In [3]:
```
# importing packages
import scipy.stats

# finding p-value
p_value = scipy.stats.norm.sf(abs(-0.67))
('p value is : ' + str(p_value))
```

Out[3]:
```
'p value is : 0.25142889509531013'
```

In [5]:
```
# importing packages
import scipy.stats

# finding p-value
p_value = scipy.stats.norm.sf(abs(1.67))
('p value is : ' + str(p_value))
```

Out[5]:
```
'p value is : 0.04745968180294733'
```

In [6]:
```
# Calculate the z-score from with scipy
import scipy.stats as stats
values = [4,5,6,6,6,7,8,12,13,13,14,18]

zscores = stats.zscore(values)
print(zscores)
```

```
[-1.2493901  -1.01512945 -0.78086881 -0.78086881 -0.78086881 -0.54660817
 -0.31234752  0.62469505  0.85895569  0.85895569  1.09321633  2.0302589 ]
```

In [8]:
```
# Calculate the Standard Deviation in Python
mean = sum(values) / len(values)
differences = [(value - mean)**2 for value in values]
sum_of_differences = sum(differences)
standard_deviation = (sum_of_differences / (len(values) - 1)) ** 0.5
(standard_deviation)
```

Out[8]:
```
4.458563432181702
```

In [ ]:
```

```

In [20]:
```
# Loading a Sample Pandas Dataframe
import pandas as pd

df = pd.DataFrame.from_dict({
    'Name': ['Nik', 'Kate', 'Joe', 'Mitch', 'Alana'],
    'Age': [32, 30, 67, 34, 20],
    'Amount': [80000, 90000, 45000, 23000, 12000],
    'Level' : [5, 7, 3, 4, 4]
})
(df.head())
```

Out[20]:

	Name	Age	Amount	Level
0	Nik	32	80000	5
1	Kate	30	90000	7
2	Joe	67	45000	3
3	Mitch	34	23000	4
4	Alana	20	12000	4

In [ ]:

In [ ]:

In [21]:
```
# Loading a Sample Pandas Dataframe
import pandas as pd

df = pd.DataFrame.from_dict({
    'Name': ['Nik', 'Kate', 'Joe', 'Mitch', 'Alana'],
    'Age': [32, 30, 67, 34, 20],
    'Amount': [80000, 90000, 45000, 23000, 12000],
    'Level' : [5, 7, 3, 4, 4]
})
(df.head())
```

Out[21]:

	Name	Age	Amount	Level
0	Nik	32	80000	5
1	Kate	30	90000	7
2	Joe	67	45000	3
3	Mitch	34	23000	4
4	Alana	20	12000	4

In [27]:
```
df['Amount zscore'] = stats.zscore(df['Amount'])
(df.head())
```

Out[27]:

	Name	Age	Amount	Level	Income zscore	Amount zscore
0	Nik	32	80000	5	0.978700	0.978700
1	Kate	30	90000	7	1.304934	1.304934
2	Joe	67	45000	3	-0.163117	-0.163117
3	Mitch	34	23000	4	-0.880830	-0.880830
4	Alana	20	12000	4	-1.239687	-1.239687

In [28]:
```
import scipy.stats as stats
import math

# Specify the sample mean (x_bar), the sample standard deviation (s), the mean
claimed in the null-hypothesis (mu_null), and the sample size (n)
x_bar = 62.1
s = 13.46
mu_null = 55
n = 30

# Calculate the test statistic
test_stat = (x_bar - mu_null)/(s/math.sqrt(n))

# Output the p-value of the test statistic (right tailed test)
print(1-stats.t.cdf(test_stat, n-1))
```

```
0.0036182783769480586
```

In [ ]:

In [ ]: