



Dipl.-Inf. Robert Ringel

The 5-Strands of Mathematical Proficiency - Weiterentwicklung zum Diagnosewerkzeug für den Lernfortschritt bei programmieren Lernen

31.01.2023

Einleitung – Forschungsgegenstand
Kognitive Modelle – 5-Strands-Modell

Konzeption von Lernstandserhebungen auf der Basis des 5-Strands-Modells

Durchführung und Auswertung der Erhebungen

Arten von Ergebnissen, Interpretation, Ableitung von Erkenntnissen

Zusammenfassende Wertung der Methode

Ziel der Arbeit: Entwicklung und Erprobung eines Rahmenkonzepts zum programmieren Lernen

Was heißt programmieren Lernen?

- geeignete kognitive Modelle für das Programmieren entwickeln

>>> den Entwicklungsstand dieser kognitiven Modelle dokumentieren

GIERL & LEIGHTON 2011: **Kognitives Modell**

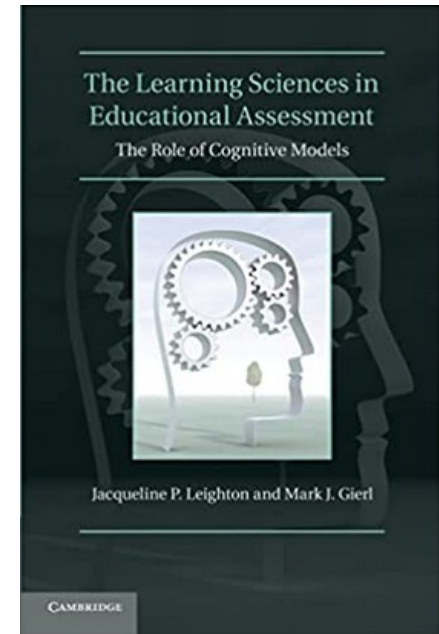
“Vereinfachte Beschreibung des menschlichen Problemlösens an standardisierten Lernaufgaben, die helfen Wissen und Können der Studenten einzuschätzen, das sie in unterschiedlichen Lernstufen erworben haben und das dabei unterstützt, studentische Leistungen zu diagnostizieren und vorherzusagen.”

Beachtung von

- | | |
|-------------------------|---|
| Granularität: | - Grad der fachlichen Detailliertheit |
| Messbarkeit: | - Validität, Reliabilität, Objektivität |
| pädagogischer Relevanz: | - Bezug zu den Lernzielen |

Neben anderen Modellen:

5-Strands-Model als Beispiel eines kognitiven Modells



KILPATRICK, SWAFFORD, FINDELL 2001:

5-Strands-Modell als Framework für Lernprozesse in Mathematik

1) Konzeptionelles Verständnis (CU)

“Conceptual understanding of mathematical principles, knowledge, operations, and relations.”

2) Handlungsroutine (PF)

“Procedural fluency in carrying out the skills associated with mathematical knowledge, in particular, showing flexibility, efficiency, and appropriateness in skill deployment.”

3) Problemlösen (PS)

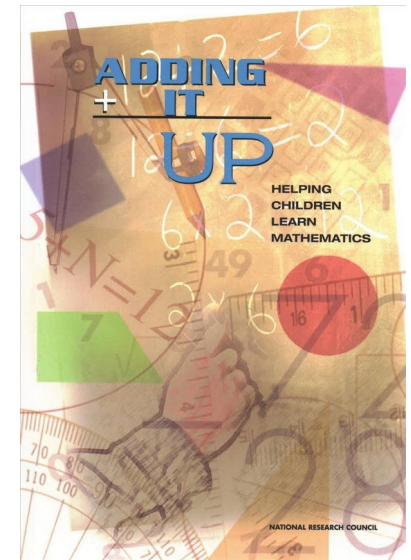
“Strategic competence as demonstrated by formulating, representing, and solving mathematical problems.”

4) Transfer (XF)

“Adaptive reasoning as shown by the ability to engage in logical thought, reflection, explanation, and justification.”

5) Positive Grundeinstellung (PD)

“Productive disposition toward viewing mathematics as useful and as a means for solving problems in the real world, and recognizing the effort and self-efficacy that must be devoted to acquiring competence in mathematics.”



>>> Transfer in Domäne “Programmieren lernen” am Beispiel des Python-Kurses

Die Grundidee des 5-Strands-Model

KILPATRICK, SWAFFORD, FINDELL 2001:

5-Strands-Modell als Framework für Lernprozesse in Mathematik

1) Konzeptionelles Verständnis (CU)

“Conceptual understanding of mathematical principles, knowledge, operations, and relations.”

2) Handlungswissen

“Procedural mathematical knowledge and appropriate application of mathematical skills.”

3) Problemlösestrategien

“Strategic competence in mathematics.”

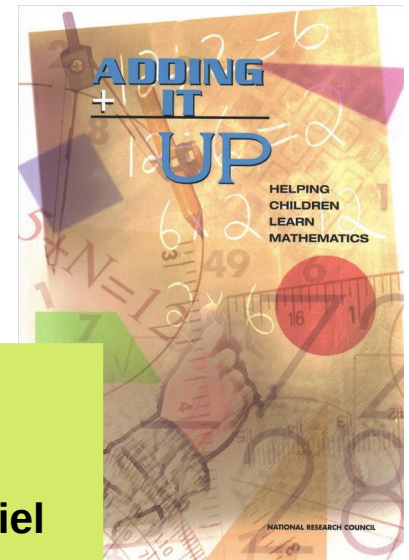
4) Transfer

“Adaptive reasoning as shown by the ability to engage in logical thought, reflection, explanation, and justification.”

5) Positive Grundeinstellung (PD)

“Productive disposition toward viewing mathematics as useful and as a means for solving problems in the real world, and recognizing the effort and self-efficacy that must be devoted to acquiring competence in mathematics.”

>>> **Transfer in Domäne “Programmieren lernen” am Beispiel des Python-Kurses**



Aufgabe

Transfer in Domäne “Programmieren lernen” am Beispiel des Python-Kurses,

um damit eine Diagnose-Werkzeug zu entwickeln.

ring

5-S-Klassifikation der Lehr-Lern-Ziele als Grundlage von Lernstandserhebungen

Strukturierung der Lehr-Lern-Ziele

Grundlagen des Programmierens

Definition von Lehr- Lern-Zielen

Begriffe und Anweisungen: Variable, Datentyp, Berechnung, Eingabe, Ausgabe, Verzweigung, Schleife, Steuerung des Programmablaufs, logischer Ausdruck, Debugging
Fähigkeiten: Programme lesen, kommentieren, anpassen, entwickeln, Fehler suchen

5-S-Klassifikation der Lehr-Lern-Ziele

CU: Begriffe erklären, klassifizieren
z.B.: logischer Ausdruck – Steuerung des Programmablaufs
PF: Routinehandlung ausführen
z.B.: Typumwandlung vornehmen
PS: Lösen einer Programmieraufgabe nach einem bekannten Schema, z.B. Währungsumrechnung
XF: Problemlösung unter Verwendung neuartiger Ansätze über Analogieschluss
z.B.: Alg. zur Grenzwertüberwachung
PD: Skala "Einstellung"

Datenstrukturen

Begriffe und Anweisungen: Liste (homogen, heterogene, Records), Tupel, Set, Dictionary, Index, Schlüssel, Wert, zusammengesetzter Schlüssel, Operationen mit Datenstrukturen
Fähigkeiten: Datenstrukturen lesen, skizzieren, anpassen, entwickeln,

CU: z.B.: Index vs. Schlüssel erklären; negativer Index?
PF: z.B.: Teilliste extrahieren, für MW-Berchng. geg. Datenstruktur skizzieren
PS: z.B.: Auswertung Lagerbestand (MW, max, min, Limits überwachen)
XF: z.B.: Initialisierung von Datenstrukturen
PD: Skala "Selbstwirksamkeit"

Programmiertechniken

Begriffe und Anweisungen: Funktion, Bibliothek, Fehlerbehandlung, Textdatei, Trennzeichen, Dateioperationen
Fähigkeiten: Funktionen lesen, dokumentieren, anpassen, entwickeln, Textdateien lesen und schreiben

CU: z.B.: Zweck eines Trennzeichens in Datei? Wozu dienen Funktionen?
PF: z.B.: Programm-Menue erweitern Default-Parameter setzen
PS: z.B.: Kursverläufe aus Textdatei einlesen
XF: z.B.: Struktur einer Textdatei entwerfen
PD: Skala "Anstrengung"

In der weiteren Betrachtung wird zunächst nicht weiter auf die Productive Disposition eingegangen.

Die Aussagen dieser Komponente sind unspezifisch und es bestehen Zweifel an der Eignung der gewählten Skalen für die Selbstauskünfte.

Die Erhebung von Productive Disposition könnte ein Thema für eine eigenständige Forschungsarbeit sein.

... mehr dazu am Ende (als Zugabe), wenn Zeit und Interesse bestehen.

Ziel: Validität der Aufgaben für die Messung der 5-S-Komponenten

Fokus auf inhaltliche Gestaltung der Aufgaben,
Formulierung der Aufgabenstellung und der Beispiellösung
als Grundlage des Lösungsbewertung

Ziel: Reliabilität >>> 5-S-Komponenten in mehreren Aufgaben messen

Pflicht-LV im 1. Semester Wirtschaftsingenieurwesen

3 LSE im Rahmen der LV in einem Semester

LSE-1: Grundlagen des Programmierens

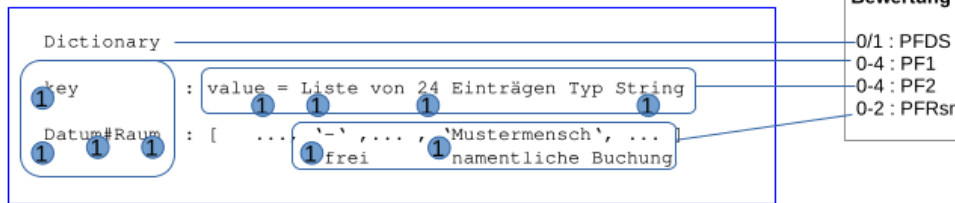
LSE-2: Datenstrukturen

LSE-3: Programmiertechniken

-
- Bearbeitung der Aufgabenblätter als Word-Datei;
 - Hilfsmittel sind zulässig; Bearbeitungszeit ca. 80 min

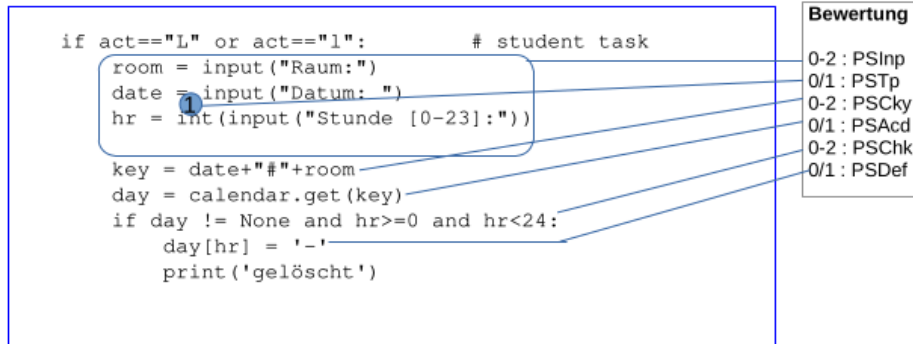
 - vor Beginn: Hinweise zur Arbeit mit der Word-Datei
kurze Info zu den Aufgaben und zur Ergebniserwartung
 - während Test: keine individuellen Hilfen
 - nach dem Test: Speichern der Word-Datei auf eigenen Gerät

Welche Datenstruktur wird benutzt? Skizzieren Sie die Datenstruktur und erklären Sie ihre Funktion, indem Sie die Bestandteile oder Inhalte benennen.



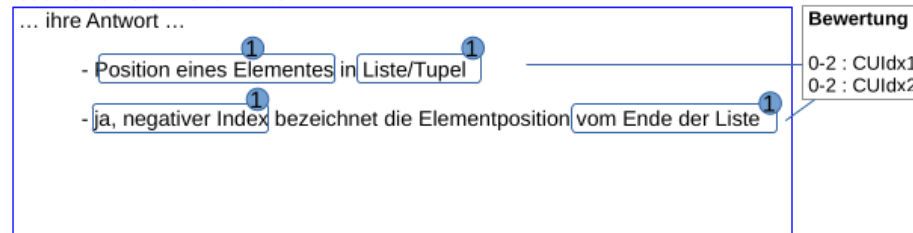
Schreiben Sie den Pythoncode zum Löschen einer Buchung. Nach dem Löschen ist der Eintrag der gelöschten Stunde wieder '-'

Stellen Sie sicher, dass nicht außerhalb gültiger Stundenzahlen gelöscht wird!
Codeumfang ca. 10 Zeilen



Verständnisfrage

Was ist ein Index? Kann ein Index auch negativ sein?
Beschreiben Sie in Stichworten!



In LV nach der Lernstandserhebung:

Phase 1 der Auswertung

- Lehrkraft stellt Aufgabe für Aufgabe Lösung vor und erklärt die Bewertung (Punktvergabe)
- Studierende bewerten demgemäß die eigene Lösung
→ bei Unklarheit fragen!
- am Ende als PDF speichern

Phase 2 der Auswertung (freiwillig):

- bewertetes PDF in OPAL hochladen
- erreichte Punktzahl mit Formular in Cloud eintragen
- Ergebnis: erster Eindruck zum Gesamtergebnis

Beobachtung: Studierende arbeiten, ehrlich und engagiert – stellen im Zweifel Fragen

Phase 3 statistische Auswertung – Ziele:

Querschnitt: Info über erreichte Lernziele anhand von
Aufgabenlösungen

Längsschnitt: Info über Entwicklung des Lernstands
in 5-S-Kategorien

Prognose: Zusammenhang zwischen den 5-S-Komponenten
und dem Prüfungsergebnis?

Methodik Nutzung von relativen Punkterreichungsgraden (Ratio)

Problem: jede 5-S-Teilkomponente und jede LSE hat
andere absolute Punktzahlen.

Lösung: $\text{Ratio} = \text{erreichte Punkte} / \text{mögliche Punkte}$

Auswertung 1: Erreichte Lernziele im Querschnitt

LSE-2 : N=29

Quote ca. 50 %

Aussage: 88% der Studierenden haben korrektes Konzeptverständnis von negativem Index

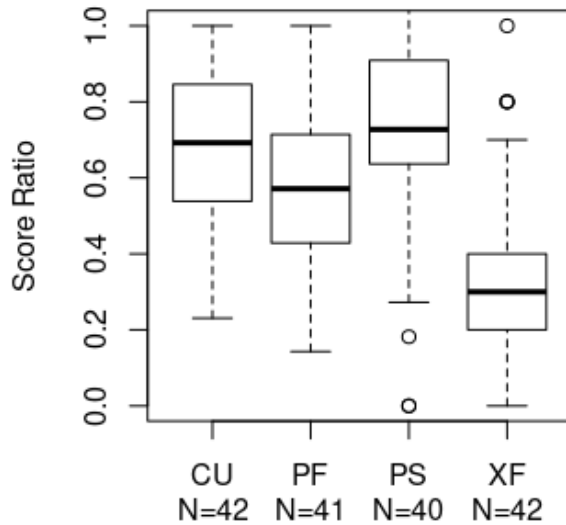
75 % Grenze

50 % Grenze

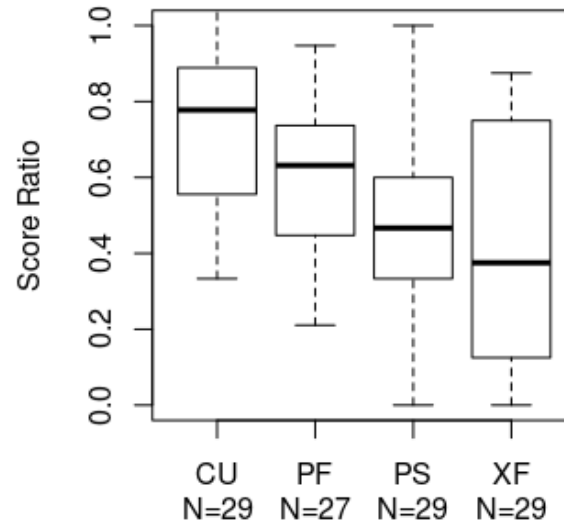
Ratio	Item	Erläuterung
1	PFBk	PF – Schleifenabbruch
0.96	PFDS	PF – DS-Dictionary
0.93	PFIIn	PF – Dateneingabe
0.9	PFLp	PF – Schleife
0.88	CUIdx2	CU – negativer Index
0.81	PFDIs	PF – Datenanzeige
0.79	PSCky	PS – Schlüssel erzeugen
0.79	PSAcD	PS – Datenzugriff Dictionary
0.78	CUIdx1	CU – Index ist Elementposition
0.74	PFCIk	PF – Zeit anzeigen
0.71	PSInp	PS – Dateneingabe
0.67	CUWh	CU – while-Schleife
0.62	PSTp	PS – Typumwandlung
0.61	XFTsk	XF – Liste Tooljobs
0.61	CUFr	CU – for-Iteration
0.55	PFShw	PF – Datenanzeige
0.53	PF1	PF – zusammengesetzter Schlüssel
0.44	PF2	PF – Value: List of Strings
0.41	PSDef	PS – Daten löschen / default setzen
0.38	PSlTr	PS – Iteration Tools
0.36	XFZro	XF – Jobtime initialisieren
0.32	PFRsn	PF – Bestandteile benennen
0.29	PSTool	PS – Anzeige Tool
0.26	PSChk	PS – Datencheck
0.26	PSTime	PS – Anzeige Zeit
0.17	XFLtj	XF – Anzeige Länge Jobliste

Auswertung 2: 5-S-Komponenten im Längsschnitt

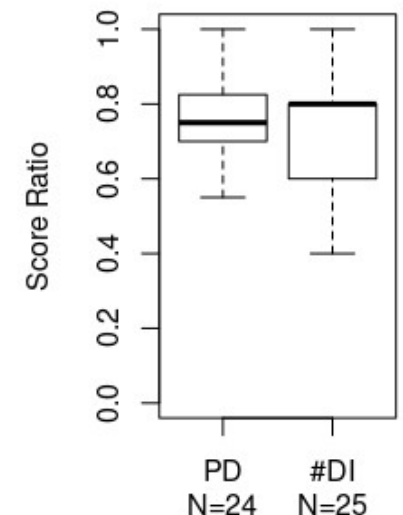
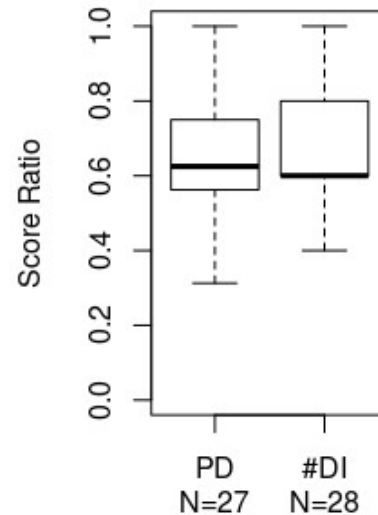
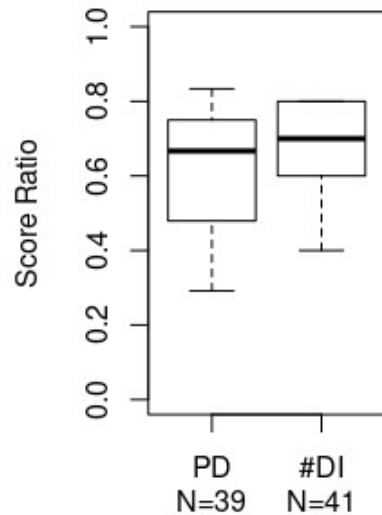
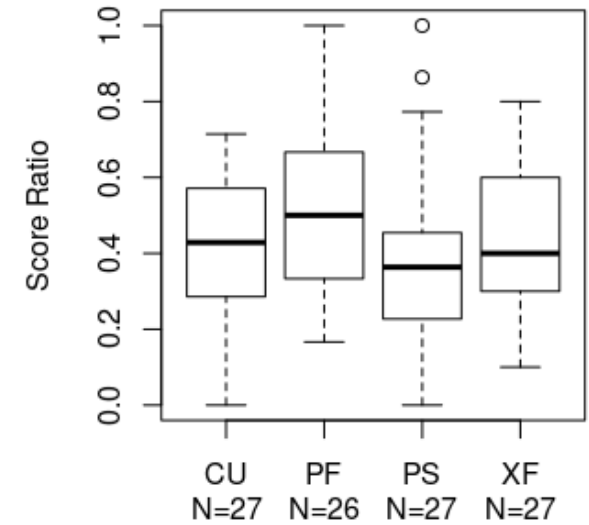
LSE-1 Wing Nov. 2022



LSE-2 Wing Dez. 2022



LSE-3 Wing Jan. 2023



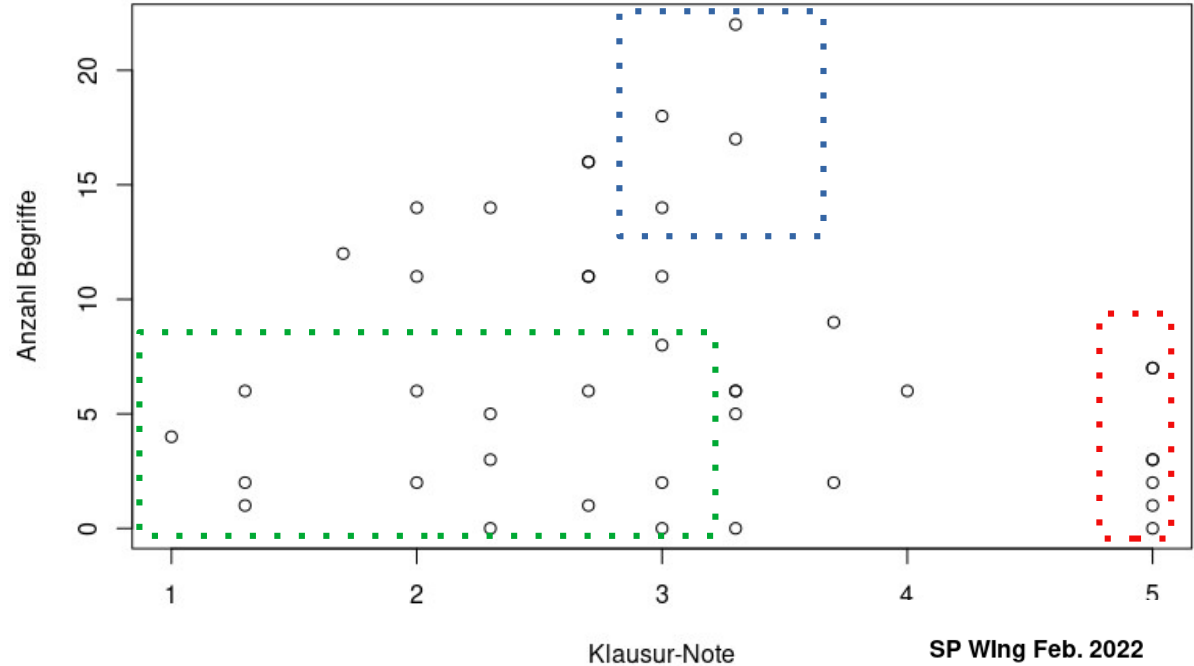
Zu Semesterbeginn: Erhebung des Vorwissens

Kennen Sie folgende Begriffe?

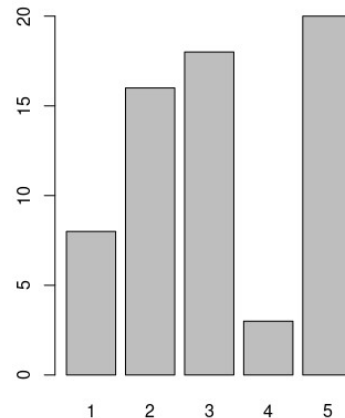
Kreuzen Sie jene Begriffe an, die Ihnen so vertraut sind, dass Sie diese erklärend beschreiben könnten!

- Variable
- Quelltext
- Datentyp
- Algorithmus
- Wertezuweisung
- Programm
- Datenstruktur
- Rückgabewert
- Array
- Anweisungsblock
- numerischer Operator
- Funktion
- Schleife
- Debugging
- Rekursion
- Programmbibliothek
- Parameterliste
- logischer Ausdruck
- Endlosschleife
- Referenz
- HashMap
- Laufzeit
- Exception
- Filehandle
- Klasse
- Instanz
- Konstruktor
- Vererbung
- Entwicklungsumgebung
- Design Pattern

Vorwissen vs. Klausurnote

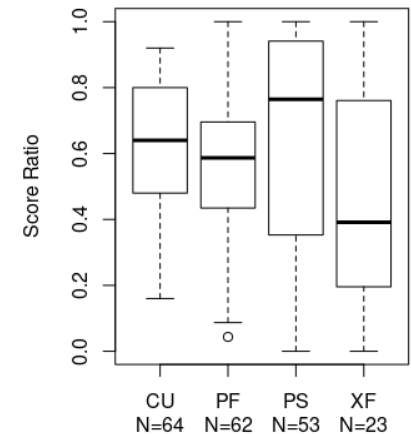


Counts of Exam Marks



N=65
78 Studierende
haben das
Semester
begonnen.

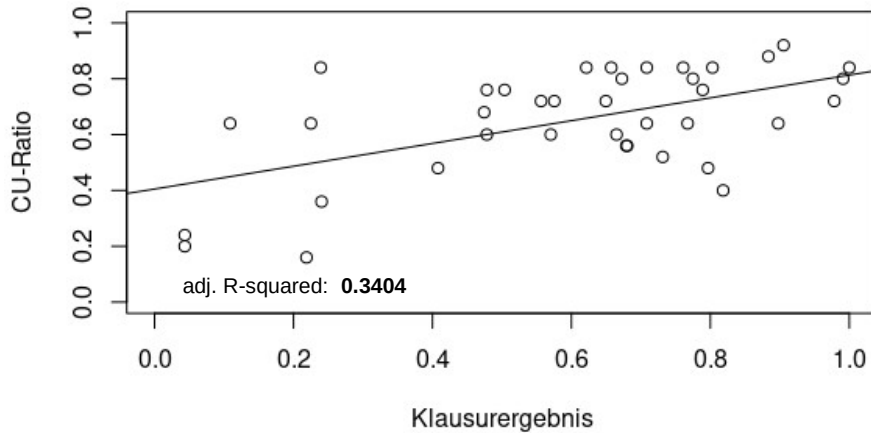
SP Wing Feb. 2022



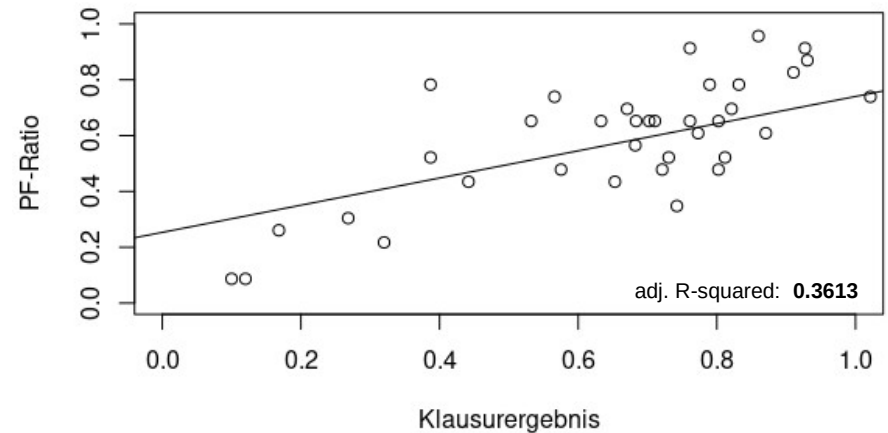
Auswertung 3: Prüfungsergebnis 2021

Kann eine der 5-S-Komponenten das Klausurergebnis vorhersagen?
Methode: lineare Regression; Klausurergebnis nicht als Note sondern Ratio

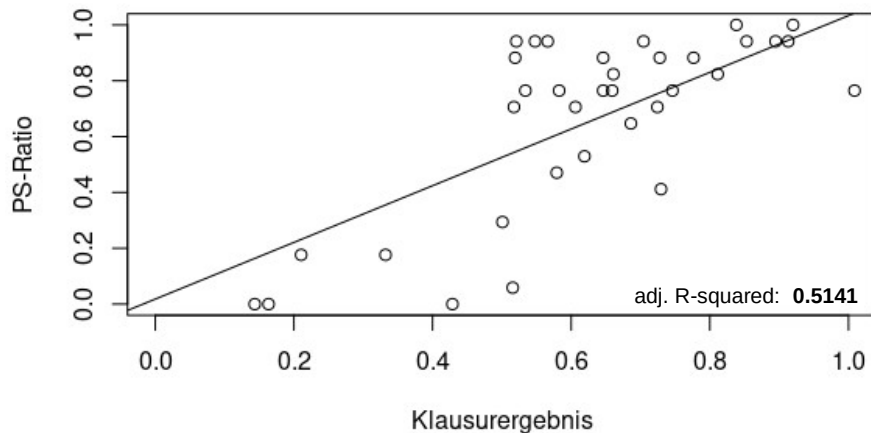
CU vs. Klausurergebnis



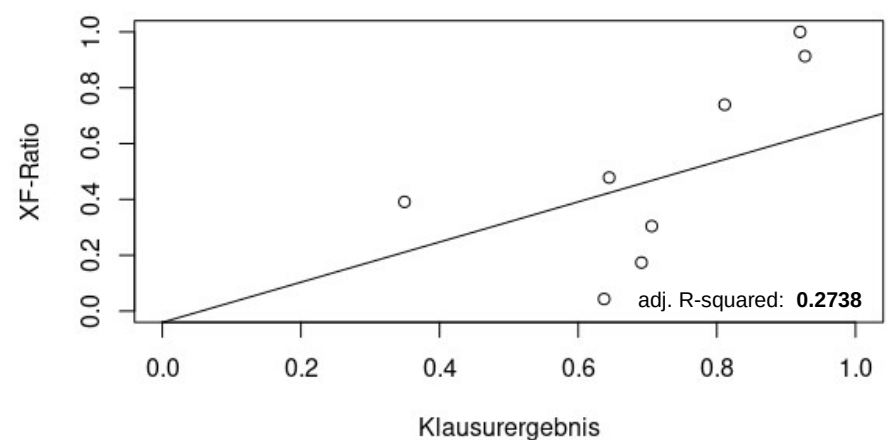
PF vs. Klausurergebnis



PS vs. Klausurergebnis

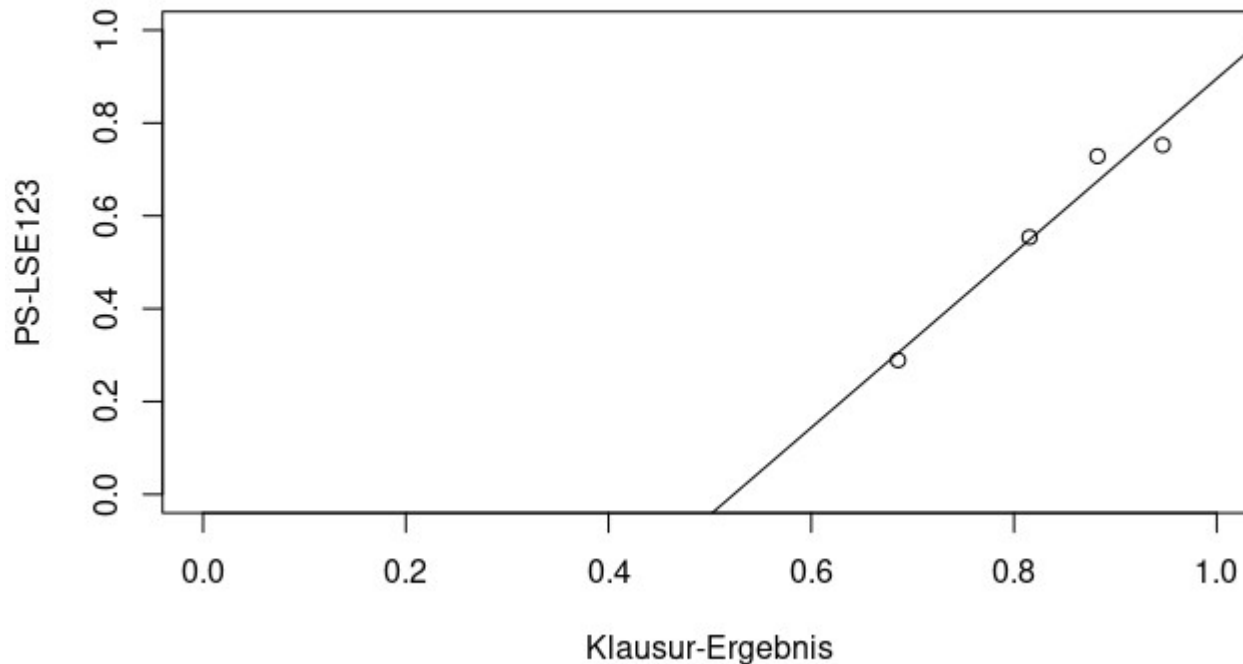


XF vs. Klausurergebnis



Kann die PS-Komponente der LSE das Klausurergebnis vorhersagen?
Methode: lineare Regression; Klausurergebnis nicht als Note sondern Ratio

PS-LSE123 vs. Klausurergebnis



	Note	ExamRatio	PS_RATIO123
1	1.0	0.9466667	0.7523810
2	2.0	0.8153794	0.5539683
3	1.7	0.8824211	0.7285714
4	3.0	0.6856266	0.2888889

Glückstreffer oder Indikation eines möglichen Zusammenhangs?

Zusammenfassung

- Für Studierende:**
- sehr gute Möglichkeit für internes Feedback
 - sehr gutes Training für die Klausur
 - Selbstbewertung – Erklärung der Lösung und Punktvergabe:
“Da sieht man, wie sie als Informatiker ticken und denken.”
 - eine Möglichkeit zum Lernen

- Für Lehrkraft:**
- gibt ein detailliertes und vielschichtiges Abbild zur Erreichung der Lernziele

Für den Lehr-Lernprozess:

- Ergebnisse geben Möglichkeit zum gezielten Üben in Defizitbereichen ... zusätzliche Übungsaufgaben bereitstellen
- Möglichkeit die Lernenden zu motivieren, insbesondere auch anhand erreichter positiver Ergebnisse
- Motivation für die Lehrkraft

-
- Stärken:**
- Methode skaliert – ist in großen Gruppen machbar
- Schwächen:**
- freiwillige Teilnahme – also keine Zufallsstichprobe!
 - Productive Disposition – Messung ist komplett unbefriedigend

Fazit: Lernstandserhebung nach 5-Strands-Model hat das Potenzial eines praktikablen kognitiven Modells des programmieren Lernens.

Vielen Dank -

Zeit für Diskussion

Auswertung 1: Erreichte Lernziele im Querschnitt

LSE-1 : N=42 Diff: 0.67

Ratio	Item	Erläuterung
1	PFPurp	PF – Programmzweck erkennen
0.93	PSLoop	PS – Schleife programmieren
0.9	CUDbg1	CU – Debugging
0.88	CULp	CU – Schleife
0.86	CUDbg2	CU – Debugging
0.83	PSErr	PS – Fehlerfall erkannt
0.81	PSType	PS – Typumwandlung
0.76	CUDes	CU – Verzweigung
0.74	PSVInit	PS – Variable initialisieren
0.74	CUDbg4	CU – Debugging
0.69	CUDbg3	CU – Debugging
0.68	PFComm	PF – Programmkommentare
0.67	PSAlg1	PS – Wertesumme/Wertezählung/MW
0.67	PSDisp	PS – Ergebnisanzeige
0.65	PFEnd	PF – Programmende ergänzen
0.64	CULpDesc	CU – Schleife beschreiben
0.64	XFUsr	XF – Haltepunkt
0.61	XFAlg2	XF – bedingte Zählung
0.57	CUlogExp3	CU – log. Ausdruck beschreiben
0.54	CUDesDesc	CU – Verzweigung beschreiben
0.54	PSIfBrk	PS – Verzweigung für Programmende
0.52	CUlogExp1	CU – log. Ausdruck beschreiben
0.38	CUlogExp2	CU – log. Ausdruck beschreiben
0.37	PFLogE	PF – log. Ausdruck anpassen
0.32	XFVInit	XF – Variable initialisieren
0.26	XFInd	XF – logischer Ausdruck
0.24	PFInc	PF – Fehlendes Bestandteil finden
0.24	XFNDbg	XF – Note-Debug
0.23	XFCond	XF – Bedingung zur Statusanzeige

LSE-2 : N=29 Diff: 0.66

Ratio	Item	Erläuterung
1	PFBk	PF – Schleifenabbruch
0.96	PFDS	PF – DS-Dictionary
0.93	PFin	PF – Dateneingabe
0.9	PFLp	PF – Schleife
0.88	CUIdx2	CU – negativer Index
0.81	PFDIs	PF – Datenanzeige
0.79	PSCKy	PS – Schlüssel erzeugen
0.79	PSAcD	PS – Datenzugriff Dictionary
0.78	CUIdx1	CU – Index ist Elementposition
0.74	PFClk	PF – Zeit anzeigen
0.71	PSInp	PS – Dateneingabe
0.67	CUWh	CU – while-Schleife
0.62	PSTp	PS – Typumwandlung
0.61	XFTsk	XF – Liste Tooljobs
0.61	CUFr	CU – for-Iteration
0.55	PFSHw	PF – Datenanzeige
0.53	PF1	PF – zusammengesetzter Schlüssel
0.44	PF2	PF – Value: List of Strings
0.41	PSDef	PS – Daten löschen / default setzen
0.38	PSlTr	PS – Iteration Tools
0.36	XFZro	XF – Jobtime initialisieren
0.32	PFRsn	PF – Bestandteile benennen
0.29	PSTool	PS – Anzeige Tool
0.26	PSChk	PS – Datencheck
0.26	PSTime	PS – Anzeige Zeit
0.17	XFLtj	XF – Anzeige Länge Jobliste

Auswertung 1: Erreichte Lernziele im Querschnitt

LSE-3 : N=27 Diff: 0.74

Ratio	Item	Erläuterung
	1 XF_Str	XF – Struktur der Textdatei
0.89	XF_Sep	XF – geeignetes Trennzeichen
0.71	PS_FOp	PS – Datei öffnen
0.7	PS_DctW	PS – Daten speichern
0.7	CU_nrt1	CU – Funktion ohne Rückgabewert
0.7	CU_nrt2	CU – Funktion ohne Rückgabewert
0.7	XF_Emp	XF – Textzeile nur Schlüsseldata
0.63	CU_tew	CU – Fehlerbehandlung Datei schreiben
0.61	PF_Comm	PF – Programmkommentare
0.59	PS_lfn	PS – Verzweigung
0.59	PS_re1	PS – return Erfolg
0.56	PF_Snow	PF – Programmresultat angeben
0.54	PF_Purp	PF – Programmzweck
0.54	PS_FCI	PS – Funktionsaufruf
0.52	PS_DctR	PS – Daten lesen
0.52	PS_Dct	PS – leeres Dictionary erzeugen
0.48	PS_re0	PS – return Daten existieren bereits
0.46	PF_Comp	PF – Code vervollständigen
0.4	XF_Lo1	XF – äußere Schleife
0.33	PS_W1	PS - Key und Trennzeichen schreiben
0.33	PS_Tx	PS – Fehlerbehandlung
0.28	PS_FCI	PS – Datei schließen
0.27	XF_W3	XF – Zeilenende schreiben
0.27	PS_EMg	PS – Fehleranzeige
0.23	XF_Lo2	XF – innere Schleife
0.16	CU_ret	CU – mehrere return-Anweisungen
0.14	PS_W2	PS - Daten und Trennzeichen schreiben