Directory, Libraries and data

In [ ]:

```
%cd /content/drive/MyDrive/Business Analyst course/Statistics and Descriptive Analytics/
Intermediary Statistics
```

```
/content/drive/MyDrive/Business Analyst course/Statistics and Descriptive Analytics/Inter
mediary Statistics
```

In [ ]:

```
#Libraries
import pandas as pd
import scipy.stats as st
import math as m
import statsmodels.stats.api as sm
```

In [ ]:

```
#Load Data
df = pd.read_csv("Wine-quality-challenge.csv")
df.head()
```

Out[ ]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

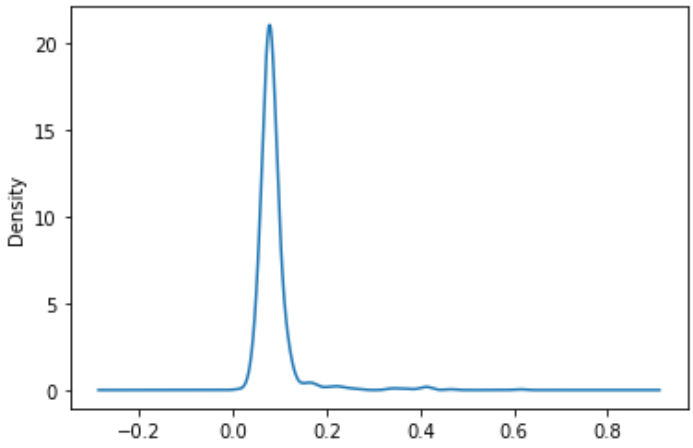
Normal distribution

In [ ]:

```
#Density distribution
df.chlorides.plot.density()
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f625cdd1fd0>
```



In [ ]:

```
#68-95-99 check
df.loc[(df.chlorides <= df.chlorides.mean() + 2 * df.chlorides.std()) &
       (df.chlorides >= df.chlorides.mean() - 2 * df.chlorides.std())].chlorides.count()
/df.chlorides.count()
```

Out[ ]:

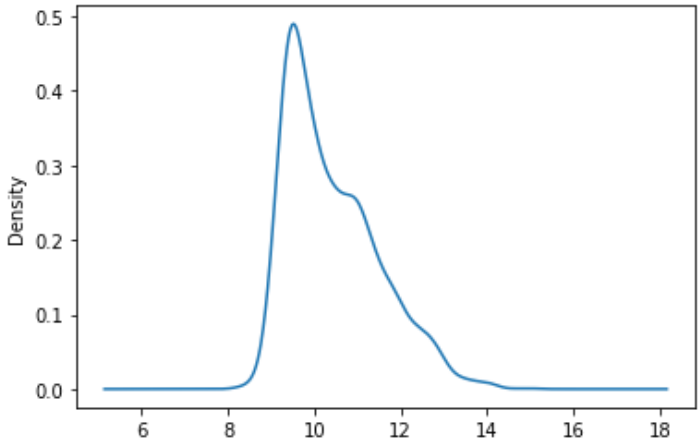
```
0.9718574108818011
```

In [ ]:

```
#plot alcohol distribution
df.alcohol.plot.density()
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f625cd47a90>
```



In [ ]:

```
#check how many observations within 3 standard deviations
df.loc[(df.alcohol <= df.alcohol.mean() + 1 * df.alcohol.std()) &
       (df.alcohol >= df.alcohol.mean() - 1 * df.alcohol.std())].alcohol.count()/df.alco
hol.count()
```

Out[ ]:

```
0.7035647279549718
```

Shapiro-Wilks Test

In [ ]:

```
#Shapiro-Wilks for normality
stat, p = st.shapiro(df.chlorides)
print(p)
#condition
if p > 0.05:
  #if yes
  print('Sample looks Gaussian/Normal (fail to reject H0')
  #if not
else:
  print('Sample does not look Gaussian/Normal (reject H0)')
```

```
0.0
Sample does not look Gaussian/Normal (reject H0)
```

In [ ]:

```
#Shapiro Wilks Test for Sulphates and create if else condition
stat, p = st.shapiro(df.sulphates)
print(p)
if p > 0.05:
  print('Sample looks Gaussian or Normal (Fail to reject)')
```

```
else:
  print('Sample does not look Gaussian / Normal (reject the H0)')
```

```
5.821617678881608e-38
Sample does not look Gaussian / Normal (reject the H0)
```

Standard Error

In [ ]:

```
#Using a function
st.sem(df.alcohol)
```

Out[ ]:

```
0.026650018979018173
```

In [ ]:

```
#Us doing the computations: Standard deviations divided by square root of observations
df.alcohol.std() / m.sqrt(df.alcohol.count())
```

Out[ ]:

```
0.026650018979018173
```

In [ ]:

```
#Standard Error of pH
print(st.sem(df.pH))
df.pH.std() / m.sqrt(df.pH.count())
```

```
0.0038608683325203784
```

Out[ ]:

```
0.0038608683325203784
```

Confidence Interval

In [ ]:

```
#Confidence interval of the mean of citric acid
print(df[['citric acid']].mean())
st.norm.interval(alpha = 0.95,
                 loc = df[['citric acid']].mean(),
                 scale = st.sem(df[['citric acid']]))
```

```
citric acid    0.270976
dtype: float64
```

Out[ ]:

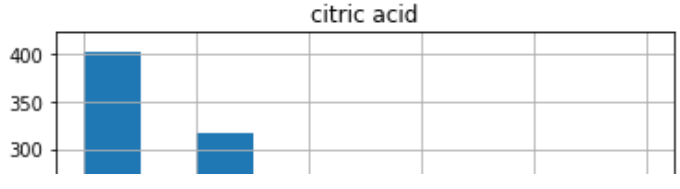
```
(array([0.26142755]), array([0.28052367]))
```

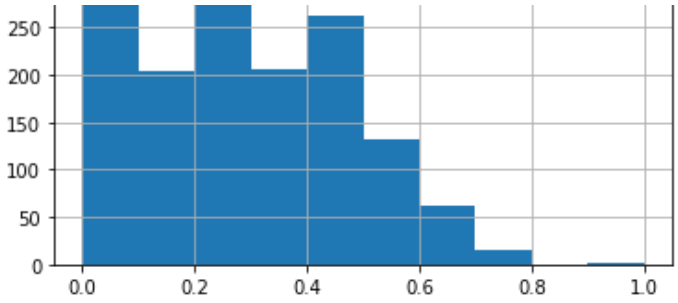
In [ ]:

```
#Histogram
df[['citric acid']].hist()
```

Out[ ]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f625c80a650>]],
      dtype=object)
```





In [ ]:

```
#Confidence interval of the Density mean
st.norm.interval(alpha = 0.95,
                 loc = df.density.mean(),
                 scale = st.sem(df.density))
```

Out[ ]:

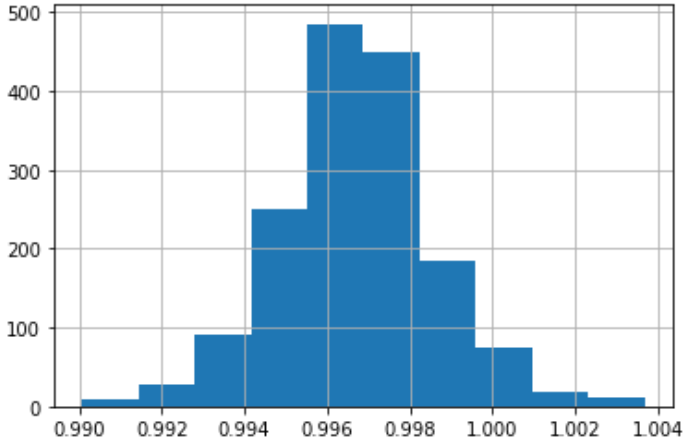
```
(0.9966541725972521, 0.9968391857517162)
```

In [ ]:

```
#Histogram of Density mean
df.density.hist()
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f625c7acf10>
```



T-test

In [ ]:

```
#load data
data = pd.read_csv("stackoverflow.csv")
data.head()
```

Out[ ]:

	Country	Salary	YearsCodedJob	OpenSource	Hobby	CompanySizeNumber	Remote	CareerSatisfaction	Data_sci
0	United Kingdom	100000.000000	20	0	1	5000	Remote	8	
1	United States	130000.000000	20	1	1	1000	Remote	9	
2	United States	175000.000000	16	0	1	10000	Not remote	7	
3	Germany	64516.129030	4	0	0	1000	Not remote	9	
4	India	6636.323594	1	0	1	5000	Not remote	5	

	Country	Salary	YearsCodedJob	OpenSource	Hobby	CompanySizeNumber	Remote	CareerSatisfaction	Data_sci

5 rows × 21 columns

In [ ]:

```
#subset
salary_uk = data.loc[data.Country == 'United Kingdom'].Salary
salary_de = data.loc[data.Country == 'Germany'].Salary
```

In [ ]:

```
#T-test
stat, p = st.ttest_ind(a = salary_uk, b = salary_de)
print(p)
if p > 0.05:
  print('Both countries have similar salaries (fail to reject H0)')
else:
  print('There is a difference in salaries (reject H0)')
```

```
0.026389999555203502
There is a difference in salaries (reject H0)
```

In [ ]:

```
#T-test in experience between India and United States
us_experience = data.loc[data.Country == 'United States'].YearsCodedJob
in_experience = data.loc[data.Country == 'India'].YearsCodedJob
stat, p = st.ttest_ind(a = us_experience, b = in_experience)
print(p)
if p > 0.05:
  print('Groups are similar (fail to reject H0)')
else:
  print('Groups are different (reject H0)')
```

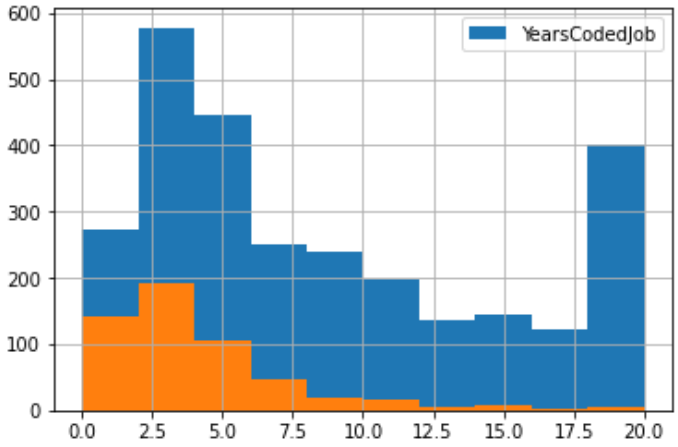
```
5.225676347614714e-58
Groups are different (reject H0)
```

In [ ]:

```
#Histograms
us_experience.hist(legend= True)
in_experience.hist()
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f625c79e3d0>
```



Chi-square test

In [ ]:

```
#cross tabulation
tab = pd.crosstab(index = data.Country,
```

```
                    columns = data.Remote)
tab
```

Out[ ]:

Remote	Not remote	Remote
Country		
Canada	457	28
Germany	717	40
India	482	56
United Kingdom	953	70
United States	2410	381

In [ ]:

```
#chi-square test
chi2, p, dof, exp = st.chi2_contingency(tab)
print(p)
if p > 0.05:
  print("there is no relationship (fail to reject H0)")
else:
  print('There is a strong relationshiop (reject H0)')
```

```
3.321120877301216e-16
There is a strong relationshiop (reject H0)
```

In [ ]:

```
#Chi square test between company size and hobbies
tab2 = pd.crosstab(index = data.Hobby,
                   columns = data.CompanySizeNumber)
chi2, p, dof, exp = st.chi2_contingency(tab2)
print(p)
if p > 0.05:
  print('There is no relationship (fail to reject H0)')
else:
  print('There is a strong relationship (reject H0)')
```

```
0.025708455559671013
There is a strong relationship (reject H0)
```