科目名：プログラミング技法 II　　担当教員名：新田　直子
学籍番号：01Z18032　　名前：トゥチェク・トム

2019 年 07 月 15 日

第三回：統計－χ^2検定・適合度検定・πシミュレーション・大数の法則・就職活動問題

課題 5：適合度検定

課題５－１では、２回目課題の 19 枚目スライドと同じく、ヒストグラムのグラフを作る。カイ二乗検定は、ウィキペディアからの公式で計算する。１０のビンに分別し、正規分布が期待度数にし、計算し、M 回繰り返し、プロットする。

図 1　ピアソンのカイ二乗検定の公式、出典：https://en.wikipedia.org/wiki/Pearson%27s_chi-squared_test

Calculating the test-statistic [edit]

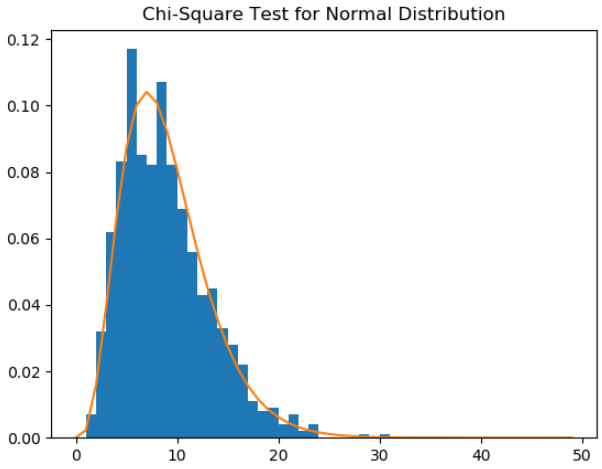
The value of the test-statistic is

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} = N \sum_{i=1}^{n} \frac{(O_i/N - p_i)^2}{p_i}$$

where

χ^2 = Pearson's cumulative test statistic, which asymptotically approaches a χ^2 distribution.
O_i = the number of observations of type i.
N = total number of observations
$E_i = Np_i$ = the expected (theoretical) count of type i, asserted by the null hypothesis that the fraction of type i in the population is p_i
n = the number of cells in the table.

図 2　課題５－１の結果

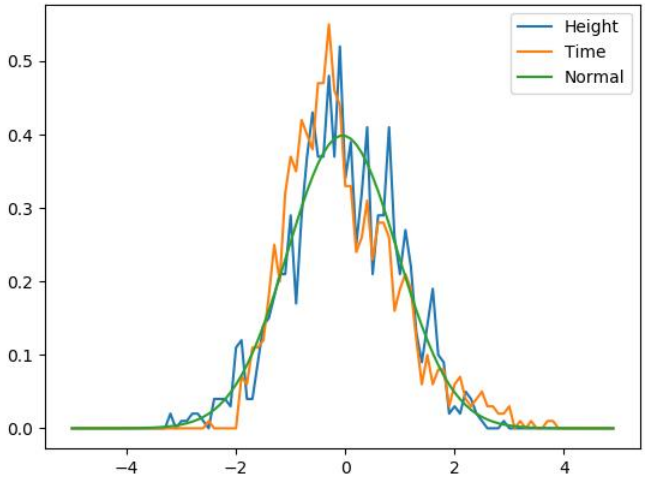


課題５－２では、私は困っていた。５－１で作った関数を使っても、scipy.stats.chisquare の関数を使っても、なかなか正しそうな結果が出てこなかった。使うデータは大丈夫かどうかを確認するようにプロットにしたが、無理だった。結局、諦めた。下記、出力とグラフを載せた。

表１　課題５－２の出力

```
-- Results of scipy chisquare test --
Power_divergenceResult(statistic=0.9031952939510415, pvalue=1.0)
Power_divergenceResult(statistic=2.14724435398335, pvalue=1.0)
-- Results of own chi square test --
13.72777566373827
65.72199113525267
```

図３　抽出されたデータは正規分布になんとか従うかどうか確認するために作ったグラフ



課題６：πを求める（シミュレーション）

課題６－１では、ランダムの点で、πをシミュレーションした。原点からの距離を計算するように、まず numpy.linalg.norm の関数を用いたが、演算が遅すぎたので、ピタゴラスの定理で自分の関数を作った。それで、結果が十分なスピードで出てきた。

話題６－２では、多様な点数 N で、前のアルゴリズムを M 回繰り返して、それの平均・標準偏差・95%信頼区間を計算し、グラフで示す。求められた「推定値πが正規分布に従うか課題

5-2 と同様に判定せよ。」という部分は、課題５－２がちゃんとできなかった原因で、それもできなかった。それでも、結果的にグラフと出力は大丈夫そうだった。

図４　課題６－２のグラフ、パイのシミュレーション

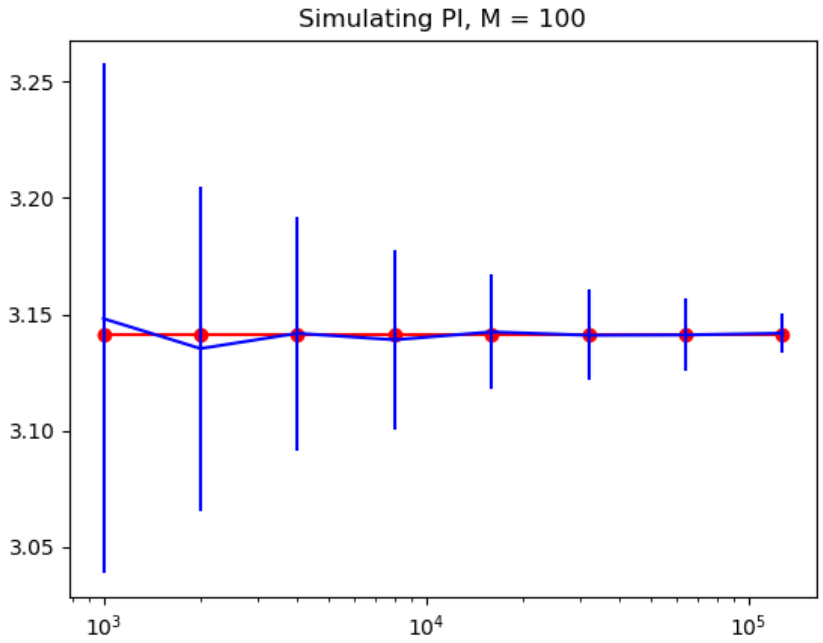


表２　課題６－２の出力　（点数・平均・標準偏差）

```
Points: 1000, mean: 3.1482, stdev: 0.055560686631735405
Points: 2000, mean: 3.13534, stdev: 0.03521369683013196
Points: 4000, mean: 3.14189, stdev: 0.02526261860912083
Points: 8000, mean: 3.1391, stdev: 0.01926988019755917
Points: 16000, mean: 3.142485, stdev: 0.012316318115378158
Points: 32000, mean: 3.14106625, stdev: 0.009586436503511538
Points: 64000, mean: 3.1412025, stdev: 0.007656655807633356
Points: 128000, mean: 3.1419196875, stdev: 0.003977270428615856
```

課題７：大数の法則と中心極限定理

課題７－１では、だいたい課題６－２と同じようなグラフを作った。選手を N 個、ランダムに抽出し、標本平均を計算し、M 回繰り返し、その標本平均の平均と標準偏差を算出した。また、課題６－２と同じく、課題５－２に基づく部分を飛ばした。

計算されたデータを、母平均とともにグラフに示した。関係について考えたら、N を増えると、精度がよくなるということが明らかになった。だが、その改善にある逓減もある。すなわち、N は大きくなると、標準偏差はもうあまり変化しない。

図 5　課題７－１のグラフ、標本平均の平均・標準偏差

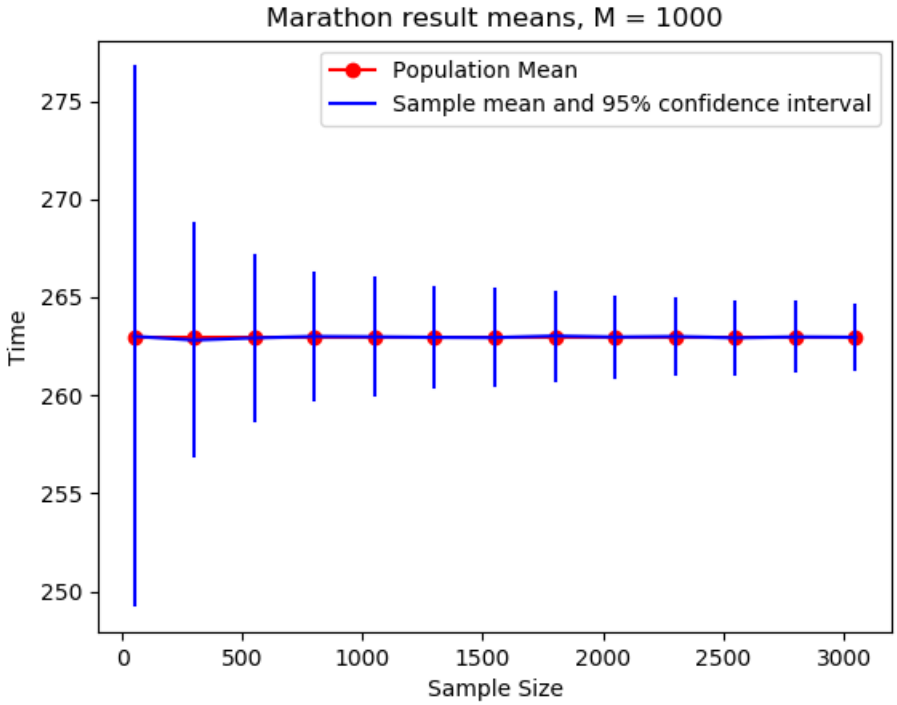
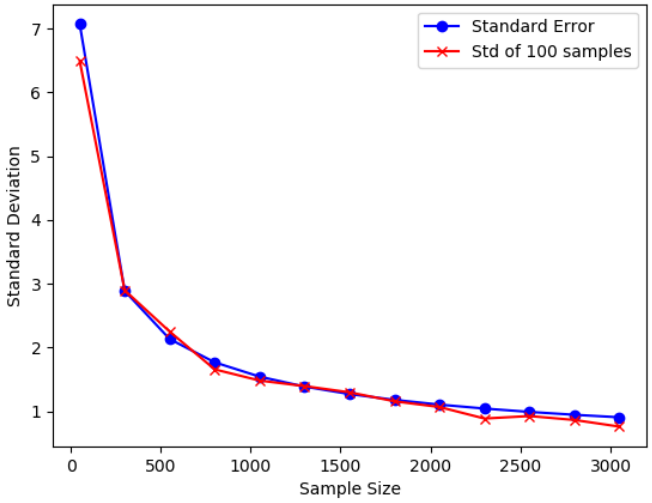


表 3　課題７－１の出力

Samples: 50, mean: 263.0066246, stdev: 7.000681512821362
Samples: 300, mean: 262.8142261333333, stdev: 3.0162576547807953
Samples: 550, mean: 262.9258973090909, stdev: 2.1524903125413344
Samples: 800, mean: 263.000103475, stdev: 1.658005188134862
Samples: 1050, mean: 262.9815420095238, stdev: 1.51059439953779
Samples: 1300, mean: 262.95379996153844, stdev: 1.277003438996385
Samples: 1550, mean: 262.9429430709677, stdev: 1.2605096777441105
Samples: 1800, mean: 263.0268313388889, stdev: 1.1418069591193505
Samples: 2050, mean: 262.97027996585365, stdev: 1.0530479246338265
Samples: 2300, mean: 262.9981461565217, stdev: 0.9833552054490209
Samples: 2550, mean: 262.93128724705883, stdev: 0.9297411816565845
Samples: 2800, mean: 262.9781665107143, stdev: 0.8976593128719663
Samples: 3050, mean: 262.9588393180328, stdev: 0.8214261935117839

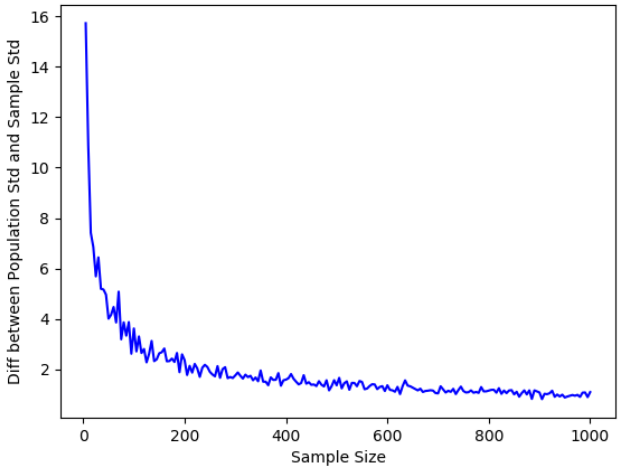
課題７－２では、標準誤差を計算し、前の標本平均の標準偏差とともにグラフで示した。比較すれば、結構似ているということが明らかになる。

図６　課題７－２のグラフ、標本平均の標準偏差・標準誤差の比較



課題７－３では、課題７－２の比較（差）の計算をＭ回繰り返し、グラフでそれの平均を示した。きれいなグラフを作るように、Ｎは５から１０００、ステップサイズ＝５に決めた。

図７　課題７－３のグラフ



課題７－４では、ランダムでN＝１００個の選手を抽出し、平均と95%信頼区間を計算し、それをM＝100000回繰り返し、母平均がその信頼区間に入らない率を出力した。約５％となるはずと書いてあり、約５％となった。信頼区間を計算するために、今度は「平均±1.96標準偏差」という公式が効かなかったので、scipy.stats.intervalという関数を用いた。

表４　課題７－４の出力　（三回の実行）

Ratio of misses: 5.1819999999999995 %
Ratio of misses: 5.067 %
Ratio of misses: 5.08 %

課題８：就職活動問題

課題８では、カードゲームのシミュレーションを行った。N＝１００個のカードで、Mをどう設定すれば勝率が最大となるか、3000回の繰り返しで、シミュレーションした。グラフに示し、最大の勝率とその値を出力する。繰り返しの数を大きくすれば、グラフはよりスムーズになる。

図８　課題８のグラフ、就職活動問題のシミュレーションの勝率

表５　課題８の出力、就職活動問題のシミュレーションの最大勝率

Highest winning percentage at: 32/100 with 39.6%
Highest winning percentage at: 36/100 with 38.86666666666667%