chap3-statistics

June 16, 2022

[1]:
```python
# import pandas library
import pandas as pd

# create dataframe
sample_data = {'name': ['John', 'Alia', 'Anna', 'Monika', 'Sylwia'],
              'gender': ['M', 'F', 'F', 'F', 'F'],
              'communication_skill_score': [30, 40, 45, 50, 55],
              'quantitative_skill_score': [38, 41, 42, 48, 32]}
data = pd.DataFrame(sample_data, columns=['name',␣
 ↪'gender','communication_skill_score', 'quantitative_skill_score'])

# find mean of communication_skill_score column
data['communication_skill_score'].mean(axis=0)
```

[1]:
```
44.0
```

[2]:
```python
# find mode of communication_skill_score column
data['communication_skill_score'].mode()
```

[2]:
```
0    30
1    40
2    45
3    50
4    55
Name: communication_skill_score, dtype: int64
```

[3]:
```python
# find median of communication_skill_score column
data['communication_skill_score'].median()
```

[3]:
```
45.0
```

[4]:
```python
# measuring dispersion
column_range=data['communication_skill_score'].
 ↪max()-data['communication_skill_score'].min()
print(column_range)
```

```
25
```

[5]:
```python
# first Quartile
q1 = data['communication_skill_score'].quantile(.25)

# third Quartile
q3 = data['communication_skill_score'].quantile(.75)

# inter Quartile Ratio
iqr = q3 - q1
print(iqr)
```

```
10.0
```

[6]:
```python
# variance of communication_skill_score
data['communication_skill_score'].var()
```

[6]: 92.5

[7]:
```python
# standard deviation of communication_skill_score
data['communication_skill_score'].std()
```

[7]: 9.617692030835672

[8]:
```python
# discribe dataframe
data.describe()
```

[8]:

	communication_skill_score	quantitative_skill_score
count	5.000000	5.000000
mean	44.000000	40.200000
std	9.617692	5.848077
min	30.000000	32.000000
25%	40.000000	38.000000
50%	45.000000	41.000000
75%	50.000000	42.000000
max	55.000000	48.000000

[9]:
```python
#skewness of communication_skill_score column
data['communication_skill_score'].skew()
```

[9]: -0.5901286563843656

[10]:
```python
# kurtosis of communication_skill_score column
data['communication_skill_score'].kurtosis()
```

[10]: -0.02191380569758916

[11]:
```python
# covariance between columns of dataframe
data.cov()
```

[11]:

	communication_skill_score	quantitative_skill_score
communication_skill_score	92.5	-3.5
quantitative_skill_score	-3.5	34.2

[12]:
```python
# correlation between columns of dataframe
data.corr(method='pearson')
```

[12]:

	communication_skill_score	quantitative_skill_score
communication_skill_score	1.000000	-0.062228
quantitative_skill_score	-0.062228	1.000000

[13]:
```python
##### Performing parametric tests #####
import numpy as np
from scipy.stats import ttest_1samp

# create data
data = np.array([63, 75, 84, 58, 52, 96, 63, 55, 76, 83])

# find mean
mean_value = np.mean(data)

print("Mean: ", mean_value)
```

```
Mean:  70.5
```

[14]:
```python
# perform one-sample t-test
t_test_value, p_value = ttest_1samp(data, 68)

print("P Value: ", p_value)

print("t-test Value: ", t_test_value)

# 0.05 or 5 % is significance level or alpha
if p_value < 0.05:
    print("Hipothesis Rejected")
else:
    print("Hipothesis Accepted")
```

```
P Value:  0.5986851106160134
t-test Value:  0.5454725779039431
Hipothesis Accepted
```

[15]:
```python
from scipy.stats import ttest_ind

# create numpy arrays
data1 = np.array([63, 75, 84, 58, 52, 96, 63, 55, 76, 83])
```

```
data2 = np.array([53, 43, 31, 113, 33, 57, 27, 23, 24, 43])

# compare samples
stat, p = ttest_ind(data1, data2)

print("p_value: ", p)

print("t_test: ", stat)

# 0.05 or 5 % is the significance level of alpha
if p < 0.05:
    print("Hypothesis Rejected")
else:
    print("Hypotesis Accepted")
```

```
p_value:  0.015170931362451255
t_test:  2.6835879913819185
Hypothesis Rejected
```

[16]:
```
from scipy.stats import f_oneway

# performance scores of Mumbai location
mumbai = [0.14730927, 0.59168541, 0.85677052, 0.27315387, 0.78591207, 0.
 ↪52426114,
         0.05007655, 0.64405363, 0.9825853, 0.62667439]

# performance scores of Chicago location
chicago = [0.99140754, 0.76960782, 0.51370154, 0.85041028, 0.19485391, 0.
 ↪25269917,
          0.19925735, 0.80048387, 0.98381235, 0.5864963]

# performace scores of London location
london = [0.40382226, 0.51613408, 0.39374473, 0.0689976, 0.28035865, 0.56326686,
         0.66735357, 0.06786065, 0.21013306, 0.86503358]
```

[17]:
```
# compare results using Oneway ANOVA
stat, p = f_oneway(mumbai, chicago, london)

print("p-value:", p)

print("ANOVA:", stat)

if p < 0.05:
    print("Hypothesis Rejected")
else:
    print("Hypothesis Accepted")
```

```
p-value: 0.27667556390705783
ANOVA: 1.3480446381965452
Hypothesis Accepted
```

[20]:
```
#### Performing non-parametric tests ####

## chi-square test
from scipy.stats import chi2_contingency

# average performing employee
average = [20, 16, 13, 7]

# outstanding performing employees
outstanding = [31, 40, 60, 13]

#contingency table
contingency_table = [average, outstanding]
```

[22]:
```
# apply test
stat, p, dof, expected = chi2_contingency(contingency_table)

print("p-value:", p)

if p < 0.05:
    print("Hypothesis Rejected")
else:
    print("Hypothesis Accepted")
```

```
p-value: 0.059155602774381234
Hypothesis Accepted
```

[23]:
```
# Mann-Whitney U test

from scipy.stats import mannwhitneyu

# sample 1
data1 = [7,8,4,9,8]

# sample 2
data2 = [3,4,2,1,1]

# apply test
stat, p = mannwhitneyu(data1, data2)

print("p-value: ", p)
```

```python
# 0.01 or 1 % is a significance level of alpha

if p < 0.01:
    print("Hypothesis Rejected")
else:
    print("Hypothesis Accepted")
```

```
p-value:  0.015333162113602824
Hypothesis Accepted
```

[24]:
```python
## Wilcoxon test

from scipy.stats import wilcoxon

# sample1

data1 = [1, 3, 5, 7, 9]

# sample-2 after treatment
data2 = [2, 4, 6, 8, 10]

# apply
stat, p = wilcoxon(data1, data2)

print("p-value: ", p)

# 0.01 or 1 % is significance level of alpha
if p < 0.01:
    print("Hypothesis Rejected")
else:
    print("Hypothesis Accepted")
```

```
p-value:  0.0625
Hypothesis Accepted
```

[29]:
```python
## Kruskal-Wallis test
from scipy.stats import kruskal

# data sample 1
x = [38, 18, 39, 83, 15, 38, 63, 1, 34, 50]

# data sample 2
y = [78, 32, 58, 59, 74, 77, 29, 77, 54, 59]

# data sample 3
z = [117, 92, 42, 79, 58, 117, 46, 114, 86, 26]
```

```python
# apply kruskal-wallis test
stat, p = kruskal(x, y, z)

print("p-value: ", p)

# 0.01 or 1 % is significance level or alpha
if p < 0.01:
    print("Hypothesis Rejected")
else:
    print("Hypothesis Accepted")
```

```
p-value:  0.01997922369138151
Hypothesis Accepted
```

[ ]:

```python

```