Process Data

Susu Zhang, Qiwei He, Sunboem Kwon

Learning Objectives

1. Understand the structure and key characteristics of process data collected from large-scale digital assessments.
2. Gain familiarity with approaches for extracting features from action sequences.
3. Implement basic data wrangling and process feature extraction in R using assessment log data.
4. Interpret the structure, distribution, and substantive meaning of process-derived features.
5. Learn how process-derived features can address questions of test design, measurement reliability, and fairness.

Module Sections

- Section 1: Introduction to Process Data
- Section 2: Extracting Process Data Features
- Section 3: *Hands-on Coding Exercise: Data Wrangling and Feature Extraction in R*
- Section 4: Applications of Process Features in Measurement and Education

About the authors

Susu Zhang, Ph.D.

Dr. Susu Zhang is an Associate Professor of Psychology and Statistics at the University of Illinois Urbana-Champaign. Her work develops methods to incorporate process data into psychometric and statistical models to improve educational measurement. She has worked with process data from PISA, PIAAC, and NAEP and has led projects funded by IES and AERA-NSF analyzing the NAEP process data. She is a co-author of the ProcData R package and has co-hosted multiple short courses on statistical learning for process data.

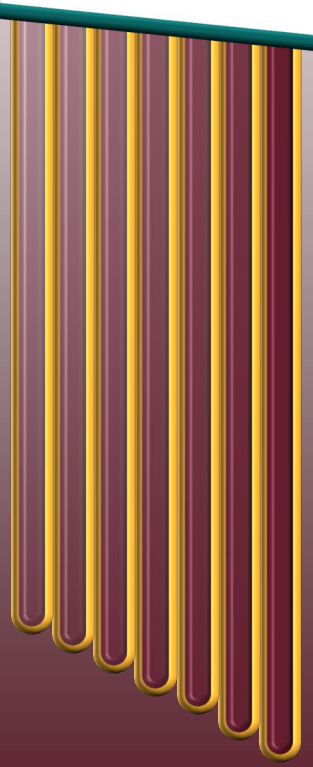
Qiwei He, Ph.D.

Dr. Qiwei He is a Provost's Distinguished Associate Professor in the Data Science and Analytics Program and the Founder and Director of the AI Measurement and Data Science Lab at Georgetown University. Her research advances psychometric and data science methodologies for multimodal process data, integrating sequence mining, text mining, psychometric modeling, and machine learning in educational and psychological assessment. She is particularly engaged in national and international large scale assessments, developing analytic frameworks that illuminate human behavior, learning processes, and measurement validity.

Sunbeom Kwon

Sunbeom Kwon is a Ph.D. candidate in Quantitative Psychology at the University of Illinois Urbana-Champaign, where he also earned an M.S. in Applied Statistics. His research focuses on psychometrics and data science, and his current work includes process data analysis, copula-based latent variable models, AI fairness evaluation, and AI-assisted psychometric models. He completed the Ida Lawrence Research Summer Internship at ETS in 2024. Previously, he earned a B.S. and an M.S. in Psychology from Sungkyunkwan University.

Section 1: Introduction to Process Data



1

1 Introduction to Process Data

Section Learning Objectives

Learning Objective 1
Define process data and learn about the data structure.

Learning Objective 2
Identify examples of process data from publicly available assessment datasets.

Learning Objective 3
Develop intuition on process data's potential utility for measurement and education.

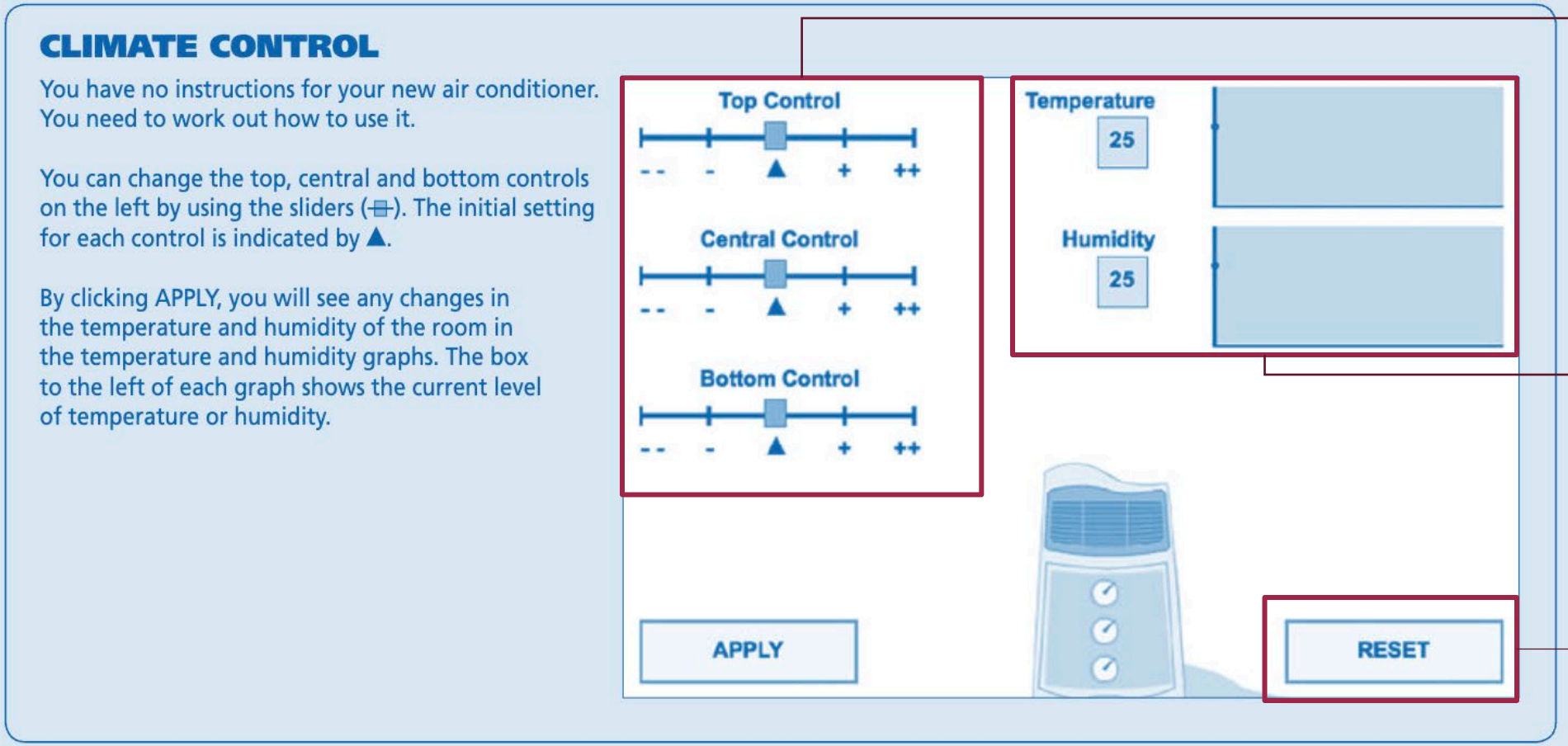
Behavioral Evidence from Computerized Tests

- The main task of educational measurement is to generate valid and reliable scores that reflect students' knowledge, skills, and attributes, enabling meaningful interpretation and informed educational decisions based on observed behavioral evidence collected from tests.

What types of behavioral evidence can we collect?

- Final scores on a test item
- Reaction times
- …
- Test-taking process data - the sequence of actions performed by an examinee in pursuit of solving a problem

Example 1: PISA 2012 Complex Problem Solving



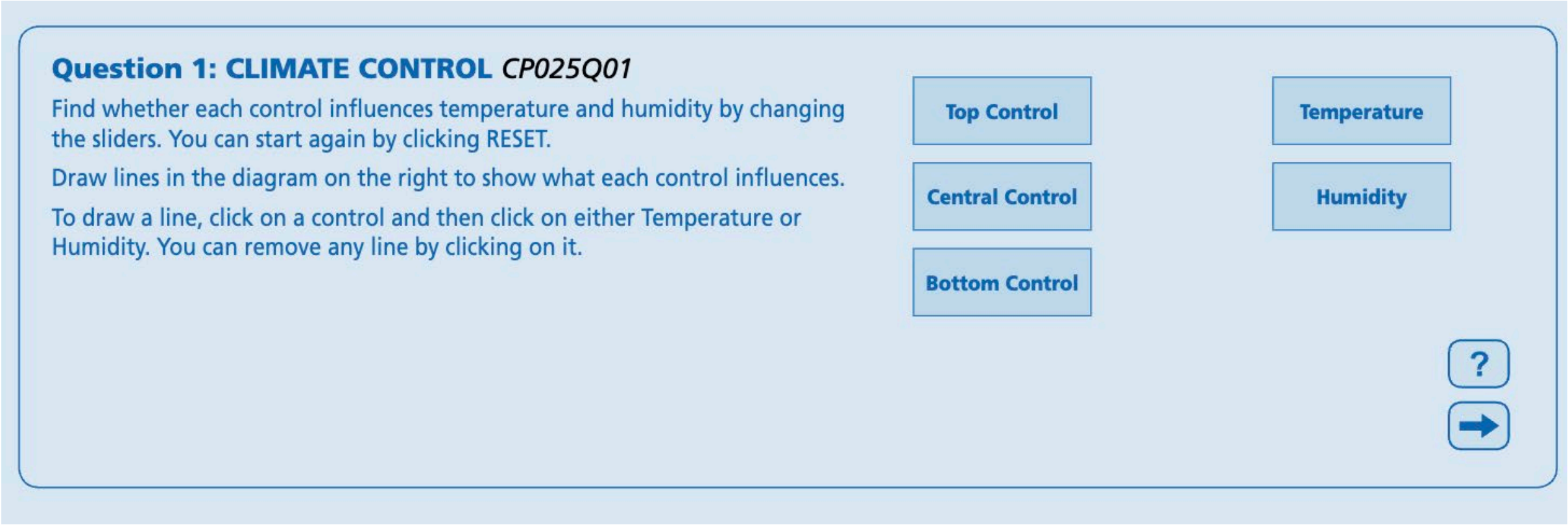
Students can move the three control bars from neutral (Δ) to 2 + /− positions.

After clicking "APPLY", these line plots add two new data points for temperature and humidity.

Clicking "RESET" will reset the three control bars to the neural (Δ) position.

PISA 2012 Climate Control item. Retrieved from OECD (2014), *PISA 2012 Technical Report*, PISA, OECD Publishing, Paris, https://doi.org/10.1787/6341a959-en.

PISA 2012 Climate Control Item – Final Score



PISA Climate Control item. Retrieved from OECD (2014), *PISA 2012 Technical Report*, PISA, OECD Publishing, Paris, https://doi.org/10.1787/6341a959-en.

Response to the matching question is used to compute the final score (0: incorrect, 1: correct).
For each examinee, we have:

- Log data containing positions of the three control bars each time they clicked "Apply" & clicks of "Reset".
- Final score (0/1) on the matching.

Example 2: eNAEP 8th Grade Math Question

Process data can also be collected with digital assessment platforms:

NAEP 2017 Mathematics item. Retrieved from the NAEP Question Tool. https://www.nationsreportcard.gov/nqt/

Log data can include:

- Keystrokes for the constructed response
- Tool usage: e.g., scratchwork draw/erase, text-to-speech, zooming, highlight
- Visits and revisits: indicated by entering/exiting the item page

What Does Process Data Look Like?

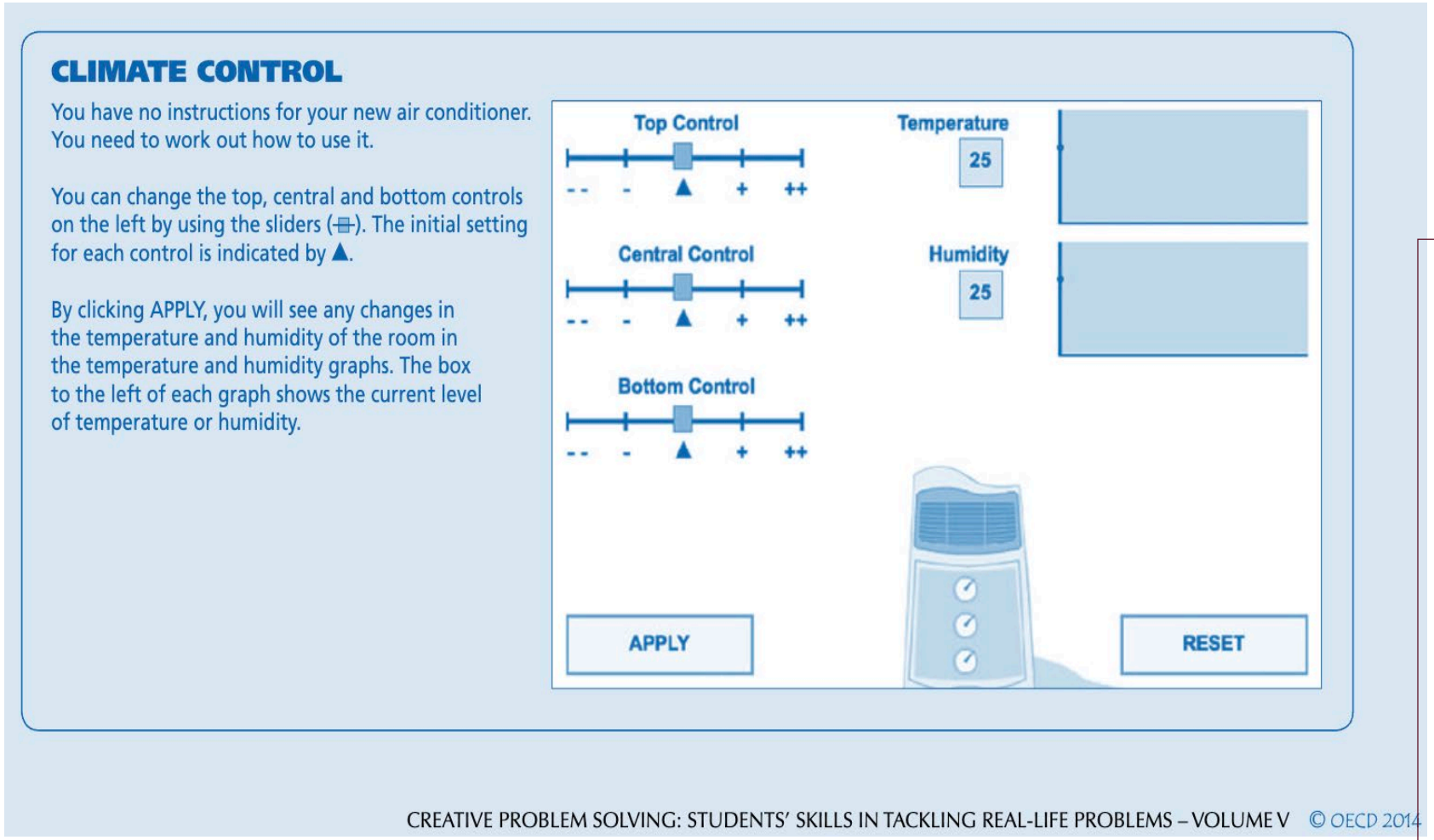
In this module, we restrict discussion to the sequence of actions performed by an examinee in pursuit of solving a problem.

- In many educational and assessment validation studies, process data can be collected from think-aloud protocols.
- Computerized assessments allow the streamlined collection of process data on a large scale, and they are commonly stored as timestamped log data, with the following components:

Examinee ID	Item ID	Event	Timestamp
1	1	Start	0.0
1	1	Click_CS	5.5
1	1	…	…

Let's see some examples.

Structure of Process Data: Example 1



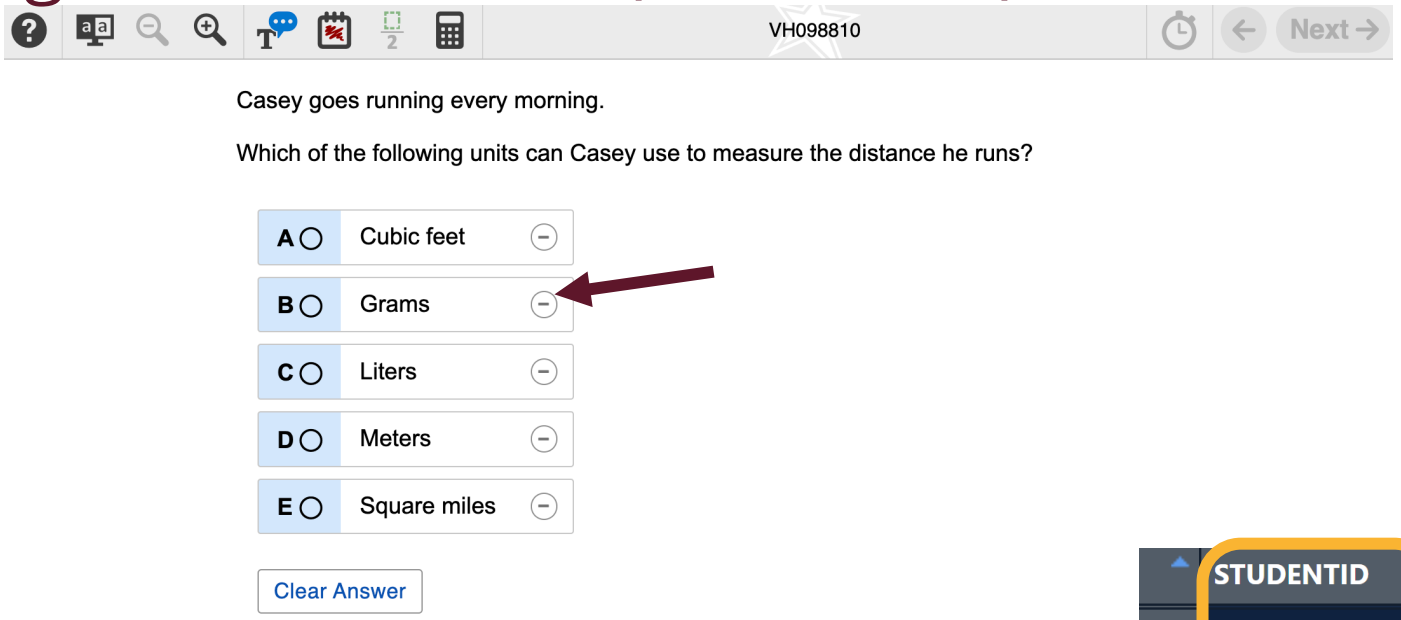
Action	Cumulative Elapsed Time
Start	0.0
1_0_0	25.1
…	…
Reset	42.5
1_1_1	56.9
end	62.2
Total Time	**62.2**
Final Score (matching question)	**1**

Setting top/middle/bottom control bars at +/Δ/Δ

PISA Climate Control Item. Retrieved from OECD (2014), *PISA 2012 Technical Report*, PISA, OECD Publishing, Paris, https://doi.org/10.1787/6341a959-en.

Structure of Process Data: Example 2

Here's a screenshot of one student's process data on one NAEP 8th grade math item (VH098810):



NAEP 2017 Mathematics item. Retrieved from the NAEP Question Tool. https://www.nationsreportcard.gov/nqt/

	STUDENTID	AccessionNumber	Observable	EventTime	Coding
432	2333000955	VH098810	Enter Item	2017-02-02 13:44:41	Enter_Item
433	2333000955	VH098810	Click Choice	2017-02-02 13:44:52	choose_4
434	2333000955	VH098810	Next	2017-02-02 13:44:57	Next
435	2333000955	VH098810	Exit Item	2017-02-02 13:44:57	Exit_Item
971	2333000955	VH098810	Click Progress Navigator	2017-02-02 14:09:01	Click_Prog
973	2333000955	VH098810	Enter Item	2017-02-02 14:09:01	Enter_Item
975	2333000955	VH098810	Exit Item	2017-02-02 14:09:03	Exit_Item

What might test-taking process tell us?

Process data may tell us **how examinees arrived at their final response**.

Examples of psychometric questions:

- Does process contain **additional construct-relevant information** unavailable from the final score alone? Can we use this additional information to reduce the **standard error of measurement**?
- Does process provide **validity** evidence that supports our **theory about how the item measures the construct?**
- Does process tell us **why** an item exhibits **differential item functioning** for specific groups?
- Do the test **design and accommodation** effectively help all examinees **demonstrate their true performance**?

Examples of educational questions:

- Can process reveal a student's mastery of specific **skills and misconceptions**?
- Do students with disabilities feature **specific types of wrong responses** or **test-taking strategies**?

Despite the potential utilities, process data can be messy and noisy.

Data Structure of Raw Action Sequences

- Raw action sequences (the "Action"/"Coding" column) are variable-length sequences of ordered categorical actions:

Variable length

Different examinees can differ in sequence length

E.g.,

- Examinee 1:

  Enter_Item, Choose_right, Choose_wrong, Choose_right, Exit_Item

- Examinee 2:

  Enter_Item, Choose_right, Exit_Item

Categorical

Each action is sampled from the set of all possible actions

E.g.,

Q1: {Enter_Item, Exit_Item, Choose_right, Choose_wrong, Clear_Answer}

Q2: {Enter_Item, Exit_Item, Part1_right, Part1_wrong, Part2_right, Part2_wrong, Clear_Answer}

Ordered

Temporal ordering of actions is substantively meaningful.

E.g.,

- Preview:

  Enter_Item, Exit_Item, Enter_Item, Choose_right, Exit_Item

- Review:

  Enter_Item, Choose_right, Exit_Item, **Enter_Item, Exit_Item**

- The next sections will discuss how to transform sequences to numerical features that are more straightforward to model.

A few publicly-available databases

Logfiles have been released for several large-scale assessments. Here are a few examples if you'd like to explore:

- Program for International Student Assessment (PISA) 2012 logfiles (OECD, n.d.):
  - PISA 2012 CBA Database. Examples of released items available in PISA 2012 technical report (OECD, 2014).
  - The ProcData R package (Tang et al., 2021) contains a preprocessed, simplified version of the Climate Control item.
- Program for International Assessment of Adult Competency (PIAAC) 2012 logfiles (OECD, n.d.):
  - Includes the interactive tasks for Problem Solving in Technology-Rich Environments (PSTRE). Apply for access to released items through the GESIS data archive.
- National Assessment of Educational Progress 2017 Mathematics process data (U.S. Department of Education, IES, & NCES, 2017):
  - Applying for a restricted-use data license
  - A small publicly available subset released for 2019 NAEP data mining competition.

Section 2: Extracting Process Data Features

2

2 Extracting Process Data Features

Section Learning Objectives

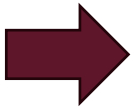
Learning Objective 1
Understand the rationale behind extracting numerical features from unstructured log data.

Learning Objective 2
Know the purpose and steps of extracting pattern-based summary indicators.

Learning Objective 3
Explain different types of data-driven process feature extraction methods.

- Raw sequence cannot be directly incorporated into many well-established statistical and measurement models, e.g.,
  - item response theory models,
  - structural equation models, or
  - hypothesis tests.
- Feature extraction methods are one class of process data analysis methods.
  - aim to transform action sequences into numerical, rectangular data that preserve original sequence information, so that, in subsequent analysis, we can work with the features instead.
  - Idea:

Event	Time
Start	0.0
Click_CS	2.9
Click_ObtNo	12.1
Button_ObtNo	16.3
Auth_No_Close	18.2
Email	21.4
Web	26.8
Back	28.1
Click_RF	33.3
Reason_Wrong	36.3
Combobox1	39.4
On_AuthBox	40.8
ASCII_7	41.7
ASCII_8	42.2
ASCII_3	43.1
ASCII_4	43.8
Off_AuthBox	44.9
Submit	50.6
Submit_Close	53.5



Examinee_ID	Feature_1	Feature_2	Feature_...
1	0	1.09452644	...
2	0	0.36402208	...
3	1	-0.06644631	...
4	1	-2.14067054	...
5	1	1.32162680	...

Types of Features to Extract from Process

- Goldhammer et al. (2021) discussed two levels of process features:
  - Low-level process feature: Small, interpretable unit of behavior derived from contextualized (i.e., task-specific) log event(s), e.g., “clicking RESET“.
  - Process indicator: Item-level summaries, e.g., time-related, count-related, sequence-related indicators. See He & Cui’s (2025) review for types of indicators.
- In this module, we will cover a few types of feature extraction methods:
  - Pattern-based summary indicators
  - n-grams features: Presence or weighted frequency of key subsequences of consecutive actions.
  - Data-driven whole-sequence-based features:
    - Multidimensional Scaling (MDS): Numerical features that preserve information about pairwise sequence dissimilarity.
    - Sequence autoencoders: Numerical features that best reconstruct original action sequences in a recurrent neural network model.

1. Pattern-based Summary Indicators

Substantively meaningful indicators (e.g., expert-rubric-based) that describe specific patterns in the log data. Examples:

Timing indicators	**Example definition (base on log data)**
Total time on page visit	Timestamp difference for entering and exiting the page.
Duration of specific states (e.g., reading, viewing, calculator/scratchwork usage)	Aggregated time (end minus start time) for each occurrence of a state.
Time from start to specific action.	Timestamp difference between specific action and start.
Counts indicators	
Total number of actions	Length of the action sequence.
Count of a specific pattern (e.g., action, answer submissions, visits to a page)	Frequency of occurrence of the pattern in the action sequence.
Sequence indicators	
Similarity to expert-derived best solution path	Longest common subsequence or Levenshtein distance compared with reference sequence (e.g., best path)
Use of a specific strategy	Presence of a specific sequence of (consecutive or non-consecutive) actions implied by the strategy.

1. Pattern-based Summary Indicators

Identifying test-taking pattern(s) of interest

What patterns may provide behavioral evidence on the examinee, task, or research question we care about?	Domain knowledge, theory, and prior literature may support examining these patterns.

Clarify the operational definition: How does the pattern manifest in the raw log data?

Example: number of item revisits	Operational definition: Number of ”Enter Item” -> “Exit Item” pairs minus 1.

Create function for feature extraction based on operational definition

Input: Examinee log file	Output: Feature value

2. n-gram Features (He & von Davier, 2016)

- "Minisequences" that disassemble a long action sequence into manageable short pieces of length n.
- Example sequence on a NAEP math question for one examinee:

  Enter_item, Scratchwork_Mode_On, Draw, Scratchwork_Mode_Off, choose_3, Exit_Item, Enter_Item, Exit_Item

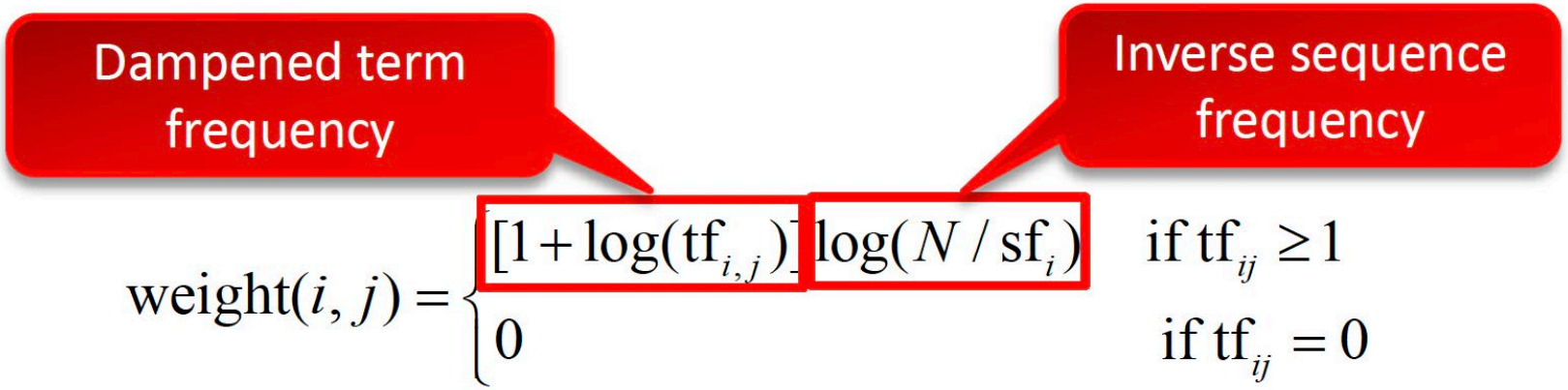
  - Unigram (n = 1): "*Enter_item*", "*Scratchwork_Mode_On*", "*Draw*", "*Scratchwork_Mode_Off*", "*choose_3*", "*Exit_Item*"
  - Bigram (n = 2): "*Enter_item, Scratchwork_Mode_On*", "*Scratchwork_Mode_On, Draw*", "*Draw, Scratchwork_Mode_Off*", "*Scratchwork_Mode_Off, choose_3*", "*choose_3, Exit_Item*", "*Exit_Item, Enter_Item*", "*Enter_Item, Exit_Item*"
  - Trigram (n = 3): "*Enter_item, Scratchwork_Mode_On, Draw*", "*Scratchwork_Mode_On, Draw, Scratchwork_Mode_Off,*", "*Draw, Scratchwork_Mode_Off, choose_3*", "*Scratchwork_Mode_Off, choose_3, Exit_Item*", "*choose_3, Exit_Item, Enter_Item*", "*Exit_Item, Enter_Item, Exit_Item*"
- Across examinees, there could be many more possible uni/bi/trigrams.
- For each possible uni/bi/trigram, we count its number of occurrences within the examinee's action sequence.

Representing n-grams as Numerical Features

1. Binary presence or absence of a particular n-gram
   - e.g., unigram of “Enter_item”: 1 (present) or 0 (absent)
2. Raw frequency
   - How many times it showed up in the sequence?
   - e.g., if the examinee entered the item twice, the frequency value of “Enter_item” is 2
3. TF-ISF weighting

n-gram features – TF-ISF weighting

We often use **TF-ISF-weighted** n-gram features: The TF-ISF value is given by:



- **Term frequency** (TF, $tf_{i,j}$): Frequency of n-gram *i* in examinee *j*'s sequence.
  - upweights behaviors that occur many times for this examinee.
- **Inverse sequence frequency** (ISF, N/sf_i): $1/(\%\ examinees\ with\ ngram\ i)$
  - downweights behaviors exhibited by most examinees and upweights rarer behaviors.

TF–ISF is high for n-grams that are

- frequent for a given examinee
- but not routine in the sample of examinees.

n-grams - Additional Remarks

- Unigrams do not capture sequential dependences (i.e., what happens right before/after an action).
  - Bigrams and trigrams can capture this sequential dependence.
- But bigrams/trigrams can occur much less often than unigrams, which can make higher-order n-grams less reliable.
  - In TF-ISF weighting, rare n-grams can receive very high weights because they are "unique," so filtering is especially important.
  - A common fix is to drop n-grams that occur fewer than ~5 times.

Not all actions/subsequences are important. How to find ones most relevant to what we care about?

Chi-square-based feature selection (He & von Davier, 2016):

It quantifies and tests the association between an outcome variable and an n-gram pattern. Taking score on task (correct/incorrect) as an example:

- For each n-gram, make 2x2 contingency table:

	Group C1 (e.g., correct)	Group C2 (e.g., incorrect)
n-gram present	n_1 = weighted count of this n-gram in C1 (sum of all TF-ISF values for correct examinees)	m_1 = weighted count of this n-gram in C2 (sum of all TF-ISF values for all incorrect examinees)
n-gram absent	Total TF-ISF in C1 − n_1	Total TF-ISF in C2 − m_1

- Compute the chi-square statistic:

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Large χ_c^2 suggests dependence between n-gram feature & outcome, i.e., people who got this item correct are more (or less) likely to present this n-gram compared to those who did not answer this item correctly.

Applications of n-grams

For test development:

- Identify the key actions/subsequences by correct/incorrect groups (He & von Davier, 2016; Liao et al., 2019)
- Identify key actions used by groups with different backgrounds (e.g., gender, race, SES) for equity and fairness checking (Han et al., 2019; Salles et al., 2020; Stadler et al., 2019)

For learning:

- Use n-grams for early prediction of task success/failure (e.g., Ulitzsch et al., 2023) to guide tailored support/interventions.

3. Data-Driven Whole Sequence Features

Aim is to transform action (or time) sequences in the sequence space into numerical vectors that preserve original sequence information in the K-dimensional vector space.

- Note that some whole-sequence features are created based on expert-defined rubrics. E.g., He, Borgonovi, & Paccagnella (2021) proposed two whole-sequence features that compare an observed sequence to an expert-defined reference sequence for optimal solution path, using on their longest common subsequence:
  - Similarity: How much does an individual's sequence deviates from a reference sequence?
  - Efficiency: How many additional actions does an individual undertake in excess of the number of actions contained in the reference sequence?
- Here we focus on data-driven features that are learned from the data. This section will introduce two methods:
  - Multidimensional Scaling (MDS; Tang et al., 2020)
  - Sequence Autoencoders (Seq2seq; Tang et al., 2021)

3.1 Multidimensional Scaling (MDS)

MDS (Tang et al., 2020) transforms variable-length action sequences ($\boldsymbol{s}_i$) to *K*-dimensional continuous features (M_i), by finding $M_1, \ldots, M_N$ that minimizes

$$\sum_{i=1}^{N} \sum_{i'=i+1}^{N} (d_{ii'} - \|\boldsymbol{M}_i - \boldsymbol{M}_{i'}\|)^2,$$

where $d_{ii'}$ is the dissimilarity between $\boldsymbol{s}_i$ and $\boldsymbol{s}_{i'}$ (for examinees i and i') based on an order-based sequence similarity metric (Gómez-Alonso & Valls, 2008).

- MDS features very accurately predicted scores on items.
- *K* can be chosen with cross-validation.
- Intuitively, MDS features aim to preserve pairwise individual differences in sequences.

Order-based Sequence Dissimilarity (Gómez-Alonso & Valls, 2008)

Sequence dissimilarity $d_{ii'}$ looks at the ordering differences on common actions $f(\boldsymbol{s_i}, \boldsymbol{s_{i'}})$ and number of unique actions $g(\boldsymbol{s_i}, \boldsymbol{s_{i'}})$:

$$d_{ii'} = \frac{f(\boldsymbol{s_i}, \boldsymbol{s_{i'}}) + g(\boldsymbol{s_i}, \boldsymbol{s_{i'}})}{L_i + L_{i'}},$$

$$f(\boldsymbol{s_i}, \boldsymbol{s_{i'}}) = \frac{\sum_{a \in C_{ii'}} \sum_{m=1}^{\min\{L_i^a, L_{i'}^a\}} |s_i^a(m) - s_{i'}^a(m)|}{\max\{L_i, L_{i'}\}},$$

$$g(\boldsymbol{s_i}, \boldsymbol{s_{i'}}) = \sum_{a \in U_{ii'}} L_i^a + \sum_{a \in U_{i'i}} L_{i'}^a .$$

Notations:

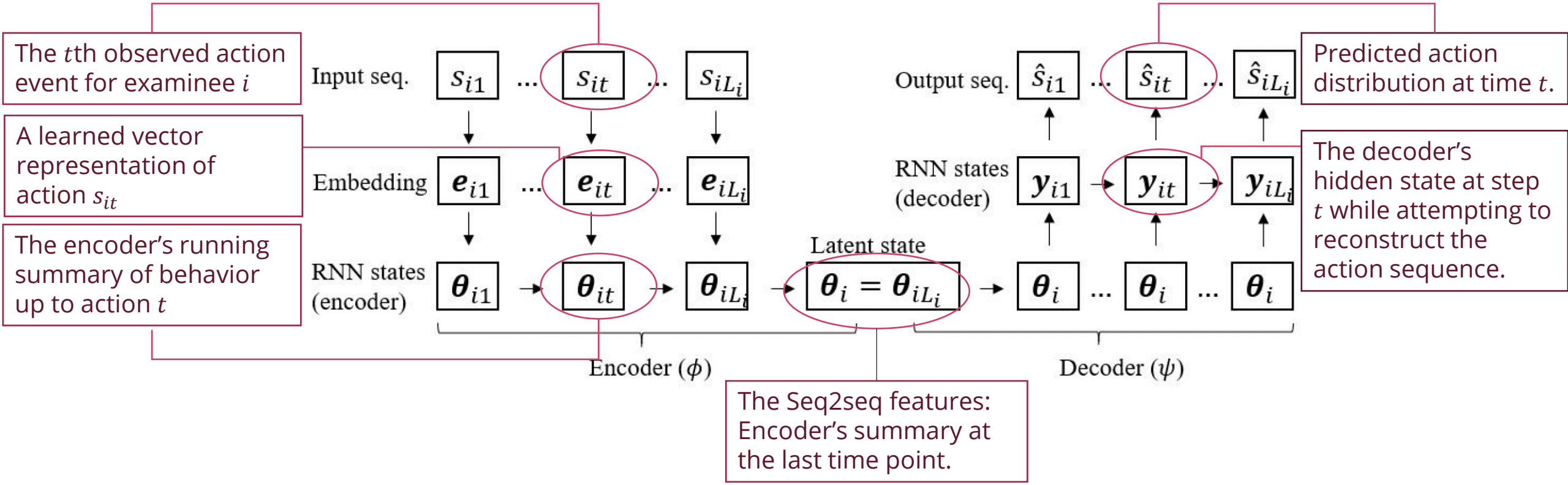
- $C_{ii'}$: the set of events shared by the two sequences
- $U_{ii'}$: the unique events of sequence i with respect to sequence i'
- L_i^a: the number of occurrences of event a in sequence $\boldsymbol{s_i}$
- $s_i^a(m)$: the position/time of the mth occurrence of event a in sequence $\boldsymbol{s_i}$.

MDS – Additional Remarks

- $d_{ii'}$ is small when:
  - Two examinees performed very **similar sets of actions**, and
  - The **ordering** of the common actions is also **similar**.
  - For two identical sequences' $d_{ii'} = 0$.
- We can use **other choices of pairwise dissimilarity** measures between sequences, e.g.,
  - Levenshtein (edit) distance (Hao, Zhu, von Davier, 2015): measures the difference between two strings by calculating the minimum number of single-character edits—insertions, deletions, or substitutions—needed to transform one string into the other
  - Dynamic time warping (He, Borgonovi, & Suárez-Álvarez, 2023): between two trajectory sequences (e.g., timing, page visit ordering), how much stretching/compressing of one sequence is needed to match the other
  - Custom definition – what types of individual differences are relevant for your application?
- **Choosing *K*** with cross-validation: Which K achieves best **out-of-sample prediction** of sequence pairwise dissimilarity?

3.2 Sequence-to-Sequence Autoencoders (Seq2seq)

- Similar to sentence embeddings in language models, we can use high-dimensional numerical vectors to represent action sequences (Tang et al., 2021).
- Neural network-based sequence model (also commonly used for neural language modeling), which extracts features that can be used to reconstruct original action sequences.



Seq2seq – Additional Remarks

- Parameters are estimated to minimize reconstruction loss of examinee's observed sequence:

$$L(\boldsymbol{s}_i, \hat{\boldsymbol{s}}_{\boldsymbol{i}}) = \frac{1}{L_i} \sum_{t=1}^{L_i} \sum_{j=1}^{J} s_{itj} \log(\hat{s}_{itj})$$

- $\boldsymbol{\theta_i}$ (encoder last state output) is the K-dimensional features representing the input sequence $\boldsymbol{s}_i$.

Remarks

1. Choice of sequence model is flexible (e.g., LSTM, GRU, transformer)
2. Choosing K (dimension of $\boldsymbol{\theta_i}$) based on cross-validation: Which K achieves the lowest out-of-sample sequence reconstruction error?

Note on Interpreting and Using Data-Driven Features

Unlike pattern-based indicators or n-grams, data-driven features are often **dense and lack inherent interpretability.**

- e.g., Toy example in Zhang et al. (2024) based on action sequences from a Problem Solving in Technology-Rich Environments (PSTRE) item from PIAAC 2012. For $K = 5$, these were the features of 3 examinees:
  - Examinee 1: "Click_W4, Toolbar_Web_Back, Response_Open, Response_4, Response_Close"
  - Examinee 2: "Click_W2"
  - Examinee 3: "Click_W1, Toolbar_Web_Back, Click_W2"

Table 1. Action sequence features on U06b for the three examinees, extracted using MDS (left) and with Seq2seq (right), both with $K = 5$ dimensions

	MDS			Seq2seq		
K	Examinee 1	Examinee 2	Examinee 3	Examinee 1	Examinee 2	Examinee 3
1	0.04	0.16	−0.04	−0.56	0.09	−0.62
2	0.15	−0.27	−0.15	0.90	0.08	0.89
3	0.20	−0.07	−0.04	−1.00	−0.90	−1.00
4	−0.07	−0.09	0.13	0.94	0.62	0.94
5	−0.01	−0.17	−0.21	−0.72	−0.36	−0.73

Note. *k* = dimension (1–5) of the MDS/Seq2seq features. The ranges of the features differed across dimensions and for MDS and Seq2Seq. For instance, across all 3,645 examinees, dimension 1 of the Seq2seq features ranged between −.92 and .27, and dimension 2 ranged between −.18 and .97.

Examinee 3's 5-dimensional Seq2seq features. Rows are the 5 dimensions.
These features are trained to preserve original sequence information, but they are not directly interpretable.

- Additional steps are often needed to use and interpret these data-driven features.
  - Some examples will be discussed in Section 4.

Wrap-up and Preview of Hands-on Examples

- In this section, we saw several types of features derived from action sequences:
  - Pattern-based summary indicators
  - n-grams features
  - Data-driven whole-sequence-based features:
    - Multidimensional Scaling (MDS)
    - Sequence autoencoders (Seq2seq)
- In the next section, we will illustrate feature extraction from PISA 2012 Climate Control item log data using these methods.

Aside from feature extraction methods, there are many other approaches to the analysis and modeling of process data that this tutorial does not cover.

Here is a very incomplete list of other methods to handle the unstructured sequences:

- **Methods that directly model the action sequences, e.g.,:**
  - Hidden Markov models (e.g., Xiao et al., 2021; Tang, 2024)
  - Point process or Gaussian process models (e.g., Chen, 2020; Chen & Zhang, 2020)
  - Recurrent neural networks (Wang et al., 2023)
- **Analyses based on sequence and trajectory (dis)similarity, e.g.:**
  - Sequence similarity and efficiency compared to expert-derived problem-solving steps (He et al., 2021)
  - Clustering based on pairwise sequence similarity or navigation trajectory similarity (Ulitzsch et al., 2021; He et al., 2023)
- **Network analysis of action relationships** (e.g., Zhu et al., 2016)

Section 3: Hands-on Coding Exercise: Data Wrangling and Feature Extraction

3

3 Hands-on Exercise

Section Learning Objectives

Learning Objective 1
Import and wrangle process data in R

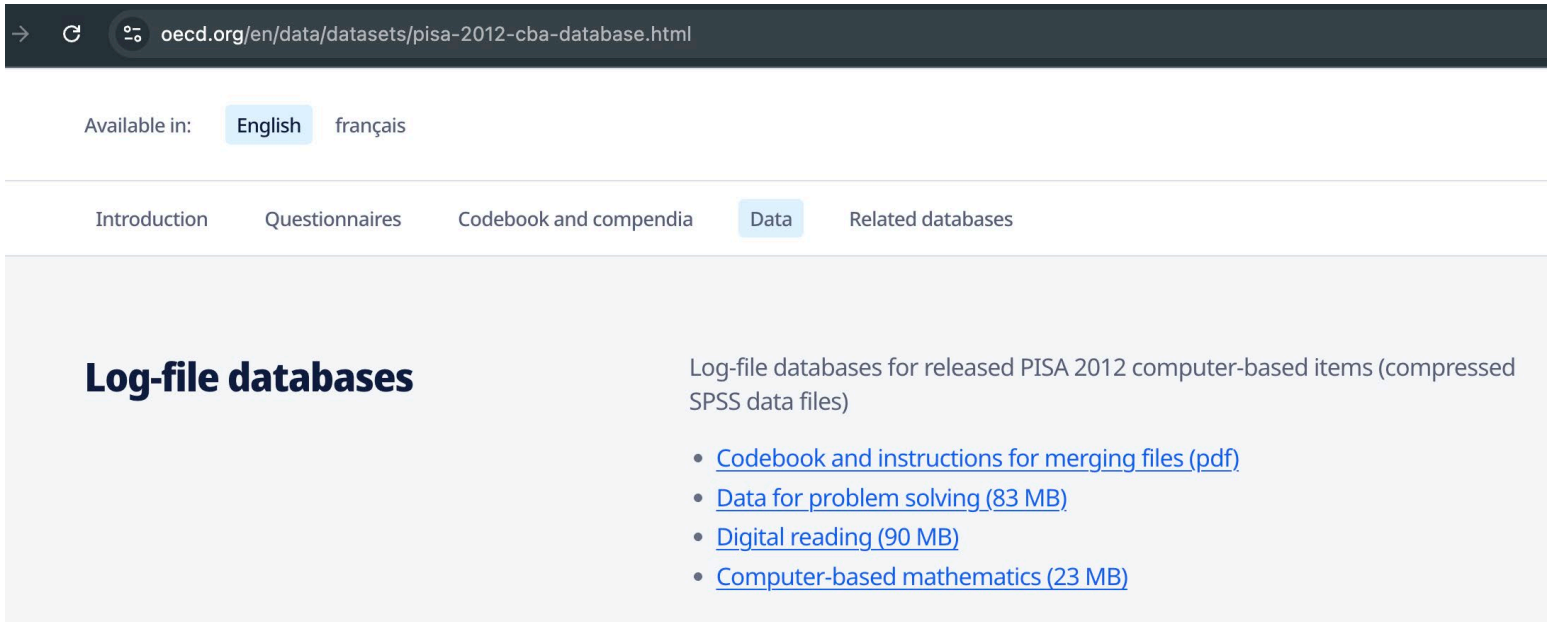
Learning Objective 2
Implement expert-derived feature extraction with PISA Climate Control item process data.

Learning Objective 3
Apply basic data-driven feature extraction using R ProcData package.

Learning Objective 4
Explore the structure, distribution, and interpretations (if applicable) of extracted features.

PISA 2012 Climate Control Process Data

- The original logfiles are available from the PISA 2012 CBA database.



Screenshot retrieved from the PISA 2012 CBA database. https://www.oecd.org/en/data/datasets/pisa-2012-cba-database.html

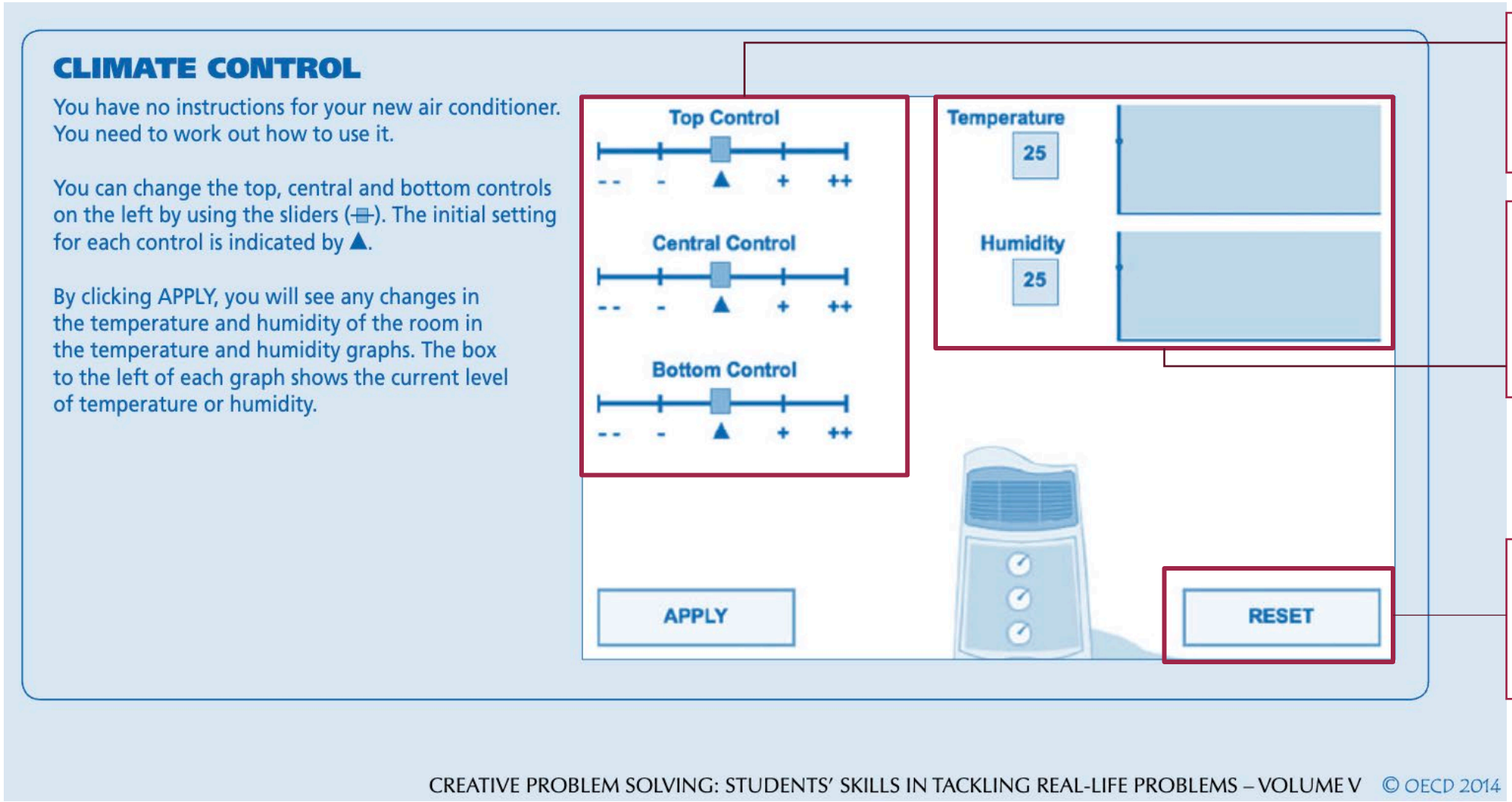
- We'll take a look at the raw data and go over common **preprocessing steps**.
- For feature extraction, we work with **pre-processed CC log data** from the ProcData R package.
- Complete R code available in EMIP_Process.Rmd.

1. Understanding task and data structure

2. Checking anomalies and preprocessing

3. Feature extraction and analysis

Recall: PISA Climate Control (CC) Item



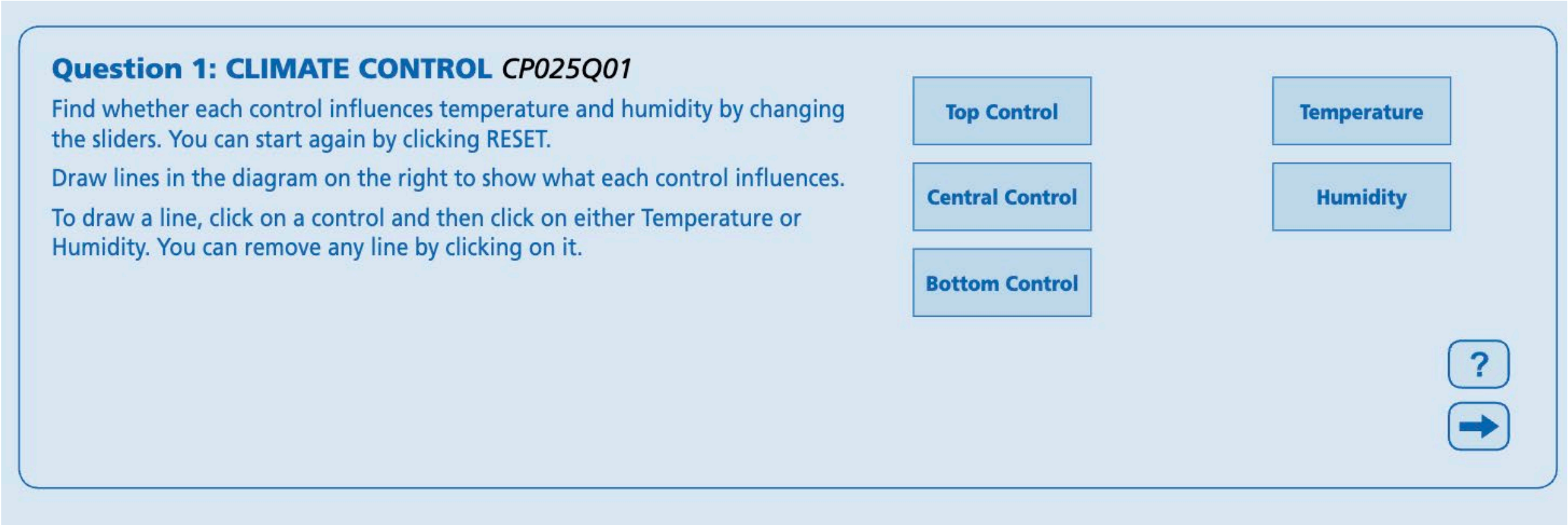
Students can move the three control bars from neutral (Δ) to 2 + /− positions.

After clicking "APPLY", these line plots add two new data points for temperature and humidity.

Clicking "RESET" will reset the three control bars to the neural (Δ) position.

PISA 2012 Climate Control item. Retrieved from OECD (2014), *PISA 2012 Technical Report*, PISA, OECD Publishing, Paris, https://doi.org/10.1787/6341a959-en.

CC Item – Final Score



PISA Climate Control item. Retrieved from OECD (2014), *PISA 2012 Technical Report*, PISA, OECD Publishing, Paris, https://doi.org/10.1787/6341a959-en.

Response to the matching question is used to compute the final score (0: incorrect, 1: correct).
For each examinee, we have:

- Log data containing positions of the three control bars each time they clicked "Apply" & clicks of "Reset".
- Final score (0/1) on the matching.

Importing Raw Data

Import Statistical Data

File/URL:

~/Downloads/CPRO_logdata_released/CBA_cp025q01_logs12_SPSS.sav

Browse...

Data Preview:

cnt Country code 3-character	**schoolid** School ID 7-digit (region ID + stratum ID + 3-digit)	**StIDStd** Student ID	**event**	**time**	**event_number**	**event_type**	**top_setting**	**central_setting**	**bottom_s**
ARE	0000189	04852	START_ITEM	1288.1	1	NULL	NULL	NULL	NULL
ARE	0000189	04852	ACER_EVENT	1291.9	2	reset	0	0	0
ARE	0000189	04852	ACER_EVENT	1338.4	3	apply	1	1	1
ARE	0000189	04852	ACER_EVENT	1346.8	4	apply	1	1	2
ARE	0000189	04852	ACER_EVENT	1350.1	5	apply	1	2	2
ARE	0000189	04852	ACER_EVENT	1354.5	6	apply	2	2	2
ARE	0000189	04852	ACER_EVENT	1361.1	7	apply	2	1	1
ARE	0000189	04852	ACER_EVENT	1361.1	8	reset	0	0	0
ARE	0000189	04852	ACER_EVENT	1375.3	9	Diagram	NULL	NULL	NULL
ARE	0000189	04852	ACER_EVENT	1376.2	10	Diagram	NULL	NULL	NULL
ARE	0000189	04852	ACER_EVENT	1400.1	11	Diagram	NULL	NULL	NULL

Previewing first 50 entries.

Import Options:

Name: CBA_cp025q01_logs12_SPSS

Model: Browse...

Format: SAV ☑ Open Data Viewer

Code Preview:

```
library(haven)
CBA_cp025q01_logs12_SPSS <- read_sav("Downloads/CPRO_logdata_released/CBA_cp025q01_logs12_SPSS
.sav")
View(CBA_cp025q01_logs12_SPSS)
```

Reading data using haven

Import Cancel

Variables in the Raw Data

	cnt Country code 3-character	schoolid School ID 7-digit (region ID + stratum ID + 3-digit)	StIDStd Student ID	event	time	event_number	event_type
1	ARE	0000189	04852	START_ITEM	1288.1	1	NULL
2	ARE	0000189	04852	ACER_EVENT	1291.9	2	reset
3	ARE	0000189	04852	ACER_EVENT	1338.4	3	apply
4	ARE	0000189	04852	ACER_EVENT	1346.8	4	apply
5	ARE	0000189	04852	ACER_EVENT	1350.1	5	apply
6	ARE	0000189	04852	ACER_EVENT	1354.5	6	apply

	event_type	top_setting	central_setting	bottom_setting	temp_value	humid_value	diag_state
1	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	reset	0	0	0	25	25	NULL
3	apply	1	1	1	27	28	NULL
4	apply	1	1	2	29	33	NULL
5	apply	1	2	2	31	36	NULL
6	apply	2	2	2	35	36	NULL
7	apply	2	1	1	36	36	NULL
8	reset	0	0	0	25	25	NULL
9	Diagram	NULL	NULL	NULL	NULL	NULL	'000000
10	Diagram	NULL	NULL	NULL	NULL	NULL	'000000
11	Diagram	NULL	NULL	NULL	NULL	NULL	'000000
12	Diagram	NULL	NULL	NULL	NULL	NULL	'000001

Remarks:

- Each student's id will be country (cnt) + school (schoolid) + student ID (StIDStd).
- There's no single "Coding"/"Event" column for a specific action, so we need to create it.
- Rows with event_type "Diagram" correspond to the matching question. We'll remove them if we only care about actions related to manipulating the control bars.

Group logfiles based on examinee IDs

```{r}
library(tidyverse)
CBA_cp025q01_logs12_SPSS %>%
  filter(
    cnt!='' & schoolid!='' & StIDStd!=''   # filter out data with missing ids
  ) %>% mutate(
  StudentID = paste0(cnt, schoolid, StIDStd, sep = '') # create a single id column "StudentID"
  ) -> log_cc

View(log_cc) # see the last column "StudentID"

log_cc %>% group_by(StudentID) -> log_cc # group the log data by Student ID
```

A tibble: 918,030 × 14 | Groups: StudentID [31,677]

We have 31,677 students' log data

cnt <chr>	schoolid <chr>	StIDStd <chr>	event <chr>	time <dbl>	event_number <dbl>	event_type <chr>	top_setting <chr>	central_setting <chr>
ARE	0000189	04852	START_ITEM	1288.1	1	NULL	NULL	NULL
ARE	0000189	04852	ACER_EVENT	1291.9	2	reset	0	0
ARE	0000189	04852	ACER_EVENT	1338.4	3	apply	1	1
ARE	0000189	04852	ACER_EVENT	1346.8	4	apply	1	1
ARE	0000189	04852	ACER_EVENT	1350.1	5	apply	1	2
ARE	0000189	04852	ACER_EVENT	1354.5	6	apply	2	2

2. Checking Anomalies

- Log data in assessments can be susceptible to data recording anomalies, so it is necessary to check for anomalies before analysis.
- Common anomalies:
  - Process information completely missing or recorded as NAs
  - Missing several or all timestamps (not necessarily a problem if there's event ordering ID and timing is not used for analysis.)
  - Incorrect log data structure, e.g.,
    - Events that should appear in all log data are missing (e.g., no "start", no "end", a page visit with "Enter" doesn't have a corresponding "Exit")
    - Mispositioned events (e.g., more events occur after "end")
  - Repeated entry of the same event.
  - …
- These will greatly help in understanding data structure and identifying anomalies:
  - Visually examine the log data of a small number of test takers.
  - Exploratory analysis: e.g., table of possible values

In our log data, there were no missing, but 118 students had anomalies in the number of or positions of start and end. We will remove their log data.

```
# Missing/multiple start/end, or events before start/after end
log_cc %>% arrange(time) %>%      # make sure events are ordered by timestamp
  summarize(
    n_start = sum(event == 'START_ITEM'),
    n_end = sum(event == 'END_ITEM'),
    first_event = event[1],
    last_event = last(event)
  ) -> tmp2
tmp2 %>% filter(n_start!=1 | n_end!=1 | first_event!='START_ITEM' | last_event!='END_ITEM') # :(
tmp2 %>% filter(n_start!=1 | n_end!=1 | first_event!='START_ITEM' | last_event!='END_ITEM') -> delete_obs
# remove these students from the log data
log_cc %>% filter(!(StudentID %in% delete_obs$StudentID)) -> log_cc

```

A tibble: 118 × 5

StudentID <chr>	n_start <int>	n_end <int>	first_event <chr>	last_event <chr>
ARE000002300541	1	1	START_ITEM	ACER_EVENT
ARE000004201060	1	1	ACER_EVENT	END_ITEM
ARE000004901228	3	1	START_ITEM	START_ITEM
ARE000010502595	1	1	START_ITEM	ACER_EVENT
ARE000010802666	1	1	START_ITEM	ACER_EVENT
ARE000010802674	2	1	START_ITEM	END_ITEM
ARE000016204066	2	1	START_ITEM	START_ITEM
ARE000016504147	2	1	START_ITEM	START_ITEM

Recoding Actions

- The goal is to recode events into a single “Action” column with unique, substantively meaningful action labels.
- Ideally, after the recoding, every action is:
  - Interpretable: Do we understand what it means in an examinee’s process?
  - Specific: Does the label precisely document what the examinee did?
  - Nonredundant: Can we delete the action without losing useful information?
- For the CC item, the recoding is relatively simple:
  - “start”, “end”, and “reset” remain.
  - For “apply”, recode into “x_y_z” containing top/central/bottom control values.
  - For all actions with event_type “Diagram” (matching actions), delete.
- Finally, we order actions by timestamps for each examinee.

```
Recoding events into unique, meaningful action labels

- START/END_ITEM & reset stay the same
- For apply, code the event based on the setting of the three controls
- For diagram, mark it as "DELETE" which will be filtered out in the final log data

```{r}
log_cc %>% arrange(time,.by_group = T) %>%
  mutate(
    Coding = case_when(
      # start/end
      event == "START_ITEM" ~ 'start',
      event == 'END_ITEM' ~ 'end',
      # reset
      event_type == 'reset' ~ 'reset',
      # apply
      event_type == 'apply' ~ paste(top_setting, central_setting, bottom_setting, sep = '_'),
      # diagram
      event_type == 'Diagram' ~ 'DELETE'
    )
  ) ->log_cc

log_cc %>% filter(
  Coding !='DELETE'          # remove rows of actions with "DELETE"
  ) %>%
  select(
    StudentID,  Coding, time # keep only id, coding, timestamp columns
  ) -> log_cc_recoded

log_cc_recoded
```
```

Resulting log_cc_recoded:

	StudentID	Coding	time
1	ARE000000100006	start	934.1
2	ARE000000100006	reset	951.0
3	ARE000000100006	0_0_0	962.3
4	ARE000000100006	0_0_0	968.6
5	ARE000000100006	0_0_0	970.4
6	ARE000000100006	0_0_0	971.0
7	ARE000000100006	end	974.5
8	ARE000000100018	start	489.1
9	ARE000000100018	1_1_1	556.0
10	ARE000000100018	1_1_0	560.3
11	ARE000000100018	1_1_-1	564.0
12	ARE000000100018	1_1_2	566.1
13	ARE000000100018	0_0_0	572.1
14	ARE000000100018	reset	574.2
15	ARE000000100018	-1_1_-1	584.0
16	ARE000000100018	reset	608.0
17	ARE000000100018	reset	709.9

3. Feature Extraction

- We will use the ProcData R package (Tang et al., 2021), which contains tools for exploratory process data analysis. ProcData contains a cleaned version of CC item's log and score data:

```r
library(ProcData)
data('cc_data')
names(cc_data)
```

```
[1] "seqs"      "responses"
```

```r
head(cc_data$responses)
```

```
 ARE000000200039 ARE000000200051 ARE000000300079 ARE000000400093 ARE000000400117 ARE000000500126
               0               1               1               1               0               0
```

```r
cc_data$seqs
```

```
'proc' object of  16763  processes

First  5  processes:

ARE000000200039
      Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7 Step 8 Step 9 Step 10 Step 11 Step 12  Step 13  Step 14
Event start  0_0_0  1_2_-2 2_2_2  2_2_2  2_2_2  2_2_2  2_2_2  2_2_-2 2_2_-2  2_-2_-2 -2_-2_-2 -2_-2_-2 -2_-2_-2
Time    0.0   49.3   55.9   61.7   62.6   63.2   63.5   63.9   66.4   68.4    71.2    74.7     75.4     75.7
      Step 15  Step 16  Step 17 Step 18 Step 19 Step 20 Step 21 Step 22 Step 23 Step 24 Step 25 Step 26 Step 27
Event -2_-2_-2 -2_-2_-2 -2_-2_0 -2_-2_0 -2_-2_0 -2_0_1  -2_0_1  -2_0_1  -2_0_1  -2_0_1  0_0_1   0_0_1   0_0_1
Time   76.0     76.2     79.5    80.5    80.9    83.6    84.1    84.6    85.0    85.4    88.2    88.6    88.9
      Step 28 Step 29 Step 30 Step 31 Step 32 Step 33 Step 34 Step 35 Step 36
Event 0_0_1   0_0_1   0_0_1   0_0_1   0_0_1   0_0_1   0_0_1   0_0_1   end
Time   89.2    89.4    91.1    91.3    91.5    91.7    91.8    92.0   100.5

ARE000000200051
      Step 1 Step 2 Step 3 Step 4  Step 5   Step 6 Step 7 Step 8 Step 9 Step 10 Step 11 Step 12 Step 13 Step 14
Event start  reset  -1_0_0 -1_-1_0 -1_-1_-1 -1_0_0 -1_0_0 reset  2_0_0  reset   0_2_0   reset   0_0_2   reset
Time    0.0   98.9  151.9  156.7   160.5    164.8  165.8  166.7  170.5  171.7   175.4   181.2   197.0   203.4
      Step 15 Step 16 Step 17 Step 18 Step 19 Step 20 Step 21
Event 0_1_0   reset   0_-1_0  reset   -1_0_0  reset   end
```

Note:

Score data (cc_data$responses) is available in the same PISA 2012 CBA database.

- "Cognitive item response data file".
- Can match to logfiles via the complete student ID.

Note on cc_data$seq

Creating the proc object from our previously preprocessed logfiles (log_cc_recoded) is easy.

Requires:

- action_seqs: list of action sequences for each examinee (each element is a vector, i.e., the Coding column for that examinee)
- time_seqs: list of timestamp sequences for each examinee (i.e., the time column for that examinee)

  Note: if you don't have timestamps, can use 1, 2, ..., L_i.
- ids: vector of examinee IDs

```r
log_cc_recoded %>% group_map( # apply to each group, i.e., each student's logfile
  ~ .x$Coding  # extract the Coding column for each examinee (list of action seqs)
) -> action_list
log_cc_recoded %>% group_map(
  ~ .x$time    # extract the time column for each examinee (list of time seqs)
) -> timestamp_list
log_cc_recoded %>% group_keys() -> Student_IDs  # student IDs

# create the proc object:
log_cc_proc <- proc(
  action_seqs = action_list,
  time_seqs = timestamp_list,
  ids = Student_IDs$StudentID
)
log_cc_proc
```

```
'proc' object of  31559  processes

First  5  processes:

ARE000000100006
      Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7
Event start  reset  0_0_0  0_0_0  0_0_0  0_0_0  end
Time  934.1  951.0  962.3  968.6  970.4  971.0  974.5

ARE000000100018
      Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7 Step 8  Step 9 Step 10 Step 11
Event start  1_1_1  1_1_0  1_1_-1 1_1_2  0_0_0  reset  -1_1_-1 reset  reset   end
Time  489.1  556.0  560.3  564.0  566.1  572.1  574.2  584.0   608.0  709.9   714.2
```

3.1 Pattern-based Summary Indicators

Let's extract three indicators:

- Number of actions
- Time to first action
- Whether the log suggests usage of the varying one thing at a time (VOTAT) strategy:
  - Manipulating a single input variable (i.e., adjusting one control bar).
  - Keeping all other input variables fixed at their initial value (i.e., leaving the other sliders at the neutral (Δ) position).
  - Observing the resulting changes (or lack thereof) in the outcome variables (i.e., the temperature and humidity levels).

1. Number of actions
Operational definition: Length of the action sequence

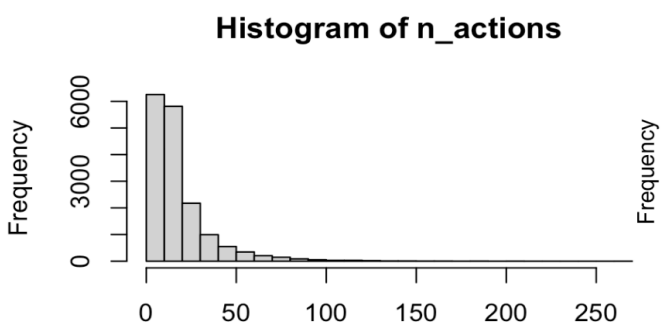
```{r}
n_actions <- sapply(seqs$action_seqs, function(x) length(x))
summary(n_actions)
```

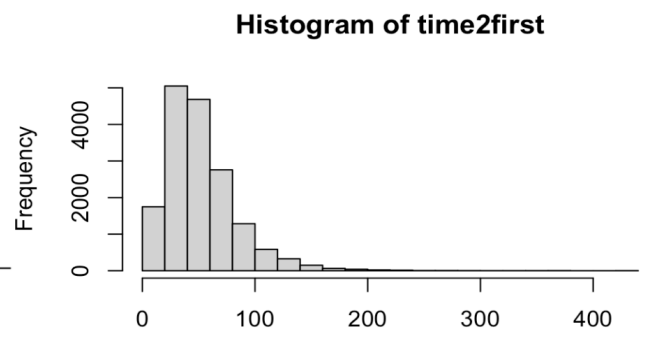
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    3.0     8.0    13.0    18.7    22.0   265.0 
```

2. Time to first action
Operational definition: Timestamp difference between 2nd action and "start"

```{r}
time2first <- sapply(seqs$time_seqs, function(x) x[2] - x[1])
summary(time2first)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00   30.70   46.20   52.51   66.20  429.70 
```





Continue to next page...

Regular expressions (regex) are useful in searching for string patterns:
https://evoldyn.gitlab.io/evomics-2018/ref-sheets/R_strings.pdf
Or just ask AI ☺

AI Overview

The regex for matching a pattern of "anything but 0" followed by "_0_0" in R is as follows:

Code

```
^[^0]+_0_0$
```

Explanation:

- `^`: Matches the beginning of the string.
- `[^0]+`: Matches one or more characters that are not a "0".
  - `[^0]` creates a character set that excludes "0".
  - `+` indicates one or more occurrences of the preceding character set.
- `_0_0`: Matches the literal string "_0_0".
- `$`: Matches the end of the string.

```
3. VOTAT strategy
Operational definition: Simultaneous presence of the following three actions in the sequence
- x_0_0: Nonzero top control only
- 0_x_0: Nonzero central control only
- 0_0_x: Nonzero bottom control only

Regular expressions are very useful for searching patterns:
https://evoldyn.gitlab.io/evomics-2018/ref-sheets/R_strings.pdf
(or ask AI :) )
```

```r
# top control
top <- lapply(seqs$action_seqs, str_detect, pattern = '^[^0]+_0_0$')
top <- sapply(top, function(x) sum(x)>0)
# central control
central <- lapply(seqs$action_seqs, str_detect, pattern = '^0_[^0]+_0$')
central <- sapply(central, function(x) sum(x)>0)
# bottom control
bottom <- lapply(seqs$action_seqs, str_detect, pattern = '^0_0_[^0]+$')
bottom <- sapply(bottom, function(x) sum(x)>0)
# votat: all three needs be true
votat <- top*central*bottom
table(votat)
```

```
votat
   0    1
7734 9029
```

Slightly more than half of the students used the VOTAT strategy.

3.2 n-gram Features

ProcData package contains a function for extracting n-gram features. Can specify:

- level: max n, e.g., level = 2 means it will extract unigrams and bigrams. level = 3 will additionally add trigrams.
- Type: binary, raw frequency, or TF-ISF weighted?

Note: for large n, the number of n-grams and computation time explode combinatorially.

```
## ngrams features

```{r}
ngrams <- seq2feature_ngram(seqs, level = 2, # uni & bigrams
                            type = 'weighted') # tf-idf weighted
dim(ngrams) # 5651 unique uni/bigrams
# View(ngrams)
```

 [1] 16763 5651
```

seq2feature_ngram {ProcData} R Documentation

ngram feature extraction

Description

`seq2feature_ngram` extracts ngram features from response processes.

Usage

```
seq2feature_ngram(seqs, level = 2, type = "binary", sep = "\t")
```

Arguments

`seqs`	an object of class "`proc`"
`level`	an integer specifying the max length of ngrams
`type`	a character string ("`binary`", "`freq`", or "`weighted`") specifying the type of ngram features.
`sep`	action seperator within ngram.

Details

Subset of unigrams:

start	0_0_0	1_2_-2	2_2_2	2_2_-2	2_-2_-2	-2_-2_-2	-2_-2_0	-2_0_1	0_0_1	end
0	0.8419037	4.859395	5.122478	6.151813	3.677196	7.204411	8.514172	13.313239	3.472410	0
0	0.0000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
0	0.0000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.021930	0
0	1.4254669	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.021930	0
0	0.0000000	0.000000	0.000000	0.000000	0.000000	2.760905	0.000000	0.000000	0.000000	0
0	0.0000000	0.000000	1.963(	0.000000	0.000000	10.237575	0.000000	0.000000	0.000000	0

Subset of bi-grams:

	-1_0_0 reset	reset 2_0_0	2_0_0 reset	reset 0_2_0	0_2_0 reset	reset 0_0_2	0_0_2 reset
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	5.320102	1.852951	1.738386	1.674951	1.608424	1.665127	1.930049
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	3.137319	2.943344	1.674951	1.608424	1.665127	3.267857
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

3.3 MDS features

ProcData package provide functions for extracting these data-driven whole-sequence features.

seq2feature_mds {ProcData} R Documentation

Feature extraction via multidimensional scaling

Description

`seq2feature_mds` extracts `K` features from response processes by multidimensional scaling.

Usage

```
seq2feature_mds(seqs = NULL, K = 2, method = "auto",
  dist_type = "oss_action", pca = TRUE, subset_size = 100,
  subset_method = "random", n_cand = 10, return_dist = FALSE,
  L_set = 1:3)
```

Arguments

`seqs`	a "`proc`" object or a square matrix. If a squared matrix is provided, it is treated as the dissimilary matrix of a group of response processes.
`K`	the number of features to be extracted.
`method`	a character string specifies the algorithm used for performing MDS. See 'Details'.
`dist_type`	a character string specifies the dissimilarity measure for two response processes. See 'Details'.
`pca`	logical. If `TRUE` (default), the principal components of the extracted features are returned.

Extracting K = 5 MDS features (and perform PCA to them):

````
199 ## MDS features
200
201 ```{r}
202 mds_fts <- seq2feature_mds(seqs, K = 5, pca = TRUE)
203 dim(mds_fts$theta)
204 ```

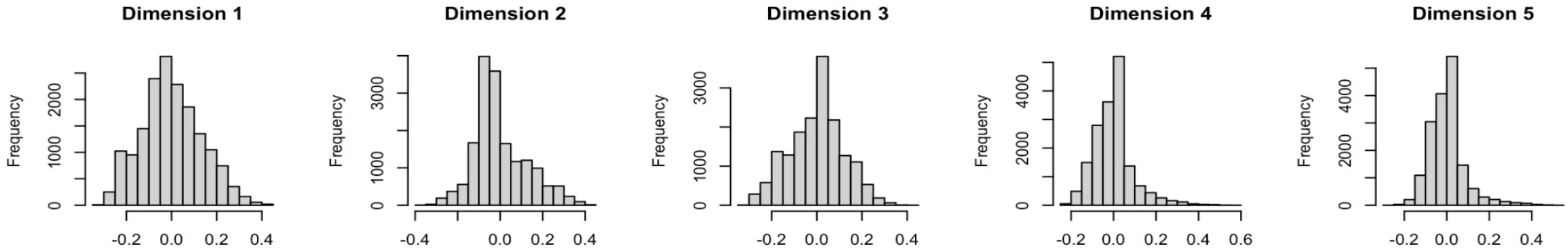
     [1] 16763     5
````

To choose K based on cross-validation? This will run MDS for K = 5, 10, and 15. It will return which one achieves lowest 5-fold cross-validation error predicting pairwise dissimilarities.

````
```{r}
chooseK_mds(seqs, K_cand = c(5,10,15))
```
````

First 8 examinees' 5-dimensional MDS features (MDS$theta):

	PC1	PC2	PC3	PC4	PC5
1	-0.030373762	0.0914882891	0.0196297784	0.0226296461	-0.0310482061
2	0.031237075	-0.1996343957	-0.0487124230	0.0525201560	0.1689125947
3	0.004326211	-0.0032278656	0.2093969243	-0.1151864631	-0.0055614476
4	0.027287183	-0.0559815020	0.0573065964	-0.0131803700	-0.0780509213
5	0.211345691	0.0090122524	-0.2852290075	0.0592053828	0.0348519415
6	-0.074442258	-0.0349178512	0.0060654210	0.0208831842	0.0085675388
7	-0.136750883	-0.0371176072	0.0246724758	0.0268497585	-0.0149831872
8	-0.125952204	0.0388762119	-0.0163756151	0.3451161041	-0.0385046327

Remarks:

- The principal components are uncorrelated but **not necessarily (almost never) independent**.
- These features are not inherently interpretable. Users take additional steps to interpret them, e.g.,
  - Interpret **clusters** based on MDS features
  - Examining **sequential pattern shifts** as feature value increases



3.4 Seq2seq Autoencoder Features

seq2feature_seq2seq {ProcData} R Documentation

Feature Extraction by autoencoder

Description

`seq2feature_seq2seq` extract features from response processes by autoencoder.

Usage

```
seq2feature_seq2seq(seqs, ae_type = "action", K, rnn_type = "lstm",
  n_epoch = 50, method = "last", step_size = 1e-04,
  optimizer_name = "adam", cumulative = FALSE, log = TRUE,
  weights = c(1, 0.5), samples_train, samples_valid,
  samples_test = NULL, pca = TRUE, verbose = TRUE,
  return_theta = TRUE)
```

Arguments

`seqs`	an object of class "`proc`".
`ae_type`	a string specifies the type of autoencoder. The autoencoder can be an action sequence autoencoder ("action"), a time sequence autoencoder ("time"), or an action-time sequence autoencoder ("both").
`K`	the number of features to be extracted.
`rnn_type`	the type of recurrent unit to be used for modeling response processes. "`lstm`" for the long-short term memory unit. "`gru`" for the gated recurrent unit.
`n_epoch`	the number of training epochs for the autoencoder.

Very similar to MDS feature extraction.

- For ae_type, it can be “action, “time”, or “both” (e.g., “both” will extract features that can recover both action and timestamp sequences.)
- rnn_type, n_epoch, step_size, etc. are settings for training the neural network.
- To choose K with cross validation, use chooseK_seq2seq()

Note:

- Recurrent neural nets are slow especially for longer and more complex (larger set of possible of actions) sequences.
- To run functions requiring neural nets in the ProcData package, user must install dependencies. See instructions here: https://cran.r-project.org/web/packages/ProcData/readme/README.html

4. Some Basic Usage of the Process Features

We can use the extracted features to answer some simple questions, e.g.,

Is the usage of the VOTAT strategy (0/1) related to the score (0/1) on the matching question?

```
score <- cc_data$responses
print(tab <- table(votat, score))
round(tab/rowSums(tab),2)
chisq.test(votat, score)
```

```
      score
votat    0    1
    0 6276 1458
    1 1358 7671
      score
votat    0    1
    0 0.81 0.19
    1 0.15 0.85

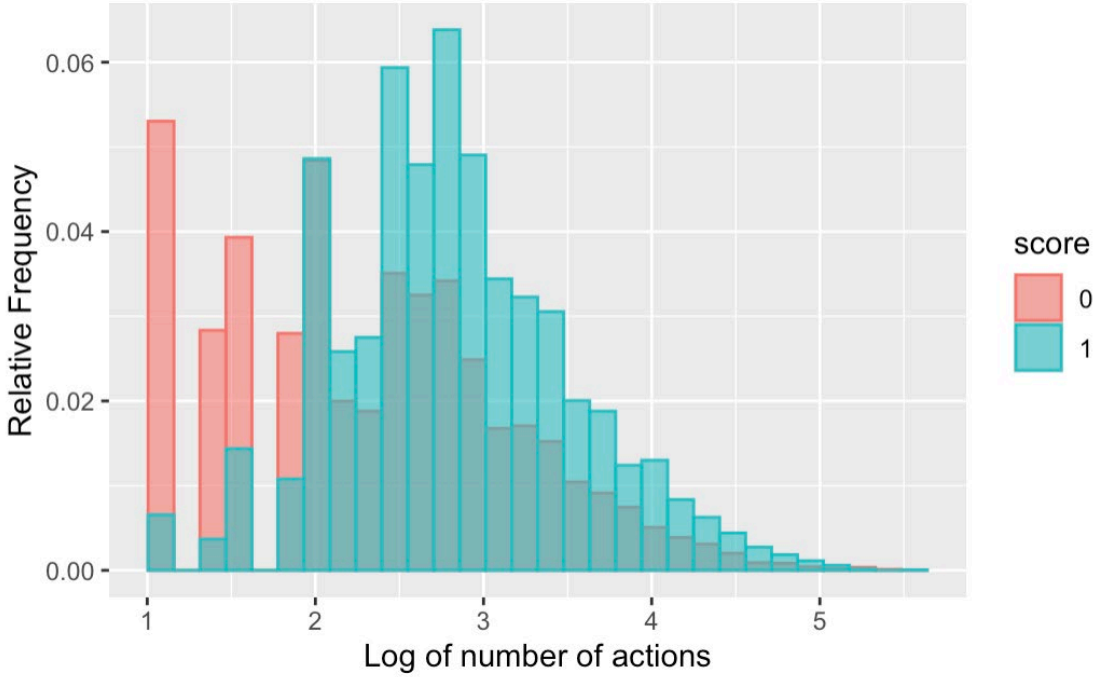
        Pearson's Chi-squared test with Yates'
continuity correction

data:  votat and score
X-squared = 7337.8, df = 1, p-value < 2.2e-16
```

19% of students who did not use VOTAT answered the question correctly, whereas 85% who used VOTAT answered correctly.

How does the distribution of number of actions differ for correct/incorrect responses?

```
plot_df <- data.frame(score = as.factor(score),
                      log_actions = log(n_actions))
plot_df %>% ggplot(aes(x = log_actions,
                       y = after_stat(count / sum(count)), # relative freq
                       fill = score, color = score)) +
  geom_histogram(position = 'identity',alpha = .6, bins =30) +
  xlab('Log of number of actions')+ ylab('Relative Frequency')
```



Section 4: Applications of Process Features in Measurement and Education

4

4 Applications of Process Features

Section Learning Objectives

Learning Objective 1

Describe case studies in which process-derived features are used to answer measurement questions.

Learning Objective 2

Know the key considerations for modeling and statistical analysis using process-derived features.

Case Studies Applying Process Features

Now that we have features extracted from process data, how do we use them in measurement tasks?

In this section, we show three case studies illustrating their usage:

- **Case study 1:**

  Examining group differences in process patterns (e.g., score/demographic) to understand the effect of NEAP extended time accommodation (Wei & Zhang, 2024)

- **Case study 2:**

  Using process features with IRT models to improve measurement precision of adult problem solving in technology-rich environments (Zhang et al., 2023)

- **Case study 3:**

  Using process features with IRT models to reduce and understand differential item functioning in assessing adult problem solving in technology-rich environments (Chen, Zhang, & Liu, 2025)

Case Study 1: Effect of NAEP Extended-time Accommodation

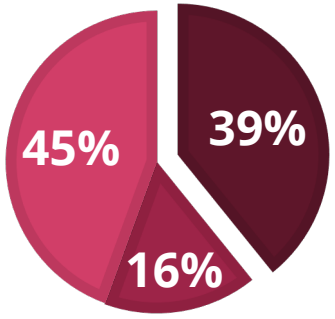
- The goal of **test accommodations** is to increase the reliability, validity, and fairness of the scores by reducing **construct-irrelevant measurement errors** (AERA, APA, & NCME, 2014).
- **Example: Extended time accommodation (ETA)**
  - NAEP's goal is to assess what students **know and can do** in mathematics
  - The standard time limit for a math test block is 30 minutes.
  - Students with **learning disabilities (LD)** are more impacted by **test-speededness**, e.g., higher anxiety, rapid guessing, and missingness due to **running out of time**.
  - **ETA** aims to help overcome the construct-irrelevant nuisance (lack of time) when speed is not an intended component of the test construct.
  - Students with ETA received **90 minutes** for a test block.
- **Do LD students who receive ETA benefit from it?**
  - NAEP provides a rare opportunity to investigate this special group with a large sample size.
  - **Process data** can capture students' **interactivity** with each question --- something speculated to depend on time pressure.

Data Description (Wei & Zhang, 2024)

Sample:
1530 students with LD who took the N03 (15-item) block in the NAEP 2017 8th Grade Math.

Note:
45% of the LD students were granted ETA but did not use more than 30 minutes (**NU**). We separated them from those who used >30 minutes (**U**) in further analyses.

- Not granted (NG)
- Granted & used (U)
- Granted not used (NU)



Instrument: 15 items (multiple choice, constructed response, drag and drop)

Item	Content	Score Range	% Full Score
1	Translate a percent to a fraction	0-1	64.27%
2	Complete a circle graph to represent the data	0-1	94.55%
3	Multiplication of two two-digit decimals	0-1	46.40%
4	Determine x and y intercept of a given line	0-2	47.68%
5	Compare measurement using unit conversions	0-2	59.21%
6	Extend a numerical pattern	0-1	44.19%
7	Calculate diameter of a circle from a given circumference	0-1	12.23%
8	Rotation of a triangle	0-1	36.89%
9	Create a proportion to find distance on a map	0-1	67.39%
10	Identify characteristics of lines	0-2	12.84%
11	Make & explain a conclusion about linear equations	0-2	14.72%
12	Identify figures that are composites of 2 given shapes	0-2	6.75%
13	Evaluate circle graph and bar graph to determine possible data sets	0-4	5.85%
14	Match box-plots to stem-and-leaf plots	0-2	19.99%
15	Write expression for polygon area using conjecture	0-2	9.35%

Key variables used for examining the effect of ETA:

- ***ETA*** *group membership* (NG, U, NU)
- ***Score*** *on each item* (0 – Full score)
- ***Process-derived*** *variables on each item*

Instead of working with raw process data, this study worked with **predefined features** that are conjectured to be associated with **interactivity**.

Sample sequence:

Enter_item, Scratchwork_Mode_On, Draw, Scratchwork_Mode_Off, choose_3, Exit_Item, Enter_Item, Exit_Item

List of process-derived features:

Feature	Value
Number of actions (# of recorded events)	8
Number of visits (page views > 3 seconds)	2
Digital drawing tool usage (0/1)	1
Text-to-speech usage (0/1)	0
Total response time (in seconds, sum across visits)	102.23

Note:

Number of actions and response time were highly skewed and log-transformed in further analysis.

Methodological Considerations

- ETA assignment is **not randomized**.
  - In the evaluation of the ETA effect, there is the confound of true **math skill/knowledge difference** without time pressure
  - Assuming there is minimal time pressure on the first 5 items, we test for **conditional independence between the ETA group and score/interactivity on items 6-15**, given a **comparable total score on items 1–5**.
- Process features for interactivity have **highly irregular distributions**.
  - e.g., some features (# of visits) are "one-inflated"
  - We adopt a **conditional permutation test for the conditional independence** to circumvent the need for distributional assumptions:

Based on **total score on first 5 items**, we divided individuals into 4 **strata (k):** 0 – 1, 2 – 3, 4 – 5, 6 – 7. Testing the effect of ETA controlling for performance on the first 5 items amounted to testing

$$H_0: E_F(X_i \mid S_i = k) = E_R(X_i \mid S_i = k), \qquad \forall k,$$
$$H_1: E_F(X_i \mid S_i = k) < E_R(X_i \mid S_i = k), \qquad \text{some } k.$$

Notations:

- X_i*:* Interactivity feature/score value of student *i*
- S_i*:* Initial score strata of student *i*

Note: The alternative is a one-sided hypothesis, where those receiving ETA (U/NU) are **higher** on expected score/interactivity.

Results: Interactivity of those who used granted ETA (U)

- Higher expected response time and number of actions across almost all of items 6 - 15.
- Higher number of visits, % of digital drawing tool usage, and % of text-to-speech usage on select items.
- Even for those who used the extra time, **the expected score was only higher on items 6, 9, 11, 13,** although interactivity was higher across all items.

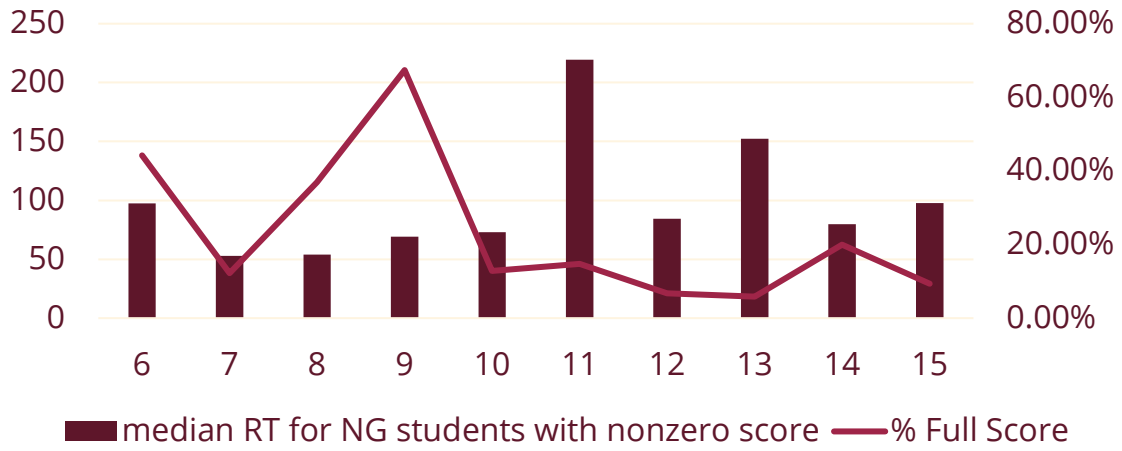
Item	Response Time	Number of Actions	Number of Visits	Digital Drawing Tool	Text-to-Speech	Final Score
6	0.43***	0.20***	0.19***	0.03*	0.10***	0.06*
7	0.55***	0.39***	0.28***	0.03	0.08**	-0.03
8	0.70***	0.19**	0.19***	0.03	0.05*	0.05
9	0.58***	0.24***	0.13***	0.04*	0.08**	0.07*
10	0.75***	0.19***	0.09	0.03*	0.13***	0
11	0.73***	0.35***	0.20***	0.06*	0.14***	0.06*
12	0.70***	0.34***	0.10**	0.10***	0.08**	0
13	0.87***	0.53***	0.21*	0.06***	0.10***	0.11*
14	1.04***	0.51***	0.34***	0.06***	0.11***	0.06
15	1.38***	0.73***	0.48***	0.11***	0.12***	0.01

**p<.05 ***p<.001*

Note: For NG vs NU group, differences in interactivity/score mostly insignificant.

How's ETA Affecting LD Students' Interactivity and Scores?

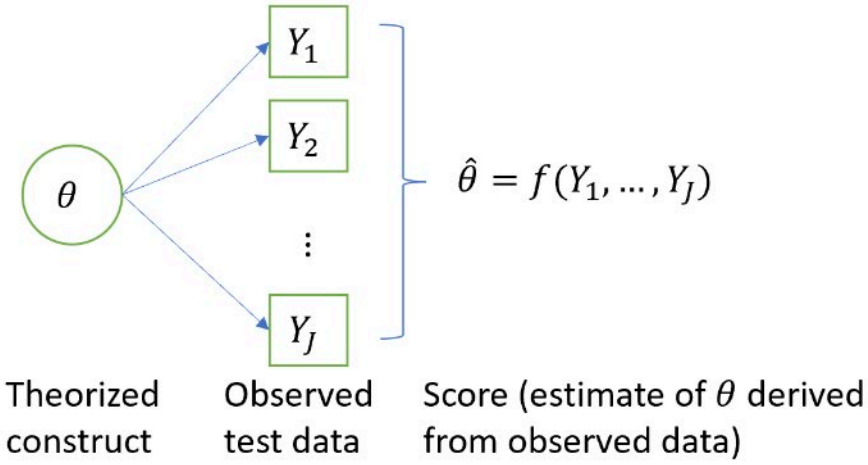
- **Many** students with LD granted ETA **did not use it** and did not do better.
- Given comparable performance on the first 5 questions, those who used the extra time **interacted more with later items than those without** ETA.
  - Spent more time
  - Had more logged actions
  - Used tools more often (scratchwork, text-to-speech)
  - Made more revisits to the same item
- If we assume that the performance on the first 5 items is an accurate reflection of what students know and can do in 8th-grade math, given the same proficiency, those given and used ETA could benefit from the extra time on certain items.
  - Speculation is that these are **items where they have the skills/knowledge to (partially) solve given sufficient time**:



- Items 6 and 9 had the **highest correct response rate**.
- Items 11 and 13 had the **highest median response time** for **individuals who received partial or full credit**.

Case Study 2: Improving Measure Precision of Adult Problem-Solving

- Suppose our aim is to estimate a latent construct (proficiency) θ from observed test behavior.
- In computer-based testing, we observe **final responses $\boldsymbol{Y}$** and also **process data $\boldsymbol{S}$** (action sequences, times, tool use).



- A test score $\hat{\theta}$ is an **estimator** of true θ, and a key goal in measurement is achieving **small estimation error** (high reliability/low mean-squared error: $E[(\hat{\theta} - \theta)^2]$).
- Reliable scoring using **responses only** ($\hat{\theta}_Y$ estimated from $\boldsymbol{Y}$) often requires long tests.

Can Process Data Reduce Test Length without Compromising Reliability?

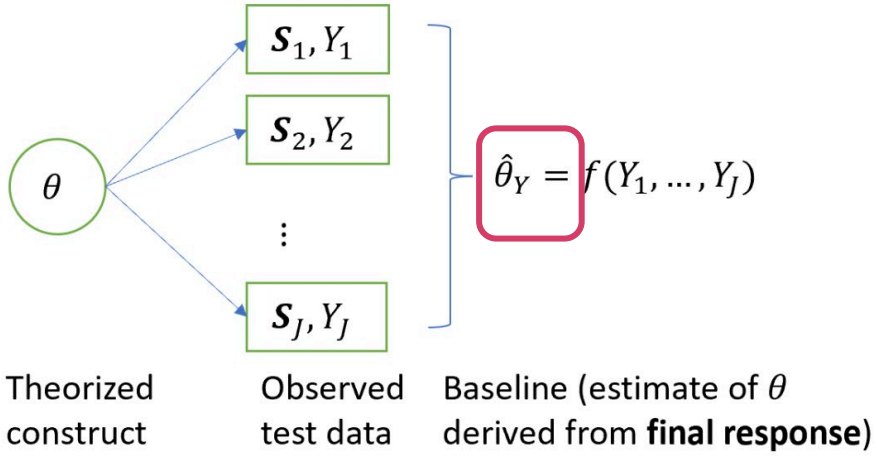
- Under Item Response Theory (e.g., generalized partial credit model for ordinal scores), ability is commonly estimated from responses via an estimator, e.g., Bayesian EAP that depends on the posterior expectation $\hat{\theta}_Y = E(\theta \mid \boldsymbol{Y})$.
- Reliability increases with more items:
  - Tradeoff: reliability vs. short test length/time.
- **Can process information add meaningful proficiency signal so we can shorten tests (use fewer items) without compromising measurement precision?**
  - Intuitively, the action sequences on item j (or MDS features extracted from them, denoted $\boldsymbol{X}_j$) may contain additional θ-relevant information compared to the final score, Y_j, e.g.:

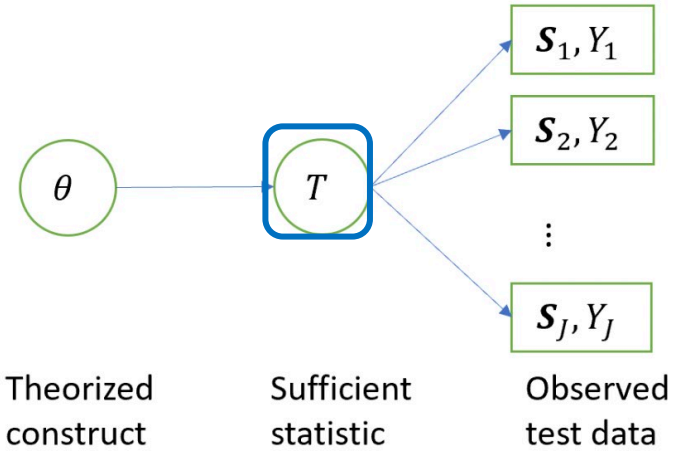
Partial mastery	Strategy choice
• Examinee has some idea on how to approach a problem, got stuck in the middle and gave up ⇒ final score is 0 • Examinee knows how to solve a numerical problem, made a careless mistake while entering a number on the calculator ⇒ final score is 0	• A spreadsheet question requires using "Search" to locate a particular row. Examinee doesn't know how to use "Search", but decided to eyeball the entire 500-row spreadsheet to find the requested info ⇒ final score is 1.

Improving $\hat{\theta}_Y$ with MDS process features

Rao-Blackwellization (Zhang et al., 2023):

Regressing the final response-based estimator ($\hat{\theta}_Y$), on a sufficient statistic (T) of θ derived from both response & process:





This gives a new process-incorporated estimator, $\hat{\theta}_X = E(\hat{\theta}_Y \mid T)$, with lower or equal MSE (comparable/higher precision).

To obtain T (intuition, see detail in paper):

- Extract MDS features from action sequences ($\boldsymbol{S}_j$) on each item j, X_j.
- For each item j, regressing the response-based score on MDS features.

Empirical Results from PIAAC PSTRE Data (Zhang et al., 2023)

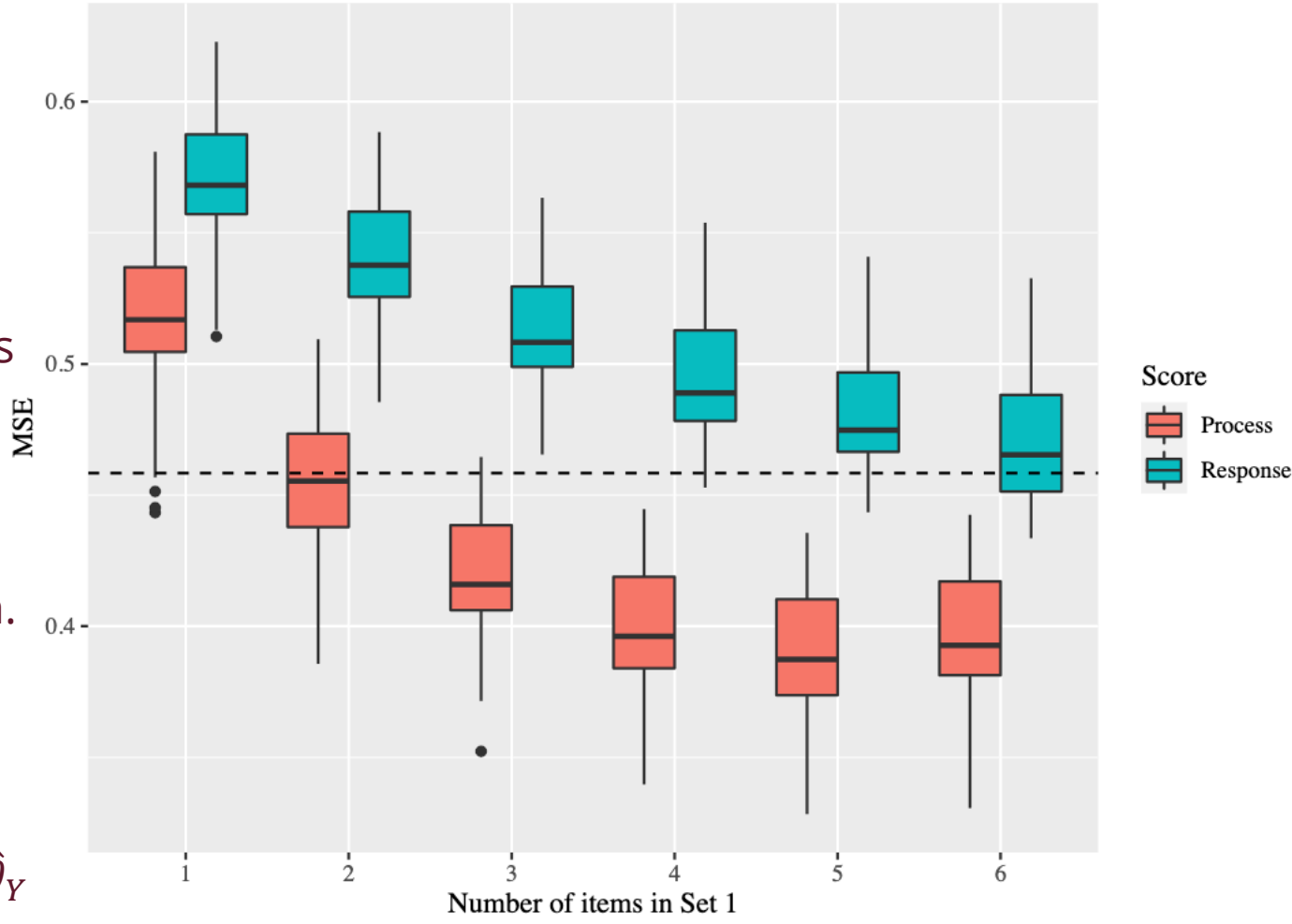
Dataset:

- OECD PIAAC adult skills assessment.
- Problem solving in technology-rich environments (PSTRE) assessment: 14 Interactive items resembling common ICT interfaces (email, spreadsheets, web browser).
- Sample: 2304 adults from 5 comparable countries and regions, who completed 14 PSTRE items.

Evaluation:

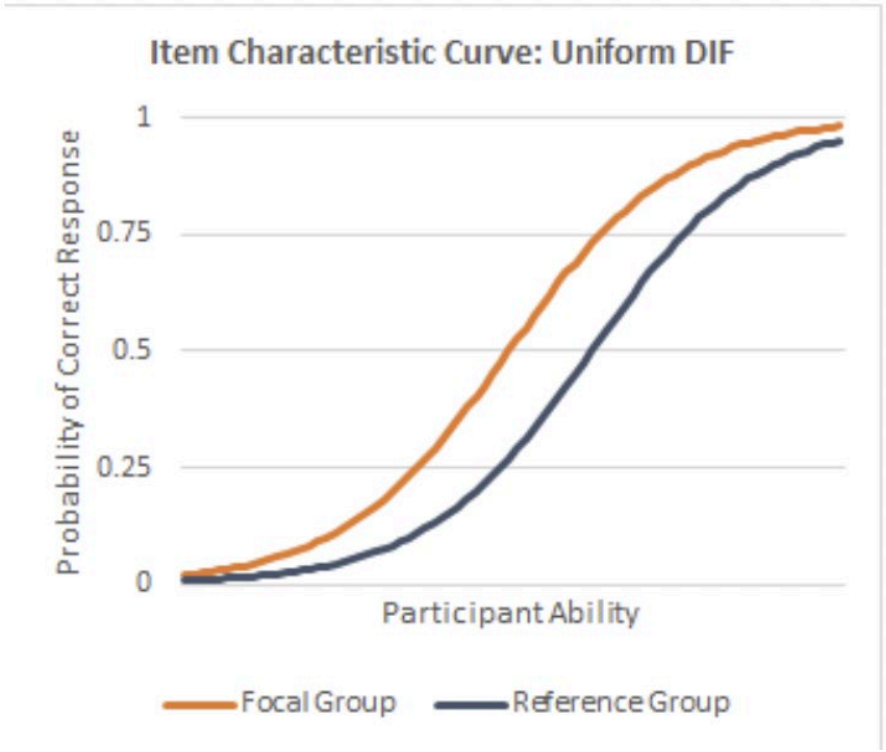
We randomly split the 14 items into 2 sets of 7 each.

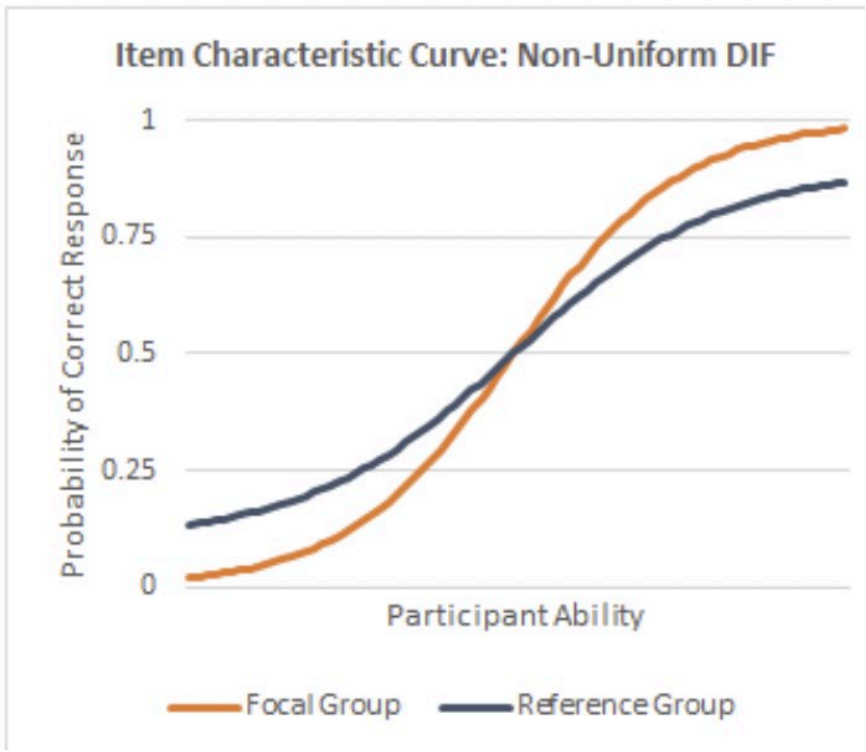
- Compute both $\hat{\theta}_Y$ (using final score only) and $\hat{\theta}_X$ (incorporating process information) on Set 1;
- Compute MSE with $\hat{\theta}_Y$ on Set 2.
- $\hat{\theta}_X$ based on 2 – 3 items achieved similar MSE as $\hat{\theta}_Y$ with all 7 items.



Case Study 3: Reducing and Understanding DIF in Adult Problem-Solving

Differential item functioning (DIF) occurs when an item behaves differently across groups for examinees with comparable proficiency.





DIF screening is routine in large-scale testing, but flagged DIF does not automatically reveal *why* the item is biased.

Can we use process to reduce/understand DIF?

- Final responses alone frequently provide insufficient evidence to diagnose the source of DIF.
- Standard DIF workflow ends with expert review, item revision, or discarding the item.
- Multidimensional IRT interpretation (Ackerman & Ma, 2024): DIF arises when the expected score depends on construct-irrelevant nuisance traits (η), whose distribution given θ differs for focal (F) and reference (R) groups, i.e., $f_F(\eta \mid \theta) \neq f_R(\eta \mid \theta)$.

  As a result, the expected score

  $$E_g(Y_j \mid \theta) \coloneqq \int E(Y_j \mid \theta, \eta) f_g(\eta \mid \theta) d\eta, \qquad g \in \{F, R\}$$

  differs across groups.
- Process data may expose how groups approached the task (e.g., interface use, strategy choices), giving a plausible path to:
  - Understanding what might be the nuisance (η) that drives DIF, and
  - Reducing DIF by jointly modeling (η, θ), rather than dropping the item.
- For an item j flagged with DIF, we estimate η using the K-dimensional MDS features extracted from item $j, \boldsymbol{X}_j \in \mathbb{R}^K$.

Reconstructing the Nuisance from Process (Chen, Zhang, & Liu, 2025)

On a DIF item j, assume response Y_{ij} follows a generalized linear model (GLM),

$$Y_{ij} \sim p(y \mid \mu_{ij}), \text{with } g(\mu_{ij}) = d + a_0\hat{\theta}_i + a_1\eta_{ij} + \lambda Z_i.$$

Notations:

- $\hat{\theta}_i$ is the target latent trait of examinee i estimated from DIF-free items;
- η_{ij} is a nuisance trait on item j of examinee i,
- Z_i is a binary grouping variable (F/R) or continuous demographic variable,
- $g(\cdot)$ is a link function (e.g., logit, identity).

If $\lambda = 0$, group differences are fully explained by η_{ij}, and the relationship between score and target trait will be group-invariant after controlling for the nuisance, eliminating the group difference in the conditional distribution of the response on measured traits (now θ and η).

Assume $\eta_{ij} = \boldsymbol{\omega}^{\top}\boldsymbol{X}_{ij}$, (i.e., nuisance is a weighted sum of MDS features).

To find the optimal weights, we solve

$$\hat{\boldsymbol{\omega}} = \text{argmin}_{||\boldsymbol{\omega}||=1}\left[\max_{d,a,\lambda} \ell(d,a,\lambda) - \max_{d,a,\lambda=0} \ell(d,a,\lambda)\right],$$

where $\ell()$ is the log-likelihood under the GLM.

This optimization finds a nuisance trait that minimizes the response distribution's dependence on group membership.

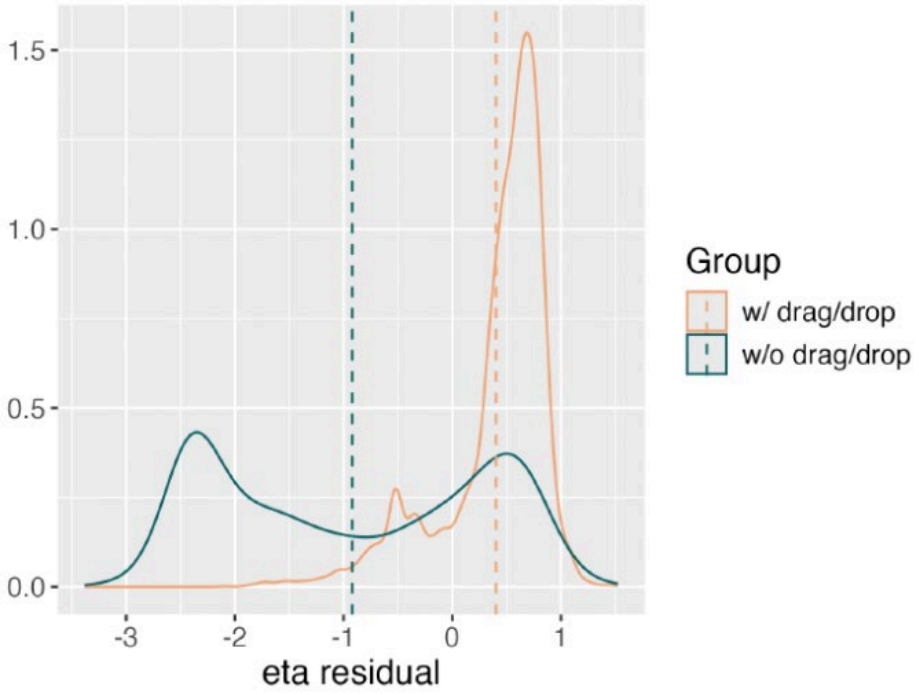
Using $\hat{\eta}_{ij}$ to Reduce and Interpret DIF

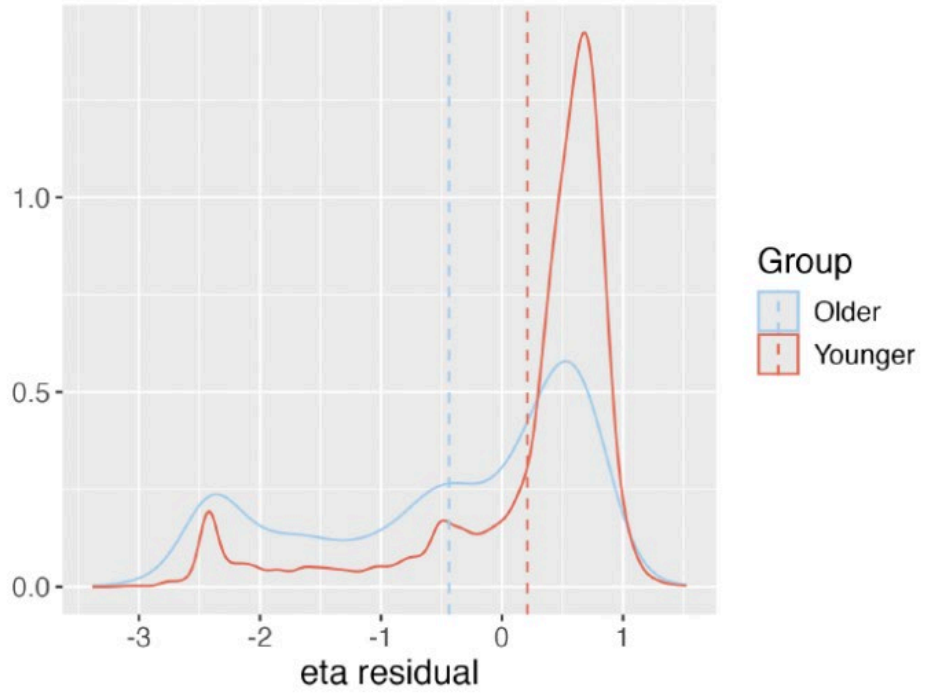
- Reducing DIF with $\hat{\eta}_{ij}$:
  - Rather than discarding item j flagged with DIF, we can now replace the item response function for j (with θ only) with the multidimensional item response function given by the GLM involving θ and η_{ij}. This aims to explicitly model the complete latent space (both θ and η_{ij}) on DIF items to mitigate DIF effect on the estimated θ (Ackerman & Ma, 2024).
  - For items without DIF, the item response function stays the same as before.
  - Plugging in the estimated $\hat{\eta}_{ij}$, we can re-estimate θ with all items (with & without DIF).
- Interpreting DIF with $\hat{\eta}_{ij}$:
  - Can look at which behavioral patterns from the log data are associated with the $\hat{\eta}_{ij} = \hat{\boldsymbol{\omega}}^{\top}\boldsymbol{X}_{ij}$.
  - The next page is an example item from the PIAAC PSTRE that was flagged with DIF related to age (PIAAC surveyed working-age adults with age between 16 and 65).

PIAAC PSTRE Nuisance Interpretations

U01a, “Party Invitation”, required examinees to move emails in the Inbox folder into the correct places – “Can Come” folder and “Cannot Come” folder.

Nuisance trait $\hat{\eta}_{ij}$ correlated with drag-and-drop usage ($r = 0.61$).
Younger examinees used drag/drop more often.





Speculations:

- Each line of email was narrow in the inbox.
- On another spreadsheet item (where each line of data is similarly narrow, older participants were also less likely to eyeball the entire spreadsheet to find some information.
- One possibility is these tasks are more visually taxing to older participants, and allowing zooming on the interface may help reduce DIF.

Concluding Remarks

- In the feature extraction stage, we transformed $\boldsymbol{S}_{ij}$ into $\boldsymbol{X}_{ij}$ (process features) that contain original sequence information but are numerical and rectangular.
- Process features can still have irregular distributions and can be noisy.
  - **Case study 1**: some process features are irregularly distributed, so we applied nonparametric hypothesis tests.
  - **Case study 2**: not all information in the MDS features is relevant to the measured trait, so we summarize θ-relevant process information (T) by regressing MDS features onto the response-based score $\hat{\theta}_Y$.
  - **Case study 3**: not all information in the MDS features is relevant to the nuisance accounting for DIF, so we summarize by finding a weight sum that best accounts for group differences in item response unaccounted for by θ.
- Use caution in making distributional assumptions, and apply regression or variable selection (if features are high-dimensional) to separate signal from noise.

References

- Ackerman, T. A., & Ma, Y. (2024). Examining differential item functioning from a multidimensional IRT perspective. *Psychometrika*, 89(1), 4-41.
- American Educational Research Association, American Psychological Association, & National Council on Measurement in Education (Eds.). (2014). *Standards for educational and psychological testing*. American Educational Research Association.
- Chen, L., Zhang, S., & Liu, J. (2025). Reducing differential item functioning via process data. *Psychometrika*.
- Chen, Y. (2020). A continuous-time dynamic choice measurement model for problem-solving process data. *Psychometrika*, 85(4), 1052-1075.
- Chen, Y., & Zhang, S. (2020). A Latent Gaussian process model for analysing intensive longitudinal data. *British Journal of Mathematical and Statistical Psychology,* 73(2), 237-260.
- Goldhammer, F., Hahnel, C., Kroehne, U., & Zehner, F. (2021). From byproduct to design factor: On validating the interpretation of process indicators based on log data. *Large-Scale Assessments in Education,* 9(1), 20.
- Gómez-Alonso, C., & Valls, A. (2008, October). A similarity measure for sequences of categorical data based on the ordering of common elements. *In International conference on modeling decisions for artificial intelligence* (pp. 134-145). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Hao, J., Shu, Z., & von Davier, A. (2015). Analyzing process data from game/scenario-based tasks: an edit distance approach. *Journal of Educational Data Mining*, 7(1), 33-50.
- Han, Z., He, Q., & von Davier, M. (2019). Predictive feature generation and selection using process data from PISA interactive problem-solving items: An application of random forests. *Frontiers in Psychology,* 10, 2461.
- He, Q., Borgonovi, F., & Paccagnella, M. (2021). Leveraging process data to assess adults' problem-solving skills: Using sequence mining to identify behavioral patterns across digital tasks. *Computers & Education,* 166, 104170.
- He, Q., Borgonovi, F., & Suárez-Álvarez, J. (2023). Clustering sequential navigation patterns in multiple-source reading tasks with dynamic time warping method. Journal of Computer *Assisted Learning,* 39(3), 719-736.

References

- He, Q., & von Davier, M. (2016). Analyzing process data from problem-solving items with n-grams: Insights from a computer-based large-scale assessment. In *Handbook of research on technology tools for real-world skill development* (pp. 750-777). IGI Global Scientific Publishing.
- He, S., & Cui, Y. (2025). A systematic review of the use of log-based process data in computer-based assessments. *Computers & Education*, 105245.
- Liao, D., He, Q., & Jiao, H. (2019). Mapping background variables with sequential patterns in problem-solving environments: An investigation of United States adults' employment status in PIAAC. *Frontiers in Psychology,* 10, 646.
- National Center for Education Statistics. (n.d.). NAEP Questions Tool [Database]. The Nation's Report Card. Retrieved January 17, 2026, from nationsreportcard.gov/nqt/
- Organisation for Economic Co-operation and Development. (n.d.). PISA 2012 computer-based administration (CBA) database [Data set]. OECD Data. Retrieved January 17, 2026, from oecd.org/en/data/datasets/pisa-2012-cba-database.html
- Organisation for Economic Co-operation and Development. (2014). *PISA 2012 technical report*. OECD Publishing. https://doi.org/10.1787/6341a959-en
- Salles, F., Dos Santos, R., & Keskpaik, S. (2020). When didactics meet data science: Process data analysis in large-scale mathematics assessment in France. *Large-scale Assessments in Education,* 8(1), 7.
- Stadler, M., Fischer, F., & Greiff, S. (2019). Taking a closer look: An exploratory analysis of successful and unsuccessful strategy use in complex problems. *Frontiers in psychology,* 10, 777.
- Tang, X., Wang, Z., He, Q., Liu, J., & Ying, Z. (2020). Latent feature extraction for process data via multidimensional scaling. *Psychometrika*, 85(2), 378-397.
- Tang, X., Wang, Z., Liu, J., & Ying, Z. (2021). An exploratory analysis of the latent structure of process data via action sequence autoencoders. *British Journal of Mathematical and Statistical Psychology,* 74(1), 1-33.

References

- Tang, X., Zhang, S., Wang, Z., Liu, J., & Ying, Z. (2021). Procdata: An R package for process data analysis. *Psychometrika*, 86(4), 1058-1083.
- Tang, X. (2024). A latent hidden Markov model for process data. *Psychometrika*, 89(1), 205-240.
- Ulitzsch, E., He, Q., Ulitzsch, V., Molter, H., Nichterlein, A., Niedermeier, R., & Pohl, S. (2021). Combining clickstream analyses and graph-modeled data clustering for identifying common response processes. *Psychometrika*, 86(1), 190-214.
- U.S. Department of Education, Institute of Education Sciences, National Center for Education Statistics. (2017). National Assessment of Educational Progress (NAEP), 2017 Mathematics Assessment [Data set]. The Nation's Report Card. nationsreportcard.gov/math_2017/
- Ulitzsch, E., Ulitzsch, V., He, Q., & Lüdtke, O. (2023). A machine learning-based procedure for leveraging clickstream data to investigate early predictability of failure on interactive tasks. *Behavior Research Methods*, 55(3), 1392-1412.
- Wang, Z., Tang, X., Liu, J., & Ying, Z. (2023). Subtask analysis of process data through a predictive model. *British Journal of Mathematical and Statistical Psychology,* 76(1), 211-235.
- Wei, X., & Zhang, S. (2024). Extended time accommodation and the academic, behavioral, and psychological outcomes of students with learning disabilities. *Journal of Learning Disabilities,* 57(4), 242-254.
- Xiao, Y., He, Q., Veldkamp, B., & Liu, H. (2021). Exploring latent states of problem-solving competence using hidden Markov model on process data. *Journal of Computer Assisted Learning*, 37(5), 1232-1247.
- Zhang, S., Wang, Z., Qi, J., Liu, J., & Ying, Z. (2023). Accurate assessment via process data. *Psychometrika*, 88(1), 76-97.
- Zhang, S., Tang, X., He, Q., Liu, J., & Ying, Z. (2024). External Correlates of Adult Digital Problem-Solving Process. *Zeitschrift für Psychologie.*
- Zhu, M., Shu, Z., & von Davier, A. A. (2016). Using networks to visualize and analyze process data for educational assessment. *Journal of Educational Measurement,* 53(2), 190-211.