スポーツデータ分析（ボウリング）

3番　荒木　渉志
10番　尾﨑　陽
28番　西山　香与

目次

テーマ

ボウリングでどのような軌道を描けば
必ずストライクが取れるのかを知る

テーマの理由

- 「ここで決めたら僕と結婚してください。」
  というシーンで必ず決めるため

- フラグを回収しないため

分析方法

プロボウリング選手の動画からデータを収集する

- <データの種類>
  - 投げるときの角度
  - ボールの速さ
  - ピンに当たるときの角度
  - 倒れたピンの本数
  - 選手の立ち位置

ヒストグラムから視覚的に分析

プログラムを用いて平均を知る

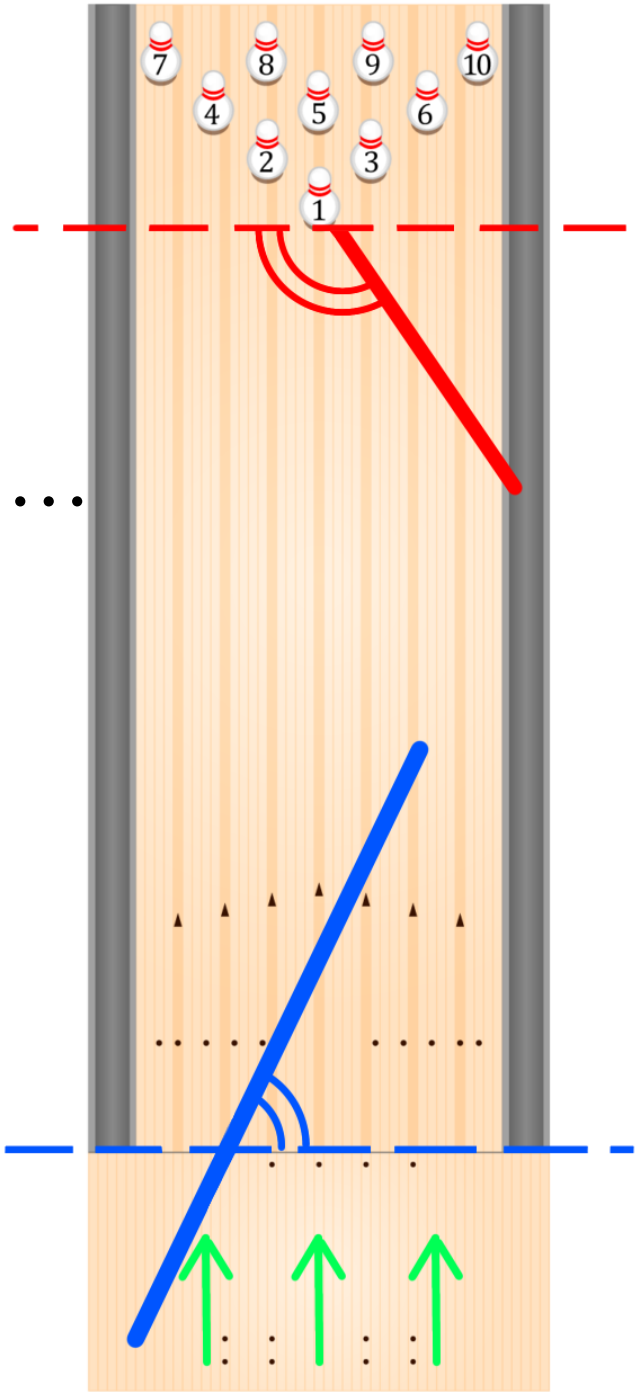
分析方法

必ずストライクが取れる軌道を知るためには…

①ピンの狙いを知る

②ピンに当たるときの角度を知る

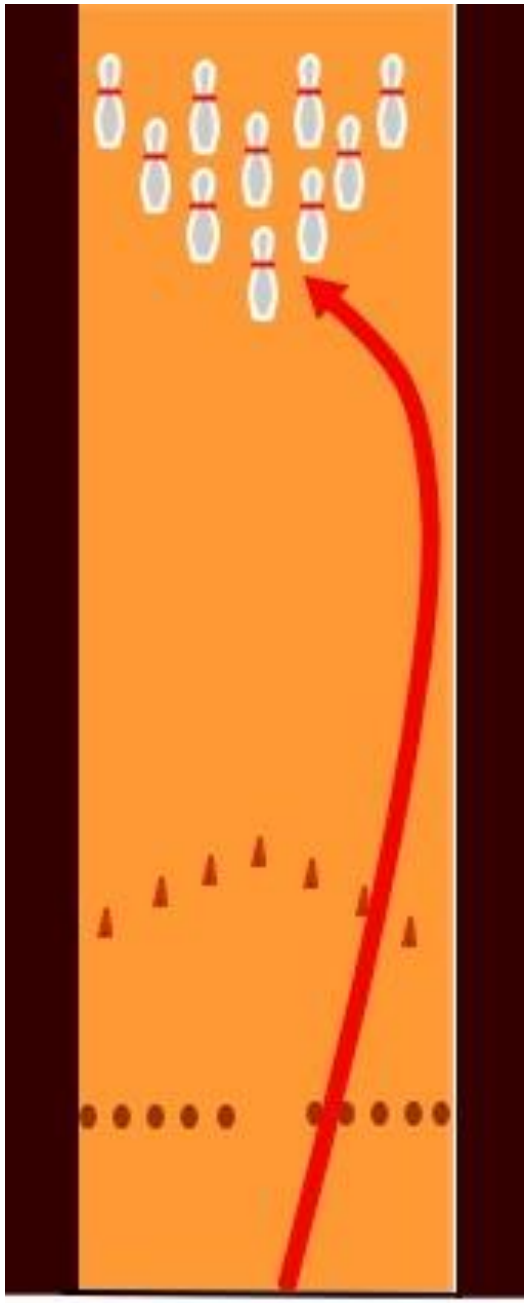
③立ち位置と投げる時の角度を知る



分析結果

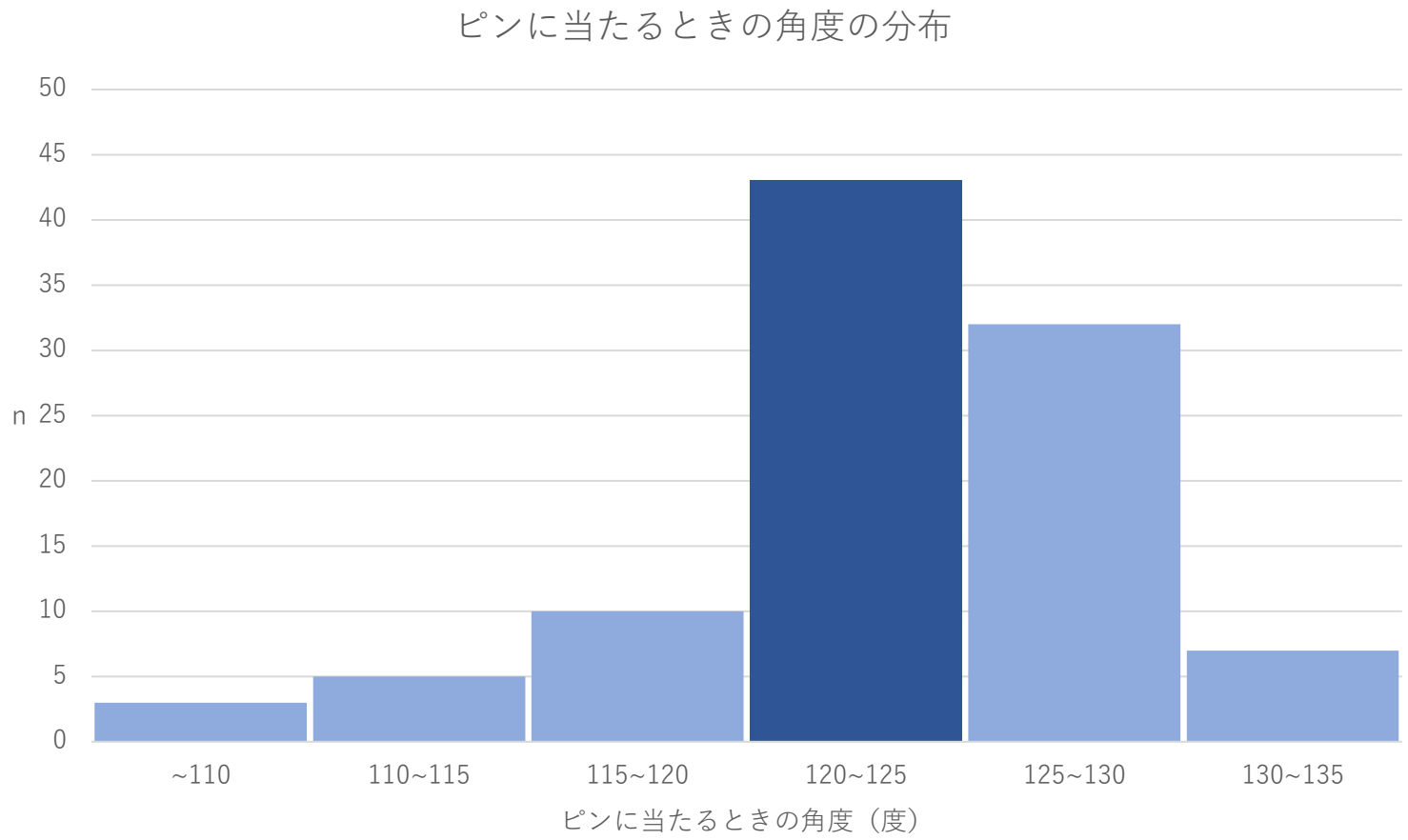
①ピンの狙い

1番ピンと３番ピンの間



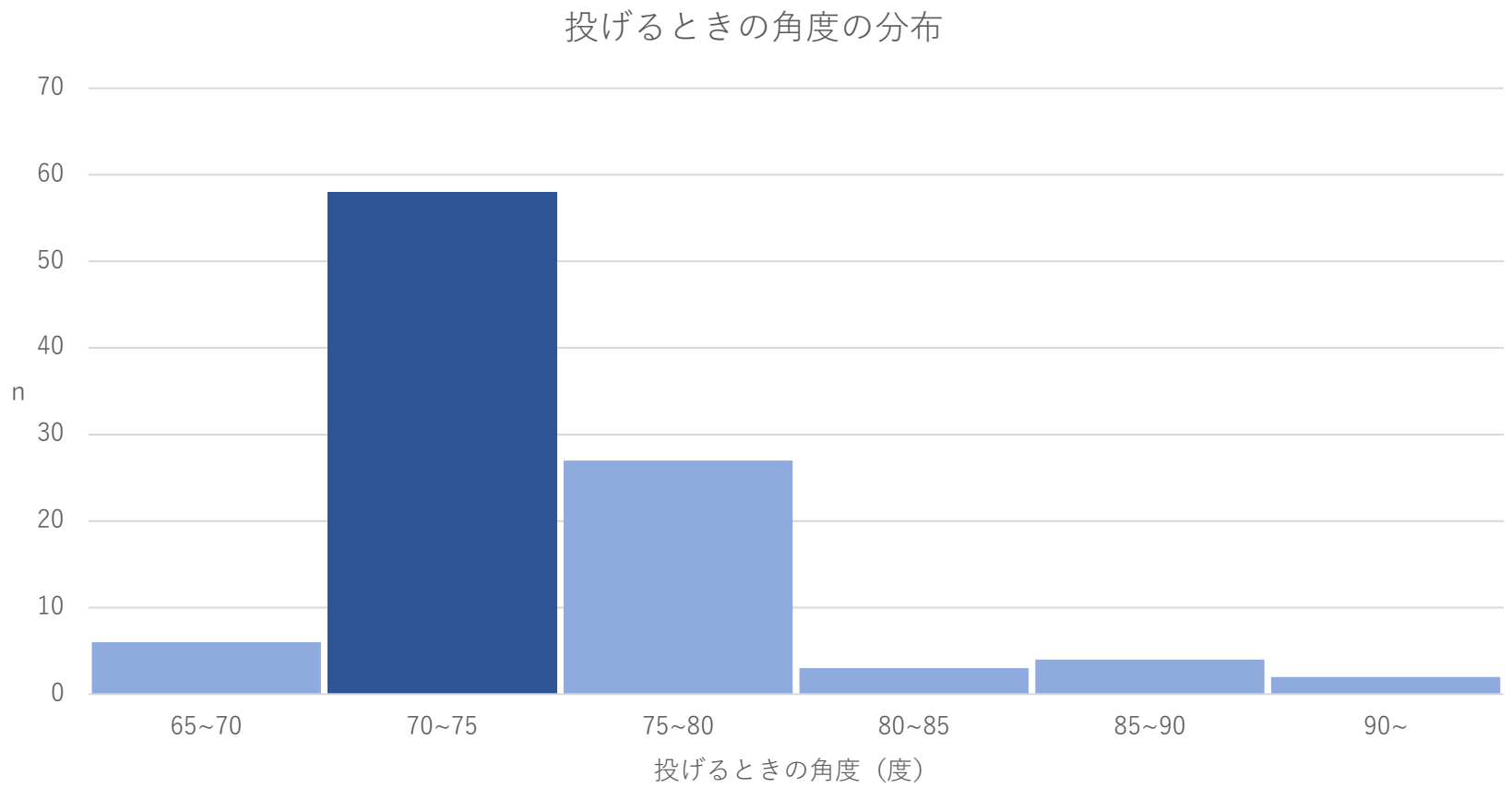
分析結果

②ピンに当たるときの角度



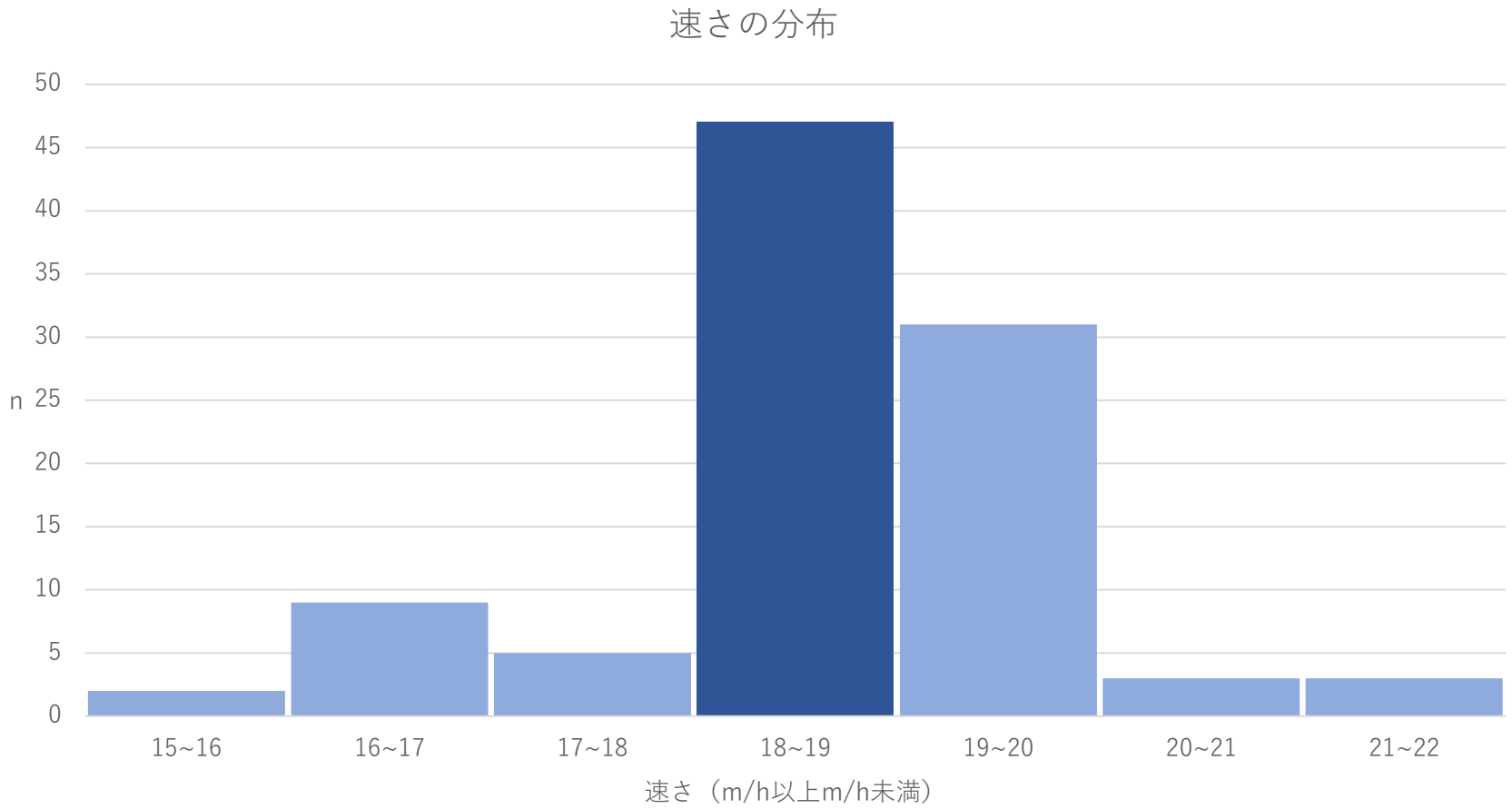
分析結果

③投げる時の角度



分析結果

- 投げる時の速さ



分析結果

```
import java.io.IOException;
public class boring_main {
    public static void main(String[] args) throws IOException {

        String filename1 = "boring_data.csv"; // UTF-8
        String[] lines1 = Library_I3_Pro2_2023.InputFromFile( filename1 );//ファイルの読み込み
        //Library_I3_Pro2_2023.printArray( lines1 );

        Member[] members = new Member[ lines1.length ];//各データを入れる配列
        for (int i = 1; i < lines1.length; i++){//データの数分繰り返す
            String[] data = lines1[i].split(",");//各行のデータをカンマごとに要素として入れる
            members[i] = new Member( data );//MemberクラスのMember( String[] data )を呼び出す
        }

        //出力
        System.out.println("データ数："+Member.n_of_members);
        System.out.println("投げるときの角度の平均："+Member.through_radian_sum/Member.n_of_members);
        System.out.println("ボールのスピードの平均："+Member.speed_sum/Member.n_of_members);
        System.out.println("ピンに当たるときの角度の平均："+Member.pin_radian_sum/Member.n_of_members);

        if(Member.position[1]>Member.position[2] && Member.position[1]>Member.position[3])
            System.out.println("左:"+Member.position[1]);
        else if(Member.position[2]>Member.position[1] && Member.position[2]>Member.position[3])
            System.out.println("真ん中:"+Member.position[2]);
        else if(Member.position[3]>Member.position[2] && Member.position[3]>Member.position[1])
            System.out.println("右："+Member.position[3]);
    }
}
```

分析結果

```
class Member {↓
    double through_radian; // 投げたときの角度↓
    double speed; // 速さ↓
    double pin_radian; // ピンに当たるときの角度↓
    double pin_count; // 倒れたピンの本数↓
    int stand_point; // 立ち位置↓
     ↓
    static int n_of_members = 0; // メンバーの人数    ↓
    static double through_radian_sum=0;// 投げたときの角度の合計↓
    static double speed_sum=0;// 速さの平均↓
    static double pin_radian_sum=0; // ピンに当たるときの角度の合計↓
    static int[] position=new int[4];//立ち位置（インデックス　左：１，真ん中２，右：３）
    /**↓
    * コンストラクタ↓
    */↓
    ↓
    Member(){//初期化↓
          this.through_radian = -1;↓
          this.speed = -1;↓
          this.pin_radian = -1;↓
          this.pin_count = -1;↓
          this.stand_point = -1;↓
    }↓
```

分析結果

```
    Member( String[] data ){
     this.through_radian = Double.parseDouble(data[1]);//指定した行の投げるときの角度
     this.speed = Double.parseDouble(data[2]);
     this.pin_radian = Double.parseDouble(data[3]);
     this.pin_count = Double.parseDouble(data[4]);
     this.stand_point = Integer.parseInt(data[5]);
     
     if(this.pin_count==10){//ストライクの時の各データごとの合計
       n_of_members++;
       through_radian_sum+=this.through_radian;
       speed_sum+=this.speed;
       pin_radian_sum+=this.pin_radian;
       position[this.stand_point]++;//立ち位置に対応するインデックスの要素をインクリメントする
     }
    }
}
```

分析結果まとめ

- ヒストグラムの結果

投げる時の角度	70~75
ボールのスピード	18~19
ピンに当たる時の角度	120~125

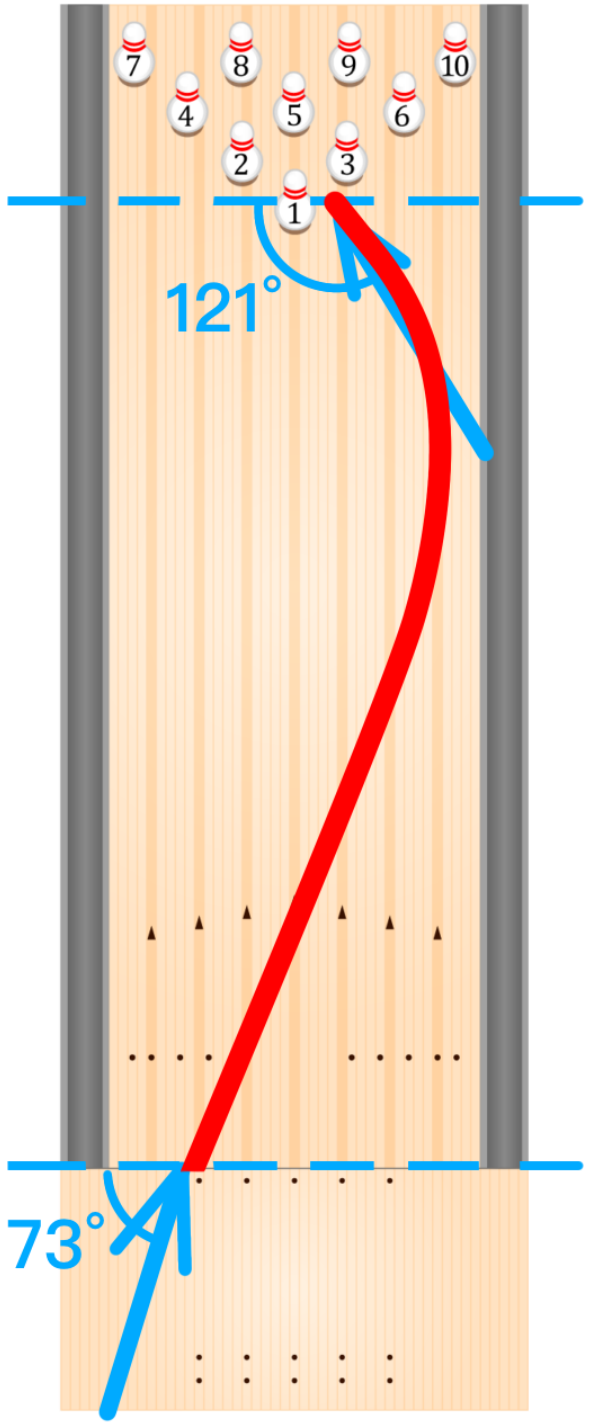
- プログラムの結果

```
データ数：69
投げるときの角度の平均：73.18115942028986
ボールのスピードの平均：18.720289855072465
ピンに当たるときの角度の平均：121.57971014492753
左:52
```

分析結果まとめ

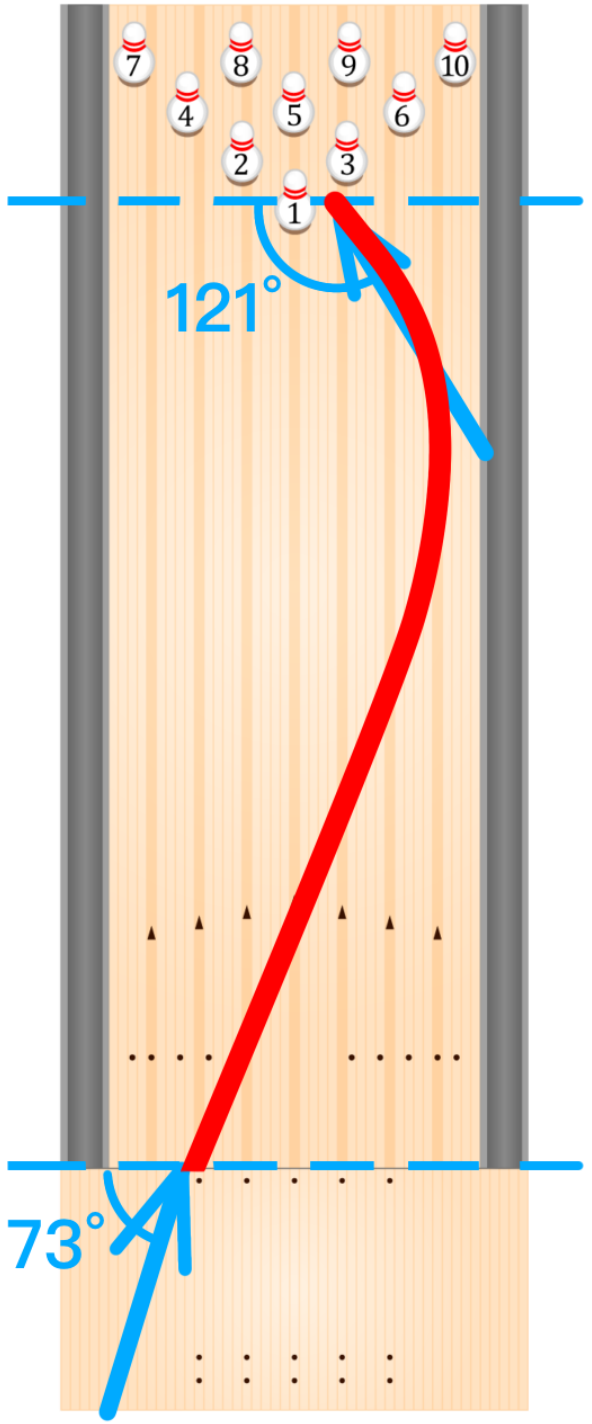
～確実にストライクが取れる軌道～

- 1番ピンと3番ピンの間を狙う
- ピンに当たる角度は121°
- 立ち位置は左より
- 投げる時の角度は73°

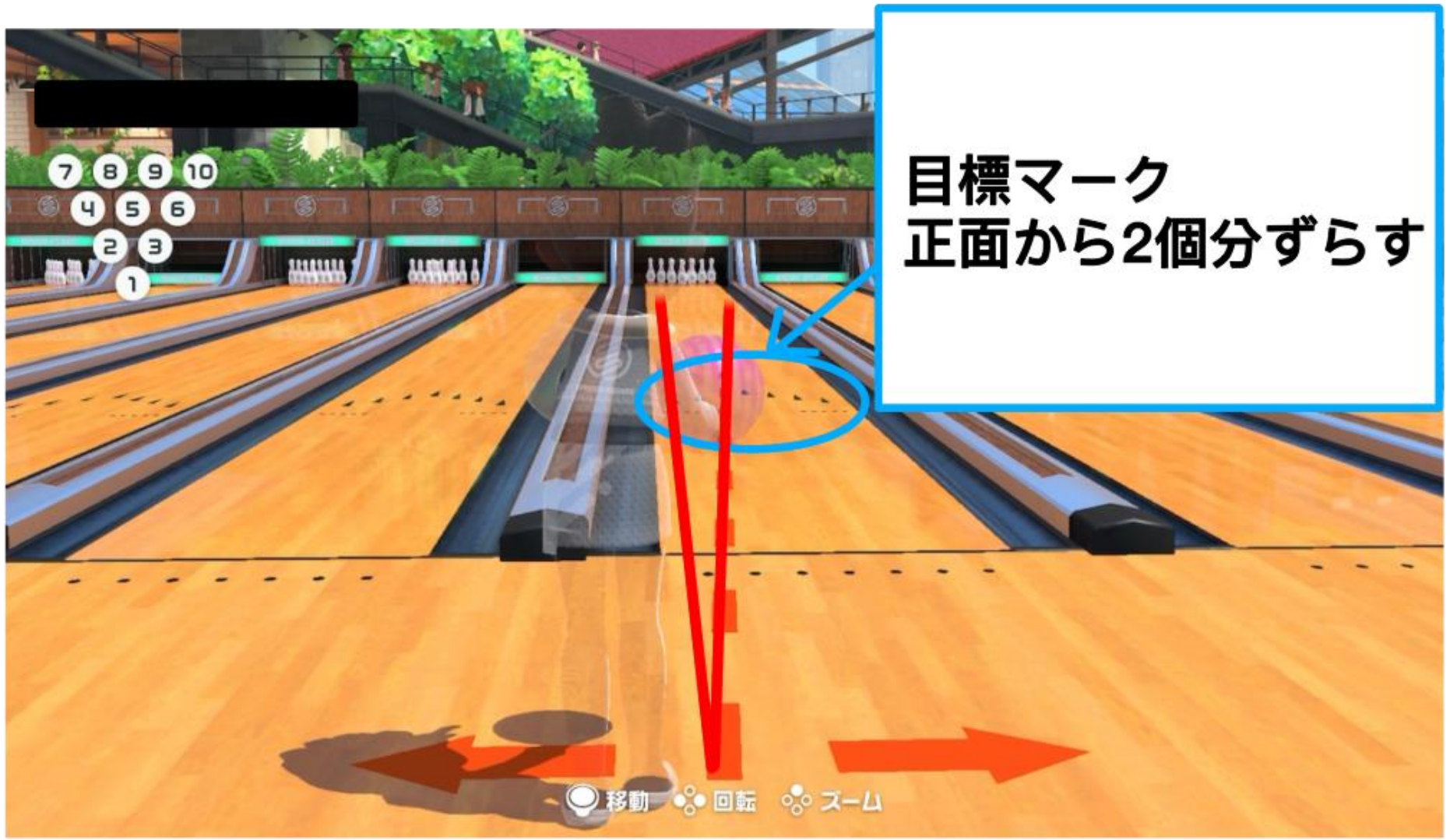


検証方法

- 任天堂ゲームソフト
  「Switchスポーツ」を用いる
- 右の図の軌道で投げた時、
  本当にストライクがとれるかどうか



検証結果



検証結果

まとめ

自信をもって「これを決めたら僕と結婚してください。」が言えるきがする！！

この位置、この角度で投げたらどんぴしゃり！！！