Statistical Analysis in Python

1. Distributions in Pandas

1.1 Binomial Distribution

In [1]:

```
import pandas as pd
import numpy as np
```

Generate a random number following binomial(1, 0.5).

In [3]:

```
np.random.binomial(1,0.5) # n=1, p=0.5
```

Out[3]:

```
0
```

Generate a list of random numbers following binomial distribution.

In [5]:

```
tornado_events = np.random.binomial(1, 0.3, 20)
tornado_events
```

Out[5]:

```
array([1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1])
```

1.2 Normal Distribution

In [11]:

```
np.random.normal(loc = 0,     # mean of the distribution
                 scale = 1,   # standard deviation
                 size = 10    # size of sample
                )
```

Out[11]:

```
array([ 0.65732337,  0.72235445, -1.30568457, -0.07198881,  0.54469687,
       -1.03769059, -0.54410273, -0.14434303,  1.14569897,  1.40586168])
```

1.3 Chi-square Distribution

In [20]:

```
chi_squared_df2 = np.random.chisquare(2,  # d.o.f
                                      size = 10000)
```

1.4 Kurtosis

Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution.

In [17]:

```
import scipy.stats as stats
distribution = np.random.normal(0.75, size = 1000)
stats.kurtosis(distribution)
```

Out[17]:

```
0.626741258328881
```

1.5 Skewness

In [18]:

```
stats.skew(distribution)
```

Out[18]:

```
-0.10232040210883142
```

In [21]:

```
stats.skew(chi_squared_df2)
```

Out[21]:

```
1.951149051921271
```

2. Hypothesis Testing

Loading dataset.

In [24]:

```
%cd D:\Data Science\GitHub\Python Learning\Python-for-Data-Science\Data Files\Python Le
arning
df = pd.read_csv('grades.csv')
df.head()
```

```
D:\Data Science\GitHub\Python Learning\Python-for-Data-Science\Data Files
\Python Learning
```

Out[24]:

	student_id	assignment1_grade	assignment1_submission	assignment2_grade	assignm
0	B73F2C11-70F0-E37D-8B10-1D20AFED50B1	92.733946	2015-11-02 06:55:34.282000000	83.030552	02
1	98A0FAE0-A19A-13D2-4BB5-CFBFD94031D1	86.790821	2015-11-29 14:57:44.429000000	86.290821	17
2	D0F62040-CEB0-904C-F563-2F8620916C4E	85.512541	2016-01-09 05:36:02.389000000	85.512541	06
3	FFDF2B2C-F514-EF7F-6538-A6A53518E9DC	86.030665	2016-04-30 06:50:39.801000000	68.824532	17
4	5ECBEEB6-F1CE-80AE-3164-E45E99473FB4	64.813800	2015-12-13 17:06:10.750000000	51.491040	12

Split the data into two parts: 'early' & 'late'

In [25]:

```
early = df[df['assignment1_submission'] <= '2015-12-31']
late = df[df['assignment1_submission'] > '2015-12-31']
```

In [26]:

```
early.mean()
```

Out[26]:

```
assignment1_grade    74.972741
assignment2_grade    67.252190
assignment3_grade    61.129050
assignment4_grade    54.157620
assignment5_grade    48.634643
assignment6_grade    43.838980
dtype: float64
```

In [27]:

```
late.mean()
```

Out[27]:

```
assignment1_grade    74.017429
assignment2_grade    66.370822
assignment3_grade    60.023244
assignment4_grade    54.058138
assignment5_grade    48.599402
assignment6_grade    43.844384
dtype: float64
```

Test whether there is a difference between early & late courses.

In [30]:

```
from scipy import stats
stats.ttest_ind(early['assignment1_grade'], late['assignment1_grade'])
```

Out[30]:

```
Ttest_indResult(statistic=1.400549944897566, pvalue=0.16148283016060577)
```

In [29]:

```
stats.ttest_ind(early['assignment2_grade'], late['assignment2_grade'])
```

Out[29]:

```
Ttest_indResult(statistic=1.3239868220912567, pvalue=0.18563824610067967)
```

In [31]:

```
stats.ttest_ind(early['assignment3_grade'], late['assignment3_grade'])
```

Out[31]:

```
Ttest_indResult(statistic=1.7116160037010733, pvalue=0.08710151634155668)
```