EE18BTECH11026_A5

February 21, 2022

[35]:
```
### IMPORTS

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

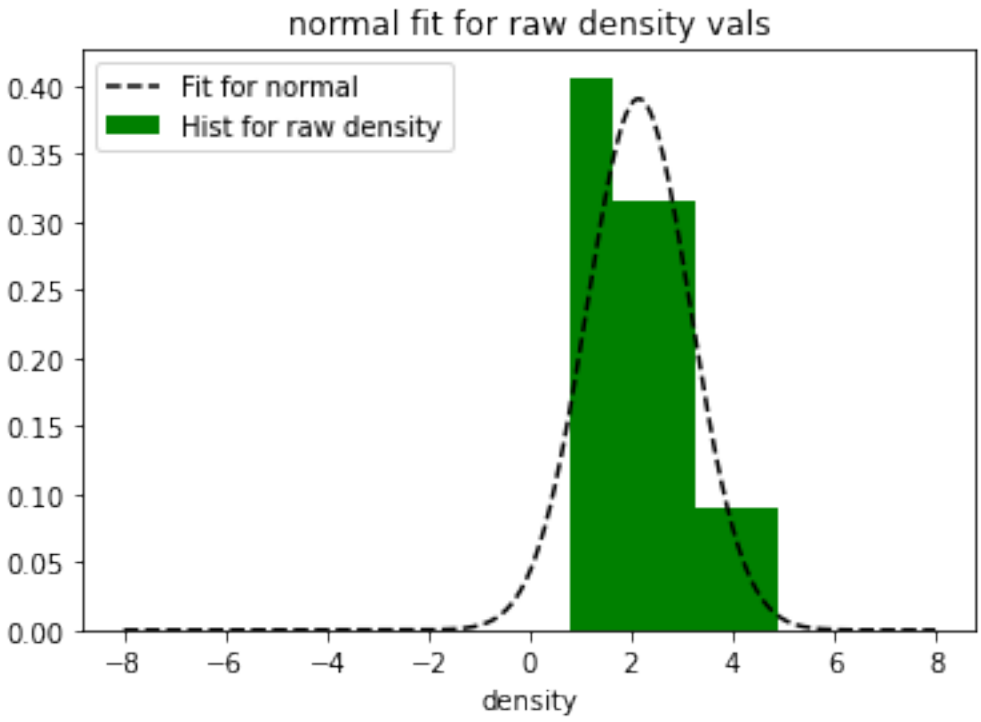
1 Q1

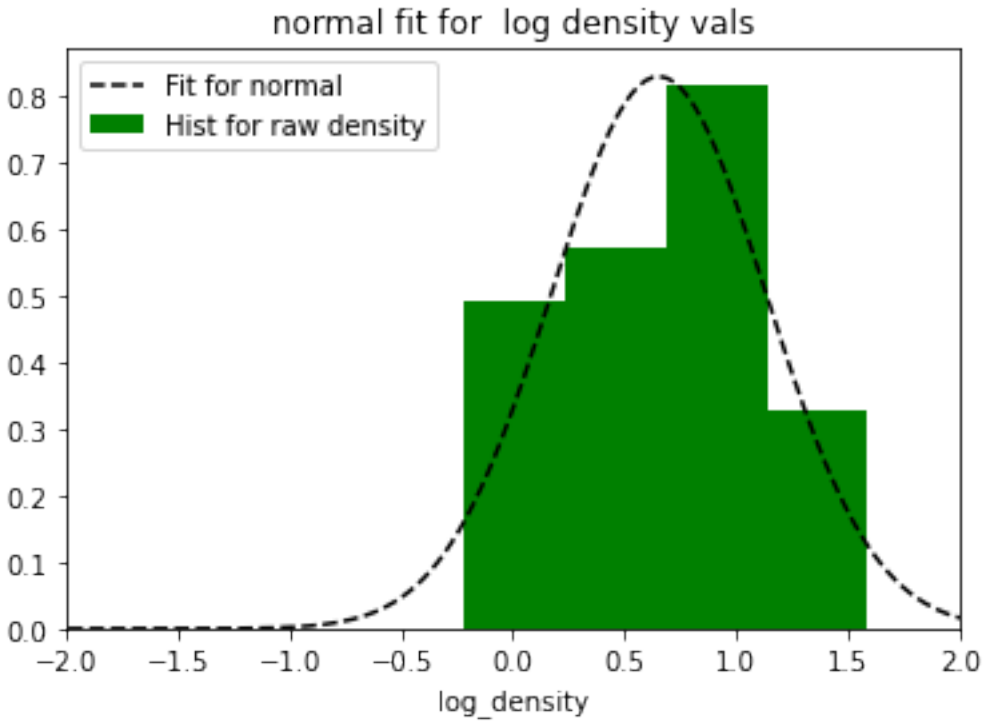
[16]:
```
##

density = np.loadtxt('Q1_data.dat')
log_density = np.log(density)

## Shapiro Wilk test
_, p_density = stats.shapiro(density)
_, p_log_density = stats.shapiro(log_density)

print('P-val from Shapiro Wilk for raw density vals {}'.format(p_density))
print('P-val from Shapiro Wilk for log density vals {}'.format(p_log_density))

print(

    '''
We can observe that p-val of Shapiro-Wilko test for raw density vals is close␣
 ↪to 0.05, while for log values it is >> 0.05.
Hence for log of density values , null hypothesis(ie, normal distribution) cant␣
 ↪be regected .
    '''
)
```

```
P-val from Shapiro Wilk for raw density vals 0.051220282912254333
P-val from Shapiro Wilk for log density vals 0.5660613775253296

We can observe that p-val of Shapiro-Wilko test for raw density vals is close to
0.05, while for log values it is >> 0.05.
```

Hence for log of density values , null hypothesis(ie, normal distribution) cant be regected .

[28]:
```
## Plotting
x=  np.linspace(-8,8, 500)
density_fit = stats.norm.pdf(x, *stats.norm.fit(density))
log_density_fit = stats.norm.pdf(x, *stats.norm.fit(log_density))

plt.plot(x, density_fit, 'k--', label = 'Fit for normal')
plt.hist(density, bins = 'fd', density = True,color = 'g', label='Hist for raw␣
 ↪density')
plt.xlabel('density')
plt.title('normal fit for raw density vals')
plt.legend()
plt.show()


plt.plot(x, log_density_fit, 'k--', label = 'Fit for normal')
plt.hist(log_density, bins = 'fd',color = 'g', density = True,  label='Hist for␣
 ↪raw density')
plt.xlabel('log_density')
plt.xlim([-2, 2])
plt.title('normal fit for  log density vals')
plt.legend()
plt.show()
```





2 Q2

[51]:
```
### importing data
data = np.genfromtxt('Q2_data.dat', skip_header=0 , skip_footer=0, names=True,␣
 ↪dtype=None, delimiter=' ')
col_RA, col_DE, col_pmRA, col_pmDE, col_BV = 2,3,5,6,8

hyade_BV , non_hyade_BV = [], []

for row in data:
    if (all([row[col_RA] >  50 , row[col_RA] <  100 , row[col_DE] > 0,␣
 ↪row[col_DE] < 25,  row[col_pmRA] > 90, row[col_pmRA] < 130, row[col_pmDE] >␣
 ↪-60 , row[col_pmDE] < -10 ])):
        hyade_BV.append(row[col_BV])
    else:
        non_hyade_BV.append(row[col_BV])

### Variance of both hyade and non-hyades
var_hyade_BV, var_non_hyade_BV = np.var(hyade_BV), np.var(non_hyade_BV)
```

```
ttest_stat, pval_ttest = stats.ttest_ind(hyade_BV, non_hyade_BV)

print('Var in Hyade star colors : {}'.format(var_hyade_BV))
print('Var in Non- Hyade star colors : {}\n'.format(var_non_hyade_BV))

print('It is observed that both variance are almost equal , thus we can proceed␣
 ↪with 2 sample ttest ...')
print('\n T - statistic for hyade and non hyade star colors : {}'.
 ↪format(ttest_stat))
print('P-val from 2 sample T-test: {}'.format(pval_ttest))

print('''
P-val << 0.05, thus we can reject null hypothesis that the color of hyades␣
 ↪stars is same as the non-hyades ones.The high value of T-statistic conveys␣
 ↪that they differ by 4 times
''')
```

```
Var in Hyade star colors : 0.10580084865302346
Var in Non- Hyade star colors : 0.10776893915957887

It is observed that both variance are almost equal , thus we can proceed with 2
sample ttest …

 T - statistic for hyade and non hyade star colors : -3.860436921860911
P-val from 2 sample T-test: 0.00011582222192442334

P-val << 0.05, thus we can reject null hypothesis that the color of hyades stars
is same as the non-hyades ones.The high value of T-statistic conveys that they
differ by 4 times
```

3 The end