

Evaluating the Quality of Multiple-Choice Tests with Automatically Generated Visualizations

*An Application of Scalable Vector Graphics
in Medical Education*

Dennis Toddenroth, Dr. med Thomas Frankewitsch

What does testing in medical education look like?

An exemplary one-best-answer item:

What is the cause of chickenpox?

- a) The bacterium *Bordetella pertussis*
- b) Morbillivirus
- c) Rubella virus
- d) *Varicella zoster virus*
- e) Human herpesvirus type six (HHV-6)

<- correct

What makes a good test item?

- Common assumption: The level of difficulty should not be extreme. An item will not reveal information about the relative competence of any candidate if everybody answers correct or incorrect.
- The item should truly measure 'competence'. In this case 'relatively competent' students are typically (although not necessarily) more likely to answer correctly.
- 'Competence' is presumed to be recognizable by better overall scores, so the set of remaining items is considered to allow the assessment of the presumed individual 'competence'.

Rationale for graphics creation

Limiting the analysis to the established parameters 'p-value' (defined as the fraction of correct choices) and 'discrimination index' (the correlation with the residual scores) is not always demonstrative and sometimes neglects information such as problems with single distractors.

„We recommend that attention be focused on the pattern of responses rather than on the difficulty level or discrimination index.“¹

„Indeed graphics can be more precise and revealing than conventional statistical computations.“²

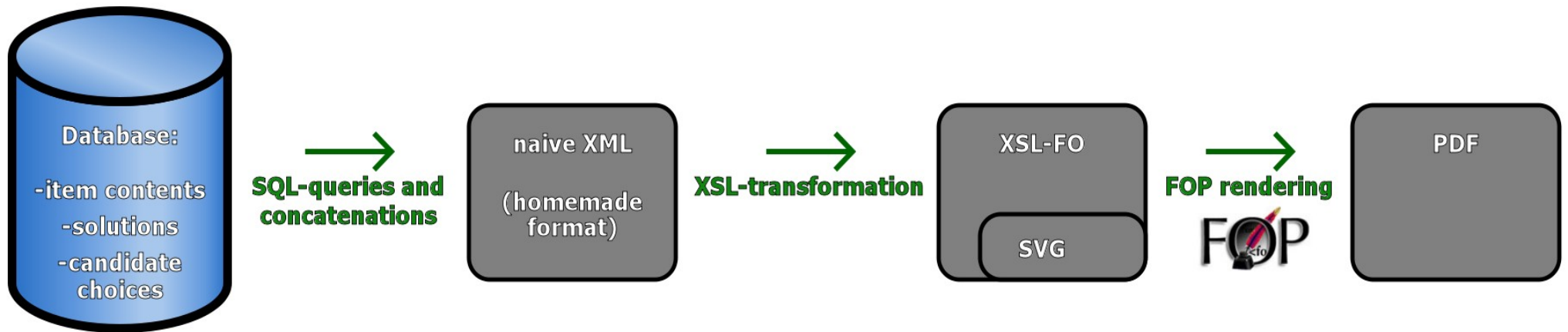
¹ Constructing Written Test Questions For the Basic and Clinical Sciences [Third Edition (Revised) January 1998] by Susan M. Case, PhD and David B. Swanson, PhD ©1996, 1998, 2001, 2002 National Board of Medical Examiners® (NBME®).

² The Visual Display of Quantitative Information [Second Edition (Revised) January 2001] by Edward R. Tufte ©2001 Graphics Press

Motivation for developing a custom-made solution

- Perpetual requirement to evaluate large amounts of data: Operating an off-the-shelf statistics software could become tedious.
- Evaluation procedure quite uniform, affording a reasonable level of automation
- Fine-grained control over layout, ability to structure outputs as needed
- General enthusiasm for programming and SVG

Key steps of the evaluation process








Program output I

```
<fo:instream-foreign-object>
  <svg width="68" height="11.5">
    <g style="fill:#9999CC; stroke:#000000">
      <rect height="10" stroke-width="0.5" x="1" y="1" width="21"/>
    </g></svg>
</fo:instream-foreign-object>

<!-- ... -->

<fo:instream-foreign-object>
  <svg width="68" height="11.5">
    <g style="fill:#666699; stroke:#000000">
      <rect height="10" stroke-width="0.5" x="1" y="1" width="3.599"/>
    </g></svg>
</fo:instream-foreign-object>
```

Antwort / wie oft gewählt	
A) 	✓ 35x (31.8%)
B) 	✗ 6x (5.5%)
C) 	✗ 12x (10.9%)
D) 	✗ 38x (34.5%)
E) 	✗ 19x (17.3%)

Note: Most namespace prefixes such as „<svg:rect/>“ have been eliminated to increase readability.

Program output II

```
<!-- sample chart for distractor 'E)' -->
<svg width="80" height="12">
  <defs>
    <linearGradient xml:id="55647">
      <stop style="stop-color: rgb(0,160,0)" offset="0%"/>
      <stop style="stop-color: rgb(160,160,160)" offset="100%"/>
    </linearGradient>
  </defs>
  <g style="stroke:#000000; fill: url(#55647)">
    <rect style="fill:black"
      height="12" width="0.5" stroke-width="0.5" y="0" x="39.5"/>
    <rect
      height="4.75" stroke-width="0.5" width="17.12" x="22.38" y="3.375"/>
  </g>
</svg>
```

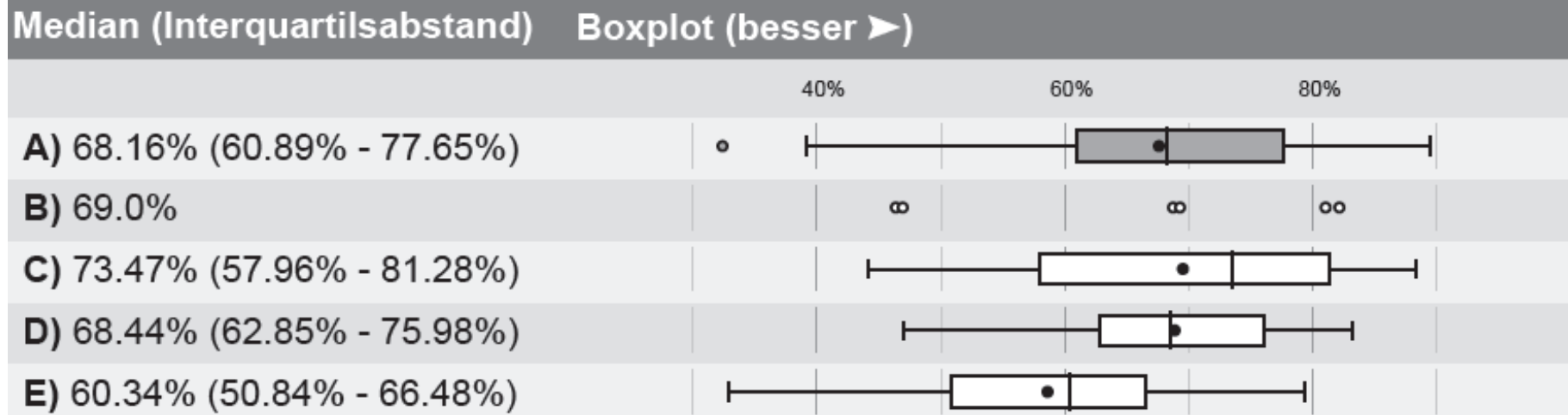
in dieser Gruppe richtig	Δ in %-Pkt. (besser ►)
67.16% (4271 von 6359)	
65.74% (706 von 1074)	↓ -1.42
69.51% (1493 von 2148)	↑ +2.35
69.10% (4599 von 6656)	↑ +1.94
58.60% (1993 von 3401)	↓ -8.56

Program output III

```

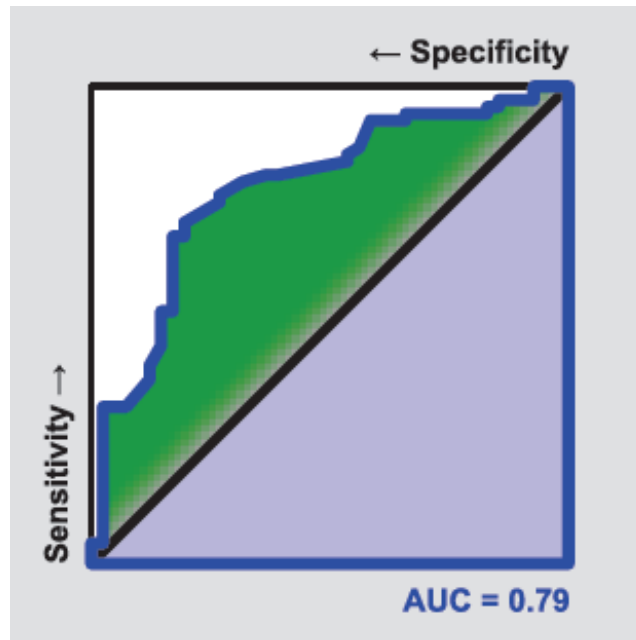
<!-- sample boxplot for candidate distribution 'A)' -->
<svg width="320" height="14">
  <g style="stroke:#000000; fill:#999999">
    <!-- ... -->
    <!-- maverick: -->
    <circle r="1.5" cy="7" cx="49.599"/>
    <!-- whiskers: -->
    <line y2="10" y1="4" x2="76.759" x1="76.75999"/>
    <line y2="10" y1="4" x2="277.56" x1="277.56"/>
    <line y2="7" y1="7" x2="277.56" x1="76.75999"/>
    <!-- quartiles: -->
    <rect height="10" width="67.040" y="2" x="163.56"/>
    <!-- median: -->
    <line y2="13" y1="1" x2="192.64" x1="192.64"/>
    <!-- mean: -->
    <circle style="stroke:#000000; fill:#000000" r="1.5" cy="7" cx="190.32"/>
  </g>
</svg>

```



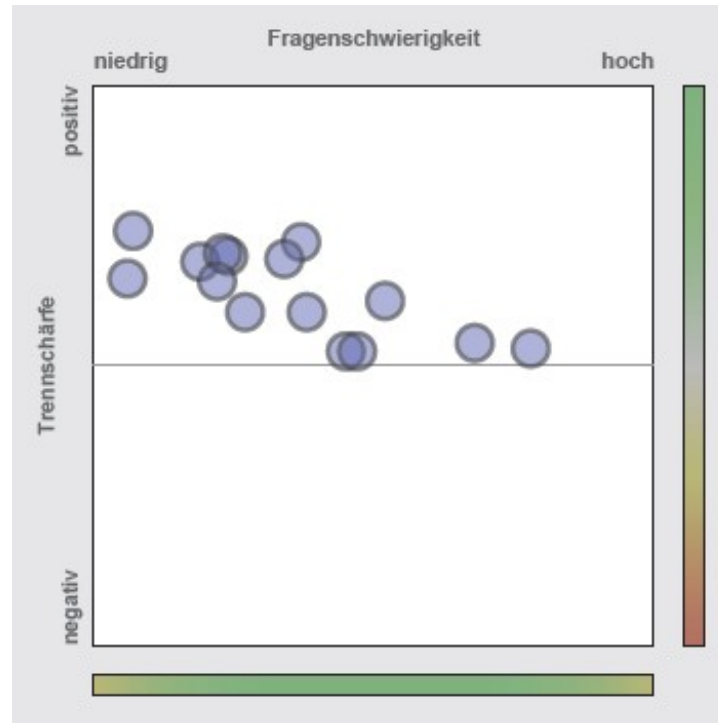
Program output IV

```
<path style="stroke:black;stroke-width:0.02;fill:url(#rocGrad13611)" d="M1 0 L1.0 0.0142 L1.0 0.0428 L0.975 0.0428 L0.926 0.328 ..."/>
```



Program output V

```
<g style="fill:blue;fill-opacity:0.32;stroke:black;stroke-  
width:.8px;stroke-opacity:0.55;">  
  <circle xml:id="datapoint_109" cx="45" cy="47.5" r="3.2"/>  
  <circle xml:id="datapoint_110" cx="19" cy="31.5" r="3.2"/>  
  <circle xml:id="datapoint_111" cx="24" cy="30.5" r="3.2"/>  
  <!-- ... -->  
</g>
```



Observation: A variety of diagrams can be consizely composed from the set of available SVG elements.

Observations on performance

- Evaluating several test items and assembling a multi-page report with numerous graphics can take minutes.
- As a consequence this approach appears not just yet ready for on-demand computation.
- Bottlenecks seem to be data aggregation and rendering PDF (involves parsing XML and producing layout).
- Workaround: a web-based frontend where users request reports. This application version subsequently emails download links, which refer to zipped collections of individually prepared files.

Alternative output target: XHTML+Javascript+SVG



Auswertung Klinische Chemie (SS 2008, 1. Klinisches Semester) - Mozilla Firefox

file:///D:/SVGOpen/1. Klinisches Semester/Auswertung_Klinische Chemie.xhtml

Klinische Chemie

[erzeugt 21.07.2008 13:16]

Fachsemester: 1. Klinisches Semester
Pruefungssemester: SS 2008
Anzahl Fragen: 20

Inhalt&Auswertung zu...

<< Frage 129 >>

The scatter plot shows the relationship between question difficulty (Fragenschwierigkeit) and discrimination (Trennschärfe). The x-axis ranges from 'niedrig' to 'hoch', and the y-axis ranges from 'negativ' to 'positiv'. A color scale on the right indicates the strength of the relationship, ranging from red (low) to green (high). A mouse cursor is pointing at a blue data point.

[Inhalt](#) | [Antworthäufigkeiten](#) | Distraktorenanalyse

Median (Interquartilsabstand)	Boxplot (besser >)
A) 66.48% (59.78% - 74.86%)	
B) 60.9%	
C) 67.32% (56.84% - 73.6%)	
D) 68.16% (60.89% - 78.21%)	
E)	

Schwierigkeitsindex: $p = 0.64$ / Trennschärfe: $r = +0.24$

Fertig

Alternative output target: XHTML+Javascript+SVG

- Firefox directly renders SVG elements integrated into XHTML.
- Javascript makes DOM-elements accessible, allowing image parts to become responsive and mutable (navigatable).
- The interactivity of the clickable scatterplot aims at making the identification of the odd one out easier.
- Most users have incompatible IE, configuration not (yet?) widely applicable

Output utilization

- PDF-reports are displayed on monitors or printed in grayscale. Vector graphics ensure a constantly impeccable visual appearance.
- Efficient PDF filesizes permit electronic report distribution without compromising image quality.
- The reports provide accessible feedback to test authors (and hopefully support the creation of high-quality test items).
- The reports are incorporated into decisions eliminating faulty items from grade evaluation.

More observations

- The design has been in operation for roughly 2½ years. The reports were gradually expanded along the way.
- Initial program versions were rather 'clipboard-composed' from FOP's examples.
- Approximately 8.500 test items and a total of 1.000.000 individual choices have been processed so far.