Homework 1-1 附錄

November 4, 2021

[1]:
```
import numpy as np
import pandas as pd
import scipy.stats as ss
```

1 第一題

1.1 第 A 小題

$$P(78 \leq s \leq 101|p = 0.568) = \sum_{i=78}^{101} \binom{N}{i} 0.568^i (1 - 0.568)^{n-i}$$

$$P(78 \leq s \leq 101|p = 0.628) = \sum_{i=78}^{101} \binom{N}{i} 0.628^i (1 - 0.628)^{n-i}$$

[2]:
```
print(ss.binom.cdf(101, 150, 0.568) - ss.binom.cdf(77, 150, 0.568))
print(ss.binom.cdf(101, 150, 0.628) - ss.binom.cdf(77, 150, 0.628))
```

```
0.8942340977568481
0.8893559102960302
```

1.2 第 C 小題

$$P(633 \leq s \leq 697|p = 0.568) = \sum_{i=633}^{697} \binom{N}{i} 0.568^i (1 - 0.568)^{n-i}$$

$$P(633 \leq s \leq 697|p = 0.628) = \sum_{i=633}^{697} \binom{N}{i} 0.628^i (1 - 0.628)^{n-i}$$

[3]:
```
print(ss.binom.cdf(697, 1112, 0.568) - ss.binom.cdf(632, 1112, 0.568))
print(ss.binom.cdf(697, 1112, 0.628) - ss.binom.cdf(632, 1112, 0.628))
```

```
0.4791774849801923
0.4782405364695516
```

2 第二題

2.1 第 A 小題

2.1.1 資料前處理

[4]:
```
raw = pd.read_csv('466920chkd.txt', sep='\t')
raw.index = pd.date_range('1961-01-01 00', '2009-12-31 23', freq='H')
raw.columns = ['stn', 'Time', 'TX01']

July = raw[raw.index.month==7]['TX01'].values.reshape(49,31*24).T
idx1 = pd.date_range('1961-07-01 00', '1961-07-31 23', freq='D').strftime('%m-%d')
idx2 = pd.date_range('1961-07-01 00', '1961-07-01 23', freq='H').strftime('%H')
midx = pd.MultiIndex.from_product([idx1, idx2], names=['Date', 'Hour'])
col = np.arange(1961, 2010, 1).astype(str)
JulyTemp = pd.DataFrame(July, index=midx, columns=col)
JulyDMax = JulyTemp.groupby(level='Date').max()
```

[5]:
```
print('No. of obs: ', ss.describe(JulyDMax.values.flatten()).nobs)
print(' Min & Max: ', ss.describe(JulyDMax.values.flatten()).minmax)
print('      Mean: {:7.4f}'.format(ss.describe(JulyDMax.values.flatten()).mean))
print('  Variance: {:7.4f}'.format(ss.describe(JulyDMax.values.flatten()).variance))
print('  Skewness: {:7.4f}'.format(ss.describe(JulyDMax.values.flatten()).skewness))
print('  Kurtosis: {:7.4f}'.format(ss.describe(JulyDMax.values.flatten()).kurtosis+3))
```

```
No. of obs:  1519
 Min & Max:  (24.1, 38.0)
      Mean: 33.6221
  Variance:  3.7008
  Skewness: -1.0385
  Kurtosis:  5.0056
```

2.1.2 資料直方圖等機率區間與個數

[6]:
```
mean = ss.describe(JulyDMax.values.flatten()).mean
std = ss.describe(JulyDMax.values.flatten()).variance**0.5
equiprob = []
for i in range(0,38):
    if i == 0:
        equiprob.append(ss.describe(JulyDMax.values.flatten()).minmax[0])
    elif i == 37:
        equiprob.append(ss.describe(JulyDMax.values.flatten()).minmax[1])
    else:
        equiprob.append(ss.norm.ppf(i/37, loc=mean, scale=std))
hist, bin_edges = np.histogram(JulyDMax.values.flatten(), bins=equiprob)
print('{:^10}{:^6}   {:^14}{:^1}'.format('區間','個數','區間','個數'))
for i in range(19):
    if i < 18:
        print('{0:5.2f}~{1:5.2f}: {2:4d}   |   {3:5.2f}~{4:5.2f}: {5:4d}'\
              .format(bin_edges[i],bin_edges[i+1],hist[i],
                      bin_edges[i+19],bin_edges[i+20],hist[i+19]))
    else:
        print('{0:5.2f}~{1:5.2f}: {2:4d}'\
              .format(bin_edges[i],bin_edges[i+1],hist[i]))
```

```
    區間       個數          區間       個數
24.10~29.92:   82   |   33.69~33.82:   77
29.92~30.53:   26   |   33.82~33.95:   31
30.53~30.93:   20   |   33.95~34.08:   42
30.93~31.24:   22   |   34.08~34.22:   73
31.24~31.50:   28   |   34.22~34.36:   31
31.50~31.73:   16   |   34.36~34.50:   42
31.73~31.93:   24   |   34.50~34.65:   72
31.93~32.11:   34   |   34.65~34.80:   42
32.11~32.28:   23   |   34.80~34.96:   52
32.28~32.44:   36   |   34.96~35.13:   63
32.44~32.60:   16   |   35.13~35.32:   52
32.60~32.75:   46   |   35.32~35.52:   44
32.75~32.89:   29   |   35.52~35.74:   41
32.89~33.03:   72   |   35.74~36.00:   49
33.03~33.16:   41   |   36.00~36.31:   42
33.16~33.29:   34   |   36.31~36.71:   40
33.29~33.43:   71   |   36.71~37.33:   21
33.43~33.56:   38   |   37.33~38.00:   10
33.56~33.69:   37
```

2.1.3 常態分布相同區間內預期個數

[7]:
```
nobs = ss.describe(JulyDMax.values.flatten()).nobs
expected = []
for i in range(37):
    lower = bin_edges[i]
    upper = bin_edges[i+1]
    prob = ss.norm.cdf(upper, mean, std) - ss.norm.cdf(lower, mean, std)
    expected.append(np.round(nobs * prob, 1))

print('{:^10}{:^6}   {:^14}{:^1}'.format('區間','個數','區間','個數'))
for i in range(19):
    if i < 18:
        print('{0:5.2f}~{1:5.2f}: {2:4.0f}   |   {3:5.2f}~{4:5.2f}: {5:4.0f}'\
              .format(bin_edges[i],bin_edges[i+1],expected[i],
                      bin_edges[i+19],bin_edges[i+20],expected[i+19]))
    else:
        print('{0:5.2f}~{1:5.2f}: {2:4.0f}'\
              .format(bin_edges[i],bin_edges[i+1],expected[i]))
```

```
    區間       個數          區間       個數
24.10~29.92:   41   |   33.69~33.82:   41
29.92~30.53:   41   |   33.82~33.95:   41
30.53~30.93:   41   |   33.95~34.08:   41
30.93~31.24:   41   |   34.08~34.22:   41
31.24~31.50:   41   |   34.22~34.36:   41
31.50~31.73:   41   |   34.36~34.50:   41
31.73~31.93:   41   |   34.50~34.65:   41
31.93~32.11:   41   |   34.65~34.80:   41
32.11~32.28:   41   |   34.80~34.96:   41
32.28~32.44:   41   |   34.96~35.13:   41
32.44~32.60:   41   |   35.13~35.32:   41
32.60~32.75:   41   |   35.32~35.52:   41
32.75~32.89:   41   |   35.52~35.74:   41
32.89~33.03:   41   |   35.74~36.00:   41
33.03~33.16:   41   |   36.00~36.31:   41
33.16~33.29:   41   |   36.31~36.71:   41
33.29~33.43:   41   |   36.71~37.33:   41
33.43~33.56:   41   |   37.33~38.00:   24
33.56~33.69:   41
```

2.1.4 Chi-square Test

[8]:
```
statistic, pvalue = ss.chisquare(hist, expected, ddof=2) # df = k-ddof-1
print('Chi-square Test Statistic: {:5.3f}'.format(statistic))
chi_square = ss.chi2.ppf(0.95, df=37-3)
print('Critical Value: {:5.3f}'.format(chi_square))
```

```
Chi-square Test Statistic: 290.079
Critical Value: 48.602
```

Test Statistic $>$ Critical Value $\Rightarrow$ Reject Null Hypothesis

2.2 第 B 小題

[9]:
```
LB = np.quantile(JulyDMax.loc[:,'1961':'2000'], 0.025, interpolation='midpoint')
UB = np.quantile(JulyDMax.loc[:,'1961':'2000'], 0.975, interpolation='midpoint')
lsLB = np.sum(JulyDMax.loc[:,'2001':'2009'].values.flatten() < LB)
grUB = np.sum(JulyDMax.loc[:,'2001':'2009'].values.flatten() > UB)
print('Lower Bound: {:4.1f}'.format(LB))
print('Upper Bound: {:4.1f}'.format(UB))
print('   Less than Lower Bound: {0:2d} ({1:.3f}%)'.format(lsLB, lsLB/279*100))
print('Greater than Upper Bound: {0:2d} ({1:.3f}%)'.format(grUB, grUB/279*100))
```

```
Lower Bound: 28.4
Upper Bound: 36.6
   Less than Lower Bound:  2 (0.717%)
Greater than Upper Bound: 10 (3.584%)
```

2.3 第 C 小題

[10]:
```
sim = ss.norm.rvs(loc=0, scale=1, size=(10000, 1519))
lsList = []
grList = []
for i in range(10000):
    lb = np.quantile(sim[i, :1240], 0.025, interpolation='midpoint')
    ub = np.quantile(sim[i, :1240], 0.975, interpolation='midpoint')
    lsList.append(np.sum(sim[i, 1240:] < lb)/279*100)
    grList.append(np.sum(sim[i, 1240:] > ub)/279*100)

lsLLB = np.quantile(lsList, 0.025, interpolation='midpoint')
lsLUB = np.quantile(lsList, 0.975, interpolation='midpoint')
grLLB = np.quantile(grList, 0.025, interpolation='midpoint')
grLUB = np.quantile(grList, 0.975, interpolation='midpoint')
print('小於 LB 的百分比門檻值： [{0:.3f}%, {1:.3f}%]'.format(lsLLB, lsLUB))
print('大於 UB 的百分比門檻值： [{0:.3f}%, {1:.3f}%]'.format(grLLB, grLUB))
```

```
小於 LB 的百分比門檻值： [0.717%, 4.659%]
大於 UB 的百分比門檻值： [0.717%, 4.659%]
```

2.4 第 E 小題

[11]:
```
data = JulyDMax.loc[:,'1961':'2000'].values.flatten()
print('No. of obs: ', ss.describe(data).nobs)
print(' Min & Max: ', ss.describe(data).minmax)
print('      Mean: {:7.4f}'.format(ss.describe(data).mean))
print('  Variance: {:7.4f}'.format(ss.describe(data).variance))
print('  Skewness: {:7.4f}'.format(ss.describe(data).skewness))
```

```
No. of obs:  1240
 Min & Max:  (24.1, 37.8)
      Mean: 33.5122
  Variance:  3.7085
  Skewness: -1.0765
```

[12]:
```
mean = ss.describe(data).mean
std  = ss.describe(data).variance**0.5
skew = ss.describe(data).skewness
equiprob = []
for i in range(0,38):
    if i == 0:
        equiprob.append(ss.describe(data).minmax[0])
    elif i == 37:
        equiprob.append(ss.describe(data).minmax[1])
    else:
        equiprob.append(ss.pearson3.ppf((37-i)/37, skew, mean, std))
hist, bin_edges = np.histogram(data, bins=equiprob)
print('{:^10}{:^6}   {:^14}{:^1}'.format('區間','個數','區間','個數'))
for i in range(19):
    if i < 18:
        print('{0:5.2f}~{1:5.2f}: {2:4d}   |   {3:5.2f}~{4:5.2f}: {5:4d}'\
              .format(bin_edges[i],bin_edges[i+1],hist[i],
                      bin_edges[i+19],bin_edges[i+20],hist[i+19]))
    else:
        print('{0:5.2f}~{1:5.2f}: {2:4d}'\
              .format(bin_edges[i],bin_edges[i+1],hist[i]))
```

```
    區間       個數          區間       個數
24.10~28.98:   41   |   33.91~34.03:   38
28.98~29.99:   29   |   34.03~34.15:   24
29.99~30.60:   28   |   34.15~34.27:   31
30.60~31.05:   21   |   34.27~34.38:   23
31.05~31.41:   23   |   34.38~34.50:   36
31.41~31.71:   24   |   34.50~34.61:   60
31.71~31.98:   20   |   34.61~34.72:   31
31.98~32.21:   48   |   34.72~34.84:   27
32.21~32.42:   34   |   34.84~34.95:   16
32.42~32.61:   38   |   34.95~35.07:   25
32.61~32.79:   17   |   35.07~35.20:   20
32.79~32.95:   51   |   35.20~35.33:   39
32.95~33.11:   77   |   35.33~35.46:   19
33.11~33.26:   29   |   35.46~35.61:   40
33.26~33.40:   28   |   35.61~35.77:   13
33.40~33.53:   62   |   35.77~35.96:   23
33.53~33.66:   34   |   35.96~36.21:   23
33.66~33.79:   26   |   36.21~37.80:   62
33.79~33.91:   60
```

[13]:
```
nobs = ss.describe(data).nobs
expected = []
for i in range(37):
    lower = bin_edges[i]
    upper = bin_edges[i+1]
    prob = ss.pearson3.cdf(upper,skew,mean,std) - ss.pearson3.cdf(lower,skew,mean,std)
    expected.append(np.round(nobs * prob, 1))

print('{:^10}{:^6}   {:^14}{:^1}'.format('區間','個數','區間','個數'))
for i in range(19):
    if i < 18:
        print('{0:5.2f}~{1:5.2f}: {2:4.0f}   |   {3:5.2f}~{4:5.2f}: {5:4.0f}'\
              .format(bin_edges[i],bin_edges[i+1],expected[i],
                      bin_edges[i+19],bin_edges[i+20],expected[i+19]))
    else:
        print('{0:5.2f}~{1:5.2f}: {2:4.0f}'\
              .format(bin_edges[i],bin_edges[i+1],expected[i]))
```

```
    區間       個數            區間       個數
24.10~28.98:   33   |   33.91~34.03:   34
28.98~29.99:   34   |   34.03~34.15:   34
29.99~30.60:   34   |   34.15~34.27:   34
30.60~31.05:   34   |   34.27~34.38:   34
31.05~31.41:   34   |   34.38~34.50:   34
31.41~31.71:   34   |   34.50~34.61:   34
31.71~31.98:   34   |   34.61~34.72:   34
31.98~32.21:   34   |   34.72~34.84:   34
32.21~32.42:   34   |   34.84~34.95:   34
32.42~32.61:   34   |   34.95~35.07:   34
32.61~32.79:   34   |   35.07~35.20:   34
32.79~32.95:   34   |   35.20~35.33:   34
32.95~33.11:   34   |   35.33~35.46:   34
33.11~33.26:   34   |   35.46~35.61:   34
33.26~33.40:   34   |   35.61~35.77:   34
33.40~33.53:   34   |   35.77~35.96:   34
33.53~33.66:   34   |   35.96~36.21:   34
33.66~33.79:   34   |   36.21~37.80:   34
33.79~33.91:   34
```

2.4.1 Chi-square Test

[14]:
```
statistic, pvalue = ss.chisquare(hist, expected, ddof=3) # df = k-ddof-1
print('Chi-square Test Statistic: {:5.3f}'.format(statistic))
chi_square = ss.chi2.ppf(0.95, df=37-4)
print('Critical Value: {:5.3f}'.format(chi_square))
```

```
Chi-square Test Statistic: 246.674
Critical Value: 47.400
```

[15]:
```
sim = ss.pearson3.rvs(skew, mean, std, size=(10000, 1519))
lsList = []
grList = []
for i in range(10000):
    lb = np.quantile(sim[i, :1240], 0.025, interpolation='midpoint')
    ub = np.quantile(sim[i, :1240], 0.975, interpolation='midpoint')
    lsList.append(np.sum(sim[i, 1240:] < lb)/279*100)
    grList.append(np.sum(sim[i, 1240:] > ub)/279*100)
```

```
lsLLB = np.quantile(lsList, 0.025, interpolation='midpoint')
lsLUB = np.quantile(lsList, 0.975, interpolation='midpoint')
grLLB = np.quantile(grList, 0.025, interpolation='midpoint')
grLUB = np.quantile(grList, 0.975, interpolation='midpoint')
print('小於 LB 的百分比門檻值： [{0:.3f}%, {1:.3f}%]'.format(lsLLB, lsLUB))
print('大於 UB 的百分比門檻值： [{0:.3f}%, {1:.3f}%]'.format(grLLB, grLUB))
```

```
小於 LB 的百分比門檻值： [0.717%, 4.659%]
大於 UB 的百分比門檻值： [0.717%, 4.659%]
```