Mining Release Cycles in the Android App Store

Maleknaz Nayebi & Guenther Ruhe

UNIVERSITY OF CALGARY

SEDS Laboratory

Table of Content

- The Process
- Methodology
- Dataset
- Confirmative analysis
- Pattern recognition
- Rough set analysis
- Rule extraction
- Summary
- Future Work

Topics Studied in This Area

What We Are Looking Into?

What We Are Looking Into?

The Process

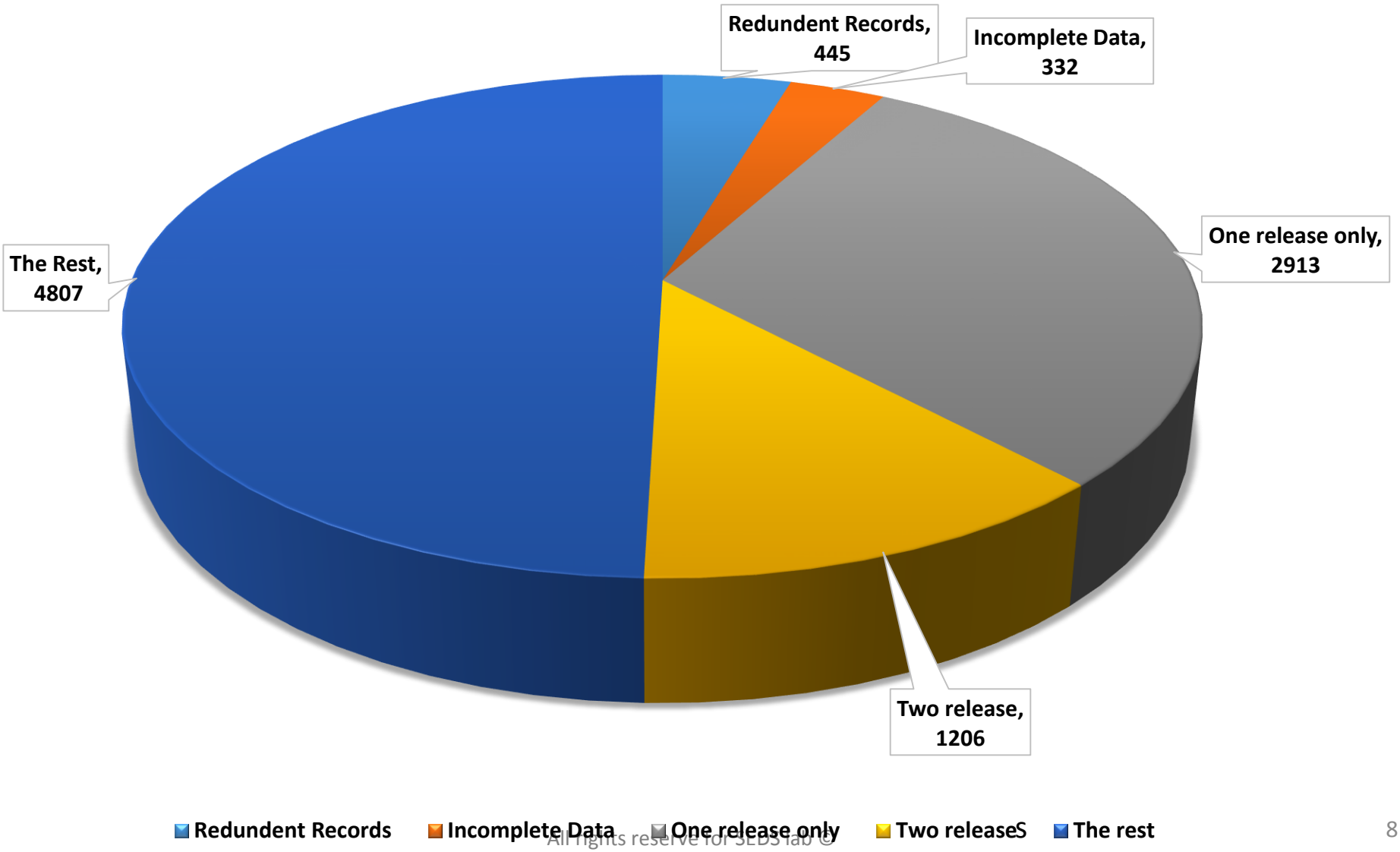
- Crawling data from Android app store and third party websites
- Data pre-processing
- Scoping (what to analyze?)
- Modeling (Dependent/Independent variables and intervals)
- Analysis
- Interpretation

Methodology

- Descriptive statistics → *Understanding*
- Inference statistics → *Validating hypotheses*
- Pattern recognition → *Detecting patterns on sequences of release cycles*
- Rough Set Analysis (RSA) → *Synthesis & explanation of results*

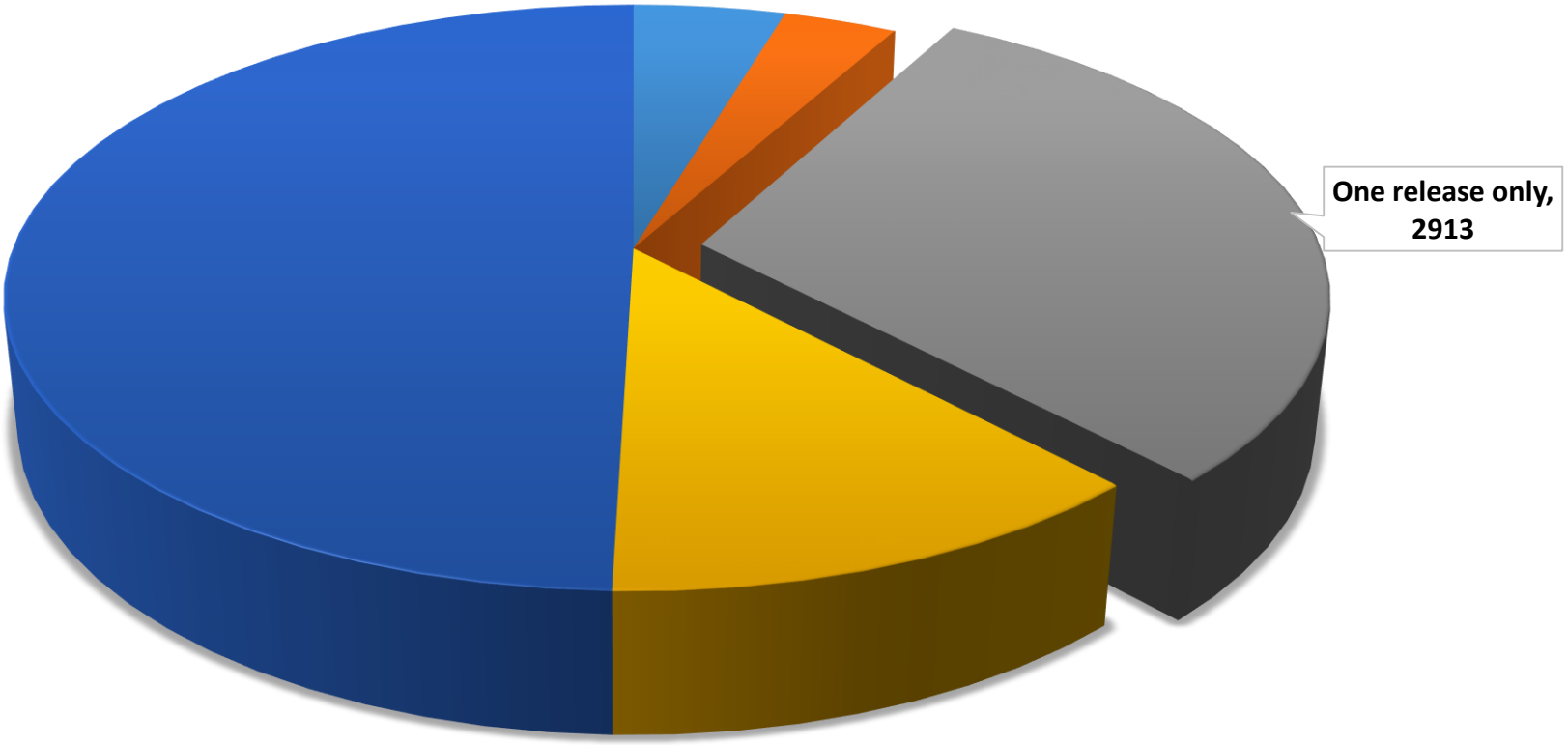
Dataset

Total number of apps' gathered data: 9703



Dataset

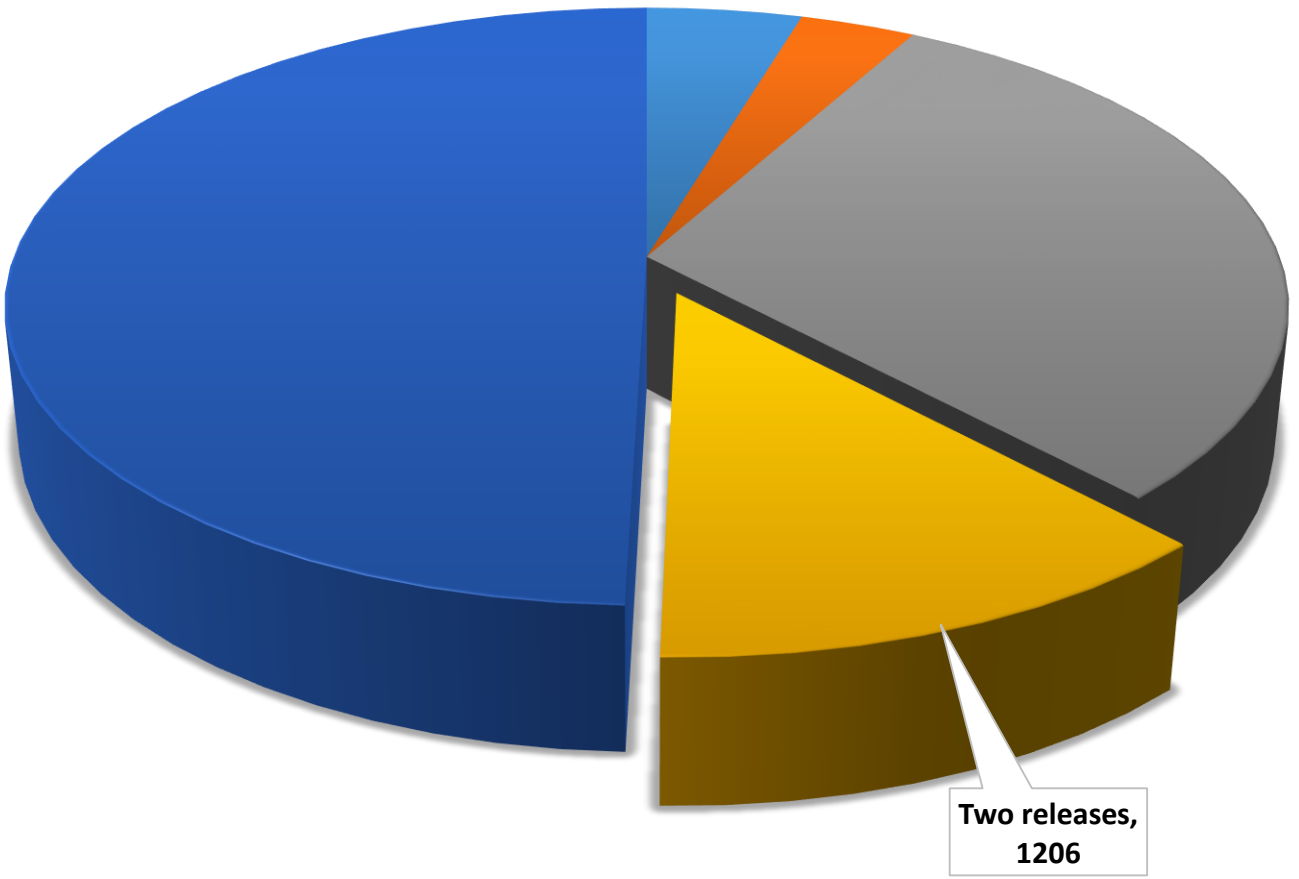
No pattern could be detected as the sequence is unavailable



Redundent Records | Incomplete Data | One release only | Two releases | The rest

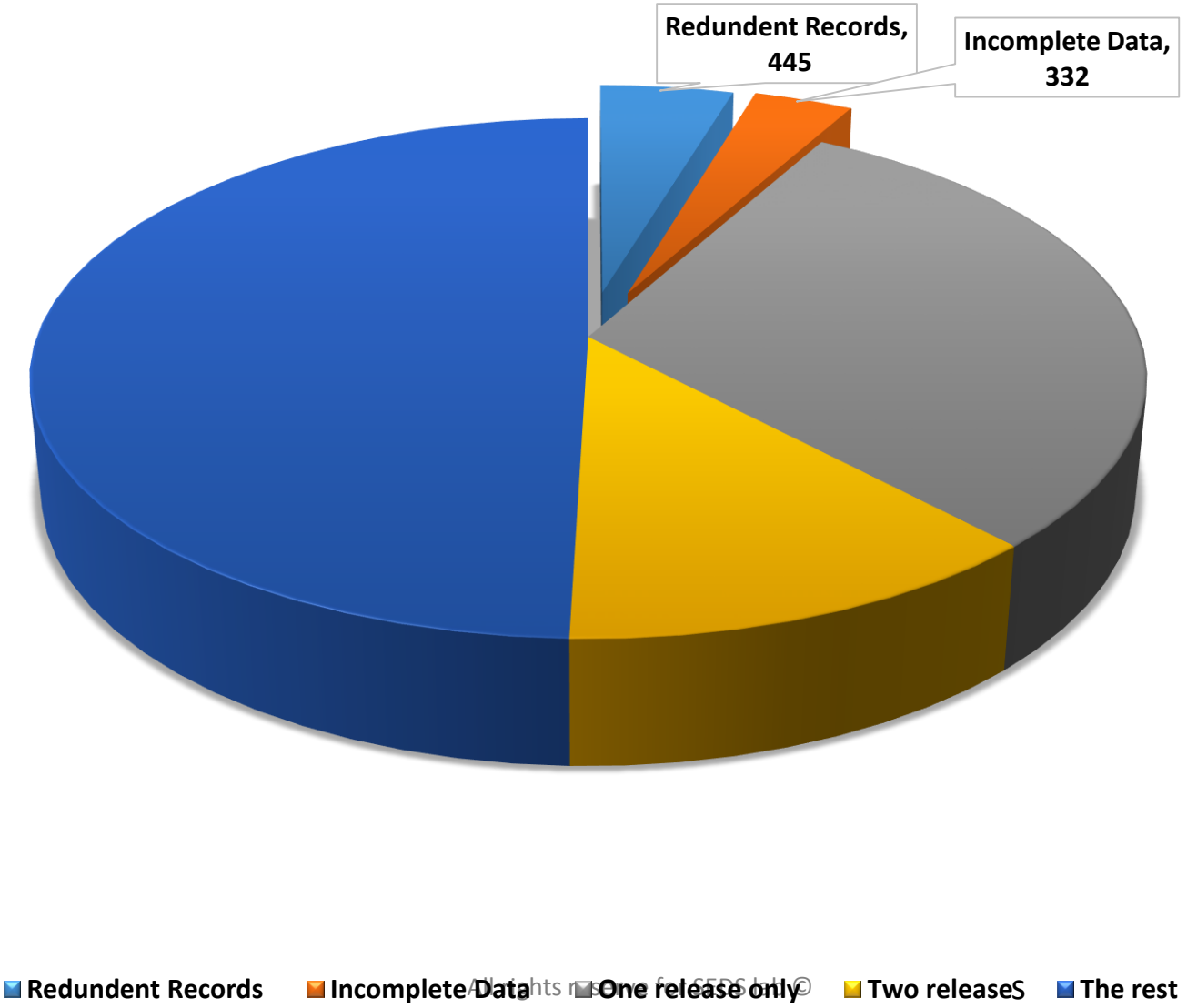
Dataset

This portion of data is not usable in variance analysis



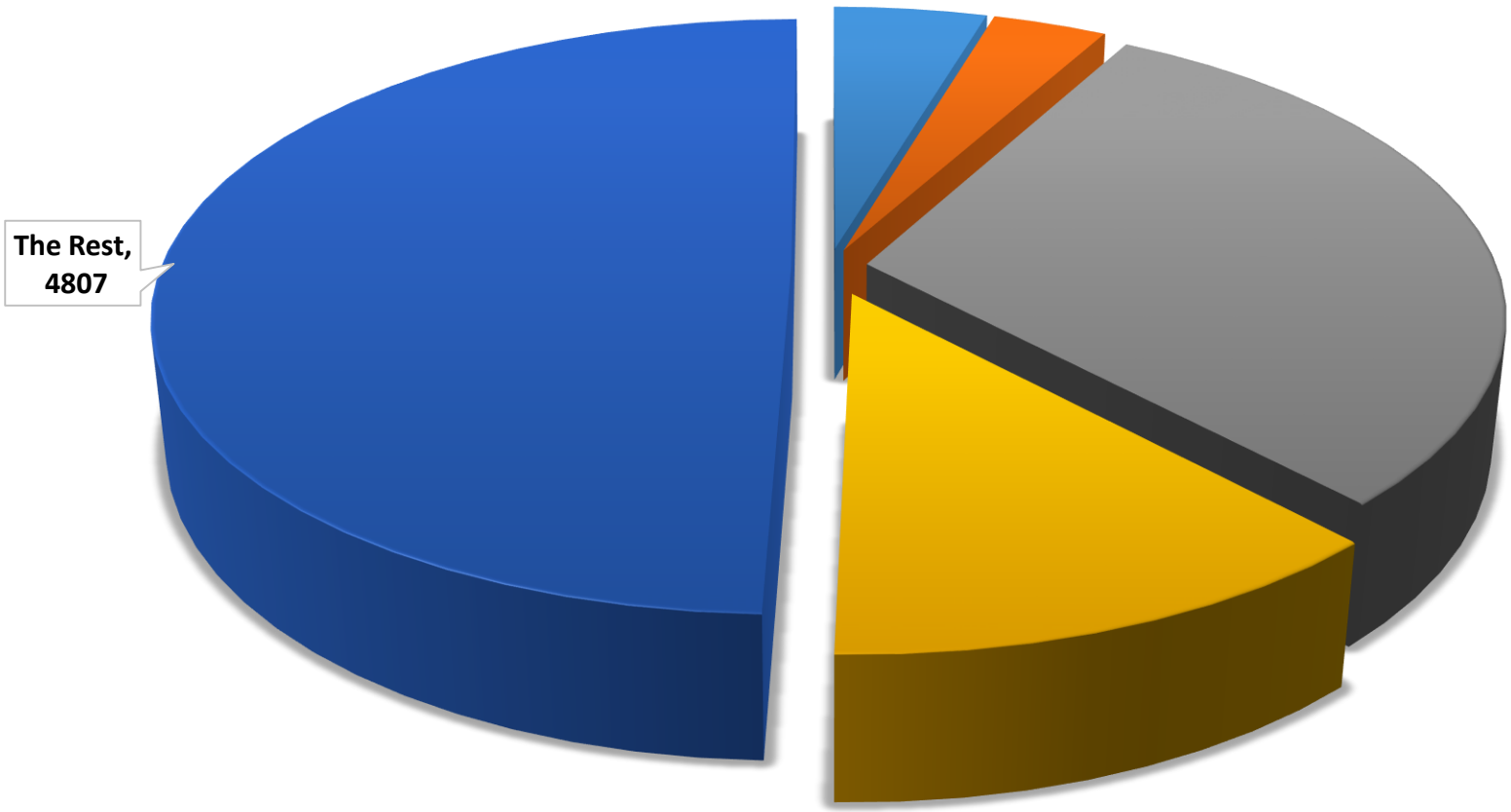
Dataset

The app is in the market but the release dates are unavailable



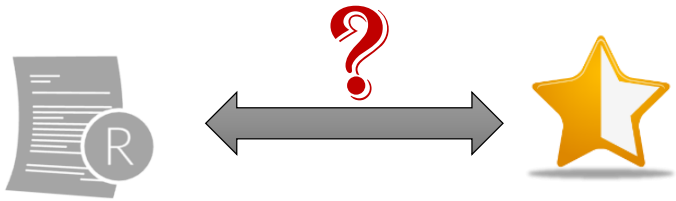
Dataset

We analyzed 6013 apps in total!



Confirmative Analysis

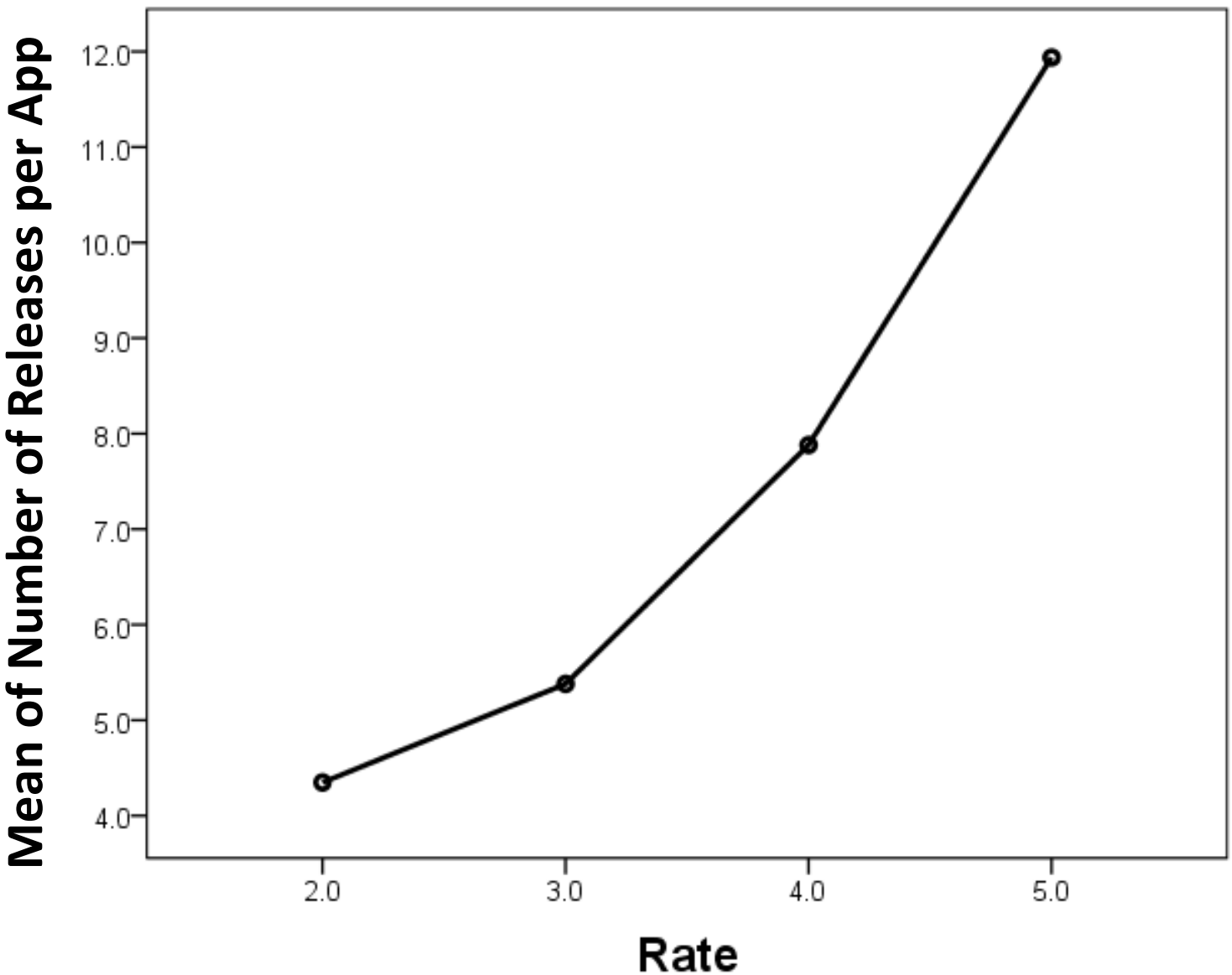
Number of Releases & Rate



H_0: There is no relation between number of releases and rate

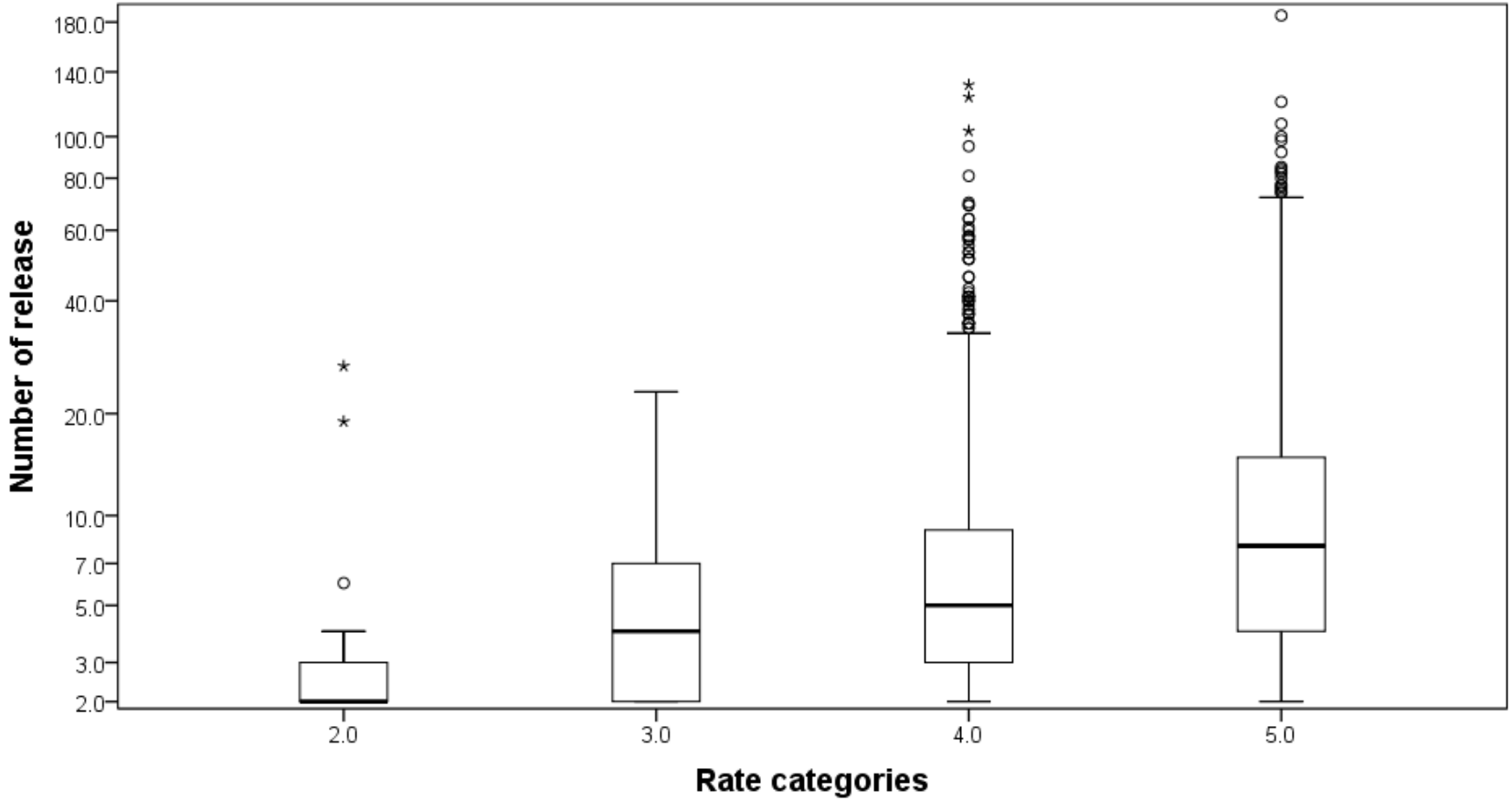
- ✓ Using ANOVA test

Number of Releases & Rate

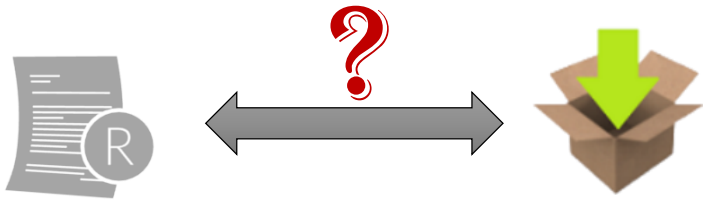


Number of Releases & Rate

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	25878.372	3	8626.124	**65.474**	**.000**
Within Groups	790224.923	5998	131.748	***Null Hypothesis is rejected!***	
Total	816103.295	6001			



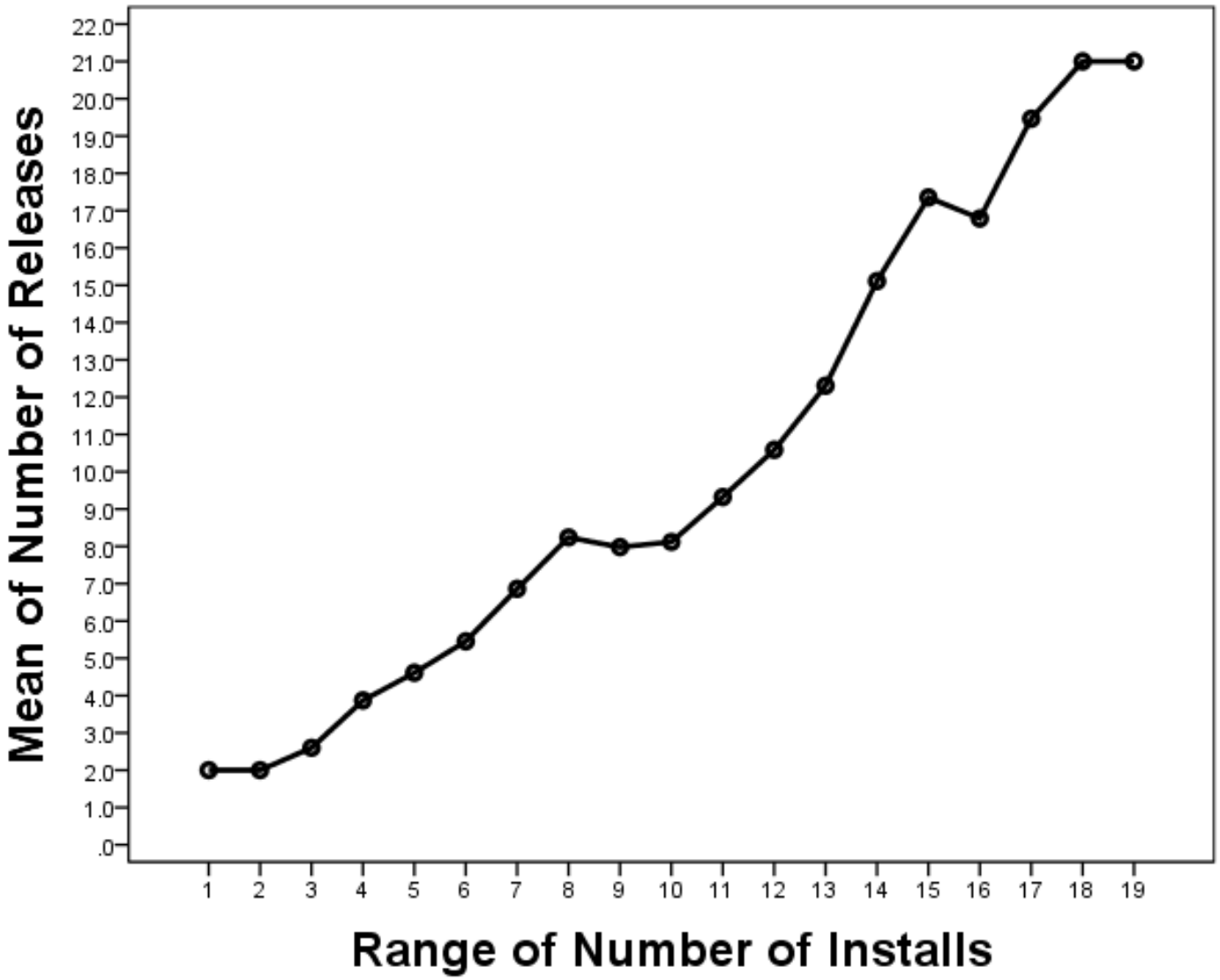
Number of Releases & Installs



H_0: There is no relation between number of releases and number of installs

- ✓ Using ANOVA test

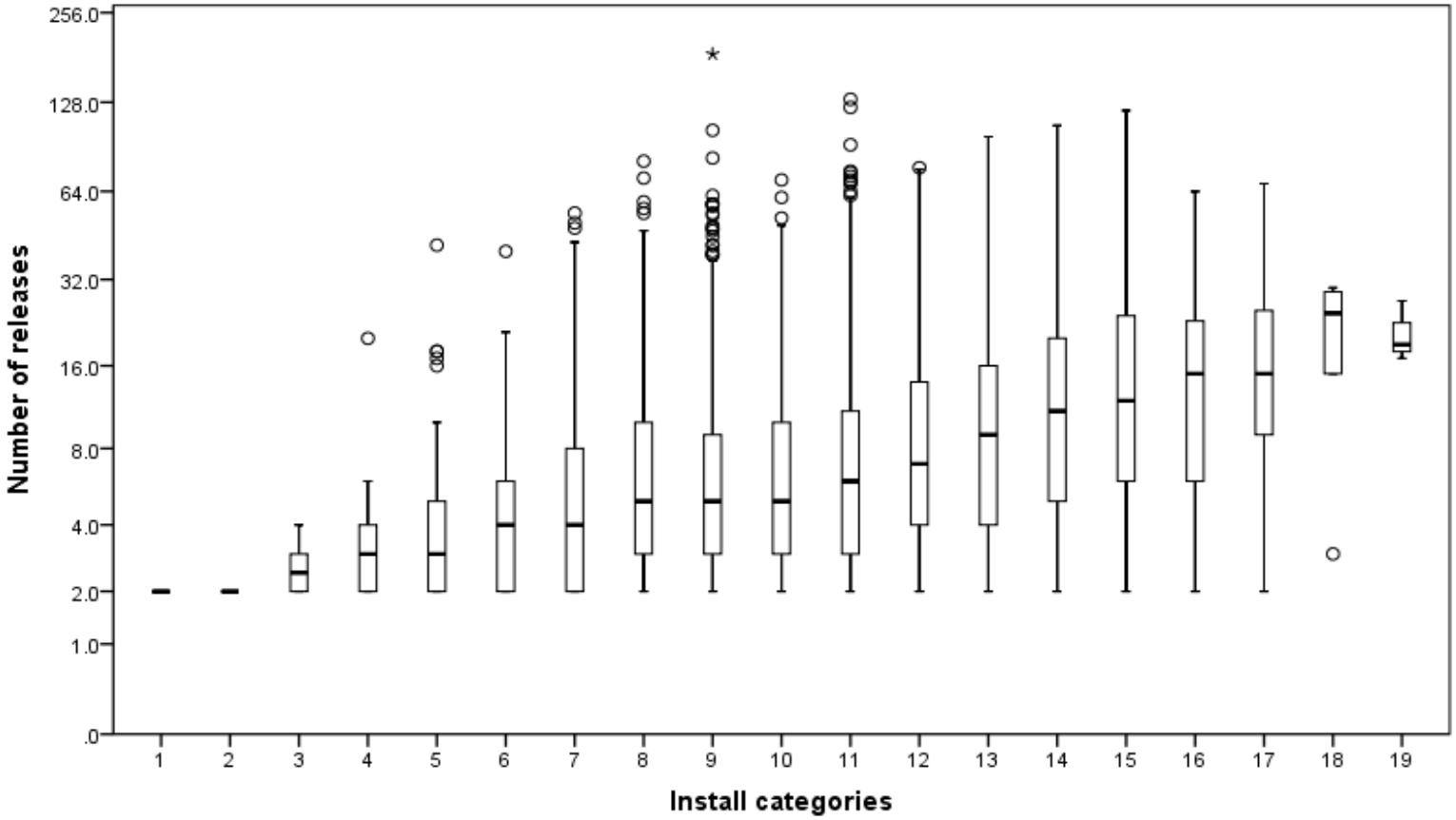
Number of Releases & Installs



Number of Releases & Installs

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	61605.596	18	3422.533	**27.140**	**.000**
Within Groups	754497.699	5983	126.107		
Total	816103.295	6001			

Null Hypothesis is rejected!



Pattern Recognition

Release Pattern Extraction

- Using K-means algorithm where *k= 3*
- Classification of release cycle time:

X: [1 82]
Y: [83 311]
Z: [312 1365]

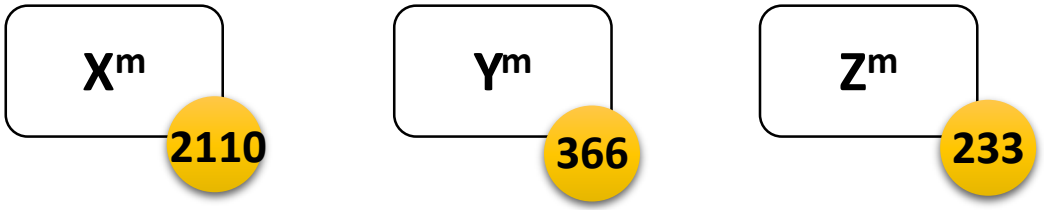
“X”, “Y” and “Z” Grammar

The language of the grammar is then the infinite set:

$$\{(X^n Y^m Z^l)^P \mid n,m,l,p \geq 0\}$$

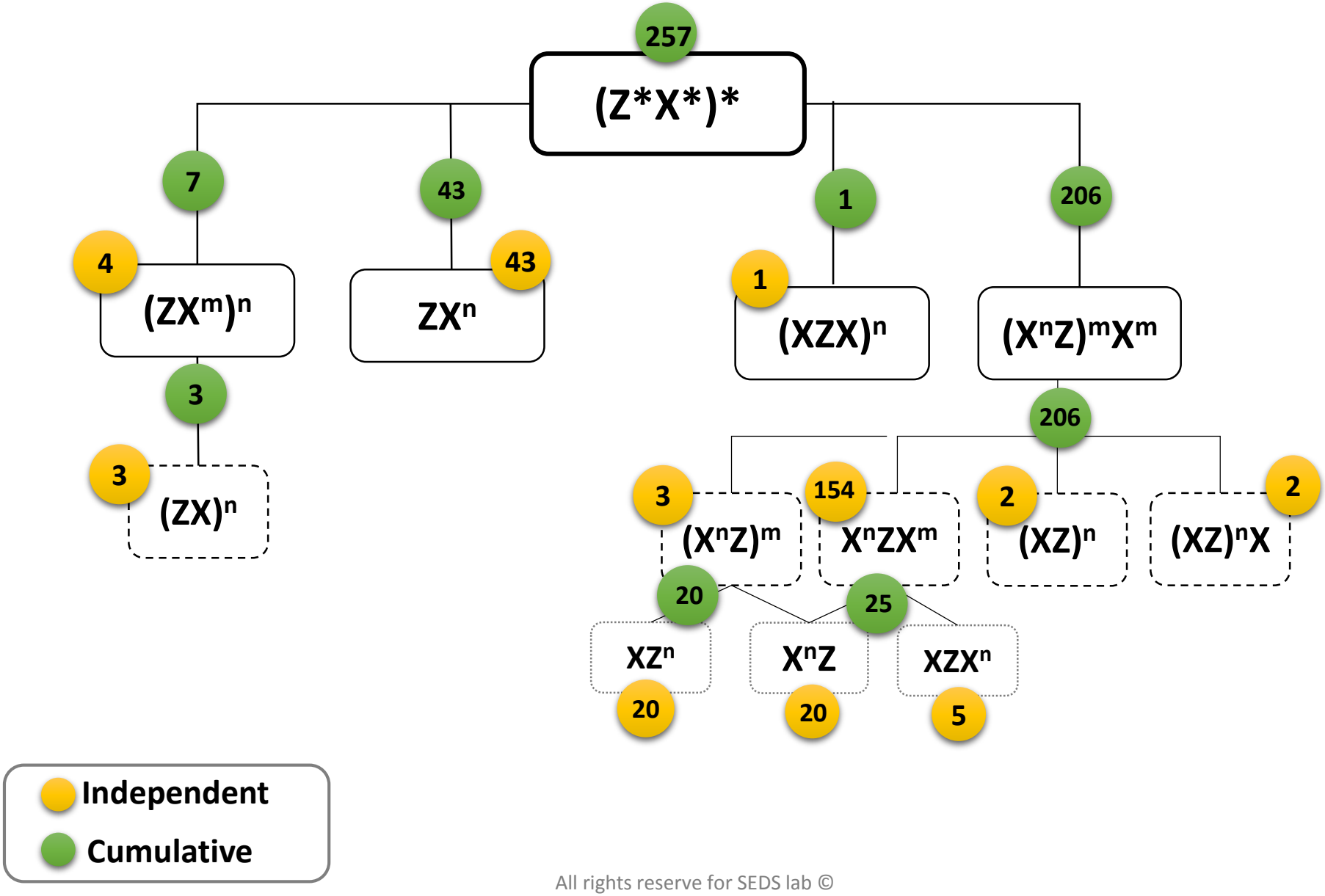
Where X^n is cycle type X repeated n times.

“X” OR “Y” OR “Z” Patterns

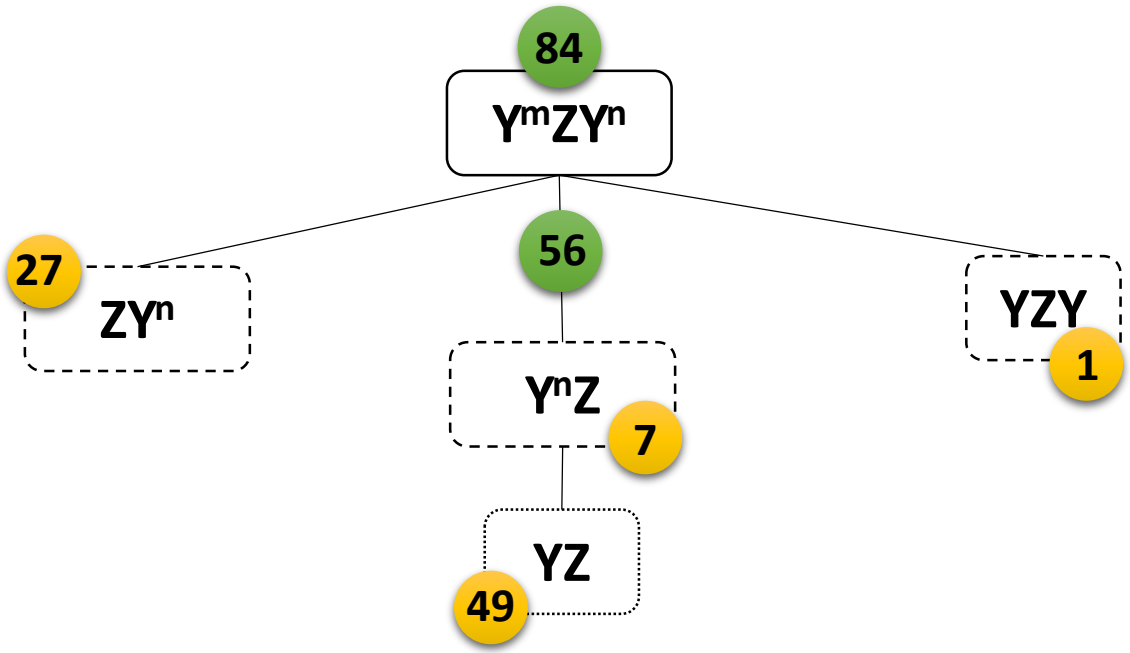


- ANOVA Test Significance
- Independent
- Cumulative

“Z” and “X” Patterns Hierarchy



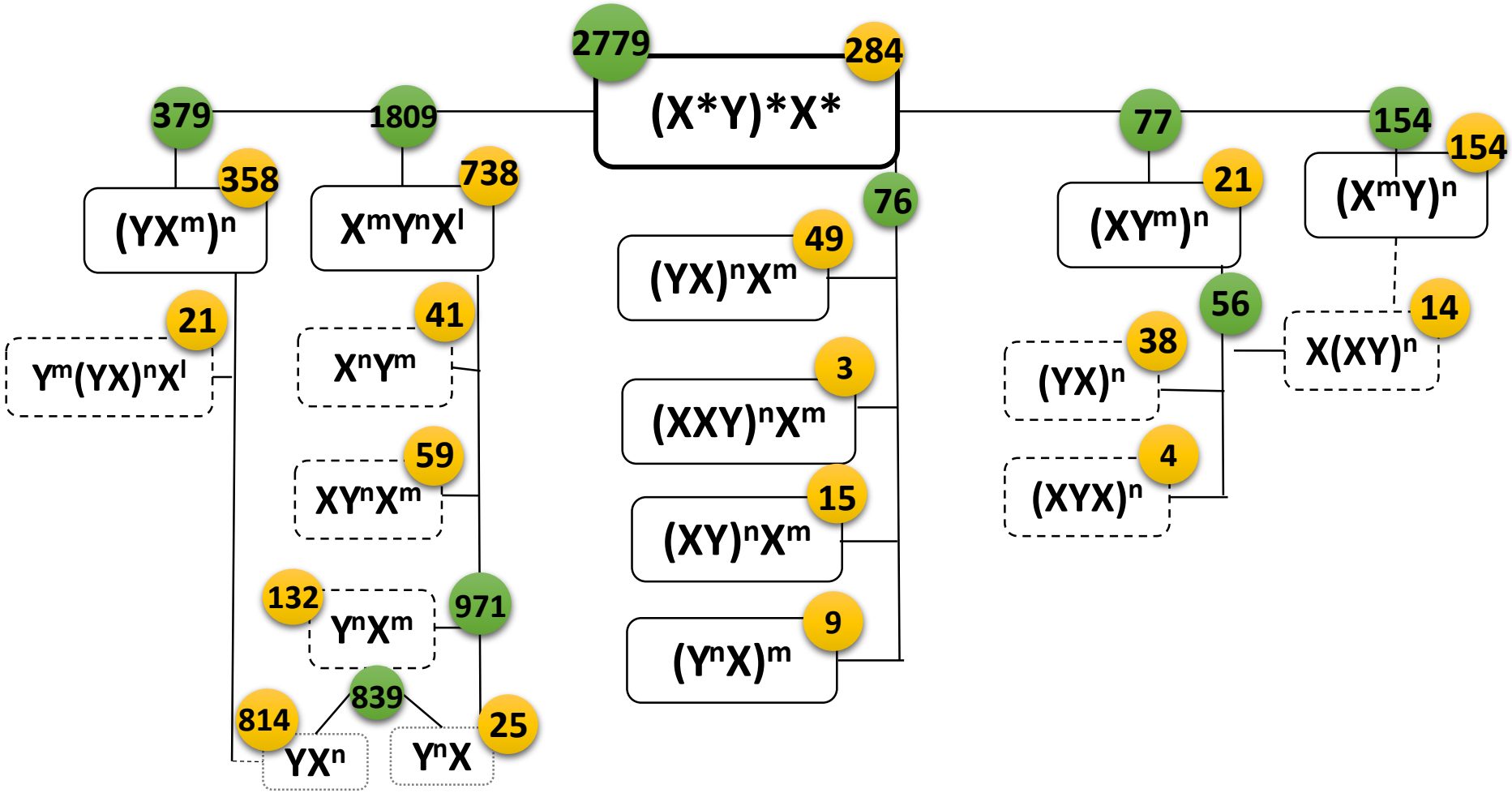
“Y” and “Z” Patterns Hierarchy



Independent

Cumulative

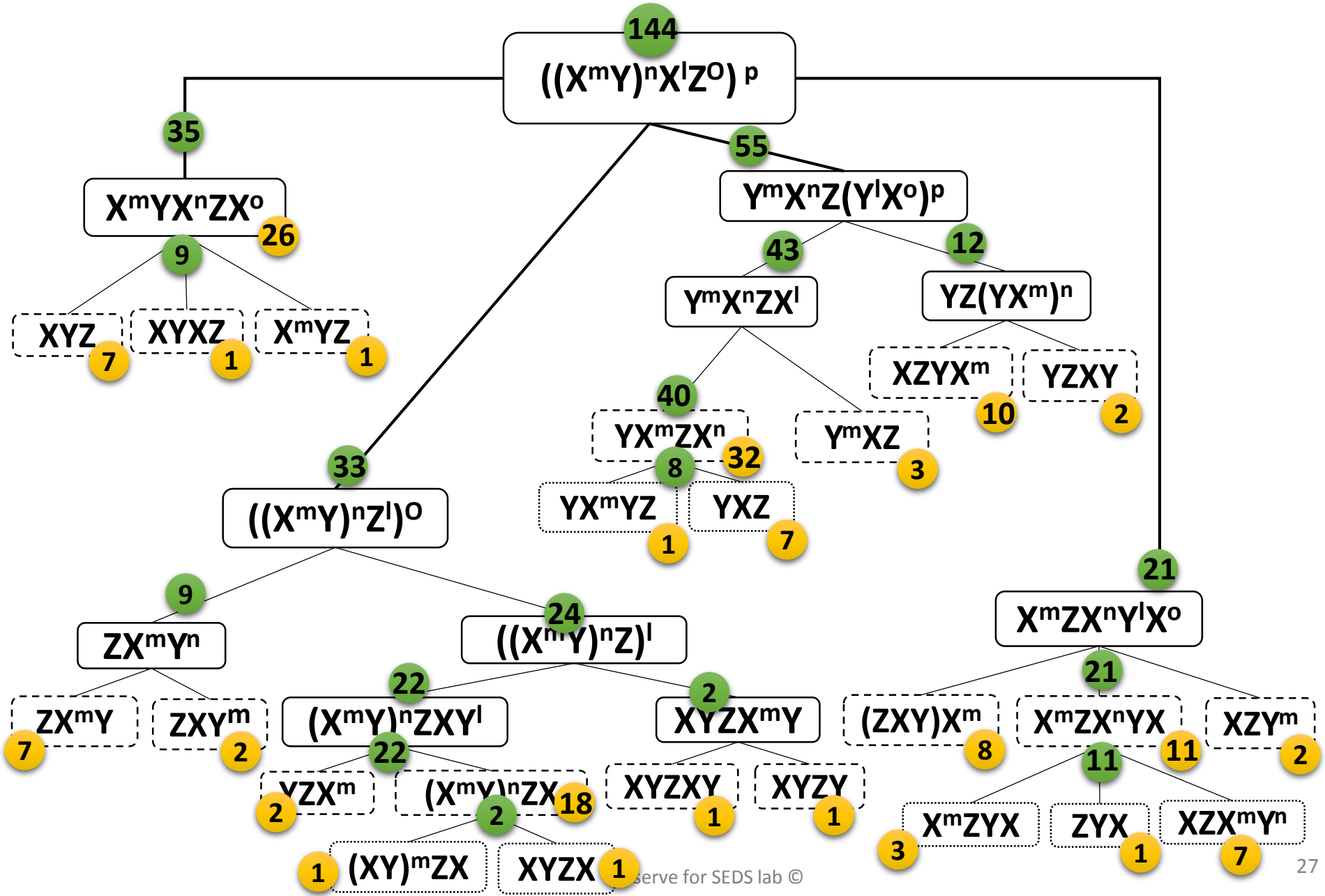
“Y” and “X” Patterns Hierarchy



Independent

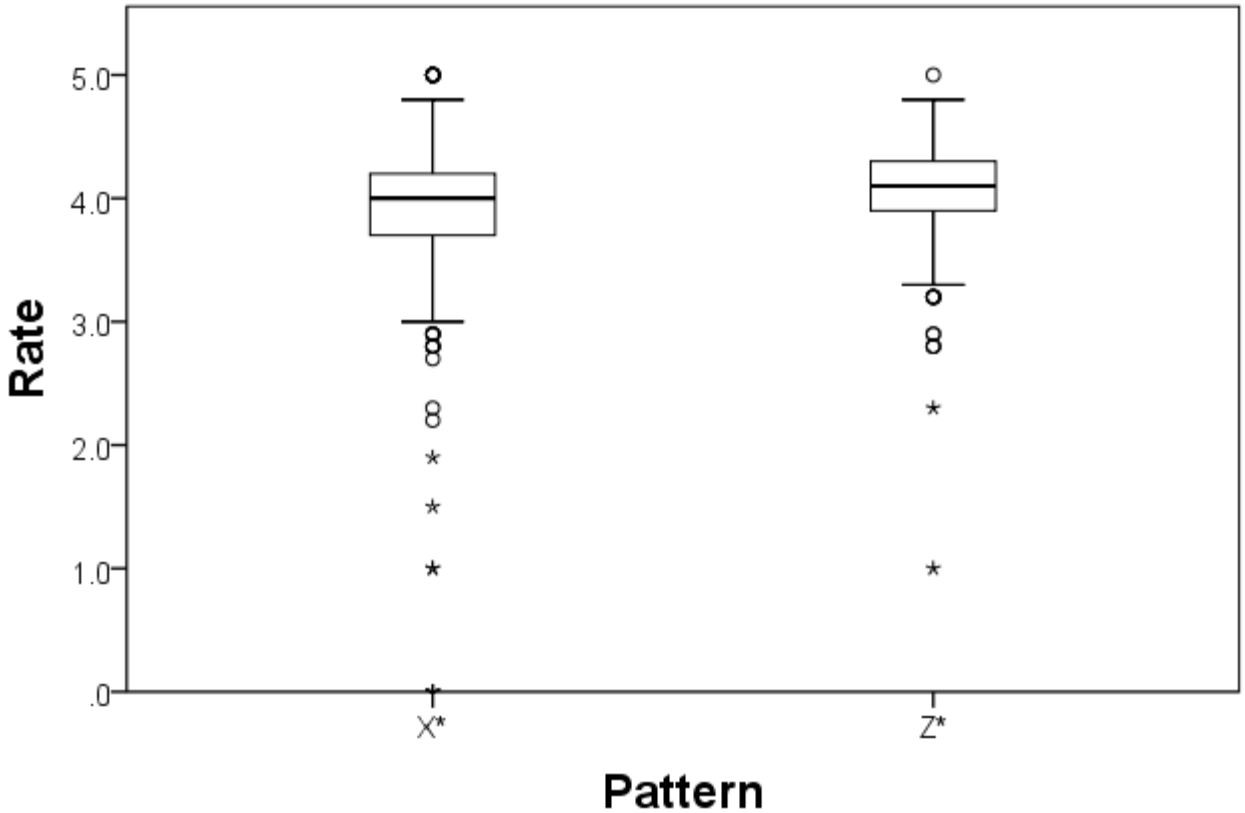
Cumulative

“Y”, “X” and “Z” Patterns Hierarchy

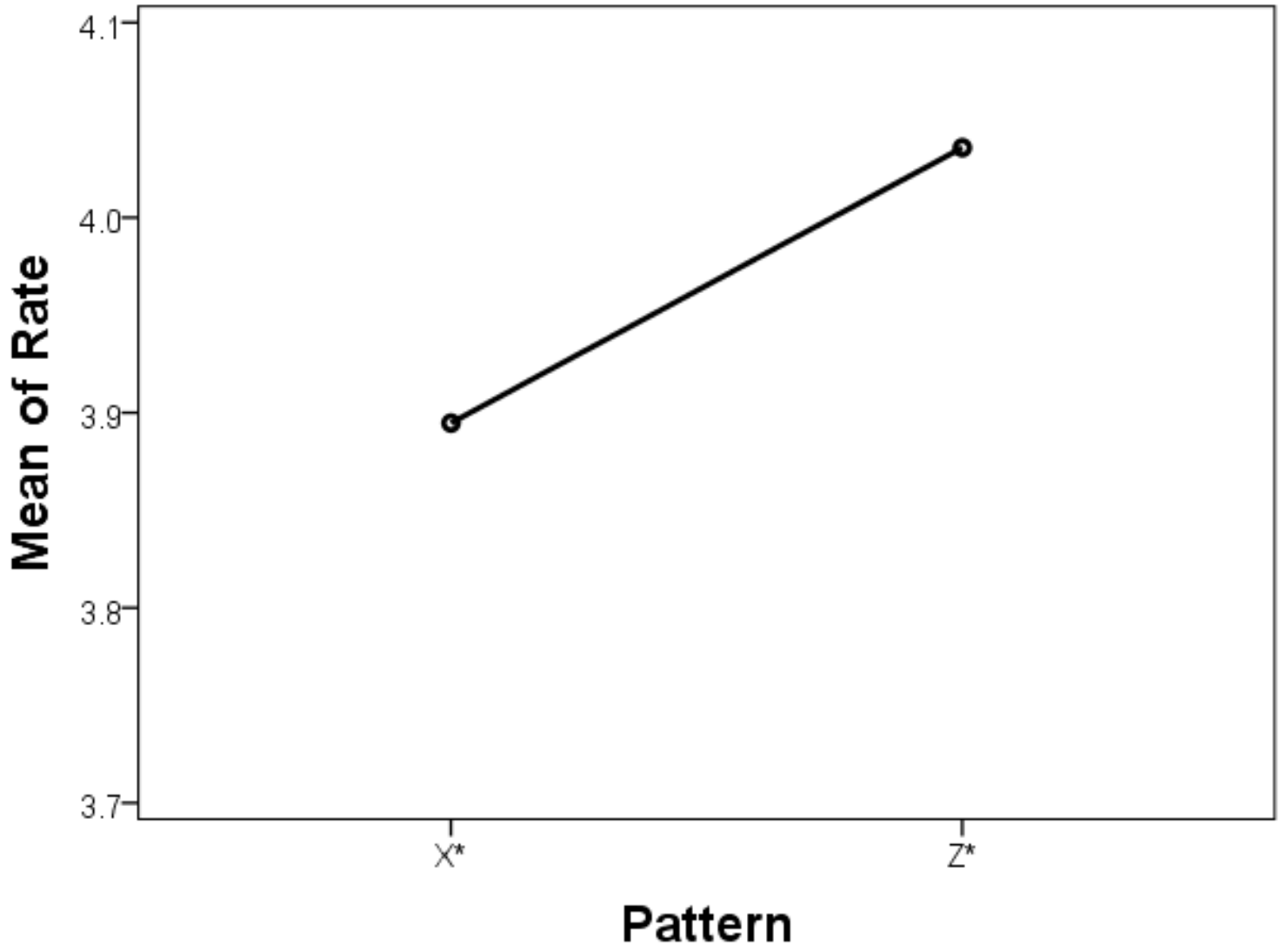


X*, Z* and Rate

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2.988	1	2.988	**8.618**	**.003**
Within Groups	229.904	663	.347		
Total	232.893	664			

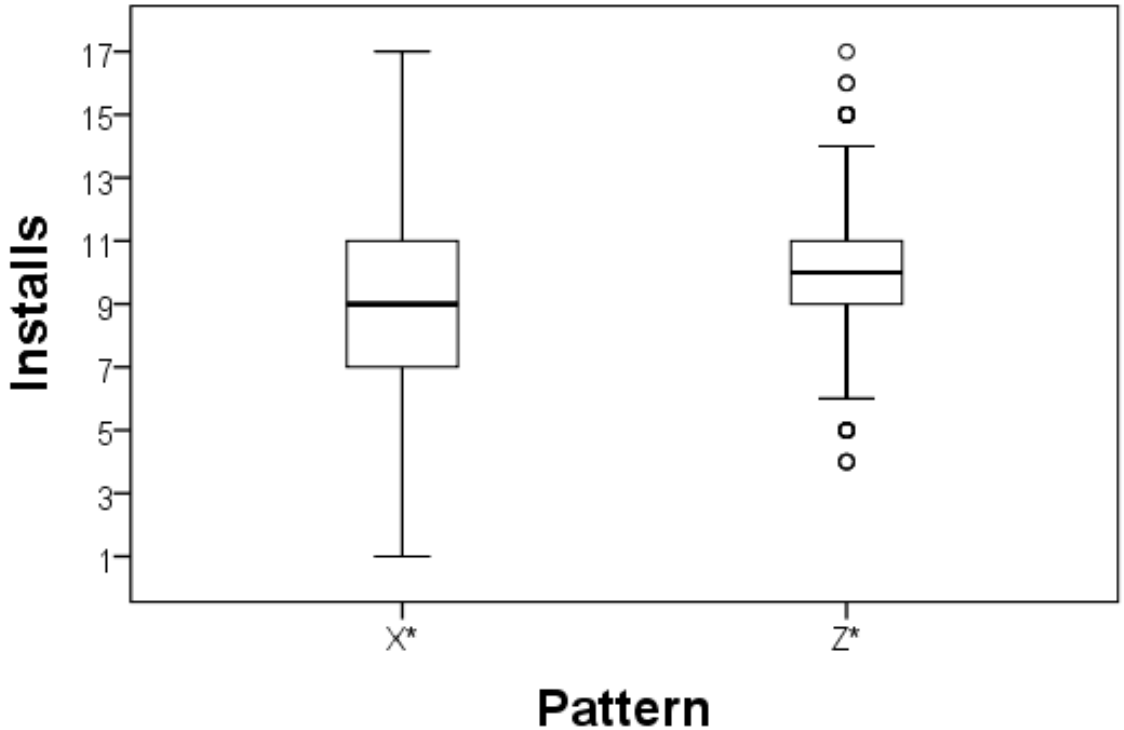


X*, Z* and Rate

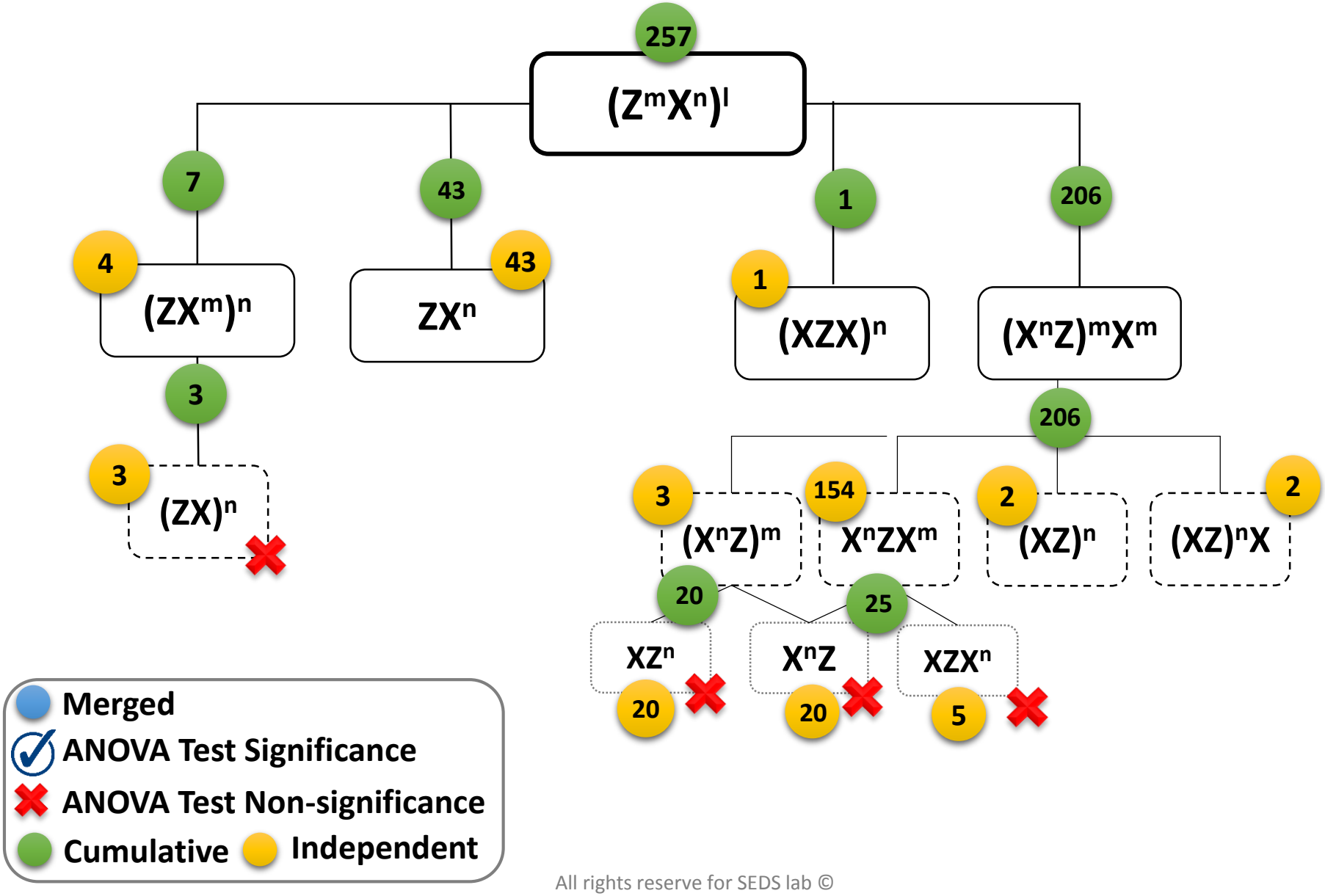


X*, Z* and Number of Installs

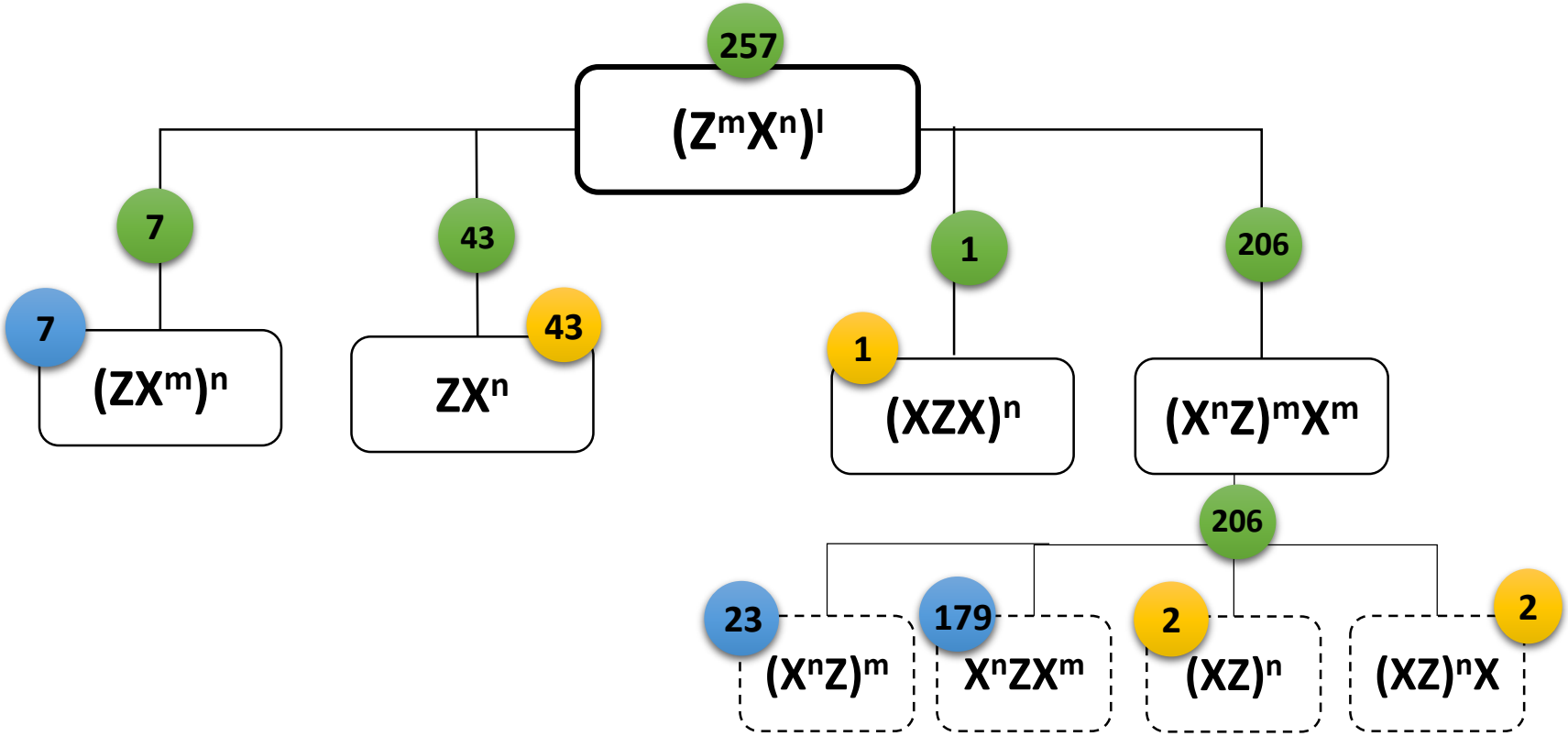
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	81.014	1	81.014	12.019	.001
Within Groups	4468.980	663	6.741		
Total	4549.994	664			



“Z” and “X” Merged Patterns

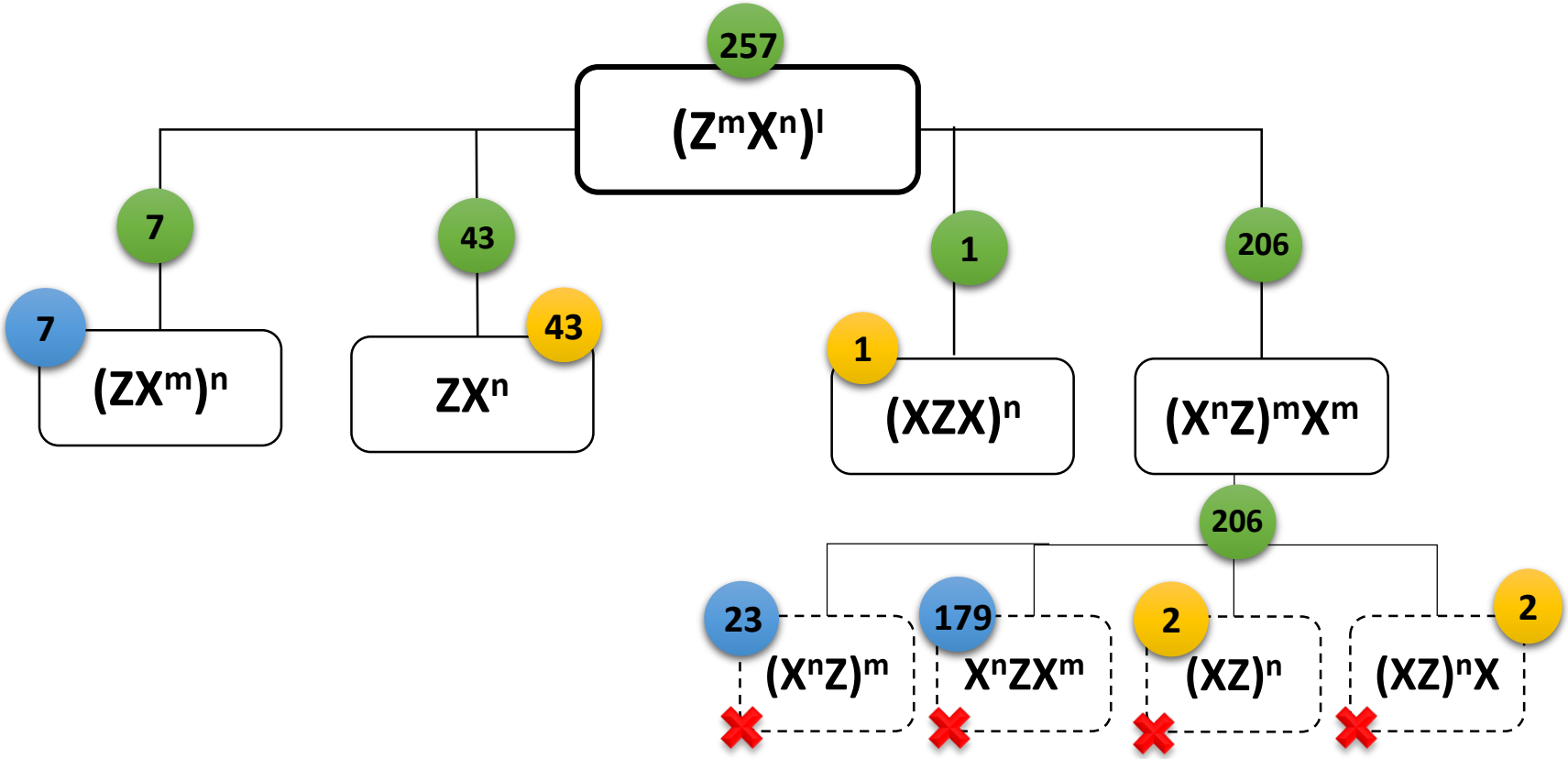


“Z” and “X” Merged Patterns



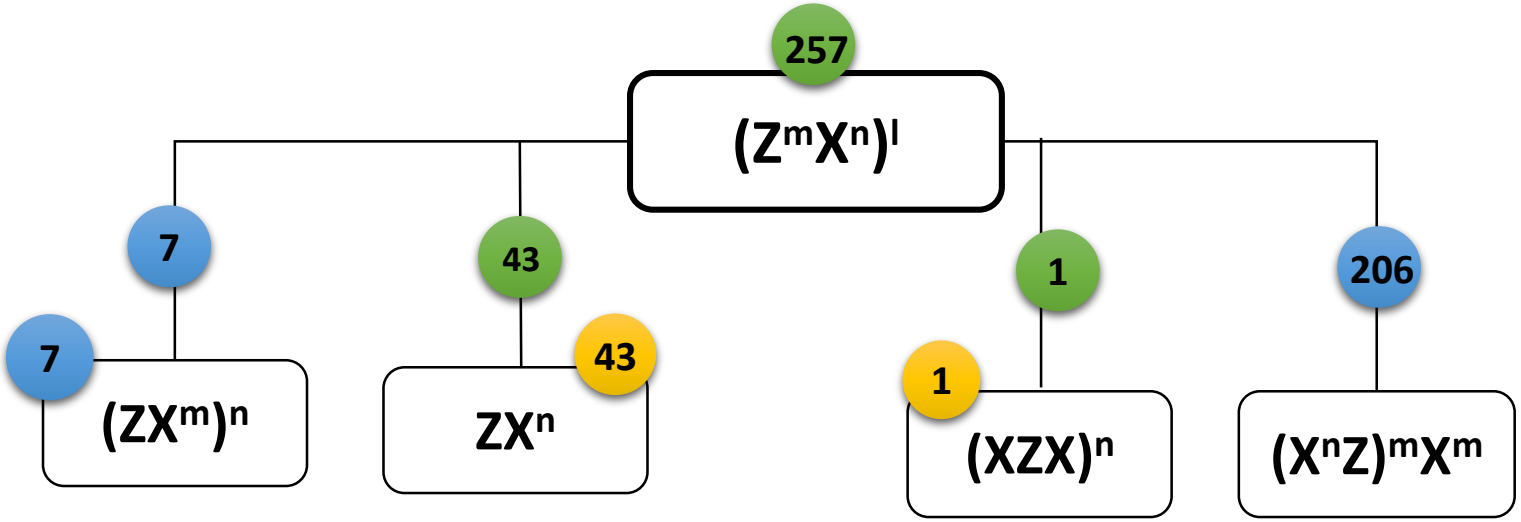
- Merged
- ANOVA Test Significance
- ANOVA Test Non-significance
- Cumulative
- Independent

“Z” and “X” Merged Patterns



- Merged
- ANOVA Test Significance
- ANOVA Test Non-significance
- Cumulative
- Independent

“Z” and “X” Merged Patterns



Merged
ANOVA Test Significance
ANOVA Test Non-significance
Cumulative
Independent

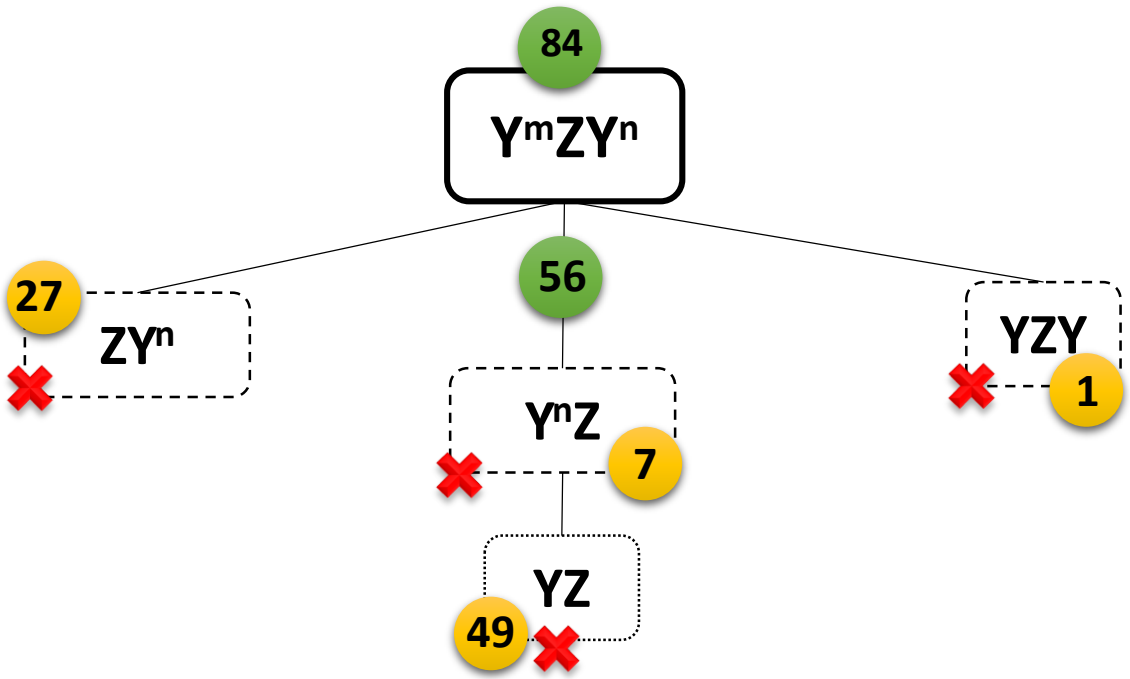
“Z” and “X” Merged Patterns

257

$(Z^m X^n)^l$

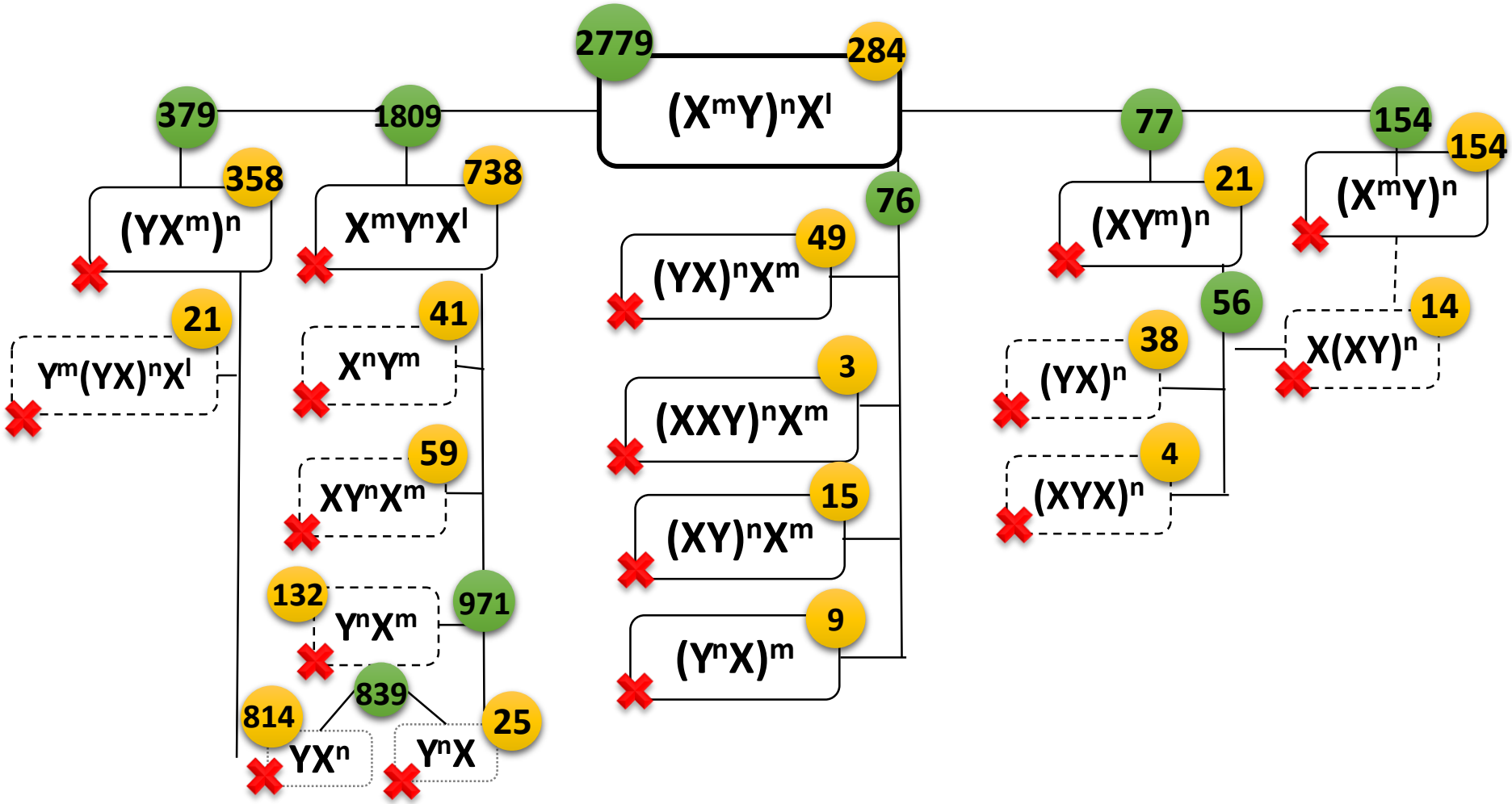
Merged
ANOVA Test Significance
ANOVA Test Non-significance
Cumulative
Independent

“Y” and “Z” Patterns Hierarchy



- Merged
- ANOVA Test Significance
- ANOVA Test Non-significance
- Cumulative
- Independent

“Y” and “X” Patterns Hierarchy

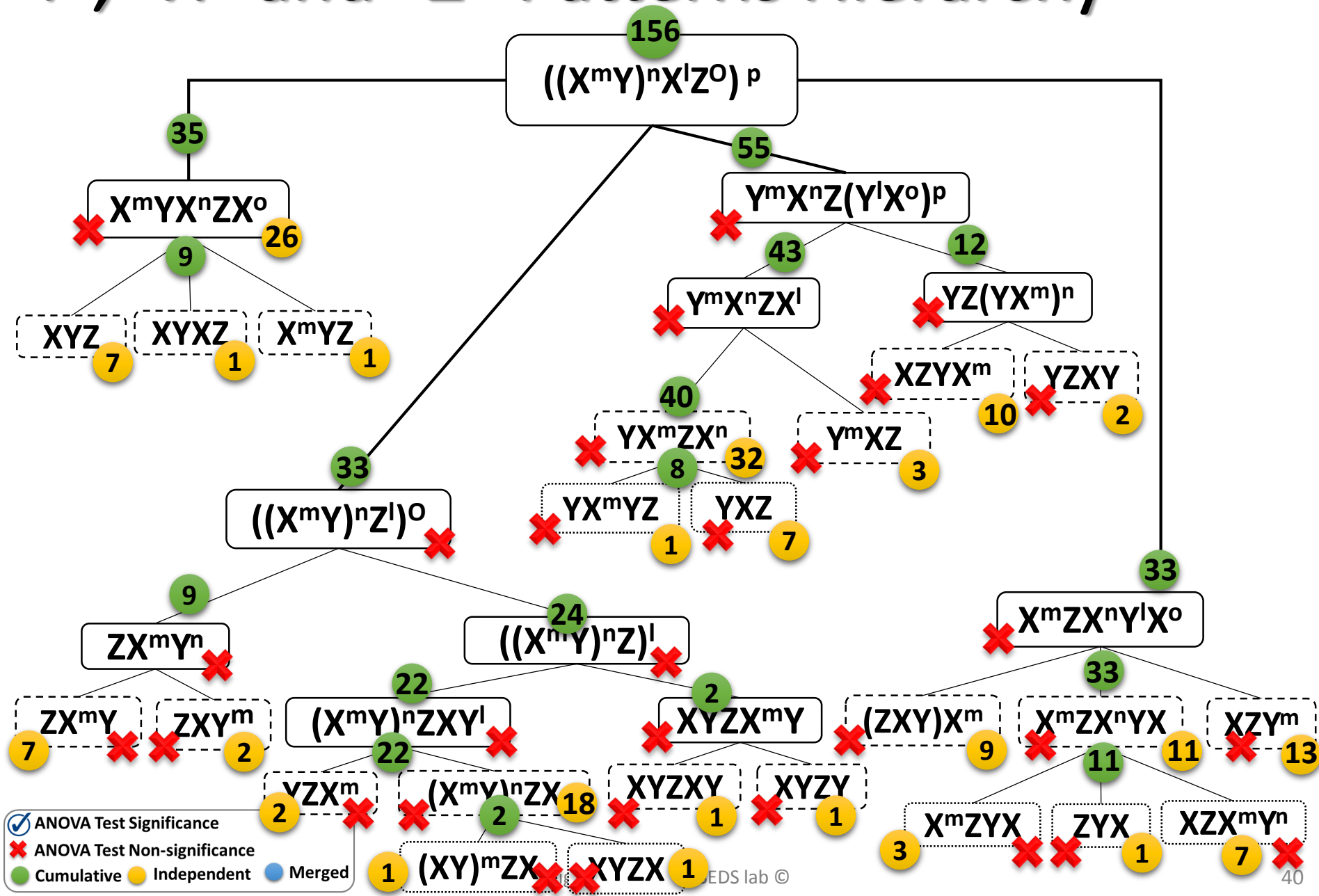


ANOVA Test Significance

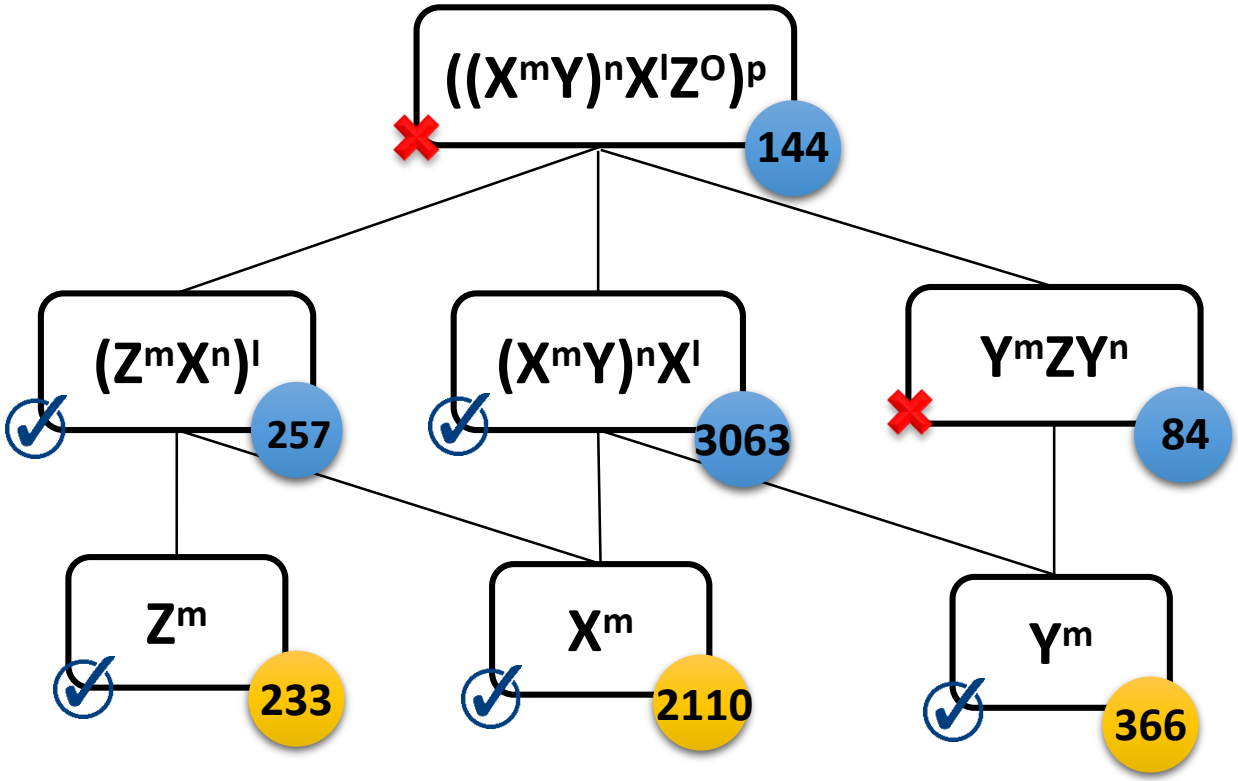
ANOVA Test Non-significance

Cumulative Independent Merged

“Y”, “X” and “Z” Patterns Hierarchy

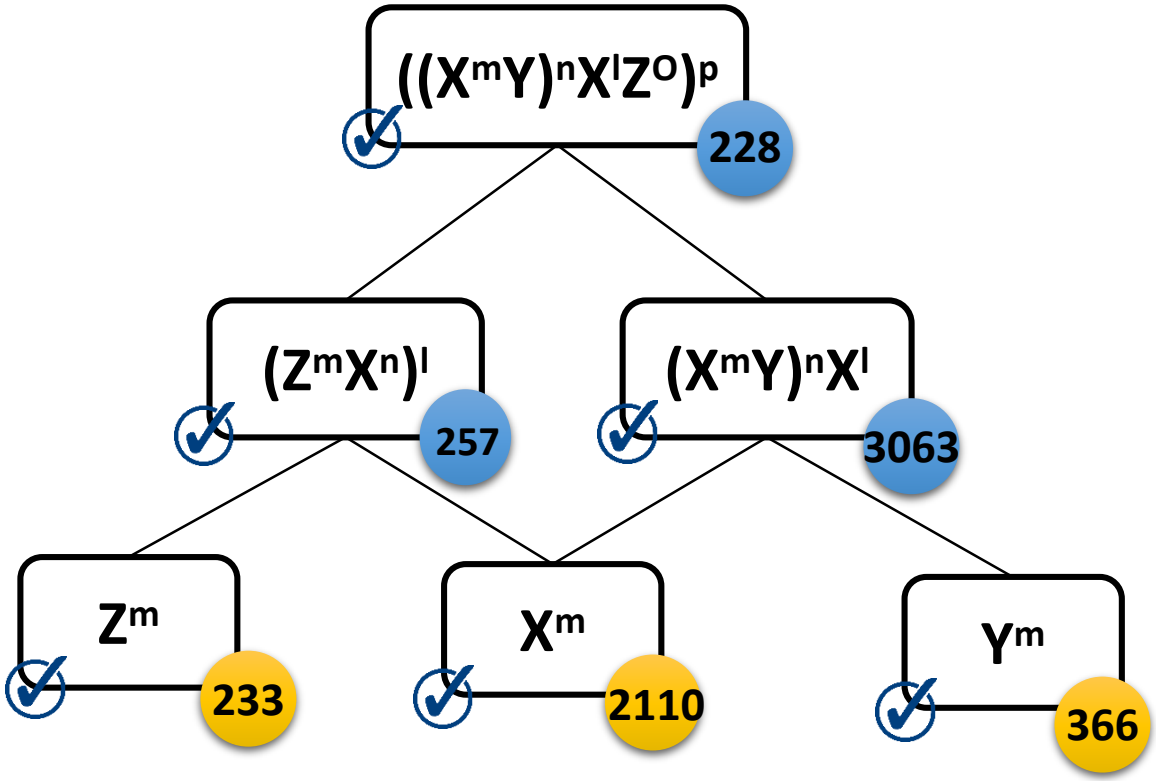


Integrated and Merged Trees



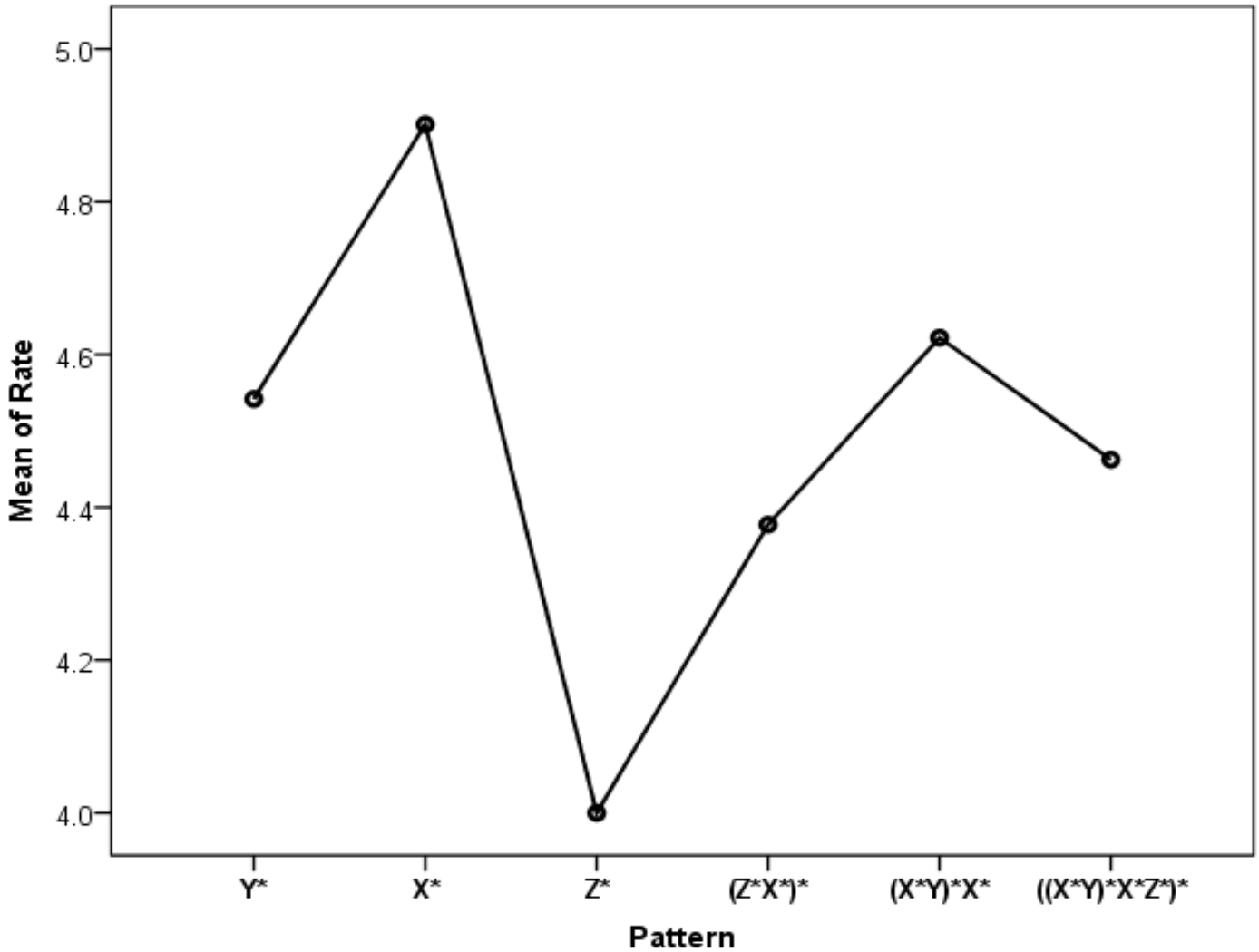
Merged
ANOVA Test Significance
ANOVA Test Non-significance
Cumulative
Independent

Integrated and Merged Trees



- Merged
- ANOVA Test Significance
- ANOVA Test Non-significance
- Cumulative
- Independent

Mean of Rate between Patterns



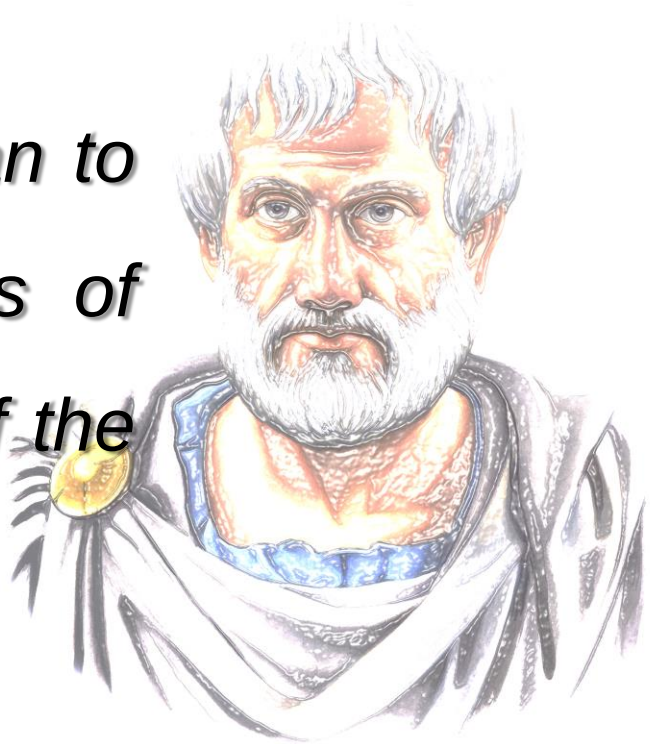
Synthesis & Explanation of Results

Rough set analysis

Rough Set Analysis (RSA)

- Rough set theory created by Pawlak (1991)
- Intended to approach inherent uncertainty
- Creates non-deterministic explanation rules
- RSA is applicable also for nominal or ordinal data

It is the mark of an educated man to look for precision in each class of things just so far as the nature of the subject admits.



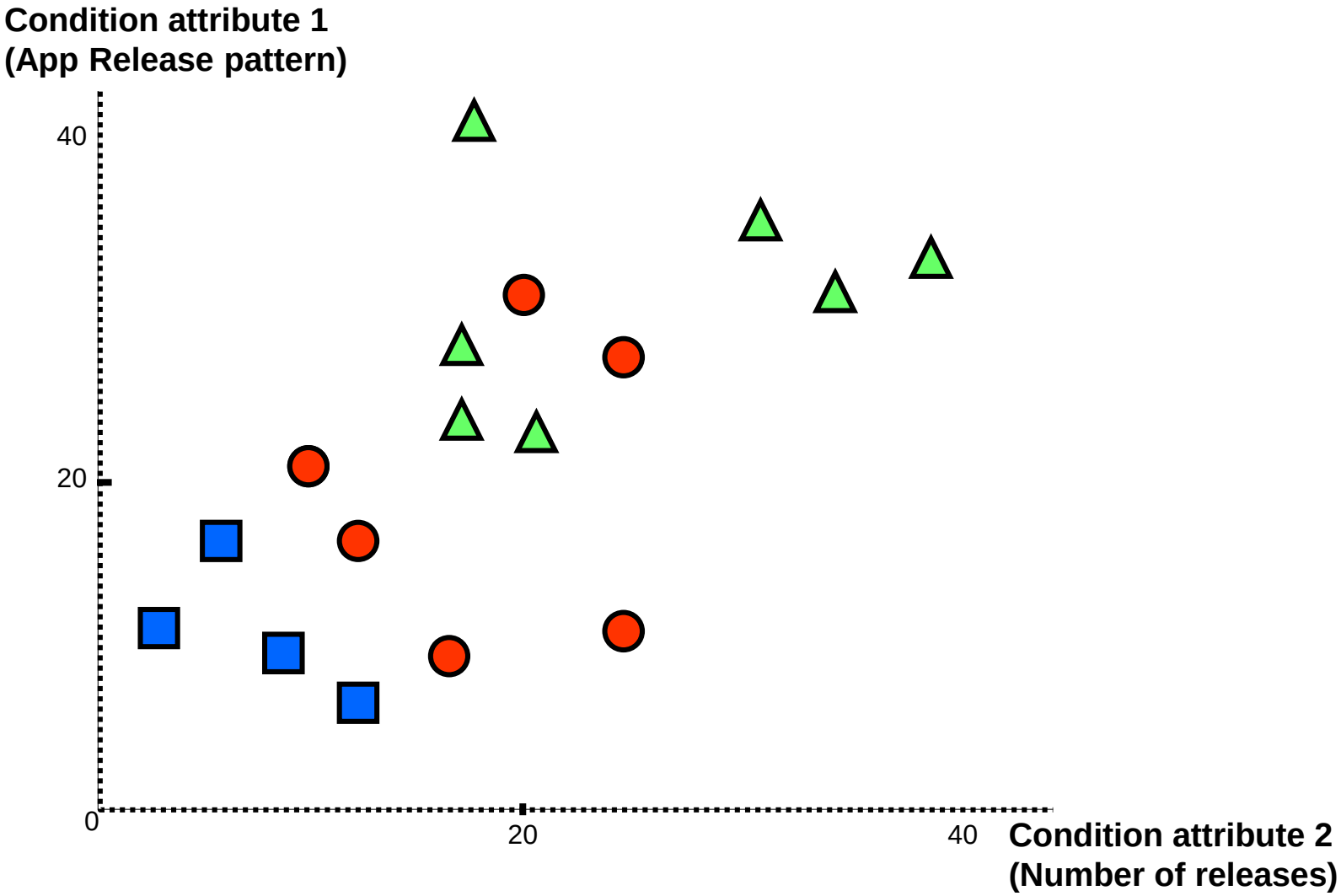
Aristotle. 325 BC

Rough Set Analysis (RSA)

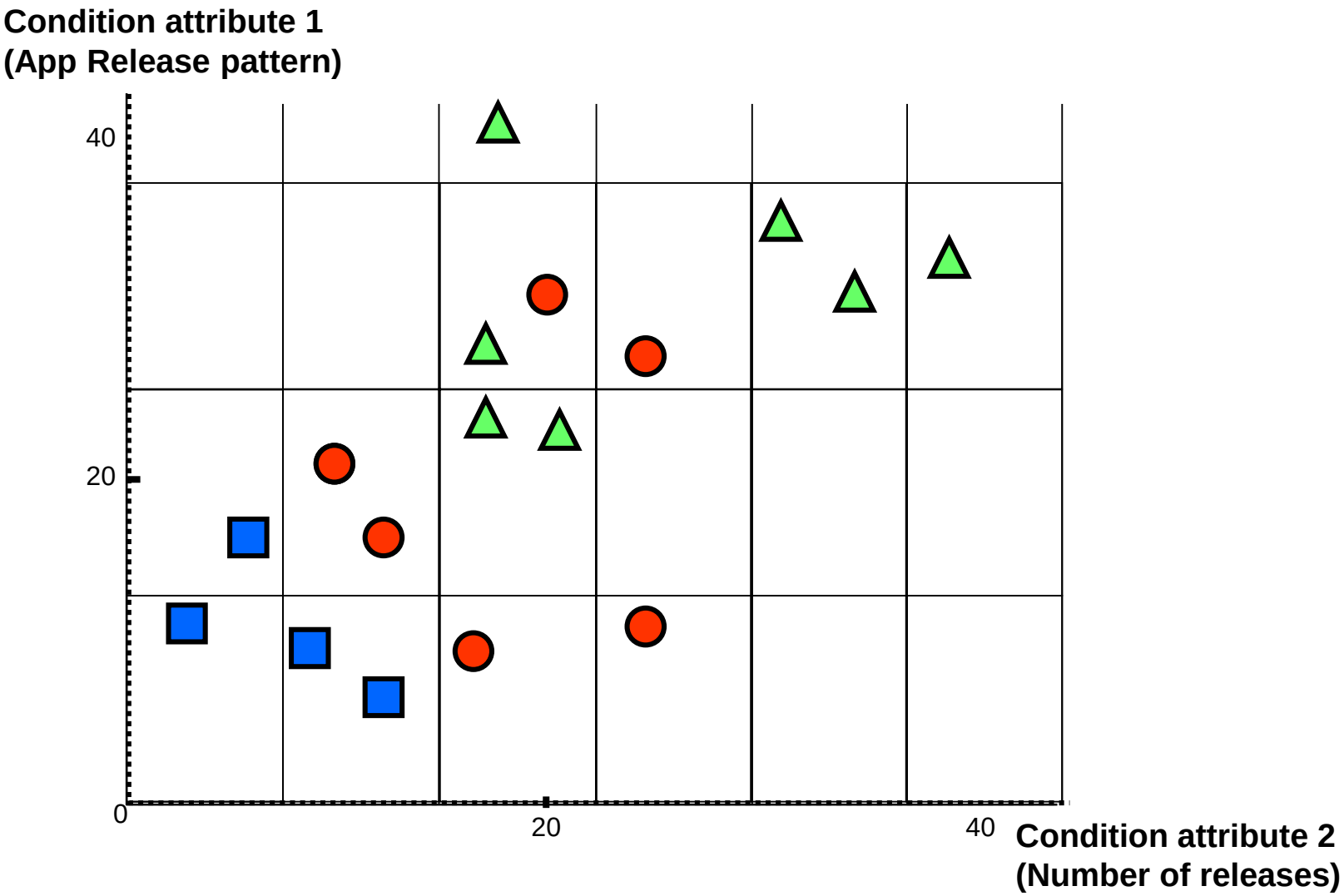
A rough set is a set of objects which, in general, cannot be precisely characterized in terms of the values of the set of attributes while a pair of a lower and an upper approximation of the collection can do.

The rough set methodology is based on the premise that **lowering the degree of precision in the data** makes the data pattern more visible.

Objects in Condition Attribute Space



Indiscernible Sets



Attributes in RSA

- **Core**

Cannot be removed from consideration without deteriorating the quality of approximation

Core Attributes in RSA

- Pattern 
- Frequency of short, medium and long releases 
- Cost 
- Number of installs 
- Number of reviewers 
- Release cycle mean 

- Release cycle variance 

Release Date Sample Rules by RSA

IF Mean is short ➡ is very high

Strengths: 70.21%

IF # of short release Is very high AND is very low ➡ # of is very high

Strength: 75.18%

IF # of short release Is low AND Mean is short AND Variance is NOT low ➡ # of is NOT very high

Strengths: 100%

Release cycle | Rate | Download | Cost

Release Date Sample Rules by RSA

IF is less than 2$ AND # of is small AND # of is low AND is short AND is very low

Mean

Variance

➡ # of Is NOT very high

Strengths: 72.2%

IF is less than 2$ AND # of is very high AND # of is very high AND is very low ➡ # of is very high

Variance

Strengths: 72.18%

Release cycle Download Cost Release

Release Date Sample Rules by RSA

IF # of short is medium AND release is medium

➡ is very high

Strengths: 21.02%

IF = X^m AND # of is low AND # of is low

➡ # of is NOT high

Strength: 90%

IF = $(X^mY)^nX^l$ AND Variance is NOT very high ➡

# of is high

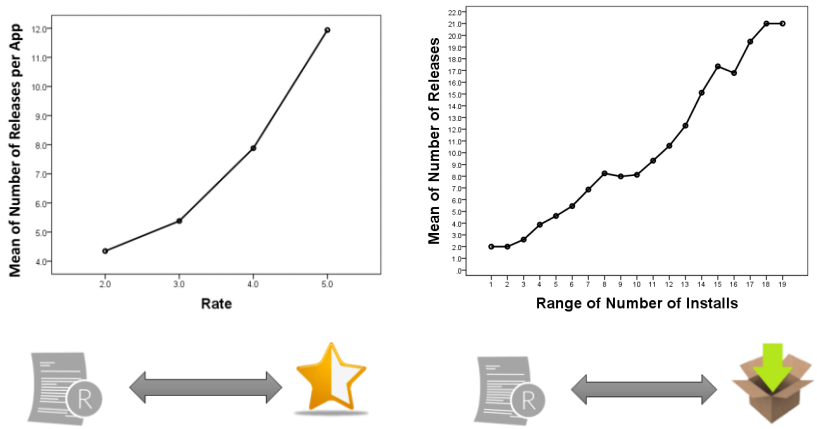
Strength: 38.9%

IF = Y^m AND # of is low ➡ # of is very low

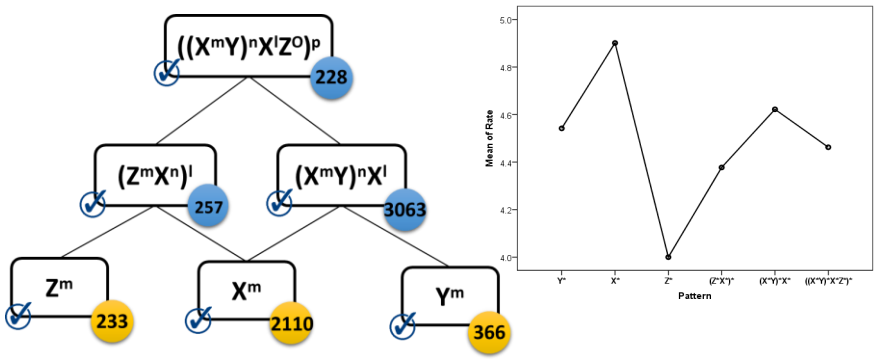
Strength: 28.57%

Release cycle | Rate | Download | Pattern | Release

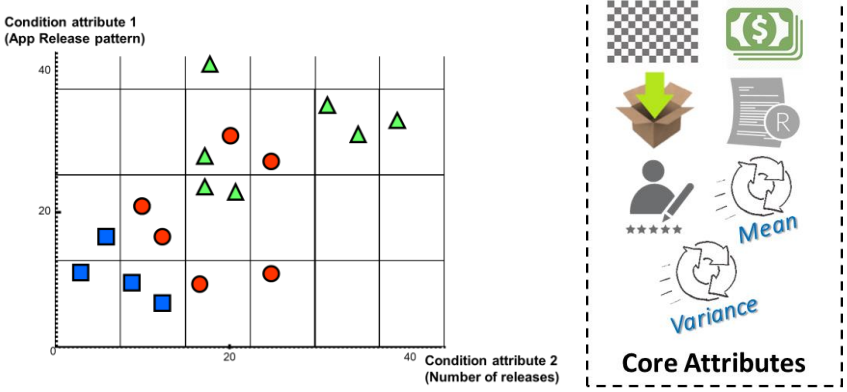
Summary



Confirmative Analysis



Pattern Recognition



Rough Set Analysis

Rule Extraction

Future Work

- Cross-validation of classification rules
- Study of release **cycle time** patterns PLUS **release type** patterns
- Varying classification parameters (interval length for "Small" releases)
- Extend analysis to further attributes
- RSA → Dominance-based RSA