UNIVERSITY OF MICHIGAN

SOLAR
SOCIETY for LEARNING ANALYTICS RESEARCH

Temporal and sequential analysis for learning analytics

Session 1A: Monday, June 21, 2021, 2-4pm PDT
Session 1B: Wednesday, June 23, 2021, 1-3pm PDT
Session 1C: Friday, June 25, 2021,1-3pm PDT

Please say hello and introduce yourself in the Zoom chat! ☺

Quan Nguyen, Ph.D.
Postdoctoral Fellow
School of Information

Email: quanngu@umich.edu
Twitter: @QuanNguyen3010

Agenda

W8. Temporal and sequential analysis for learning analytics ***Quan Nguyen – University of Michigan*** **Zoom**: https://us02web.zoom.us/j/85872971310 **Slack**: Invitation by email **R packages**: TraMineR, arules		
Session	**Activity**	**Resources**
1A: Monday, June 21, 2021 from 2pm to 4pm PDT	**Group discussion**: Overview of temporal analysis techniques in LA and discussion on the type of RQs and learning constructs that are suitable for temporal analysis	• Knight, S., Friend Wise, A., & Chen, B. (2017). Time for Change: Why Learning Analytics Needs Temporal Analysis. *Journal of Learning Analytics*, 4(3), 7–17. • Chen, B., Knight, S., & Wise, A. F. (2018). Critical Issues in Designing and Implementing Temporal Analytics. *Journal of Learning Analytics*, 5(1), 1–9.
	Tutorial: Exploratory data analysis for states sequences data (visualization, basic descriptive statistics for sequences), using package TraMineR	• Gabadinho, A., G. Ritschard, N.S. Müller and M. Studer (2011). "Analyzing and Visualizing State Sequences in R with TraMineR." Journal of Statistical Software, 40(4), 1–37. • Gabadinho, A., G. Ritschard, M. Studer and N. S. Müller Mining sequence data in R with the TraMineR package: A user's guide. University of Geneva, 2010
1B: Wednesday, June 23, 2021 from 1pm to 3pm PDT	**Tutorial**: How can we detect common learning patterns from students' log data using sequential analysis? (Sequence similarities, optimal matching, clustering sequences)	
1C: Friday, June 25, 2021 from 1pm to 3pm PDT	**Tutorial**: How can we identify courses that are frequently taken together and provide course recommendations? (Association rule mining with apriori algorithm)	https://www.kirenz.com/post/2020-05-14-r-association-rule-mining/

Note: Slides & codes will be posted on https://github.com/quan3010/temporal_analysis

Group discussion

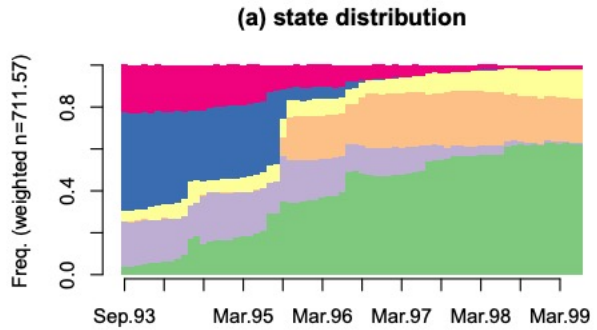
Data source	Time granularity	RQs/Learning constructs	Analysis technique
SIS/Canvas		Changes in behaviors, understanding, or knowledge	Regression MLM ANOVA Recurrence quantification Markov Growth MLM CNN transition diagnostic classification models
CLickstream		SRL calibration accuracy	
Eye-tracking		Dynamics of thinking (thought types) during learning sessions	
Mobile app		writing flow from keystroke data	
Keystroke		What is the path learners take in an online course?	
Location data		changes of learning behavior due to covid-19	
Assessment		forecasting dropout and fail risk	
Writing		Timie management	
		Time on task	
		Course design	
		Location	

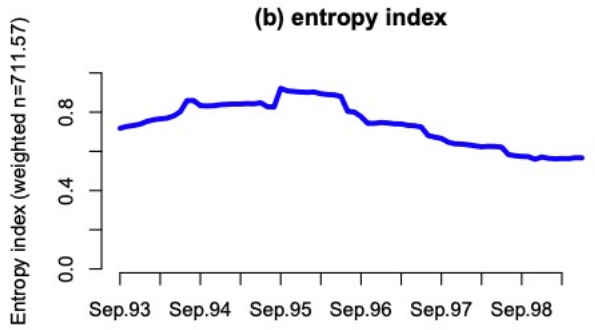
Overview

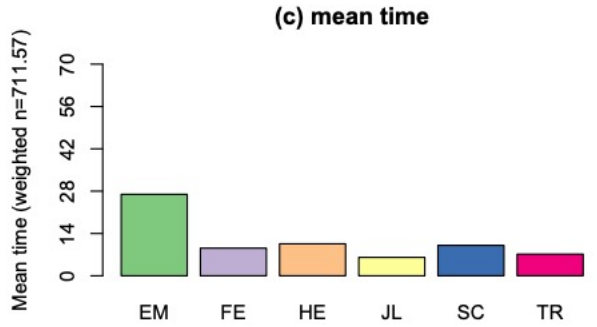
- Sequence analysis → Learning strategies, tactics → cluster by performance
- Process mining → transitional probabilities between activities
- Time-series analysis (ARIMA and the like) → I haven't seen much application in LA, partly because of the data type (panel data instead of univariate time-series)
- ML (LSTM, RNN) → Predict learner's outcome such as dropout on a weekly basis
- Temporal network analysis → SOAMS, REM, ENA
- Theory-driven → procrastination, spaced learning, SRL phases, etc…

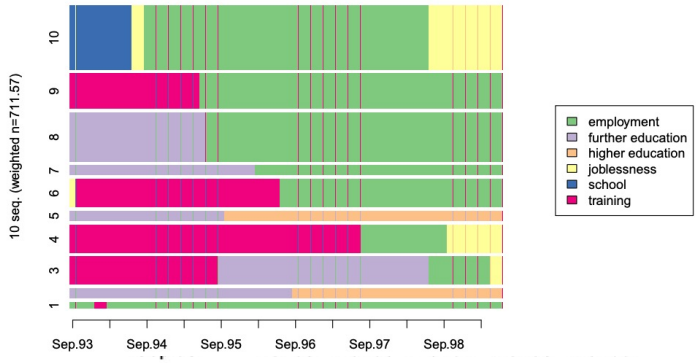
TraMineR

- A toolbox for exploring, rendering and analyzing categorical sequence data
- Visualize
- Descriptive stats
- Clustering
- Association rules

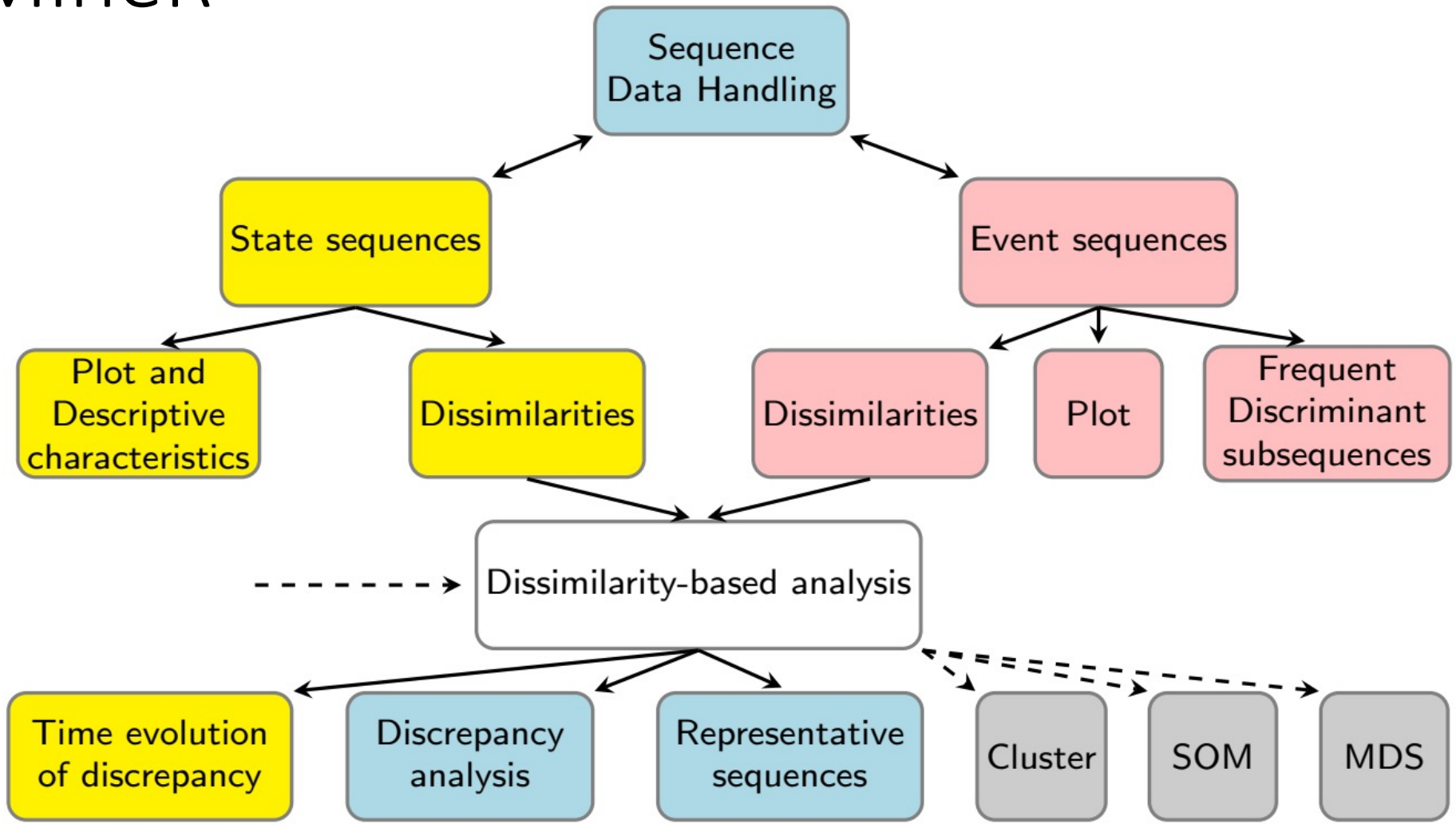








TraMineR


Sequence Data Handling
State sequences
Event sequences
Plot and Descriptive characteristics
Dissimilarities
Dissimilarities
Plot
Frequent Discriminant subsequences
Dissimilarity-based analysis
Time evolution of discrepancy
Discrepancy analysis
Representative sequences
Cluster
SOM
MDS

TraMineR

- Handling of longitudinal data and conversion between various sequence formats
- Plotting sequences (distribution plot, frequency plot, index plot and more)
- Individual longitudinal characteristics of sequences (length, time in each state, longitudinal entropy, turbulence, complexity and more)
- Sequence of transversal characteristics by position (transversal state distribution, transversal entropy, modal state)
- Other aggregated characteristics (transition rates, average duration in each state, sequence frequency)
- Dissimilarities between pairs of sequences (Optimal matching, Longest common subsequence, Hamming, Dynamic Hamming, Multichannel and more)
- Representative sequences and discrepancy measure of a set of sequences
- ANOVA-like analysis and regression tree of sequences
- Rendering and highlighting frequent event sequences
- Extracting frequent event subsequences
- Identifying most discriminating event subsequences
- Association rules between subsequences

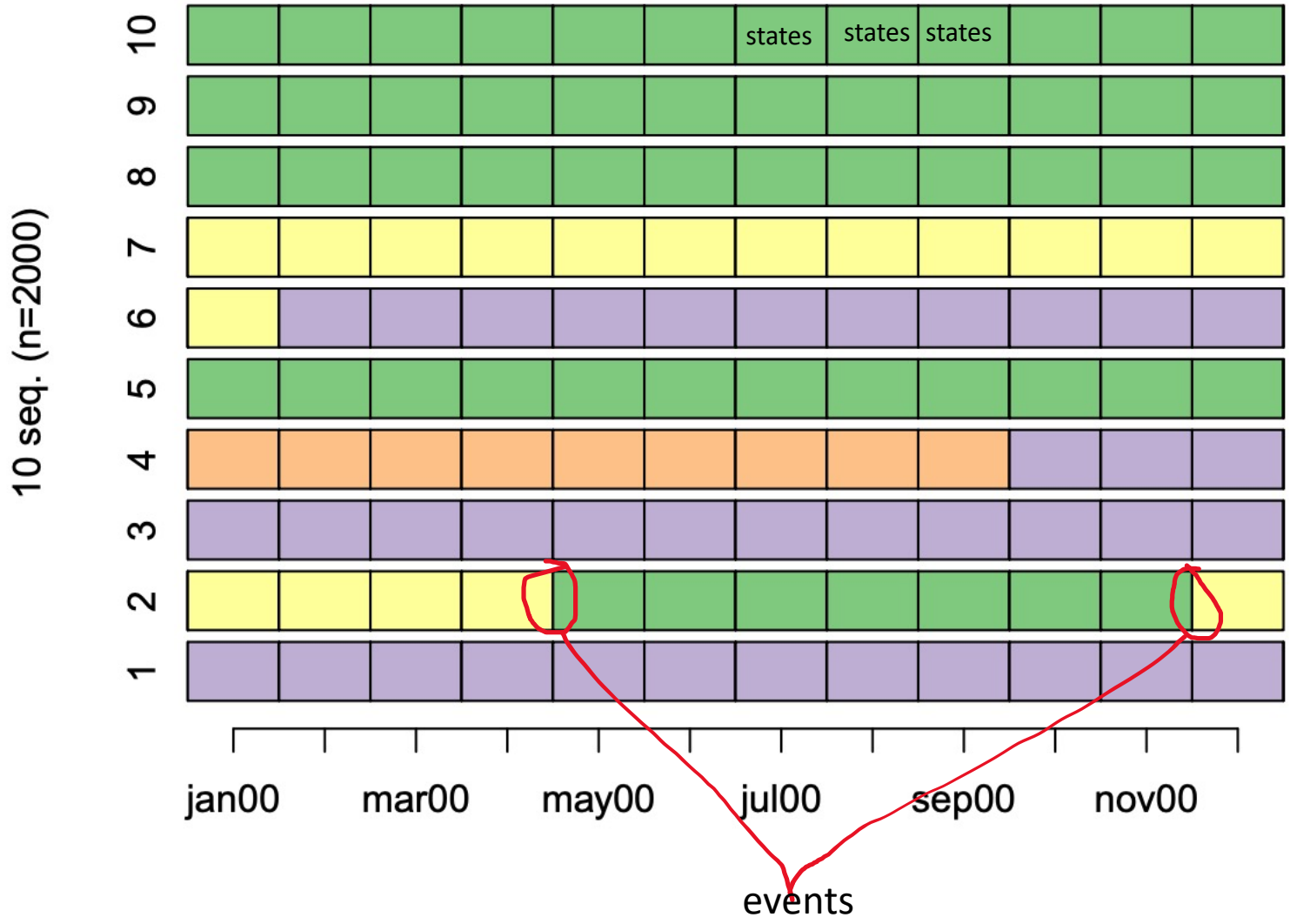
TraMineR – mvad data

- Life-trajectories

Table 3.5: List of Variables in the *MVAD* data set

id	unique individual identifier
weight	sample weights
male	binary dummy for gender, 1=male
catholic	binary dummy for community, 1=Catholic
Belfast	binary dummies for location of school, one of five Education and Library Board areas in Northern Ireland
N.Eastern	"
Southern	"
S.Eastern	"
Western	"
Grammar	binary dummy indicating type of secondary education, 1=grammar school
funemp	binary dummy indicating father's employment status at time of survey, 1=father unemployed
gcse5eq	binary dummy indicating qualifications gained by the end of compulsory education, 1=5+ GCSEs at grades A-C, or equivalent
fmpr	binary dummy indicating SOC code of father?s current or most recent job,1=SOC1 (professional, managerial or related)
livboth	binary dummy indicating living arrangements at time of first sweep of survey (June 1995), 1=living with both parents
jul93	Monthly Activity Variables are coded 1-6, 1=school, 2=FE, 3=employment, 4=training, 5=joblessness, 6=HE
:	"
jun99	"

States vs events



Define states sequences

View(mvad.seq)

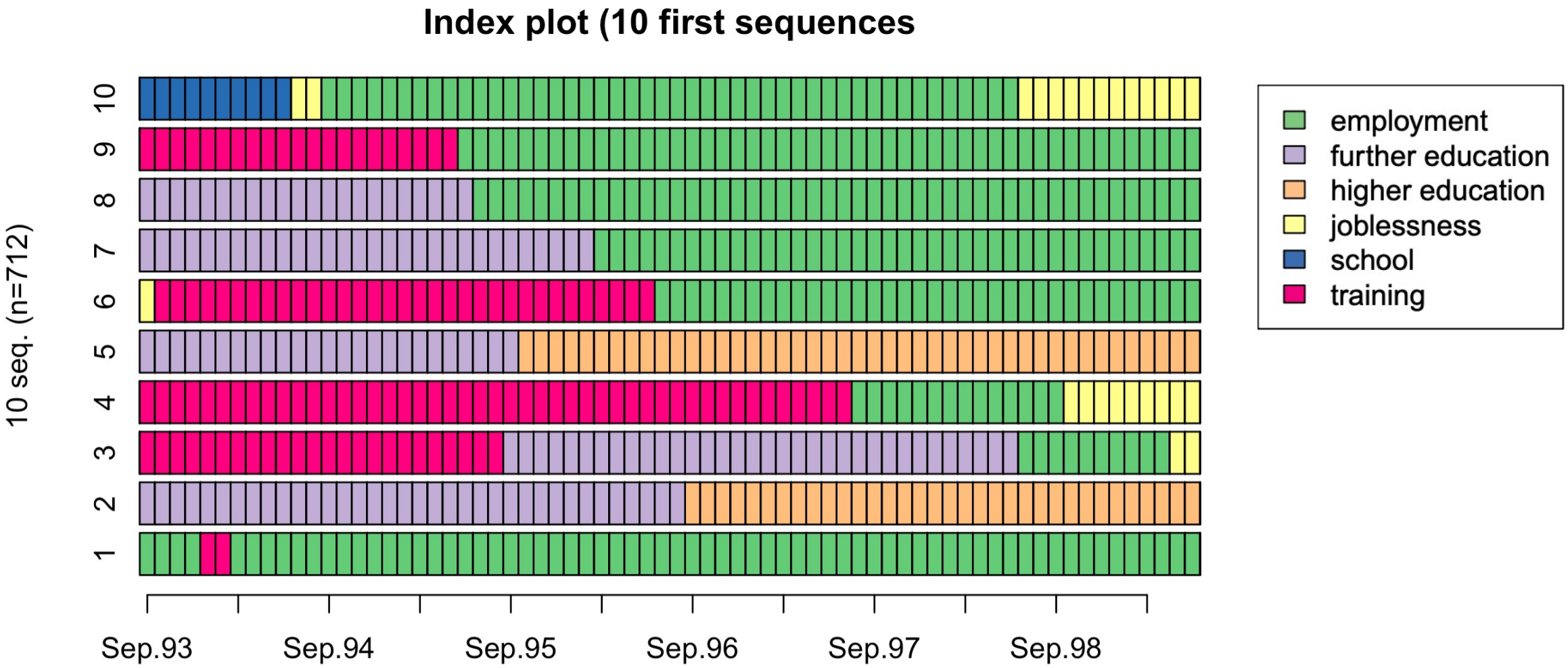
Sep.93	Oct.93	Nov.93	Dec.93	Jan.94	Feb.94
EM	EM	EM	EM	TR	TR
FE	FE	FE	FE	FE	FE
TR	TR	TR	TR	TR	TR
TR	TR	TR	TR	TR	TR
FE	FE	FE	FE	FE	FE
JL	TR	TR	TR	TR	TR
FE	FE	FE	FE	FE	FE
FE	FE	FE	FE	FE	FE
TR	TR	TR	TR	TR	TR
SC	SC	SC	SC	SC	SC
FE	FE	FE	FE	FE	FE
SC	SC	SC	SC	SC	SC
SC	SC	SC	SC	SC	SC

Two sequences describing family formation histories of two individuals. S=single, M=married, MC=married with children, D=divorced

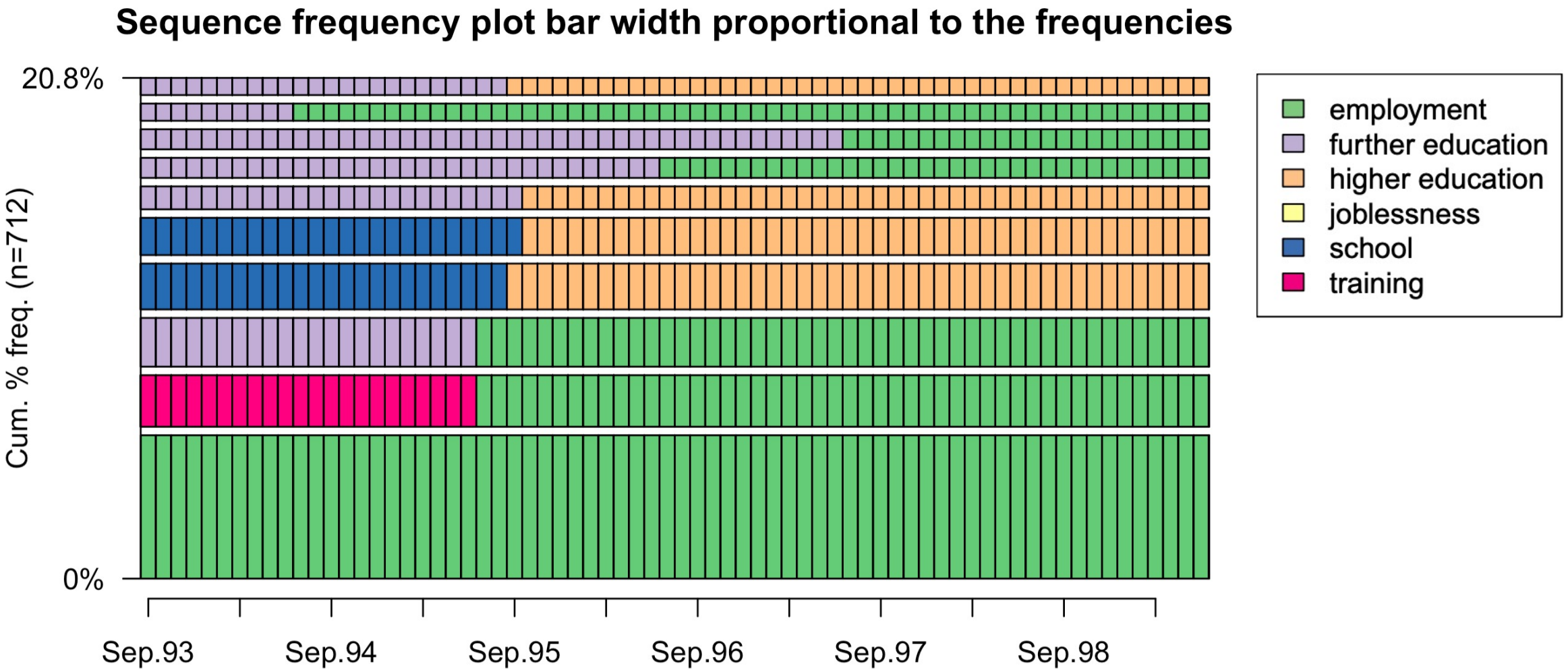
Table 4.2: Sequence data representations: Examples

Code	Example										
STS	**Id**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**
	101	S	S	S	M	M	MC	MC	MC	MC	D
	102	S	S	S	MC	MC	MC	MC	MC	MC	MC
SPS (1)	**Id**	**State 1**		**State 2**		**State 3**		**State 4**		**State 5**	
	101	(S,3)		(M,2)		(MC,4)		(D,1)			
	102	(S,3)		(MC,7)							
SPS (2)	**Id**	**State 1**		**State 2**		**State 3**		**State 4**		**State 5**	
	101	S/3		M/2		MC/4		D/1			
	102	S/3		MC/7							
DSS	**Id**	**State 1**		**State 2**		**State 3**		**State 4**		**State 5**	
	101	S		M		MC		D			
	102	S		MC							
TSE	**id**	**time**	**event**								
	101	21	Marriage								
	101	23	Child								
	101	27	Divorce								
	102	21	Marriage								
	102	21	Child								
SPELL	**id**	**index**	**from**	**to**	**status**						
	101	1	18	20	Single						
	101	2	21	22	Married						
	101	3	23	26	Married w Children						
	101	4	27	..	Divorced						
	102	1	18	20	Single						
	102	2	21	27	Married w Children						

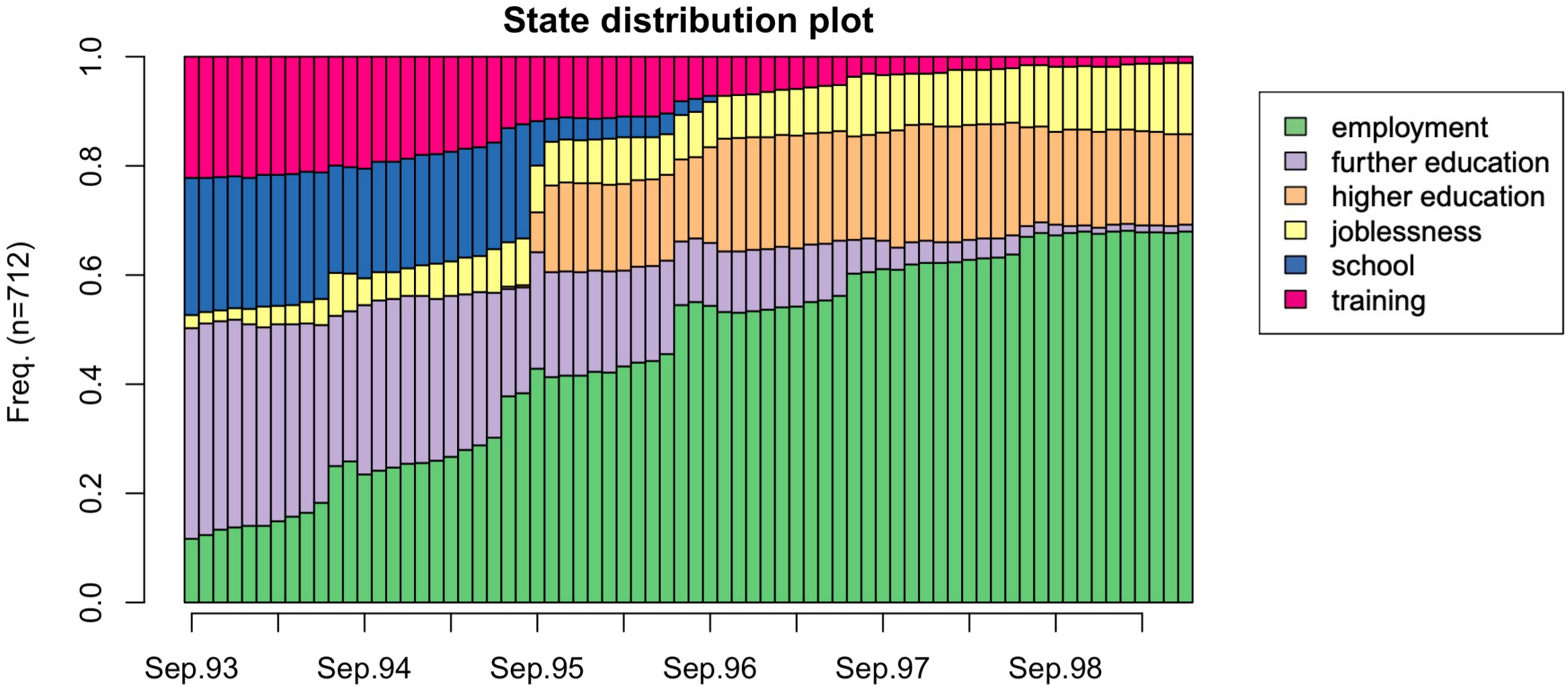
What do people's life trajectories look like?



Which sequences are the most common?



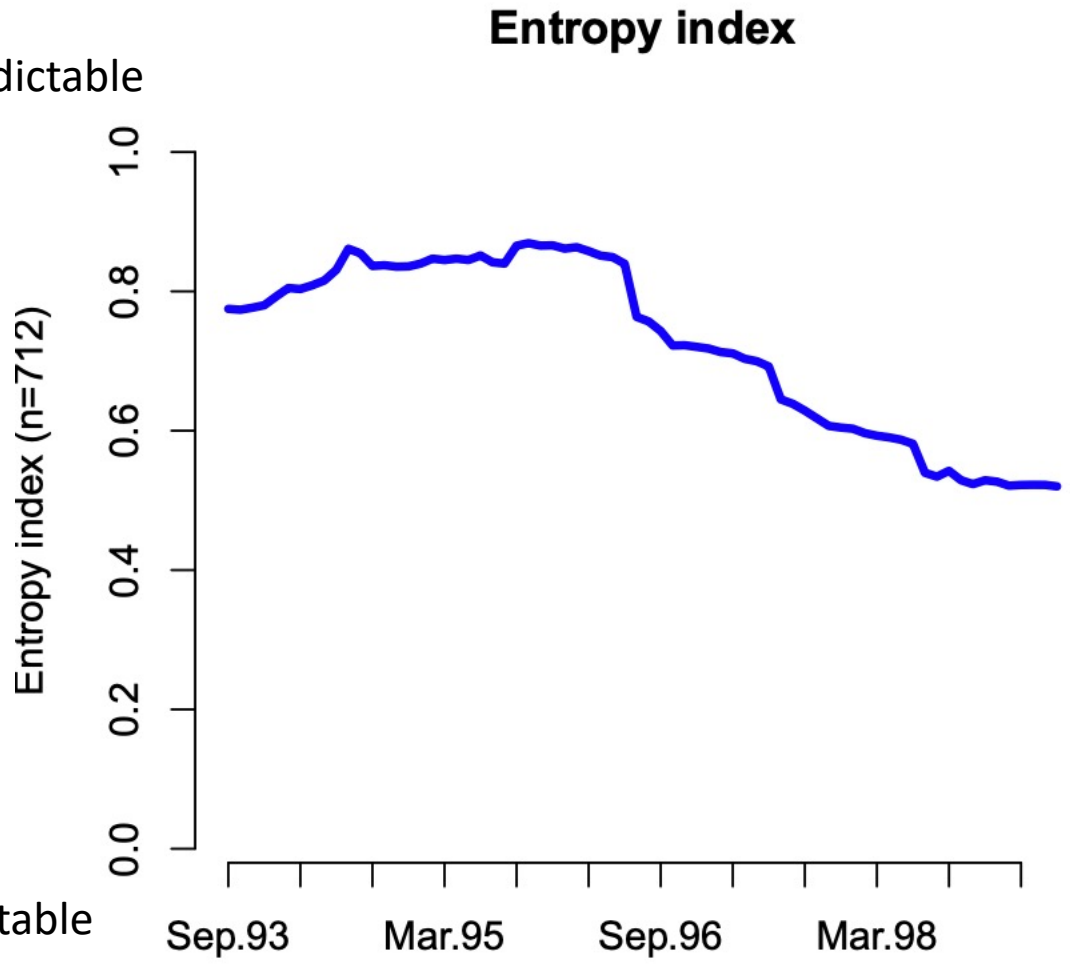
Sequence distribution over time

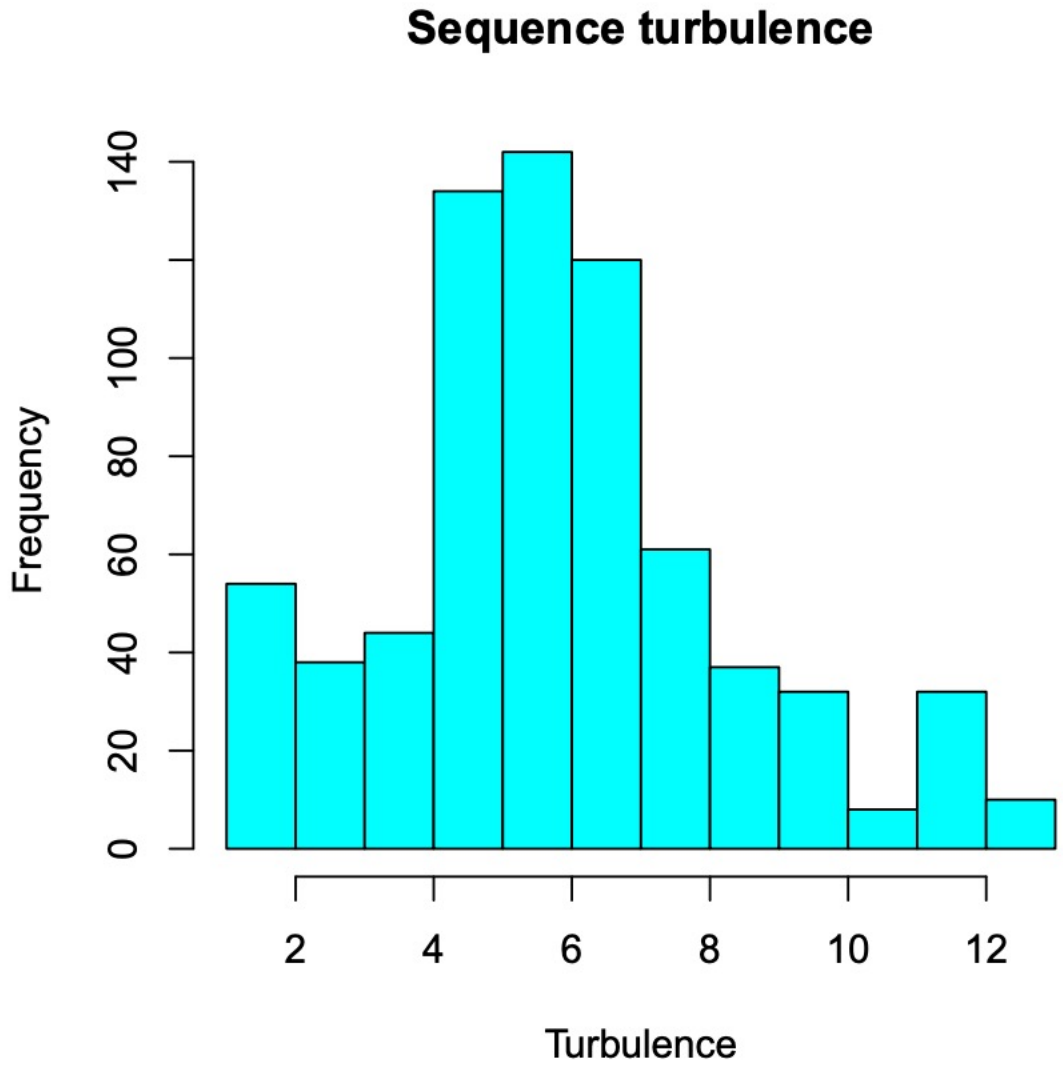


How stable are these sequences over time?

unpredictable

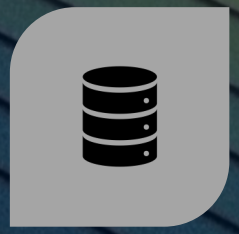
predictable





Case study 1

AN 8 MONTH LONG ONLINE COURSE



LOG DATA

GRADES

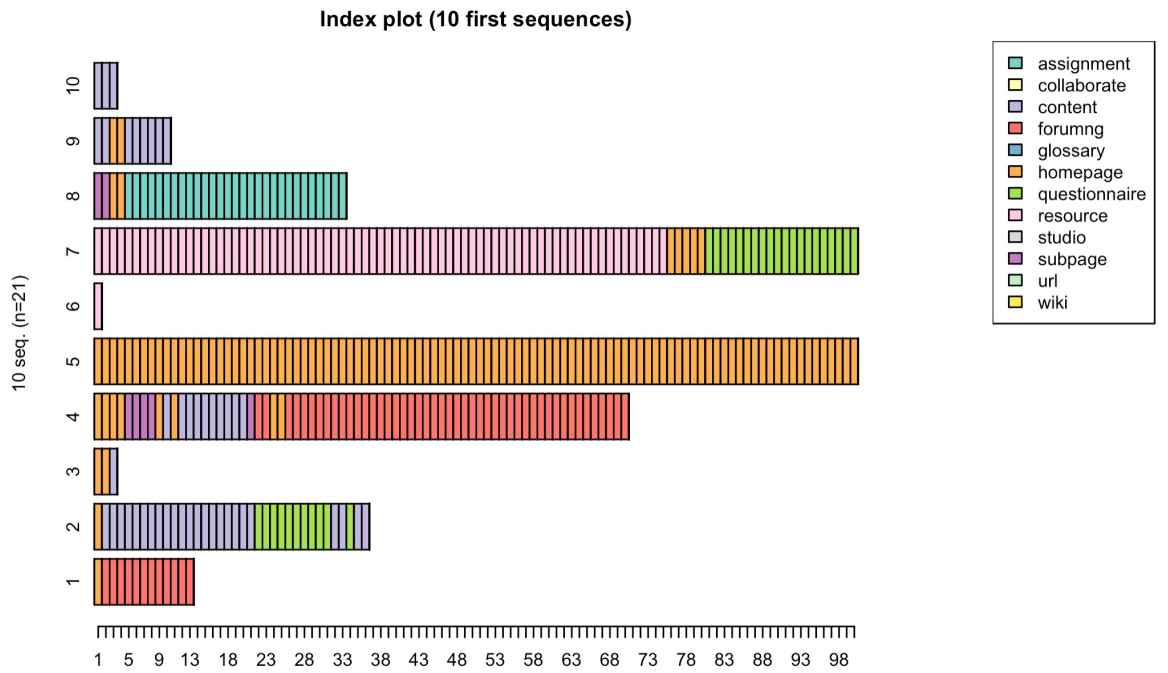
RQ: WHAT ARE SOME COMMON LEARNING PATTERNS AND HOW DO THEY RELATE TO ACADEMIC PERFORMANCE?

Analysis path

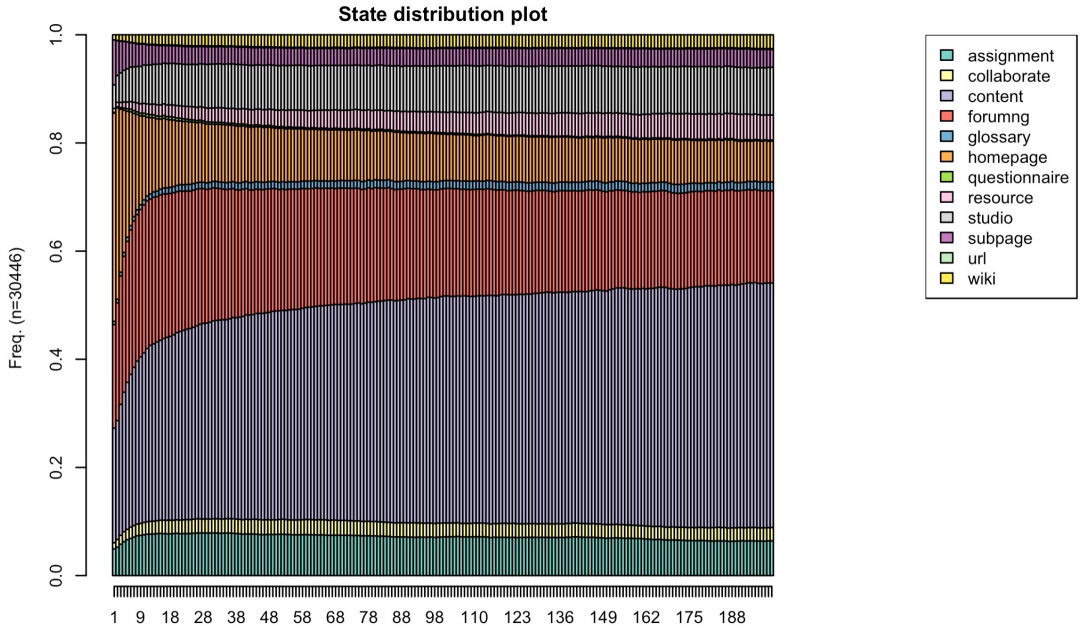
1. Define unit of analysis (e.g., a learning session where no consecutive logs are greater than 30 mins, a student during a reading session, etc…)
2. Define time unit (e.g., seconds, minutes, 15m, order)
3. Label sequences
4. Format sequences (create a sequence object)
5. Visualize
6. Descriptive stats
7. Compute distance matrix of dissimilarities
8. Clustering & other types of statiscs

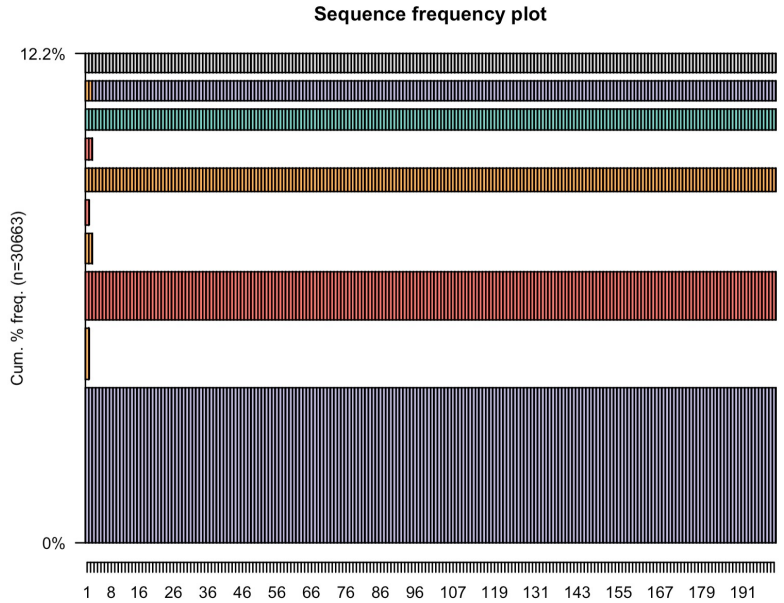
Recap

	id	date_time	spent_time	site_type	instancename	PassFlag
204713	44	2016-09-07 23:55:00	9000	content	Block 1	1
204839	44	2016-09-07 23:55:00	18000	content	Block 1	1
206001	44	2016-09-07 23:56:00	27000	content	Course introduction	1
206462	44	2016-09-07 23:56:00	8000	homepage	homepage	1
206856	44	2016-09-07 23:56:00	3000	content	Module guide	1
206915	44	2016-09-07 23:56:00	10000	homepage	homepage	1
203578	44	2016-09-07 23:57:00	97000	content	Block 1	1
205889	44	2016-09-07 23:57:00	4000	homepage	homepage	1
201227	44	2016-09-07 23:58:00	3549000	content	Block 1	1
612717	182	2016-09-08 00:10:00	92000	homepage	homepage	1
606166	182	2016-09-08 00:12:00	34000	homepage	homepage	1
607979	182	2016-09-08 00:12:00	5000	homepage	homepage	1
609434	182	2016-09-08 00:12:00	16000	homepage	homepage	1
617476	182	2016-09-08 00:13:00	55000	content	Module guide	1

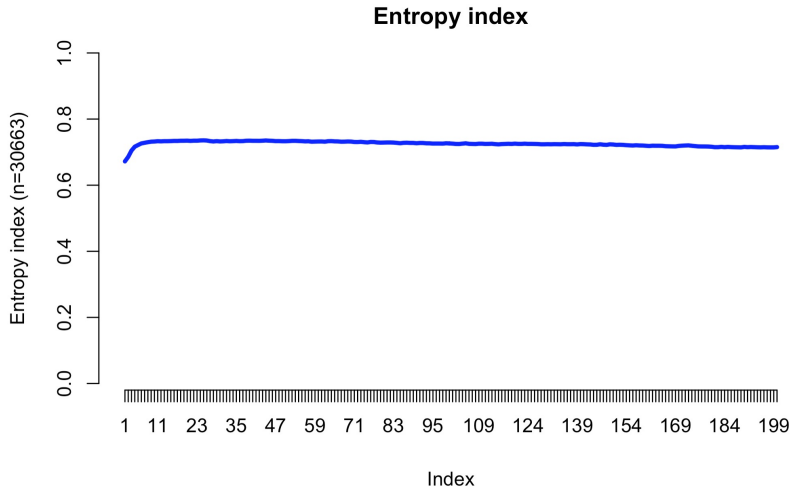


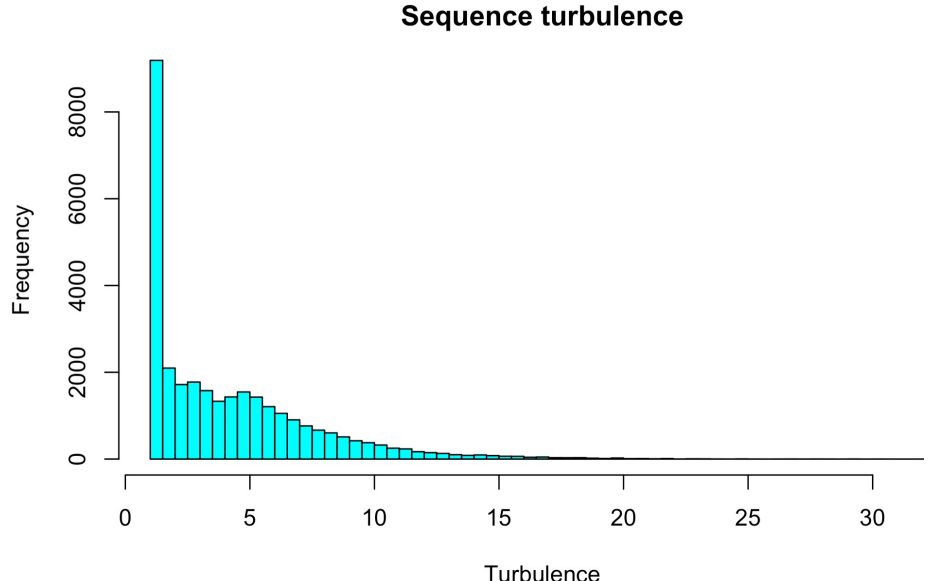
Recap





	Freq	Percent
content/200	1455	4.78
forumng/200	481	1.58
homepage/1	393	1.29
homepage/2	315	1.03
forumng/2	237	0.78
homepage/200	217	0.71
homepage/2-content/198	204	0.67
assignment/200	200	0.66
forumng/1	186	0.61
studio/200	183	0.60





Our goal today

- Find common learning patterns (find groups of sequences that are ‘close’ to each other)
- Intuition:

1. Compute distance matrix between sequences
2. Apply clustering technique
3. Correlate covariates with clustering membership

Sequences dissimilarity

- S1 = {A B C A C}
- S2 = {A C B B C}

Dissimilarity can be measured by:

- **Common attributes**: The more common attributes, the more similar
- **Edit distance**: The lower the edit cost, the more similar

Sequences dissimilarity

- Hamming distance: counting common states position-wise
- S1 = {A B C A C}
- S2 = {A C B B C}

Hamming distance = 5 – 2 = 3

Hamming distance compares two sequences of equal length

Sequences dissimilarity

- S1 = {A B C A C}
- S2 = {A C B B C}

Common attributes: states appearing in the same order and for a same duration → length of the longest common subsequence (LCS)

Can be used for sequences of different lengths

Sequences dissimilarity

Optimal Matching (OM): distance between two sequence x and y is minimal cost of transforming x into y, using:

- indel (**insert or delete**) with assigned indel cost(s)
- **substitution** between states with assigned substitution costs

	S1 = {A B C A C}
Operation	S S S
	S2 = {A C B B C}

Substitution cost = 2

Total cost = 2 * 3 = 6

	S1 = {A - B - C - A C}
Operation	I D I D I D
	S2 = {A C - B - B - C}

Indel cost = 1

Total cost = 1 * 6 = 6

Sequences dissimilarity

Strategies for defining the costs in OM:

- **Constant** costs (indel cost = 1, substitution cost = 2)

	assignment->	collaborate->	content->	forumng->	glossary->	homepage->
assignment->	0	2	2	2	2	2
collaborate->	2	0	2	2	2	2
content->	2	2	0	2	2	2
forumng->	2	2	2	0	2	2
glossary->	2	2	2	2	0	2
homepage->	2	2	2	2	2	0

- Costs based on observed **transition rates**:
  The lower the transition rate, the higher the cost

	assignment->	collaborate->	content->	forumng->	glossary->	homepage->
assignment->	0.00	2.00	2.00	2.00	2.00	1.99
collaborate->	2.00	0.00	2.00	2.00	2.00	2.00
content->	2.00	2.00	0.00	2.00	1.99	1.97
forumng->	2.00	2.00	2.00	0.00	2.00	1.98
glossary->	2.00	2.00	1.99	2.00	0.00	2.00
homepage->	1.99	2.00	1.97	1.98	2.00	0.00

Sequences dissimilarity

Output of OM between 12 sequences = an n x n distance matrix

S1=AABC and S2=ABC

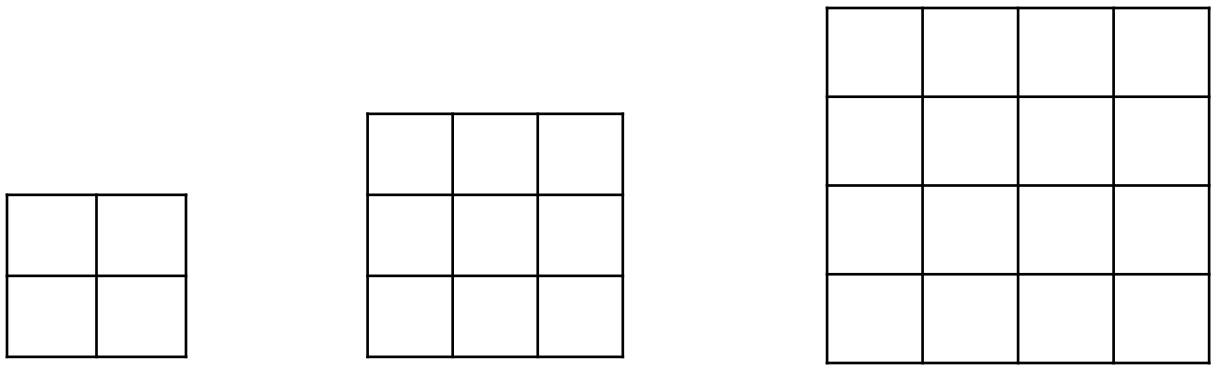
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	98	8	264	137	8	8	14	54	90	398
2	0	0	98	8	264	137	8	8	14	54	90	398
3	98	98	0	98	166	39	98	98	98	98	10	300
4	8	8	98	0	256	137	4	4	12	50	88	392
5	264	264	166	256	0	127	256	256	250	260	174	212
6	137	137	39	137	127	0	137	137	137	137	49	261
7	8	8	98	4	256	137	0	0	8	50	88	392
8	8	8	98	4	256	137	0	0	8	50	88	392
9	14	14	98	12	250	137	8	8	0	50	88	384
10	54	54	98	50	260	137	50	50	50	0	88	396
11	90	90	10	88	174	49	88	88	88	88	0	310
12	398	398	300	392	212	261	392	392	384	396	310	0

Creating cluster

1. Compute similarity distances
2. Apply clustering techniques (e.g., Ward)

Sequence similarity on big data

- Problem while computing the n x n distance matrix.
- The matrix size increases exponentially as the number of sequence increases)
- Need to use parallel computing

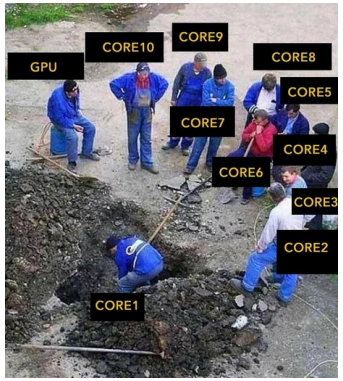


N=2 N=3 N=4

Sequence similarity on big data

- Normal matrix calculation in R without parallel processing

Core 1	Core 1	Core 1	Core 1
Core 1	Core 1	Core 1	Core 1
Core 1	Core 1	Core 1	Core 1
Core 1	Core 1	Core 1	Core 1



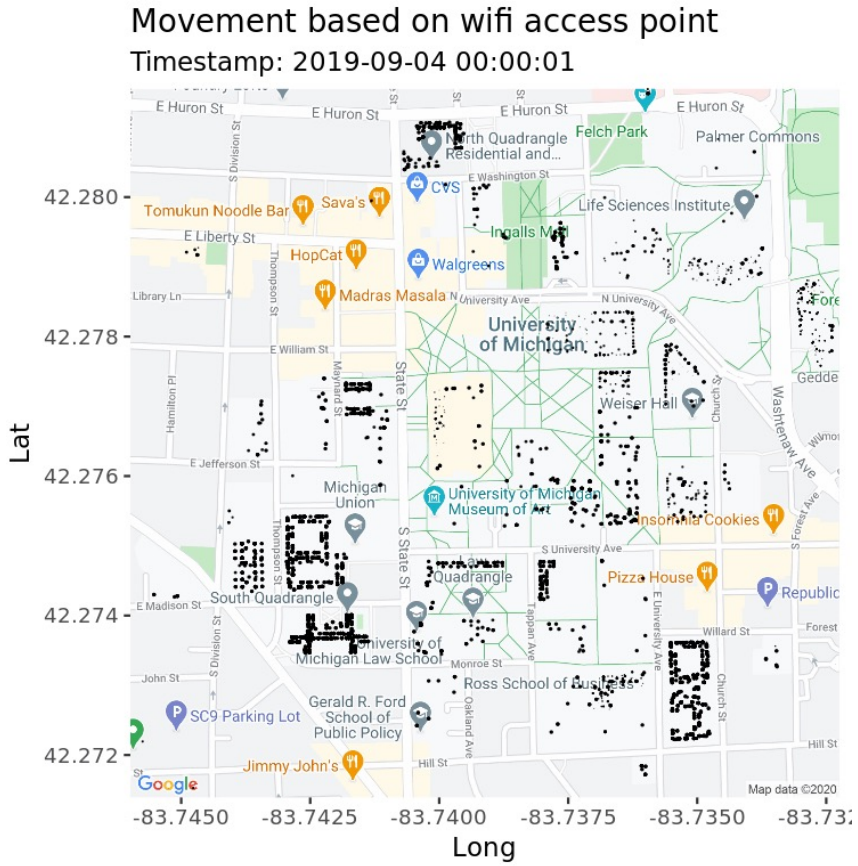
- With parallel processing

Core 1	Core 1	Core 1	Core 1
Core 2	Core 2	Core 2	Core 2
Core 3	Core 3	Core 3	Core 3
Core 4	Core 4	Core 4	Core 4

Case study

- How can we model social interactions on campus based on WiFi location data?

User	Timestamp (connected)	Timestamp (disconnected)	Access point	MAC
1	2018-09-24 08:00:00	NA	NQUAD-1023	XYZ123
1	2018-09-24 08:02:00	NA	NQUAD-2013	XYZ123
2	2018-09-24 08:00:03	2018-09-24 08:00:55	NQUAD-1210	XYZ125
2	2018-09-24 09:00:05	NA	NQUAD-3734	XYZ125

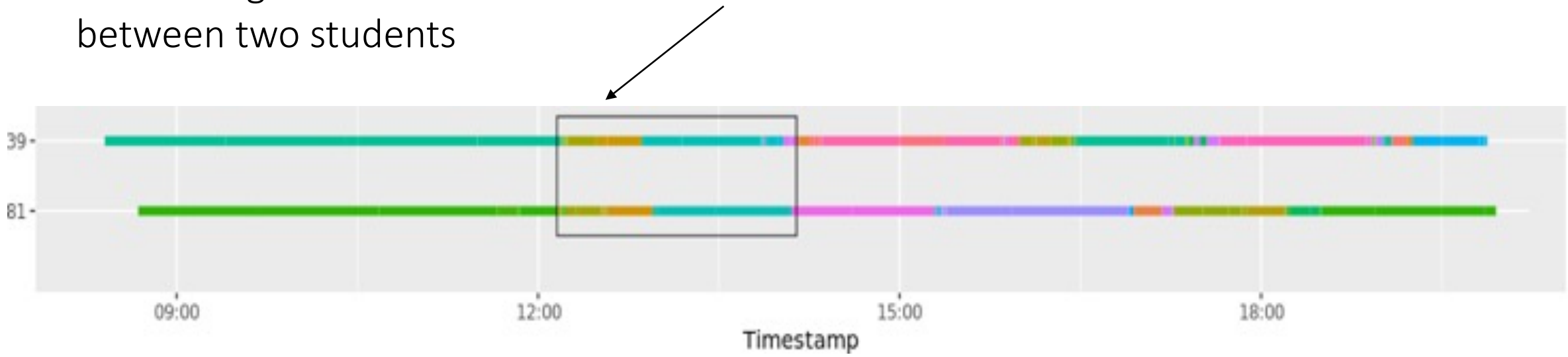


Inferring co-location from WiFi signals

- Students are presumably in a close proximity when they are connected to the same WiFi access point
- A proxy for in-person interactions (being in the same place at the same time)

Co-location network inference

- A tie was inferred when two users were connected to the sample access point during the same time window
- A tie's weight = the total amount of co-located time between two students



Problem

- >50k students
- > 1B log data points over 4 months
- Compute the overlap between 50k sequences → a 50k x 50k matrix → 1,249,975,000 calculations

Solution

- data.table: manipulate data faster with built-in parallel processing
- foverlap: find sequence overlap

https://www.rdocumentation.org/packages/data.table/versions/1.14.0/topics/foverlaps

- mclapply: trigger parallel processing

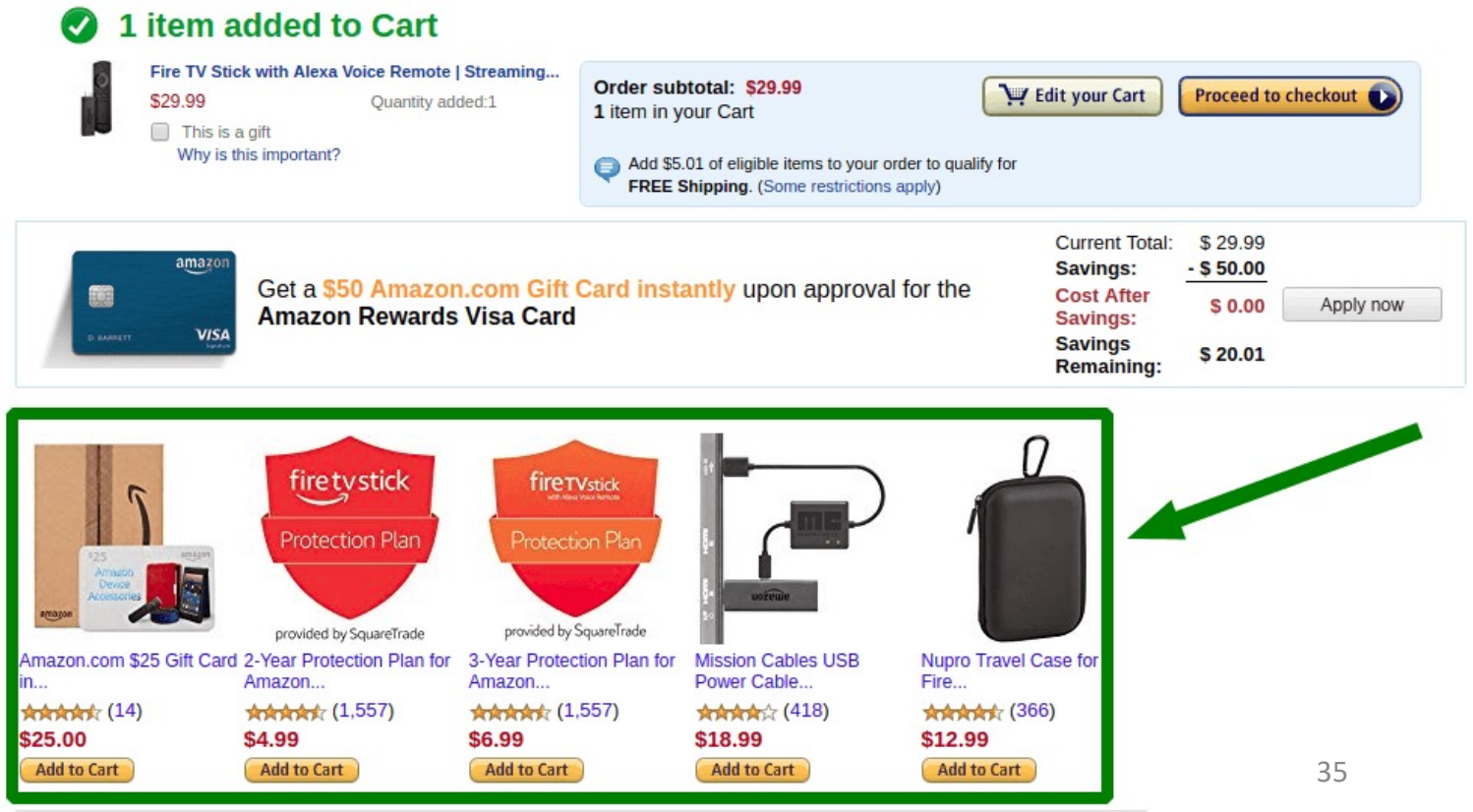
https://www.rdocumentation.org/packages/parallel/versions/3.4.1/topics/mclapply

Take-home challenge

- Create a Rmd file that showcase how you analyze your own dataset
- I am looking for 1-2 people who are willing to present their analysis in the next session

Association rule mining

- Which courses are likely to be taken together?
- Can we make a recommender system from historical course enrollment patterns?



Case study

Course enrollment pattern in 1 semester

ID	Courselist
1	PHY,MATH,PSY,ENG
2	CHEM,STATS,ENG,PSY
3	MATH,PHY,POLS,ENG
4	PSY, POLS, ENG, PHIL
5	SOC,PSY,ECON,STATS

Support: How popular an itemset is

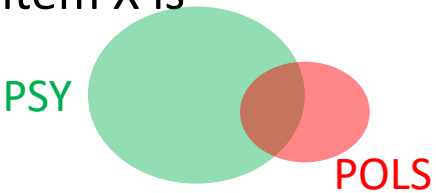
Support(X) = P(X) = Count X / Total number of transactions

Support(POLS) = 2/5 = 0.4 Support(PSY) = 4/5 = 0.8

Support(POLS, PSY) = 1/5 = 0.2

Confidence: How likely item Y is purchased when item X is purchased

Confidence(X → Y) = $\frac{P(X,Y)}{P(X)}$



Confidence(POLS, PSY) = 0.2/0.8 = 0.25

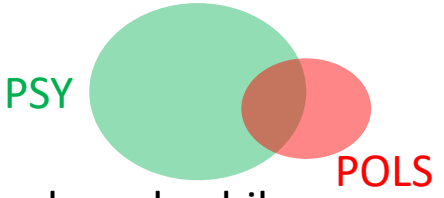
However, confidence alone might misrepresent the importance of an association

Case study

Course enrollment pattern in 1 semester

ID	Courselist
1	PHY,MATH,PSY,ENG
2	CHEM,STATS,ENG,PSY
3	MATH,PHY,POLS,ENG
4	PSY, POLS, ENG, PHIL
5	SOC,PSY,ECON,STATS

However, confidence alone might misrepresent the importance of an association



Lift: How likely item Y is purchased when item X is purchased, while controlling for how popular item Y is

Lift(X → Y) = $\frac{P(X,Y)}{P(X)P(Y)}$

Lift (POLS, PSY) = 0.2/(0.8*0.4) = 0.625

- Lift =1: no association between X and Y
- Lift > 1: item Y is *likely* to be bought if item X is bought
- Lift <1 : item Y is *unlikely* to be bought if item X is bought

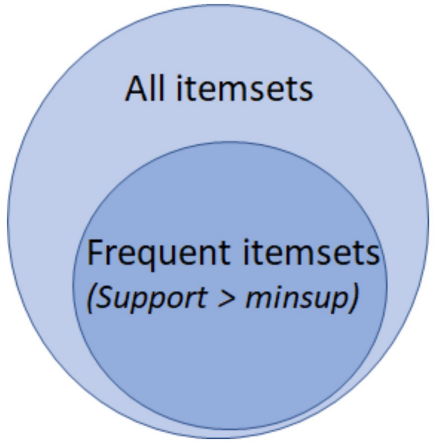
Association rule

- Rule-generation is a two-step process

1. Generating itemsets from a list of items
2. Generating all possible rules from the frequent itemsets

Association rule

1. Generating itemsets from a list of items



Frequent itemset: An itemset whose support is greater than or equal to a min_support threshold

Association rule

1. Generating itemsets from a list of items (**min_sup = 0.3**)

Candidate 1-itemsets

itemset	support
PHY	0.2
MATH	0.4
ENG	0.8
CHEM	0.2
STATS	0.4
POLS	0.4
PHIL	0.2
SOC	0.2
ECON	0.2

Frequent 1-itemsets

itemset	support
MATH	0.4
ENG	0.8
STATS	0.4
POLS	0.4

Candidate 2-itemsets

itemset	support
MATH-ENG	0.4
MATH-STATS	0
MATH-POLS	0.2
ENG-STATS	0
ENG-POLS	0.4
STATS-POLS	0

Frequent 2-itemsets

itemset	support
ENG-POLS	0.4
MATH-ENG	0.4

ID	Courselist
1	PHY,MATH,PSY,ENG
2	CHEM,STATS,ENG,PSY
3	MATH,PHY,POLS,ENG
4	PSY, POLS, ENG, PHIL
5	SOC,PSY,ECON,STATS

Association rule

2. Generating all possible rules from the frequent itemsets

min_confidence = 0.6

Frequent itemset = {ENG,POLS}

Rule	Confidence
ENG → POLS	0.4/0.4=1
POLS → ENG	0.4/0.8=0.5

Course recommender

- Support
- Confidence
- Lift
- arules
- arulesSequences