

PSYCHOMETRICS

The initial part of this project involved researching psychometric values that are commonly used in assessment evaluation. The report includes the following psychometric calculations.

Reliability Coefficient (Cronbach’s Alpha)
This value is used to determine whether the test can be considered a reliable measuring tool. It is defined by the following equation:

α = (K / (K - 1)) * (1 - (Σσ_{yi}² / σ_x²))

K is the sample size, σ_{yi}² is the variance of the individual test scores, and σ_x² is the variance of the total test scores.

Difficulty (P-value)
The question difficulty, called the p-value, is defined by the following equation:

p = (question average score / question max. score)

The p-value can range from 0 (hard) to 1 (easy). The target difficulty range is between 0.3 and 0.8.

Discrimination Index
The discrimination index determines how well a question helps to distinguish between students who did well on the test and students who did not do well on the test. It is determined by the following equation:

D = ((upper - lower) / group size)

upper and lower are the number of students who answered correctly in the top 27% and bottom 27% of test takers respectively, and group size is the size of the largest group. The discrimination index can range from -1 to 1.

Rit and Rir Values
Rit, item-test correlation, and Rir, item-rest correlation, indicate the correlation between an individual’s score on the question and their total exam score. Rit and Rir differ in that Rir subtracts the question score from the total test score. They are defined by the following equation:

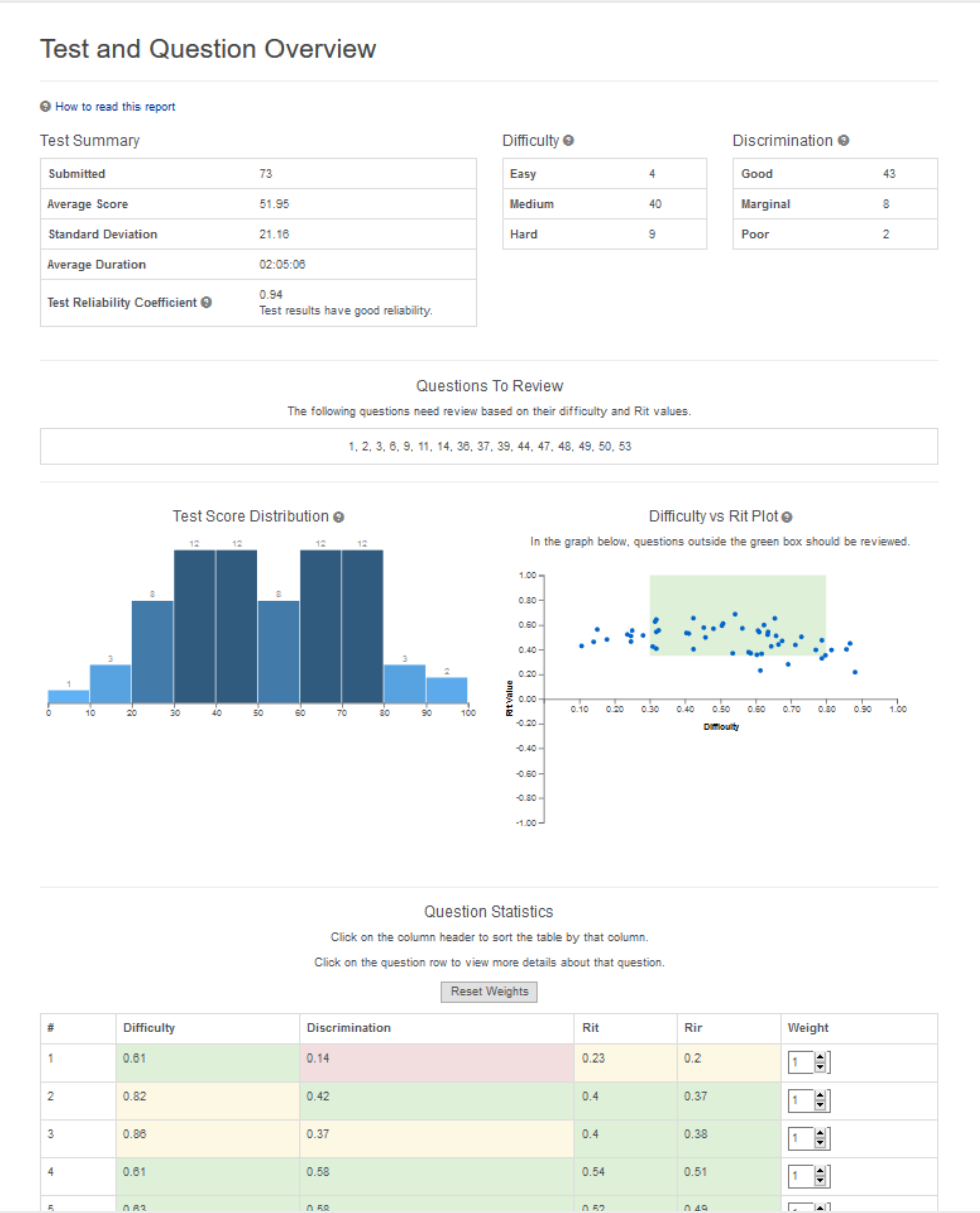
r_{xy} = ((N * Σx_iy_i - Σx_iΣy_i) / (N * S_xS_y))

N is the sample size, x_i is the individual question score, y_i is the individual total test score (Rit) or the individual total test score minus x_i (Rir), and S_x and S_y are the standard deviations of x and y respectively. The Rit and Rir values can range from -1 to 1.



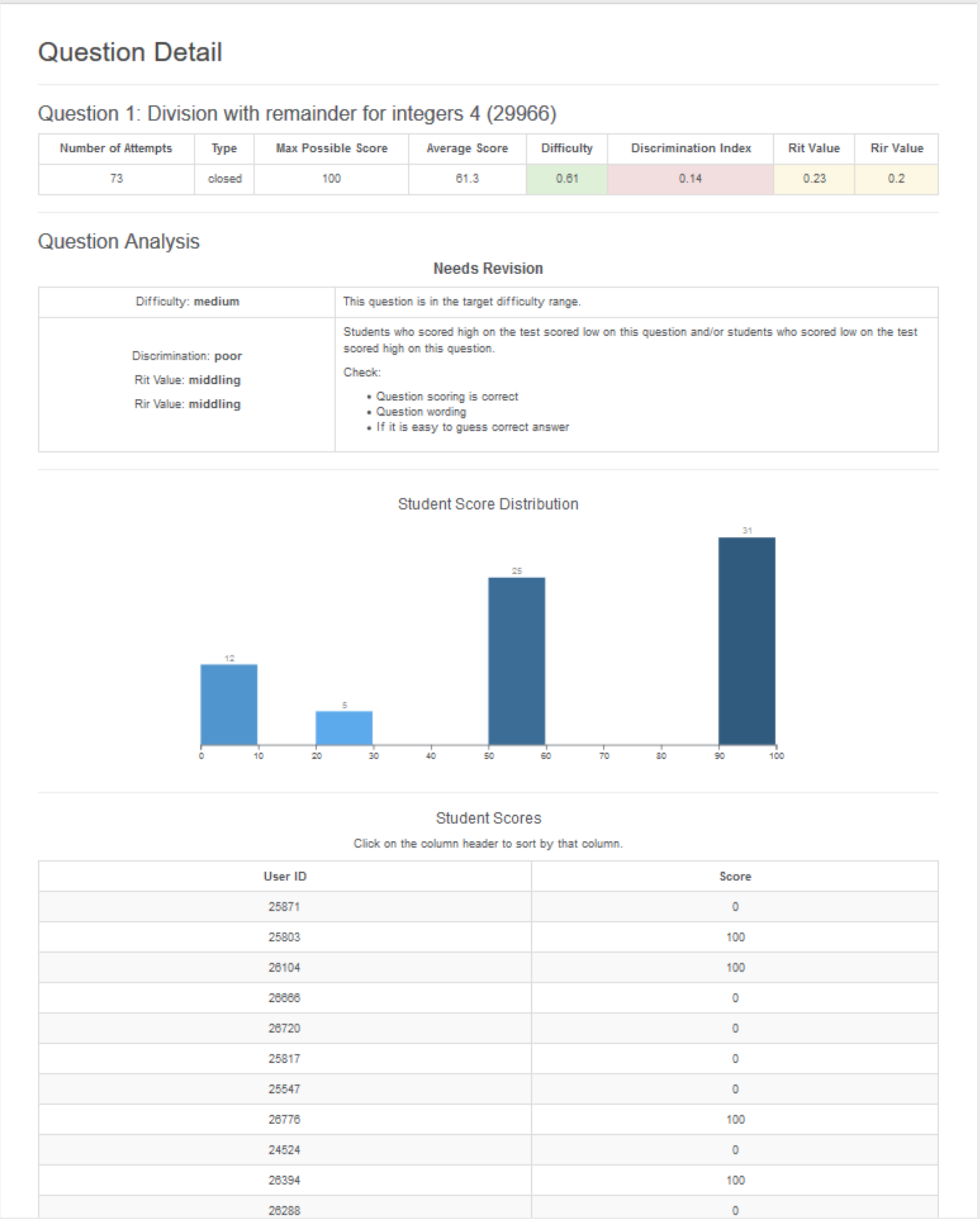
SOWISO ASSESSMENT PERFORMANCE REPORT

Team Members: Jack Middlebrook, Patrick Spencer, Stephanie Wiele



DESCRIPTION

Sowiso is a company based in Amsterdam that offers an online e-learning platform for math and science. Our team developed an assessment performance report for the Sowiso platform that instructors can use to psychometrically evaluate an assessment. The report is generated from the set of student scores on a given assessment and gives a summary of the overall quality of the assessment as well as the psychometric values associated with each question in the assessment. The main page shows an overview of the assessment results which includes the following: the average total score; the test reliability coefficient; the number of easy, medium, and difficult questions; the number of good, middling, and poor questions based on their discrimination index; a graph of the grade distribution; a graph and a list indicating which questions need to be reviewed; and a table of the psychometric values associated with each question. Clicking on a row in the question table brings up the detail page for that question which helps explain what the psychometric values indicate, as well as showing the student score distribution for that particular question.



IMPLEMENTATION

- Data used in the example report is loaded from csv files of anonymized test data that we received from Sowiso. Once the project is integrated in to the Sowiso platform, this will be live test data from students who are taking exams on the Sowiso platform.
- Assessment results data is passed to algorithms that calculate the psychometrics statistics for each question. The statistics are then displayed and visualized for a teacher to analyze.
- A customized question analysis is performed to give teachers insights into how students performed on the question and if it should be reviewed.
- Teachers can adjust the weighting of questions and see how the change in weights would update the Test Score Distribution and the Summary Test Statistics



```
public function calcDiscIndex($question_scores, $high_students, $low_students, $group_size)
{
    $high_students_score_total = 0;
    foreach ($high_students as $test_id => $average_test_score) {
        $high_students_score_total += $question_scores[$test_id];
    }
    $high_students_score_avg = $high_students_score_total / $group_size;

    $low_students_score_total = 0;
    foreach ($low_students as $test_id => $average_test_score) {
        $low_students_score_total += $question_scores[$test_id];
    }
    $low_students_score_avg = $low_students_score_total / $group_size;

    return ($high_students_score_avg - $low_students_score_avg) / 100;
}
```

```
if (($file = fopen($test_data_filepath, 'rb+')) !== false) {
    while (($data = fgetcsv($file, 0, 'delimiter:', ',')) !== false) {
        if ($row_count === 0) {
            foreach ($data as $header) {
                if (strpos($header, 'Question') === 0) {
                    $question_number = (int)substr($header, $question_number_start);
                    if (!array_key_exists($question_number, $question_idx_lookup)) {
                        $question_idx_lookup[$question_number] = $column_idx;
                    }
                    $column_idx++;
                }
            }
        } elseif ($row_count === 1) {
            $question_names = $data;
        } elseif ($row_count === 2) {
            $score_headers = $data;
        } else {
            $user_data[] = $data;
        }
        $row_count++;
    }
    fclose($file);
}
```

```
const bins = histogram(data_source);
y.domain([0, d3.max(bins, function(d) { return d.length; })]);

const colorScale = d3.scaleLinear()
    .domain(y.domain())
    .range([d3.rgb(barColor).brighter(), d3.rgb(barColor).darker()]);

const chartWrapper = svg.append('g')
    .attr('name: 'transform', 'values: 'translate(' + margin.left + ', ' + margin.top + '));

const bars = chartWrapper.selectAll('.bar')
    .data(bins)
    .enter().append('g')
    .attr('name: 'fill', barColor)
    .attr('name: 'class', 'value: 'bar');

const rects = bars.append('rect')
    .attr('name: 'fill', 'value: function(d) { return colorScale(d.length) });

const texts = bars.append('text')
    .attr('name: 'y', 'value: -4) // to move the text labels above the bar
    .attr('name: 'text-anchor', 'value: 'middle')
    .text('value: function(d) {
        if (d.length > 0) {
            return formatCount(d.length);
        }
    });
```

LANGUAGES & FRAMEWORKS

- Backend: PHP & CodeIgniter
- Frontend: JavaScript, HTML, CSS, Bootstrap
- Data Visualizations: D3.js

