さいころのシミュレーション

疑似さいころを100回/1000回振ってさいころとして妥当か考えてみよう

疑似さいころの検証(モデル)

- 乱数を使ってさいころを振る
- 出現回数を調べ妥当性を考える

例:さいころを100回振ってみると

- 乱数を使ってさいころを振る

- Excelでシミュレーション

- Pythonでシミュレーション

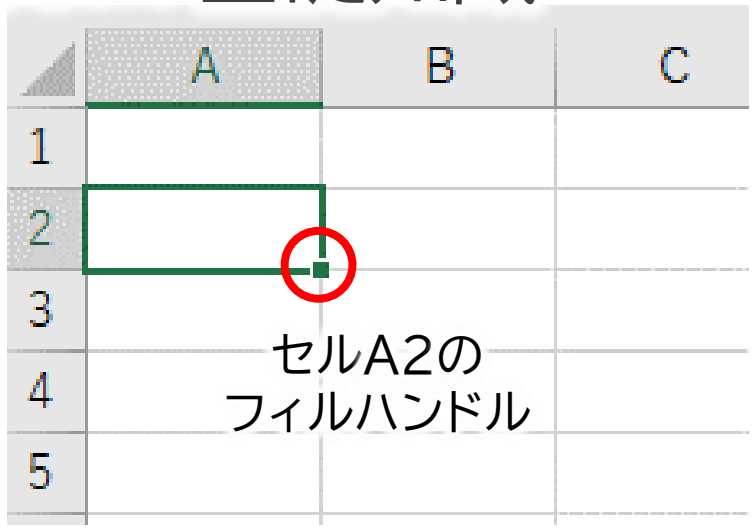
例:さいころを100回振ってみると

Excelでシミュレーション

- 手順
- 1. 表の作成
- 2. さいころの式を作る
- 3. 出目のカウント
- 4. グラフ作成
- 5. 数値を変えてみる

- 手順の説明は次ページへ

Excelの基礎知識



おかしいときは直そうとせず
[Esc]→[Ctrl]+[Z]で
やらなかったことに

例:さいころを100回振ってみると

Excelでシミュレーション

- 1. 表の作成
  - セルB1に　出現回数　と入力
  - セルA2に　１　と入力
  - セルA3に　２　と入力
  - セルA2からセルA3を
    ドラッグして範囲選択したら、
    フィルハンドルをセルA7まで
    ドラッグし、6まで作る
  - セルA10に　目　と入力
  - saikoroというファイル名で自分のドライブに保存

日本語入力は
こまめに
切っておく

	A	B
1		出現回数
2	1	
3	2	
4	3	
5	4	
6	5	
7	6	
8		
9		
10	目	
11		

完成イメージ

☆重要

例:さいころを100回振ってみると

Excelでシミュレーション

- 2. さいころの式を作る
  - セルA11に　=randbetween(1,6）と入力
  - セルA11のフィルハンドルを下にドラッグして
    セルA110までドラッグする

数式は
小文字で
入力できる

- RANDBETWEEN(開始,終了)
  - 開始値から終了値の整数による乱数を発生する

☆重要

例:さいころを100回振ってみると

Excelでシミュレーション

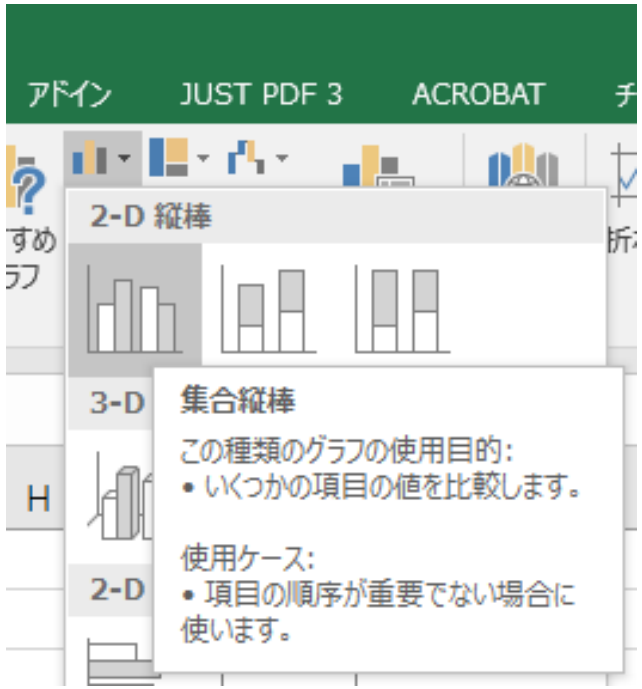
- 3. 出目のカウント
  - セルB2に　=countif(a11:a110,a2)を入力
  - セルB2のフィルハンドルをダブルクリック

- COUNTIF(範囲,条件)
  - 範囲の中から条件に合うものをカウントする

例:さいころを100回振ってみると

Excelでシミュレーション

- 4. グラフ作成
  - セルA1～B7をドラッグ
  - [挿入]→[グラフ]→[縦棒グラフ]→[集合縦棒]



例:さいころを100回振ってみると

Excelでシミュレーション

- 4. グラフ作成-検討
- 何回か実行する
  - セルA1をクリック
  - [F2]キーを押して[Enter]キーを押す

- 結果のデータや度数分布を見て、さいころとして妥当か考えてみよう

例:さいころを100回振ってみると

Excelでシミュレーション

- 5. 数値を変えてみる
  - さいころを1000回振るとどうなるか
    - セルB2～B7の式は修正が必要

- 結果のデータや度数分布を見て、さいころとして妥当か考えてみよう
- 妥当かどうか判断する方法を今後数学で学ぶはず

さいころを100回振ってみると
Excel版　終了
次の課題へ>>

例:さいころを100回振ってみると

Pythonでシミュレーション

- 手順
- 1. プログラムの作成
- 2. 出目のカウント
- 3. グラフ作成のために改造
- 4. グラフ表示プログラムの追加
- 5. 数値を変えて実行

- 手順の説明は次ページへ

例:さいころを100回振ってみると

Pythonでシミュレーション

- 1. プログラムの作成
  1. 新しいノートブックを用意
  2. 以下のプログラムを入力
  3. 入力できたら実行して動作確認
     - [4, 6, 2, 6, 1, 1, 3, 2, 3, 3]のように表示される
  4. 100回に改造

```
import random
saikoro=[]
for i in range(10):
    saikoro.append(random.randint(1,6))
print(saikoro)
```

☆重要

例:さいころを100回振ってみると

Pythonでシミュレーション

- 1. プログラムの作成-プログラムの意味

```
import random
saikoro=[]
for i in range(10):
    saikoro.append(random.randint(1,6))
print(saikoro)
```

numpyにnpという呼び名を付けた

- 100回に改造する

モジュールrandom をインポート
リストsaikoroを用意
10回繰り返し
　　1から6の整数の乱数をsaikoroに追加
saikoroを表示

例:さいころを100回振ってみると

- 2. 出目のカウント
  - 以下の部分を追加して実行し動作確認

```
import random
saikoro=[]
for i in range(100):
    saikoro.append(random.randint(1,6))
print(saikoro)
deme=[]
for j in range(6):
    deme.append(saikoro.count(j+1))
print(deme)
```

☆重要

例:さいころを100回振ってみると

Pythonでシミュレーション

- 2. 出目のカウント-プログラムの意味

```
deme=[]
for  j  in range(6):
     deme.append(saikoro.count(j+1))
print(deme)
```

リストdemeを用意
jを0から6未満まで変えながら繰り返す
　　リストsaikoroにあるj+1を数えdemeに代入
demeを表示

- リスト
  複数のデータを入れる変数(実はsaikoroもリスト)

例:さいころを100回振ってみると

- 3. グラフ作成のために改造
  - 以下の部分を追加して実行し動作確認

```
import random
import matplotlib.pyplot as plt
saikoro=[]
for i in range(100):
    saikoro.append(random.randint(1,6))
print(saikoro)
deme=[]
for j in range(6):
    deme.append(saikoro.count(j+1))
print(deme)
```

☆重要

例:さいころを100回振ってみると

Pythonでシミュレーション

- 3. グラフ作成のために改造-プログラムの意味

```
import matplotlib.pyplot as plt
```

モジュールmatplotlib.pyplotをpltとしてインポート

- プログラム中に何度もmatplotlib.pyplotと打つのは面倒なので、plt という略称をつけておく

例:さいころを100回振ってみると

Pythonでシミュレーション

- 4. グラフ表示プログラムの追加
  - 以下の部分を追加して実行し動作確認

```
left=[1,2,3,4,5,6]
plt.title('Saikoro Simulation')
plt.xlabel('Me')
plt.ylabel('Kaisu')
plt.bar(left,deme,align='center')
plt.show()
```

```
import random
import matplotlib.pyplot as plt
saikoro=[]
for i in range(100):
    saikoro.append(random.randint(1,6))
print(saikoro)
deme=[]
for j in range(6):
    deme.append(saikoro.count(j+1))
print(deme)
left=[1,2,3,4,5,6]
plt.title('Saikoro Simulation')
plt.xlabel('Me')
plt.ylabel('Kaisu')
plt.bar(left,deme,align='center')
plt.show()
```

完成イメージ

例:さいころを100回振ってみると

Pythonでシミュレーション

- 4. グラフ表示プログラムの追加-検討

- 何回か実行する
- 結果のデータや度数分布を見て、さいころとして妥当か考えてみよう

例:さいころを100回振ってみると

Pythonでシミュレーション

- 5. 数値を変えて実行
- さいころを1000回振るとどうなるか

- 結果のデータや度数分布を見て、さいころとして妥当か考えてみよう
- 妥当かどうか判断する方法を今後数学で学ぶはず

さいころを100回振ってみると
Python版　終了
次の課題へ>>

ガチャのシミュレーション

ガチャ100回やったら当たるのか？

ガチャの検証(モデル)

- 排出確率1%のガチャ
  - 1～100までの整数の乱数を発生
  - あたりの数を1つ決めてシミュレーション

- ガチャを100回やったらどれぐらい当たるか
- 100人が100回やったらどれぐらい当たるか

例:ガチャを100回やると

- 排出確率1%のガチャ
  - 1～100までの整数の乱数を発生
  - 当たりの数を1つ決めてシミュレーション

- Excelでシミュレーション

- Pythonでシミュレーション

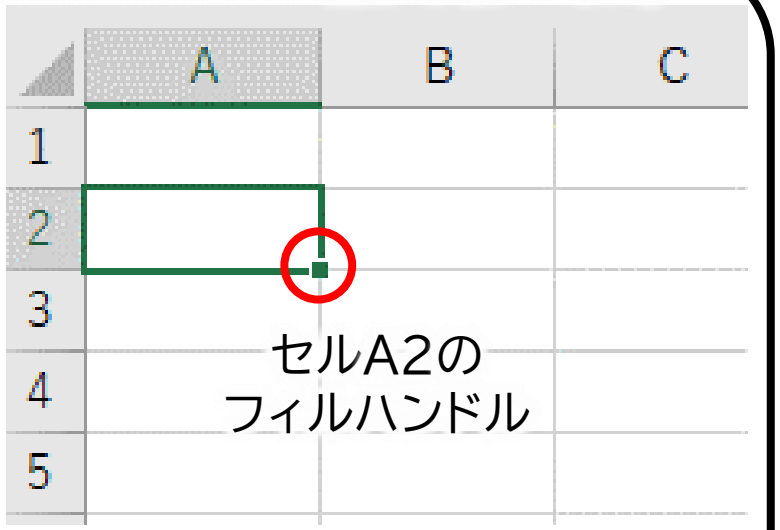
例:ガチャを100回やると

Excelでシミュレーション

- 手順
- 1. 表の作成
- 2. ガチャの式を作る
- 3. 当たりの判定
- 4. 100人分実行
- 5. 排出回数、当選人数をカウント
- 6. 数値を変えて実行

- 手順の説明は次ページへ

Excelの基礎知識



おかしいときは直そうとせず
[Esc]→[Ctrl]+[Z]で
やらなかったことに

例:ガチャを100回やると

Excelでシミュレーション

- 1. 表の作成
  - セルA1に　当たり　と入力
  - セルA3に　排出回数　と入力
  - セルA4に　当選人数　と入力
  - セルB10に 1,セルC10に 2と入力
  - セルB10からセルC10を
    ドラッグして範囲選択したら、
    フィルハンドルをセルCW10まで
    右にドラッグし、100まで作る
  - gacha　というファイル名で自分のドライブに保存

日本語入力は
こまめに
切っておく

完成イメージ

	A	B	C	D	E
1	当たり				
2					
3	排出回数				
4	当選人数				
5					
6					
7					
8					
9					
10		1	2	3	4
11					

☆重要

例:ガチャを100回やると

Excelでシミュレーション

- 2. ガチャの式を作る
  - セルB11に　=randbetween(1,100）と入力
  - セルB11のフィルハンドルを右にドラッグして
    セルCW11までドラッグする

数式は
小文字で
入力できる

- RANDBETWEEN(開始,終了)
  - 開始値から終了値の整数による乱数を発生する

☆重要

例:ガチャを100回やると

Excelでシミュレーション

- 3. 当たりの判定
  - セルB1に　50　を入力
    - 当選番号を50にする/別の数にしても良い
  - セルA11に　＝countif(b11:cw11,b1)を入力

- COUNTIF(範囲,条件)
  - 範囲の中から条件に合うものをカウントする

例:ガチャを100回やると

Excelでシミュレーション

- 4. 100人分実行
  - セルA11からセルCW11をドラッグして範囲選択
  - 選択範囲右下のフィルハンドルを下にドラッグし、CW110までドラッグする

- ガチャ100回100人分の表が完成

例:ガチャを100回やると

Excelでシミュレーション

- 5. 排出回数、当選人数をカウント
  - セルB3に　=sum(a11:a110）を入力
  - セルB4に　=countif(a11:a110,”>0”）を入力

- SUM(範囲)
  - 範囲にある数値の合計
- COUNTIF(範囲,条件)
  - 範囲の中から条件に合うものをカウントする

例:ガチャを100回やると

Excelでシミュレーション

- 5. 排出回数、当選人数をカウント-検討
- 何回か実行する
  - セルA1をクリック
  - [F2]キーを押して[Enter]キーを押す

- 結果のデータを見て考えよう
  - 思っていたようなデータか
  - ガチャとして妥当か

例:ガチャを100回やると

Excelでシミュレーション

- 6. 数値を変えて実行
- 100回のガチャを1000人でやるとどうなるか
  - セルB3,セルB4の式は修正が必要

- 結果のデータを見て考えよう
  - 思っていたようなデータか
  - ガチャとして妥当か

ガチャを100回やると
Excel版　終了
次の課題へ>>

例:ガチャを100回やると

Pythonでシミュレーション

- 手順
- 1. プログラムの作成
- 2. 当たりの判定
- 3. 100人分のために改造
- 4. 100人分実行
- 5. 排出回数、当選人数をカウント
- 6. 数値を変えて実行

- 手順の説明は次ページへ

例:ガチャを100回やると

Pythonでシミュレーション

- 1. プログラムの作成
  1. 新しいノートブックを用意
  2. 以下のプログラムを入力
  3. 入力できたら実行して動作確認
  4. 100回やるよう改造

```
import random
gacha=[]
for i in range(10):
    gacha.append(random.randint(1,100))
print(gacha)
```

☆重要

例:ガチャを100回やると

Pythonでシミュレーション

- 1. プログラムの作成-プログラムの意味

```
import random
gacha=[]
for i in range(10):
    gacha.append(random.randint(1,100))
print(gacha)
```

- 100回に改造する

モジュールrandomをインポート
リストgachaを用意
10回繰り返し
　　1～100までの乱数をgachaに追加
gachaを表示

例:ガチャを100回やると

- 2. 当たりの判定
  - 以下の部分を追加して実行し動作確認

```
import random
atari=50
gacha=[]
for i in range(100):
    gacha.append(random.randint(1,100))
print(gacha)
tousen=gacha.count(atari)
print(tousen)
```

☆重要

例:ガチャを100回やると

Pythonでシミュレーション

- 2. 当たりの判定-プログラムの意味
  - 当選番号は1～100の間で変えてもよい

当選番号を
50にする

```
import random
atari=50
gacha=[]
for i in range(100):
    gacha.append(random.randint(1,100))
print(gacha)
tousen=gacha.count(atari)
print(tousen)
```

tousenを
表示する

tousen にgachaにあるatari
の個数(排出回数)を代入

例:ガチャを100回やると

- 3.100人分のために改造
  - 以下の部分を修正して実行し動作確認

```
import random
atari=50
def gacha100():
    gacha=[]
    for i in range(100):
        gacha.append(random.randint(1,100))
    tousen=gacha.count(atari)
    return tousen

print(gacha100())
```

☆重要

例:ガチャを100回やると

Pythonでシミュレーション

- 3.100人分のために改造-プログラムの意味
  - gacha100()関数を定義/排出回数を返す

関数gacha100()を定義

関数gachaの戻り値として当たりの回数を返す

関数gacha100()を呼び出し結果を表示

```
import random
atari=50
def gacha100():
    gacha=[]
    for i in range(100):
        gacha.append(random.randint(1,100))
    tousen=gacha.count(atari)
    return tousen

print(gacha100())
```

例:ガチャを100回やると

Pythonでシミュレーション

- 4. 100人分実行
  - print(gacha100())を消して
  - 以下の部分を追加して実行し動作確認

```
kekka=[]
for  i  in  range(100):
      kekka.append(gacha100())

print(kekka)
```

```
1 import random
2 atari=50
3 def gacha100():
4     gacha=[]
5     for i in range(100):
6         gacha.append(random.randint(1,100))
7     tousen=gacha.count(atari)
8     return tousen
9
10 kekka=[]
11 for i in range(100):
12     kekka.append(gacha100())
13 print(kekka)
```

完成イメージ

☆重要

例:ガチャを100回やると

Pythonでシミュレーション

- 4. 100人分実行-プログラムの意味
  - gacha100()を100回実行し排出回数をリストkekkaに代入する

```
kekka=[]
for  i  in range(100):
      kekka.append(gacha100())

print(kekka)
```

空のリスト kekka を用意

100回繰り返す

gacha100()の結果をリストkekkaに追加

- リスト
  複数のデータを入れる変数(実はgachaもリスト)

例:ガチャを100回やると

Pythonでシミュレーション

- 5. 排出回数、当選人数をカウント
  - kekka＝[]以降を修正して実行し動作確認

```
kekka=[]
hit=0;luck=0
for  i  in range(100):
     kekka.append(gacha100())
     hit=hit+kekka[i]
     if kekka[i]>0:
          luck=luck+1
print(kekka)
print('hit',hit,'luck',luck)
```

完成イメージ

```
1 import random
2 atari=50
3 def gacha100():
4     gacha=[]
5     for i in range(100):
6         gacha.append(random.randint(1,100))
7     tousen=gacha.count(atari)
8     return tousen
9
10 kekka=[]
11 hit=0;luck=0
12 for i in range(100):
13     kekka.append(gacha100())
14     hit=hit+kekka[i]
15     if kekka[i]>0:
16         luck=luck+1
17 print(kekka)
18 print('hit',hit,'luck',luck)
```

☆重要

例:ガチャを100回やると

Pythonでシミュレーション

- 5. 排出回数、当選人数をカウント-プログラムの意味
  - kekka＝[]以降を修正して実行し動作確認

変数
hit(排出回数)
luck(当選人数)
を0にする

```
kekka=[]
hit=0;luck=0
for  i  in range(100):
      kekka.append(gacha100())
      hit=hit+kekka[i]
      if kekka[i]>0:
            luck=luck+1
print(kekka)
print('hit',hit,'luck',luck)
```

変数hitに
排出回数を加える

排出回数＞0なら
（当選したなら）
変数luckに1を加える

変数
hit(排出回数)
luck(当選人数)
を表示する

例:ガチャを100回やると

Pythonでシミュレーション

- 5. 排出回数、当選人数をカウント-検討
- 何回か実行する

- 結果のデータを見て考えよう
  - 思っていたようなデータか
  - ガチャとして妥当か

例:ガチャを100回やると

Pythonでシミュレーション

- 6. 数値を変えて実行
- 100回のガチャを1000人でやるとどうなるか

- 結果のデータを見て考えよう
  - 思っていたようなデータか
  - ガチャとして妥当か

ガチャを100回やると
Python版　終了
次の課題へ>>

時間があれば取り組もう

- それぞれの課題でまだやっていないものに取り組もう

- さいころのシミュレーション
  - Excelでシミュレーション
  - Pythonでシミュレーション
- ガチャのシミュレーション
  - Excelでシミュレーション
  - Pythonでシミュレーション