Khalil MASMOUDI

Journée Statistique, Mathématique et Data Science 2019

Distributions in Pandas

In [1]:

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

In [4]:

```
np.random.binomial(1, 0.5,10)
```

Out[4]:

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

In [5]:

```
np.random.binomial(1000, 0.5)/1000
```

Out[5]:

```
0.494
```

In [6]:

```
chance_of_tornado = 0.01/100
np.random.binomial(100000, chance_of_tornado)
```

Out[6]:

```
13
```

In [7]:

```
chance_of_tornado = 0.01

tornado_events = np.random.binomial(1, chance_of_tornado, 1000000)

two_days_in_a_row = 0
for j in range(1,len(tornado_events)-1):
    if tornado_events[j]==1 and tornado_events[j-1]==1:
        two_days_in_a_row+=1

print('{} tornadoes back to back in {} years'.format(two_days_in_a_row, 1000000/365))
```

```
93 tornadoes back to back in 2739.72602739726 years
```

In [8]:

```
np.random.uniform(0, 1)
```

Out[8]:

```
0.4483134735515122
```

In [9]:

```
np.random.normal(0.75)
```

Out[9]:

```
0.5744400118902134
```

Formula for standard deviation

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

In [8]:

```
distribution = np.random.normal(0.75,size=1000)

np.sqrt(np.sum((np.mean(distribution)-distribution)**2)/len(distribution))
```

Out[8]:

```
0.9906245129774512
```

In [9]:

```
np.std(distribution)
```

Out[9]:

```
0.9906245129774512
```

In [10]:

```
import scipy.stats as stats
stats.kurtosis(distribution)
```

Out[10]:

```
-0.2051648792431222
```

In [11]:

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

Out[11]:

```
-0.02197682854117612
```

In [12]:

```
chi_squared_df2 = np.random.chisquare(2, size=10000)
stats.skew(chi_squared_df2)
```

Out[12]:

```
1.8929664385287919
```

In [13]:

```
chi_squared_df5 = np.random.chisquare(5, size=10000)
stats.skew(chi_squared_df5)
```

Out[13]:

```
1.2929994659448996
```

In [10]:

```
import scipy.stats as sc
sc.norm.
```

In [14]:

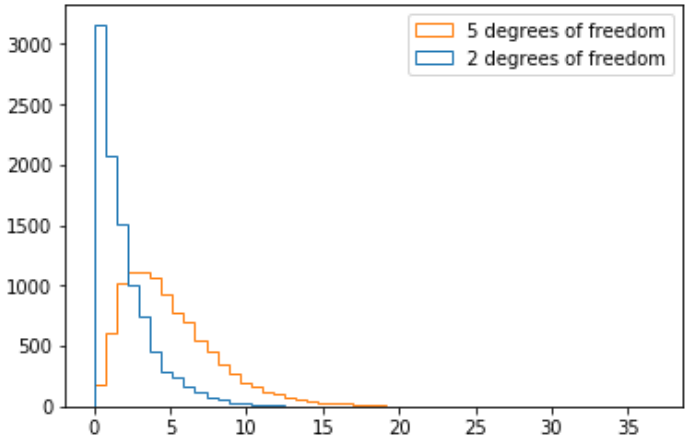
```
%matplotlib inline
import matplotlib
```

```
import matplotlib.pyplot as plt

output = plt.hist([chi_squared_df2,chi_squared_df5], bins=50, histtype='step',
                  label=['2 degrees of freedom','5 degrees of freedom'])
plt.legend(loc='upper right')
```

Out[14]:

```
<matplotlib.legend.Legend at 0x17a33b70240>
```



Hypothesis Testing

In [19]:

```
df = pd.read_csv('grades.csv')
```

In [20]:

```
df.head()
```

Out[20]:

	student_id	assignment1_grade	assignment1_submission	assignment2_grade	assignment2_submission	assignment3_g
0	B73F2C11-70F0-E37D-8B10-1D20AFED50B1	92.733946	2015-11-02 06:55:34.282000000	83.030552	2015-11-09 02:22:58.938000000	67.16
1	98A0FAE0-A19A-13D2-4BB5-CFBFD94031D1	86.790821	2015-11-29 14:57:44.429000000	86.290821	2015-12-06 17:41:18.449000000	69.77
2	D0F62040-CEB0-904C-F563-2F8620916C4E	85.512541	2016-01-09 05:36:02.389000000	85.512541	2016-01-09 06:39:44.416000000	68.41
3	FFDF2B2C-F514-EF7F-6538-A6A53518E9DC	86.030665	2016-04-30 06:50:39.801000000	68.824532	2016-04-30 17:20:38.727000000	61.94
4	5ECBEEB6-F1CE-80AE-3164-E45E99473FB4	64.813800	2015-12-13 17:06:10.750000000	51.491040	2015-12-14 12:25:12.056000000	41.93

In [21]:

```
len(df)
```

Out[21]:

```
2315
```

In [22]:

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

In [23]:

```
early.mean()
```

Out[23]:

```
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 [24]:

```
late.mean()
```

Out[24]:

```
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
```

In [25]:

```
from scipy import stats
stats.ttest_ind?
```

In [26]:

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

Out[26]:

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

In [27]:

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

Out[27]:

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

In [28]:

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

Out[28]:

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