

情報 1 2 学期末考査

福井県立勝山高等学校

2024 年 11 月 28 日 1 限目

注意事項

- 開始のチャイムが鳴るまで開かないこと.
- チャイムの前に問題用紙・解答用紙に記名して良い.
- 終了後, 問題冊子は持ち帰ること.
- 時間配分を考えて解くこと.

【マーク記入時の注意】

- 解答は, 解答用紙の問題番号に対応した解答欄にマークすること.

- (ウ) (エ) などの に数字を入れる場合, 以下の通りに記入せよ.

(例 1) (ウ) (エ) に 34 と解答する場合.

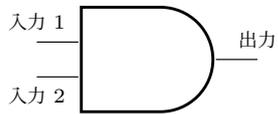
(ウ) に 3 を, (エ) に 4 を記入.

(例 2) (ウ) (エ) に 1 と解答する場合.

(ウ) に 0 を, (エ) に 1 を記入.

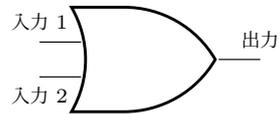
論理回路, フローチャート, python の例文については, 以下を参考にすること.

(A)AND 回路



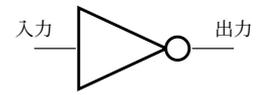
| 入力 1 | 入力 2 | 出力 |
|------|------|----|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

(B)OR 回路



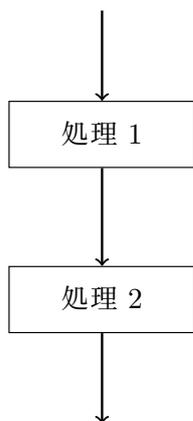
| 入力 1 | 入力 2 | 出力 |
|------|------|----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

(C)NOT 回路



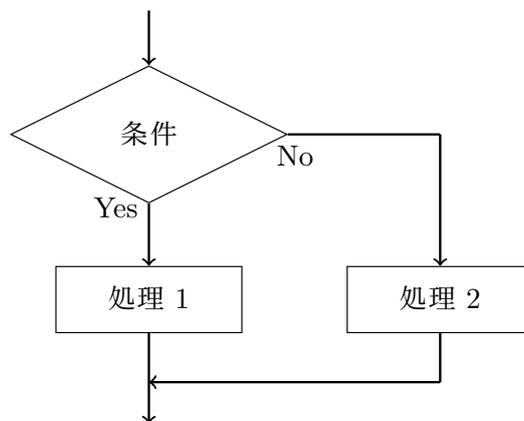
| 入力 | 出力 |
|----|----|
| 0 | 1 |
| 1 | 0 |

(A) 順次構造



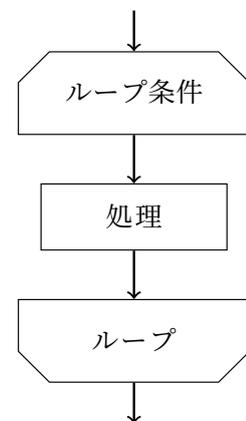
1つの処理の終了後に, 次の処理へと移る.

(B) 分岐構造



条件によって処理を選択し, 実行する.

(C) 反復構造



条件を満たす間, 同じ処理を繰り返し続ける.

python コード 基本文

1) 表示

```
1 print("Hello World")
2 print(321)
```

実行結果

```
1 Hello World
2 321
```

2) 繰り返し (for 文)

```
1 for i in range(3):
2     print(i, "Hello")
```

実行結果

```
1 0 Hello
2 1 Hello
3 2 Hello
```

3) 繰り返し (while 文)

```
1 i=0
2 while i<3:
3     print(i, "Hello")
4     i=i+1
```

実行結果

```
1 0 Hello
2 1 Hello
3 2 Hello
```

4) 条件分岐

```
1 a=int(input())
2 if a==1:
3     print("Hello")
4 elif a==2:
5     print("Bye")
6 else:
7     print("See You")
```

実行結果

実行後 2 を入力した場合

```
1 Bye
```

5) 演算

| | 演算子 | 記入例 | 結果 |
|-----|-----|----------|-----|
| 足し算 | + | $a + b$ | 7 |
| 引き算 | - | $a - b$ | 3 |
| 掛け算 | * | $a * b$ | 10 |
| 割り算 | / | a/b | 2.5 |
| 商 | // | $a//b$ | 2 |
| 余り | % | $a\%b$ | 1 |
| 累乗 | ** | $a ** b$ | 25 |

1 1996年1月から2024年8月までの各月毎のデータをまとめ、分析する。各問いに答えよ。【データ分析：25点】

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---------|-----------|-----------|-----------|-----------|----------|--------|----------|------------|------------|------------|
| 1 | date | start | high | low | end | UP | UP_per | doll-yen | trade | service | syuusi |
| 2 | 1996.01 | 20,618.00 | 20,812.74 | 20,080.92 | 20,812.74 | 194.74 | 1% | 105.81 | 2647.12591 | -5746.3597 | 341.517322 |
| 3 | 1996.02 | 20,935.12 | 21,118.30 | 19,919.97 | 20,125.37 | -809.75 | -4% | 105.7 | 8521.36439 | -4706.9767 | 8609.02696 |
| 4 | 1996.03 | 20,168.63 | 21,406.85 | 19,734.70 | 21,406.85 | 1,238.22 | 6% | 105.85 | 13175.1118 | -4741.9246 | 13154.0163 |
| 5 | 1996.04 | 21,560.39 | 22,282.05 | 21,424.24 | 22,041.30 | 480.91 | 2% | 107.4 | 5276.05495 | -5600.8535 | 6176.03206 |
| 6 | 1996.05 | 21,815.01 | 22,147.21 | 21,171.82 | 21,956.19 | 141.18 | 1% | 106.49 | 4060.60105 | -5102.0309 | 4079.92115 |
| 7 | 1996.06 | 21,588.52 | 22,666.80 | 21,588.52 | 22,630.75 | 942.23 | 4% | 108.82 | 9006.27397 | -5574.1668 | 6360.94251 |

図1 データセット

データセットの項目概要は、以下の通りである。

| 項目 | 説明 |
|----------|--------------------------------|
| date | 年.月 |
| start | 各月の日経平均株価の初期値 |
| high | 各月の日経平均株価の最高値 |
| low | 各月の日経平均株価の最低値 |
| end | 各月の日経平均株価の終値 |
| UP | 株価の上昇幅. (end-start) |
| UP_per | 初期値に対する上昇割合. (end-start)/start |
| doll-yen | 為替 (円/1ドル) |
| trade | 貿易国際収支. |
| service | サービス国際収支. |
| syuusi | 経常収支 |

| | date | start | high | low | end | UP | UP_per | doll.yen | trade | service | syuusi |
|----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|--------------|-------------|-------------|
| date | 1.00000000 | 0.57159980 | 0.57812577 | 0.57714399 | 0.58282488 | 0.13408225 | 0.11323904 | 0.069578616 | -0.651196190 | 0.45884860 | 0.20437364 |
| start | 0.57159980 | 1.00000000 | 0.99696472 | 0.99347559 | 0.99131750 | 0.01923680 | -0.01565819 | 0.609038408 | -0.325583945 | 0.06248739 | 0.28604117 |
| high | 0.57812577 | 0.99696472 | 1.00000000 | 0.99525819 | 0.99636341 | 0.08042399 | 0.04089248 | 0.610900157 | -0.333192394 | 0.06322927 | 0.28331792 |
| low | 0.57714399 | 0.99347559 | 0.99525819 | 1.00000000 | 0.99585104 | 0.10276389 | 0.06433961 | 0.603211685 | -0.326633057 | 0.06514214 | 0.28067374 |
| end | 0.58282488 | 0.99131750 | 0.99636341 | 0.99585104 | 1.00000000 | 0.15053540 | 0.10717282 | 0.605388158 | -0.333380428 | 0.07209787 | 0.28579907 |
| UP | 0.13408225 | 0.01923680 | 0.08042399 | 0.10276389 | 0.15053540 | 1.00000000 | 0.93263949 | 0.024155525 | -0.087030329 | 0.07842595 | 0.02256062 |
| UP_per | 0.11323904 | -0.01565819 | 0.04089248 | 0.06433961 | 0.10717282 | 0.93263949 | 1.00000000 | -0.015500048 | -0.069084083 | 0.08064828 | 0.02092346 |
| doll.yen | 0.06957862 | 0.60903841 | 0.61090016 | 0.60321169 | 0.60538816 | 0.02415553 | -0.01550005 | 1.000000000 | 0.008617121 | -0.15743515 | 0.25561120 |
| trade | -0.65119619 | -0.32558395 | -0.33319239 | -0.32663306 | -0.33338043 | -0.08703033 | -0.06908408 | 0.008617121 | 1.000000000 | -0.14357987 | 0.43658104 |
| service | 0.45884860 | 0.06248739 | 0.06322927 | 0.06514214 | 0.07209787 | 0.07842595 | 0.08064828 | -0.157435155 | -0.143579872 | 1.000000000 | 0.41222294 |
| syuusi | 0.20437364 | 0.28604117 | 0.28331792 | 0.28067374 | 0.28579907 | 0.02256062 | 0.02092346 | 0.255611200 | 0.436581043 | 0.41222294 | 1.000000000 |

図2 相関行列

相関行列とは、行と列の交差する点の値が、その2項目の相関係数となる行列のことである。

株価の上昇率が為替や日本の経常収支に与える影響を分析するため、3つの行を抽出して分析を行った。

まず, 各項目の外れ値なしの箱ヒゲ図を描いた. (単位はすべて円)

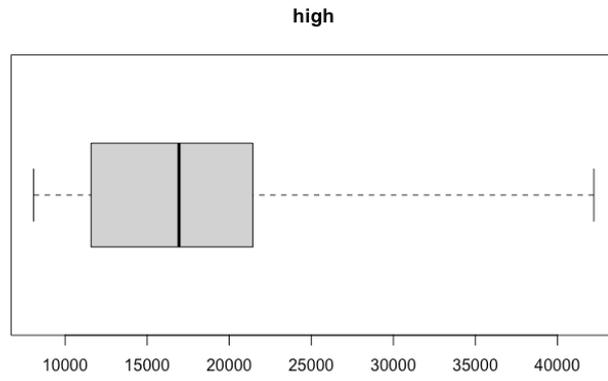


図 3 散布図 1

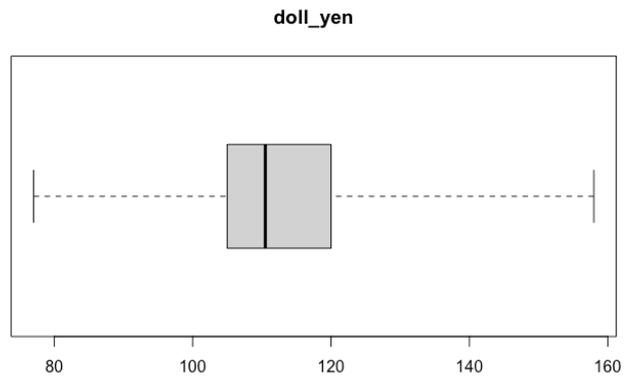


図 4 散布図 2

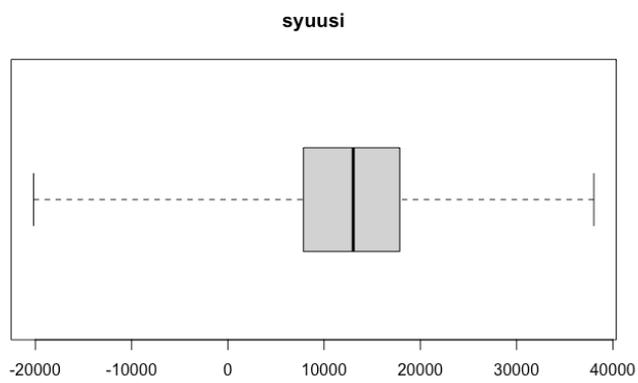


図 5 散布図 3

日経平均株価の月最高値, 為替, 経常収支の3項目に注目して, 分析を行うことにした. (単位はすべて円)

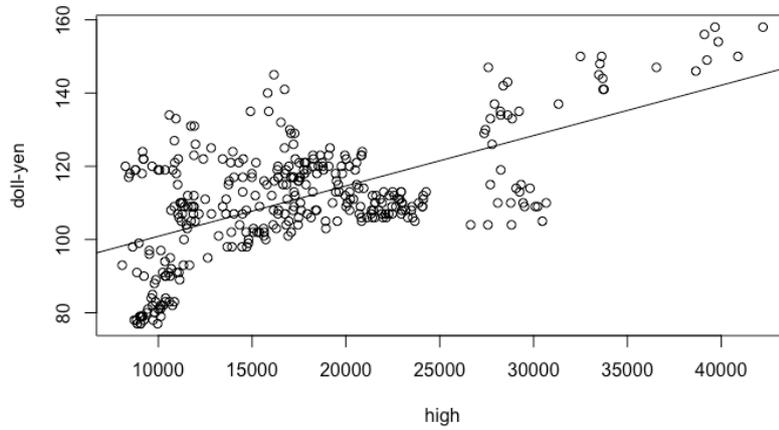


図6 散布図1

| | |
|--------|-------------------------|
| 相関係数 | 0.6109 |
| 回帰直線の式 | $87.126421 + 0.001375x$ |

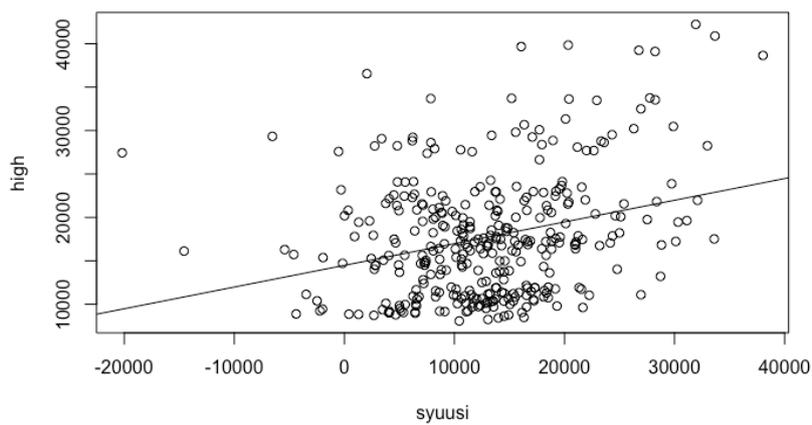


図7 散布図2

| | |
|--------|-------------------|
| 相関係数 | 0.2833 |
| 回帰直線の式 | $14480 + 0.2498x$ |

(ア) 図1のデータセットから、分析の仮説を立てる。仮説として適切でないものを選び。

- ① 日経平均株価が高いほど、為替レートは高い傾向にあるのではないか。
- ② 日経平均株価が低いほど、日本人の貯金額が増える傾向があるのではないか。
- ③ 貿易収支は、為替には左右されないだろう。
- ④ 貿易国際収支とサービス国際収支は、連動して動いているだろう。

(イ) 相関の有無を調べたい。その際の相関行列の見方について、正しいものを選び。

- ① 値が1に近いものに注目すればよい。
- ② 値が0に近いものに注目すればよい。
- ③ すべての関係性に注目するべきである。
- ④ 絶対値が大きいものに注目すればよい。
- ⑤ 絶対値が小さいものに注目すればよい。

(ウ) 箱ひげ図(図3, 4, 5)から読み取れるものとして正しいものを選び。

- ① 中央値が最も大きいのは syuusi である。
- ② 四分位範囲が最も小さいのは doll-yen である。
- ③ high の平均値は 20000 以上である。
- ④ すべてのデータは正の値をとる。

(エ) 回帰直線の係数の決定法はいくつかある。そのうち「最小二乗法」は、次のような手順である。

a) $x = [x_1, x_2, \dots, x_n]$ と $y = [y_1, y_2, \dots, y_n]$ という n 個のデータセットに対して...

b) 回帰直線 $y = ax + b$ の係数 a, b を,

$$S = (y_1 - (ax_1 + b))^2 + (y_2 - (ax_2 + b))^2 + (y_3 - (ax_3 + b))^2 + \dots + (y_n - (ax_n + b))^2$$

の値が最小となるように定める。

正しいものを選べ。

- ① 相関係数 r が、 $-1 < r < 1$ のとき、 a, b が 0 になることはない。
- ② 相関係数の値に関わらず、 b の値が 0 になる可能性がある。
- ③ a の値が大きいほど、関係性が強いということである。
- ④ b の値が大きいほど、関係性が強いということである。

(オ) 図 6 と図 7, それらに付随する情報から読み取ったこととして正しいものを選べ。

- ① 回帰直線の傾きが大きい「syuusi と high」の関係性の方が、「high と doll-yen」よりも関係性が強い。
- ② 回帰直線の傾きが小さい「syuusi と high」の関係性の方が、「high と doll-yen」よりも関係性が強い。
- ③ 相関係数の値が大きい「syuusi と high」の関係性の方が、「high と doll-yen」よりも関係性が強い。
- ④ 相関係数の値が大きい「syuusi と high」の関係性の方が、「high と doll-yen」よりも関係性が弱い。

(カ) high= 50000 のときの為替 (円/1 ドル) を、回帰直線の式をもとに求めよ。(選択肢は、小数第 3 位を四捨五入)

- ① 155.88 円
- ② 150.88 円
- ③ 259.70 円
- ④ 269.70 円

問題は続きます.

- 2 ある学校の2年生対象に、アンケートを行った。その結果をテキストマイニングした結果を、以下に示す。【テキストマイニング：25点】

| 質問番号 | 内容 |
|------|--|
| (1) | プログラミングの授業を受ける前のプログラミングに対する印象を教えてください。 |
| (2) | 現在のプログラミングに対する印象を教えてください。 |
| (3) | プログラミングのスキルは、将来役に立つと思いますか。その理由を... |

また、テキストマイニングについて、以下の通り。

| 単語 | 概要 |
|----------|--|
| ワードクラウド | スコアが高い単語を複数選び出し、その値に応じた大きさに図示している。単語の色は品詞の種類で異なり、青色が名詞、赤色が動詞、緑色が形容詞、灰色が感動詞を表す。ここでのスコアは、頻出度としている。 |
| 共起ネットワーク | 文章中に出現する単語の出現パターンが似たものを線で結んだ図。出現数が多い語ほど大きく、また共起の程度は強い方から順に太い実線 > 細い実線 > 破線 で描画される。 |

(3) プログラミングのスキルは, 将来役に立つと思いますか. その理由を...



図 12 ワードクラウド

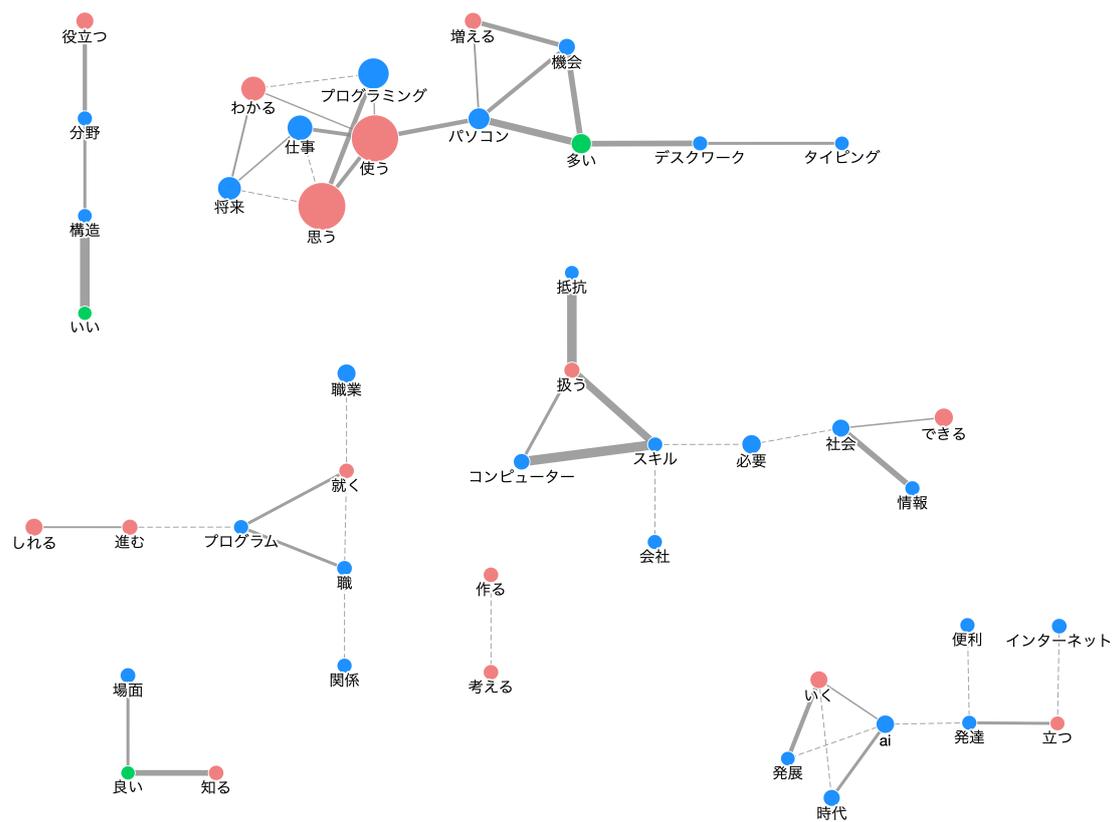


図 13 共起ネットワーク

(ア) 質問番号 (1) について、分析結果として適切なものを選べ。

- ① 図 8 を参照すると、プログラミングを使いこなせる人が多いことがわかる。
- ② 図 8 を参照すると、プログラミングに対してネガティブな印象を持っている人が多いことがわかる。
- ③ 図 8 を参照すると、プログラミングは単純なものであるという印象の人と、複雑であるという印象の人の割合が同程度である事がわかる。
- ④ 図 8 を参照すると、プログラミングは難しいから触れづらいつ感じている人が多い。

(イ) 質問番号 (1) について、分析結果として適切でないものを選べ。

- ① 図 9 を参照すると、プログラミングは頭がいい人が使うものであると感じている人が多い事がわかる。
- ② 図 9 を参照すると、プログラミングは小難しく触れづらいつ感じている人がいる事がわかる。
- ③ 図 9 を参照すると、プログラミングを行うにあたって数式は必要ないと感じている人がいる事がわかる。
- ④ 図 9 を参照すると、めんどくさいが理解できると面白いということを大半の人が感じている事がわかる。

(ウ) 質問番号 (2) について、分析結果として適切なものを選べ。

- ① 図 10 を参照すると、プログラミングは AI がやる事であると感じている人が多い事がわかる。
- ② 図 10 を参照すると、エラーを見つけ出すことに難しさを感じている人が多い事がわかる。
- ③ 図 10 を参照すると、将来ゲームクリエイターとして使用すると感じている人がいる事がわかる。
- ④ 図 10 を参照すると、達成感を感じる人もいる事がわかる。

(エ) 質問番号 (2) について、分析結果として適切なものを選べ。

- ① 図 11 を参照すると、プログラミングが得意な人が多くいる事がわかる。
- ② 図 11 を参照すると、成功時に達成感を感じる人が大半である事がわかる。
- ③ 図 11 を参照すると、成功が達成感とつながっている人がいる事がわかる。
- ④ 図 11 を参照すると、エラーを見つけ出す事が理解につながると感じている人がいる事がわかる。

(オ) 質問番号 (3) について、分析結果として適切なものを選べ。

- ① 図 12 を参照すると、プログラミングが役に立たないと感じている人が多い事がわかる。
- ② 図 12 を参照すると、プログラミングのスキルが様々な分野で必要になると感じている人が多い。
- ③ 図 12 を参照すると、プログラミングのスキルが役に立つと感じている人が多数派である事がわかる。
- ④ 図 12 を参照すると、プログラミングのスキルが特定の職業で必要になると感じている人が多数いる事がわかる。

(カ) 質問番号 (3) について、分析結果として適切なものを選べ。

- ① 図 13 を参照すると、プログラムの構造を学ぶことが職につく上でいいと感じている人がいる。
- ② 図 13 を参照すると、プログラミングのスキルは将来使うと思っている人が多い事がわかる。
- ③ 図 13 を参照すると、プログラミングのスキルよりもタイピングの方に重要性を感じている人がいる事がわかる。
- ④ 図 13 を参照すると、インターネットは便利だと感じている人が多くいる事がわかる。

(キ) テキストマイニングについて、適切な説明を選べ。

- ① 回答を数値で行うと、テキストマイニングが行いやすい。
- ② 共起ネットワークの線で結ばれていない線に注目する必要はない。
- ③ 回答者が全員、一単語での回答をした場合、有効な共起ネットワークが生成できない。
- ④ ワードクラウドの結果で出てきた最も大きい文字に注目して分析していく必要がある。

3 様々なシミュレーションについて、2人が話している。会話文を読んで各問いに答えよ。【シミュレーション：25点】

Aさん. 授業で、シミュレーションについて学んだね。

Bさん. そうだっけ。

Aさん. 寝てたんだ。Bさんはテストでは点数取れないね。あの単元すごく難しいから。

Bさん. 赤点は取りたくないな。助けてよ。

Aさん. 仕方ない。確率モデルを使ったシミュレーション手法の一つである「モンテカルロ法」について教えてあげる。テストに出るって先生言ってたし。

Bさん. 教えることも理解を深めるのに役に立つし、いい機会だね。

Aさん. 「モンテカルロ法を用いて円周率の近似解を得る」ことを目指そう。

Bさん. おおよその値を求めるとのことだね。

Aさん. 正解。以下のような手順で近似解を求めるんだ。

モンテカルロ法による円周率の計算 (例)

(1) 半径1の円を描き、その円に外接する正方形を描く。

(2) その正方形の中に、ランダムに点を打っていく。

(3) 多くの点を打った後、以下の計算を行う。

$$\frac{\text{円の中に入った点の数}}{\text{打った点の数}}$$

(4) この値を4倍したものが、円周率の近似解になる。

Bさん. 疑問点がいくつかあるんだけど..

Aさん. なんだい。なんでも答えてあげるよ。

Bさん. 点を打つ個数はどうしたらいいの。

Aさん.

Bさん. じゃあ、シミュレーションは誰がやっても結果は変わらないの。

Aさん.

Bさん. そうなんだ。ちなみに最初に描く半径を2にしてもいいの。

Aさん. いいよ。

Bさん. そうなんだね。すごく勉強になったよ。ありがとう。

Aさん. ちなみに、円周率の近似解を得るシミュレーション手法として、 というものもあるんだ。

Bさん. それはどんな手法なの。

Aさん. Bさんにわかるかな。以下のような手順で近似解を求めるんだ。

(エ) による円周率の計算 (例)

(1) 平面上に間隔 l で平行線を引く.

(2) ながさ d ($d \leq l$) の針を無作為に投げた後, 以下の計算を行う.

$$\frac{\text{投げた針の本数}}{\text{線と交わった針の本数}} \times \frac{2l}{d}$$

(3) この値が, 円周率の近似解になる.

B さん. わかったよ. この例だと, 理論上針が線と交わる確率は (オ) なんだね. 証明してみたいな.

A さん. そうだね. やってみたいけど, 数学の知識がいるから情報とは関係ないかもね. ちなみに, 大学で学ぶ確率論の知識も必要だけど, 私は分かるけど, B さんには分からないだろうから気にしないでいいよ. また興味があったら聞いてきてね.

B さん.

(ア) 当てはまるものとして最も適切なものを選べ.

- ① 点を打つ回数を何回にしても, シミュレーションの精度は変わらないよ.
- ② 点を打つ回数を増やせば増やすほど, シミュレーションの精度は上がるよ.
- ③ 点を打つ回数を減らせば減らすほど, シミュレーションの精度は上がるよ.
- ④ 点を打つ回数よりも, コンピュータの性能を上げることが, 精度を上げることにつながるよ.

(イ) 当てはまるものとして最も適切なものを選べ.

- ① 誰がやっても, 同じ結果になるよ.
- ② 使うパソコンが同じであれば, 同じ結果になるよ.
- ③ 違う人がやれば, 必ず違う結果になるよ.
- ④ 同じ結果になることもあるし, ならないこともあるよ.

(ウ) 当てはまるものとして最も適切なものを選べ.

- ① 他の数値を変えなくても, 円周率の近似値が出てくるよ.
- ② (4)での「4倍」を「2倍」に変えればいいよ.
- ③ (4)での「4倍」を「16倍」に変えればいいよ.
- ④ (4)での「4倍」を「 $\frac{1}{4}$ 倍」に変えればいいよ.

(エ) 当てはまるものとして最も適切なものを選べ.

- ① スーパーコンピュータ
- ② ビュフォンの針
- ③ クイックソート
- ④ 静的モデル

(オ) 当てはまるものとして最も適切なものを選べ. (ただし, π は円周率の値とする.)

- ① π
- ② $\frac{\pi}{4}$
- ③ $\frac{2l\pi}{d}$
- ④ $\frac{2l}{\pi d}$

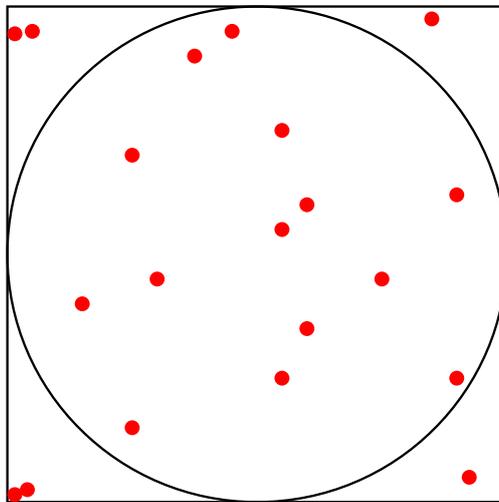
(カ) モンテカルロ法について述べた文として正しいものを選び.

- ① 点を 1000 回打つ場合, 3.1415 のように小数第 4 位までの近似解を得る事ができる.
- ② 点を 1000 回打つ場合, 3.141 のように小数第 3 位までの近似解を得る事ができる.
- ③ 点を 1000 回打つ場合, 3.14 のように小数第 2 位までの近似解を得る事ができる.

(問) モンテカルロ法でのシミュレーションの結果, 以下のようになった. この際の円周率の近似値を求めよ.

ただし, 点を打つ回数 は 20 回であるとし, その点は下図中に赤で示している. また, この円の半径は 3 としている.

回答は, . に回答すること. (3.1 の場合は, キに 3 を, クに 1 を回答)



4 【プログラミング：25点】

どの目の出方も同様に確からしいの六面サイコロを振るシミュレーションを行う。

手順書 (簡易)

- (1) 各目が何回ずつ出たかのカウントを行う配列を作る。
- (2) サイコロを振る回数を、ユーザに入力させて読み取る。
- (3) サイコロを振る回数分、以下の操作を繰り返す。
 - i. 1以上6以下の乱数を生成する。
 - ii. 生成された乱数の値から1引いた値の配列番号の数値を1増やす。
- (4) 結果を表示する。

この手順書を、python コードで示したものが以下である。

```
1 import random
2
3 ME=[0,0,0,0,0,0]
4 N=int(input("times of dice = "))
5 for i in range(N):
6     me=random.randint(1,6)
7     ME[me-1]=ME[me-1]+1
8 print(ME)
```

[問題]

(ア) このコードについて説明した文章として適切なものを選べ.

- ① 5行目の乱数関数の2つの値を変えるだけで, 20面サイコロや8面サイコロをシミュレーションする事ができる.
- ② 4面サイコロをシミュレーションしたい場合, 6行目の乱数関数の2つの値を変更するのみで可能である.
- ③ ユーザーの入力するNの値をどれだけ増やしても, 結果が出るまでの時間は変わらない.
- ④ 6行目の「me」の取りうる値の範囲は, 1以上6未満の整数である.

(イ) ユーザーが100と入力した際の出力結果として, 起こり得るものを選べ.

- ① [10, 10, 20, 20, 20, 10]
- ② [0, 0, 0, 100, 0]
- ③ [15, 15, 20, 15, 25, 10]
- ④ [1, 1, 1, 1, 1, 94, 1]

(ウ) プログラムの構造上, 実現したいシミュレーションに影響を与えない入れ替えを選べ.

- ① 3行目と4行目
- ② 6行目と7行目
- ③ 5行目と6行目
- ④ 4行目と5行目

(エ) 出た目の値の総和と、平均を求めるプログラムを組みたい。9行目以降に追加するプログラムとして正しいものを選べ。

①

```
9 SUM=0
10 for i in range(len(ME)):
11     SUM=SUM+(i+1)*ME[i]
12 AVE=SUM/N
13 print("SUMMARY = ",SUM)
14 print("AVERAGE = ",AVE)
```

②

```
9 SUM=0
10 for i in range(len(ME)):
11     SUM=SUM+i*ME[i-1]
12 AVE=SUM/N
13 print("SUMMARY = ",SUM)
14 print("AVERAGE = ",AVE)
```

③

```
9 SUM=0
10 for i in range(len(ME)):
11     SUM=SUM+(i+1)*ME[i+1]
12 AVE=SUM/N
13 print("SUMMARY = ",SUM)
14 print("AVERAGE = ",AVE)
```

④

```
9 SUM=0
10 for i in range(len(ME)):
11     SUM=SUM+i*ME[i]
12 AVE=SUM/N
13 print("SUMMARY = ",SUM)
14 print("AVERAGE = ",AVE)
```

サイコロに歪みが生じている場合のシミュレーションを行いたいと考え、以下のようなコードを組んだ。

```
1 import random
2
3 ME=[0,0,0,0,0,0]
4 N=int(input("times of dice = "))
5 for i in range(N):
6     me=random.randint(1,8)
7     if me==7:
8         ME[1]=ME[1]+1
9     elif me==8:
10        ME[3]=ME[3]+1
11    else:
12        ME[me-1]=ME[me-1]+1
13 print(ME)
```

(問) どのような歪みを再現しているか。適切なものを2個選び、, に回答せよ。ただし、順序不同である。

- ① 1の目が出やすい。
- ② 2の目が出やすい。
- ③ 3の目が出やすい。
- ④ 4の目が出やすい。
- ⑤ 5の目が出やすい。
- ⑥ 6の目が出やすい。
- ⑦ 他にはない。

