualSugar, ax=axs[1], orient='v', width=0.2)
```

```
plt.suptitle("Distribution of Residual Sugar")
```

```
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Chlorides

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
chlorides = data[['chlorides']]
```

```
# Create figure
```

```
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# Histogram
```

```
sns.histplot(chlorides, ax=axs[0])
```

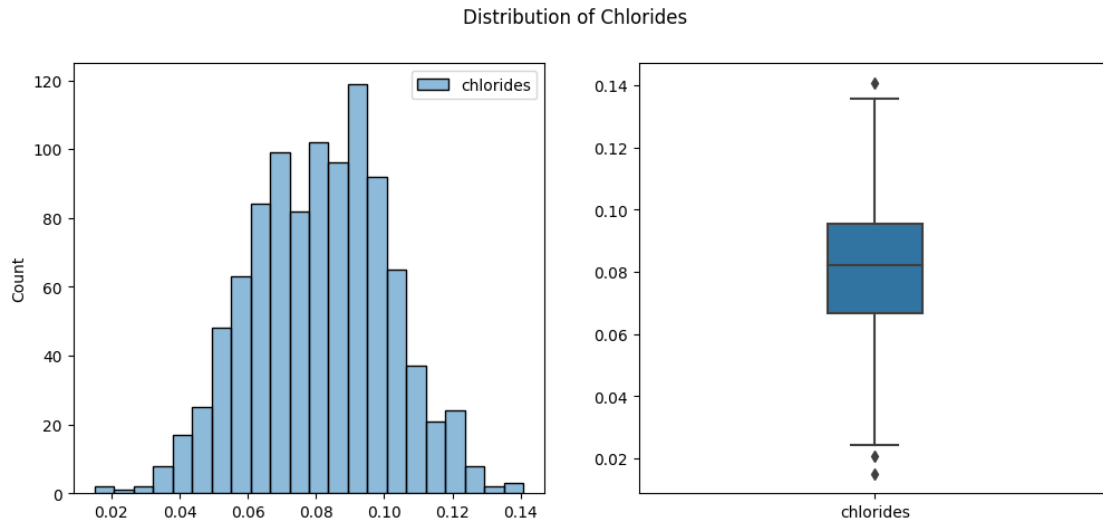
```
# Boxplot
```

```
sns.boxplot(chlorides, ax=axs[1], orient='v', width=0.2)
```

```
plt.suptitle("Distribution of Chlorides")
```

```
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Free Sulfur Dioxide

Pada histogram, distribusi terlihat membentuk puncak namun tidak simetris. Pada boxplot terlihat terdapat outlier. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat bukan normal secara grafis.

```
freeSulfurDioxide = data[['free sulfur dioxide']]

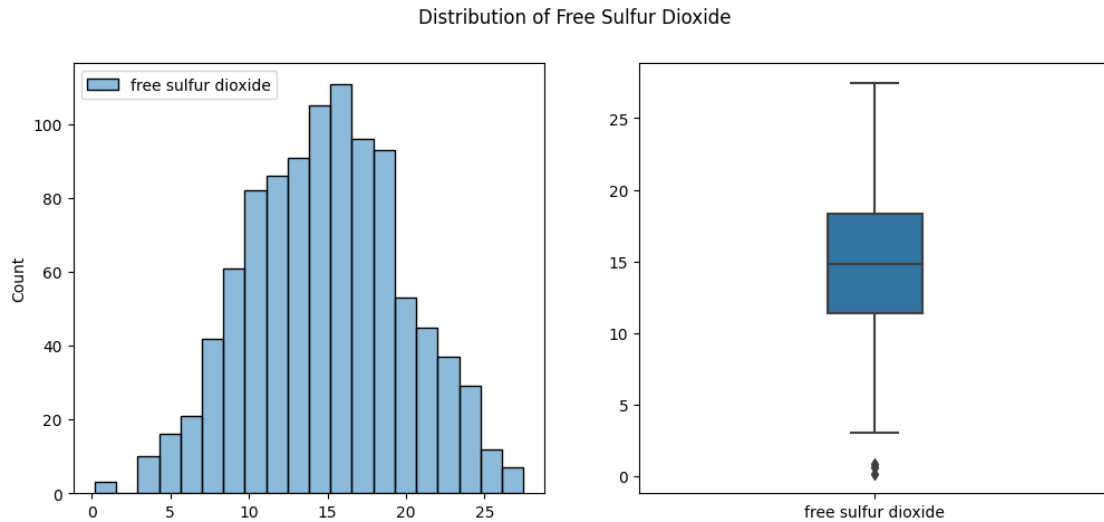
# Create figure
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))

# Histogram
sns.histplot(freeSulfurDioxide, ax=axs[0])

# Boxplot
sns.boxplot(freeSulfurDioxide, ax=axs[1], orient='v', width=0.2)

plt.suptitle("Distribution of Free Sulfur Dioxide")
plt.show

<function matplotlib.pyplot.show(close=None, block=None)>
```

Total Sulfur Dioxide

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
totalSulfurDioxide = data[['total sulfur dioxide']]
```

```
# Create figure
```

```
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# Histogram
```

```
sns.histplot(totalSulfurDioxide, ax=axs[0])
```

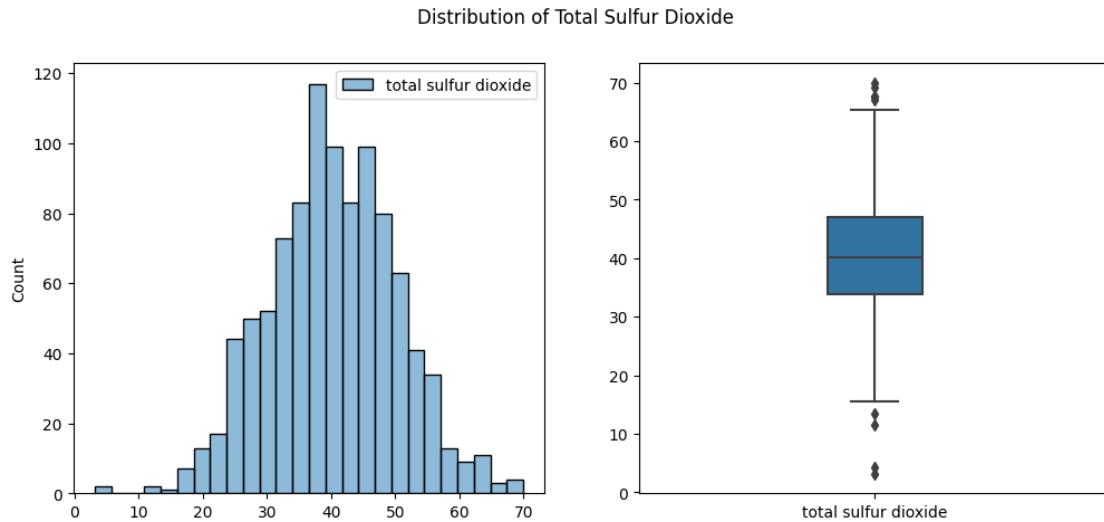
```
# Boxplot
```

```
sns.boxplot(totalSulfurDioxide, ax=axs[1], orient='v', width=0.2)
```

```
plt.suptitle("Distribution of Total Sulfur Dioxide")
```

```
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Density

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
density = data[['density']]
```

```
# Create figure
```

```
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# Histogram
```

```
sns.histplot(density, ax=axs[0])
```

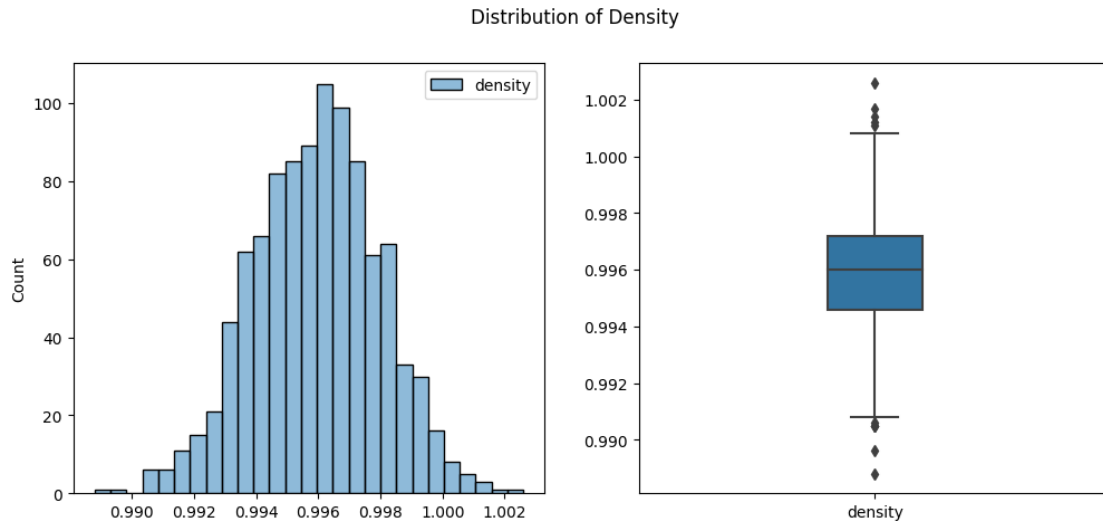
```
# Boxplot
```

```
sns.boxplot(density, ax=axs[1], orient='v', width=0.2)
```

```
plt.suptitle("Distribution of Density")
```

```
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



pH

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
pH = data[['pH']]
```

```
# Create figure
```

```
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# Histogram
```

```
sns.histplot(pH, ax=axs[0])
```

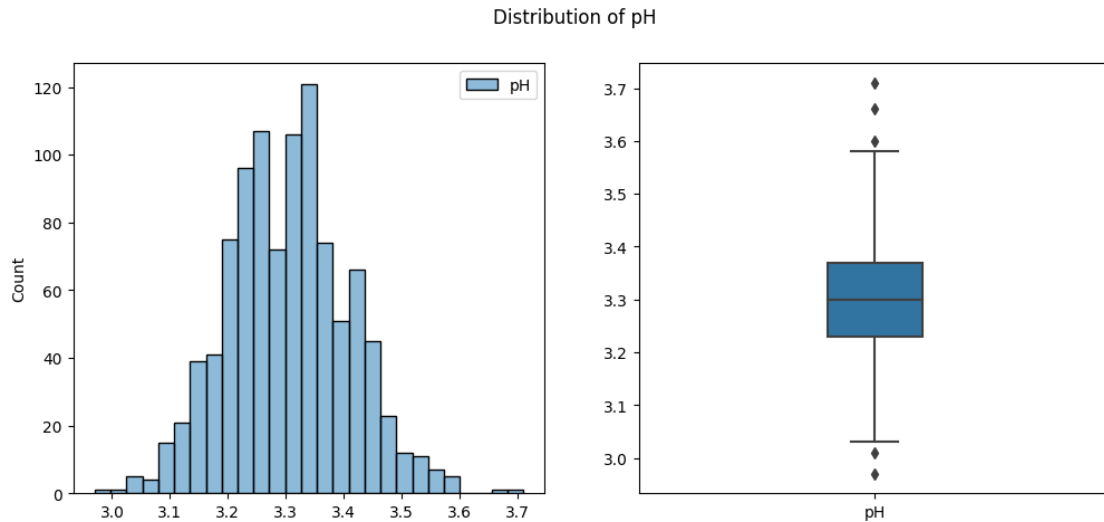
```
# Boxplot
```

```
sns.boxplot(pH, ax=axs[1], orient='v', width=0.2)
```

```
plt.suptitle("Distribution of pH")
```

```
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Sulphates

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
sulphates = data[['sulphates']]
```

```
# Create figure
```

```
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# Histogram
```

```
sns.histplot(sulphates, ax=axs[0])
```

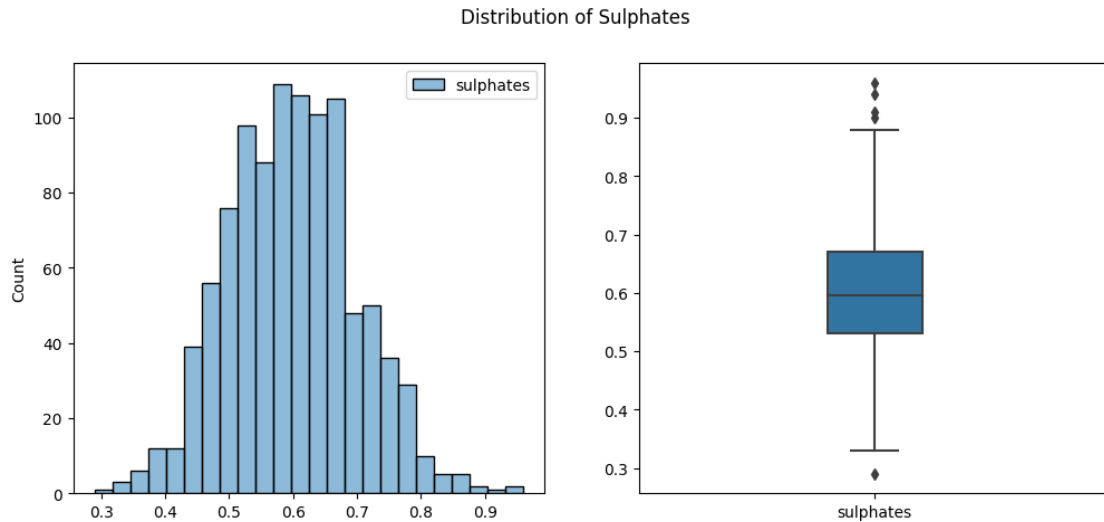
```
# Boxplot
```

```
sns.boxplot(sulphates, ax=axs[1], orient='v', width=0.2)
```

```
plt.suptitle("Distribution of Sulphates")
```

```
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Alcohol

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
alcohol = data[['alcohol']]
```

```
# Create figure
```

```
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# Histogram
```

```
sns.histplot(alcohol, ax=axs[0])
```

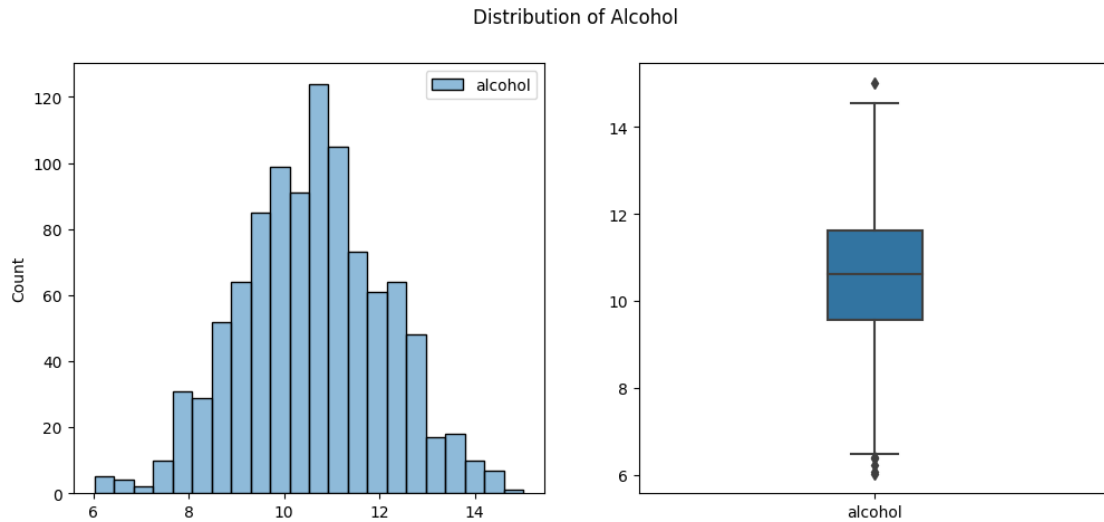
```
# Boxplot
```

```
sns.boxplot(alcohol, ax=axs[1], orient='v', width=0.2)
```

```
plt.suptitle("Distribution of Alcohol")
```

```
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Quality

Pada histogram, distribusi terlihat membentuk puncak simetris. Pada boxplot juga terlihat garis Q2 berada di tengah IQR. Sehingga distribusi bersifat normal secara grafis.

```
quality = data[['quality']]
```

```
# Create figure
```

```
fig, axs = plt.subplots(ncols=2, figsize=(12, 5))
```

```
# Histogram
```

```
sns.histplot(quality, ax=axs[0])
```

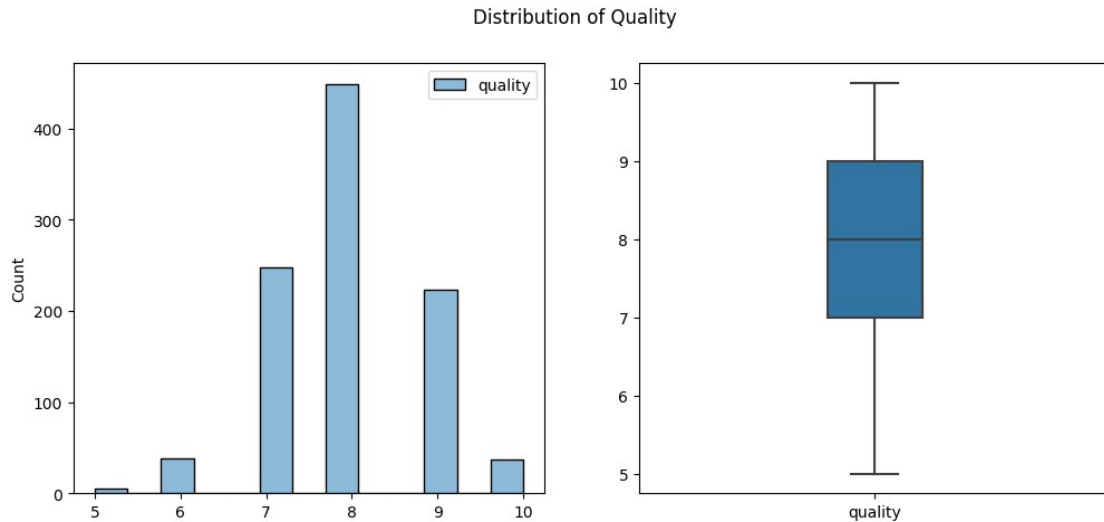
```
# Boxplot
```

```
sns.boxplot(quality, ax=axs[1], orient='v', width=0.2)
```

```
plt.suptitle("Distribution of Quality")
```

```
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Soal 3

Menentukan setiap kolom numerik berdistribusi normal atau tidak. Cara menentukan suatu distribusi normal atau tidak dapat menggunakan 2 cara. Cara pertama yaitu cara graphical yaitu membandingkan sampel data dengan garis normal pada histogram. Cara kedua yaitu cara analytical yaitu mencari p-value menggunakan normality test seperti Kolmogorov–Smirnov, Shapiro–Wilk, dan D’Agostino–Pearson tests. Apabila p-value > alpha maka suatu distribusi bersifat normal. Alpha yang sering digunakan pada normality test yaitu 0.05.

Fixed Acidity

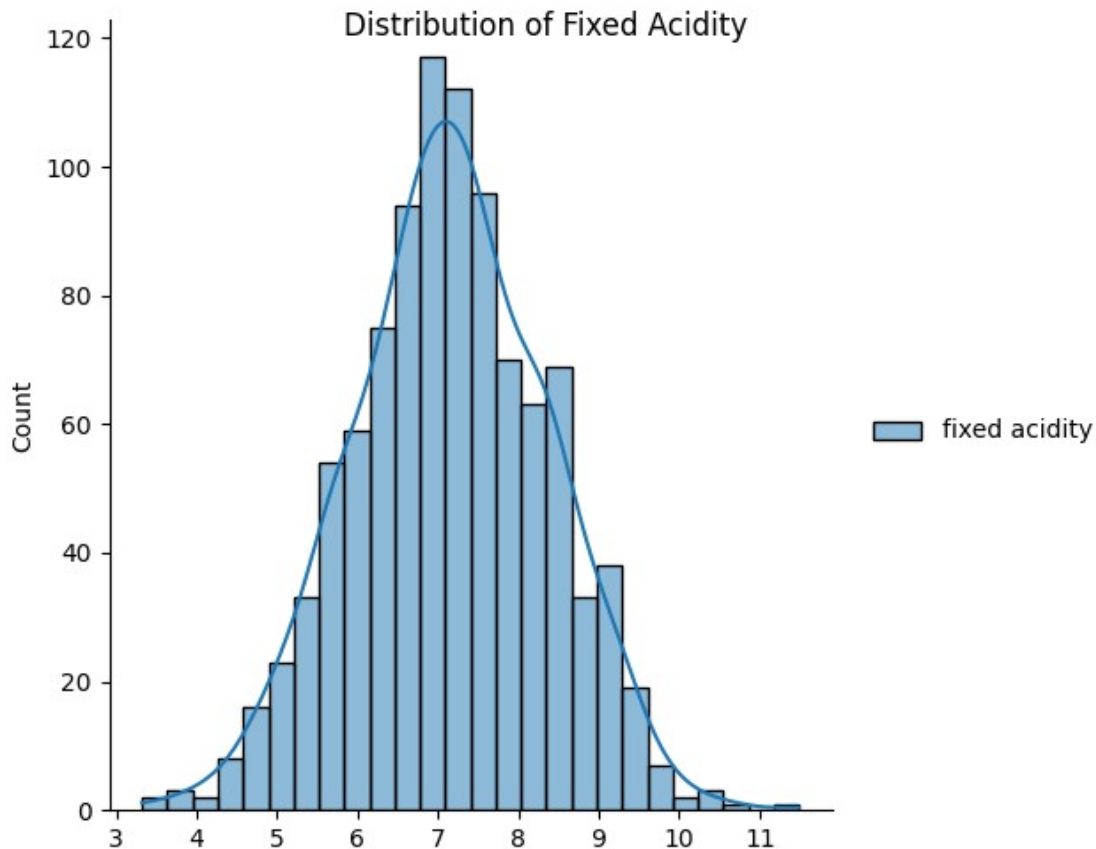
Distribusi Fixed Acidity bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi Fixed Acidity juga bersifat normal secara analytical dengan p-value > 0.05.

```
sns.displot(fixedAcidity, kde=True, ax=axis[0])
plt.suptitle("Distribution of Fixed Acidity")
plt.show
```

```
c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-
packages\seaborn\distributions.py:2142: UserWarning: `displot` is a
figure-level function and does not accept the ax= parameter. You may
wish to try histplot.
```

```
warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['fixed acidity'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")

print(f"Skewness pada distribusi = {(3 * (data['fixed acidity'].mean()
- data['fixed acidity'].median()))/data['fixed acidity'].std()}")
```

Distribusi Normal dengan p-value = 0.9308584274486692
 Skewness pada distribusi = 0.006316590636063725

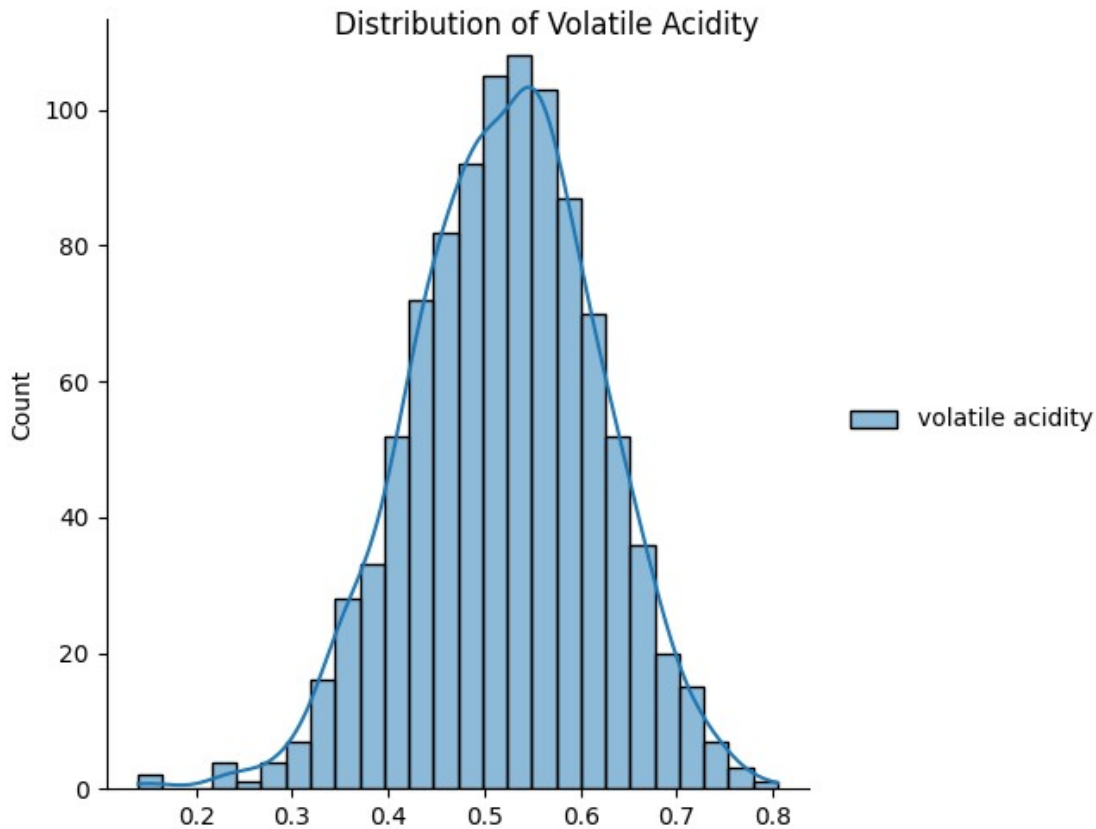
Volatile Acidity

Distribusi Volatile Acidity tidak bersifat normal secara graphical. Distribusi pada histogram membentuk puncak namun tidak simetris. Distribusi Volatile Acidity juga tidak bersifat normal secara analytical dengan p-value < 0.05. Distribusi ini menggambarkan skewness negatif.

```
sns.displot(volatileAcidity, kde=True)
plt.suptitle("Distribution of Volatile Acidity")
plt.show
```



```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['volatile acidity'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")

print(f"Skewness pada distribusi = {(3 * (data['volatile
acidity'].mean() - data['volatile acidity'].median()))/data['volatile
acidity'].std()}")
```

Bukan Distribusi Normal dengan p-value = 0.022581461594113835
 Skewness pada distribusi = -0.1255578164407037

Citric Acid

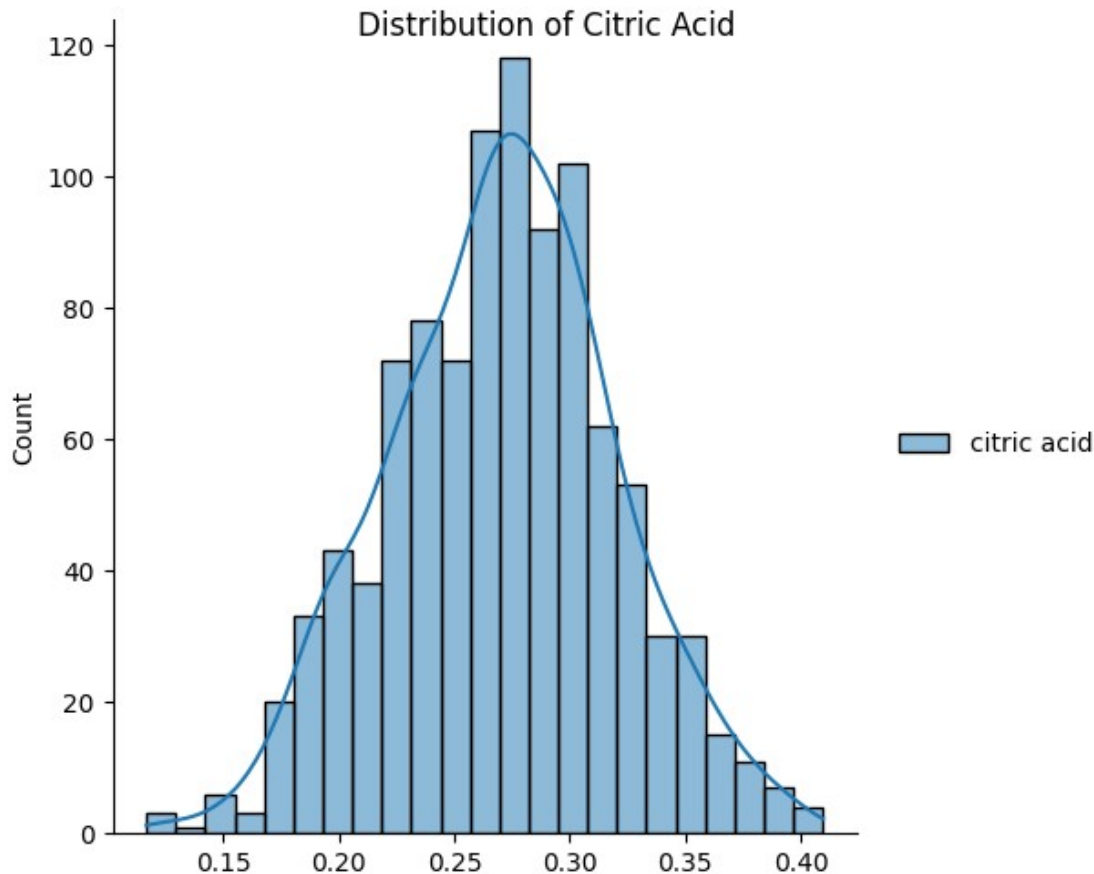
Distribusi Citric Acid bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi Citric Acid juga bersifat normal secara analytical dengan p-value > 0.05.

```
sns.displot(citricAcid, kde=True, ax=axes[0])
plt.suptitle("Distribution of Citric Acid")
plt.show
```

```
c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
```

```
warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['citric acid'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")

print(f"Skewness pada distribusi = {(3 * (data['citric acid'].mean() -
data['citric acid'].median()))/data['citric acid'].std()}")
```

```
Distribusi Normal dengan p-value = 0.6816899375976969
Skewness pada distribusi = -0.10283436799948968
```

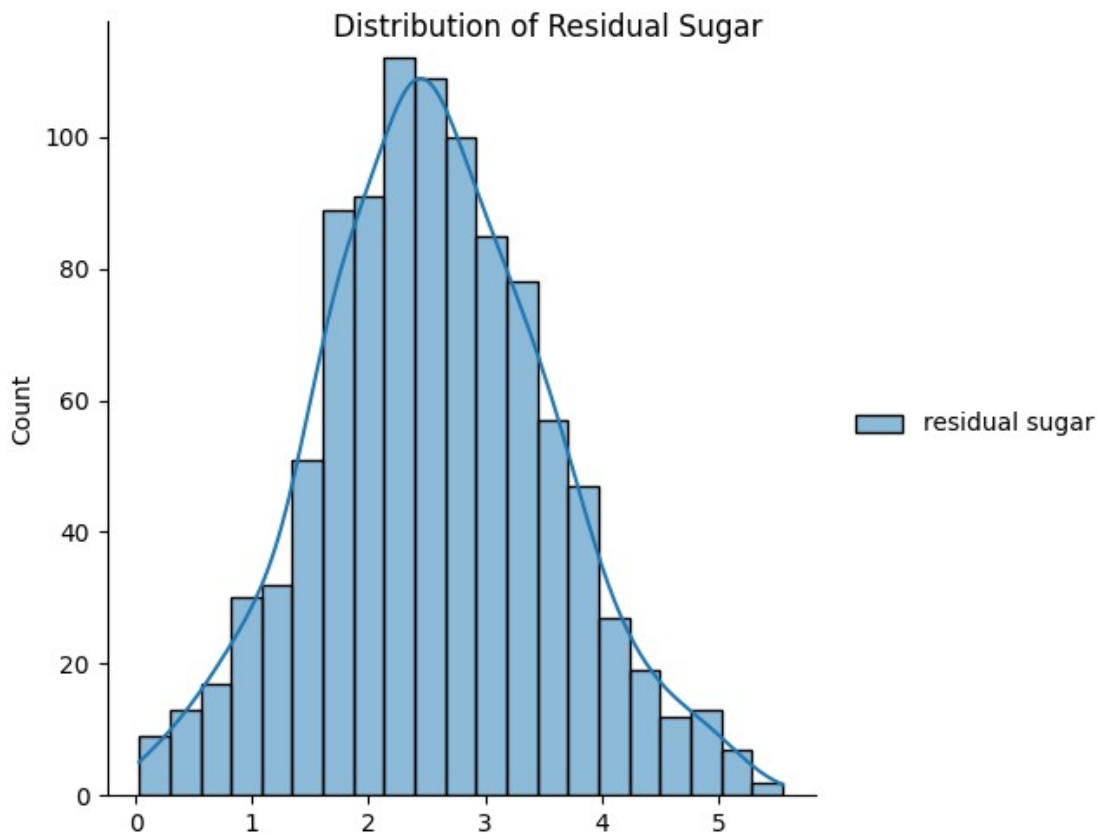
Residual Sugar

Distribusi Residual Sugar bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi Residual Sugar juga bersifat normal secara analytical dengan p-value > 0.05.

```
sns.displot(residualSugar, kde=True, ax=axes[0])
plt.suptitle("Distribution of Residual Sugar")
plt.show
```

```
c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-
packages\seaborn\distributions.py:2142: UserWarning: `displot` is a
figure-level function and does not accept the ax= parameter. You may
wish to try histplot.
  warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['residual sugar'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")
```

```
print(f"Skewness pada distribusi = {(3 * (data['residual
sugar'].mean() - data['residual sugar'].median()))/data['residual
sugar'].std()})")
```

Distribusi Normal dengan p-value = 0.2246670332131056
 Skewness pada distribusi = 0.1447697076471561

Chlorides

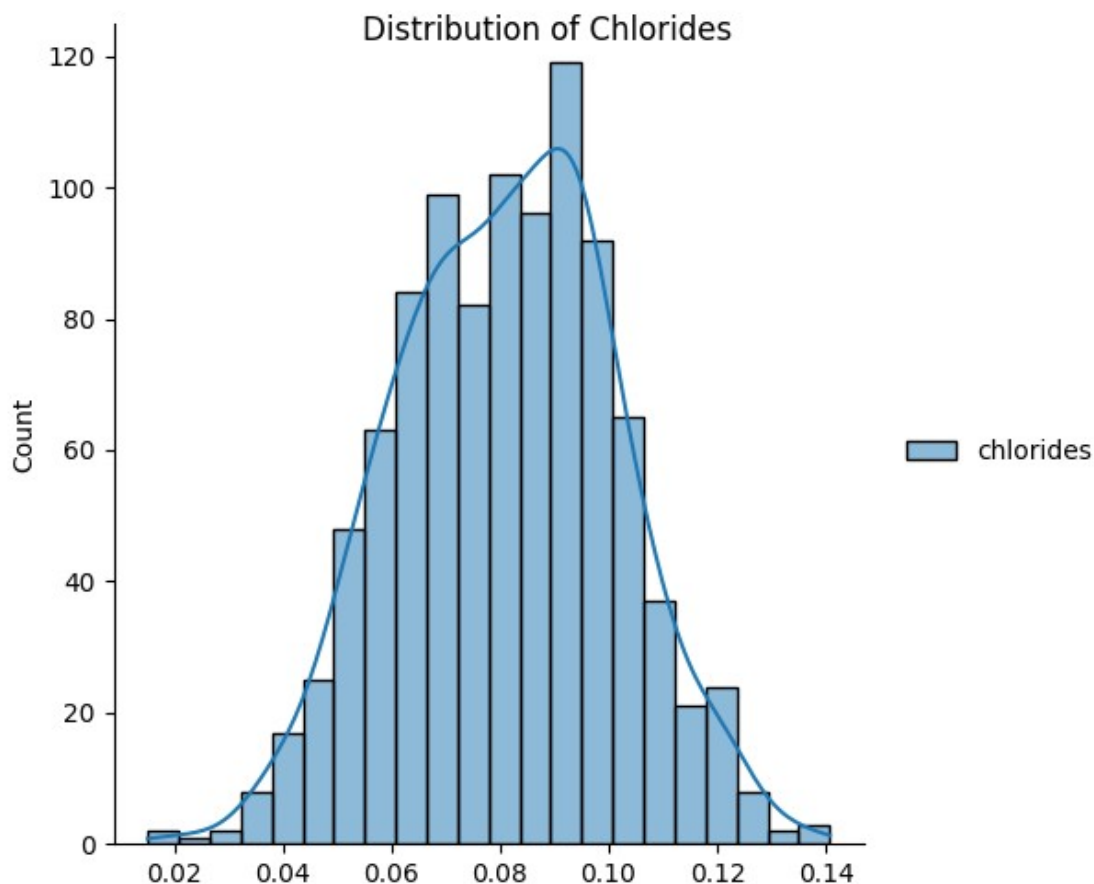
Distribusi Chlorides bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi Chlorides juga bersifat normal secara analytical dengan p-value > 0.05.

```
sns.displot(chlorides, kde=True, ax=axes[0])
plt.suptitle("Distribution of Chlorides")
plt.show
```

c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.

```
warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```

stat, p = stats.normaltest(data['chlorides'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")
print(f"Skewness pada distribusi = {(3 * (data['chlorides'].mean() -
data['chlorides'].median()))/data['chlorides'].std()}")

```

Distribusi Normal dengan p-value = 0.17048274704296862
 Skewness pada distribusi = -0.14496047465064746

Free Sulfur Dioxide

Distribusi Free Sulfur Dioxide bersifat normal secara graphical. Distribusi pada histogram membentuk kurva yang cukup simetris dengan puncak berada di tengah distribusi. Namun, distribusi Fixed Acidity tidak bersifat normal secara analytical dengan p-value < 0.05. Distribusi ini menggambarkan skewness positif.

```

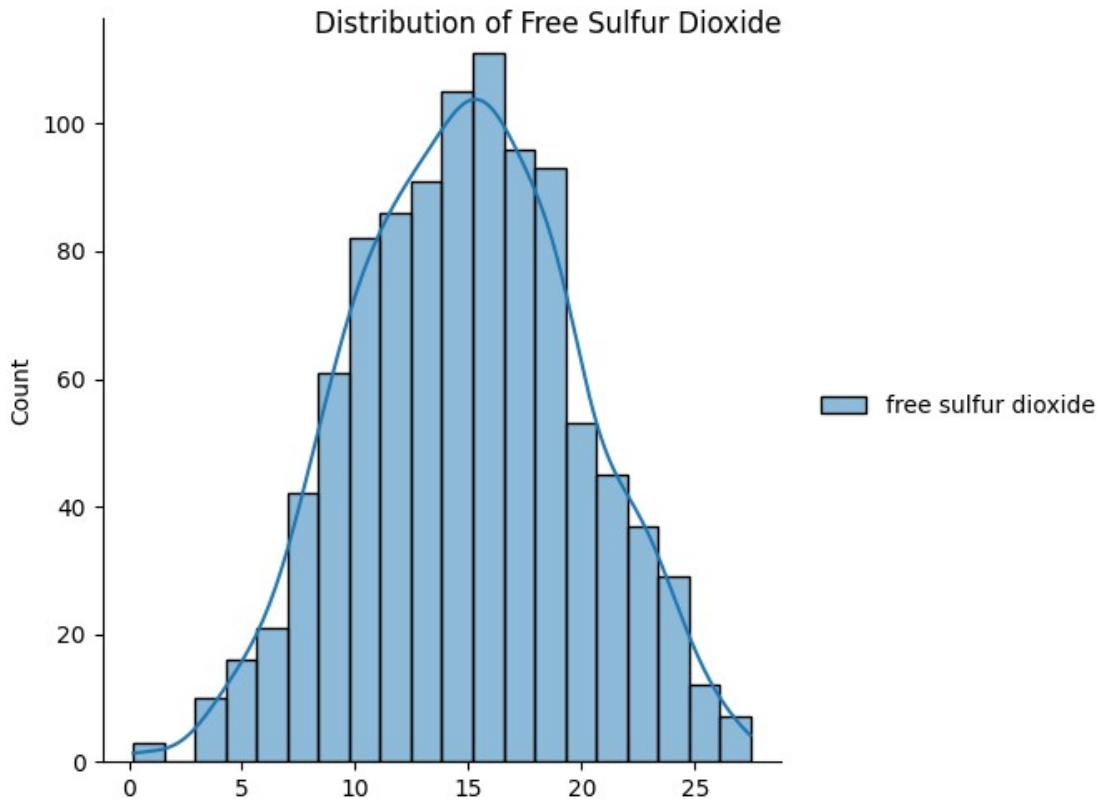
sns.displot(freeSulfurDioxide, kde=True, ax=axes[0])
plt.suptitle("Distribution of Free Sulfur Dioxide")
plt.show

```

c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.

```
warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['free sulfur dioxide'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")
print(f"Skewness pada distribusi = {(3 * (data['free sulfur dioxide'].mean() - data['free sulfur dioxide'].median()))/data['free sulfur dioxide'].std()}")
```

Bukan Distribusi Normal dengan p-value = 0.01743043451827735
 Skewness pada distribusi = 0.02904994822740363

Total Sulfur Dioxide

Distribusi Total Sulfur Dioxide bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi Total Sulfur Dioxide juga bersifat normal secara analytical dengan p-value > 0.05.

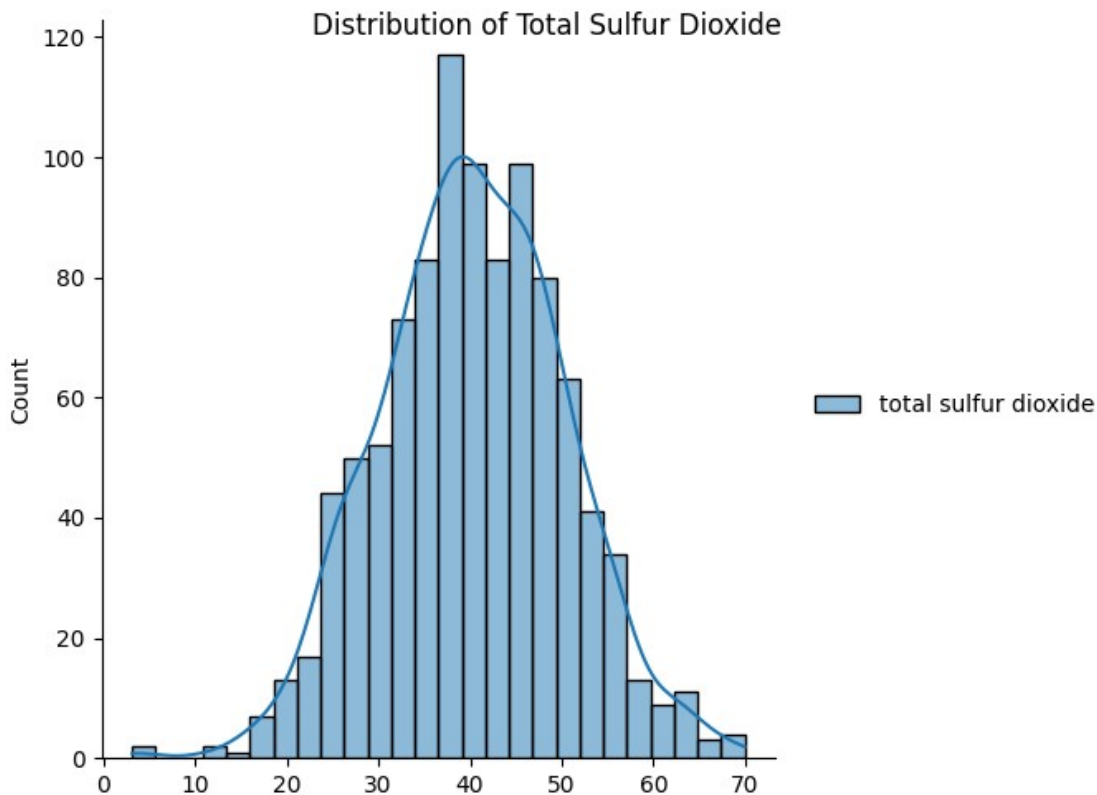
```
sns.displot(totalSulfurDioxide, kde=True, ax=axes[0])
plt.suptitle("Distribution of Total Sulfur Dioxide")
plt.show
```

c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may

wish to try histplot.

```
warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['total sulfur dioxide'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")
print(f"Skewness pada distribusi = {(3 * (data['total sulfur
dioxide'].mean() - data['total sulfur dioxide'].median()))/data['total
sulfur dioxide'].std()}")
```

Distribusi Normal dengan p-value = 0.8488846101395726

Skewness pada distribusi = 0.030148205216689577

Density

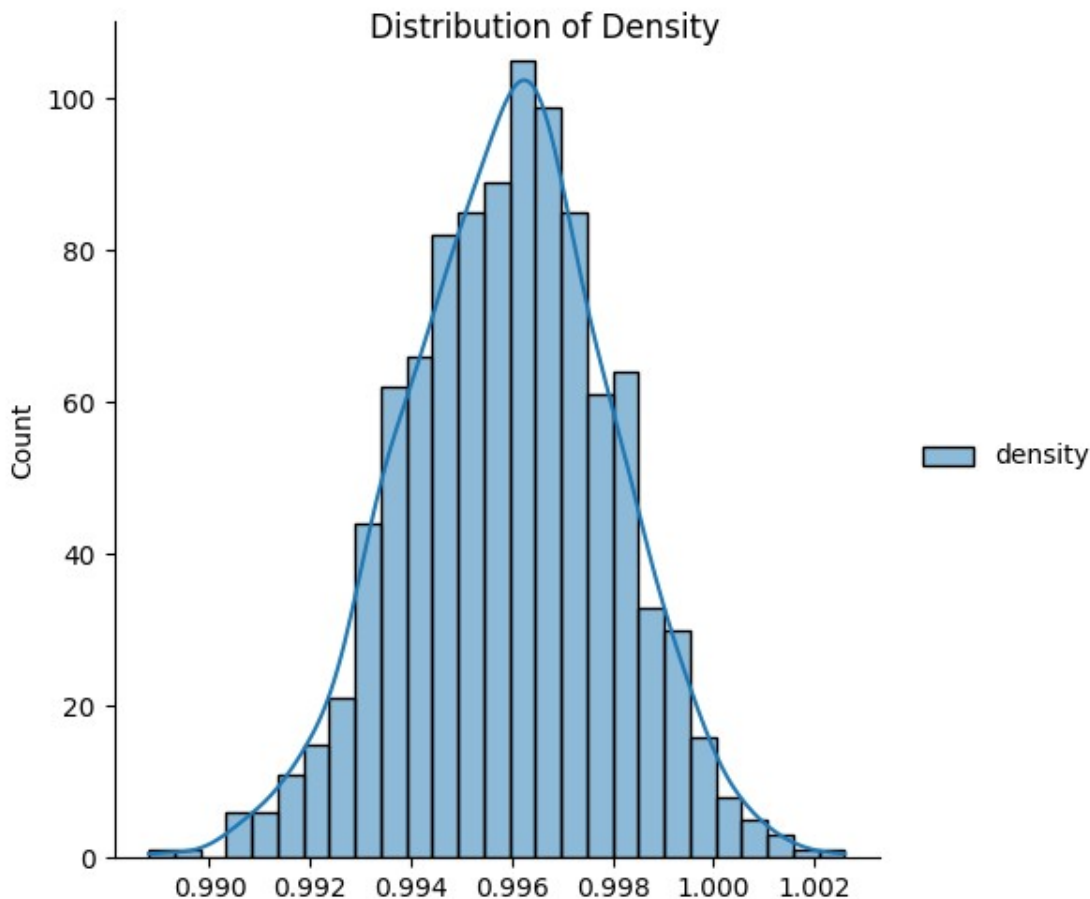
Distribusi Density bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi Density juga bersifat normal secara analytical dengan p-value > 0.05.

```
sns.displot(density, kde=True, ax=axes[0])
plt.suptitle("Distribution of Density")
plt.show
```

```
c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
```

```
warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['density'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")
print(f"Skewness pada distribusi = {(3 * (data['density'].mean() - data['density'].median()))/data['density'].std()}")
```

```
Distribusi Normal dengan p-value = 0.5985227325531981
Skewness pada distribusi = -0.11093065738249801
```

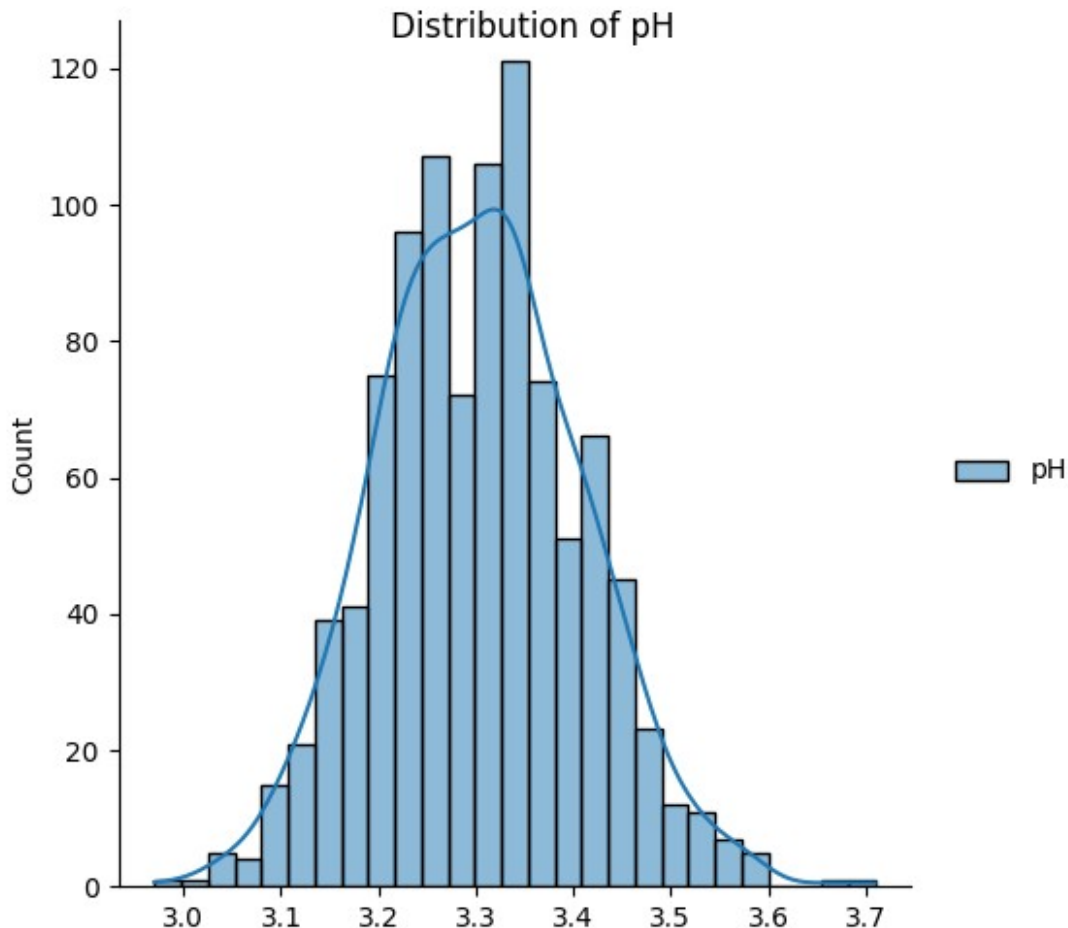

pH

Distribusi pH bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi pH juga bersifat normal secara analytical dengan p-value > 0.05.

```
sns.displot(pH, kde=True, ax=axes[0])  
plt.suptitle("Distribution of pH")  
plt.show
```

```
c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-  
packages\seaborn\distributions.py:2142: UserWarning: `displot` is a  
figure-level function and does not accept the ax= parameter. You may  
wish to try histplot.  
warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['pH'])  
if(p > 0.05):  
    print("Distribusi Normal", end='')  
else:
```

```

    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")
print(f"Skewness pada distribusi = {(3 * (data['pH'].mean() -
data['pH'].median()))/data['pH'].std()}")

```

Distribusi Normal dengan p-value = 0.13678740824860436

Skewness pada distribusi = 0.10326531781094268

Sulphates

Distribusi Sulphates bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi Sulphates juga bersifat normal secara analytical dengan p-value > 0.05.

```

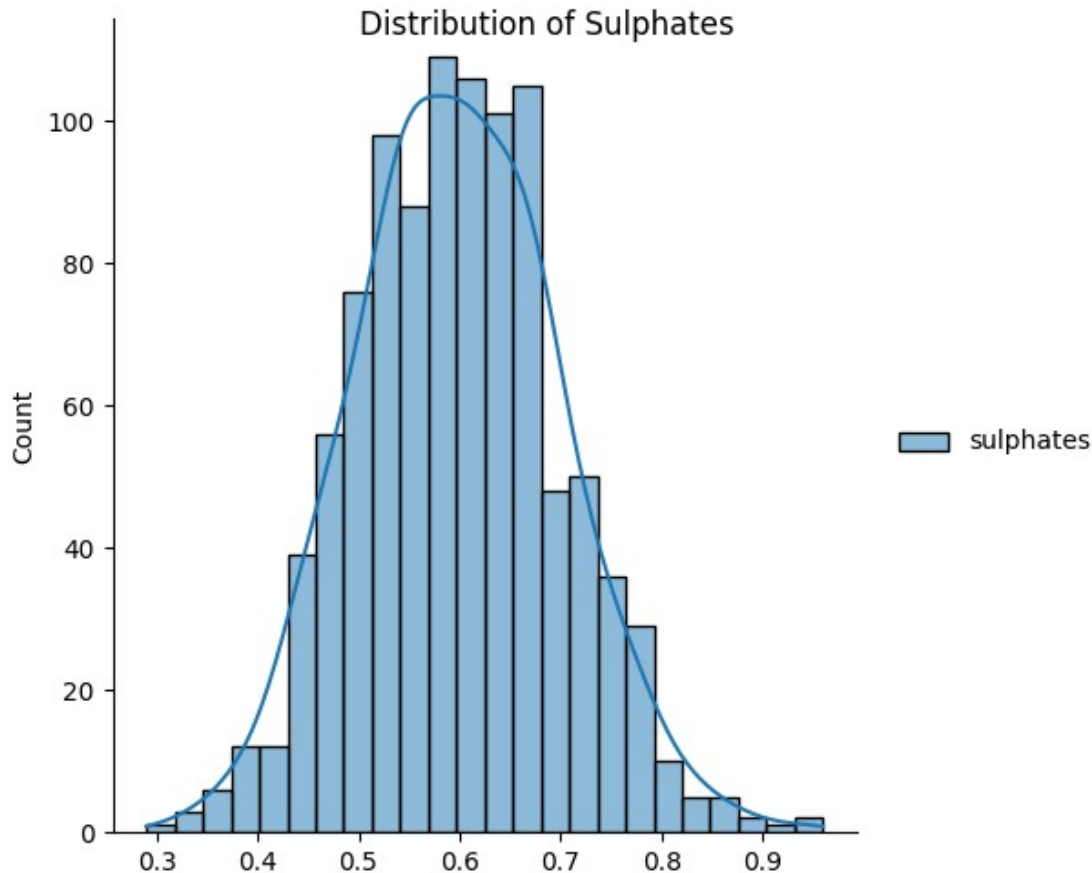
sns.displot(sulphates, kde=True, ax=axis[0])
plt.suptitle("Distribution of Sulphates")
plt.show

```

c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.

```
warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['pH'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")
print(f"Skewness pada distribusi = {(3 * (data['pH'].mean() -
data['pH'].median()))/data['pH'].std()}")
```

Distribusi Normal dengan p-value = 0.13678740824860436
 Skewness pada distribusi = 0.10326531781094268

Alcohol

Distribusi Alcohol bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi Alcohol juga bersifat normal secara analytical dengan p-value > 0.05.

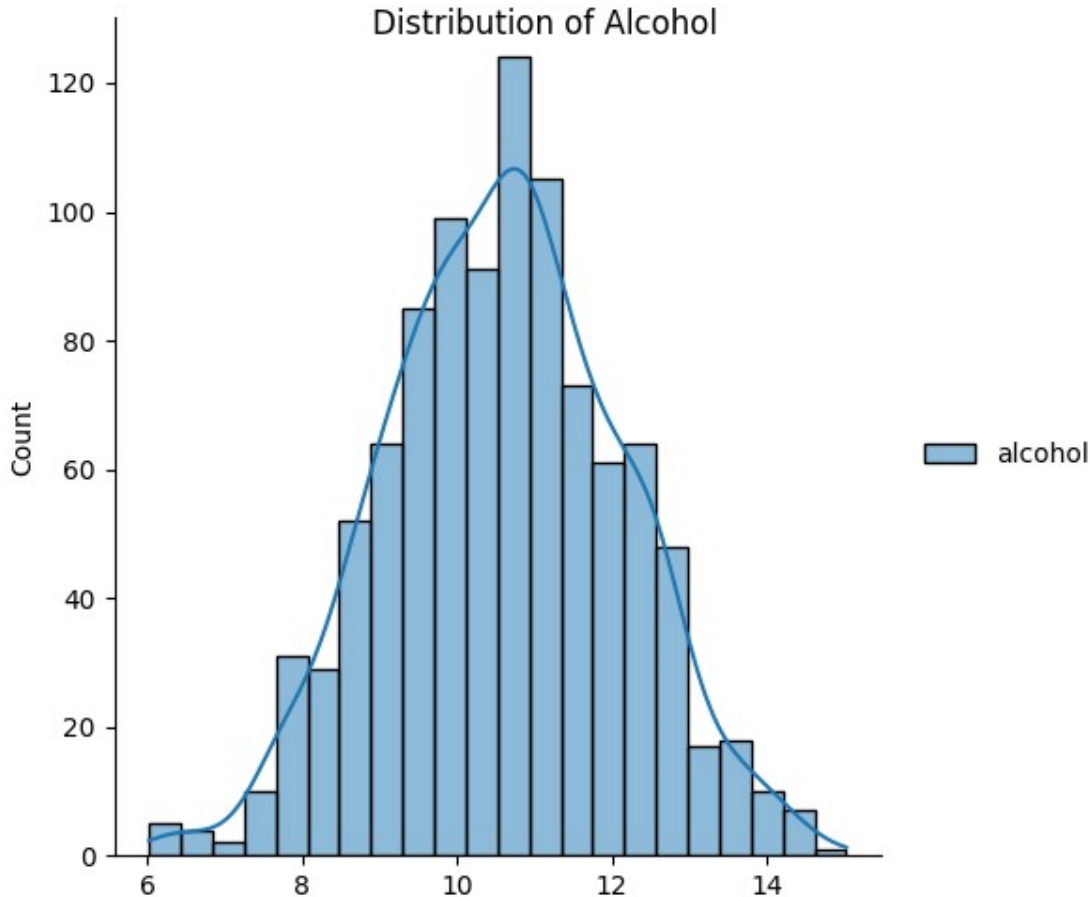
```
sns.displot(alcohol, kde=True, ax=axes[0])
plt.suptitle("Distribution of Alcohol")
plt.show
```

c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a

figure-level function and does not accept the ax= parameter. You may wish to try histplot.

```
warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['alcohol'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")
print(f"Skewness pada distribusi = {(3 * (data['alcohol'].mean() -
data['alcohol'].median()))/data['alcohol'].std()}")
```

Distribusi Normal dengan p-value = 0.6790884901361043

Skewness pada distribusi = -0.03518884535840057

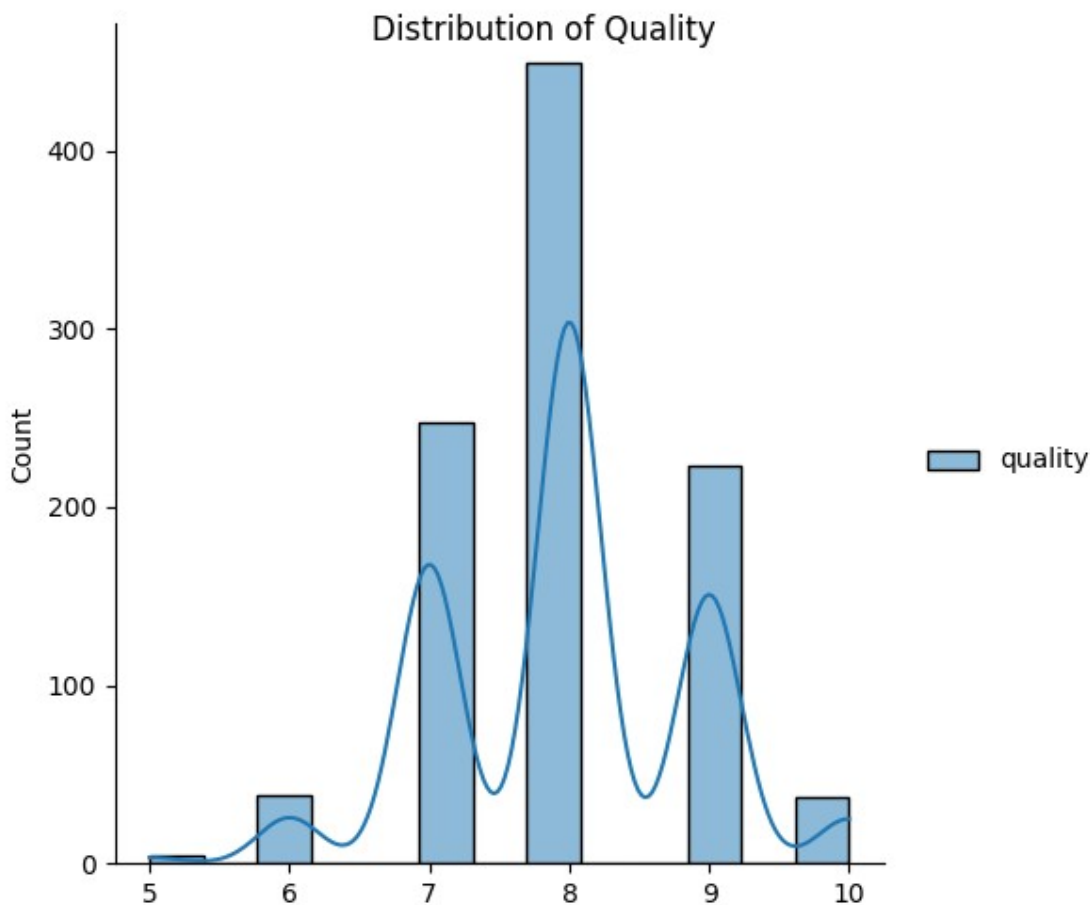
Quality

Distribusi Quality bersifat normal secara graphical. Distribusi pada histogram membentuk kurva simetris dengan puncak berada di tengah distribusi. Distribusi Quality juga bersifat normal secara analytical dengan p-value > 0.05.

```
sns.displot(quality, kde=True, ax=axes[0])
plt.suptitle("Distribution of Quality")
plt.show
```

```
c:\Users\Ditra\AppData\Local\Programs\Python\Python311\Lib\site-
packages\seaborn\distributions.py:2142: UserWarning: `displot` is a
figure-level function and does not accept the ax= parameter. You may
wish to try histplot.
  warnings.warn(msg, UserWarning)
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
stat, p = stats.normaltest(data['quality'])
if(p > 0.05):
    print("Distribusi Normal", end='')
else:
    print("Bukan Distribusi Normal", end='')
print(f" dengan p-value = {p}")
print(f"Skewness pada distribusi = {(3 * (data['quality'].mean() -
data['quality'].median()))/data['quality'].std()}")
```

Distribusi Normal dengan p-value = 0.3888139394184818
Skewness pada distribusi = -0.13956552038002426

Soal 4

Tes hipotesis 1 sampel

Soal 4.a

Nilai rata-rata pH di atas 3.29?

```
# === Langkah 1
```

```
=====
```

```
=====
```

```
# Tentukan Hipotesis 0 (H0)
```

```
# H0: pH = 3.29
```

```
# === Langkah 2
```

```
=====
```

```
=====
```

```
# Tentukan Hipotesis Alternatif (H1)
```

```
# H1: pH > 3.29
```

```
# === Langkah 3
```

```
=====
```

```
=====
```

```
# Tentukan tingkat signifikan alpha yaitu 5%
```

```
alpha = 0.05
```

```
# === Langkah 4
```

```
=====
```

```
=====
```

```
# Tentukan daerah kritis one-tailed
```

```
z_crit = 1.645
```

```
# === Langkah 5
```

```
=====
```

```
=====
```

```
# Hitung nilai t-value dan p-value
```

```
z_value = (data['pH'].mean() - 3.29) / (data['pH'].std() /
```

```
np.sqrt(data['pH'].count()))
```

```
p_value = 1 - stats.norm.cdf(z_value)
```

```
# === Langkah 6
```

```
=====
```

```

=====
# Ambil keputusan berdasarkan z-value dengan z-crit dan p-value dengan
alpha
print('Keputusan berdasarkan z-value:')
print(f'z-value = {z_value}')
print(f'z-crit = {z_crit}')
if z_value > z_crit:
    print('H0 ditolak')
else:
    print('H0 diterima')

print('')

print('Keputusan berdasarkan p-value:')
print(f'p-value = {p_value}')
print(f'alpha = {alpha}')
if p_value < alpha:
    print('H0 ditolak')
else:
    print('H0 diterima')

Keputusan berdasarkan z-value:
z-value = 4.1037807933651145
z-crit = 1.645
H0 ditolak

Keputusan berdasarkan p-value:
p-value = 2.032263004325685e-05
alpha = 0.05
H0 ditolak

```

Soal 4.b

Nilai rata-rata residual sugar tidak sama dengan 2.5?

=== Langkah 1

```

=====
=====

```

```

# Tentukan Hipotesis 0 (H0)
# H0: Residual Sugar = 2.5

```

=== Langkah 2

```

=====
=====

```

```

# Tentukan Hipotesis Alternatif (H1)
# H1: Residual Sugar /= 2.5

```

=== Langkah 3

```

=====
=====
# Tentukan tingkat signifikan alpha yaitu 5%
alpha = 0.05

# === Langkah 4
=====
=====
# Tentukan daerah kritis two-tailed
z_crit_r = 1.96
z_crit_l = -1.96

# === Langkah 5
=====
=====
# Hitung nilai z-value dan p-value
z_value = (data['residual sugar'].mean() - 2.5) / (data['residual
sugar'].std() / np.sqrt(data['residual sugar'].count()))
p_value = 2 * (1 - stats.norm.cdf(np.abs(z_value)))

# === Langkah 6
=====
=====
# Ambil keputusan berdasarkan z-value dengan z_crit_r dan z_crit_l dan
p-value dengan alpha
print('Keputusan berdasarkan z-value:')
print(f'z-value = {z_value}')
print(f'z-crit-r = {z_crit_r}')
print(f'z-crit-l = {z_crit_l}')
if z_value < z_crit_l or z_value > z_crit_r:
    print('H0 ditolak')
else:
    print('H0 diterima')

print('')

print('Keputusan berdasarkan p-value:')
print(f'p-value = {p_value}')
print(f'alpha = {alpha}')
if p_value < alpha:
    print('H0 ditolak')
else:
    print('H0 diterima')

Keputusan berdasarkan z-value:
z-value = 2.1479619435539523
z-crit-r = 1.96
z-crit-l = -1.96

```


H0 ditolak

Keputusan berdasarkan p-value:
p-value = 0.031716778818727365
alpha = 0.05
H0 ditolak

Soal 4.c

Nilai rata-rata 150 baris pertama kolom sulphates bukan 0.65?

=== Langkah 1

=====

=====

Tentukan Hipotesis 0 (H0)
H0: sulphates = 0.65

=== Langkah 2

=====

=====

Tentukan Hipotesis Alternatif (H1)
H1: sulphates /= 2.5

=== Langkah 3

=====

=====

Tentukan tingkat signifikan alpha yaitu 5%
alpha = 0.05

=== Langkah 4

=====

=====

Tentukan daerah kritis two-tailed
t_crit_r = 1.96
t_crit_l = -1.96

=== Langkah 5

=====

=====

Hitung nilai z-value dan p-value
z_value = (data['sulphates'].iloc[:150].mean() - 3.29) /
(data['sulphates'].iloc[:150].std() /
np.sqrt(data['sulphates'].iloc[:150].count()))
p_value = 2 * (1 - stats.norm.cdf(np.abs(z_value)))

=== Langkah 6

=====

=====

Ambil keputusan berdasarkan z-value dengan z-crit-r dan z-crit-l dan p-value dengan alpha

```
print('Keputusan berdasarkan z-value:')
print(f'z-value = {z_value}')
print(f'z_crit_r = {z_crit_l}')
print(f'z_crit_l = {z_crit_r}')
if z_value < z_crit_l or z_value > z_crit_r:
    print('H0 ditolak')
else:
    print('H0 diterima')
```

```
print('')
```

```
print('Keputusan berdasarkan p-value:')
print(f'p-value = {p_value}')
print(f'alpha = {alpha}')
if p_value < alpha:
    print('H0 ditolak')
else:
    print('H0 diterima')
```

Keputusan berdasarkan z-value:
z-value = -301.9554753801894
z_crit_r = -1.96
z_crit_l = 1.96
H0 ditolak

Keputusan berdasarkan p-value:
p-value = 0.0
alpha = 0.05
H0 ditolak

Soal 4.d

Nilai rata-rata total sulfur dioxide di bawah 35?

=== Langkah 1

=====

=====

Tentukan Hipotesis 0 (H0)
H0: pH = 35

=== Langkah 2

=====

=====

Tentukan Hipotesis Alternatif (H1)

```
# H1: pH < 35
```

```
# === Langkah 3
```

```
=====
```

```
=====
```

```
# Tentukan tingkat signifikan alpha yaitu 5%  
alpha = 0.05
```

```
# === Langkah 4
```

```
=====
```

```
=====
```

```
# Tentukan daerah kritis one-tailed  
z_crit = -1.645
```

```
# === Langkah 5
```

```
=====
```

```
=====
```

```
# Hitung nilai z-value dan p-value  
z_value = (data['total sulfur dioxide'].mean() - 35) / (data['total  
sulfur dioxide'].std() / np.sqrt(data['total sulfur  
dioxide'].count()))  
p_value = stats.norm.cdf(z_value)
```

```
# === Langkah 6
```

```
=====
```

```
=====
```

```
# Ambil keputusan berdasarkan z-value dengan z-crit dan p-value dengan  
alpha
```

```
print('Keputusan berdasarkan z-value:')
```

```
print(f'z-value = {z_value}')
```

```
print(f'z-crit = {z_crit}')
```

```
if z_value < z_crit:  
    print('H0 ditolak')
```

```
else:  
    print('H0 diterima')
```

```
print('')
```

```
print('Keputusan berdasarkan p-value:')
```

```
print(f'p-value = {p_value}')
```

```
print(f'alpha = {alpha}')
```

```
if p_value < alpha:  
    print('H0 ditolak')
```

```
else:  
    print('H0 diterima')
```

Keputusan berdasarkan z-value:
z-value = 16.786387372296744
z-crit = -1.645
H0 diterima

Keputusan berdasarkan p-value:
p-value = 1.0
alpha = 0.05
H0 diterima

Soal 4.e

Proporsi nilai total Sulfat Dioxide yang lebih dari 40, adalah tidak sama dengan 50% ?

=== Langkah 1

=====

Tentukan Hipotesis 0 (H0)
H0: pH = 0.5

=== Langkah 2

=====

Tentukan Hipotesis Alternatif (H1)
H1: pH /= 0.5

=== Langkah 3

=====

Tentukan tingkat signifikan alpha yaitu 5%
alpha = 0.05

=== Langkah 4

=====

Tentukan daerah kritis two-tailed
z_crit_r = 1.96
z_crit_l = -1.96

=== Langkah 5

=====

Hitung nilai z-value dan p-value
z_value = ((data[data['total sulfur dioxide'] > 40]['total sulfur dioxide'].count() / data['total sulfur dioxide'].count()) - 0.5) / np.sqrt(0.5 * (1 - 0.5) / data['total sulfur dioxide'].count())

```
p_value = 2 * (1 - stats.norm.cdf(np.abs(z_value)))
```

```
# === Langkah 6
```

```
=====
```

```
=====
```

```
# Ambil keputusan berdasarkan z-value dengan z-crit-r dan z-crit-l dan p-value dengan alpha
```

```
print('Keputusan berdasarkan z-value:')
```

```
print(f'z-value = {z_value}')
```

```
print(f'z-crit-r = {z_crit_r}')
```

```
print(f'z-crit-l = {z_crit_l}')
```

```
if z_value < z_crit_l or z_value > z_crit_r:
```

```
    print('H0 ditolak')
```

```
else:
```

```
    print('H0 diterima')
```

```
print('')
```

```
print('Keputusan berdasarkan p-value:')
```

```
print(f'p-value = {p_value}')
```

```
print(f'alpha = {alpha}')
```

```
if p_value < alpha:
```

```
    print('H0 ditolak')
```

```
else:
```

```
    print('H0 diterima')
```

```
Keputusan berdasarkan z-value:
```

```
z-value = 0.7589466384404118
```

```
z-crit-r = 1.96
```

```
z-crit-l = -1.96
```

```
H0 diterima
```

```
Keputusan berdasarkan p-value:
```

```
p-value = 0.4478844782641116
```

```
alpha = 0.05
```

```
H0 diterima
```

Soal 5

Tes hipotesis 2 sampel

Soal 5.a

Data kolom fixed acidity dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata kedua bagian tersebut sama?

```
# === Definisi
```

```
=====
```

```
=====
```

```
n = len(data["fixed acidity"])
```

```

fa_r = data.loc[:n//2-1, "fixed acidity"]
fa_l = data.loc[n//2:, "fixed acidity"]
mean_r = fa_r.mean()
mean_l = fa_l.mean()
std_r = fa_r.std()
std_l = fa_l.std()
n_r = fa_r.count()
n_l = fa_l.count()
alpha = 0.05

# Perform F-Test two-tailed
if std_r ** 2 > std_l ** 2:
    f_value = (std_r ** 2 / std_l ** 2)
else:
    f_value = (std_l ** 2 / std_r ** 2)
f_crit_r = stats.f.ppf(1 - alpha, n_r - 1, n_l - 1)
f_crit_l = stats.f.ppf(alpha, n_r - 1, n_l - 1)

# Ambil keputusan berdasarkan varians kedua bagian
if f_value < f_crit_l or f_value > f_crit_r:
    print("Varians tidak equal")
else:
    print("Varians equal")
print("")

# === Langkah 1
=====
# Tentukan Hipotesis 0 (H0)
# H0: mean_r = mean-l

# === Langkah 2
=====
# Tentukan Hipotesis 1 (H1)
# H0: mean_r /= mean-l

# === Langkah 3
=====
# Tentukan tingkat signifikan alpha yaitu 5%
alpha = 0.05

# === Langkah 4
=====

```

```

=====
# Tentukan batas daerah kritis two-tailed
z_crit_r = 1.96
z_crit_l = -1.96

# === Langkah 5
=====
# Tentukan nilai statistik dan hitung p-value
z_value = (mean_r - mean_l) / np.sqrt(std_r ** 2 / n_r + std_l ** 2 /
n_l)
p_value = 2 * (1 - stats.norm.cdf(np.abs(z_value)))

# === Langkah 6
=====
# Ambil keputusan berdasarkan z-value dengan z-crit-r dan z-crit-l dan
p-value dengan alpha
print('Keputusan berdasarkan z-value:')
print(f'z-value = {z_value}')
print(f'z-crit-r = {z_crit_r}')
print(f'z-crit-l = {z_crit_l}')
if z_value < z_crit_l or z_value > z_crit_r:
    print('H0 ditolak')
else:
    print('H0 diterima')

print('')

print('Keputusan berdasarkan p-value:')
print(f'p-value = {p_value}')
print(f'alpha = {alpha}')
if p_value < alpha:
    print('H0 ditolak')
else:
    print('H0 diterima')

```

Varians equal

Keputusan berdasarkan z-value:
z-value = 0.02604106999906379
z-crit-r = 1.96
z-crit-l = -1.96
H0 diterima

Keputusan berdasarkan p-value:
p-value = 0.9792245804254096

alpha = 0.05
H0 diterima

Soal 5.b

Data kolom chlorides dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata bagian awal lebih besar daripada bagian akhir sebesar 0.001?

```
# === Definisi
=====
n = len(data["chlorides"])
fa_r = data.loc[:n//2-1, "chlorides"]
fa_l = data.loc[n//2:, "chlorides"]
mean_r = fa_r.mean()
mean_l = fa_l.mean()
std_r = fa_r.std()
std_l = fa_l.std()
n_r = fa_r.count()
n_l = fa_l.count()
d0 = 0.001
alpha = 0.05

# Perform F-Test one-tailed
if std_r ** 2 > std_l ** 2:
    f_value = (std_r ** 2 / std_l ** 2)
else:
    f_value = (std_l ** 2 / std_r ** 2)
f_crit = stats.f.ppf(1 - alpha, n_r - 1, n_l - 1)

# Ambil keputusan berdasarkan varians kedua bagian
if f_value > f_crit:
    print("Varians tidak equal")
else:
    print("Varians equal")
print("")

# === Langkah 1
=====
# Tentukan Hipotesis 0 (H0)
# H0: mean_r - mean_l = 0.001

# === Langkah 2
=====
# Tentukan Hipotesis 1 (H1)
# H0: mean_r - mean_l > 0.001
```



```

# === Langkah 3
=====
=====
# Tentukan tingkat signifikan alpha yaitu 5%
alpha = 0.05

# === Langkah 4
=====
=====
# Tentukan batas daerah kritis one-tailed
z_crit = 1.645

# === Langkah 5
=====
=====
# Tentukan nilai statistik dan hitung p-value
z_value = (mean_r - mean_l - d0) / np.sqrt(std_r ** 2 / n_r + std_l **
2 / n_l)
p_value = 1 - stats.norm.cdf(z_value)

# === Langkah 6
=====
=====
# Ambil keputusan berdasarkan z-value dengan z-crit-r dan z-crit-l dan
p-value dengan alpha
print('Keputusan berdasarkan z-value:')
print(f'z-value = {z_value}')
print(f'z-crit = {z_crit}')
if z_value > z_crit:
    print('H0 ditolak')
else:
    print('H0 diterima')

print('')

print('Keputusan berdasarkan p-value:')
print(f'p-value = {p_value}')
print(f'alpha = {alpha}')
if p_value < alpha:
    print('H0 ditolak')
else:
    print('H0 diterima')

```

Variances equal

Keputusan berdasarkan z-value:

z-value = -0.4673171228521319

z-crit = 1.645

H0 diterima

Keputusan berdasarkan p-value:

p-value = 0.6798634962094465

alpha = 0.05

H0 diterima

Soal 5.c

Benarkah rata-rata sampel 25 baris pertama kolom Volatile Acidity sama dengan rata-rata 25 baris pertama kolom Sulphates ?

```
# === Definisi
```

```
=====
```

```
va = data.loc[:24, "volatile acidity"]
```

```
s = data.loc[:24, "sulphates"]
```

```
va_mean = va.mean()
```

```
s_mean = s.mean()
```

```
va_std = va.std()
```

```
s_std = s.std()
```

```
va_n = va.count()
```

```
s_n = s.count()
```

```
alpha = 0.05
```

```
# Perform F-Test two-tailed
```

```
if va_std ** 2 > s_std ** 2:
```

```
    f_value = (va_std ** 2 / s_std ** 2)
```

```
else:
```

```
    f_value = (s_std ** 2 / va_std ** 2)
```

```
f_crit_r = stats.f.ppf(1 - alpha, va_n - 1, s_n - 1)
```

```
f_crit_l = stats.f.ppf(alpha, va_n - 1, s_n - 1)
```

```
# Ambil keputusan berdasarkan varians kedua bagian
```

```
if f_value < f_crit_l or f_value > f_crit_r:
```

```
    print("Variances tidak equal")
```

```
else:
```

```
    print("Variances equal")
```

```
print("")
```

```
# === Langkah 1
```

```
=====
```

```
# Tentukan Hipotesis 0 (H0)
```

```

# H0:  $\mu = s_{mean}$ 

# === Langkah 2
=====
=====
# Tentukan Hipotesis 1 (H1)
# H0:  $\mu \neq s_{mean}$ 

# === Langkah 3
=====
=====
# Tentukan tingkat signifikan alpha yaitu 5%
alpha = 0.05

# === Langkah 4
=====
=====
# Tentukan batas daerah kritis two-tailed
z_crit_r = 1.96
z_crit_l = -1.96

# === Langkah 5
=====
=====
# Tentukan nilai statistik dan hitung p-value
z_value = (va_mean - s_mean) / np.sqrt(va_std ** 2 / va_n + s_std ** 2 / s_n)
p_value = 2 * (1 - stats.norm.cdf(np.abs(z_value)))

# === Langkah 6
=====
=====
# Ambil keputusan berdasarkan z-value dengan z-crit-r dan z-crit-l dan p-value dengan alpha
print('Keputusan berdasarkan z-value:')
print(f'z-value = {z_value}')
print(f'z-crit-r = {z_crit_r}')
print(f'z-crit-l = {z_crit_l}')
if z_value < z_crit_l or z_value > z_crit_r:
    print('H0 ditolak')
else:
    print('H0 diterima')

print('')

```

```

print('Keputusan berdasarkan p-value:')
print(f'p-value = {p_value}')
print(f'alpha = {alpha}')
if p_value < alpha:
    print('H0 ditolak')
else:
    print('H0 diterima')

```

Variances equal

Keputusan berdasarkan z-value:
z-value = -2.63748216767487
z-crit-r = 1.96
z-crit-l = -1.96
H0 ditolak

Keputusan berdasarkan p-value:
p-value = 0.008352401685453703
alpha = 0.05
H0 ditolak

Soal 5.d

Bagian awal kolom residual sugar memiliki variansi yang sama dengan bagian akhirnya?

=== Definisi

=====

```

n = len(data["residual sugar"])
fa_r = data.loc[:n//2-1, "residual sugar"]
fa_l = data.loc[n//2:, "residual sugar"]
mean_r = fa_r.mean()
mean_l = fa_l.mean()
std_r = fa_r.std()
std_l = fa_l.std()
n_r = fa_r.count()
n_l = fa_l.count()
alpha = 0.05

# Perform F-Test two-tailed
if std_r ** 2 > std_l ** 2:
    f_value = (std_r ** 2 / std_l ** 2)
else:
    f_value = (std_l ** 2 / std_r ** 2)
f_crit_r = stats.f.ppf(1 - alpha, n_r - 1, n_l - 1)
f_crit_l = stats.f.ppf(alpha, n_r - 1, n_l - 1)

# Ambil keputusan berdasarkan varians kedua bagian
print(f"f-value = {f_value}")

```

```

print(f"f-crit-r = {f_crit_r}")
print(f"f-crit-l = {f_crit_l}")
if f_value < f_crit_l or f_value > f_crit_r:
    print("Varians tidak equal")
else:
    print("Varians equal")
print("")

f-value = 1.0615664973153538
f-crit-r = 1.1588265949626437
f-crit-l = 0.8629418796107595
Varians equal

```

Soal 5.e

Proporsi nilai setengah bagian awal alcohol yang lebih dari 7, adalah lebih besar daripada, proporsi nilai yang sama di setengah bagian akhir alcohol?

```

# === Definisi
=====
n = len(data["alcohol"])
al_r = data.loc[:n//2-1, "alcohol"]
al_l = data.loc[n//2:, "alcohol"]
n_r = al_r.count()
n_l = al_l.count()
std_r = al_r.std()
std_l = al_l.std()
p_r = (al_r > 7).sum() / n_r
p_l = (al_l > 7).sum() / n_l
p = (al_r > 7).sum() / n
alpha = 0.005

# Perform F-Test one-tailed
if std_r ** 2 > std_l ** 2:
    f_value = (std_r ** 2 / std_l ** 2)
else:
    f_value = (std_l ** 2 / std_r ** 2)
f_crit = stats.f.ppf(1 - alpha, n_r - 1, n_l - 1)

# Ambil keputusan berdasarkan varians kedua bagian
if f_value > f_crit:
    print("Varians tidak equal")
else:
    print("Varians equal")
print("")

# === Langkah 1

```

```

=====
=====
# Tentukan Hipotesis 0 (H0)
# H0:  $p_r = p_l$ 

# === Langkah 2
=====
=====
# Tentukan Hipotesis 1 (H1)
# H0:  $p_r > p_l$ 

# === Langkah 3
=====
=====
# Tentukan tingkat signifikan alpha yaitu 5%
alpha = 0.05

# === Langkah 4
=====
=====
# Tentukan batas daerah kritis one-tailed
z_crit = 1.645

# === Langkah 5
=====
=====
# Tentukan nilai statistik dan hitung p-value
z_value = (p_r - p_l) / np.sqrt(p * (1 - p) * (1/n_r + 1/n_l))
p_value = 1 - stats.norm.cdf(z_value)

# === Langkah 6
=====
=====
# Ambil keputusan berdasarkan z-value dengan z-crit-r dan z-crit-l dan
p-value dengan alpha
print('Keputusan berdasarkan z-value:')
print(f'z-value = {z_value}')
print(f'z-crit = {z_crit}')
if z_value > z_crit:
    print('H0 ditolak')
else:
    print('H0 diterima')

print('')

```

```

print('Keputusan berdasarkan p-value:')
print(f'p-value = {p_value}')
print(f'alpha = {alpha}')
if p_value < alpha:
    print('H0 ditolak')
else:
    print('H0 diterima')

```

Varians equal

Keputusan berdasarkan z-value:

z-value = 0.0

z-crit = 1.645

H0 diterima

Keputusan berdasarkan p-value:

p-value = 0.5

alpha = 0.05

H0 diterima

Referensi

- <https://www.liputan6.com/hot/read/4470780/mean-adalah-nilai-rata-rata-ketahui-rumus-contoh-dan-jenisnya#:~:text=Dalam%20bahasa%20Indonesia%2C%20mean%20juga,mudah%20dihitung%20dengan%20rumus%20sederhana>
- <https://dailysocial.id/post/median-pengertian-contoh-rumus-dan-cara-menghitungnya>
- <https://www.merdeka.com/jateng/modus-adalah-nilai-data-dalam-statistik-ketahui-jenis-dan-cara-menghitungnya-kln.html>
- <https://money.kompas.com/read/2021/06/23/150642926/standar-deviasi-adalah-perhitungan-statistik-simak-rumus-dan-kegunaannya>
- <https://www.akseleran.co.id/blog/rumus-varians/>
- [https://akupintar.id/info-pintar/-/blogs/simpangan-kuartil-pengertian-rumus-jangkauan-antar-kuartil-dan-contoh-soal#:~:text=Kuartil%20adalah%20nilai%20yang%20membagi,dan%20kuartil%20atas%20\(Q3\)](https://akupintar.id/info-pintar/-/blogs/simpangan-kuartil-pengertian-rumus-jangkauan-antar-kuartil-dan-contoh-soal#:~:text=Kuartil%20adalah%20nilai%20yang%20membagi,dan%20kuartil%20atas%20(Q3))
- <https://dt-sense.com/2021/09/26/belajar-mengenai-ukuran-letak-data/>
- <https://www.editage.com/blog/normality-test-methods-of-assessing-normality/>
- <https://support.minitab.com/en-us/minitab/21/help-and-how-to/statistics/basic-statistics/how-to/normality-test/interpret-the-results/key-results/#:~:text=If%20the%20p%2Dvalue%20is,not%20follow%20a%20normal%20distribution.&text=If%20the%20p%2Dvalue%20is%20larger%20than%20the%20significance%20level,to%20reject%20the%20null%20hypothesis.>

- R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, Probability & Statistics for Engineers & Scientists, 9th ed. London: Pearson Education, 2016, p. 791.