

令和5年度神奈川県情報部会主催
「情報科実践事例報告会」

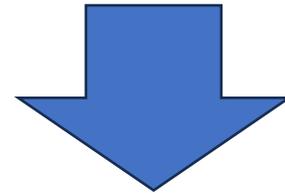
ChatGPTを活用した
「情報Ⅰ」 大学入学共通テスト
対策問題制作の実践報告

帝京大学大学院 理工学研究科 修士課程 植垣 新一

1年で定期テストレベルから共通テストレベルまで
500問以上作問 (書籍執筆+企業委託関連など)

共通テストレベル

アイデア出し→整合性のとれた問題作成 時間を要する



産みの苦しみ



ChatGPT活用で産みの苦しみが大幅に軽減



今日は何をお手伝いしましょうか？

情報Ⅰ 大学入学共通テスト対策 参考書＋特典問題集 会話型テキストと動画でよくわかる (インプレス)



植垣 新一(著)
稲垣 俊介(監修)

オールカラー 288ページ
本はすべてインプット教材(全13教科書を徹底分析)
＋問題集(約200ページ) を特典提供(PDF/HTML)

全編解説動画付き！ 作問事例を公表

※詳細は概要欄にリンクを記載

1月18日発売予定
インプレス 1980円 (税込)

講義形式で学ぶ

2024年4月 刊行予定

「情報I」大学入学共通テスト 問題集

能城茂雄 (東京都立三鷹中等教育学校)

著者 秋吉祐樹 (大分県立中津南高等学校)

植垣新一 (情報教育人気 YouTuber)

予価 1,100円 (本体1,000円+税10%)

体裁 B5判・2色刷・192ページ (予定)

一番最初に解いてほしい問題集!

共通テスト対策には、過去問の学習が欠かせません。大学入試センターから公開された3回の問題を学習することは、共通テストに向けた準備に非常に役立ちます。

この問題集を学習することで、各問題のポイント、詳細な解説を通じて、「情報I」を深く理解することができます。

東京都立三鷹中等教育学校 能城茂雄



大学入試センター発表問題

- ・ 試作問題解説
- ・ サンプル問題解説
- ・ 検討用イメージ解説
- オリジナル問題
- ・ 演習問題
- ・ 実践問題

今回の発表部分

全編動画解説付き!



「問題」で 要点確認

「解説」で 深く理解

<p>Point 1</p> <p>3回分の問題を徹底解説</p> <p>大学入試センターから公開された全3回分の試作問題、サンプル問題をもとに、情報科教育のエキスパートが講義形式で丁寧に解説しました。</p>	<p>Point 2</p> <p>領域別編集で教科の内容を定着</p> <p>学習指導要領の4領域構成に従って問題を分類。1年次の学習内容を体系的に復習したり、苦手な分野を集中的に対策したりするのに便利です。</p>	<p>Point 3</p> <p>演習と実践問題で実力アップ</p> <p>各章末に、その領域の内容を共通テスト形式で確認できる演習問題を設置。さらに巻末には実践形式の予想問題も収録しています。</p>	<p>Point 4</p> <p>わかりやすい解説動画を完備</p> <p>演習問題・実践問題を含むすべての問題に解説動画をご用意。生徒の自学・自習にもご活用いただけます。</p>
--	--	---	--

サンプルはこちら!

実践事例 シミュレーションプログラム作成



ChatGPT活用前

作問アイデアが浮かんでも
プログラムの作成が困難

数時間～数日の時間を要する

産みの苦しき…



ChatGPT活用後 数分に短縮



ChatGPT

正六面体のサイコロを同時に500回振り、サイコロの出目の合計数をグラフ化するプログラムを以下に示します。このプログラムでは、NumPyとMatplotlibを使用します。

python

Copy code

```
import numpy as np
import matplotlib.pyplot as plt

# 同時に振るサイコロの数
dice_count = 500

# サイコロを振る回数
rolls = 500

# サイコロの出目を記録するリスト
sum_of_rolls = []

# サイコロを振るシミュレーション
for _ in range(rolls):
    # 同時にサイコロを振る
    dice_rolls = np.random.randint(1, 7, dice_count)
    # 出目の合計を計算し、リストに追加
    sum_of_rolls.append(np.sum(dice_rolls))
```

実践事例 シミュレーションプログラム作成

■第2問 B

B 次の文章を読み、後の問い（問1～3）に答えよ。

ケンジさんは文化祭の実行委員をしている。今度の文化祭では、ビンゴ大会を行う予定である。ビンゴ大会を行うのははじめてであり、ビンゴカードのマス数や、割り振る数字の範囲によって、ゲームの時間が変わってくるのではないかと考えた。ビンゴ大会の時間は限られているため、最適なビンゴカードのマス数や数字の範囲をあらかじめ決めておく必要がある。そこで、**ビンゴカードのマス数や使う数字の範囲を変更した場合、何回目で完了する人が多いのかのシミュレーションを行うことにした。**

問1 次の文章の空欄【オ】【カ】に当てはまる数字を答えよ。また、【キ】にあてはまる文章を解答群の中から一つ選べ。

ビンゴゲームのルールは、以下の通りである。

- ・マス目にランダムに番号が振られたカードを参加者に配布する。
- ・司会者はランダムに番号を1つずつ発表していき、参加者はカードに該当する番号があった場合、その番号をマークする。
- ・早く縦・横・斜めのいずれか連ねた者が勝ち（以下、ビンゴという）となる。

まず、ケンジさんは卓上でシミュレーションしてみることにした。図1のような3×3のカードを用意し、1～20までのいずれかの数字をランダムに割り振った。このマス数の場合、最短【オ】個目の数字が発表された時点でビンゴとなる。

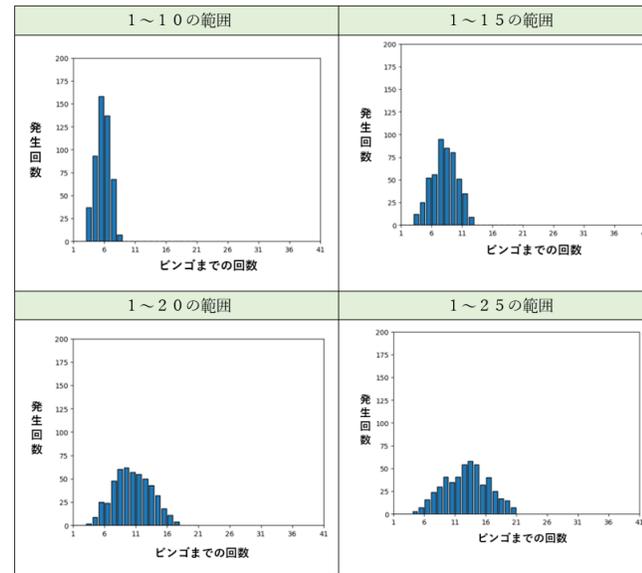
9	20	1
18	5	8
11	13	7

図1 ビンゴカード

※初稿の原稿なので
体裁はこれから整います。

問2

次に3×3のマスのままで、数字の範囲を変更した場合、どのようにシミュレーション結果が変化するかを知りたいと考えた。数字の範囲の最大値を10、15、20、25とした場合のシミュレーションをそれぞれ500回行った。図3はその結果である。



問3

次に数字の範囲と同時に、マス目の数を変更した場合にどのようなかを検証することにした。

数字の範囲を1～30、1～40にしたものとマス目を4×4、5×5にしたもののパターンを組み合わせて一覧にしたものである。シミュレーション結果について空欄【ケ】～【サ】にあてはまる最も適当なグラフをそれぞれ①～③のうちから1つずつ選びなさい。

	数字の範囲1～30	数字の範囲1～40
マス目 4×4		【ケ】
マス目 5×5	【コ】	【サ】

実践事例 シミュレーションプログラム作成

【ChatGPTへの依頼①】（指示概要）

3×3マスのビンゴゲームをするとき20個の数字を用いて、何回でビンゴするかをシミュレーションするプログラム作成して

ChatGPT

3×3マスのビンゴゲームをシミュレートするPythonプログラムを作成することができます。ビンゴゲームでは、通常1から75までの数字が使用されますが、あなたの要件に合わせて1から20までの数字を使用することができます。

以下は、3×3マスのビンゴカードを作成し、ランダムに数字を選んでビンゴが成立するまでの回数をプリントするシンプルなプログラムの例です。

```
python Copy code

import random
import numpy as np

def generate_bingo_card():
    card = np.zeros((3, 3), dtype=int)
    numbers = random.sample(range(1, 21), 9)
    for i in range(3):
        for j in range(3):
            card[i, j] = numbers[i * 3 + j]
    return card

def check_bingo(card, drawn_numbers):
    for row in card:
        if set(row) <= set(drawn_numbers):
            return True
    for col in card.T:
        if set(col) <= set(drawn_numbers):
            return True
    if set(np.diag(card)) <= set(drawn_numbers) or set(np.diag(np.fliplr(card))) <= set(drawn_numbers):
        return True
    return False
```

```
def simulate_bingo():
    card = generate_bingo_card()
    print("ビンゴカード:")
    print(card)
    drawn_numbers = []
    for _ in range(20):
        number = random.choice([n for n in range(1, 21) if n not in drawn_numbers])
        drawn_numbers.append(number)
        if check_bingo(card, drawn_numbers):
            print("ビンゴ!")
            print("引かれた数字:", drawn_numbers)
            print("ビンゴまでの回数:", len(drawn_numbers))
            break
    else:
        print("ビンゴせず!")
        print("引かれた数字:", drawn_numbers)

if __name__ == "__main__":
    simulate_bingo()
```

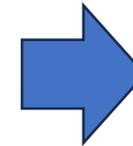
このプログラムでは、ビンゴカードは3x3のNumPy配列で表され、1から20までの数字がランダムに配置されます。ビンゴが成立するまでランダムに数字を選び、その数字を`drawn_numbers`リストに追加していきます。ビンゴが成立したかどうかは`check_bingo`関数でチェックされ、ビンゴが成立した場合はその時点でプログラムが終了します。

プログラムを実行すると、ビンゴカードと引かれた数字、ビンゴまでの回数が表示されます。

Googleコラボラトリーで実行

```
simulate_bingo()
```

```
ビンゴカード:
[[ 6 17 10]
 [12 16 14]
 [ 3 15  2]]
ビンゴ!
引かれた数字: [7, 17, 18, 10, 13, 16, 14, 3]
ビンゴまでの回数: 8
```



意図した結果となるかを十分に検証する。

→ChatGPTに再度問いかける。

※実際のやり取りを公開しています。（概要欄参照）

実践事例 シミュレーションプログラム作成

【ChatGPTへの依頼②】（指示概要）

今のプログラムをベースに、500回シミュレーションを行って何回目でビンゴしたかの分布を作成したい

ChatGPT

500回のビンゴゲームのシミュレーションを行い、何回目でビンゴが成立したかの分布を作成するために、プログラムを修正します。この結果をヒストグラムで表示することができます。

まず、ビンゴが成立するまでの回数を記録するために、`simulate_bingo`関数を修正してビンゴが成立するまでの回数を返すようにします。

次に、500回のシミュレーションを行い、結果をリストに保存します。最後に、そのリストを使用してヒストグラムを描画します。

以下は、修正されたプログラムの例です。

```
python Copy code

import random
import numpy as np
import matplotlib.pyplot as plt

def generate_bingo_card():
    card = np.zeros((3, 3), dtype=int)
    numbers = random.sample(range(1, 21), 9)
    for i in range(3):
        for j in range(3):
            card[i, j] = numbers[i * 3 + j]
    return card

def check_bingo(card, drawn_numbers):
    for row in card:
        if set(row) <= set(drawn_numbers):
            return True
    for col in card.T:
        if set(col) <= set(drawn_numbers):
            return True
    if set(np.diag(card)) <= set(drawn_numbers) or set(np.diag(np.fliplr(card))) <= set(drawn_numbers):
        return True
    return False
```

```
def simulate_bingo():
    card = generate_bingo_card()
    drawn_numbers = []
    for turn in range(1, 21):
        number = random.choice([n for n in range(1, 21) if n not in drawn_numbers])
        drawn_numbers.append(number)
        if check_bingo(card, drawn_numbers):
            return turn
    return 20 # ビンゴが成立しなかった場合

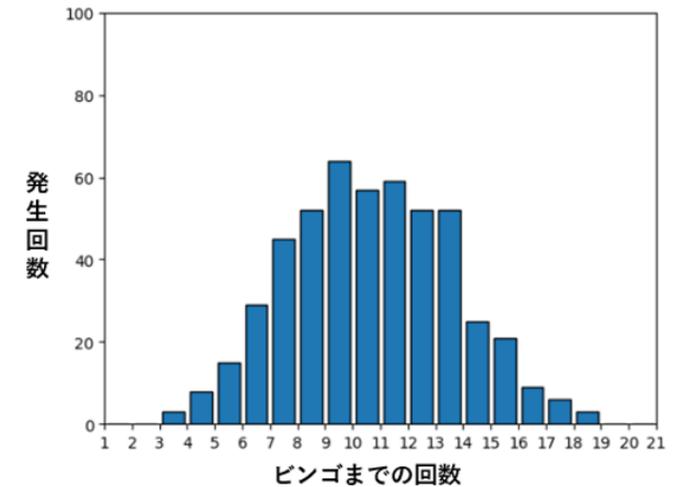
if __name__ == "__main__":
    results = [simulate_bingo() for _ in range(500)]

    plt.hist(results, bins=range(1, 22), edgecolor='k', rwidth=0.8)
    plt.xlabel('ビンゴまでの回数')
    plt.ylabel('頻度')
    plt.title('ビンゴまでの回数の分布 (500回シミュレーション)')
    plt.show()
```

このプログラムを実行すると、500回のビンゴゲームのシミュレーションが行われ、ビンゴが成立するまでの回数の分布がヒストグラムで表示されます。ヒストグラムのx軸はビンゴまでの回数を、y軸はその回数でビンゴが成立したゲームの数を表しています。

matplotlibがインストールされていない場合は、以下のコマンドでインストールしてください。

Googleコラボラトリーで実行



意図した結果となるかを十分に検証する。

※実際のやり取りを公開しています。（概要欄参照）

問題が完成しました！



実践事例 モデル化とシミュレーションの共通テストレベル問題

■第2問 B

B 次の文章を読み、後の問い（問1～3）に答えよ。

ケンジさんは文化祭の実行委員をしている。今度の文化祭では、ビンゴ大会を行う予定である。ビンゴ大会を行うのははじめてであり、ビンゴカードのマス数や、割り振る数字の範囲によって、ゲームの時間が変わってくるのではないかと考えた。ビンゴ大会の時間は限られているため、最適なビンゴカードのマス数や数の範囲をあらかじめ決めておく必要がある。そこで、ビンゴカードのマス数や使う数の範囲を変更した場合、何回目まで完了する人が多いかのシミュレーションを行うことにした。

問1 次の文章の空欄【オ】【カ】に当てはまる数字を答えよ。また、【キ】にあてはまる文章を解答群の中から一つ選べ。

ビンゴゲームのルールは、以下の通りである。

- ・マス目にランダムに番号が振られたカードを参加者に配布する。
- ・司会者はランダムに番号を1つずつ発表していき、参加者はカードに該当する番号があった場合、その番号をマークする。
- ・早く縦・横・斜めのいずれか連ねた者が勝ち（以下、ビンゴという）となる。

まず、ケンジさんは卓上でシミュレーションしてみることにした。図1のような3×3のカードを用意し、1～20までのいずれかの数字をランダムに割り振った。このマス数の場合、最短【オ】個目の数字が発表された時点でビンゴとなる。

9	20	1
18	5	8
11	13	7

図1 ビンゴカード

仮に、司会者が、以下の順で番号を発表したとする。

7→19→14→18→8→12→3→20→1→13→6→5→17→9→4→11→10→16→15→2

この場合、図1のカードは【カ】個目の数字が発表された時点でビンゴとなる。

次にケンジさんは、3×3のカードで1～20までのいずれかの数字の範囲をランダムに割

り振った場合についてシミュレーションプログラムを作成した。

プログラムを500回実行した結果の分布は次のようになった。横軸がビンゴまでの回数で、縦軸はその発生回数である。

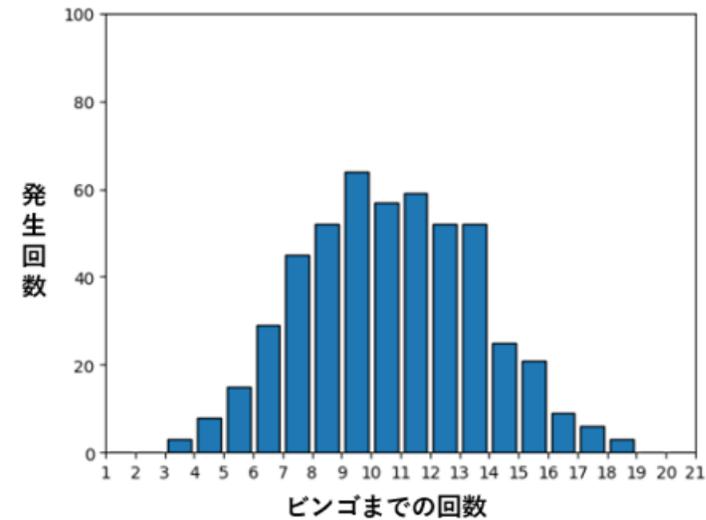


図2 シミュレーション結果

これより【キ】ということが読み取れる。

【キ】の解答群

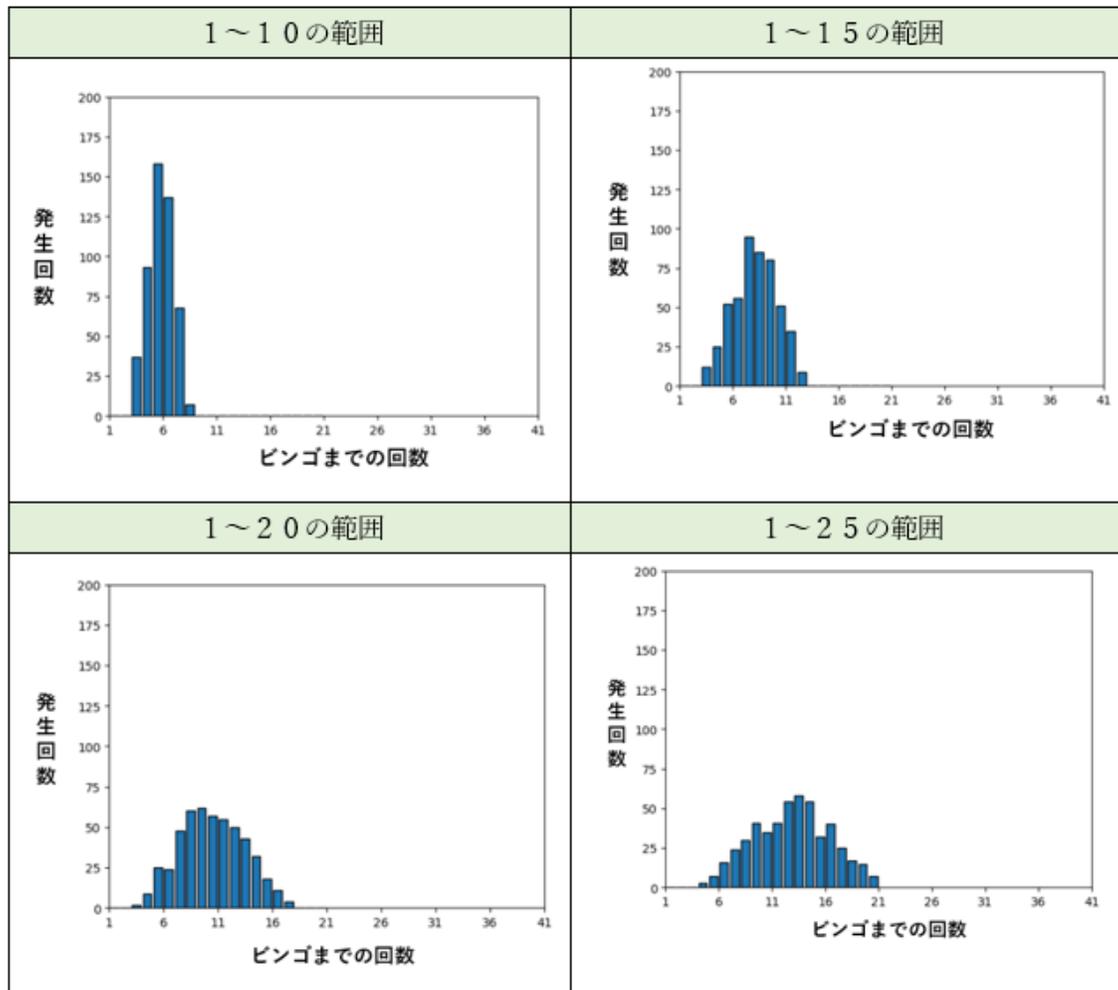
- ① 少なくとも5回、最後の20個目の数字でビンゴしている
- ② 11回目でビンゴした割合が最も多い
- ③ 6回目でビンゴする割合は、15回目より多い。
- ④ もう一度シミュレーションした場合は、発生回数が80回を超えるものが必ず現れる

実践事例 モデル化とシミュレーションの共通テストレベル問題

問2

次に 3×3 のマスそのまま、数字の範囲を変更した場合、どのようにシミュレーション結果が変化するかを知りたいと考えた。

数の範囲の最大値を10、15、20、25とした場合のシミュレーションをそれぞれ500回行った。図3はその結果である。



この例の場合において、シミュレーション結果から読みとれないことを次の①～③のうちから一つ選べ。【ク】

【ク】の解答群

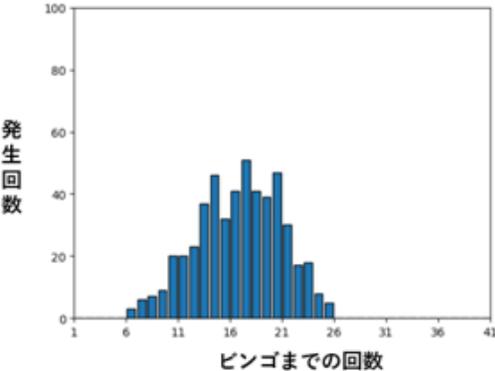
- ① 数字の範囲が大きくなるにしたがって、ビンゴまでの回数のばらつきも大きくなる傾向がある。
- ② 1～10の範囲では、過半数が7回以内にビンゴすることが推測できる。
- ③ 最頻値に注目すると数字の範囲が大きくなるにしたがって、ビンゴまでの回数も多くなる傾向にある。
- ④ 数字の範囲を広くするほど、最も早くビンゴする人の割合も多くなると推測できる。

実践事例 モデル化とシミュレーションの共通テストレベル問題

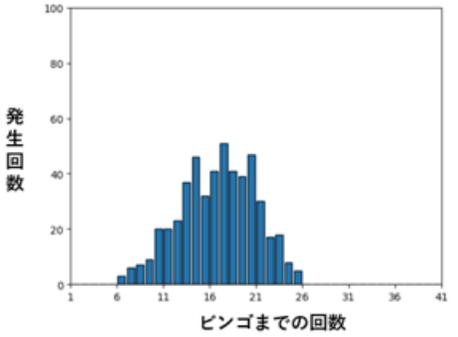
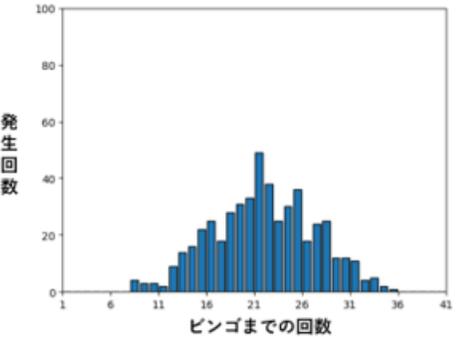
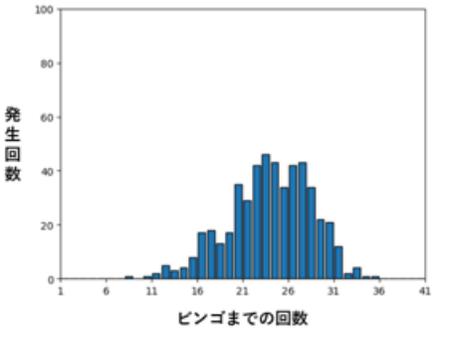
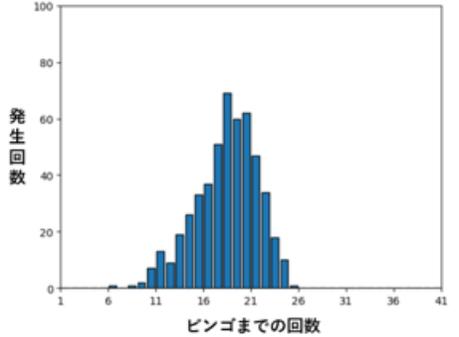
問3

次に数字の範囲と同時に、マス目の数を変更した場合にどのようなかを検証することにした。

数字の範囲を1～30、1～40にしたものとマス目を4×4、5×5にしたもののパターンを組み合わせ一覧にしたものである。シミュレーション結果について空欄【ケ】～【サ】にあてはまる最も適切なグラフをそれぞれ①～③のうちから1つずつ選びなさい。

	数字の範囲1～30	数字の範囲1～40
マス目 4×4		【ケ】
マス目 5×5	【コ】	【サ】

【解答群】

①	②	③
		
		

大学入学共通テスト対策

情報Ⅰ

植垣新一 著 稲垣俊介 監修

The Common Test for University Admissions Examination



会話型
テキストと
動画で
よくわかる

得点力がつく!

プログラミング問題もこわくない!

本書の3大特長

- 公開資料と全13教科書の分析で見えた要点を、**完全網羅!**
- 超基礎から語りかける授業形式の解説で、**挫折ゼロ!**
- 共通テスト用プログラム表記の**解説が充実!**

豊富な特典
学習を強力サポート

インターネットで提供
スマホ対応

- 本書 ↔ 教科書対照表
 - 用語知識の定着を促す音声ファイル&暗記カードWebアプリ
 - 要点チェック問題/章のまとめ問題/総仕上げ問題
- 動画では実行環境「Python」を使って解説。「こう書けば、こう動く」がわかりやすい!
- インプレス

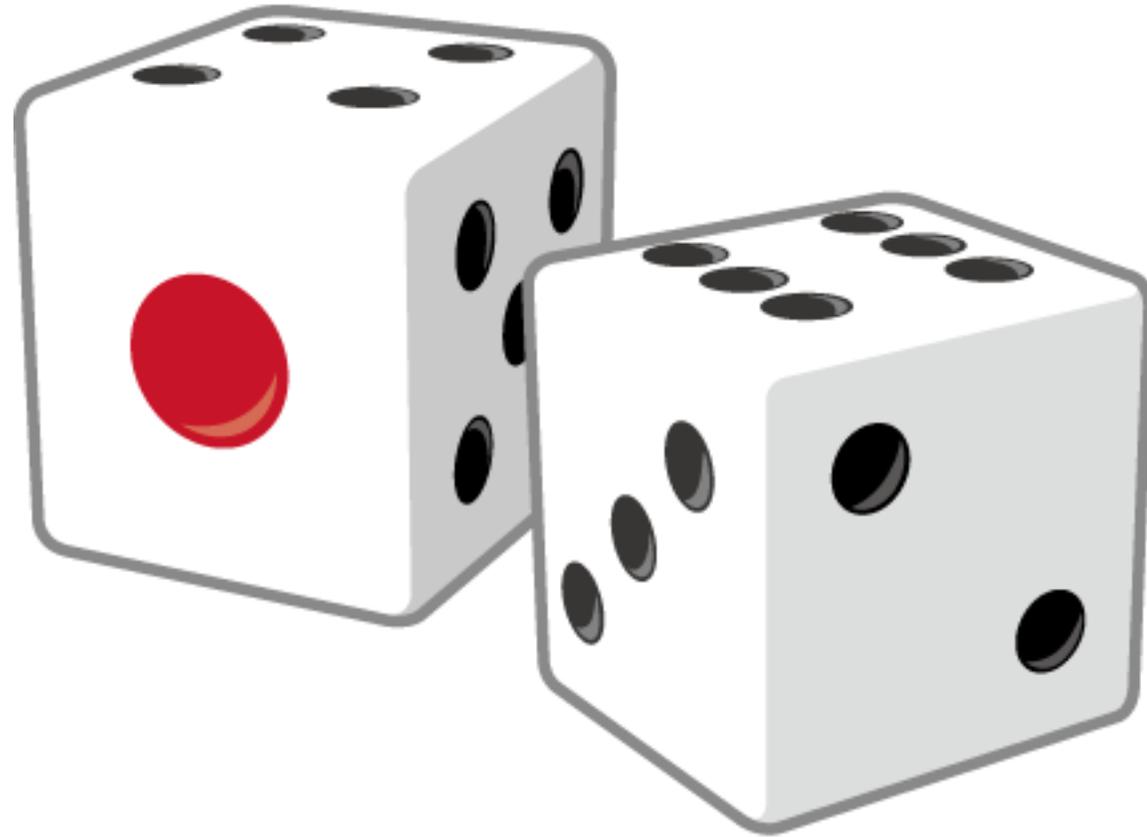
モデル化と シミュレーション関連

ChatGPTを活用した 作問事例

※問題集は特典として電子媒体（PDF、HTML）での提供

実践事例 モデル化とシミュレーションの確認テストレベル問題

2つの正六面体のサイコロを振る（合計を求める）。
それを500回繰り返した時の合計値の分布を表示する
シミュレーションプログラム



実践事例 モデル化とシミュレーションの確認テストレベル問題



匿名

正六面体のサイコロを同時に振ることを500回やった時のサイコロの出目の合計数をグラフ化するプログラム作成して



ChatGPT

正六面体のサイコロを同時に500回振り、サイコロの出目の合計数をグラフ化するプログラムを以下に示します。このプログラムでは、NumPyとMatplotlibを使用します。

```
python Copy code

import numpy as np
import matplotlib.pyplot as plt

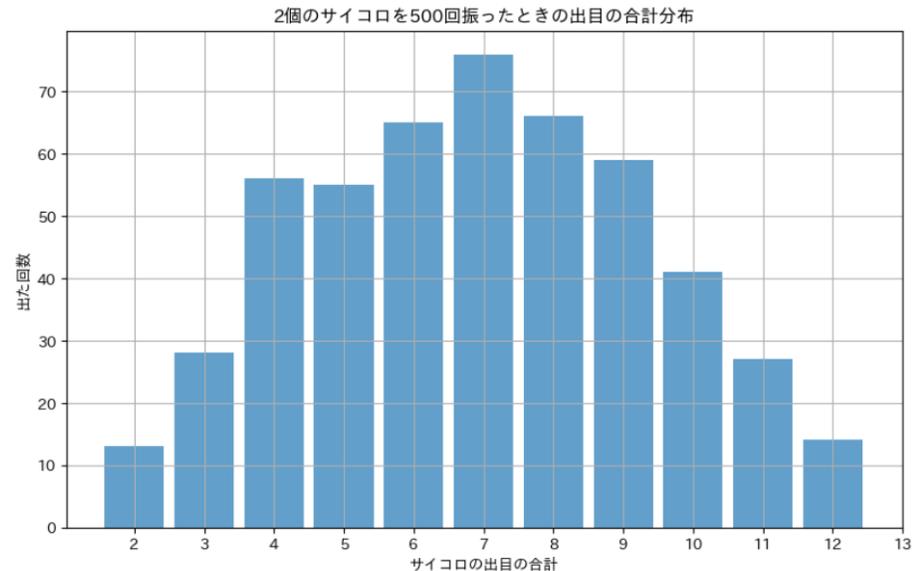
# 同時に振るサイコロの数
dice_count = 500

# サイコロを振る回数
rolls = 500

# サイコロの出目を記録するリスト
sum_of_rolls = []

# サイコロを振るシミュレーション
for _ in range(rolls):
    # 同時にサイコロを振る
    dice_rolls = np.random.randint(1, 7, dice_count)
    # 出目の合計を計算し、リストに追加
    sum_of_rolls.append(np.sum(dice_rolls))

# グラフの描画
plt.figure(figsize=(10, 6))
plt.hist(sum_of_rolls, bins=range(dice_count, 7 * dice_count + 2), dens
plt.xlabel('サイコロの出目の合計')
plt.ylabel('確率')
plt.title(f'{dice_count}個のサイコロを{rolls}回振ったときの出目の合計分布')
plt.grid(True)
plt.show()
```



このプログラムは、指定した回数のサイコロを同時に振って、出目の合計の分布をヒストグラムとして表示します。変数`dice_count`と`rolls`を調整することで、同時に振るサイコロの数とサイコロを振る回数を変更できます。

実践事例 モデル化とシミュレーションの確認テストレベル問題

【ChatGPTへの依頼②】（指示概要）
今のプログラムの結果を散布図形式で縦軸に1つ目のサイコロ、横軸に2つめのサイコロの出目とできる？



ChatGPT

もちろん、正六面体のサイコロを同時に2つ振った結果を散布図形式で表示することができます。以下はそのプログラム例です。

```
python Copy code

import numpy as np
import matplotlib.pyplot as plt

# 同時に振るサイコロの数
dice_count = 2

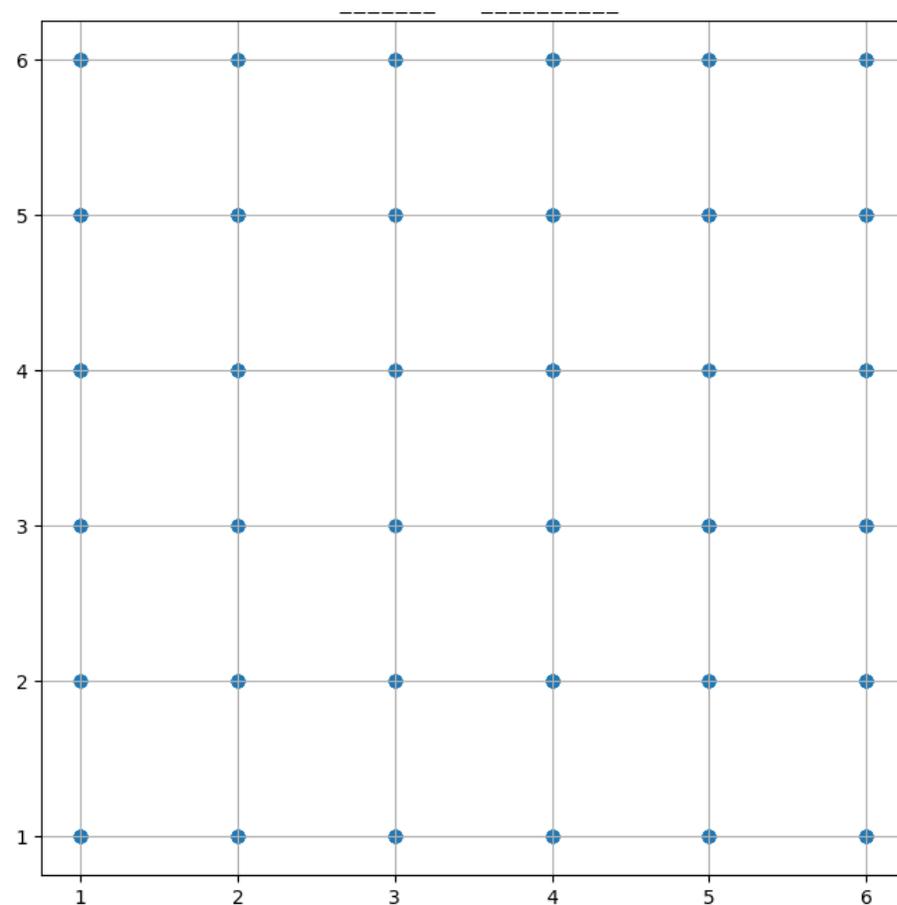
# サイコロを振る回数
rolls = 500

# サイコロの出目を記録するリスト
dice1_rolls = []
dice2_rolls = []

# サイコロを振るシミュレーション
for _ in range(rolls):
    # 同時にサイコロを指定の数だけ振る
    dice_rolls = np.random.randint(1, 7, dice_count)
    # 各サイコロの出目を記録
    dice1_rolls.append(dice_rolls[0])
    dice2_rolls.append(dice_rolls[1])

# 散布図の描画
plt.figure(figsize=(8, 8))
plt.scatter(dice2_rolls, dice1_rolls, alpha=0.5)
plt.xlabel('2つ目のサイコロの出目')
plt.ylabel('1つ目のサイコロの出目')
plt.title(f'{dice_count}個のサイコロを{rolls}回振った結果の散布図')
plt.grid(True)
plt.xticks(range(1, 7))
plt.yticks(range(1, 7))
plt.show()
```

このプログラムは、1つ目のサイコロの出目を縦軸、2つ目のサイコロの出目を横軸として、500回のサイコロ振りの結果を散布図形式で表示します。



実践事例 モデル化とシミュレーションの確認テストレベル問題

【ChatGPTへの依頼③】（指示概要） 出た回数によって、円の大きさ変えてくれる？

出た回数によって円の大きさを変えて散布図を表示するには、`scatter` メソッドの `s` パラメータを使用して、各データポイント（サイコロの組み合わせ）ごとに円の大きさを設定することができます。以下は、そのプログラム例です。

```
python Copy code

import numpy as np
import matplotlib.pyplot as plt

# 同時に振るサイコロの数
dice_count = 2

# サイコロを振る回数
rolls = 500

# サイコロの出目を記録するリスト
dice1_rolls = []
dice2_rolls = []

# サイコロの組み合わせごとの出現回数を記録する辞書
roll_counts = {}

# サイコロを振るシミュレーション
for _ in range(rolls):
    # 同時にサイコロを指定の数だけ振る
    dice_rolls = np.random.randint(1, 7, dice_count)
    # 各サイコロの出目を記録
    dice1_rolls.append(dice_rolls[0])
    dice2_rolls.append(dice_rolls[1])

# 出目の組み合わせを文字列に変換して辞書のキーとし、出現回数をカウント
roll_str = f"{dice_rolls[0]}-{dice_rolls[1]}"
if roll_str in roll_counts:
    roll_counts[roll_str] += 1
else:
    roll_counts[roll_str] = 1

# 散布図の描画
plt.figure(figsize=(10, 8))

# 各サイコロの組み合わせに対して、出現回数に応じて円の大きさを設定
sizes = [roll_counts[f"{dice1}-{dice2}"] * 10 for dice1, dice2 in zip(d

plt.scatter(dice2_rolls, dice1_rolls, s=sizes, alpha=0.5)
plt.xlabel('2つ目のサイコロの出目')
plt.ylabel('1つ目のサイコロの出目')
```

問題が完成しました！



良ければ次の画面の動画を止めて解いてみてください

■問題■

正六面体のサイコロを2つ振りその合計を求める処理を500回行うシミュレーションプログラムを作成した。図1はサイコロの出目の合計と回数の分布，図2は1つ目を振ったサイコロの出目と2つ目を振ったサイコロの出目の組み合わせで，円の大きさは出現回数に対応している。このシミュレーション結果について誤っているものを1つ選び，番号で答えなさい。

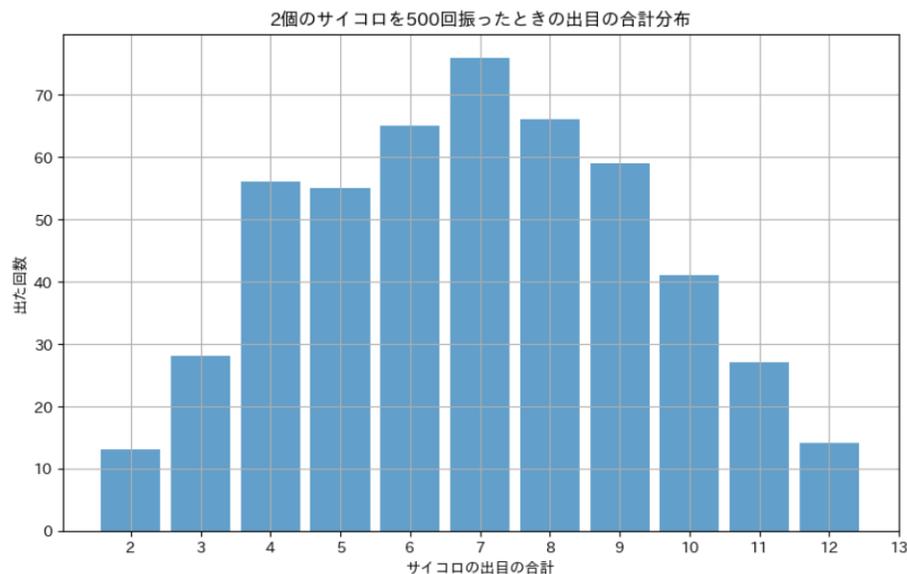


図1

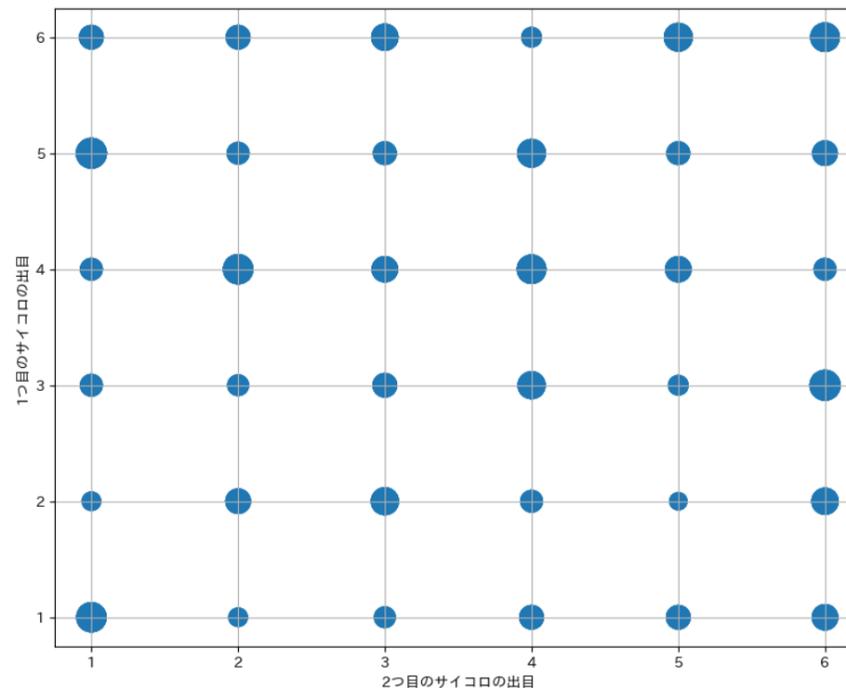
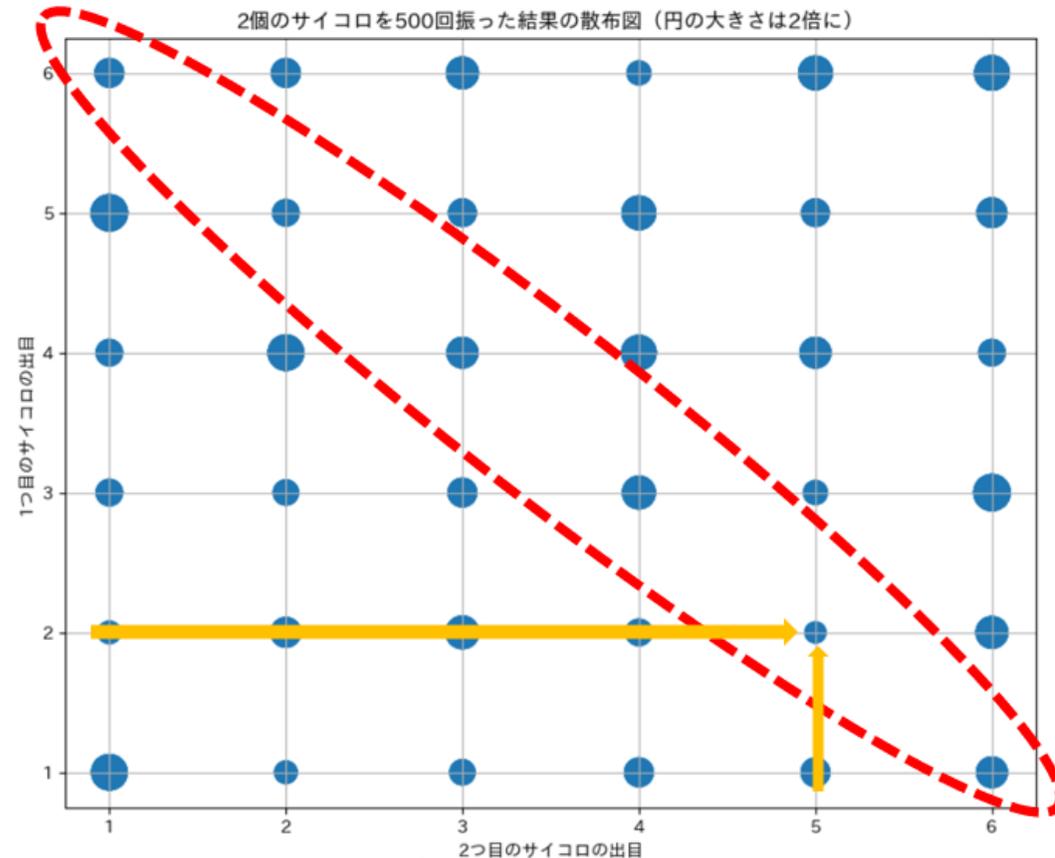


図2

- ① もう一度シミュレーションを行った場合は必ずしも同じ結果になるとは限らない。
- ② 1つ目で5が出た場合，2つ目で6が出たパターンが多いことが読み取れる。
- ③ サイコロの合計値は7になる確率が最も高いことが読み取れる。
- ④ 合計値が7になるパターンの中で，1つ目が2で，2つ目5が出るパターンが最も多い。

正解（誤りの内容を選ぶ問題）

→③ 合計値が7になるパターンの中で，1つ目が2で，2つ目5が出るパターンが最も多い。



赤い点線が，合計が7になるパターンですが，
1つ目が2で，2つ目5が出るパターンの●が他に比べて小さいことから，
そのパターンの割合は，他の合計が7になる組み合わせに比べて少ないことがわかります。
※500回程度ではばらつきが大きい。試行回数を増やせば，同じ●の大きさに近づく。

その他のChatGPT活用実践事例

- 問題の検証
- 解説作成の補助
- 類似事例の作成
- 将来的な期待

問題の検証

答えが一意に定まる必要がある。
作問時に別解を見つけられないことがある。 → ChatGPTに解いてもらう

SP

あなた

以下の問題を解いて解説して

Society 5.0の社会を生き抜くうえで身につけていく力として、誤っているものを1つ選び、番号で答えなさい。

- ④ 学校の授業で得た知識を日常生活で積極的に活用して問題解決していく力
- ① 新たに生まれてくるサービスに興味を持ち、活用していく力
- ② 問題を発見し、その解決のためにIT技術を駆使して、解決を目指す力
- ③ 今のやり方が最適と考え、新たに生まれてくる技術を否定する力

設定

×

④ 一般

チャット履歴とトレーニング

このブラウザでの新しいチャットを履歴に保存し、モデルの改善に使用できるようにします。保存されていないチャットは、30日以内にシステムから削除されます。この設定はブラウザやデバイス間で同期されません。詳しくはこちら

🔊 ベータ機能

🔒 データ制御

共有されたリンク

管理

👤 ビルダープロフィール

データをエクスポート

エクスポート

アカウントを削除

削除

🌀

ChatGPT

Society 5.0は、テクノロジーと社会が融合する未来を指しており、この社会を生き抜くためには特定のスキルやマインドセットが求められます。上記の選択肢の中で、Society 5.0の社会に適応するために必要な力として誤っているものを選ぶと、答えは「③ 今のやり方が最適と考え、新たに生まれてくる技術を否定する力」になります。

解説：

- ④ 学校の授業で得た知識を日常生活で積極的に活用して問題解決していく力: Society 5.0では、教育で得た知識を日常生活や実世界の問題解決に応用する能力が重要です。
- ① 新たに生まれてくるサービスに興味を持ち、活用していく力: 新しい技術やサービスに対して柔軟にアプローチし、それらを積極的に活用する姿勢は、変化の速い社会に適応するために不可欠です。
- ② 問題を発見し、その解決のためにIT技術を駆使して、解決を目指す力: Society 5.0では、IT技術を駆使して問題を解決する能力が非常に重要とされています。これには、新しい技術に対する理解と応用能力が必要です。
- ③ 今のやり方が最適と考え、新たに生まれてくる技術を否定する力: これはSociety 5.0の考え方に反するものです。新しい技術やアプローチに対して開かれた姿勢を持つことが、この新しい社会で成功するためには非常に重要です。過去のやり方に固執し、新しい技術を否定する態度は、革新的な変化に適応する上での障害となります。

👍 🗣️ 🔄 🗑️

注) 機密性の高い情報を扱う場合は情報漏洩につながらないよ
チャットGPTの記録をOFFにしておくこと

解説作成の補助

横100画素，縦50画素，1画素あたり24ビットフルカラーの画像のデータ量は
何バイトになるか，答えなさい。

ChatGPT

この問題では、横100ピクセル、縦50ピクセル、そして各ピクセルが24ビットのフルカラーで表現される画像のデータ量を計算します。以下の手順で計算を行います。

1. ピクセル数の計算:

まず、画像の全ピクセル数を計算します。これは横のピクセル数と縦のピクセル数を掛け合わせることで求められます。

$$\text{ピクセル数} = \text{横のピクセル数} \times \text{縦のピクセル数}$$

つまり、

$$\text{ピクセル数} = 100 \times 50 = 5000 \text{ピクセル}$$

2. ビット数の計算:

各ピクセルは24ビットのフルカラーで表現されています。したがって、画像全体で使われるビット数は、全ピクセル数にピクセルあたりのビット数を掛けたものになります。

$$\text{総ビット数} = \text{ピクセル数} \times \text{ピクセルあたりのビット数}$$

つまり、

$$\text{総ビット数} = 5000 \times 24 = 120000 \text{ビット}$$

3. バイト数への変換:

コンピュータでは通常、データ量をバイトで表します。1バイトは8ビットに相当するため、ビット数を8で割ることでバイト数に変換できます。

$$\text{データ量 (バイト)} = \frac{\text{総ビット数}}{8}$$

つまり、

$$\text{データ量} = \frac{120000}{8} = 15000 \text{バイト}$$

結果として、この画像のデータ量は15,000バイトとなります。

類似事例の作成

4-13

標本調査と仮説検定の考え方

動画 第4章 ▶ 4-13_標本調査と仮説検定の考え方

きょうの授業はこんな話

① 母集団と標本、推定

母集団と標本の違いについて学び、標本調査について理解しよう。

② 仮説検定の考え方

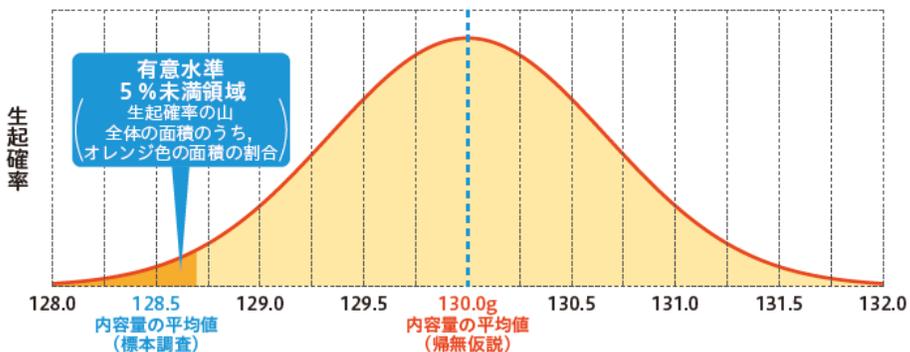
母集団がどのような値をもっているかということについて、標本データから得られた仮説を立証するときの考え方を学ぼう。



最近肌荒れがひどいです。30%増量キャンペーン中だからって、兄が大量に買って来たスナック菓子を、食べ過ぎてしまったからだと思います。



増量キャンペーンでオトナ買える気持ち、わかります。ところで、製品の内容量について、当然、メーカーはきちんと管理をしています。前回取り上げた統計学の知見は、そうした管理に役立ちます。今回は、具体例を通して、「標本調査」と「仮説検定」について学んでいきましょう。



② 仮説検定の考え方



標本データから母集団に関する仮説の真偽を判断する手法を、**仮説検定**といいます。たとえば、あるメーカーが、普段は内容量 100g で売っているスナック菓子を 30% 増量し、130g として 10,000 個売り出すキャンペーンを開催すると決めたとします。原材料を手配し、製造に関係する機械を調整し、試作品として 100 個作ってみてそれらについて全数調査をしたら、多少ばらつきはあったものの、内容量の平均値はちゃんと 130g になりました。

そして、念のため、128g 以下または 132g 以上の個体を検知したら、それを異常値と認めてアラート（警報）を鳴らすセンサーを、製造ラインの終点近くにとりつけました。そのうえで、いよいよ 10,000 個の製造開始です。

ところが、量産開始初日から、アラートが思ったよりもよく鳴ります。そこで、アラートが鳴った個体、鳴らなかった個体をよく混ぜてから 30 個無作為に抽出し、標本調査を行った結果、重量の標本平均は 128.5g でした。この結果からいえるのは、「平均の内容量は 130g より少ない」ということです。

この例では、128.5g という値は、1 つの個体の内容量としては正常です。とはいえ、標本調査の結果としては少々問題です。そのまま放置すると、数多くの廃棄すべき不良品を出してしまうことになりかねません。面倒ではありますが、機器が正しく動作しているかなど、製造工程の点検が必要な場面かもしれません。

しかし、ここで工場長が、「いや、10,000 個の全数調査をしたら、平均は 130g になるはず。だから、製造工程の点検は不必要。このまま、センサーがアラートを鳴らした個体を、その都度取り除くことを続けられればいいだけだ」と主張したとします。

このとき、「平均の内容量は 130g より少ない」という、標本調査をした人が立証したい仮説を対立仮説といい、対立仮説に反対する仮説（ここでは「平均の内容量は 130g」）を帰無仮説といいます。

仮説検定では、帰無仮説が正しいという前提において、対立仮説の根拠となっている標本調査の結果はどのくらいの確率で起こりうるのかを計算します。この確率を p 値（有意確率）といいます。 p 値が低く「普通は起きないこと」と判断されたら、それが標本調査で起きたのは問題である、という話になります。



ということは、 p 値には、「普通は起きないこと」と判断するかしらないかの分かれ目があるのですかね？



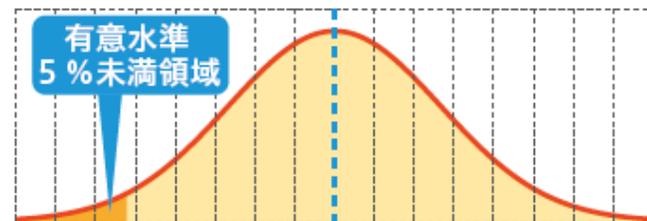
はい。統計学の世界では、一般的に、 p 値が 5% 未満の場合は、帰無仮説の提起時点で普通は起きないとしたことが、標本調査の結果として起きたという考えのもと、対立仮説が正しいと判断します。このことを、「帰無仮説を棄却する」といいます。

p 値が 5% 未満なら、対立仮説の根拠となっている調査結果について「有意性がある」ということができ、この 5% の基準を有意水準といいます。

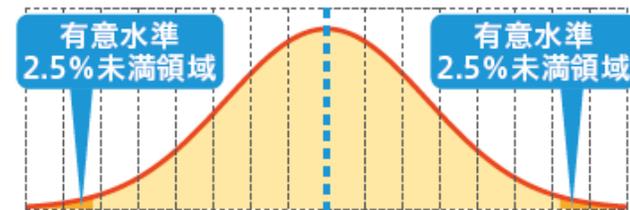
p 値が 5% 以上だと、対立仮説の根拠となっている調査結果は、帰無仮説から想定できる範囲の事象とされ、帰無仮説は棄却されません。

この例では、量産前に実施した母集団（試作品）の全数調査で、内容量の平均値はピッツ

片側検定



両側検定



出典：情報Ⅰ 大学入学共通テスト対策

会話型テキストと動画でよくわかる（インプレス）

類似事例の作成

② 仮説検定の考え方



標本データから母集団に関する仮説の真偽を判断する手法を、**仮説検定**といいます。
たとえば、あるメーカーが、普段は内容量 100g で売っているスナック菓子を 30% 増量し、130g として 10,000 個売り出すキャンペーンを開催すると決めたとします。原材料を手配し、製造に関係する機械を調整し、試作品として 100 個作ってみてそれらについて全数調査をしたら、多少ばらつきはあったものの、内容量の平均値はちゃんと 130g になりました。

そして、念のため、128g 以下または 132g 以上の個体を検知したら、それを異常値と認めてアラート（警報）を鳴らすセンサーを、製造ラインの終点近くにとりつけました。そのうえで、いよいよ 10,000 個の製造開始です。

ところが、量産開始初日から、アラートが思っていたよりもよく鳴ります。そこで、アラートが鳴った個体、鳴らなかった個体をよく混ぜてから 30 個無作為に抽出し、標本調査を行った結果、重量の標本平均は 128.5g でした。この結果からいえるのは、「平均の内容量は 130g より少ない」ということです。

この例では、128.5g という値は、1 つの個体の内容量としては正常です。とはいえ、標本調査の結果としては少々問題です。このまま放置すると、数多くの廃棄すべき不良品を出してしまうことになりかねません。面倒ではありますが、機器が正しく動作しているかなど、製造工程の点検が必要な場面かもしれません。

しかし、ここで工場長が、「いや、10,000 個の全数調査をしたら、平均は 130g になるはず。だから、製造工程の点検は不必要。このまま、センサーがアラートを鳴らした個体を、その都度取り除くことを続けられればいいだけだ」と主張したとします。

このとき、「平均の内容量は 130g より少ない」という、標本調査をした人が立証したい仮説を**対立仮説**といい、対立仮説に反対する仮説（ここでは「平均の内容量は 130g」）を**帰無仮説**といいます。

仮説検定では、**帰無仮説が正しいという前提**において、対立仮説の根拠となっている標本調査の結果はどのくらいの確率で起こりうるのかを計算します。この確率を **p 値**（有意確率）といいます。p 値が低く「普通は起きないこと」と判断されたら、それが標本調査で起きたのは問題である、という話になります。



ということは、p 値には、「普通は起きないこと」と判断するかしないかの分かれ目があるのですね？



はい。統計学の世界では、一般的に、p 値が 5% 未満の場合は、**帰無仮説の提起時点で普通は起きないとしたことが、標本調査の結果として起きたという考えのもと、対立仮説が正しいと判断**します。このことを、「**帰無仮説を棄却する**」といいます。

p 値が 5% 未満なら、対立仮説の根拠となっている調査結果について「**有意性がある**」ということができ、この 5% の基準を**有意水準**といいます。

p 値が 5% 以上だと、対立仮説の根拠となっている調査結果は、**帰無仮説から想定できる範囲の事象**とされ、**帰無仮説は棄却**されません。

この例では、量産前に実施した母集団（試作品）の全数調査で、内容量の平均値はピツタ

別の商品で 類似の事例を作成して



要点チェック問題

(3) 次の文章を読んで、帰無仮説と対立仮説と片側・両側検定の組み合わせとして適当なものを 2 つ選び、番号で答えなさい。

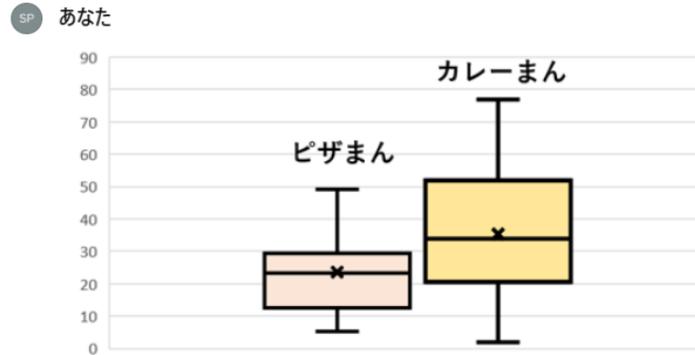
あるソフトドリンクメーカーが、普段は 500ml のボトルで販売している飲料を 20% 増量し、600ml として 50,000 本のキャンペーンを開催することにしました。
原材料を手配し、生産ラインを調整し、試作品として 200 本作って全数調査を行ったところ、多少のばらつきはありましたが、内容量の平均値はちゃんと 600ml でした。そして、598ml 以下または 602ml 以上のボトルを異常と判断するセンサーを設置しました。しかし、量産開始後、センサーのアラートが頻繁に鳴ります。そこで、50 本を無作為に抽出し、標本調査を行った結果、内容量の標本平均は 599ml でした。
ここで工場長が、「いや、50,000 個の全数調査をしたら、平均は 600ml になるはず。だから、製造工程の点検は不必要。このまま、センサーがアラートを鳴らした個体を、その都度取り除くことを続けられればいいだけだ」と主張しました。
あなたは工場長の考えは違うと思い、仮説検定を用いて考えの真偽を立証することにしました。

	帰無仮説	対立仮説	片側・両側
①	内容量の平均は 600ml である。	内容量の平均は 600ml ではない。	両側検定
②	内容量の標本平均は 600ml ではない。	内容量の平均は 600ml である。	片側検定
③	内容量の平均は 600ml である。	内容量の平均は 600ml より少ない。	片側検定
④	内容量の標本平均は 600ml ではない。	内容量の平均は 600ml である。	両側検定

出典：情報 I 大学入学共通テスト対策
会話型テキストと動画でよくわかる（インプレス）

将来的な期待（画像解析）

ChatGPT 4Vから画像解析が可能に！



箱ひげ図同士を比較して考察して

ChatGPT

この箱ひげ図では、2つの異なるデータセット（おそらく「ピンクさん」と「カレーさん」とラベル付けされている）の統計的分布を比較しています。箱ひげ図は、データの中央値、四分位数、外れ値などの要約統計を視覚的に表現するためによく使われます。

比較するときに注目すべき点は以下の通りです：

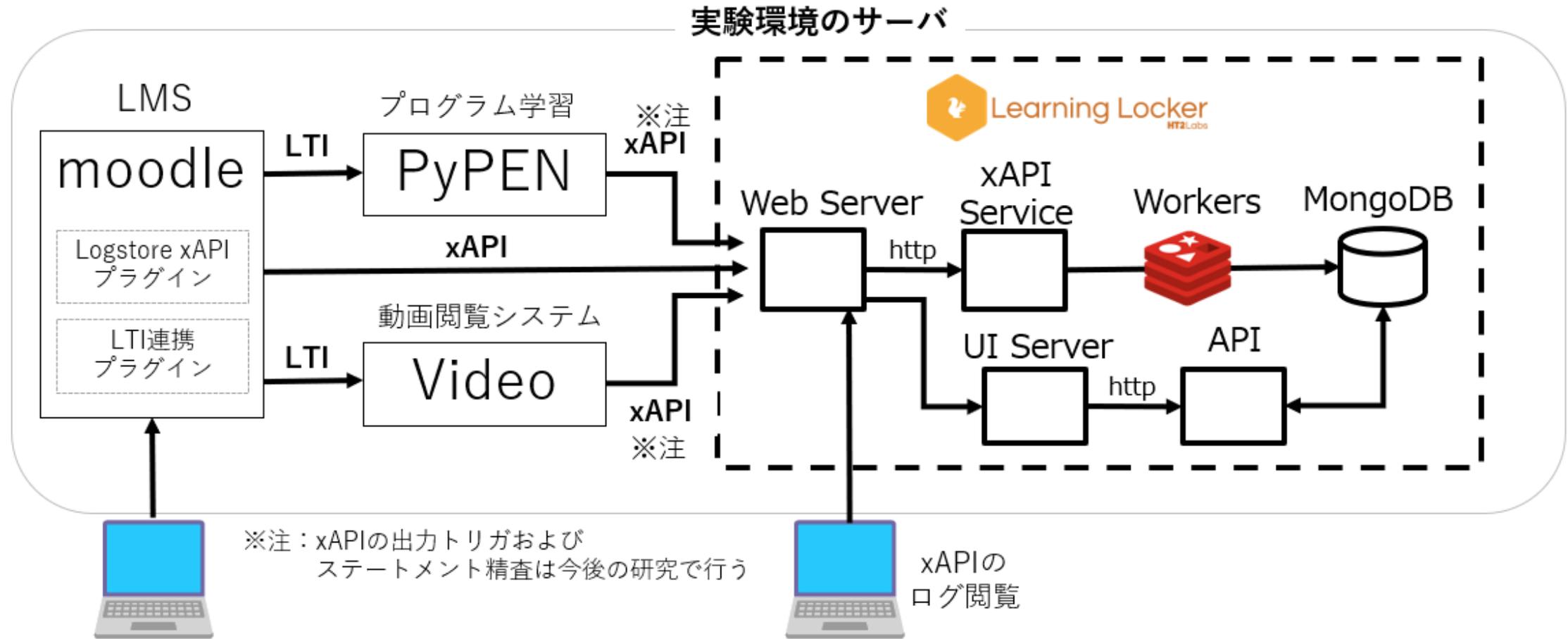
- **中央値**：箱の中の線（またはX印）は各データセットの中央値を示しており、カレーさんの中央値がピンクさんよりも高いことがわかります。
- **四分位範囲**：箱の上下の端はそれぞれ第3四分位数と第1四分位数を表しており、この範囲がデータセットの中央50%を占めています。カレーさんの四分位範囲がピンクさんよりも広いことが見て取れます。
- **外れ値**：図にある小さな記号は外れ値を表している可能性があります。これは通常、全体のデータセットから大幅に逸脱している値です。
- **範囲**：最高値と最低値の間の距離は、全体のバリエーションを示します。カレーさんのデータセットの方が範囲が広いことがわかります。

これらの観察から、カレーさんのデータセットはピンクさんのデータセットよりも中央値が高く、より大きなバリエーションがあるという結論が導かれます。ただし、具体的な文脈やデータが何を表しているのかにもよりますが、これらの情報をもとにデータセットについての更なる分析や解釈を行うことができます。

精度は未だ低い？

【番外編】大学院の研究実施中（教育工学）

高等学校「情報Ⅰ」ラーニングアナリティクス 実験環境構築



3月の情報処理学会全国大会（学生セッション）で発表予定

今回発表した内容

ChatGPTを活用した…

- シミュレーション問題作成
- 問題の検証
- 解説の作成の補助
- 類似事例の作成
- 画像解析（将来的な期待）

他にも沢山あると思います！

よい使い方ありましたら共有して頂けると嬉しいです。