UNCOVERING RETAIL CUSTOMER SEGMENTATION FROM LARGE TRANSACTION RECORDS: A NUANCED COMPARISON OF CLUSTERING ALGORITHMS USING ROUGH SET REDUCED DATASET

Final Report

by

SYED AHMAD ZAKI BIN SYED SAKAF AL-ATTAS

In Partial Fulfillment
of the Requirements for the Capstone Project Module in the
Master in IT in Business (MITB) Programme at the
Singapore Management University

Supervisor: Prof Kam Tin Seong

22nd December 2022

Abstract

This project explores the use of Rough Set's reduct algorithm on synthesizing a dataset down to its most significant variables, and its impact on clustering outcomes. This capstone project also explores and compares the use of non-traditional clustering algorithms such as fuzzy *k*-means and Gaussian mixture model, against the more traditional *k*-means clustering. These algorithms would be tested on a retail transaction dataset as an actual real-world application. These algorithms would also be executed primarily by R Code within a SAS Enterprise Miner 14.1 environment. Findings indicate positive contributions and clearer clustering outcomes, from the use of non-traditional algorithms such as fuzzy *k*-means, Gaussian mixture, and Rough Set's reduct. It is hoped that these three algorithms would be a welcome addition to a data analyst's ever burgeoning toolbox.

Contents

1.0 Introduction 5
2.0 Problem Statement and Objectives 6
3.0 Literature Review 6
3.1 Gaussian Mixture Model 7
3.2 Fuzzy Clustering 8
3.3 Rough Set Theory 9
4.0 Research Design and Methods 11
4.1 Pre-Clustering (Rough Set Variable Reduction) 13
4.2 Employing Relevant Clustering Methods – k-Means Clustering 15
4.3 Employing Relevant Clustering Methods – Fuzzy k-Means Clustering 21
4.4 Employing Relevant Clustering Methods – Gaussian Mixture Model 24
5.0 Analysis and Results 27
5.1 Overview of Clustering Output Results 27
5.2 TKM Clustering Output Results 29
5.3 FKM Clustering Output Results 38
5.4 Revisiting FKM-R 42
5.5 GMM Clustering Output Results 43
5.6 Revisiting GMM-R 51
6.0 Extended Analysis and Results 60
6.1 Revisiting Reduct-Based Outputs 60
6.2 Overview of Extended Clustering Output Results 65
6.3 Tying Back Clustering Output Results To Household Data 83
6.4 TKM Clustering Output Household Characteristics 85
6.5 FKM Clustering Output Household Characteristics 90
6.6 GMM Clustering Output Household Characteristics 92
6.7 3 Reduct Variables Clustering Output Household Characteristics 100
7.0 Discussion 109
8.0 Conclusions and Contributions 111
9.0 Milestones and Reflections 111
10.0 References 113
11.01 Appendix 1: Clustering Comparisons 115
11.02 Appendix 2: Summary of Clustering Methodology 116
11.03 Appendix 3: Remaining TVM Variables 117
11.04 Appendix 4: Remaining Rough Set Reduct Variables 122
11.05 Appendix 5: Rough Set Reduct R Code – 1st Run 123
11.06 Appendix 6: Rough Set Reduct R Code – 2nd Run 123
11.07 Appendix 7: Rough Set Reduct R Code – 3rd Run 123
11.08 Appendix 8: TKM-N-T R Code – Exploration 123
11.09 Appendix 9: TKM-N-T R Code – 2 Optimised Clusters 124
11.10 Appendix 10: TKM-A-T R Code – Exploration 124
11.11 Appendix 11: TKM-A-T R Code – 3 Optimised Clusters 125
11.12 Appendix 12: TKM-N-R R Code – Exploration 126
11.13 Appendix 13: TKM-N-R R Code – 2 Optimised Clusters 126
11.14 Appendix 14: TKM-A-R R Code – Exploration 127
11.15 Appendix 15: TKM-A-R R Code – 2 Optimised Clusters 127
11.16 Appendix 16: FKM-T R Code – Exploration 128
11.17 Appendix 17: FKM-T R Code – 2 Optimised Clusters 128
11.18 Appendix 18: FKM-R R Code – Exploration 129
11.19 Appendix 19: FKM-R R Code – 2 Optimised Clusters 129
11.20 Appendix 20: FKM-R R Code – 6 Optimised Clusters 130
11.21 Appendix 21: FKM-R R Code – 7 Optimised Clusters 131
11.22 Appendix 22: FKM-R R Code – 9 Optimised Clusters 132
11.23 Appendix 23: GMM-T R Code – Exploration 133
11.24 Appendix 24: GMM-T R Code – 8 Optimised Clusters 133
11.25 Appendix 25: GMM-R R Code – Exploration 134
11.26 Appendix 26: GMM-R R Code – 19 Optimised Clusters 134
11.27 Appendix 27: GMM-R R Code – 8 Optimised Clusters 135
11.28 Appendix 28: TKM-N-R3 R Code – Exploration 135
11.29 Appendix 29: TKM-N-R3 R Code – 2 Optimised Clusters 136

11.30 Appendix 30: TKM-A-R3 R Code – Exploration 137
11.31 Appendix 31: TKM-A-R3 R Code – 2 Optimised Clusters 137
11.32 Appendix 32: GMM-R3 R Code – Exploration 138
11.33 Appendix 33: GMM-R3 R Code – 8 Optimised Clusters 138
11.34 Appendix 34: FKM-R3 R Code – Exploration 139
11.35 Appendix 35: FKM-R3 R Code – 5 Optimised Clusters 139
11.36 Appendix 36: Segment Profile R Code 140
11.37 Appendix 37: Household Profile R Code 141
11.38 Appendix 38: Cluster Comparison on Randomised Data 143
11.39 Appendix 39: Reproducible R Code for Randomised Data 144
11.40 Appendix 40: Reproducible R Code for Clustering Example 146

1.0 Introduction

Cluster analysis is an important data mining technique, which segregates data into separate groups, which themselves possess uniquely distinct attributes. Hierarchical and *k*-means clustering are two such established clustering methods, with differing approaches towards a similar clustering outcome. Since the introduction of both these established methods, data has grown increasingly complex and imprecise.

Though today's technology allows ease of data recording more so than ever before, data remains disparately stored i.e., held in often unconnected sources with differing structures. Putting them together into a single homogenous dataset becomes a gargantuan task, and this is even before any proper analytical work is performed. The combined dataset invariably contains inherent inconsistencies (such as imprecisions or incompleteness), that traditional and established clustering methods are not designed for.

Two different approaches were adopted to resolve these shortcomings. The first approach was to address the limitations of existing clustering methods. Gaussian Mixture Model (GMM) was introduced to specifically address *k*-means and its limitations, namely its fundamental assumptions of sphericity, equal variance, and hard clustering. Given that its methodology remains akin to *k*-means, GMM's application is similarly wide-ranging across diverse fields. That said, analysing such imperfect data was not limited to just tweaking existing clustering methods.

The other approach was to develop entirely new fields, capable of generating meaningful insights from inconsistent data. One novel research field was fuzzy set theory (Zadeh, 1965). All objects are said to belong to all different sets, albeit to varying degrees of membership, similar to 'soft' clustering. Conversely, this theory also recognizes objects that have binary set membership ie. that they either belong to a set or not. This is similar to 'hard' clustering like *k*-means. With the success of fuzzy set theory in analysing imperfect data, another novel research field was introduced over a decade later that is equally adaptable to recognizing both soft and hard clustering.

Rough Set theory (Pawlak, 1982) introduces the concept of approximating hard or crisp sets. These approximations are divided into two regions: the lower approximation (positive region) and the upper approximation (negative region). The former (lower approximation) is similar to 'hard' clustering, where the objects are unambiguously and *positively* assigned to a set. Conversely, the latter (upper approximation) recognizes objects that are *possibly* assigned to a set, similar to 'soft' clustering. As such, both fuzzy and rough sets achieve similar outcomes of identifying both 'hard' and 'soft' sets or clusters, through their own distinctive methodologies.

Whether it be Rough Set or fuzzy set theory, both novel research fields were created out of a need to analyse inconsistent data in a more robust way. This complementary nature has led to numerous works incorporating both theories together. Computationally efficient on managing uncertainty in large complex data, both theories have had major research and applications in machine learning, data mining and other domains (Pieta & Szmuc, 2021). Despite their robust applications in a highly technical research field such as artificial intelligence (AI), there have been surprisingly little applications in a much less complex, real-world field such as customer segmentation.

Thus, this study aims to understand and compare clustering approaches, both established and novel, specifically on customer segmentation, to the eventual benefit of the everyday data analyst.

The flow of this capstone report starts off with its problem statement and objectives. It is then followed by a literature review of these topics. An explanation of the capstone's six step approach is covered within the research design and methods and is then followed with an extensive analysis and results exploration, as well as a subsequent discussion section. This capstone then rounds off its study with its conclusions and contributions as well as the author's milestones and reflections.

2.0 Problem Statement and Objectives

Using a real-world dataset, this study explored the impact of Rough Set's reduct variable selection. Two different forms of dataset are fed into each clustering method: traditional variable manipulation (TVM) and the Rough Set-based variable reduction. TVM was completed during the progress report phase, while the Rough Set-based variable reduction has been expanded in this report.

This study also examined and compared the clustering outcome of *k*-means (both traditional and fuzzy) and GMM. The comparison would entail the following:

- Detailed the different outcomes
- Uncovered merits and shortcomings of each clustering methods
- Suggested situations where each approach would excel

Lastly, given SAS scholarship's requirements, this study used JMP Pro and SAS Enterprise Miner (EM) for data manipulation and clustering respectively. Given that these four algorithms (traditional *k*-means, fuzzy *k*-means, GMM and Rough Set) are not available within EM, this study explored the use of R and its statistical packages within EM's environment.

The table below shows the clustering methods used on the differing dataset.

DATA MANIPULATION & VARIABLE SELECTION METHODS	***Traditional Variable Manipulation (TVM)***	***Rough Set Based Variable Reduction (RSVR)***
CLUSTERING METHODS		
Traditional *k*-means (TKM)	TKM-T	TKM-R
Fuzzy *k*-means (FKM)	FKM-T	FKM-R
Gaussian Mixture Models (GMM)	GMM-T	GMM-R

Table 1. A Breakdown of the Clustering Methods and Variable Selection Techniques Involved

In summary, this study aims to understand whether either GMM, Rough Set-based variable reduction, fuzzy *k*-means, or all of them would be a welcomed complement to current established clustering methods such as *k*-means clustering.

3.0 Literature Review

Cormack (1971) explains that a traditional clustering approach assigns individual objects into initially unclear classes, where all entities within a class share similarities with one another. The converse is also true. Objects outside of their assigned classes are highly dissimilar to those within their respective classes. A well-regarded clustering methodology is the *k*-means clustering. Being an established non-hierarchical clustering approach, objects are partitioned such that the squared Euclidean distance between each object and the centroid of the class it resides in, is as small as the

centroids of the other classes (Steinley, 2006). Garla & Gaeth (2012) and Steinley (2006) explains the iterative *k*-means algorithm in detail:

1. Specify the number of clusters or classes known as *k*. This number is chosen *a priori* ie. based on theoretical deduction.
2. Randomly select *k* cluster centres, or initial seeds in the data space, as defined by P-dimensional vectors ($S_1^{(k)}$,…, $S_P^{(k)}$), for $1 \leq k \leq K$
3. Allocate objects to clusters, where the squared Euclidean distance, $d^2(i,k)$, between *i*th object and the *k*th seed vector, is at its minimum

$$d^2(i,k) = \sum_{j=1}^{P} (x_{ij} - s_j^{(k)})^2 \tag{1}$$

4. Re-compute new cluster centroids by:
   a. Calculate the centroid value for the *j*th variable in cluster C_k

$$\bar{x}_j^{(k)} = \frac{1}{n_k} \sum_{i \in C_k} x_{ij} \tag{2}$$

   b. Calculate complete centroid vector for cluster C_k by averaging all centroid values within

$$\bar{x}^{(k)} = (\bar{x}_1^{(k)}, \bar{x}_2^{(k)}, \ldots, \bar{x}_P^{(k)})'. \tag{3}$$

5. Repeat steps 3. and 4. Above until convergence criterion is achieved i.e., the centroids do not change

3.1 Gaussian Mixture Model

Raykow et. al (2016) explains that *k*-means suffers from sphericity, outliers, and geometric closeness. Sphericity assumes that all clusters, within the dataset, are spherical in nature with the same radius. Conversely, given that *k*-means heavily relies on linear distance, any outliers that are unusually far away from the rest of the points within the cluster would impair the results of the *k*-means. Lastly, *k*-means implicitly assumes each cluster within the dataset has the same volume and fails to take into consideration clusters with differing geometric closeness and densities. These three limitations prevent *k*-means from clustering improper data in a meaningful way. To overcome this, this capstone is influenced by the work of Garla and Gaeth (2012). There, a comparison was made between *k*-means, normal mixtures, and probabilistic-D clustering for a B2B segmentation study on customers' perceptions. Given that normal mixtures model, otherwise known as Gaussian Mixture Model (GMM), is not a common clustering method, this capstone aims to explore this on top of the traditional and fuzzy *k*-means.

Gaussian Mixture Model (GMM) was introduced to overcome the *k*-means assumptions of hard clustering, sphericity, outliers, and geometric closeness. It is also suited for an incomplete dataset (Melchior & Goulding, 2018). Patel and Kushwaha (2020) denotes each Gaussian or Normal distribution as a continuous probability distribution as follows:

$$N(X|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\Sigma|}} \exp\left\{-\frac{(X-\mu)^T\Sigma^{-1}(X-\mu)}{2}\right\} \tag{4}$$

where
1. μ is a D-dimensional mean vector
2. Σ is a D x D covariance matrix, describing the shape of the Gaussian
3. $|\Sigma|$ denotes the determinant of Σ

Each dataset is deemed to contain several different Gaussian (normal) distributions on the same linear plane, with each distribution being regarded as one cluster:

$$p(X) = \sum_{k=1}^{K} \pi_k N(X|\mu_k,\Sigma_k) \tag{5}$$

where
1. K is the number of components or clusters in the mixture model
2. π_k denotes the mixing coefficient, giving an estimate of the density of each Gaussian component
3. $N(X|\mu_k,\Sigma_k)$ denotes the Gaussian density as per (5), with mean μ_k, covariance Σ_k and the mixing coefficient π_k

Unlike traditional *k*-means which assigns each object to only one cluster, GMM assigns each object a probability of how closely it fits to each distribution. It thus has a 'fit' score relating to each cluster and would naturally be included in the cluster with the highest score. While it's akin to a *k*-means clustering method in that it uses the mean values, GMM goes one step further by incorporating variance into the methodology. This allows it to identify and separate clusters that are close to each other, overcoming some of its limitations.

3.2 Fuzzy Clustering

In the traditional deterministic form of *k*-means clustering, it is impossible to assess the impact of the imprecise data (Chaudhuri & Bhowmik, 1998). Given that *k*-means is a hard clustering methodology, objects are either assigned to a particular set or not. This becomes increasingly difficult with imprecise data, where there is every possibility that an object may not just belong to a single set. For *k*-means to adopt a soft clustering approach, it would need to be paired with an additional concept. As such, there have been numerous studies to introduce fuzzy logic into the traditional *k*-means clustering.

Building on classical set theory where objects in a set either belong to a set or otherwise, Zadeh (1965) theorised fuzzy sets, where these same objects have degrees of membership to a set. Granted imprecision within a data space, Bezdek (1981) explains the use of fuzzy *k*-means clustering to 'soft' classify its observations into *k* clusters through a minimisation algorithm. This allows the traditional *k*-means clustering method to adopt a more flexible approach, especially where imprecise data would generally not fit into a 'hard' cluster. Steinley (2006) expresses the fuzzy *k*-means as follows:

$$\text{Fuzzy } k\text{-means} = \sum_{i=1}^{N}\sum_{k=1}^{K} w_{ik}^{r} d^2(i,k) \tag{6}$$

where
1. w_{ik} indicates membership values or the degree to which object *i* belongs to cluster *k*
2. all w_{ik} sums to unity or 1
3. *r* is a hyper-parameter that controls the fuzziness of the solution and is always greater than or equal to 1. The higher the r value is, the 'fuzzier' the solution
4. $d^2(i,k)$ is the squared Euclidean distance between the *i*th point and the *k*th seed vector

On reflection, while *k*-means clustering also tries to minimise (4), its w_{ik} or membership values are only binary in nature. Separately, while r takes values between 1 and ∞ ie. $r \in \{1, \infty\}$, a larger r results in fuzzier clusters. In the case of m →1, the membership, w_{ik}, converses to either 0 or 1, and Fuzzy *k*-means would match the behaviour of traditional *k*-means.

3.3 Rough Set Theory

This study thus far has considered traditional *k*-means, fuzzy *k*-means and GMM as part of the clustering concepts to explore. To efficiently discover clusters, feature selection is an important *pre*-clustering process. The task of finding important and significant variables within the unsupervised dataset allows for highly efficient processing, without eroding or compromising the overall quality of the dataset. A common feature selection would be to identify non-collinear variables for subsequent *k*-means clustering. As an alternative feature selection, this capstone explored Rough Set theory's reduct algorithm.

The Rough Set theory begins by separating these same objects into indiscernible classes, which can then be constructed as clusters themselves (Nayak et al., 2012). Chelly Dagdia et. al (2020) expounds the theory by starting with a tuple $S = (U, A)$, where $U = \{u_1, u_2, \dots, u_N\}$ is a non-empty finite set of N objects aptly named as *universe* and A is a non-empty set of $(n + k)$ *attributes*. Feature set $A = C \cup D$ can broken down to two subsets: *conditional* feature set, $C = \{a_1, a_2, \dots, a_n\}$, is made up of *n conditional* attributes or predictors, while decision attribute, $D = \{d_1, d_2, \dots, d_k\}$, is made up of *k decision* attributes or output variables. Each feature $a \in A$ is defined with a set of possible values V_a, or known as the *domain* of *a*. For each non-empty subset of attributes $P \subset C$, a binary relation named P-indiscernibility relation, is a key tenet of the Rough Set theory and is illustrated as follows:

$$IND(P) = \{(u_1, u_2) \in U \times U : \forall_a \in P, a(u_1) = a(u_2)\}. \quad (7)$$

where $a(u_i)$ refers to the value of attribute *a* for the instance u_i

If $(u_1, u_2) \in IND(P)$, then u_1 is indiscernible from u_2 by the attributes P. The induced set of equivalence classes $[u]p$,where $u \in U$, separates U into different parts or blocks denoted as U/P. The Rough Set theory approximates a target set of objects $X \subseteq U$ using the equivalence classes induced using P as follows:

$$\underline{P}(X) = \{u : [u]_P \subseteq X\}. \text{ (lower approximation)} \quad (8)$$

where $\underline{P}(X)$ denotes the P-lower approximations of *X*, or positively and certainly assigned to set *X*

$$\bar{P}(X) = \{u: [u]_P \cap X \neq \emptyset\}. \text{ (upper approximation)} \tag{9}$$

where $\bar{P}(X)$ denotes the P-upper approximations of X, or possibly assigned to set X

The difference between the two approximations is called the *boundary region* and is made up of the set of instances that are not positively, but possibly classified in a certain way. X is said to be a *crisp* set if the boundary region is an empty set where $\underline{P}(X) = \bar{P}(X)$. Failure to meet this condition would yield X as a *rough set*.

A dependency measure is defined to compare subsets of attributes. For example, the dependency measure of an attribute subset Q on another attribute subset P is given as:

$$_{\gamma P}(Q) = \frac{|POS_P(Q)|}{|U|} \tag{10}$$

where $0 \leq {}_{\gamma P}(Q) \leq 1$, $\cup$ denotes the union operation, $||$ denotes the set cardinality, and $POS_P(Q)$ is defined as:

$$POS_P(Q) = \bigcup_{X \in [u]_Q} \underline{P}\,(X) \tag{11}$$

$POS_P(Q)$ is the positive region of subset Q in relation to subset P, and is the set of all objects of U that can be uniquely classified to blocks of the partition $[u]_Q$, by means of P. As $_{\gamma P}(Q)$ nears to 1, the greater Q depends on P.

The additional benefit of Rough Set theory is the *reduct* concept. It is the ability to reduce the number of attributes that objects possess in such a way that it remains distinguishable as before. A subset $R \subseteq C$ is considered a *reduct* of C where:

$$_{\gamma R}(D) = {}_{\gamma C}(D) \tag{12}$$

and there is no $R' \subset R$ such that $_{\gamma R'}(D) = {}_{\gamma R}(D)$.

These *reduct* concept, distilling only 'significant' attributes, remain a vital part of the cluster generation. Recent research applications of Rough Set theory mainly focus on attribute reduction, rule acquisition and intelligent algorithm (Zhang et. al., 2016). Conversely, clustering-specific Rough Set-based research has been few. It was this capstone's initial direction to venture into this somewhat uncharted field. Upadhyaya, Arora, & Jain (2008) and Singh & Mandal (2017) applied *rough set*'s *reduct* algorithm on the attributes within each cluster, *post* clustering, to reduce each cluster's insignificant attributes and retain its significant attributes. With this method, each cluster would retain attributes pertinent to its unique cluster characteristics. While this capstone acknowledges this *reduct* approach as a possible direction to take, nonetheless this capstone's eventual direction slightly differs. Rather than incorporate the *reduct* algorithm *post* clustering, this capstone aims to examine the effects of incorporating a *reduct* algorithm *pre*-clustering. By comparing a 'normal' dataset with another dataset *post reduct*, this capstone aims to examine and compare its effects on cluster accuracy.

4.0 Research Design and Methods

This study's approach are broadly classified into the following six sequential steps, of which steps 3 to 6 are encapsulated in Figure 1 below.

1. Overall Intent of Study
2. Original Data Overview and Table Selection
3. Data Manipulation
4. *Pre*-Clustering
5. Employing Relevant Clustering Methods
6. Clustering Result Comparison

Figure 1 below shows the workflow steps beginning from data manipulation to *pre*-clustering techniques to actual clustering applications, before finally comparing the cluster output against its original household data.

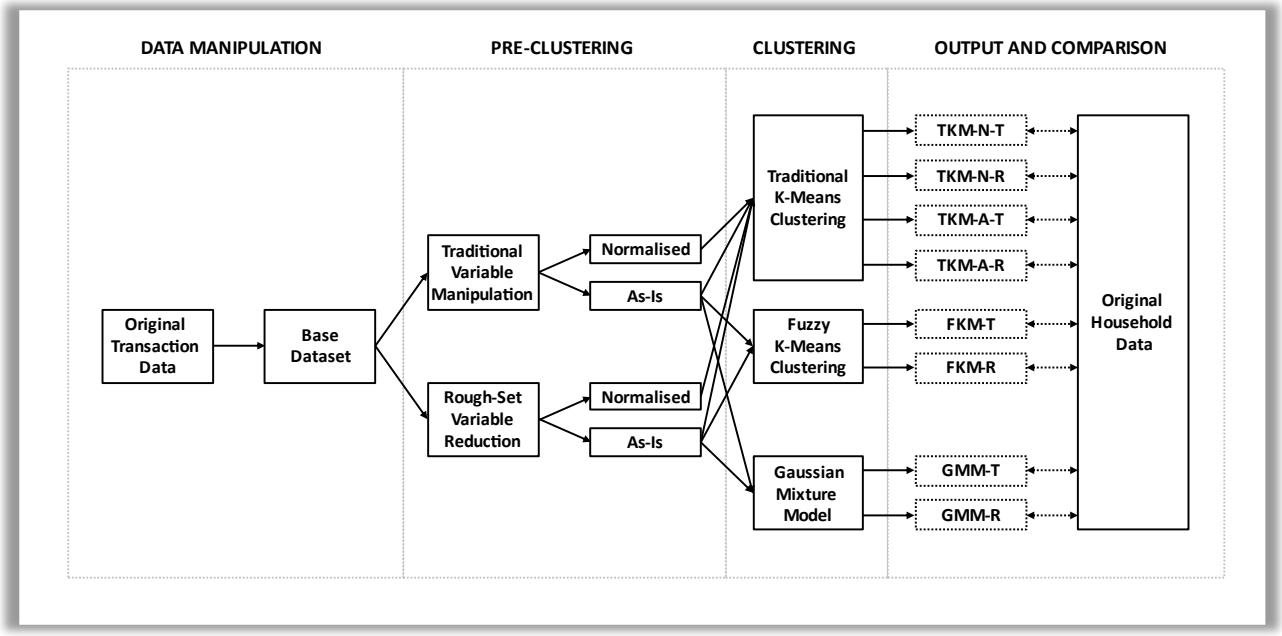
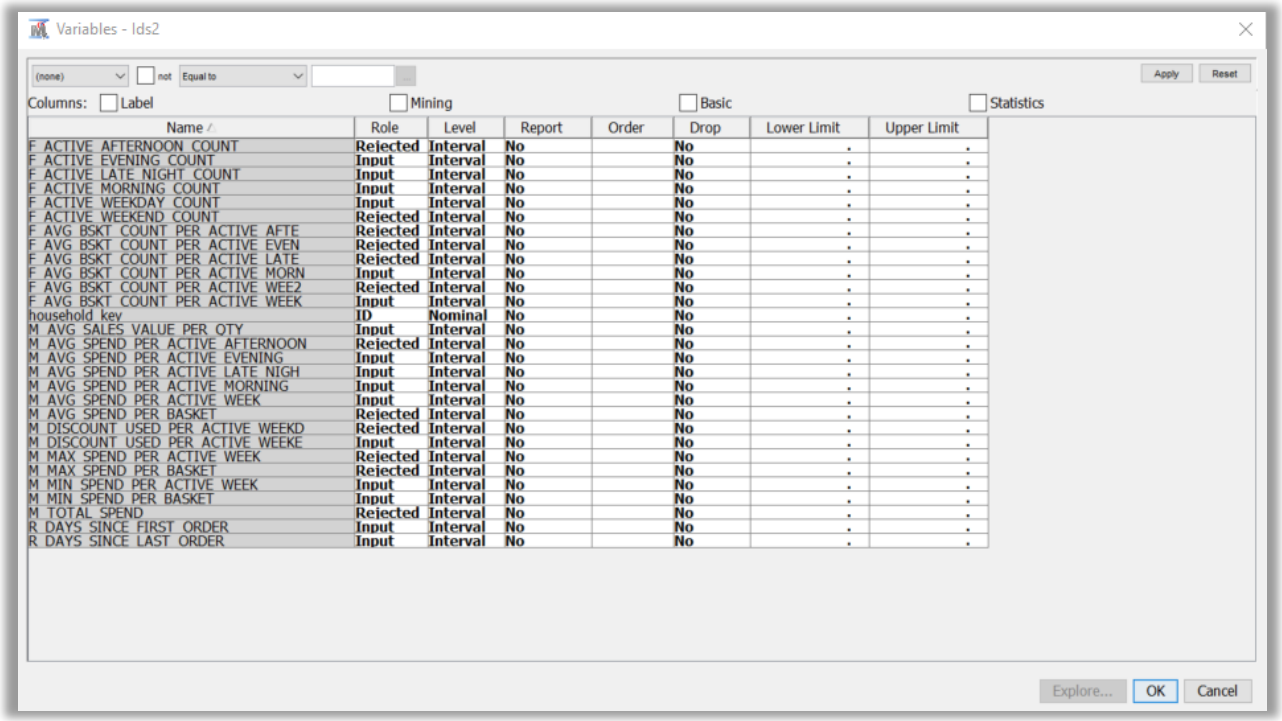


Figure 1: Capstone Workflow

To add, the completion of steps 1 to 3 i.e., overall intent of study, original data overview and table selection as well as data manipulation were covered in the earlier progress report. At the same time, the traditional variable manipulation part of step 4 was similarly completed. Excluding the household ID variable, this traditional variable manipulation part identified 16 non-collinear variables from the overall 28 variables as per Figure 2. These non-collinear variables are crucial assumptions within the *k*-means algorithm.



Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
F_ACTIVE_AFTERNOON_COUNT	Rejected	Interval	No		No	.	.
F_ACTIVE_EVENING_COUNT	Input	Interval	No		No	.	.
F_ACTIVE_LATE_NIGHT_COUNT	Input	Interval	No		No	.	.
F_ACTIVE_MORNING_COUNT	Input	Interval	No		No	.	.
F_ACTIVE_WEEKDAY_COUNT	Input	Interval	No		No	.	.
F_ACTIVE_WEEKEND_COUNT	Rejected	Interval	No		No	.	.
F_AVG_BSKT_COUNT_PER_ACTIVE_AFTE	Rejected	Interval	No		No	.	.
F_AVG_BSKT_COUNT_PER_ACTIVE_EVEN	Rejected	Interval	No		No	.	.
F_AVG_BSKT_COUNT_PER_ACTIVE_LATE	Rejected	Interval	No		No	.	.
F_AVG_BSKT_COUNT_PER_ACTIVE_MORN	Input	Interval	No		No	.	.
F_AVG_BSKT_COUNT_PER_ACTIVE_WEE2	Rejected	Interval	No		No	.	.
F_AVG_BSKT_COUNT_PER_ACTIVE_WEEK	Input	Interval	No		No	.	.
household_key	ID	Nominal	No		No	.	.
M_AVG_SALES_VALUE_PER_QTY	Input	Interval	No		No	.	.
M_AVG_SPEND_PER_ACTIVE_AFTERNOON	Rejected	Interval	No		No	.	.
M_AVG_SPEND_PER_ACTIVE_EVENING	Input	Interval	No		No	.	.
M_AVG_SPEND_PER_ACTIVE_LATE_NIGH	Input	Interval	No		No	.	.
M_AVG_SPEND_PER_ACTIVE_MORNING	Input	Interval	No		No	.	.
M_AVG_SPEND_PER_ACTIVE_WEEK	Input	Interval	No		No	.	.
M_AVG_SPEND_PER_BASKET	Rejected	Interval	No		No	.	.
M_DISCOUNT_USED_PER_ACTIVE_WEEKD	Rejected	Interval	No		No	.	.
M_DISCOUNT_USED_PER_ACTIVE_WEEKE	Input	Interval	No		No	.	.
M_MAX_SPEND_PER_ACTIVE_WEEK	Rejected	Interval	No		No	.	.
M_MAX_SPEND_PER_BASKET	Rejected	Interval	No		No	.	.
M_MIN_SPEND_PER_ACTIVE_WEEK	Input	Interval	No		No	.	.
M_MIN_SPEND_PER_BASKET	Input	Interval	No		No	.	.
M_TOTAL_SPEND	Rejected	Interval	No		No	.	.
R_DAYS_SINCE_FIRST_ORDER	Input	Interval	No		No	.	.
R_DAYS_SINCE_LAST_ORDER	Input	Interval	No		No	.	.

Figure 2: 16 Non-Collinear Variables Kept as Inputs For k-Means; Rest Are Rejected

Thus, this final report would continue from the Rough Set variable reduction part of step 4.

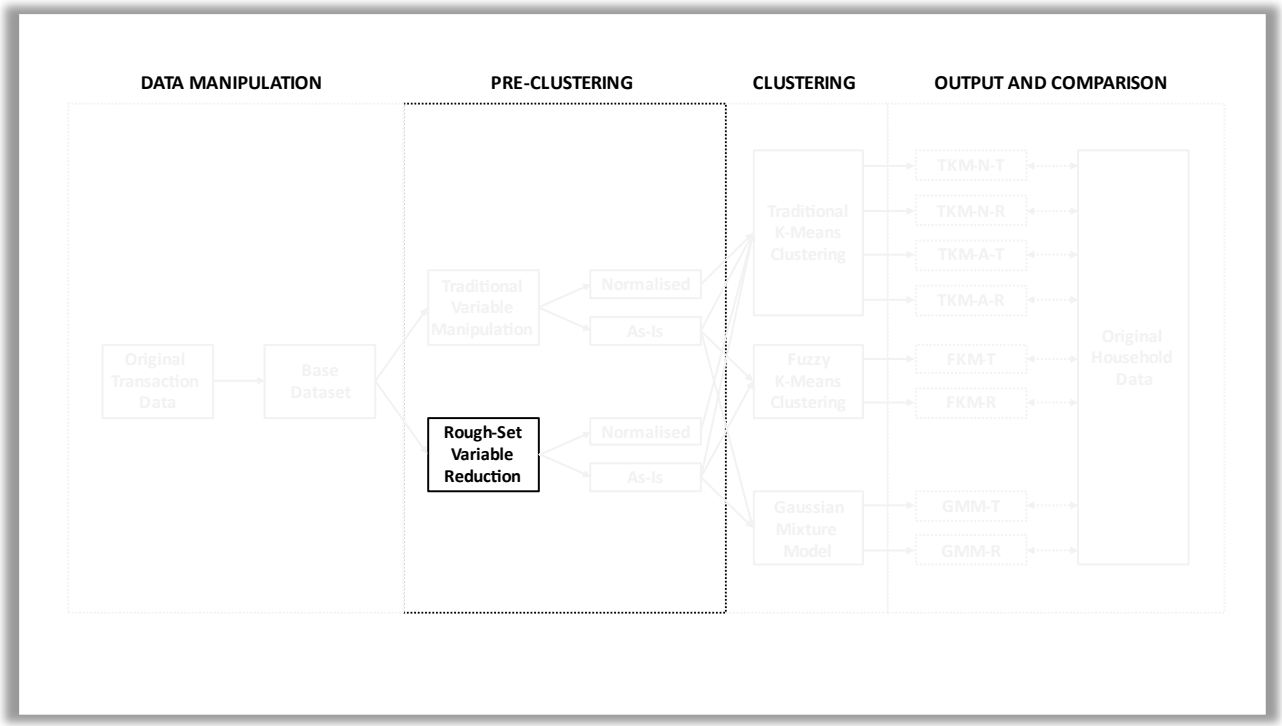


Figure 3: Capstone Workflow with emphasis on current step

4.1 *Pre*-Clustering (Rough Set Variable Reduction)

This sub-section implements Rough Set's reduct feature selection algorithm onto the base dataset. Though Rough Set algorithm exists within a multitude of programming language packages (R, C++, and Java) (Riza et. al, 2014), Enterprise Miner does not include Rough Set reduct within its standard software. As a workaround, Enterprise Miner has an Open-Source Integration (OSI) Node that allows for both R and Python scripts to run within the EM environment. Since Rough Set already exists as an R package, using this node allows for Rough Set to be run within EM. Similarly, R packages of *k*-means, fuzzy *k*-means and Gaussian Mixture Models algorithms would also be executed with the Open-Source Integration node in similar fashion.

To ensure successful R integration with EM, compatibility of both versions need to be carefully considered. This capstone uses EM version 14.1. As such, SAS recommend 64-Bit R versions between 2.13.0 and 3.2.5. At the time of this capstone, the latest stable R version is 4.2.2. Since EM version 14.1 only recognizes R version 3.2.5 and nothing later, it must be said that certain R packages, considered useful to this capstone, may have limited usage on this earlier R version, or may even find itself completely unusable. Nonetheless, it is this capstone's intention to find workable alternatives to ultimately deliver meaningful analysis and insights. Lastly, for housekeeping purposes and to ensure a streamlined and coherent report, all R codes within this capstone will be included in the appendix.

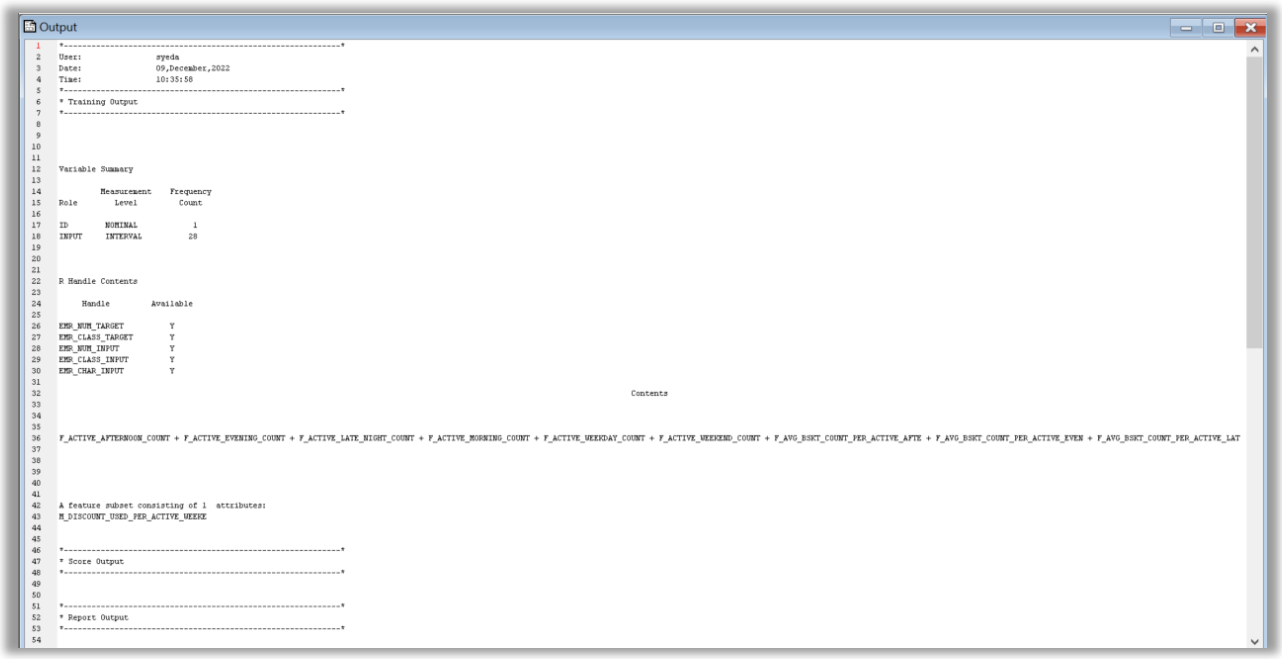
As seen in Figure 4 below, the Open-Source Integration Node is attached to the Data_All Node containing the base dataset of 28 input variables and 1 ID variable. The Rough Set reduct R code, using the *RoughSets* R package, is then entered within the Code Editor of the OSI Node. The Rough Set reduct R code used within this node closely follows Riza et. al (2014).

Figure 4: 16 Attaching Open-Source Integration Node to Data_All Node

The base dataset is first converted as a decision table. Unlike R's usual *dataframe* or *tibble* type, a decision table has attribute descriptions, types of attributes and an index of the decision attribute. The *household_ID* is classified as an index of the decision attribute, with a nominal property. As a *pre*-processing step, this decision table is then discretized or converted from real-valued attributes into nominal attributes. This step maintains the discernibility between objects in information systems. Of the many different discretization approaches, there is one option for either global (discretizing values over the whole continuous attribute) or local (discretizing values over localized regions). Riza et. al. (2014) used global discretization, and thus this capstone followed suit. The eventual discretized values are then applied back to the decision table to generate a new decision table. Though there are other methods (quickreduct.rst, quickreduct.frst), feature selection is performed using the greedy.heuristic.superreduct method. The R code detailing this whole Rough Set reduct process is included in Appendix 5, and 6.

Figure 5 shows the Rough Set reduct output. Out of 28 possible input variables to choose from, the R code within the OSI Node chose only one input variable: M_DISCOUNT_USED_PER_ACTIVE_WEEKEND. This meant that the amount of discounts on

an active weekend was THE attribute that is akin to the entire set of shopping consumer behaviour attributes.



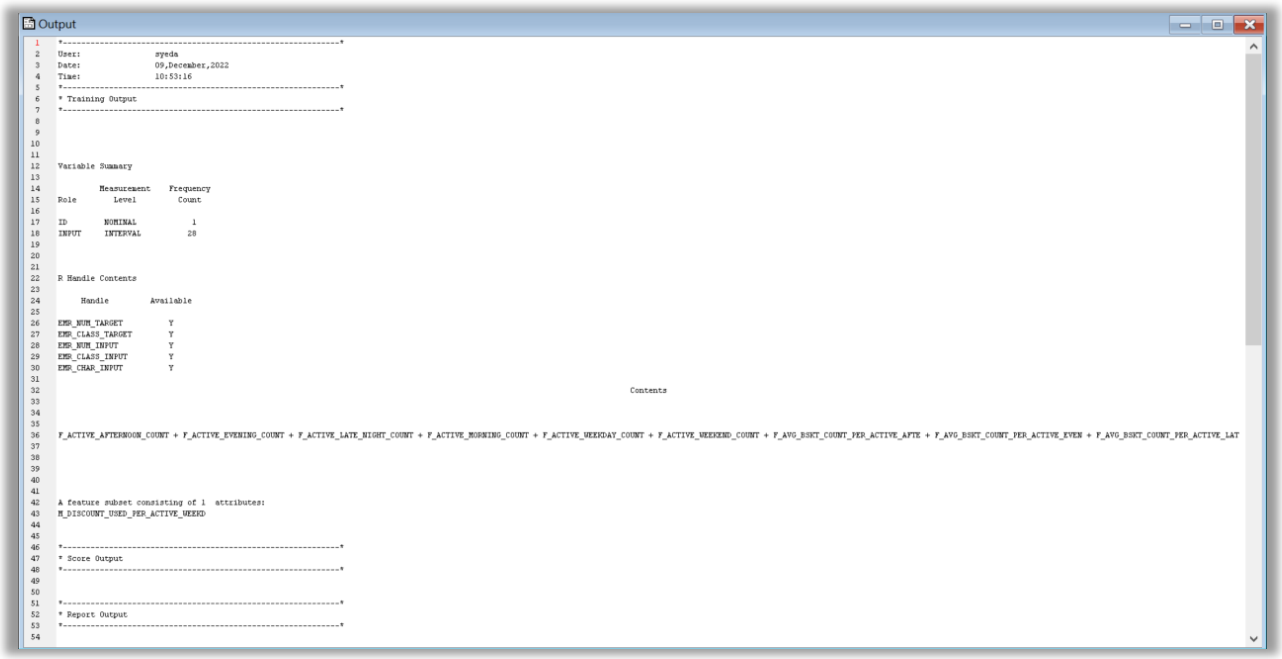
```
Output
*------------------------------------------------------------*
User:                 syeda
Date:                 09,December,2022
Time:                 10:35:58
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*



Variable Summary

          Measurement    Frequency
Role        Level         Count

ID        NOMINAL            1
INPUT     INTERVAL          28



R Handle Contents

      Handle          Available

EMP_NUM_TARGET          Y
EMP_CLASS_TARGET        Y
EMP_NUM_INPUT           Y
EMP_CLASS_INPUT         Y
EMP_CHAR_INPUT          Y

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                
Contents

F_ACTIVE_AFTERNOON_COUNT + F_ACTIVE_EVENING_COUNT + F_ACTIVE_LATE_NIGHT_COUNT + F_ACTIVE_MORNING_COUNT + F_ACTIVE_WEEKDAY_COUNT + F_ACTIVE_WEEKEND_COUNT + F_AVG_BSKT_COUNT_PER_ACTIVE_AFTE + F_AVG_BSKT_COUNT_PER_ACTIVE_EVEN + F_AVG_BSKT_COUNT_PER_ACTIVE_LAT



A feature subset consisting of 1  attributes:
M_DISCOUNT_USED_PER_ACTIVE_WEEKE

*------------------------------------------------------------*
* Score Output
*------------------------------------------------------------*


*------------------------------------------------------------*
* Report Output
*------------------------------------------------------------*
```

Figure 5: Rough Set Reduct Output from Data_All

Unfortunately, given that clustering algorithms require a minimum of two variables, at least one other input is required. To meet this requirement, this capstone then removed the 'M_DISCOUNT_USED_PER_ACTIVE_WEEKEND' input variable from the Data_All Node. Since the OSI Node takes in the whole data node regardless of rejected variables, filtering out the 'M_DISCOUNT_USED_PER_ACTIVE_WEEKEND' input variable would need to be done within the OSI Node.

Figure 6: 16 Attaching Open-Source Integration Node (Removed One Input and Redid Reduct) to Data_All Node

Figure 6 shows the second run of Rough Set reduct output. Out of 27 possible input variables to choose from, the R code within the OSI Node chose only one input variable as before. This time, the input variable 'M_DISCOUNT_USED_PER_ACTIVE_WEEKDAY' was chosen. Thus, the reduct dataset (originally from the base dataset) would now comprise of one ID variable (household_ID), and two input variables (M_DISCOUNT_USED_PER_ACTIVE_WEEKDAY and M_DISCOUNT_USED_PER_ACTIVE_WEEKEND).



```
Output
*-----------------------------------------------------------------*
User:                 syeda
Date:                 09,December,2022
Time:                 10:53:16
*-----------------------------------------------------------------*
* Training Output
*-----------------------------------------------------------------*



Variable Summary

          Measurement    Frequency
Role        Level         Count

ID        NOMINAL            1
INPUT     INTERVAL          28



R Handle Contents

      Handle          Available

EMR_NUM_TARGET          Y
EMR_CLASS_TARGET        Y
EMR_NUM_INPUT           Y
EMR_CLASS_INPUT         Y
EMR_CHAR_INPUT          Y
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                
F_ACTIVE_AFTERNOON_COUNT + F_ACTIVE_EVENING_COUNT + F_ACTIVE_LATE_NIGHT_COUNT + F_ACTIVE_MORNING_COUNT + F_ACTIVE_WEEKDAY_COUNT + F_ACTIVE_WEEKEND_COUNT + F_AVG_BSKT_COUNT_PER_ACTIVE_AFTE + F_AVG_BSKT_COUNT_PER_ACTIVE_EVEN + F_AVG_BSKT_COUNT_PER_ACTIVE_LAT
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                
A feature subset consisting of 1  attributes:
M_DISCOUNT_USED_PER_ACTIVE_WEEKD
*-----------------------------------------------------------------*
* Score Output
*-----------------------------------------------------------------*
*-----------------------------------------------------------------*
* Report Output
*-----------------------------------------------------------------*
```

Figure 7: Rough Set Reduct Output from Data_All
(Excluding M_DISCOUNT_USED_PER_ACTIVE_WEEKEND)

4.2 Employing Relevant Clustering Methods – *k*-Means Clustering

As mentioned in the progress report, four out of the eight clustering outputs are generated from *k*-means clustering. The outputs differ due to:

- Type of Dataset Used:
  - 16 non-collinear input variables and 1 ID variable
  - 2 Rough Set reduct variables and 1 ID variable
- Whether dataset was normalized (using the Transform Variables Node) before *k*-means clustering was performed

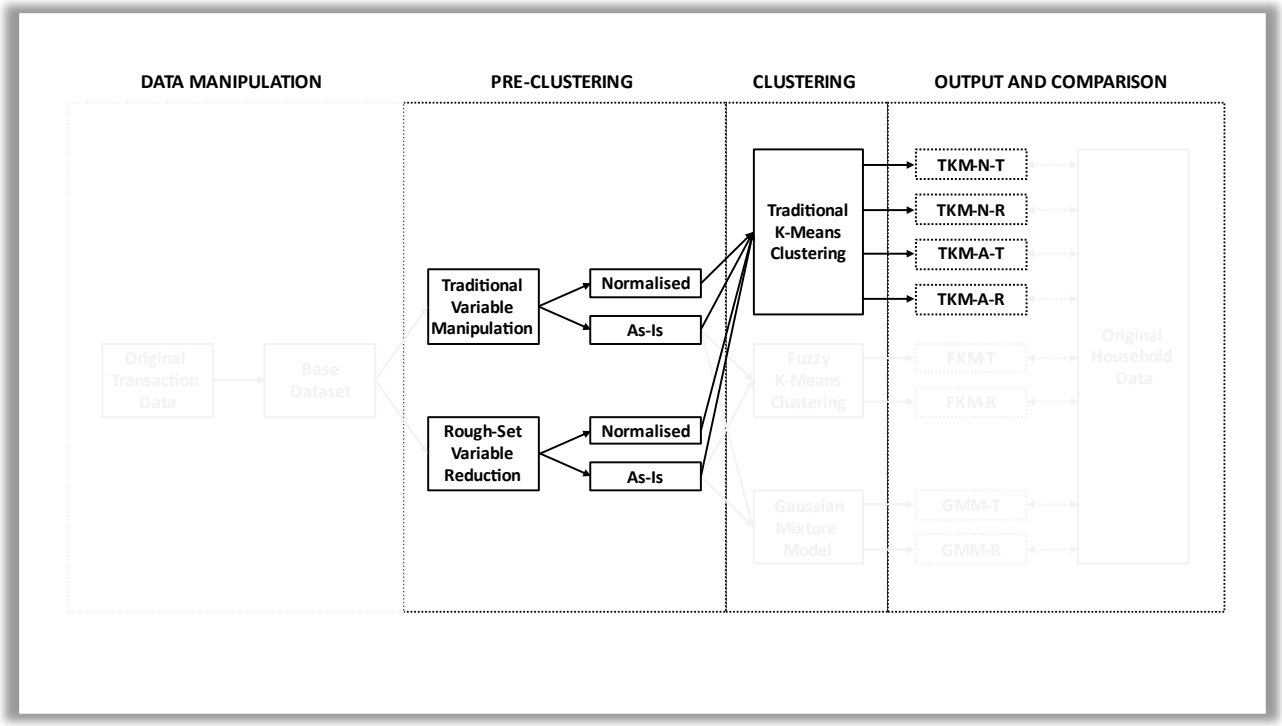


Figure 8: Capstone Workflow with emphasis on current step

It's worth noting that since *k*-means is a distance-based algorithm, its input variables need to be scaled in a similar fashion. This scaling will be done in the R code. This scale function, already within base R, centres and scales the values, using either standard deviation or root mean square.

Separately, the clustering algorithm in the Cluster Node within SAS EM is closer to hierarchical clustering, than *k*-means clustering. The following details were lifted from SAS EM 14.1's Reference Help, with emphasis on the hierarchical clustering.

> *"The Automatic setting (default) configures SAS Enterprise Miner to automatically determine the optimum number of clusters to create.*
>
> *When the Automatic setting is selected, the value in the Maximum Number of Clusters property in the Number of Clusters section is not used to set the maximum number of clusters. Instead, SAS Enterprise Miner first makes a preliminary clustering pass, beginning with the number of clusters that is specified as the Preliminary Maximum value in the Selection Criterion properties.*
>
> *After the preliminary pass completes, the multivariate means of the clusters are used as inputs for a second pass that uses* ***agglomerative, hierarchical algorithms*** *to combine and reduce the number of clusters. Then, the smallest number of clusters that meets all four of the following criteria is selected."*

Since *k*-means clustering is not found within SAS EM, similar to Rough Set, this section will focus on using R code within OSI to perform *k*-means clustering. R already has *k*-means clustering built into its base stats package. To determine the optimal number of *k* clusters, this capstone would use

the silhouette coefficient, with values between -1 and +1. It is a measure of how similar a data-point is within-cluster (cohesion), compared to other clusters (separation). Rousseeuw (1987) first defines a value of *s(i)* in the case of dissimilarities for object *i*. Object *i* is then assigned to cluster *A*. When cluster *A* has other objects apart from *i*, the following can be computed:

$$a(i) = \text{average dissimilarity distance of } i \text{ to all other objects within cluster } A \tag{13}$$

Considering another cluster C, which is different from cluster *A*, the following can be computed:

$$d(i, C) = \text{average dissimilarity distance of } i \text{ to all objects of cluster } C \tag{14}$$

After computing *d(i, C)* for all clusters $C \neq A$, the smallest of these numbers are then selected, and denoted by:

$$b(i) = \text{minimum } d(i, C), \text{ where } C \neq A \tag{15}$$

Thus, the number *s(i)* is obtained by combining *a(i)* and *b(i)* as follows:

$$s(i) = \begin{cases} 1 - b(i) / a(i) & if\ a(i) < b(i) \\ 0 & if\ a(i) = b(i) \\ b(i) / a(i) - 1 & if\ a(i) > b(i) \end{cases}, or \tag{16}$$

$$s(i) = \frac{b(i) - a(i)}{max\,\{a(i),\ b(i)\}} \tag{17}$$

A silhouette coefficient close to +1 shows that the object *i* has been assigned to an appropriate cluster, while a silhouette coefficient close to -1 shows that the same object has been misclassified to its current cluster. A silhouette coefficient close to zero would mean that object *i* lies equally far from two clusters.

This section covers both the TKM-N-T and TKM-A-T outputs, using the dataset with 16 non-collinear input variables and one ID variable, as shown in the EM diagram below.

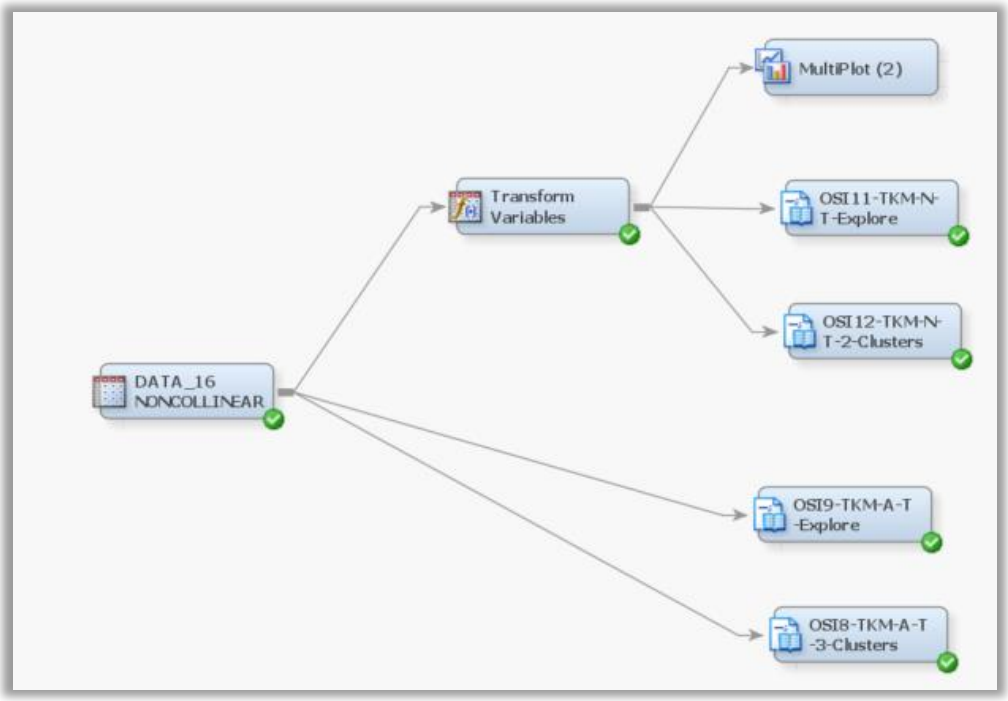


Figure 9: EM Diagram for TKM-N-T and TKM-A-T outputs
(Using 16 non-collinear inputs and 1 ID variable)

<u>TKM-N-T (Traditional *k*-means clustering on normalized form of 16 non-collinear inputs)</u>
First, the optimal number of clusters needs to be computed. Information on the transformation of these 16 non-collinear inputs, within the Transform Variables Node, is included in Appendix 3. These are then ingested into the OSI Node called 'OSI11-TKM-N-T-Explore.' The number 12 here refers to the node number within the EM Workspace and allows for ease of retrieval of .csv and plot files. Setting the seed value to 1234, and then calling the '*cluster*' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 50. This silhouette index is then plotted within EM, as seen below.

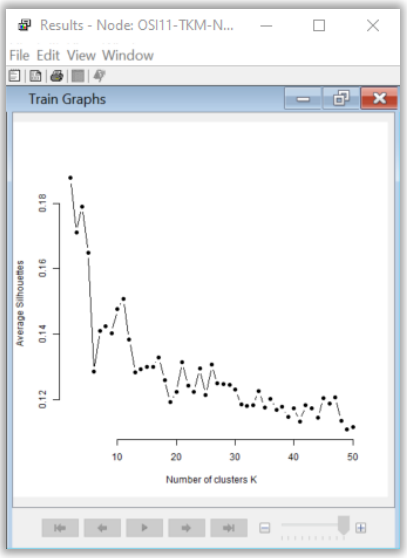


Figure 10:
Silhouette Index for TKM-N-T output

The silhouette index starts out strongest at $k = 2$, and then goes downwards sharply by $k = 6$, before gradually decreasing by $k = 50$. It's worth noting that the silhouette index within the graph is closer to zero than to 1. This might mean that the objects within the dataset are more evenly spread out, with little cluster concentrations. For now, we will continue to regard $k = 2$ as its most optimized cluster count.

Another OSI node, called 'OSI12-TKM-N-T-2-Clusters,' is now attached to the same Transform Variable Node. This time, instead of exploring, we will generate the $k = 2$ cluster output. Apart from the same R code above, a plot is created to compare the histograms of each cluster against overall for all variables. Analysis of these results will be covered in a later section.

TKM-A-T (Traditional *k*-means clustering on as-is form of 16 non-collinear inputs)

Similarly, the optimal number of clusters needs to be computed here. These are then directly ingested into the OSI Node called 'OSI9-TKM-A-T-Explore.' Setting the seed value to 1234, and then calling the '*cluster*' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 50. This silhouette index is then plotted within EM, as seen below.

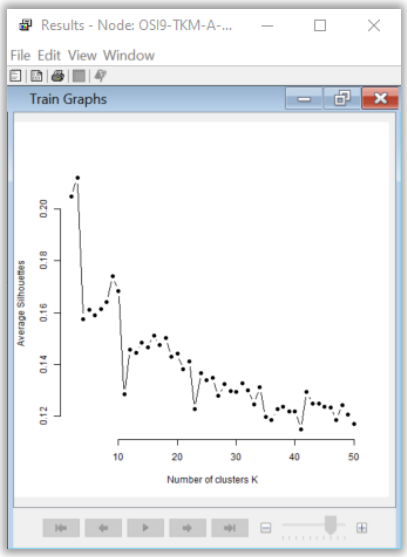


Figure 11:
Silhouette Index for TKM-A-T output

The silhouette index is strongest at $k = 3$, and then goes downwards sharply by $k = 4$, before gradually decreasing by $k = 50$. Again, it's worth noting that the silhouette index within the graph is closer to zero than to 1. Having said that, TKM-A-T's silhouette index is slightly higher than TKM-N-T. This suggests that the clustering was slightly impacted from the earlier normalisation. For now, we will regard $k = 3$ as its most optimized cluster count.

Another OSI Node, called 'OSI8-TKM-A-T-3-Clusters,' is now attached to the 16 non-collinear data node. The $k = 3$ cluster output will be generated together with its histogram plots. Analysis of these results will also be covered in a later section.

This section covers both the TKM-N-R and TKM-A-R outputs, using the dataset with 16 non-collinear input variables and one ID variable, as shown in the EM diagram below.

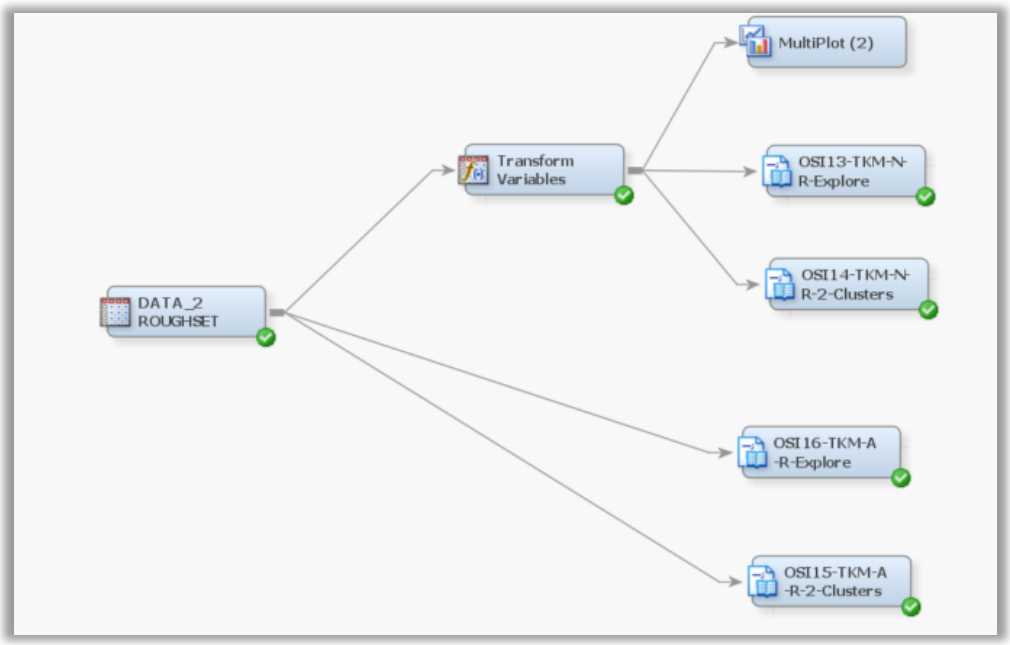


Figure 12: EM Diagram for TKM-N-R and TKM-A-R outputs
(Using 2 reduct inputs and 1 ID variable)

TKM-N-R (Traditional *k*-means clustering on normalized form of 2 reduct inputs)

Both discount input variables are first transformed within the Transform Variables Node. Information on both these transformations is included in Appendix 3. These are then ingested into the OSI Node called 'OSI13-TKM-N-R-Explore.' Setting the seed value to 1234, and then calling the 'cluster' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 50. This silhouette index is then plotted within EM, as seen below.

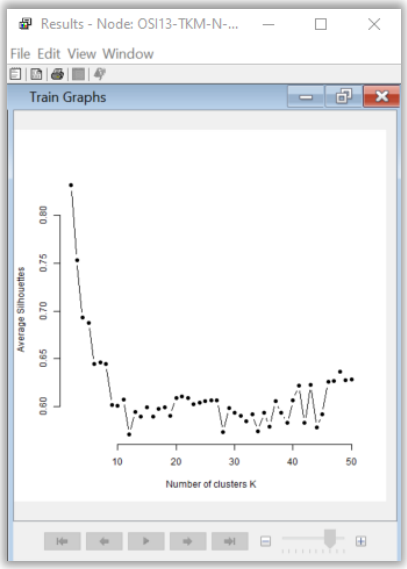


Figure 13:
Silhouette Index for TKM-N-R output

The silhouette index starts out strongest at $k = 2$, and then goes downwards sharply by $k = 12$, before plateauing til $k = 50$. This time, the silhouette index is much closer than before to 1. We can infer that the reduct algorithm had a strong impact in the clustering accuracy. As such, we will regard $k = 2$ as its most optimized cluster count.

Another OSI Node, called 'OSI14-TKM-N-R-2-Clusters,' is now attached to the same Transform Variable Node. This time, instead of exploring, we will generate the $k = 2$ cluster output. Apart from the same R code above, a plot is created to compare the histograms of each cluster against overall for all variables. Analysis of these results will be covered in a later section.

TKM-A-R (Traditional *k*-means clustering on as-is form of 2 reduct inputs)

Similarly, the optimal number of clusters needs to be computed here. These are then directly ingested into the OSI Node called 'OSI16-TKM-A-R-Explore.' Setting the seed value to 1234, and then calling the 'cluster' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 50. This silhouette index is then plotted within EM, as seen below.

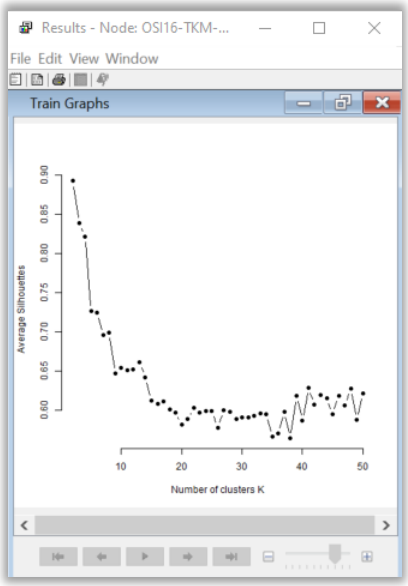


Figure 14:
Silhouette Index for TKM-A-R output

The silhouette index is strongest at $k = 2$, before gradually decreasing to $k = 50$. Similar to TKM-N-R, the silhouette index within the graph is very much close to 1. At the same time, TKM-A-R's silhouette index is slightly higher than TKM-N-R. This suggests that the clustering was slightly impacted from the earlier normalisation. Thus, we will regard $k = 2$ as its most optimized cluster count.

Another OSI Node, called 'OSI15-TKM-A-R-2-Clusters,' is now attached to the 2 reduct data node. The $k = 2$ cluster output will be generated together with its histogram plots. Analysis of these results will also be covered in a later section.

4.3 Employing Relevant Clustering Methods – Fuzzy *k*-Means Clustering

Similarly, as mentioned in the progress report, there will be two outputs generated from fuzzy *k*-means clustering. The difference is in the dataset used ie. either the full dataset with 28 input variables and 1 ID variable, or the reduct dataset with 2 input variables and 1 ID variable. This dataset will then be scaled within the OSI Node using the scale R package, before being fed into the fuzzy *k*-means R package, *fclust*.

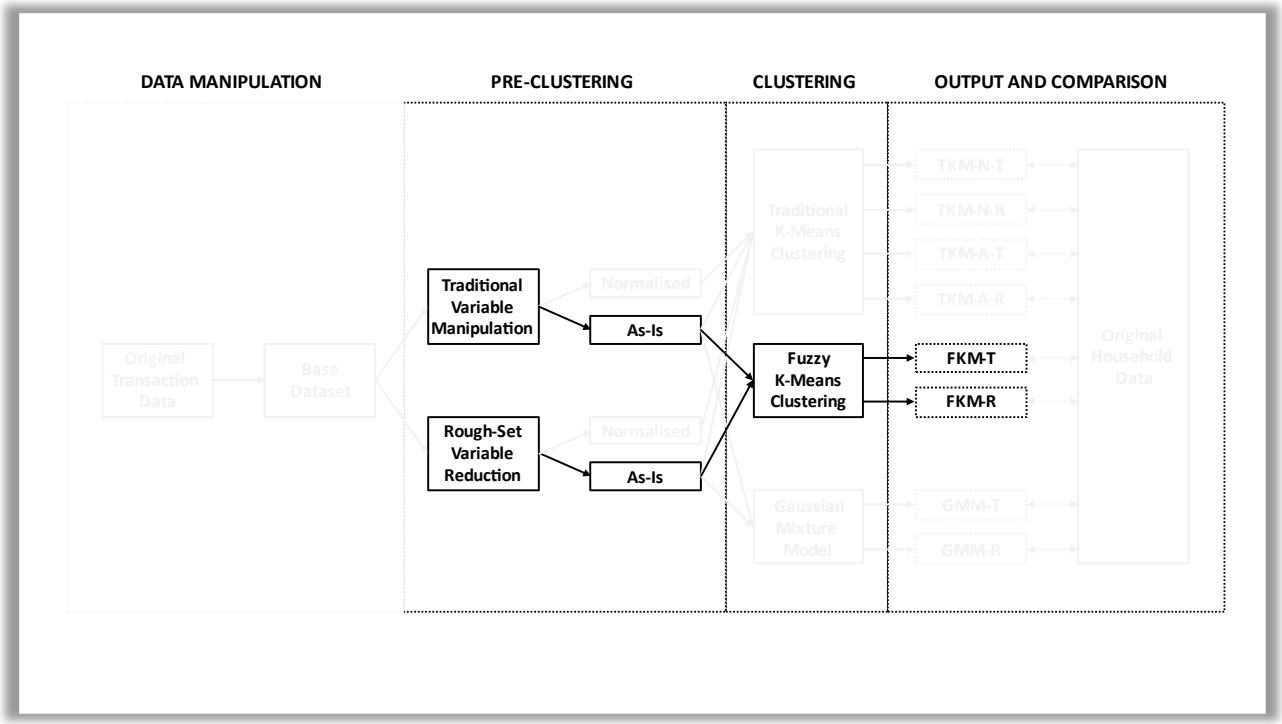


Figure 15: Capstone Workflow with emphasis on current step

Since fuzzy *k*-means is built on the traditional *k*-means clustering method with an additional fuzzy logic aspect, it is only natural that the same silhouette coefficient index is used to determine its optimal cluster count. The only difference is that the traditional silhouette coefficient calculates the fit of object *i* between its assigned cluster and otherwise. This is very much hard clustering and suited to the traditional *k*-means clustering. However, fuzzy *k*-means allows the same object *i* memberships to multiple clusters at the same time ie. soft clustering. As such, the silhouette index needs to be tweaked to incorporate the fuzzy logic element. Campello & Hruschka (2006) demonstrated exactly this in its aptly named form, Fuzzy Silhouette (FS):

$$FS = \frac{\sum_{j=1}^{N}(\mu_{pj} - \mu_{qj})^{\alpha} s_j}{\sum_{j=1}^{N}(\mu_{pj} - \mu_{qj})^{\alpha}}, \tag{18}$$

where
1. s_j is the silhouette value of object *j* as per Equation 17
2. μ_{pj} and μ_{qj} are the first and second largest elements of the *j*th column of the fuzzy partition matrix
3. $\alpha \geq 0$ is a weight coefficient

Equation 18 highlights the weighted average nature of the individual silhouettes, as per Equation 17. The weight of each term is determined by the difference in membership degrees of the corresponding object to its first and second best matching fuzzy clusters, respectively. Having said that, the FS range of values are very much the same as the Silhouette Index and are to be read as such.

As shown in the EM diagram below, the FKM-T (fuzzy *k*-means) and GMM-T (Gaussian) pull from the Data_All Node, while the FKM-R and GMM-R pulls from the Reduct Data Node. Both fuzzy *k*-means outputs would be covered first in the immediate section below, followed by both Gaussian outputs.

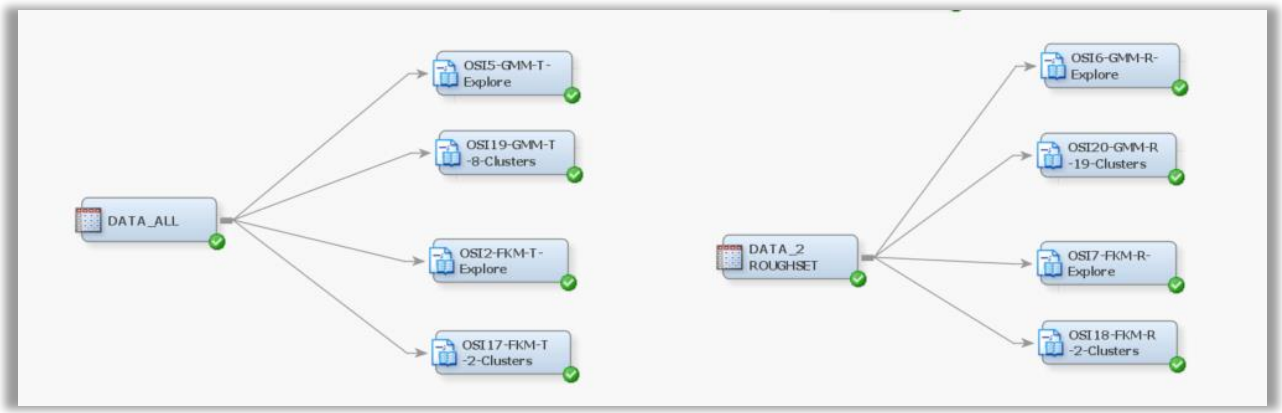


Figure 16: EM Diagram for FKM-T, GMM-T, FKM-R and GMM-R outputs

<u>FKM-T (Fuzzy *k*-means clustering on the as-is form of traditional variable manipulation)</u>
Similarly, the optimal number of clusters needs to be computed here. These are then directly ingested into the OSI Node called 'OSI2-FKM-T-Explore.' Setting the seed value to 1234, and then calling the '*cluster*' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 20. This is a much lower initial cluster exploration range, compared to the *k*-means count range of 2 to 50. This is due to the

extremely long processing time, as the cluster count increases. This fuzzy silhouette index is then plotted within EM, as seen below.

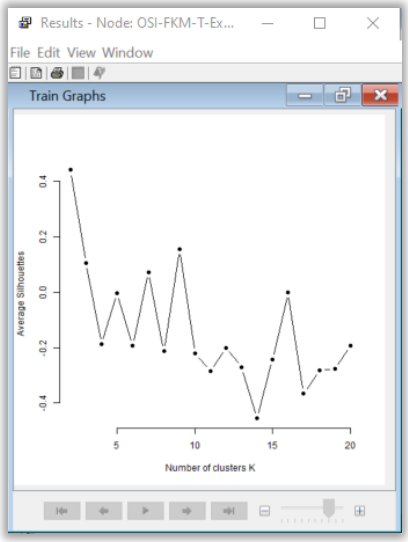


Figure 17:
Silhouette Index for FKM-T output

The silhouette index is strongest at $k = 2$. It then bounced up from $k = 4$, and then dropped down from $k = 5$, before stopping at $k = 6$. It continued this up-and-down motion for an extended bit. Thus, the highest fuzzy silhouette index is at $k = 2$. Similar to the outputs generated above, the fuzzy silhouette index within the graph is near the 0.5 mark. This is in between the scores generated for the TKM reduct series and the TKM as-is series. Thus, we will regard $k = 2$ as its most optimized cluster count.

Another OSI Node, called 'OSI17-FKM-T-2-Clusters,' is now attached to the Data_All Node. The $k = 2$ cluster output will be generated together with its histogram plots. Analysis of these results will also be covered in a later section.

<u>FKM-R (Fuzzy *k*-means clustering on the as-is form of 2 reduct inputs)</u>
Similarly, the optimal number of clusters needs to be computed here. These are then directly ingested into the OSI Node called 'OSI17-FKM-R-Explore.' Calling the '*cluster*' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 20. This is a much lower initial cluster exploration range, compared to the *k*-means count range of 2 to 50. This is due to the extremely long processing time, as the cluster count increases. This fuzzy silhouette index is then plotted within EM, as seen below.

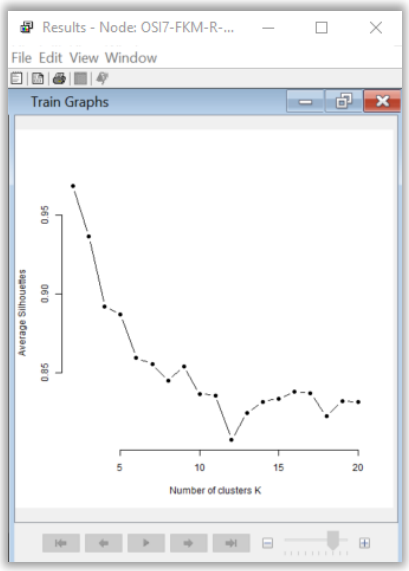


Figure 18:
Silhouette Index for FKM-R output

The silhouette index is strongest at $k = 2$. It then gradually reduces, before stopping at $k = 6$. There were no further k cluster counts after 20. Thus, the highest fuzzy silhouette index is at $k = 2$. This fuzzy silhouette index is almost double the one generated for FKM-T, mainly due to the reduct variables. Thus, we will regard $k = 2$ as its most optimized cluster count.

Another OSI Node, called 'OSI18-FKM-R-2-Clusters,' is now attached to the Data_All Node. The $k = 2$ cluster output will be generated together with its histogram plots. Analysis of these results will also be covered in a later section.

4.4 Employing Relevant Clustering Methods – Gaussian Mixture Model

Similarly, as mentioned in the progress report, there will be two outputs generated from GMM clustering. It uses the same dataset as the fuzzy *k*-means clustering ie. either the full dataset with 28 input variables and 1 ID variable, or the reduct dataset with 2 input variables and 1 ID variable. This dataset will then be scaled within the OSI Node using the scale R package, before being fed into the GMM R package, *mclust*.

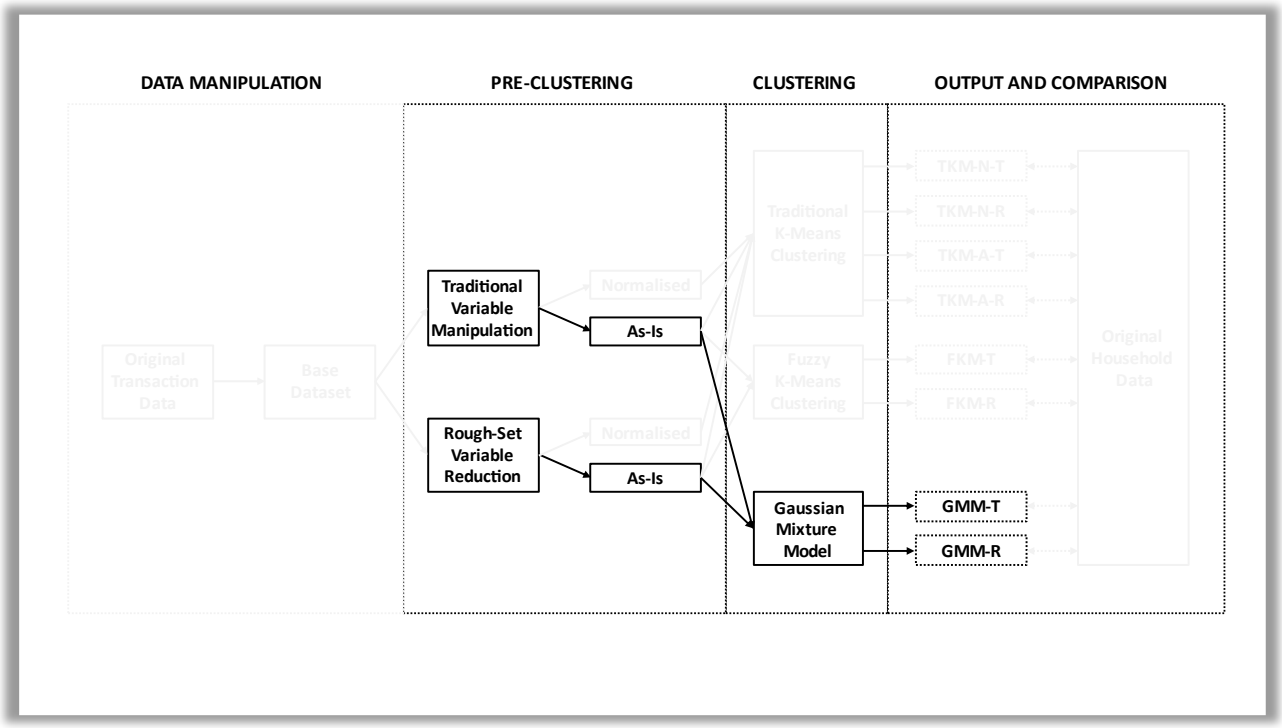


Figure 19: Capstone Workflow with emphasis on current step

For GMM, a common method for determining the optimal number of clusters is using Bayesian Information Criterion (BIC). Fraley & Raftery (2002) explains the integrated likelihood for regular models can be approximated by the following BIC formula. A maximum BIC value (the approximate Bayes factors with the BIC approximation) would determine the optimal cluster *k* for the dataset.

$$BIC_k = 2 \log p\,(\,\mathrm{D} \mid \mathrm{M}_k\,) \approx 2 \log p\,(\,\mathrm{D} \mid \hat{\theta}_k, \mathrm{M}_k) - \nu_k \log(\,n\,)\,, \tag{19}$$

where
1. ν_k are the number of independent parameters to be estimated in model M_k (Schwarz, 1978)
2. $p\,(\,\mathrm{D} \mid \mathrm{M}_k\,)$ is the probability of data D, given model M_k
3. $\hat{\theta}_k$ are the parameters of the *k*th component in the mixture

Within the *mclust* R package, the BIC values are calculated for different univariate and multivariate mixtures. Scrucca et. al. (2016) nicely summarizes the models names and their respective traits.

Mixture Type	*mclust* Model Name	Distribution	Volume	Shape	Orientation
Univariate	E	Equal variance	-	-	-
	V	Variable / unequal variance	-	-	-
Multivariate	EII	Spherical	Equal	Equal	-
	VII	Spherical	Variable	Equal	-
	EEI	Diagonal	Equal	Equal	Coordinate Axes
	VEI	Diagonal	Variable	Equal	Coordinate Axes
	EVI	Diagonal	Equal	Variable	Coordinate Axes
	VVI	Diagonal	Variable	Variable	Coordinate Axes
	EEE	Ellipsoidal	Equal	Equal	Equal
	VEE	Ellipsoidal	Variable	Equal	Equal
	EVE	Ellipsoidal	Equal	Variable	Equal
	VVE	Ellipsoidal	Variable	Variable	Equal
	EEV	Ellipsoidal	Equal	Equal	Variable
	VEV	Ellipsoidal	Variable	Equal	Variable
	EVV	Ellipsoidal	Equal	Variable	Variable
	VVV	Ellipsoidal	Variable	Variable	Variable

Table 2. Parameterisations of the within-group covariance matrix Σk for multidimensional data available in the mclust package, and the corresponding geometric characteristics.

<u>GMM-T (Gaussian Mixture Model on the as-is form of traditional variable manipulation)</u>
Similarly, the optimal number of clusters needs to be computed here. These are then directly ingested into the OSI Node called 'OSI5-GMM-T-Explore.' Calling the *mclust* package, a function to record the BIC value of a particular cluster *k* is created, and then looped through cluster counts 2 to 20. Again, this is a much lower initial cluster exploration range, compared to the *k*-means count range of 2 to 50. This is due to the long processing time, as the cluster count increases. These BIC values are then plotted within EM, as seen below.

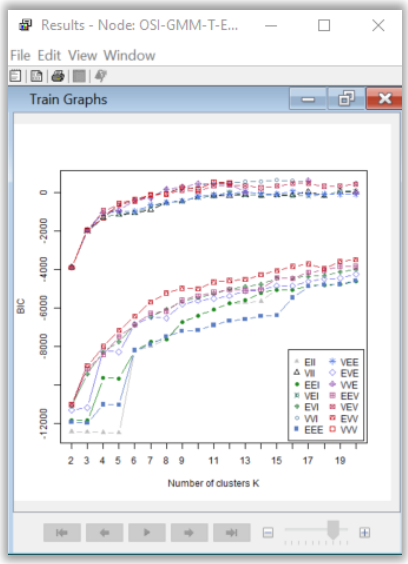


Figure 20:
Silhouette Index for GMM-T output

The BIC values generally improve as the number of clusters *k* increases. Using VEV values as recommended by the *mclust* package, cluster 8 had the highest BIC values. Thus, we will regard *k* = 8 as its most optimized cluster count.

Another OSI Node, called 'OSI19-GMM-T-8-Clusters,' is now attached to the Data_All Node. The *k* = 8 cluster output will be generated together with its histogram plots. Analysis of these results will also be covered in a later section.

<u>GMM-R (Gaussian Mixture Model on the as-is form of 2 reduct inputs)</u>
Similarly, the optimal number of clusters needs to be computed here. These are then directly ingested into the OSI Node called 'OSI6-GMM-R-Explore.' Calling the *mclust* package, a function to record the BIC value of a particular cluster *k* is created, and then looped through cluster counts 2 to 20. Again, this is a much lower initial cluster exploration range, compared to the *k*-means count range of 2 to 50. This is due to the long processing time, as the cluster count increases. These BIC values are then plotted within EM, as seen below.

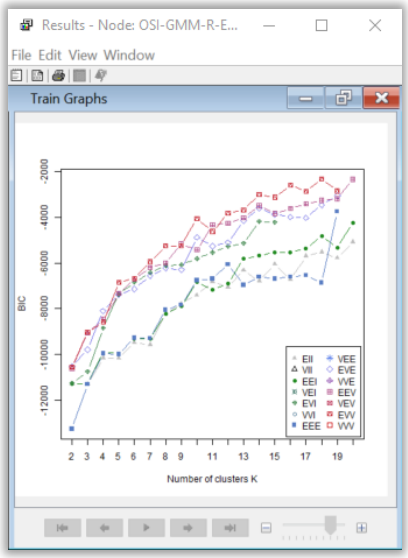


Figure 21:
Silhouette Index for GMM-R output

Similarly, the BIC values generally improve as the number of clusters *k* increases. Using EVV values as recommended by the *mclust* package, cluster 19 had the highest BIC values. Thus, we will regard k = 19 as its most optimized cluster count.

Another OSI Node, called 'OSI20-GMM-T-19-Clusters,' is now attached to the Data_All Node. The *k* = 19 cluster output will be generated together with its histogram plots. Analysis of these results will also be covered in a later section.

5.0 Analysis and Results

5.1 Overview of Clustering Output Results

After identifying the optimal cluster counts for each output, this section covers the distribution of each cluster, as well as explore its individual characteristics. Tweaks to the outputs may be explored as well, where needed. Figure below shows the outputs for exploration.

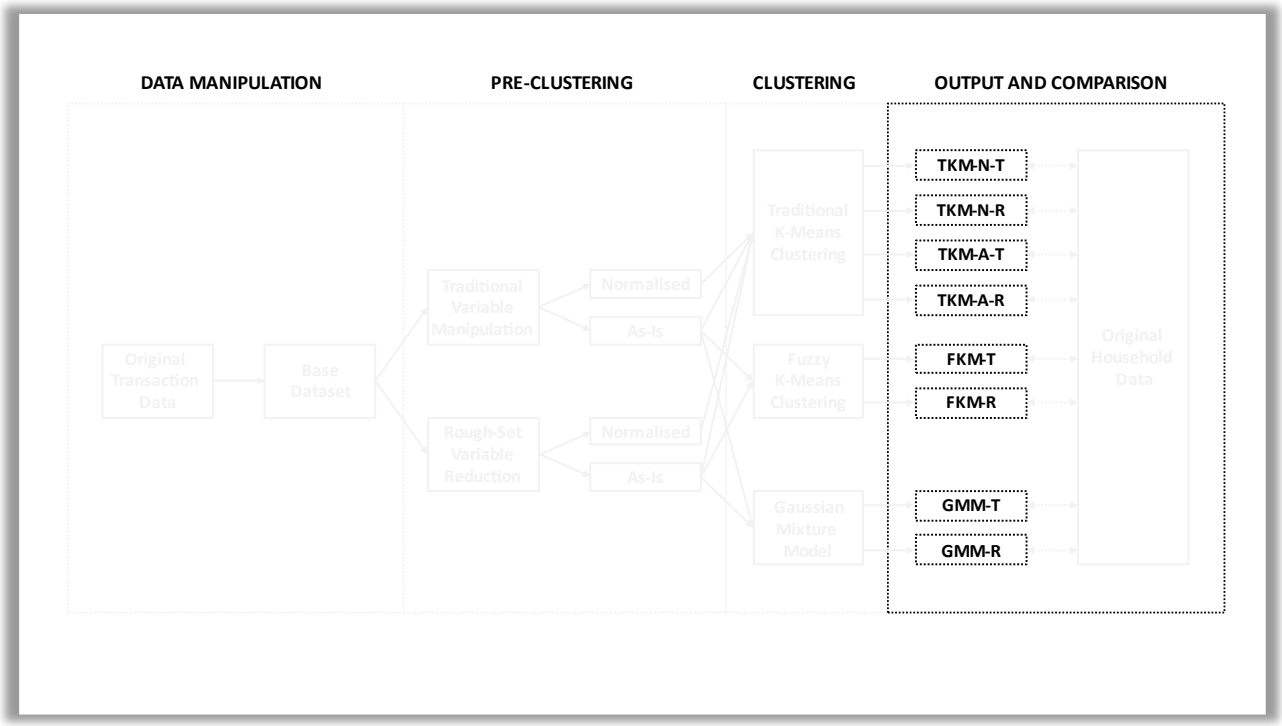


Figure 22: Capstone Workflow with emphasis on current step

Most of the optimal cluster counts hovered at the 2-cluster mark. This was generally true for both the traditional and fuzzy *k*-means clustering outputs, except for the TKM-A-T. GMM produced a much higher cluster count at 8 and 19 respectively.

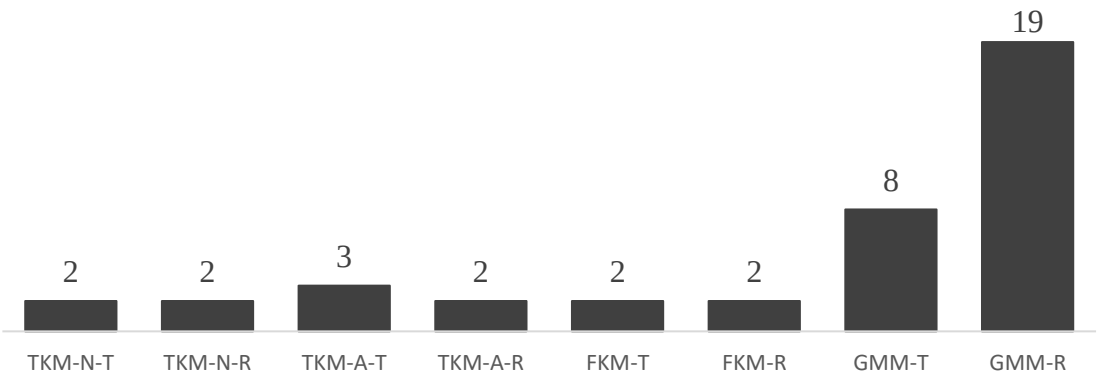


Figure 23: Optimal Cluster Counts for Each Clustering Output

Table 3 shows the various output distribution within each cluster. It would appear that both the traditional and fuzzy *k*-means output, from a reduct dataset, had very low distribution in their first cluster, when compared to their second. Coversely, there is a more even distribution for the same clustering outputs, from a regular dataset. For the GMM outputs, the reduct dataset yielded a very uneven spread, compared to the regular dataset.

Cluster No.	TKM-N-T	TKM-N-R	TKM-A-T	TKM-A-R	FKM-T	FKM-R	GMM-T	GMM-R
1	1,287	143	774	74	1,421	100	283	389
2	1,183	2,327	1,452	2,396	1,049	2,370	504	11
3	-	-	244	-	-	-	298	73
4	-	-	-	-	-	-	146	40
5	-	-	-	-	-	-	274	1,759
6	-	-	-	-	-	-	552	51
7	-	-	-	-	-	-	368	10
8	-	-	-	-	-	-	45	10
9	-	-	-	-	-	-	-	6
10	-	-	-	-	-	-	-	22
11	-	-	-	-	-	-	-	6
12	-	-	-	-	-	-	-	9
13	-	-	-	-	-	-	-	16
14	-	-	-	-	-	-	-	22
15	-	-	-	-	-	-	-	8
16	-	-	-	-	-	-	-	15
17	-	-	-	-	-	-	-	7
18	-	-	-	-	-	-	-	3
19	-	-	-	-	-	-	-	13

Table 3. Distribution of 2,470 IDs across the various clustering outputs

The clustering outputs, together with their respective cluster traits, will be explored next. Given that there's 28 variables to consider, only 9 of the more significant variables will be included here for each output.

5.2 TKM Clustering Output Results

TKM-N-T (Traditional *k*-means clustering on normalized form of 16 non-collinear inputs)
Cluster 1 shoppers are a good mixture of frequent and infrequent shoppers. They are price sensitive, since they use a large number of discounts and sales per quantity are generally on the lower side. Their total spending is sizable.

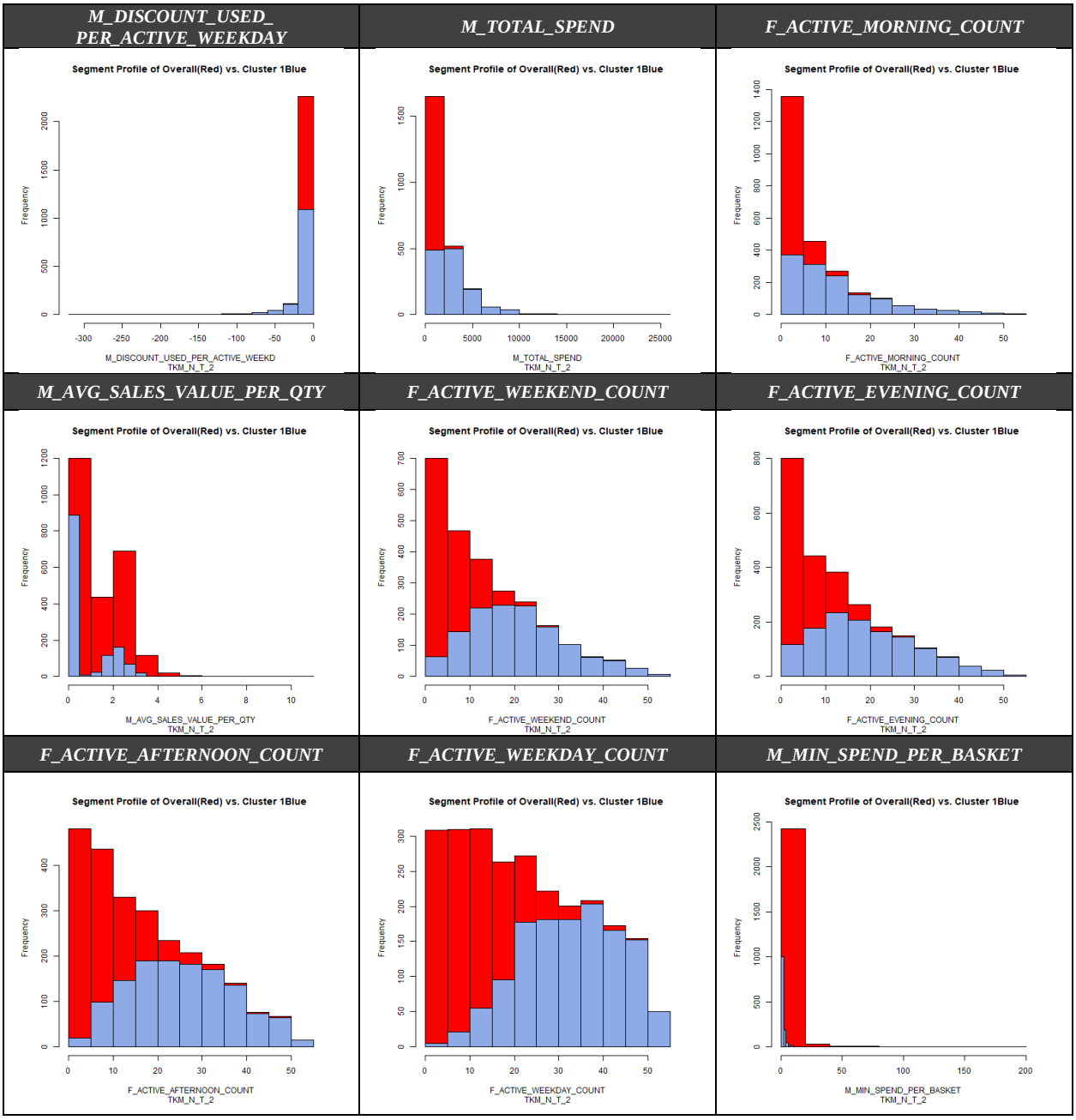


Figure 24: 9 Significant Attributes for Cluster 1 of TKM-N-T

Cluster 2 shoppers are somewhat infrequent shoppers. They are not as price sensitive as they tend to use less discounts on weekdays. Their total spending is much lower than Cluster 1 shoppers. Given their infrequency, they tend to spend slightly more for each basket.

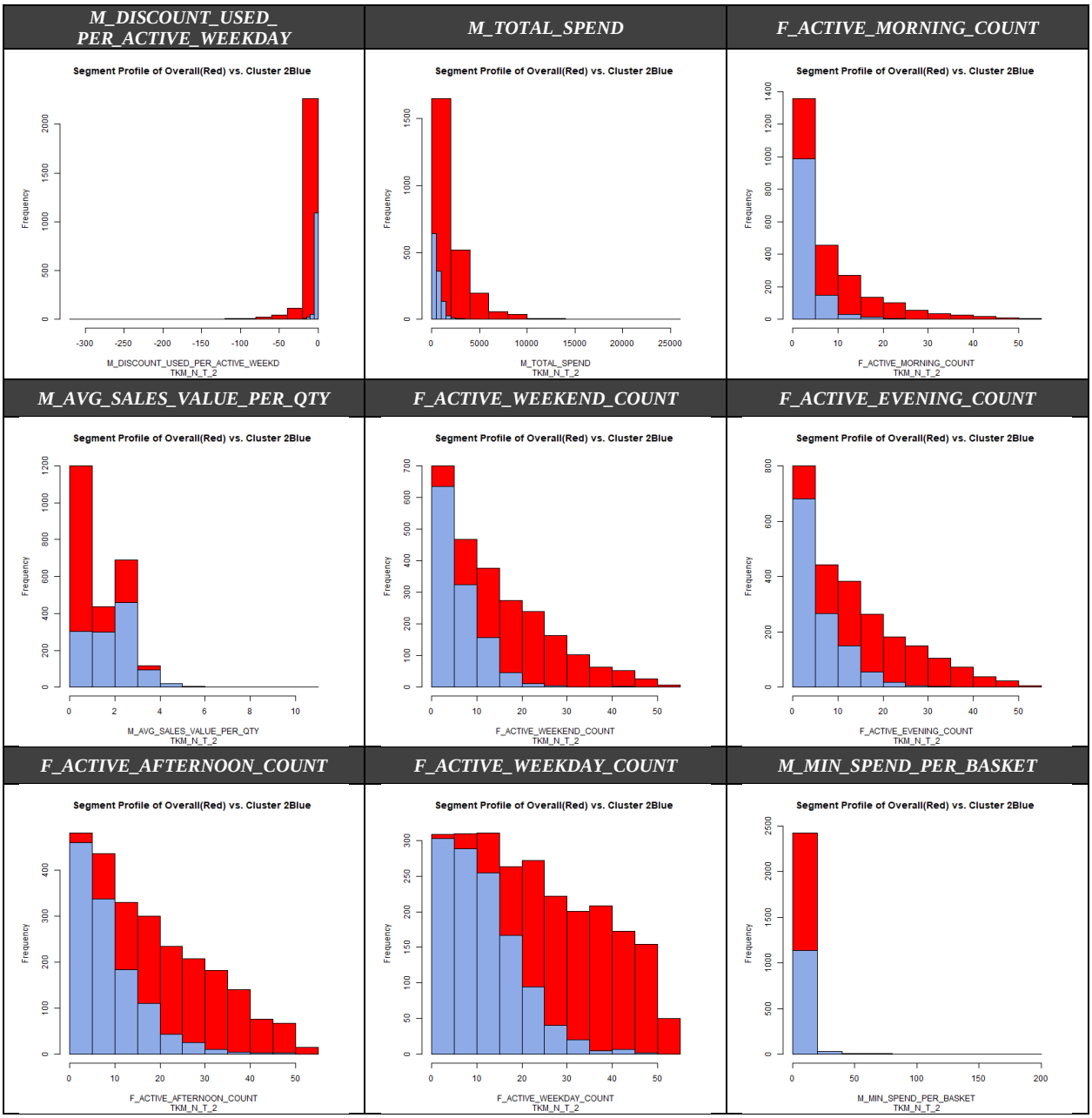


Figure 25: 9 Significant Attributes for Cluster 2 of TKM-N-T

TKM-A-T (Traditional *k*-means clustering on as-is form of 16 non-collinear inputs)

Cluster 1 shoppers tend to be a more frequent shopper with a preference for weekday shopping. They spend more overall, and yet are price conscious due to their use of discounts. Given their frequency, they spend less for each basket.

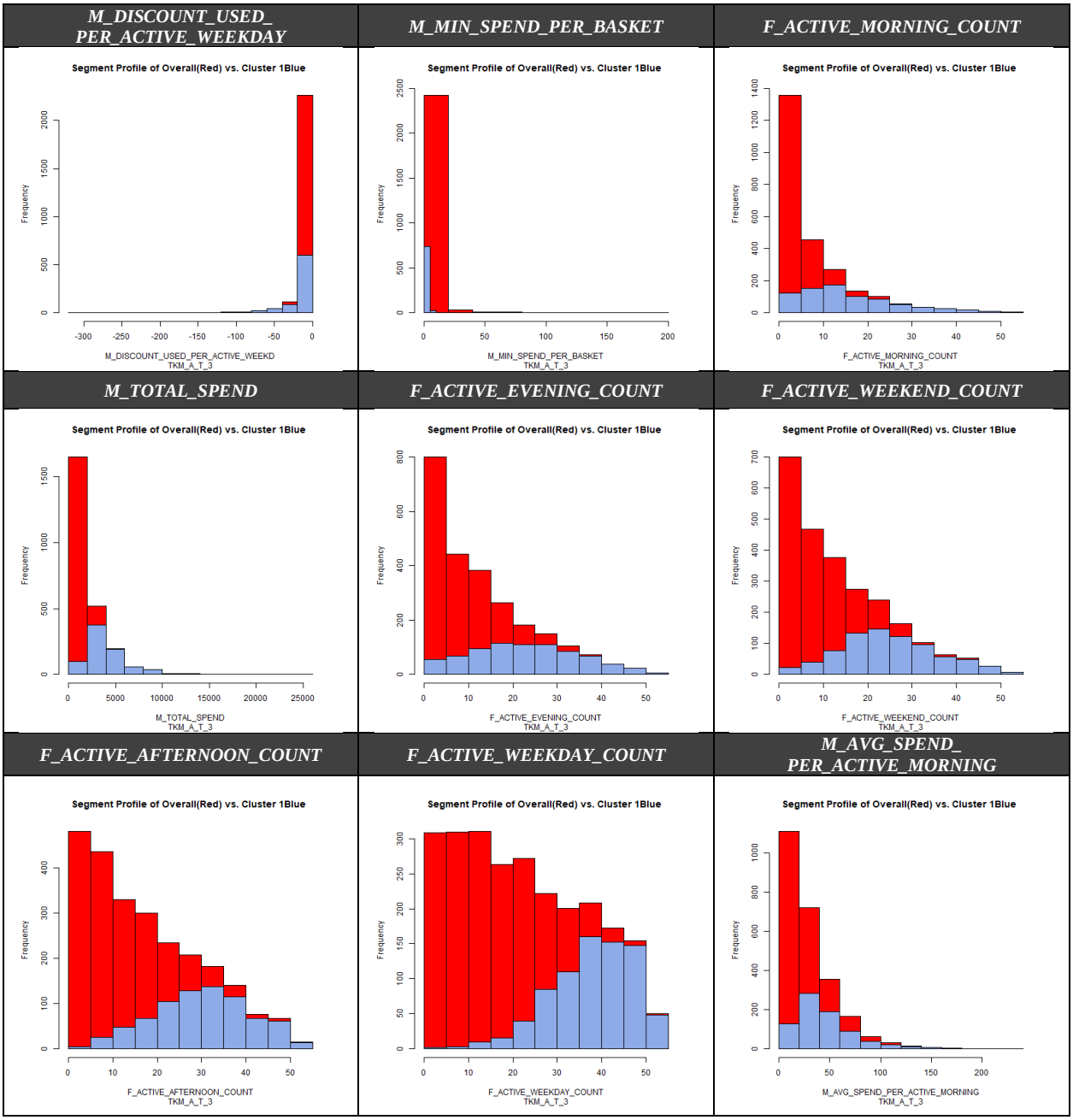


Figure 26: 9 Significant Attributes for Cluster 1 of TKM-A-T

Cluster 2 shoppers are somewhat infrequent shoppers. They spend less each time, and hardly make use of discounts on weekdays. Overall spending is also on the lower side

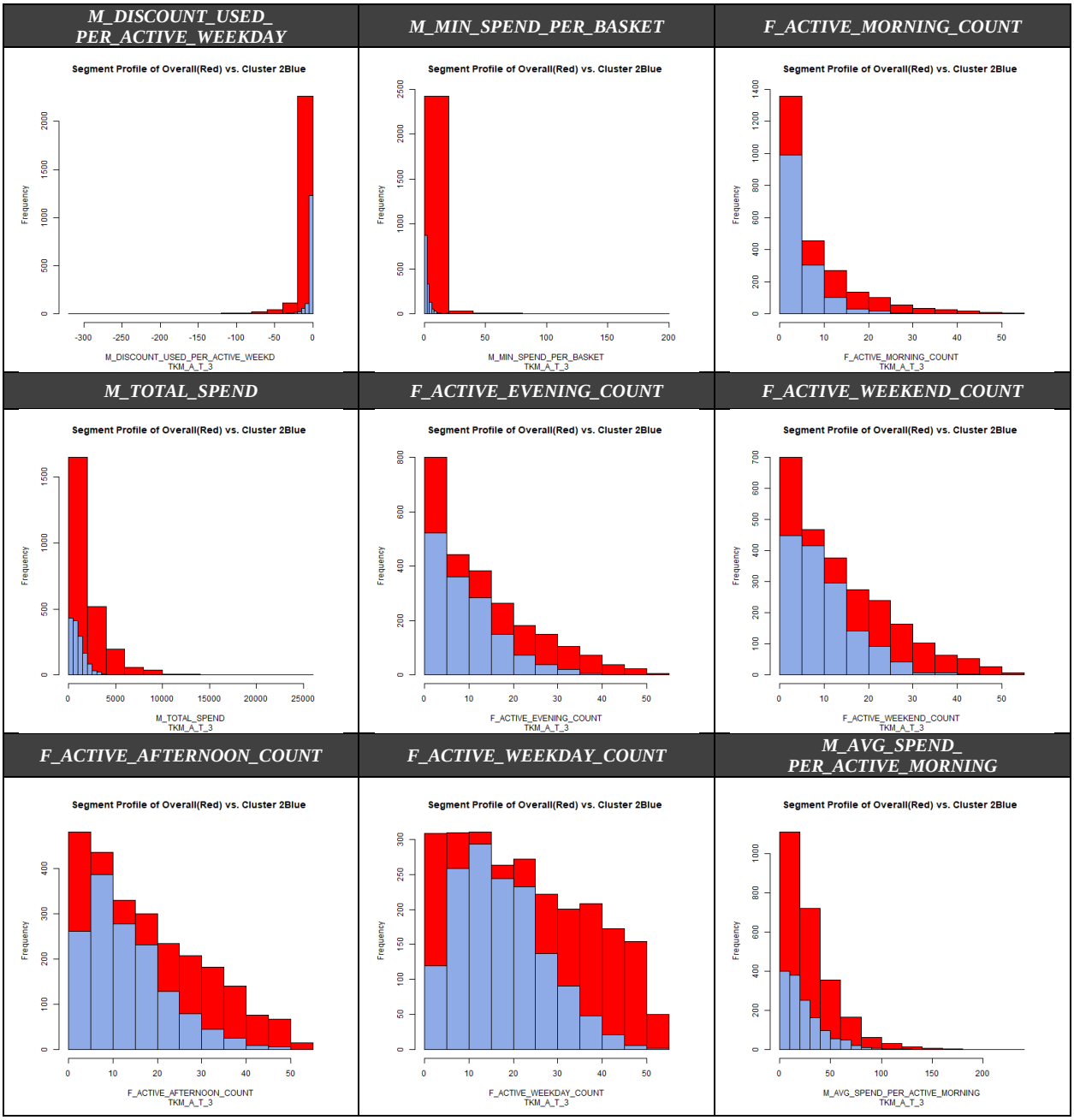


Figure 27: 9 Significant Attributes for Cluster 2 of TKM-A-T

Cluster 3 shoppers are highly infrequent shoppers. They tend to use little to no discounts on weekdays and spend less overall than Cluster 2 shoppers. They are much less active through the week. Given their frequency, they tend to spend much more for each basket.

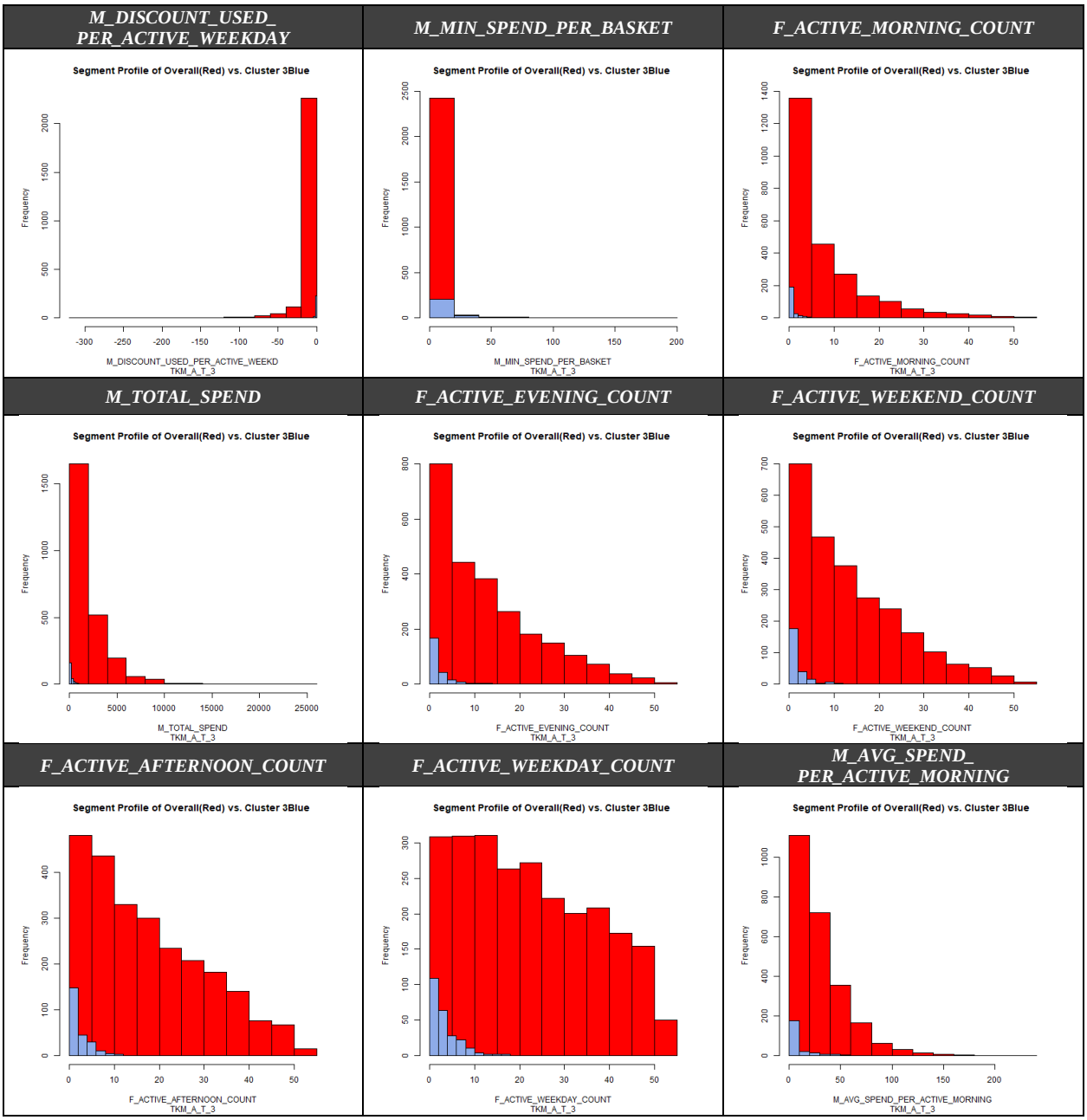


Figure 28: 9 Significant Attributes for Cluster 3 of TKM-A-T

<u>TKM-N-R (Traditional *k*-means clustering on normalized form of 2 reduct inputs)</u>

Cluster 1 shoppers are a good spread of frequent and infrequent shoppers. They are price conscious and have a propensity to use more discounts. They don't spend much overall. Given their frequency, they spend less for each basket.

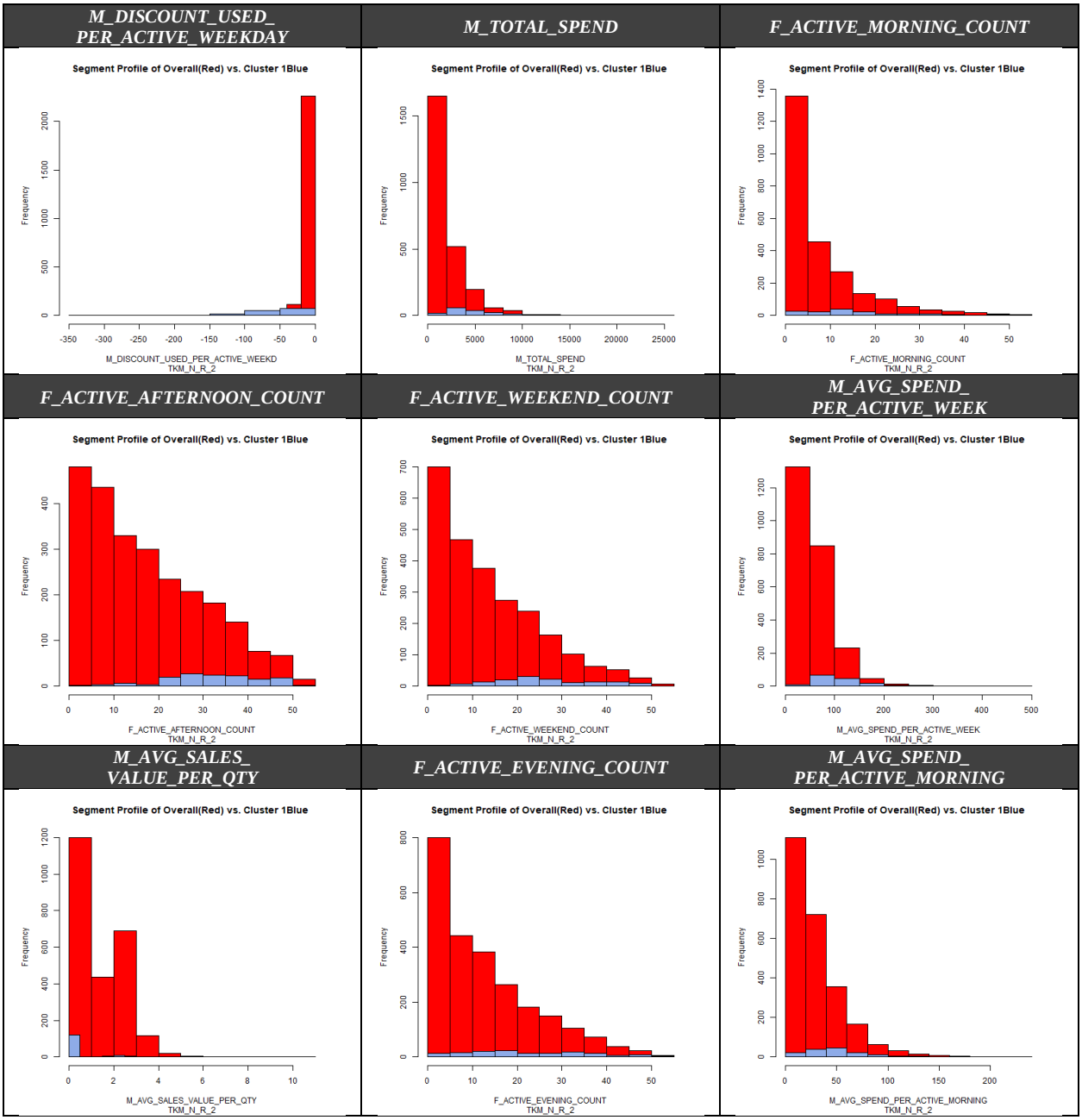


Figure 29: 9 Significant Attributes for Cluster 1 of TKM-N-R

Cluster 2 shoppers are highly frequent shoppers with deeper pockets. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each basket.

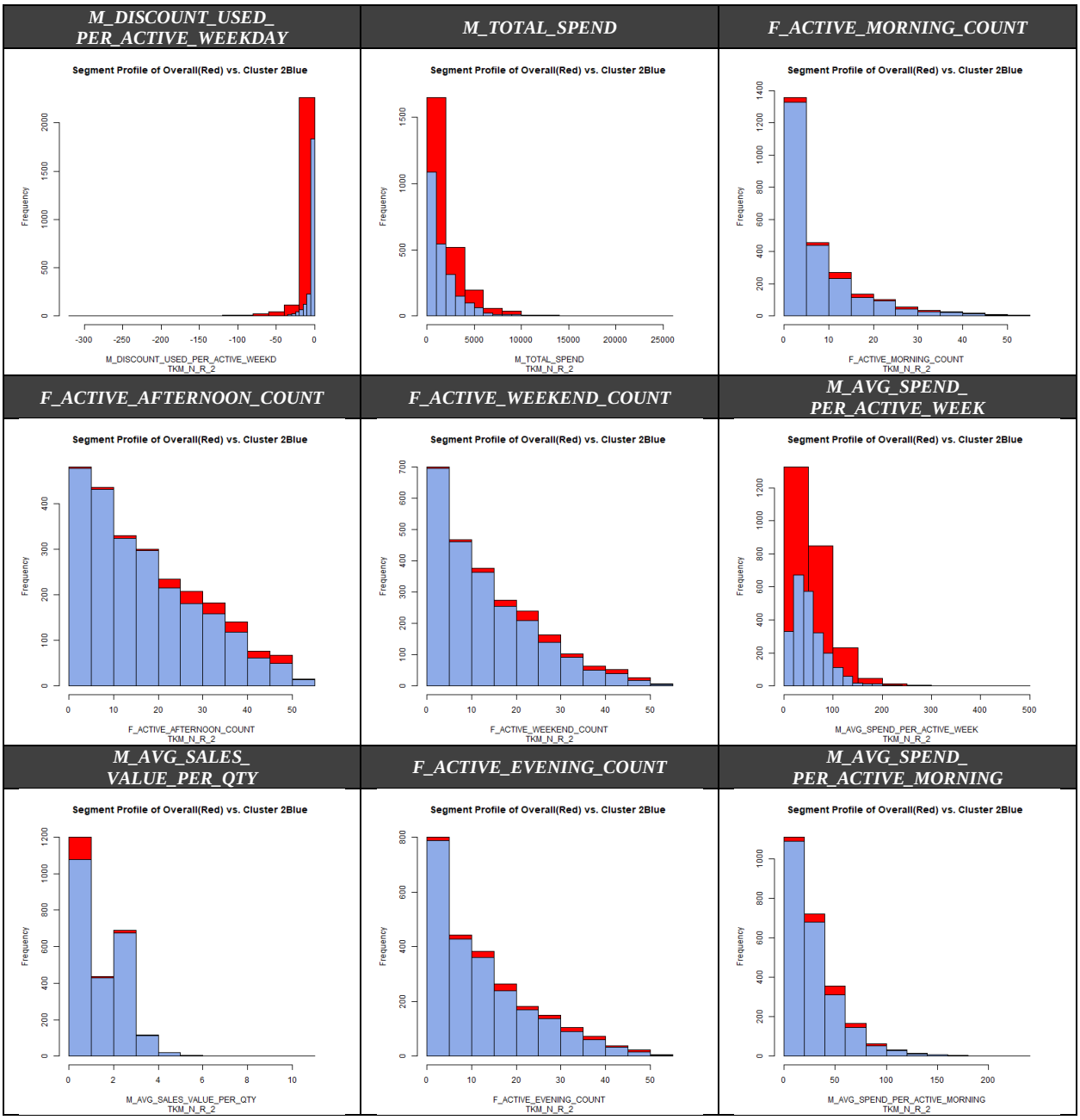


Figure 30: 9 Significant Attributes for Cluster 2 of TKM-N-R

<u>TKM-A-R (Traditional *k*-means clustering on as-is form of 2 reduct inputs)</u>
Cluster 1 shoppers are highly infrequent shoppers and are price conscious (with low average sales per quantity). They have a propensity not to use any discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.

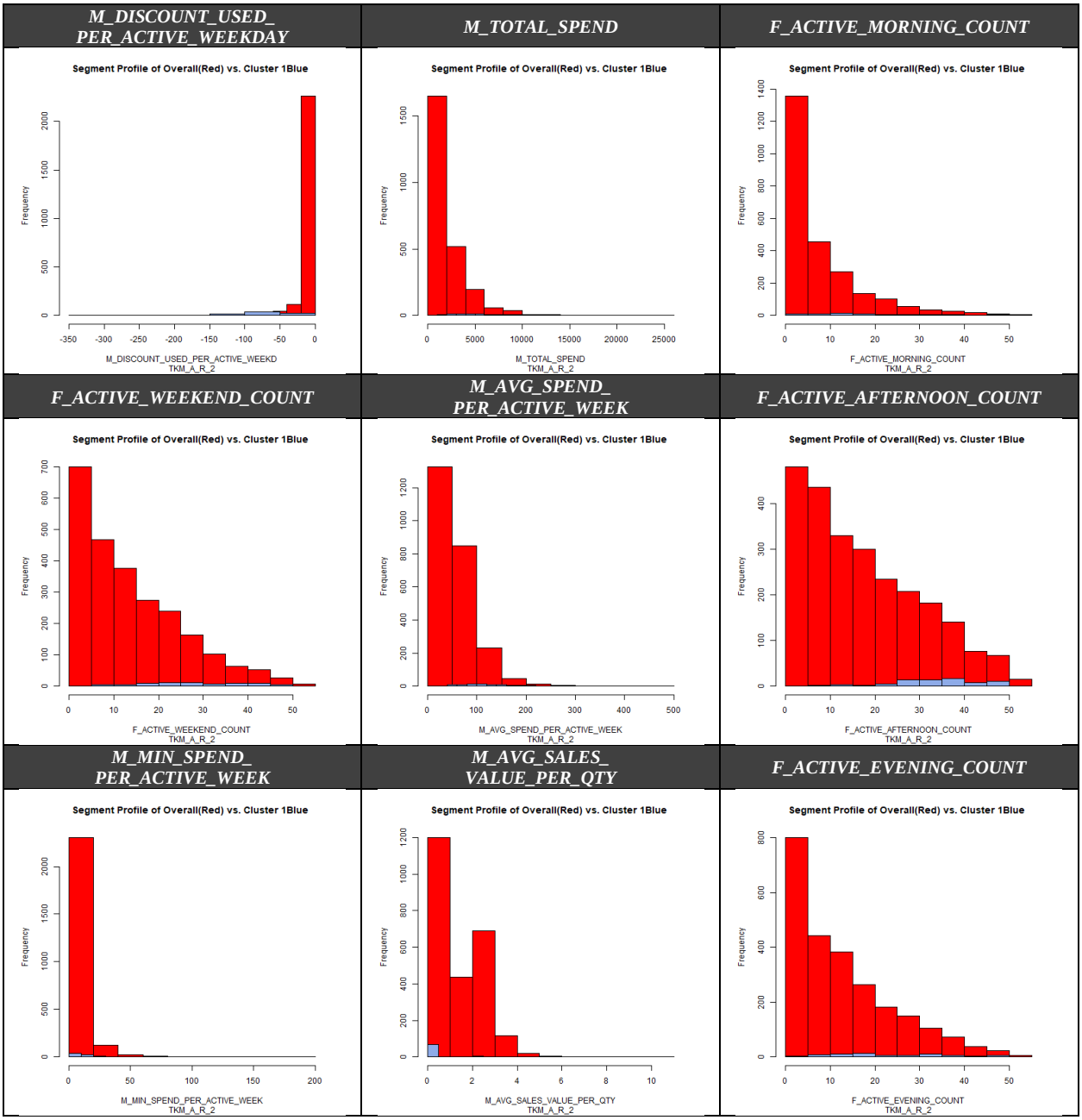


Figure 31: 9 Significant Attributes for Cluster 1 of TKM-A-R

Cluster 2 shoppers are highly frequent shoppers with deeper pockets. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each item.

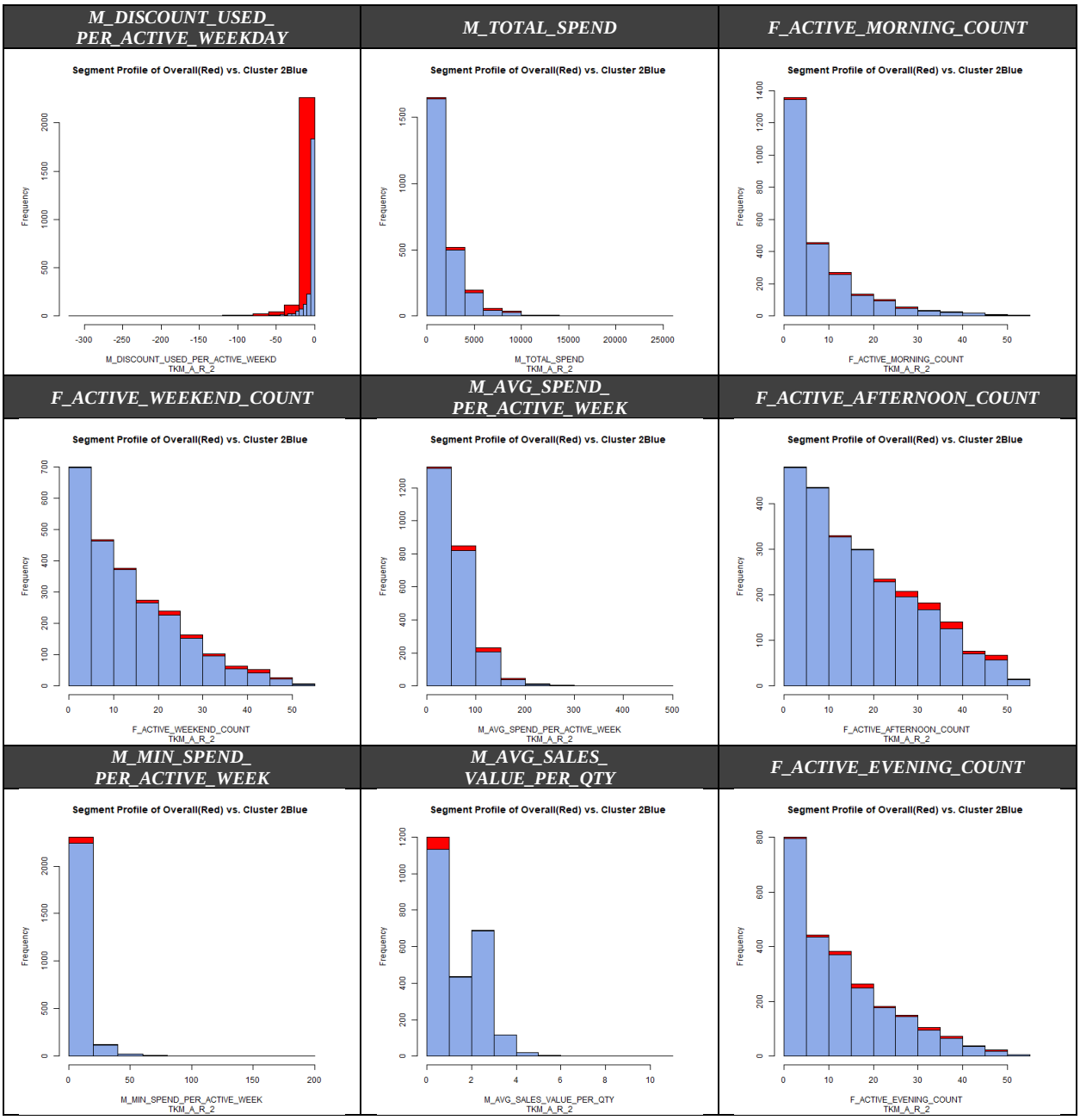


Figure 32: 9 Significant Attributes for Cluster 2 of TKM-A-R

5.3 FKM Clustering Output Results

FKM-T (Fuzzy *k*-means clustering on the as-is form of traditional variable manipulation)
Cluster 1 shoppers are somewhat infrequent shoppers but are not price conscious. They have a propensity not to use any discounts. They don't spend much overall, but when they do, they don't mind paying more for each product. They somewhat shop all week around.

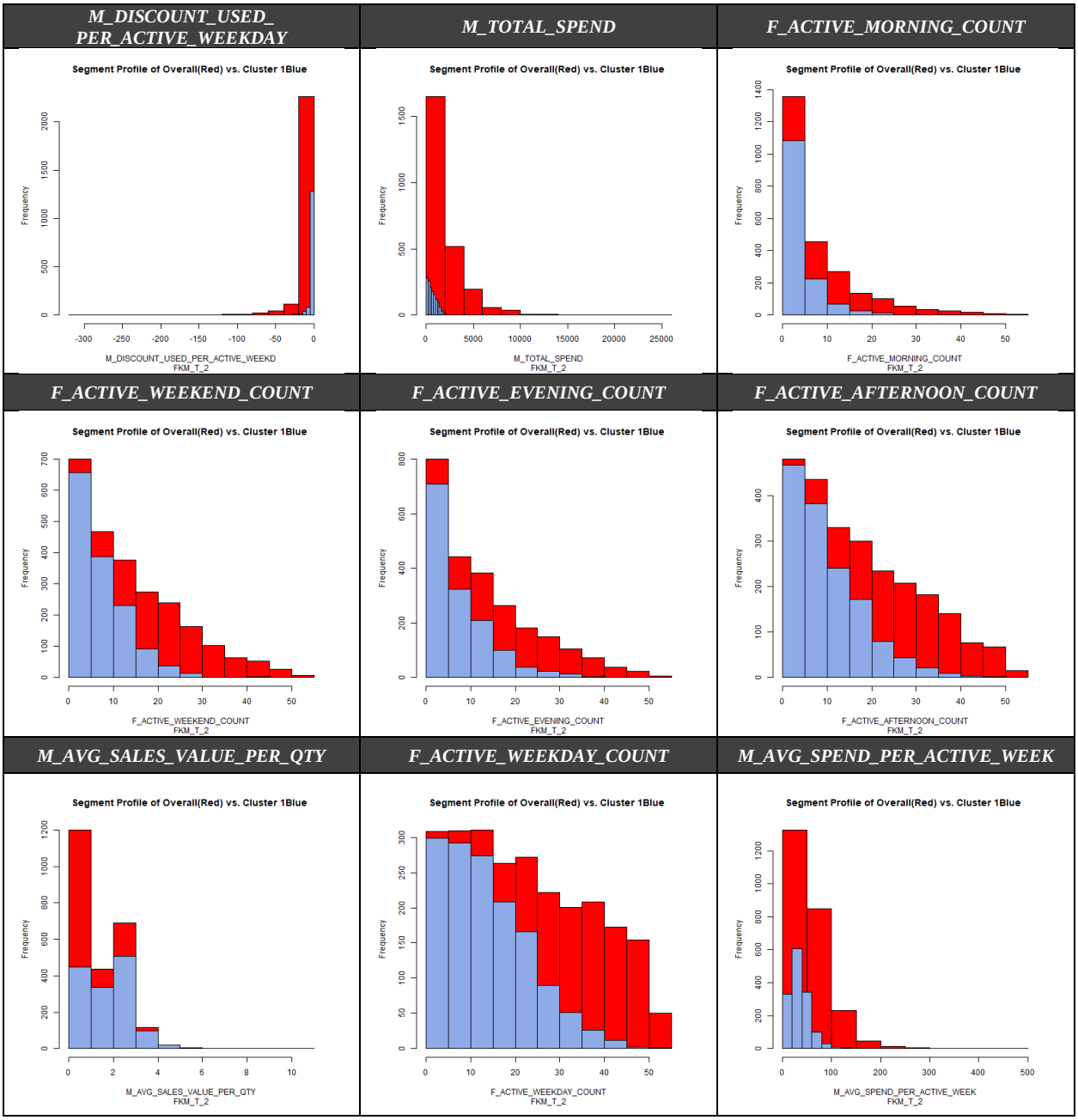


Figure 33: 9 Significant Attributes for Cluster 1 of FKM-T

Cluster 2 shoppers are a good mix of frequent and infrequent shoppers. They spend a lot and have no qualms about using discounts. This leads to a lower sales value per quantity. They actively shop all week around.

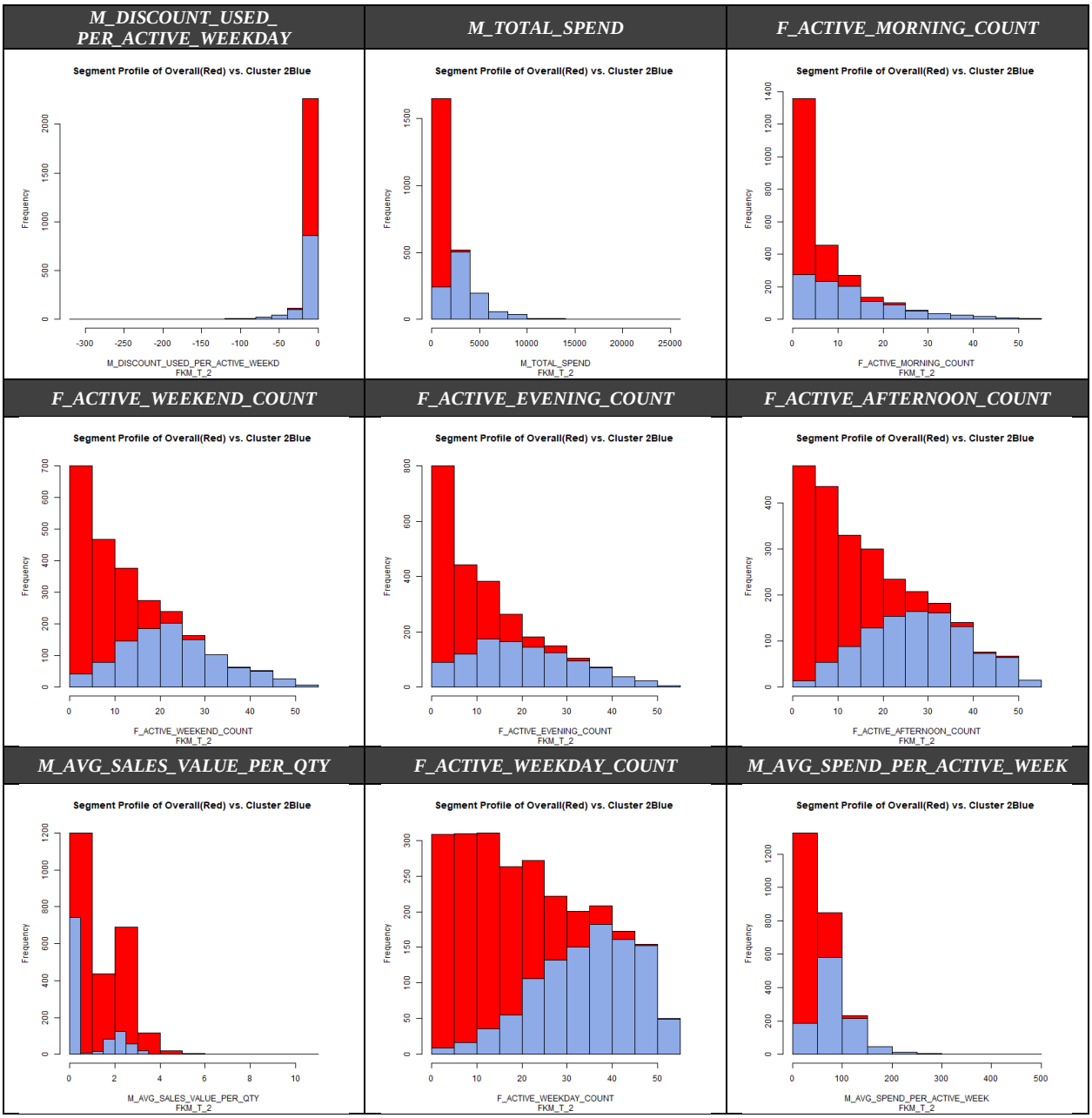


Figure 34: 9 Significant Attributes for Cluster 2 of FKM-T

FKM-R (Fuzzy *k*-means clustering on the as-is form of 2 reduct inputs)

Cluster 1 shoppers are a good mix of frequent and infrequent shoppers and don't spend much. When they do shopping, it's usually in the afternoons.

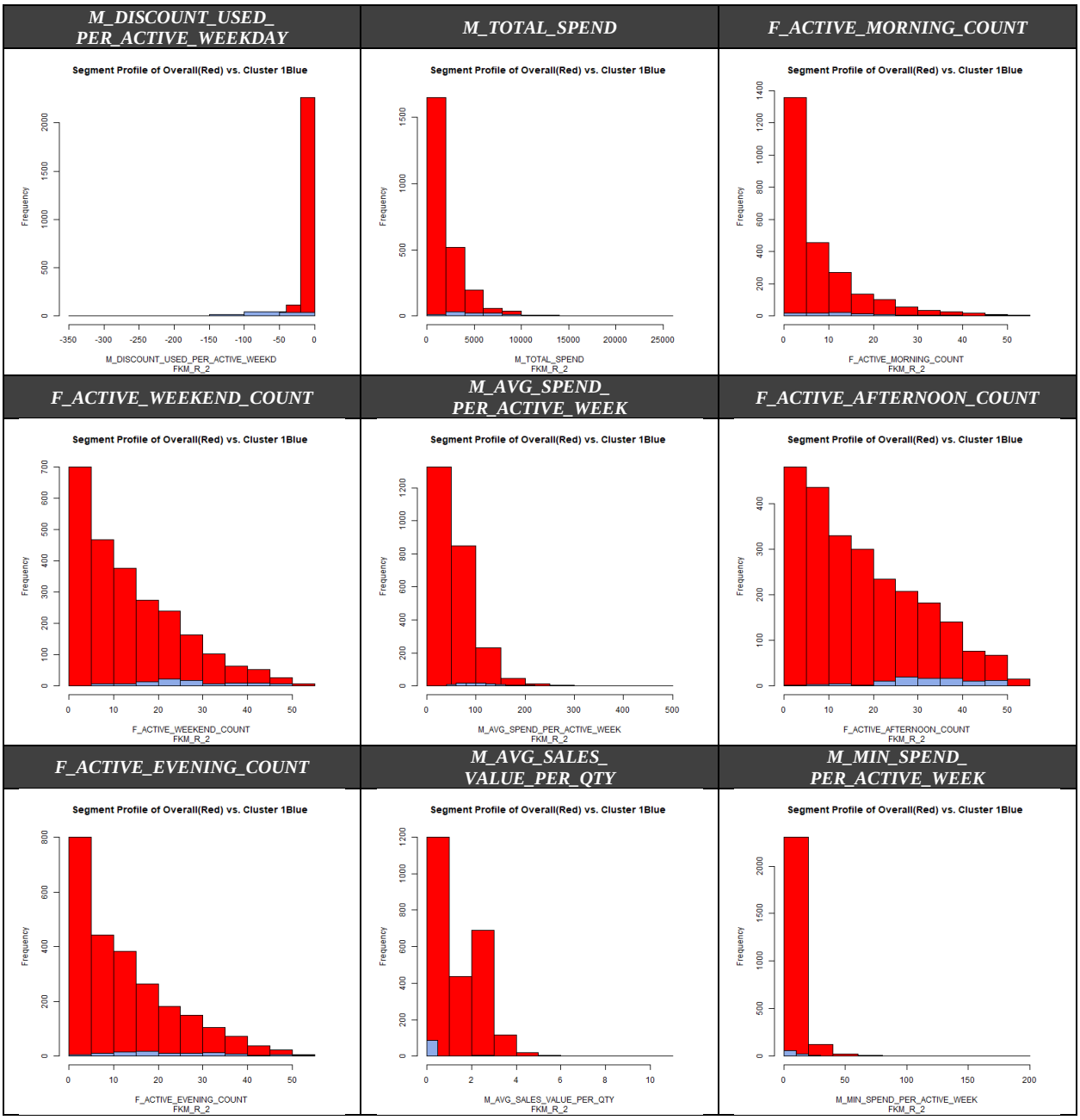


Figure 35: 9 Significant Attributes for Cluster 1 of FKM-R

Cluster 2 shoppers are highly frequent shoppers with deep pockets. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each item.

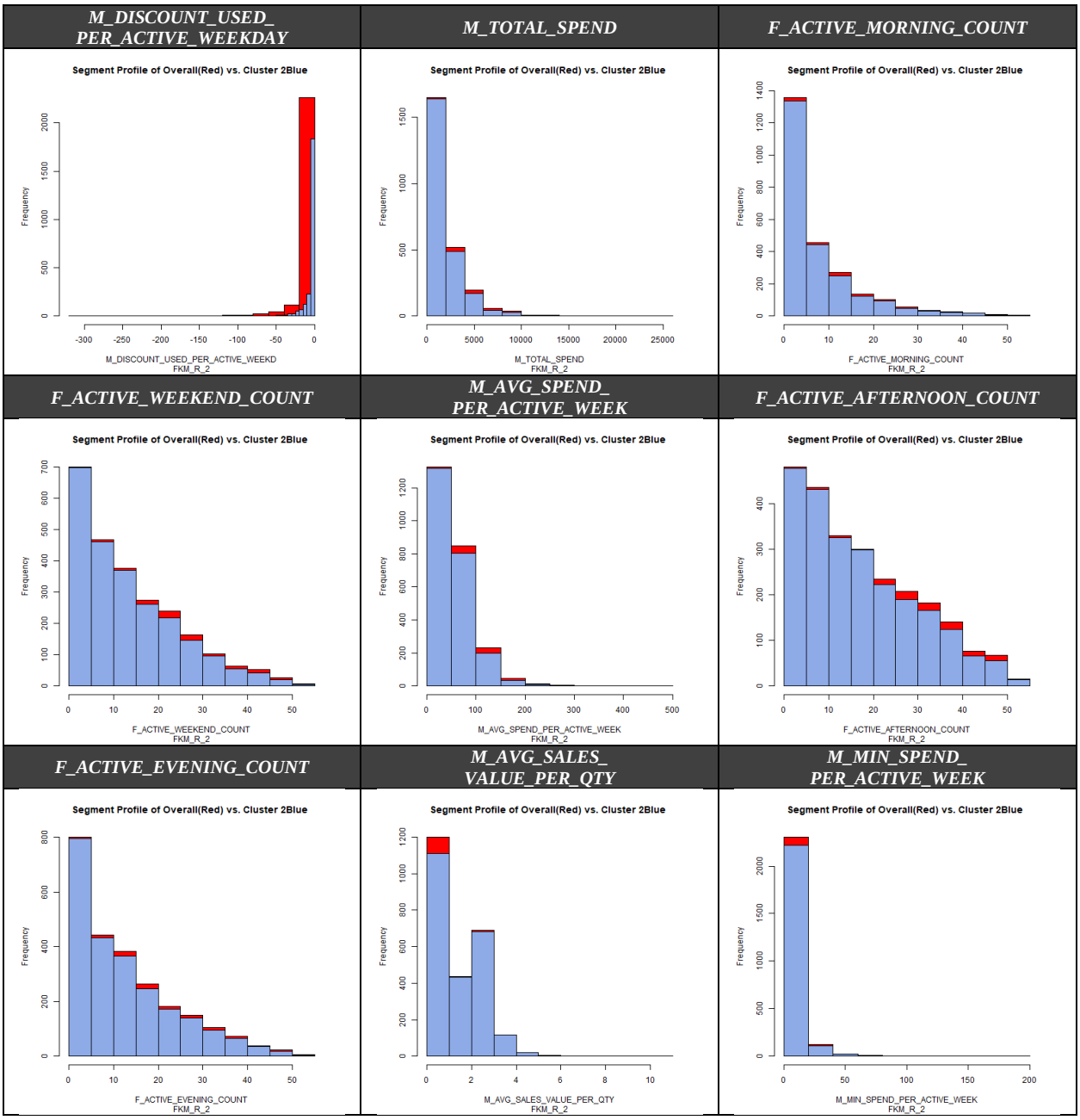


Figure 36: 9 Significant Attributes for Cluster 2 of FKM-R

5.4 Revisiting FKM-R

Even though both the cluster attributes for FKM-R show very distinct differences, the split in IDs between both clusters, as per table 4, were not even. Cluster 1 had 100 IDs, while Cluster 2 had 2,370 IDs. This study aims to explore whether finding alternative cluster values would aid in having a better distribution of IDs across its clusters. Looking at Figure 37, cluster values < 9 all had a silhouette index value of above 0.85. With this, clusters, *k*=7 and *k*=9 were chosen, and rerun separately.

Looking through the results, the distribution did not get better. The lowest ID distribution in both *k* = 7 and *k* = 9 is 11, and the highest is 1,803.

Cluster No.	FKM-R-2	FKM-R-7	FKM-R-9
1	100	11	329
2	2,370	1,803	39
3	-	43	20
4	-	13	116
5	-	47	30
6	-	144	1,764
7	-	409	12
8	-	-	149
9	-	-	11

Table 4. Distribution of 2,470 IDs across the various clustering outputs

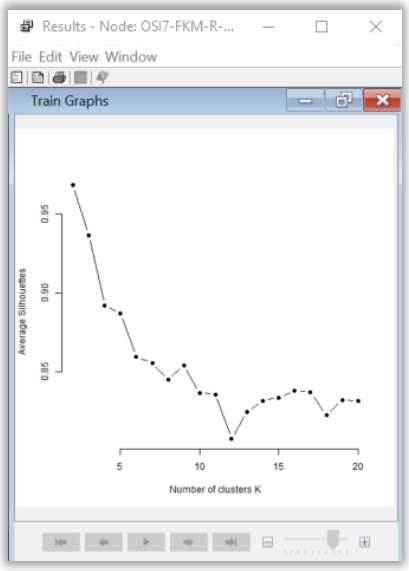


Figure 37:
Silhouette Index for FKM-R output

5.5 GMM Clustering Output Results

GMM-T (Gaussian Mixture Model on the as-is form of traditional variable manipulation)
Cluster 1 shoppers are highly infrequent shoppers and are price conscious. They have a propensity to use plenty of discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.

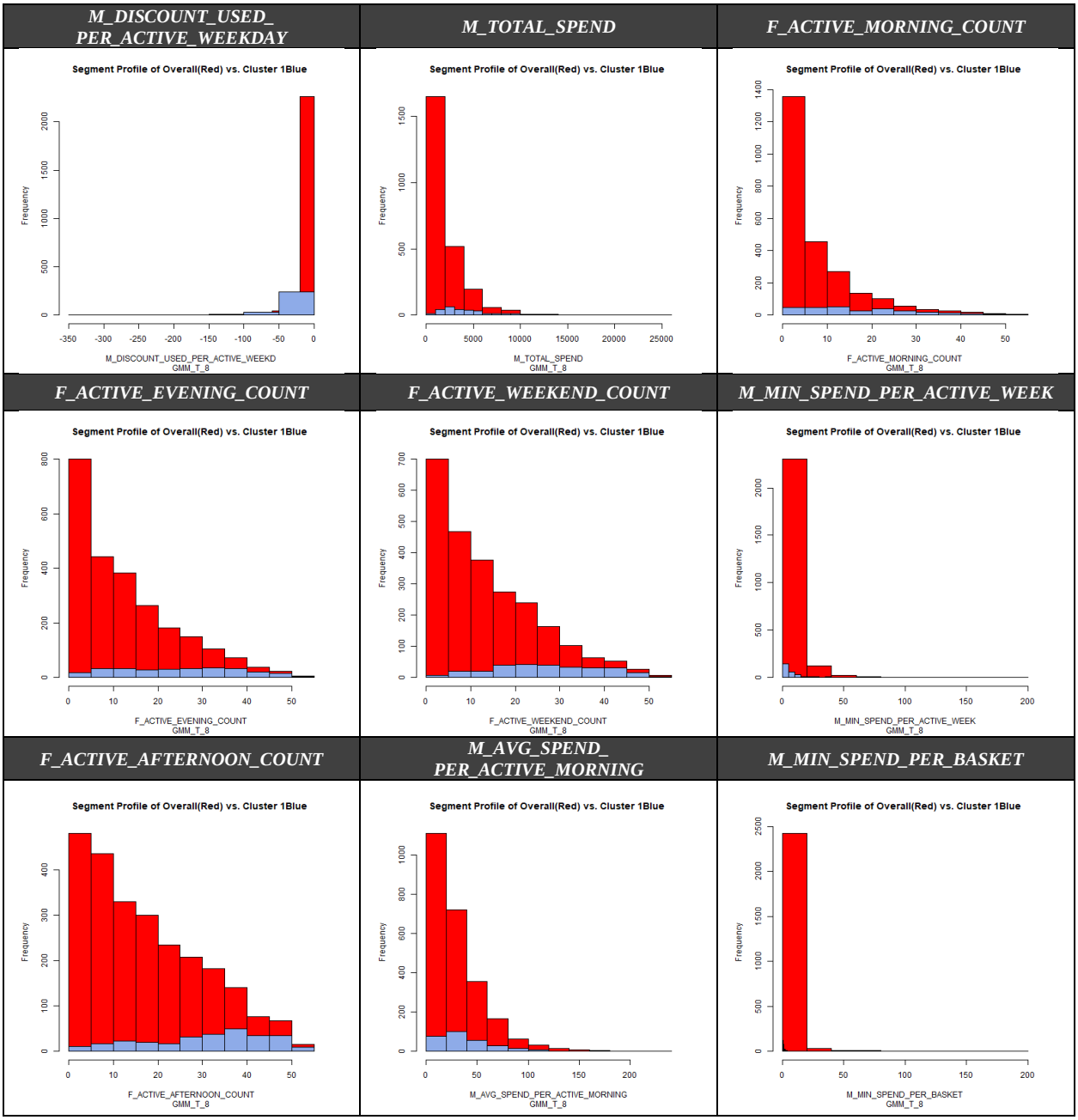


Figure 38: 9 Significant Attributes for Cluster 1 of GMM-T

Cluster 2 shoppers are somewhat frequent shoppers and are price conscious. They have a propensity not to use any discounts. They don't spend much overall. They prefer afternoons or evenings to shop.

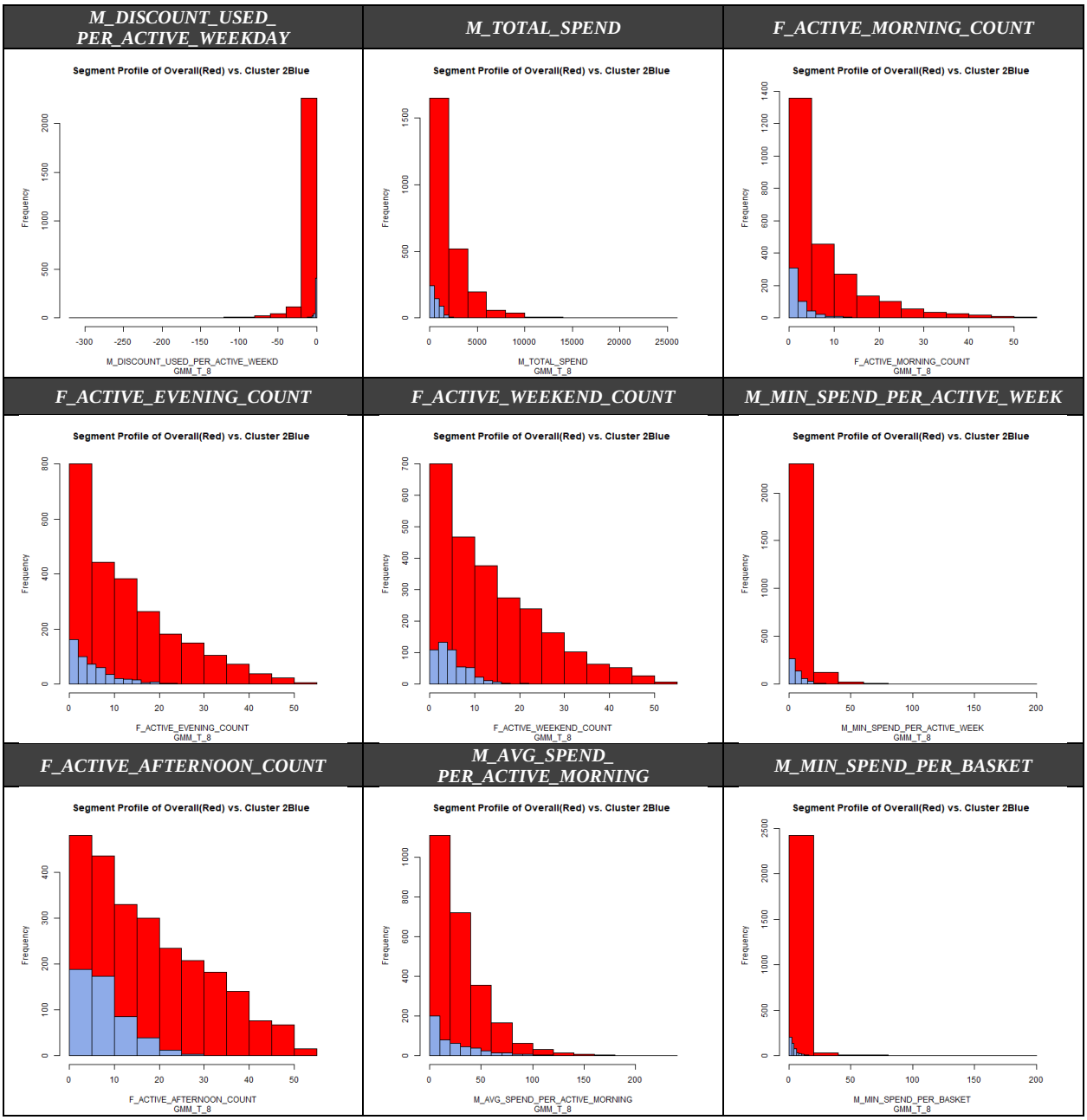


Figure 39: 9 Significant Attributes for Cluster 2 of GMM-T

Cluster 3 shoppers have a mixed spread of frequent and infrequent shoppers. They prefer not to use any discounts and spend very little overall.

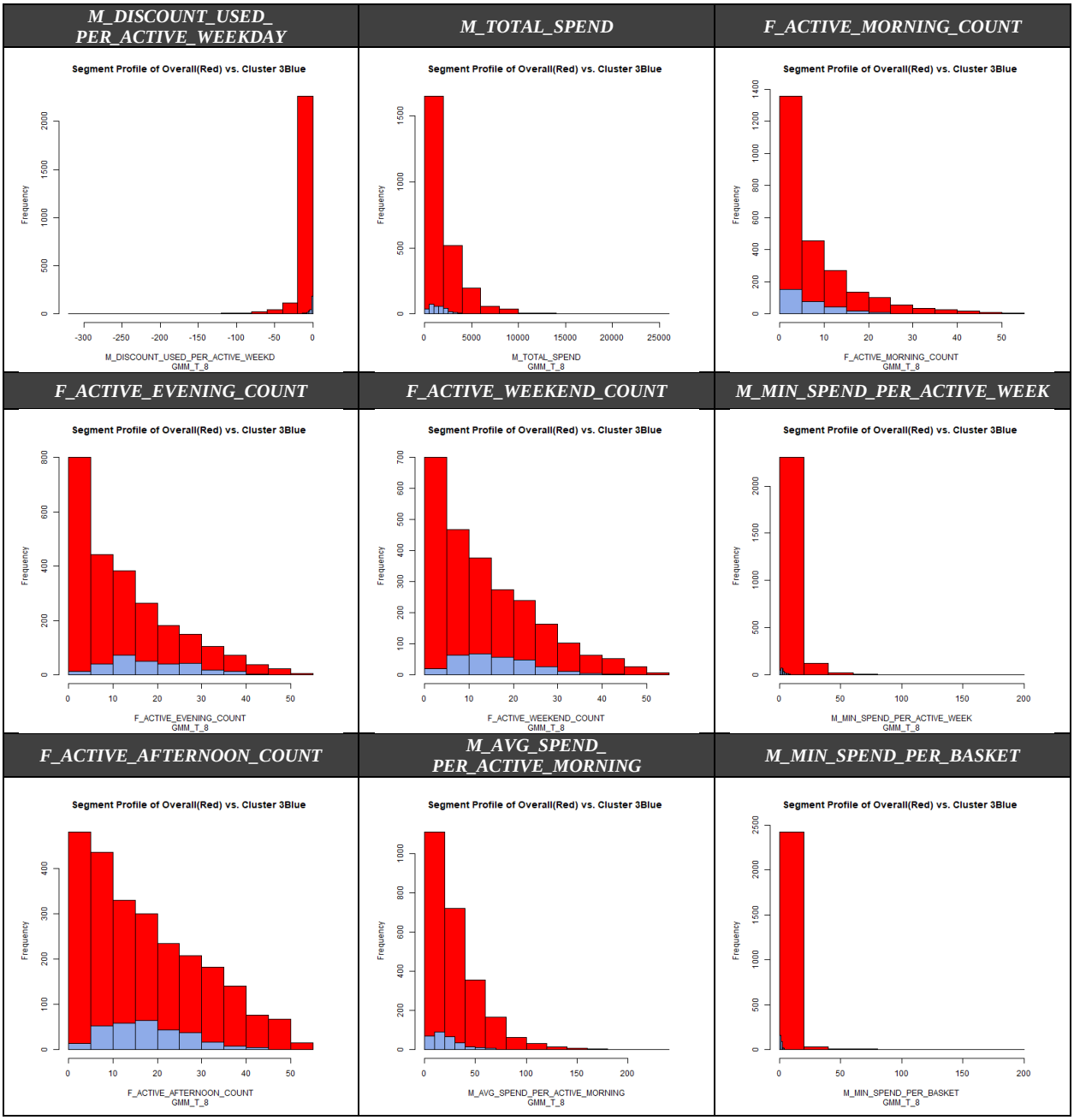


Figure 40: 9 Significant Attributes for Cluster 3 of GMM-T

Cluster 4 shoppers is a small group with a mix of frequent and infrequent shoppers. They have a tendency not to use any discounts.

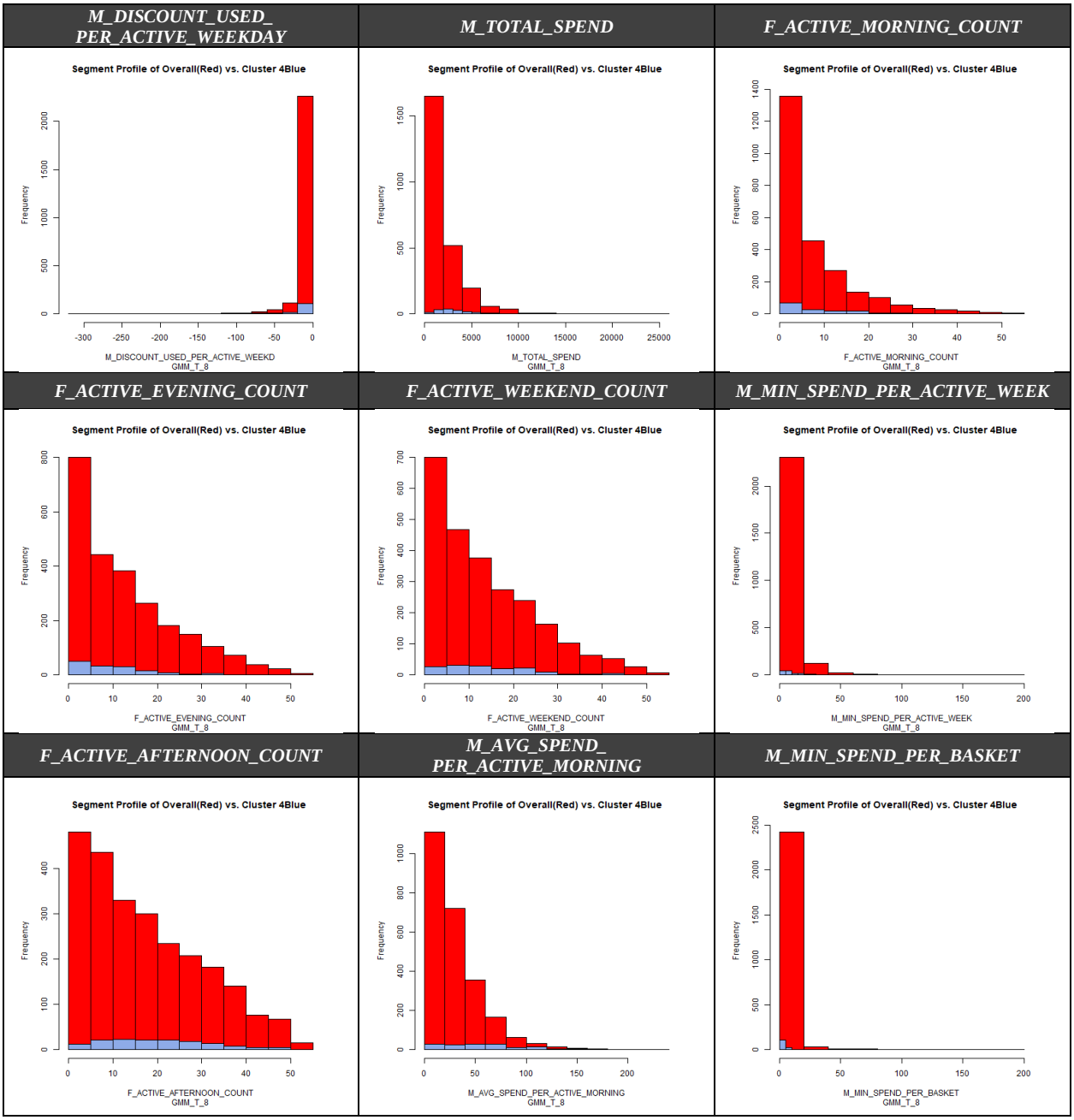


Figure 41: 9 Significant Attributes for Cluster 4 of GMM-T

Cluster 5 shoppers are highly infrequent shoppers and do not use any discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.

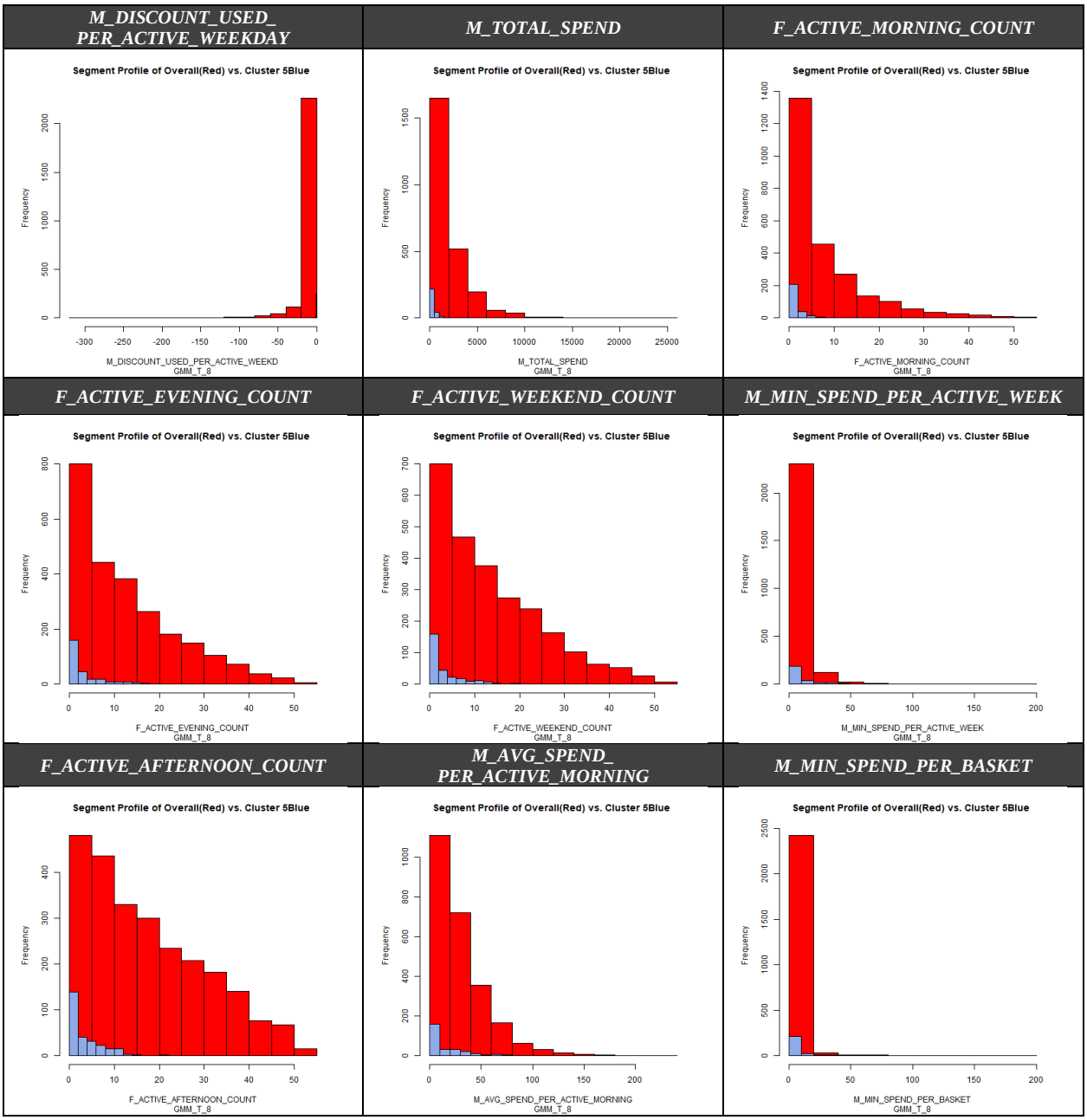


Figure 42: 9 Significant Attributes for Cluster 5 of GMM-T

Cluster 6 shoppers are highly frequent shoppers, who much prefer weekend shopping. They also prefer to shop either in the morning, afternoon, or evening. They don't spend much and hardly use discounts.

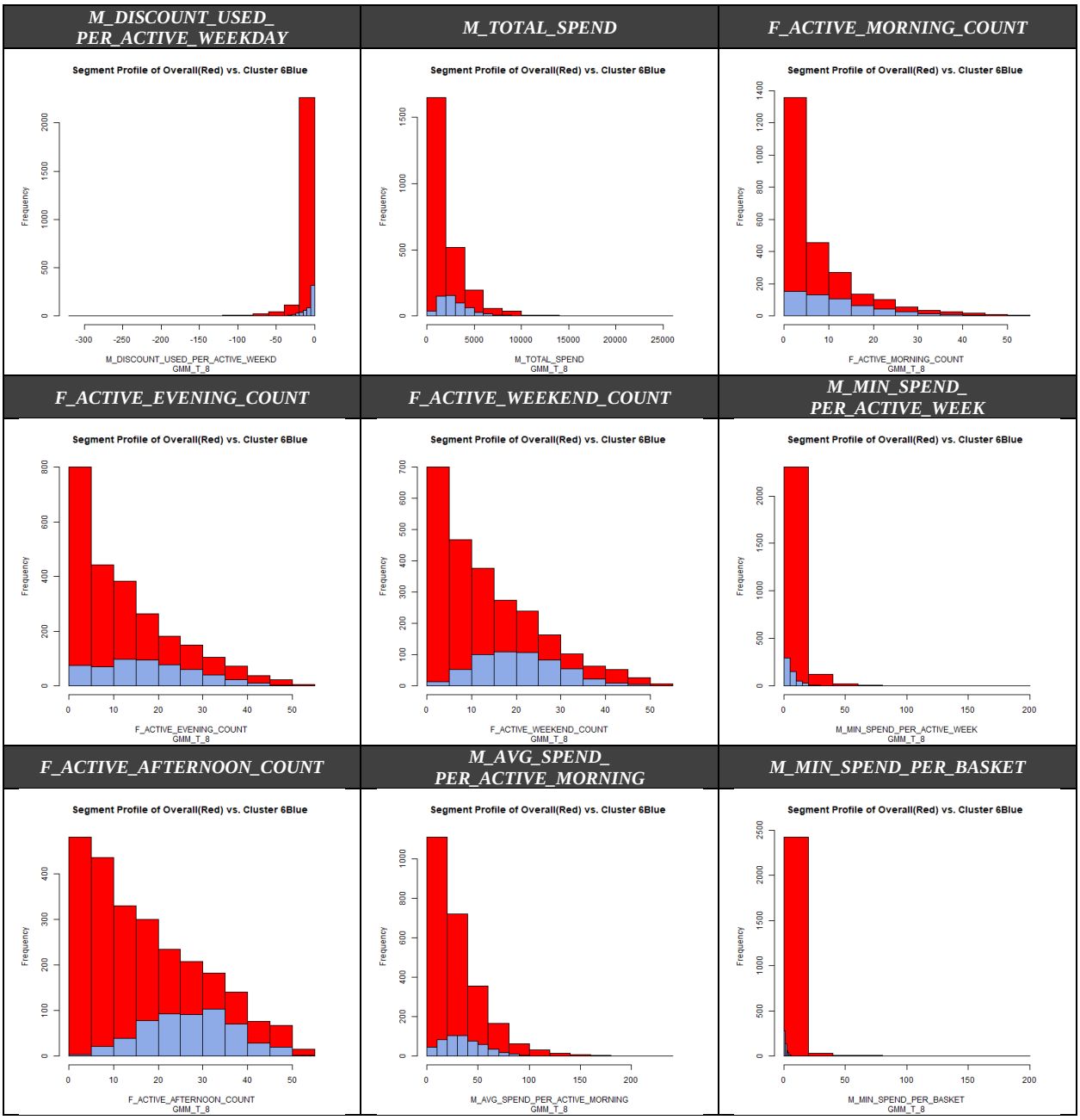


Figure 43: 9 Significant Attributes for Cluster 6 of GMM-T

Cluster 7 shoppers are very similar to Cluster 6, but less frequency in shopping and spends less as a result.

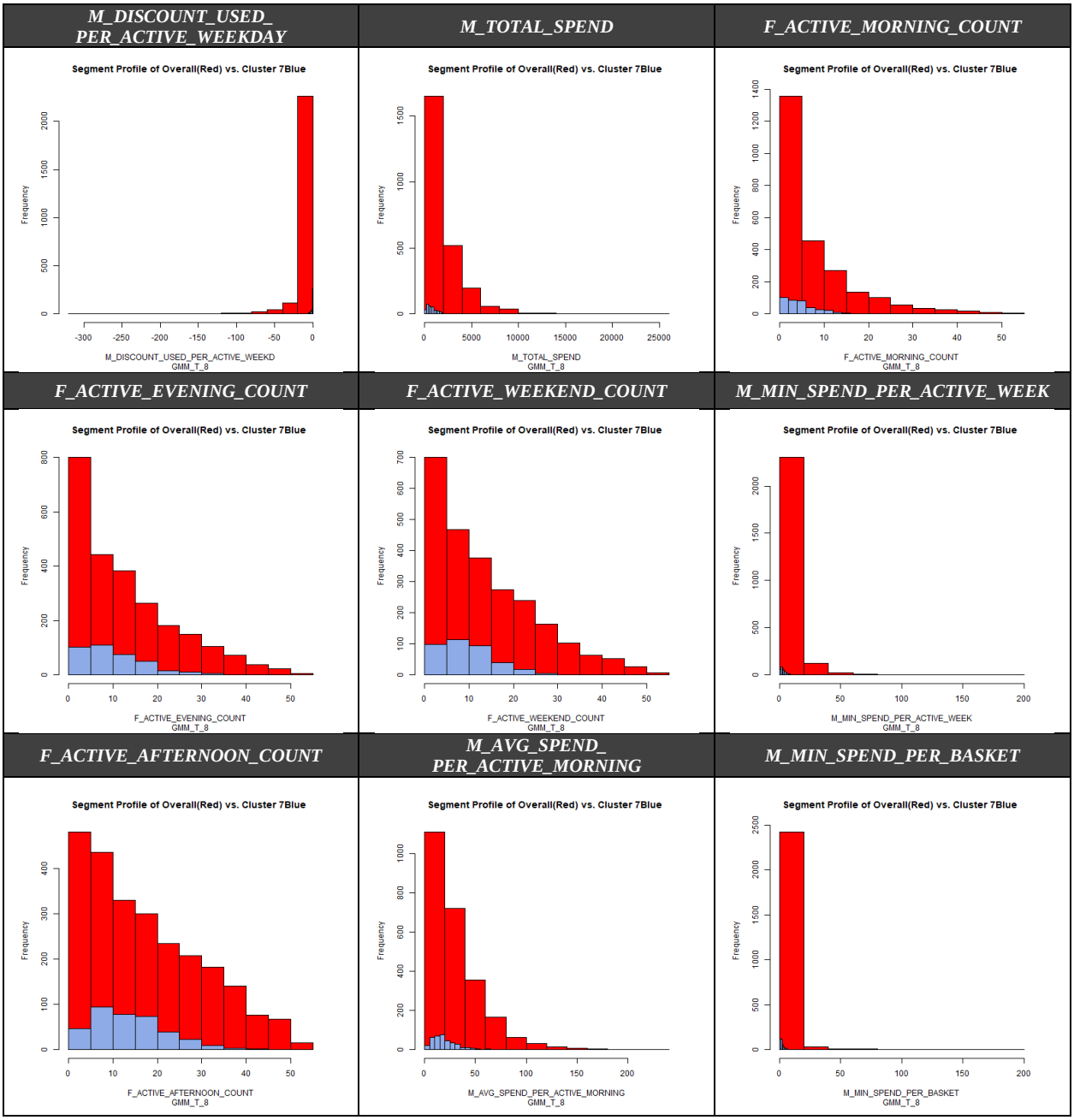


Figure 44: 9 Significant Attributes for Cluster 7 of GMM-T

Cluster 8 shoppers are a small group of shoppers with a good spread in terms of preferred visit frequency. They have very little spending. When they do spend, they use discounts.

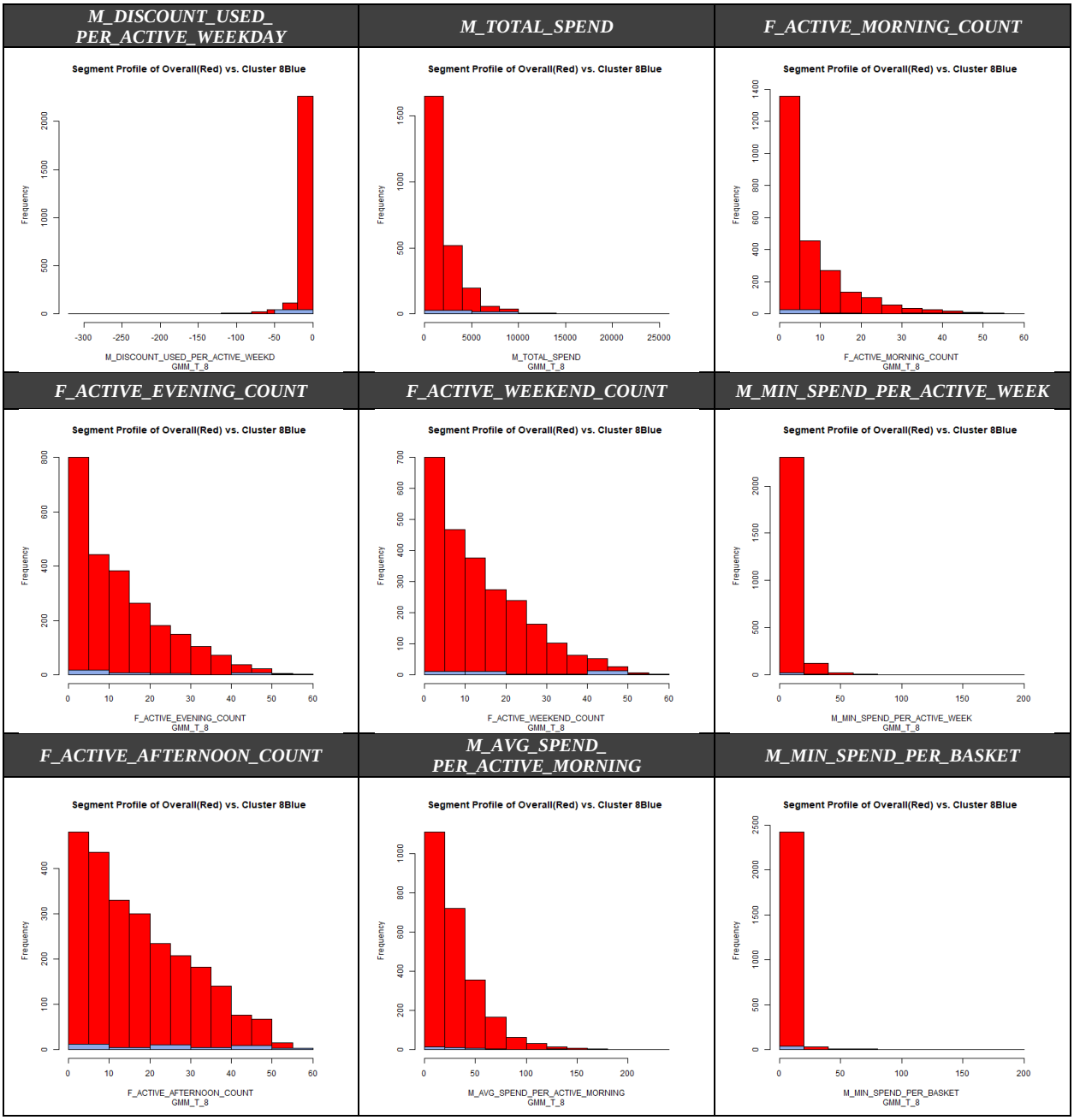


Figure 45: 9 Significant Attributes for Cluster 8 of GMM-T

5.6 Revisiting GMM-R

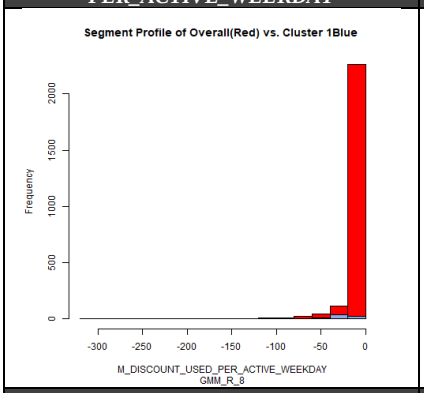
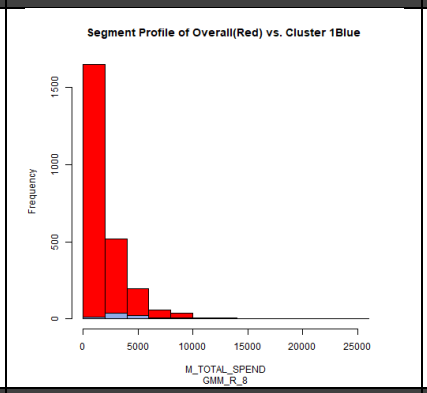
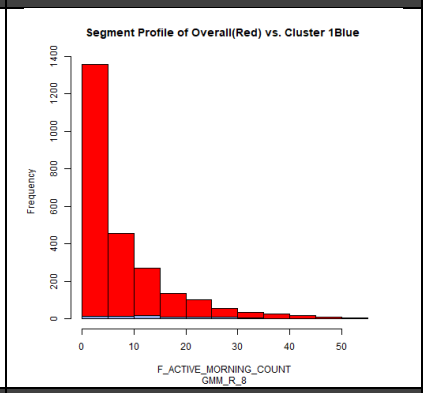
Revisiting the GMM outputs, the optimal cluster count for GMM-T and GMM-R are 8 and 19 respectively. As we examine the spread of IDs within each GMM-R's 19 clusters, some clusters noticeably have only a handful of IDs, namely cluster 18, with 3 IDs and clusters 9 and 11 with 6 IDs each. The converse is also true. Cluster 5 has a much bigger proportion of IDs than the next nearest cluster which is cluster 1. Hence 19 clusters for GMM-R, though optimal, may not yield the best spread of IDs.

In this sense, for an apple-to-apple comparison to GMM-T, we will rerun GMM-R to generate 8 clusters. As can be seen on Table 5, its distribution still leaves plenty to be desired.

Cluster No.	GMM-T	GMM-R	GMM-R-8
1	283	389	54
2	504	11	1,808
3	298	73	103
4	146	40	16
5	274	1,759	7
6	552	51	28
7	368	10	429
8	45	10	25
9	-	6	-
10	-	22	-
11	-	6	-
12	-	9	-
13	-	16	-
14	-	22	-
15	-	8	-
16	-	15	-
17	-	7	-
18	-	3	-
19	-	13	-

Table 5. Distribution of 2,470 IDs across GMM clustering outputs

Cluster 1 shoppers are a good mix of frequent and infrequent shoppers. They are price conscious and have a propensity to use plenty of discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.

M_DISCOUNT_USED_ PER_ACTIVE_WEEKDAY	*M_TOTAL_SPEND*	*F_ACTIVE_MORNING_COUNT*
		
F_ACTIVE_EVENING_COUNT	*F_ACTIVE_WEEKEND_COUNT*	*M_MIN_SPEND_PER_ACTIVE_WEEK*

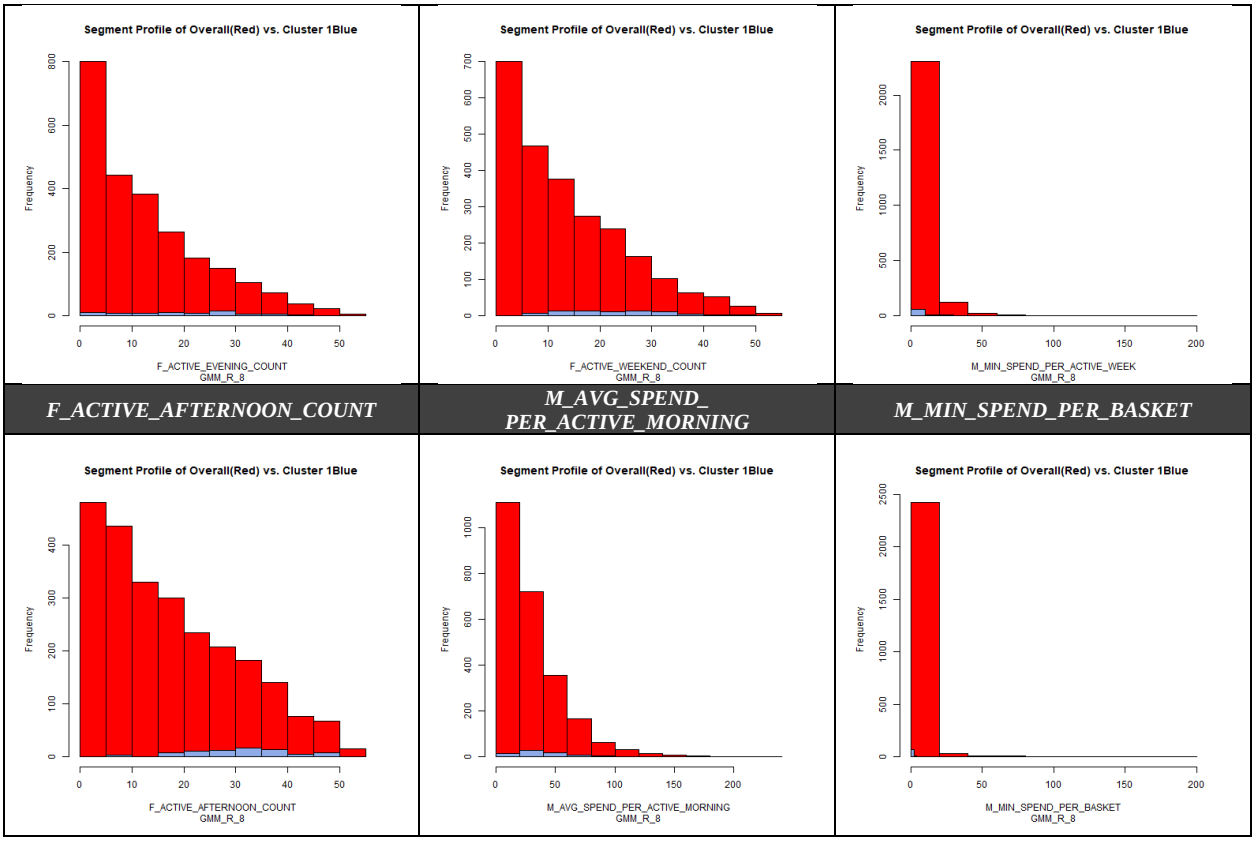


Figure 46: 9 Significant Attributes for Cluster 1 of GMM-R

Cluster 2 shoppers are very active throughout the week. They use very few discounts and spend quite a sizable amount.

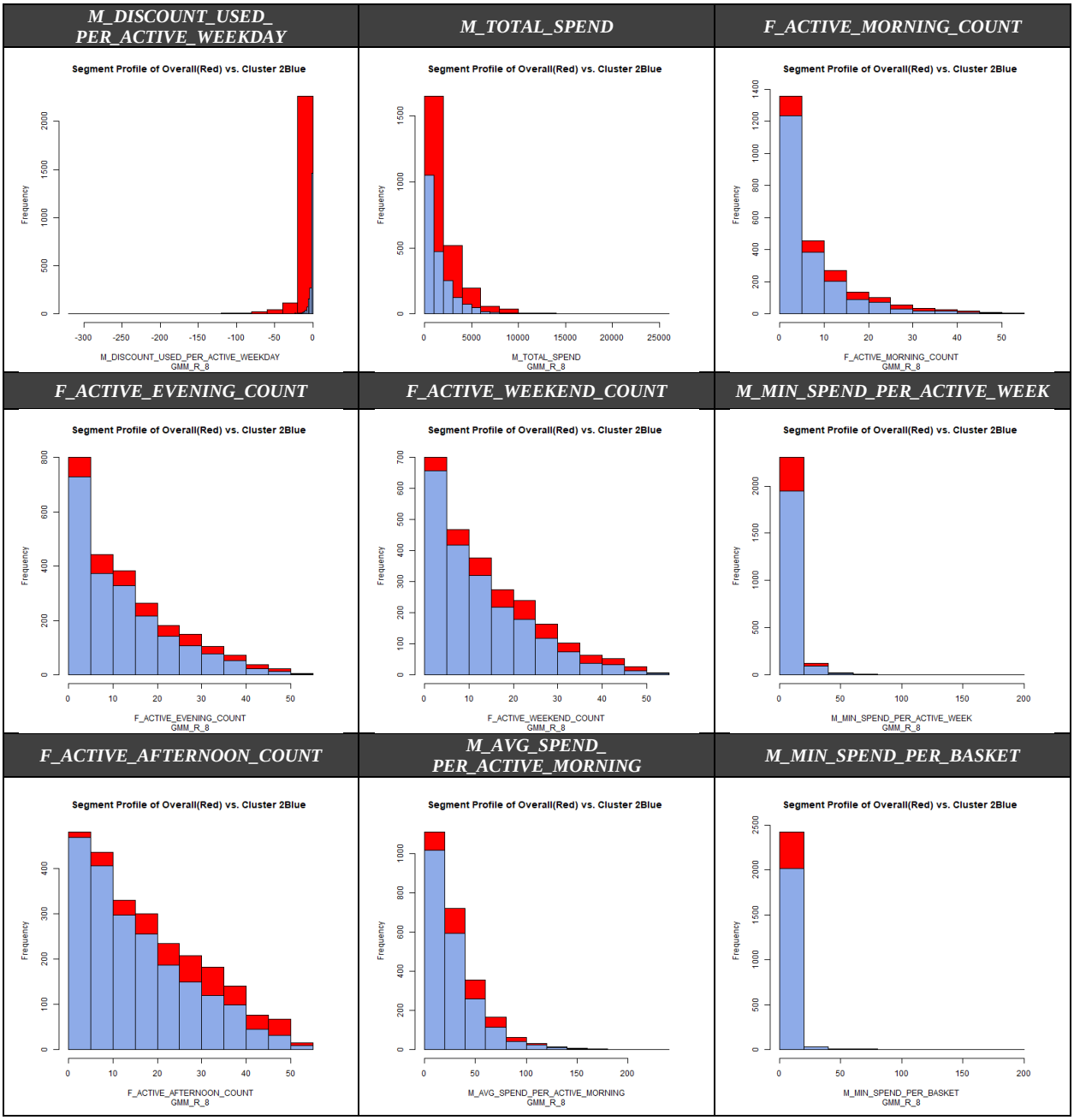


Figure 47: 9 Significant Attributes for Cluster 2 of GMM-R

Cluster 3 is a much bigger group than Cluster 1. Nonetheless, they shop thru the week and spend not as much overall. No other noteworthy differences seen.

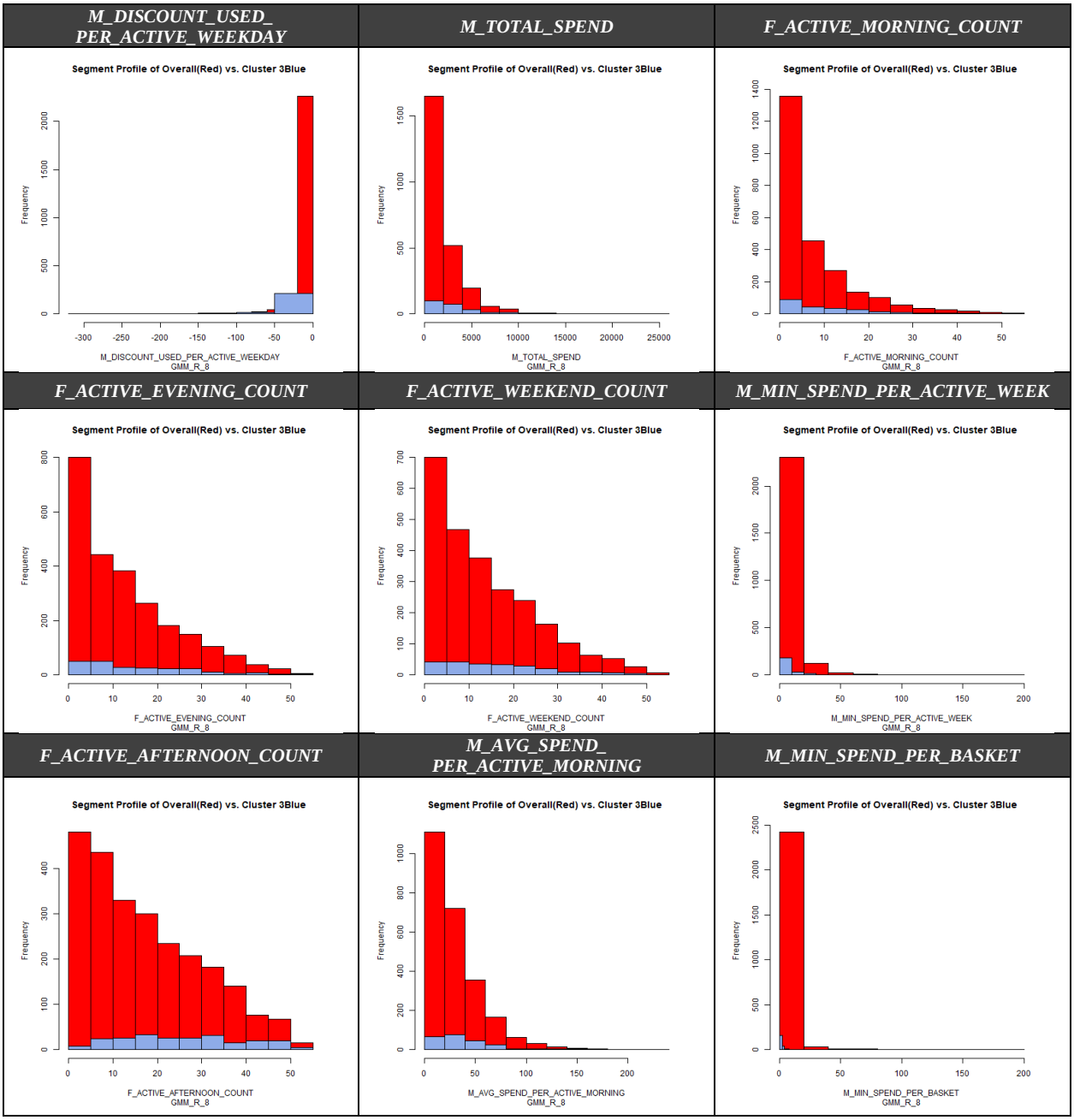


Figure 48: 9 Significant Attributes for Cluster 3 of GMM-R

Cluster 4 shoppers are a much smaller group than Cluster 1. Very little traits can be drawn from the figure below.

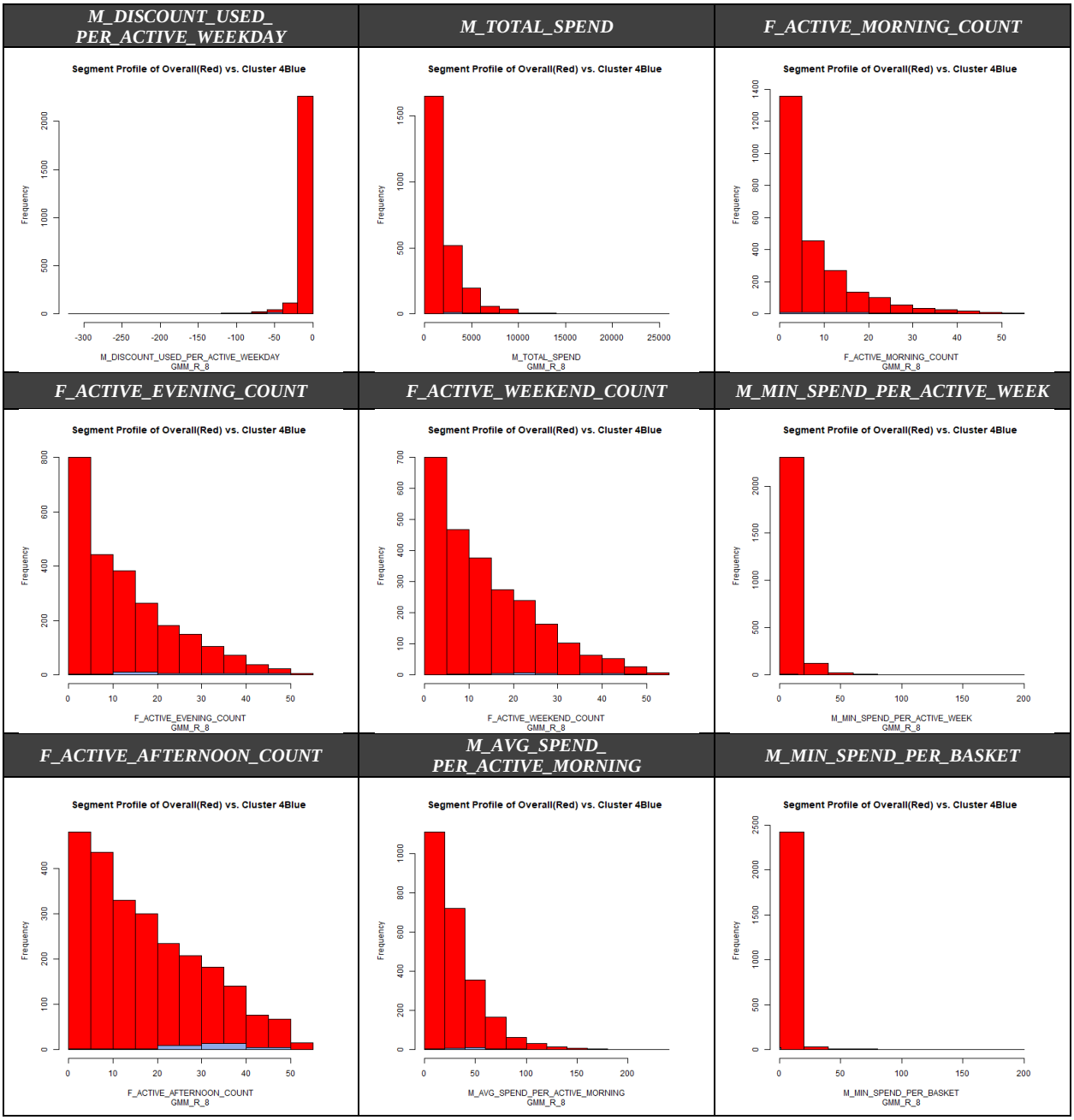


Figure 49: 9 Significant Attributes for Cluster 4 of GMM-R

Though highly similar to Cluster 4, Cluster 5 shoppers shop slightly more frequently.

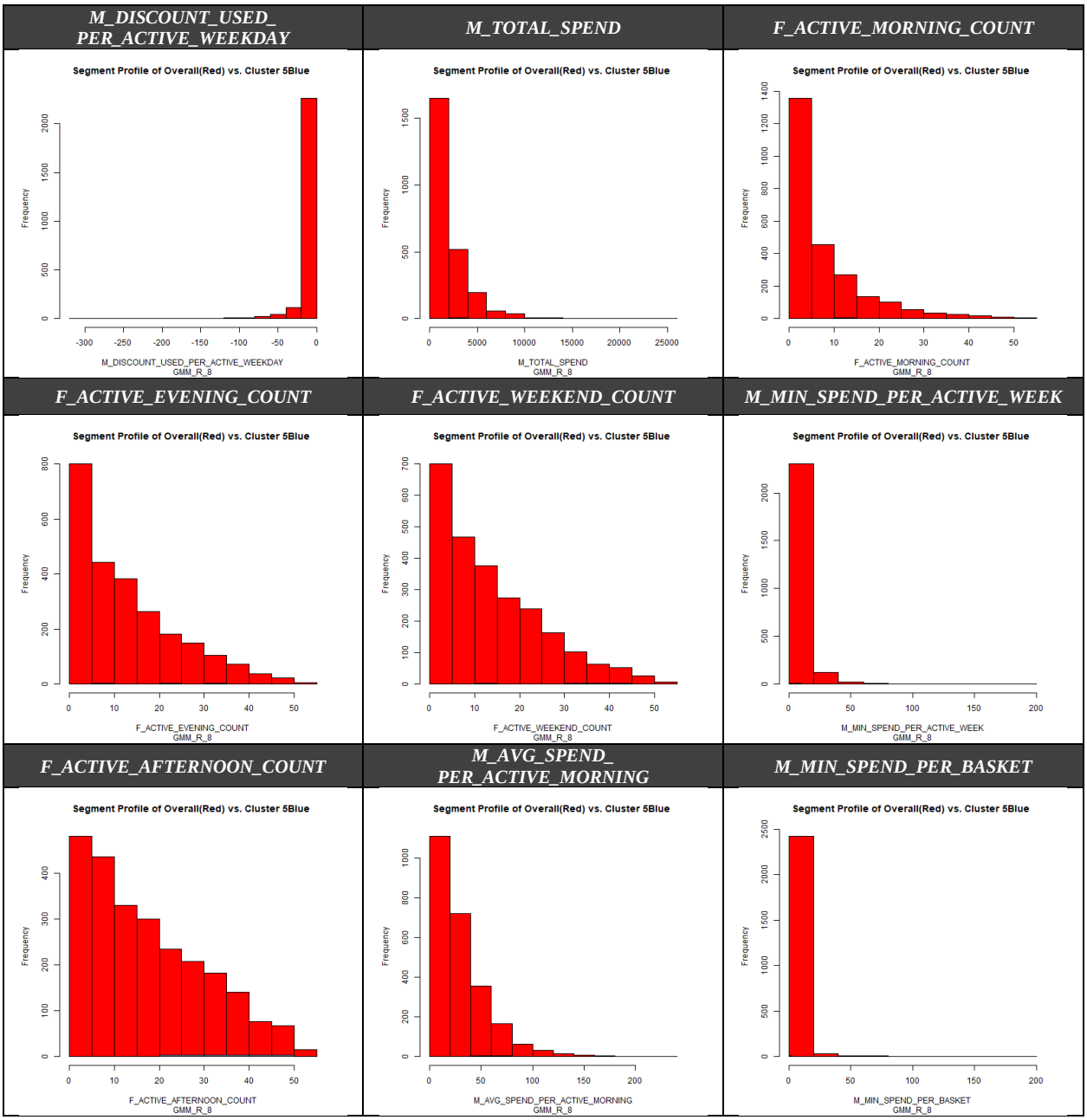


Figure 50: 9 Significant Attributes for Cluster 5 of GMM-R

Though highly similar to Cluster 4, Cluster 6 shoppers shop even less.

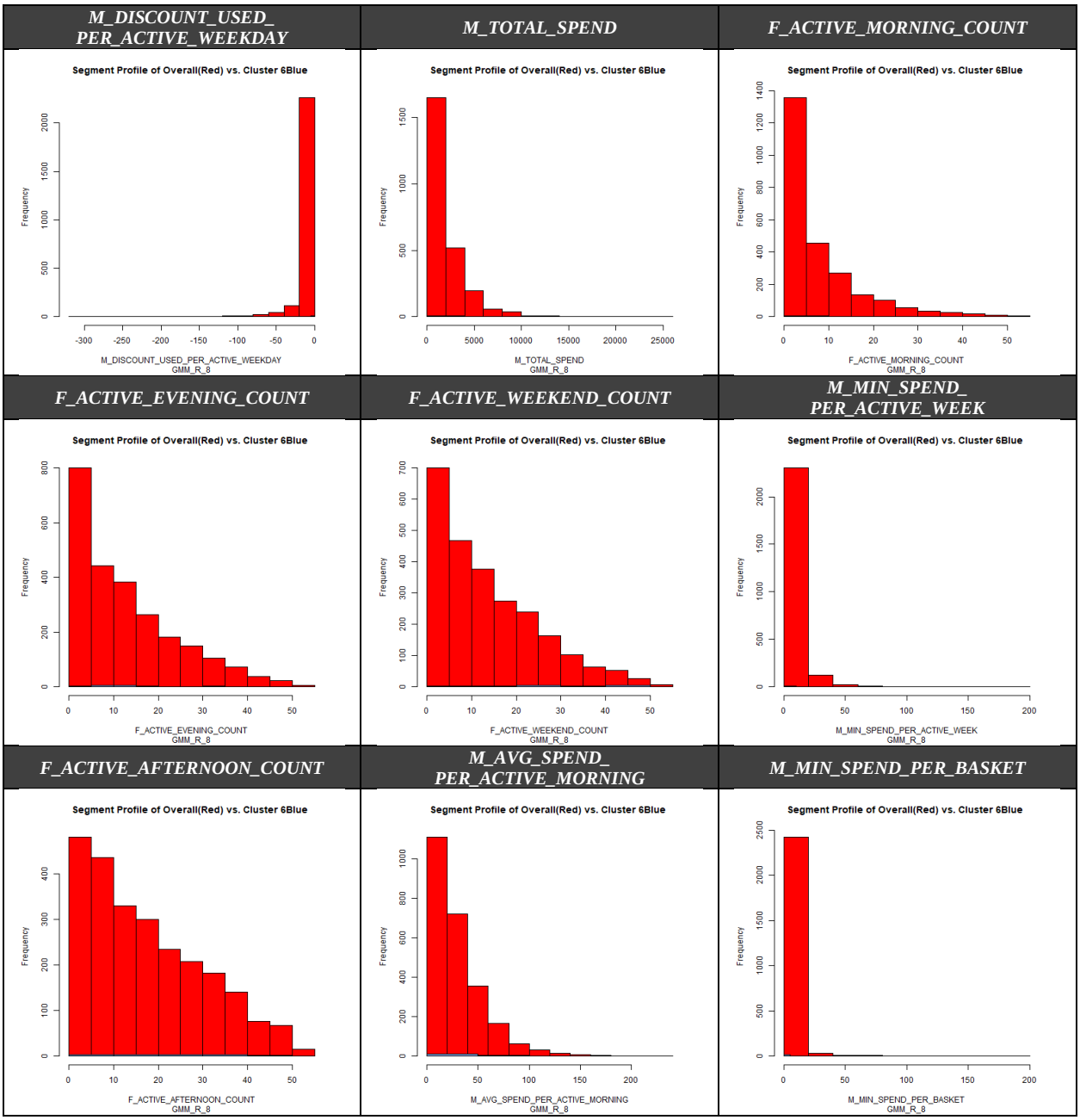


Figure 51: 9 Significant Attributes for Cluster 6 of GMM-R

Both Clusters 6 and 7 are highly similar.

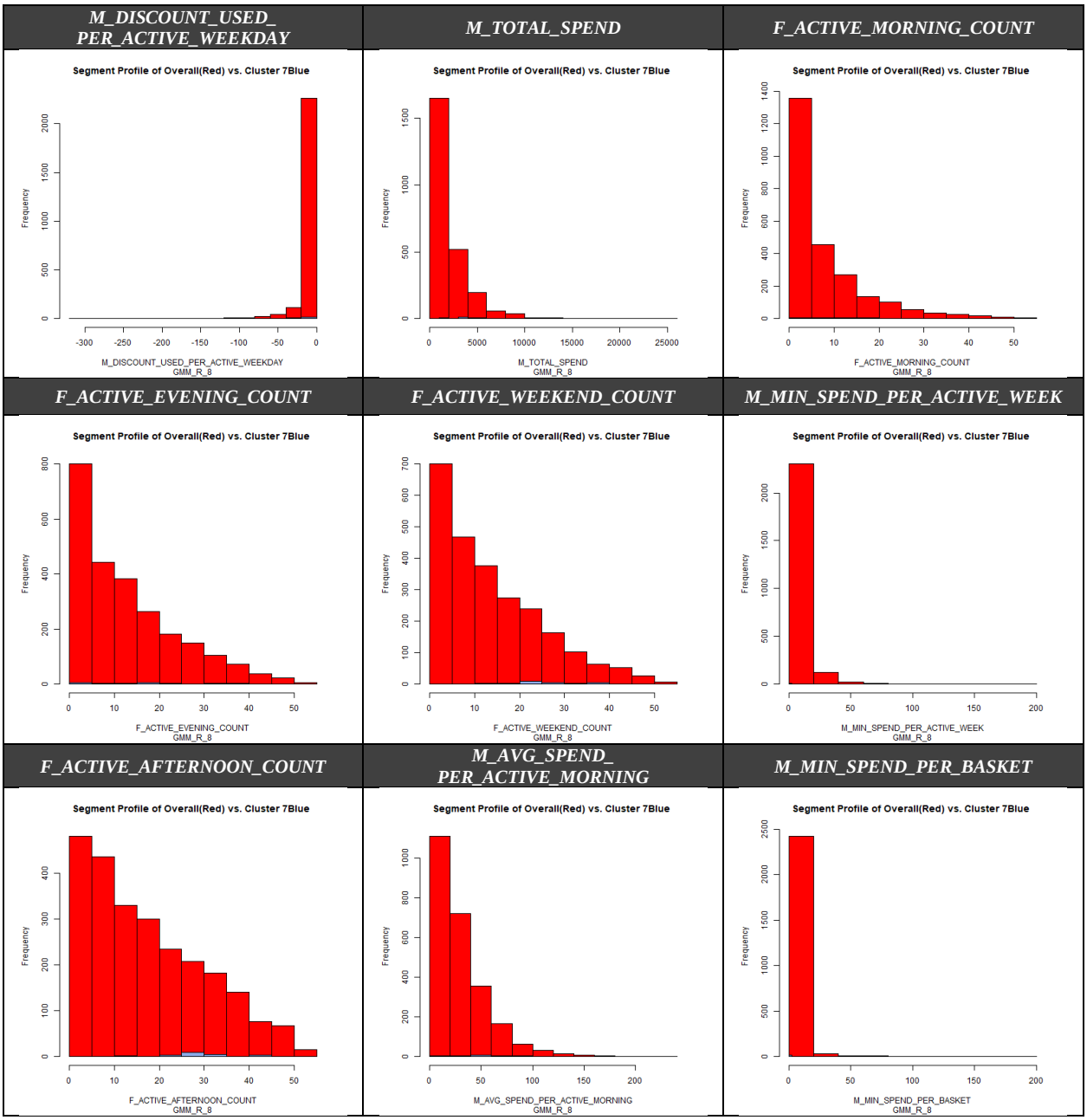


Figure 52: 9 Significant Attributes for Cluster 7 of GMM-R

Both Clusters 7 and 8 are highly similar, as well, in terms of its low activity and spending.

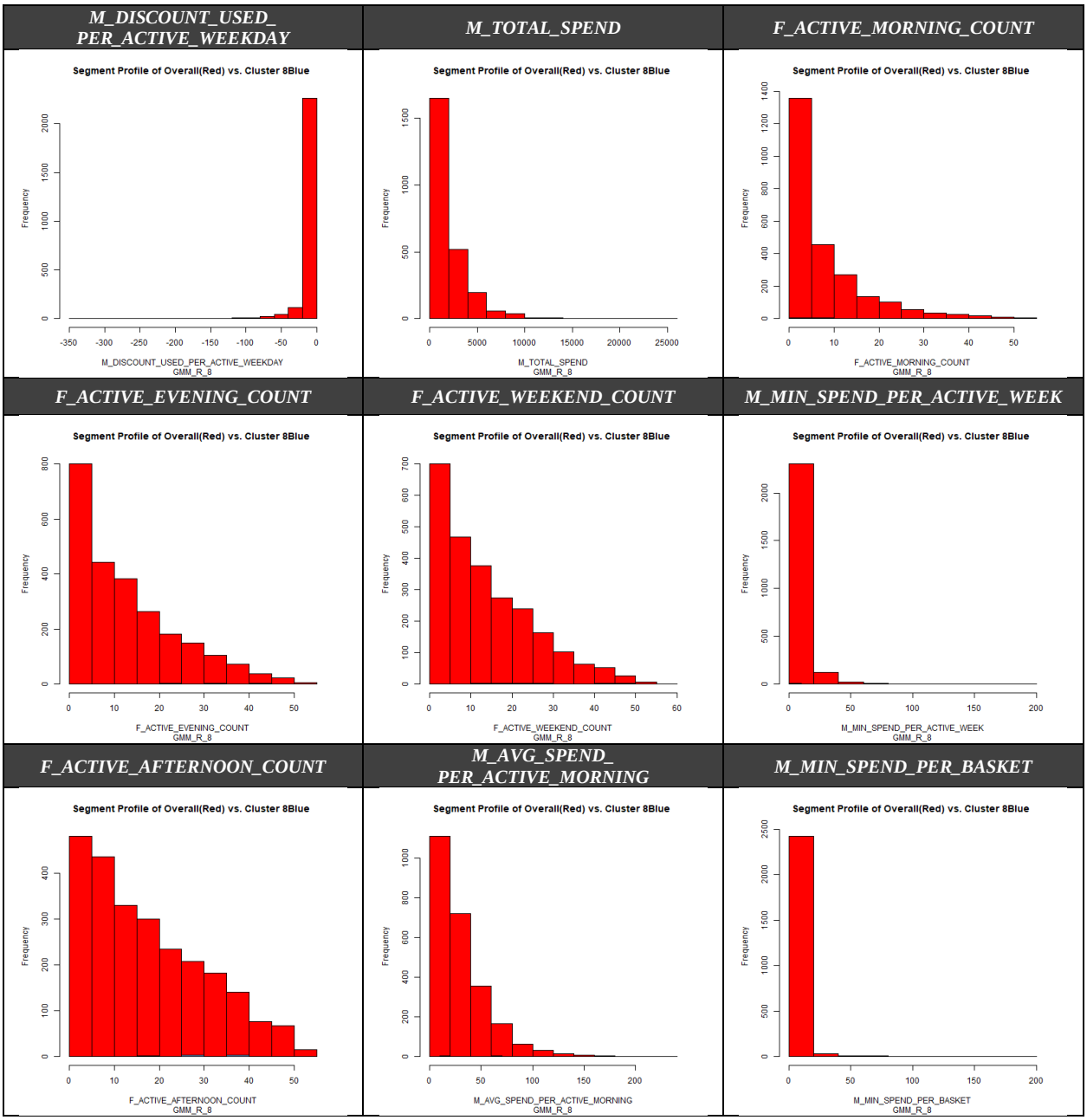


Figure 53: 9 Significant Attributes for Cluster 8 of GMM-R

6.0 Extended Analysis and Results

Before moving on to the next step of attaching household information to these clusters and exploring their relationship, the reduct-based outputs required further exploration. This extended analysis section covers this specific exploration in detail.

6.1 Revisiting Reduct-Based Outputs

There was an interesting trend relating to reduct based outputs. These outputs were clustered, based on just two reduct variables 'M_DISCOUNT_USED_PER_ACTIVE_WEEKDAY' and 'M_DISCOUNT_USED_PER_ACTIVE_WEEKEND.' Table 6 shows the uneven distribution between the various clusters. With the exception of GMM-R, the clusters attributes of TKM-N-R, TKM-A-R and FKM-R nonetheless exhibited highly distinct characteristics. Reflecting on this, it appears that the two reduct variables had a great impact on the clusters and their distinct characteristics. Thus, this capstone aims to explore the impact adding a third reduct variable.

Cluster No.	TKM-N-R	TKM-A-R	FKM-R	GMM-R
1	143	74	100	389
2	2,327	2,396	2,370	11
3	-	-	-	73
4	-	-	-	40
5	-	-	-	1,759
6	-	-	-	51
7	-	-	-	10
8	-	-	-	10
9	-	-	-	6
10	-	-	-	22
11	-	-	-	6
12	-	-	-	9
13	-	-	-	16
14	-	-	-	22
15	-	-	-	8
16	-	-	-	15
17	-	-	-	7
18	-	-	-	3
19	-	-	-	13

Table 6. Distribution of 2,470 IDs across the various clustering outputs

To get the third reduct variable, the earlier reduct steps are repeated. This capstone removed both these variables from the DATA_ALL Node, using the R Code within the OSI Node.

Figure 54: 16 Attaching Open-Source Integration Node (Removed Two Inputs and Redid Reduct) to Data_All Node

Figure 55 shows the third run of the Rough Set reduct output. Out of 26 possible input variables to choose from, the R code within the OSI Node chose only one input variable as before. This time,

the input variable ‘M_AVG_SPEND_PER_ACTIVE_LATE_NIGHT’ was chosen. Thus, the reduct dataset (originally from the base dataset) would now comprise of one ID variable (household_ID), and three input variables (M_DISCOUNT_USED_PER_ACTIVE_WEEKDAY, M_DISCOUNT_USED_PER_ACTIVE_WEEKEND and M_AVG_SPEND_PER_ACTIVE_LATE_NIGHT).

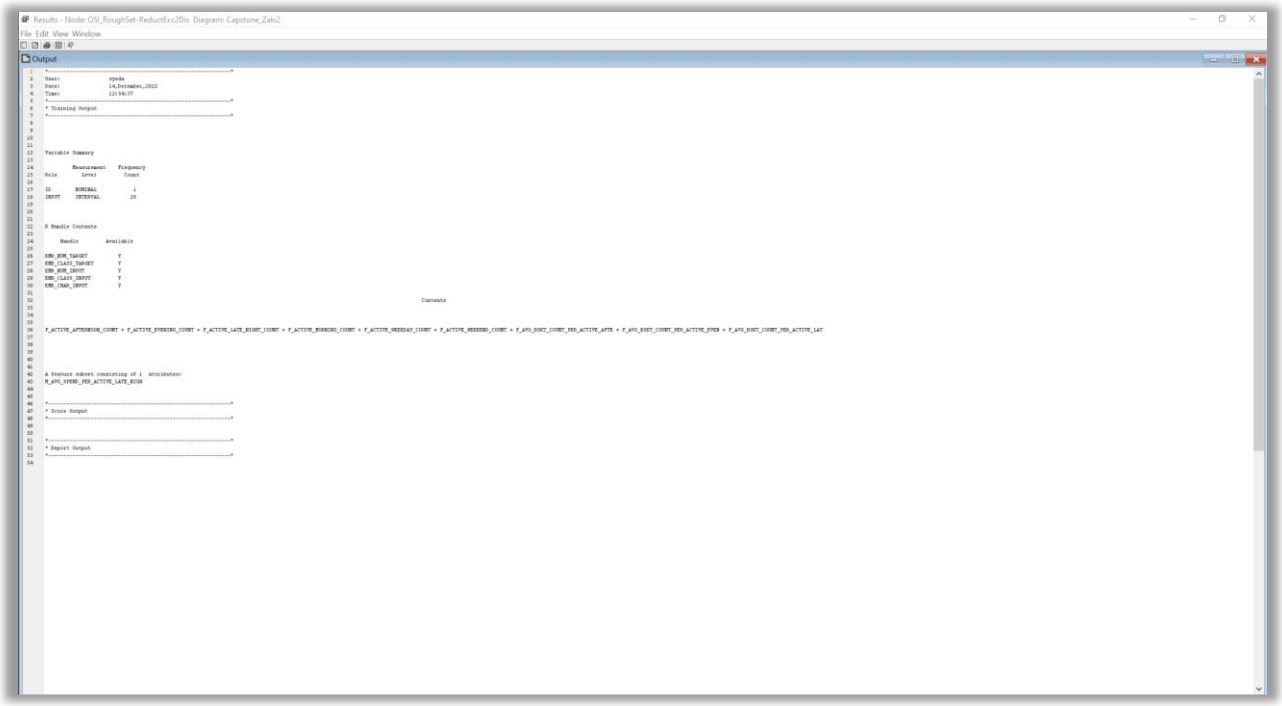


Figure 55: Rough Set Reduct Output from DATA_ALL
(Excluding Two Previous Reduct Variables)

This Rough Set reduct output of 3 input and 1 ID variable are then reran to the same four outputs, as per Figure 56.

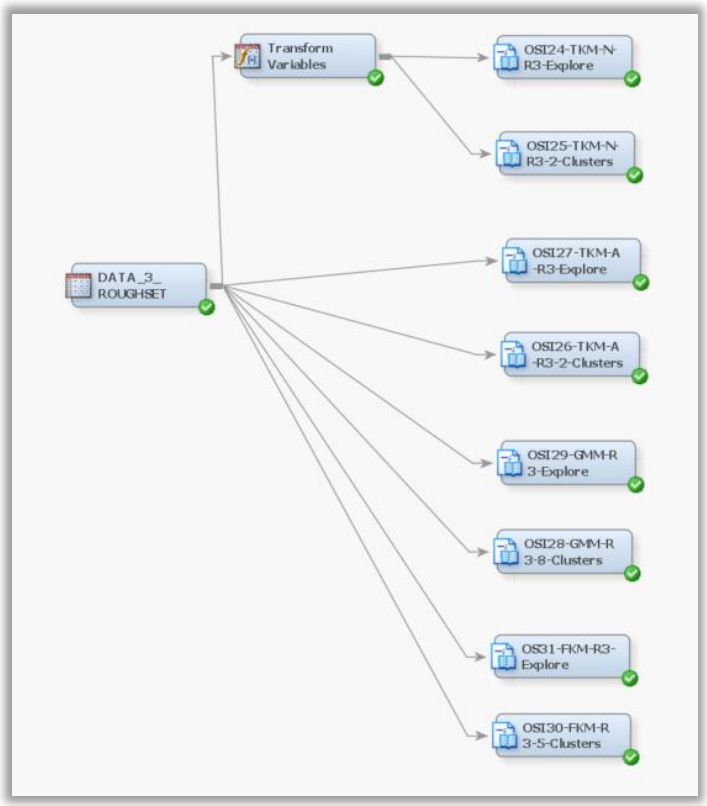
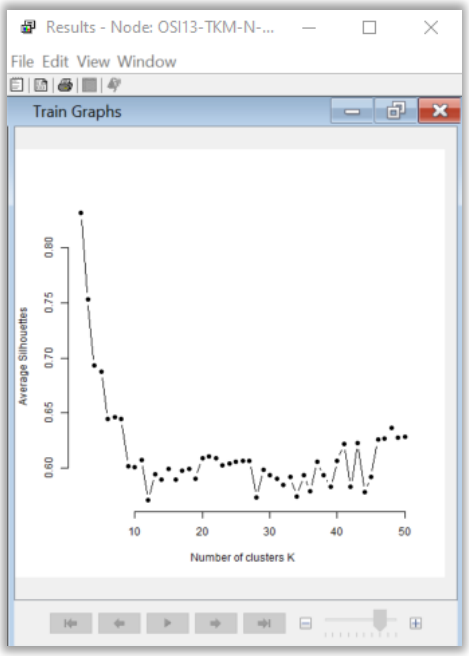


Figure 56: Four Outputs (TKM-N-R, TKM-A-R, GMM-R, FKM-R) Reran with Reduct Dataset of 3 Input Variables

<u>TKM-N-R (Traditional *k*-means clustering on normalized form of 3 reduct inputs)</u>
Three reduct input variables are first transformed within the Transform Variables Node. These are then ingested into the OSI Node called 'OSI24-TKM-N-R3-Explore.' Setting the seed value to 1234, and then calling the '*cluster*' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 50. This silhouette index is then plotted within EM, as seen below.

Both graphs are highly similar. The only difference is the 2 reduct dataset starts out higher at the ~0.85 before plateauing around the 0.55 mark. On the other hand, the 3 reduct dataset starts out lower at the ~0.75 market, before plateauing around the 0.50 mark. Again, we will regard $k = 2$ as its most optimized cluster count.



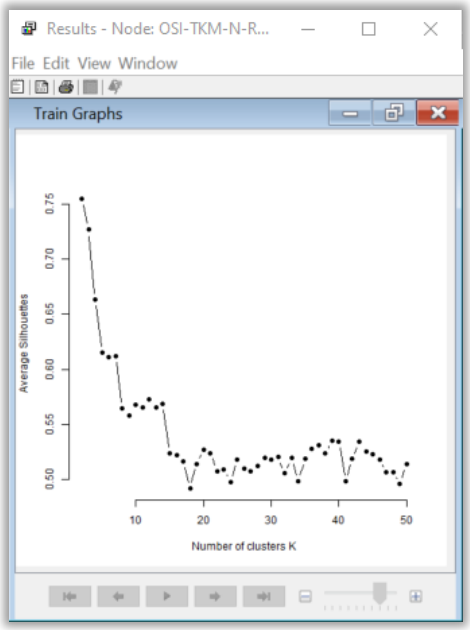
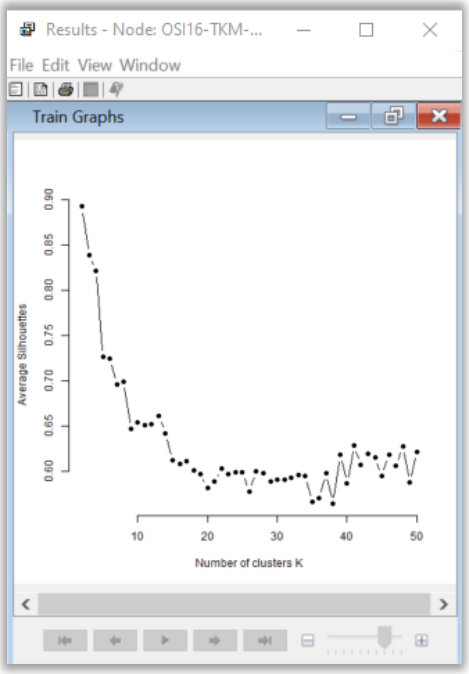


Figure 57: Comparison of Silhouette Index of TKM-N-R output from 2 reduct dataset (left) and 3 reduct dataset (right)

<u>TKM-A-R (Traditional *k*-means clustering on as-is form of 3 reduct inputs)</u>

Without the normalization step, the three reduct input variables are ingested into the OSI Node called 'OSI27-TKM-A-R3-Explore.' Setting the seed value to 1234, and then calling the '*cluster*' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 50. This silhouette index is then plotted within EM, as seen below.

Both graphs are highly similar yet again, and the two reduct dataset starts off higher than the three reduct dataset. Again, we will regard $k = 2$ as its most optimized cluster count.



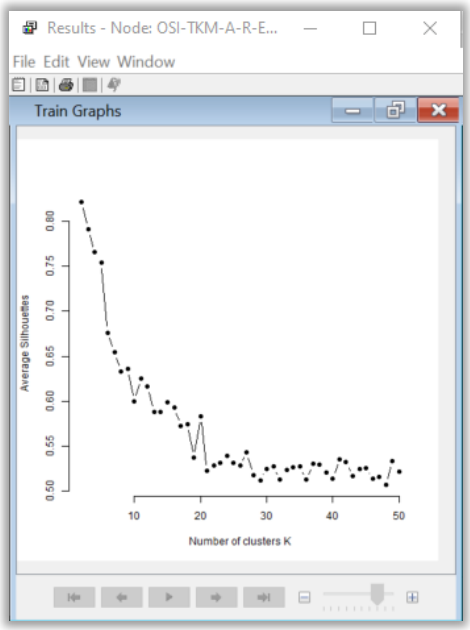


Figure 58: Comparison of Silhouette Index of TKM-A-R output from 2 reduct dataset (left) and 3 reduct dataset (right)

<u>FKM-R (Fuzzy *k*-means clustering on the as-is form of 3 reduct inputs)</u>
This time, the 3 reduct dataset are directly ingested into the OSI Node called 'OSI31-FKM-R3-Explore.' Calling the '*cluster*' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 20. This is a much lower initial cluster exploration range, compared to the *k*-means count range of 2 to 50. This is due to the extremely long processing time, as the cluster count increases. This fuzzy silhouette index is then plotted within EM, as seen below.

In similar fashion, both have the same trendline and that the two reduct dataset starts off from a higher mark than the three reduct dataset. Again, we will regard $k = 2$ as its most optimized cluster count.

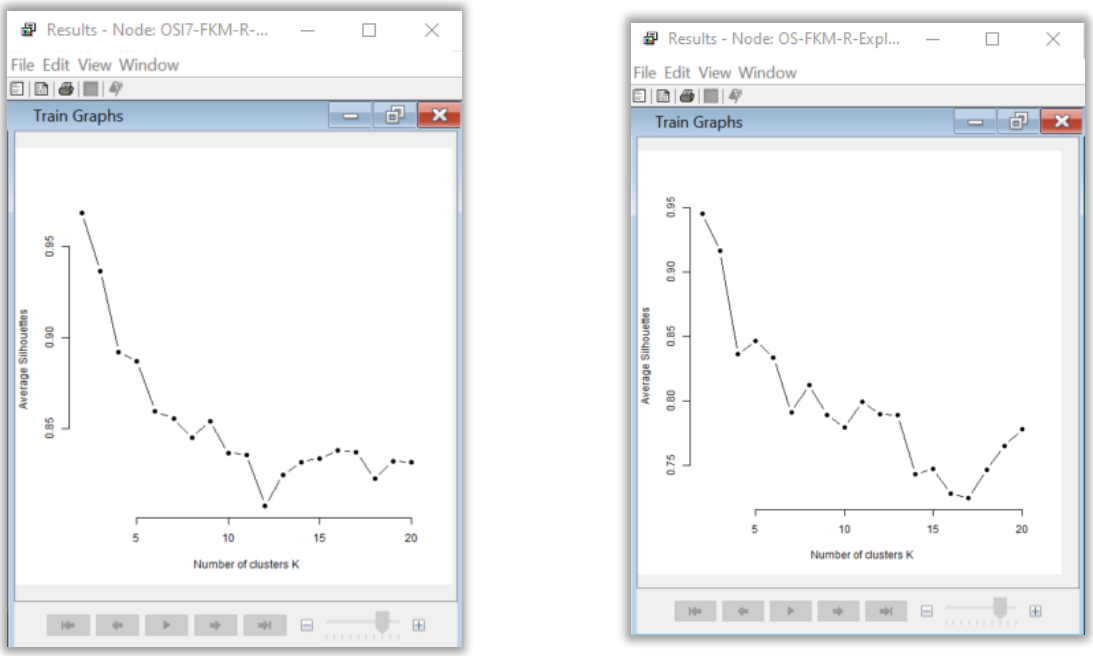


Figure 59: Comparison of Silhouette Index of FKM-R output from 2 reduct dataset (left) and 3 reduct dataset (right)

<u>GMM-R (Gaussian Mixture Model on the as-is form of 3 reduct inputs)</u>
The 3 reduct dataset are again directly ingested into the OSI Node called 'OSI29-GMM-R3-Explore.' Calling the '*cluster*' and '*purrr*' packages, a function to record the silhouette index of a particular cluster *k* is created, and then looped through cluster counts 2 to 20. This is a much lower initial cluster exploration range, compared to the *k*-means count range of 2 to 50. This is due to the extremely long processing time, as the cluster count increases. This fuzzy silhouette index is then plotted within EM, as seen below.

Here both graphs look highly dissimilar. The *mclust* package recommends either $k = 2$, 8 or 20 in that order of highest BIC values. Given that $k = 2$ might provide too little granularity and that GMM-T had an optimal output of $k = 8$, we will regard $k = 8$ as its most optimized cluster count.

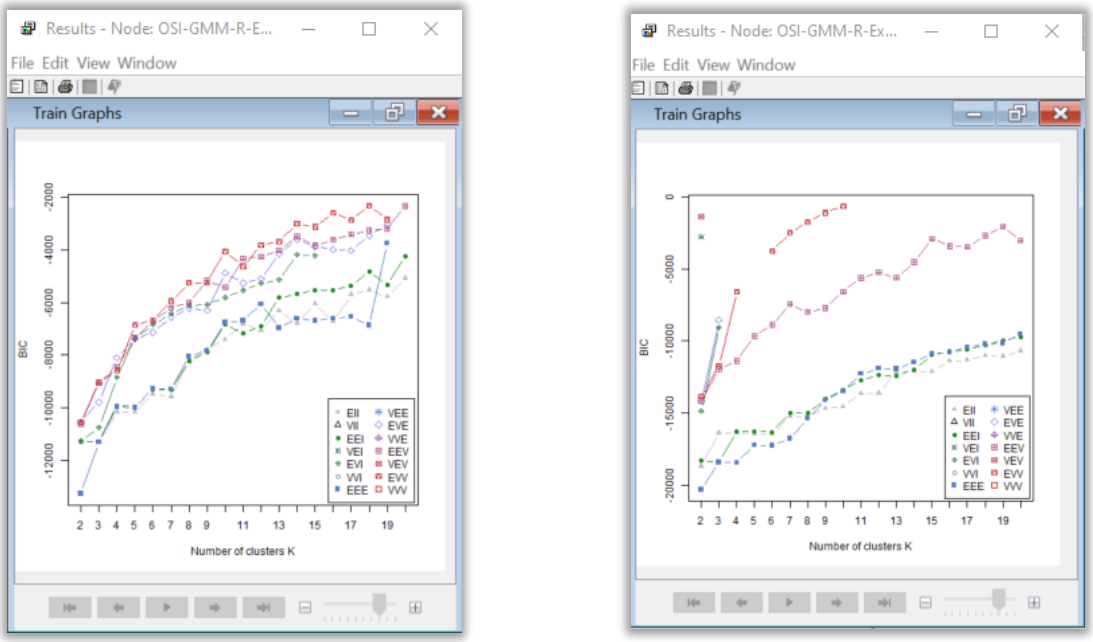


Figure 60: Comparison of Silhouette Index of GMM-R output from 2 reduct dataset (left) and 3 reduct dataset (right)

6.2 Overview of Extended Clustering Output Results

Table 7 shows the various output distribution within each cluster i.e. the number of objects or households that have been allocated to each cluster. These results reveal that both the TKM-N-R and TKM-N-R3 are mirror opposites of each other. Both TKM-A-R outputs also have a similar trend. Even with high cluster counts, both FKM-R3 and GMM-R3 look highly uneven. To have a suitable comparison, the same 9 cluster variables from each of the two reduct output would be used for the three reduct outputs.

Cluster No.	TKM-N-R	TKM-N-R3 (New)	TKM-A-R	TKM-A-R3 (New)	FKM-R	FKM-R3 (New)	GMM-R	GMM-R3 (New)
1	143	2,327	74	2,384	100	57	389	54
2	2,327	143	2,396	86	2,370	233	11	1,808
3	-		-		-	59	73	103
4	-		-		-	1,856	40	16
5	-		-		-	265	1,759	7
6	-		-		-		51	28
7	-		-		-		10	429
8	-		-		-		10	25
9	-		-		-		6	
10	-		-		-		22	
11	-		-		-		6	
12	-		-		-		9	
13	-		-		-		16	
14	-		-		-		22	
15	-		-		-		8	
16	-		-		-		15	
17	-		-		-		7	
18	-		-		-		3	
19	-		-		-		13	

Table 7. Distribution of 2,470 IDs across the various clustering outputs (Newer outputs are highlighted in yellow)

TKM-N-R3 (Traditional *k*-means clustering on normalized form of 3 reduct inputs)

Figure 61 reveals TKM-N-R3's Cluster 1's significant attributes. Cluster 1 shoppers are highly frequent shoppers with deep pockets. They use some discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more per item.

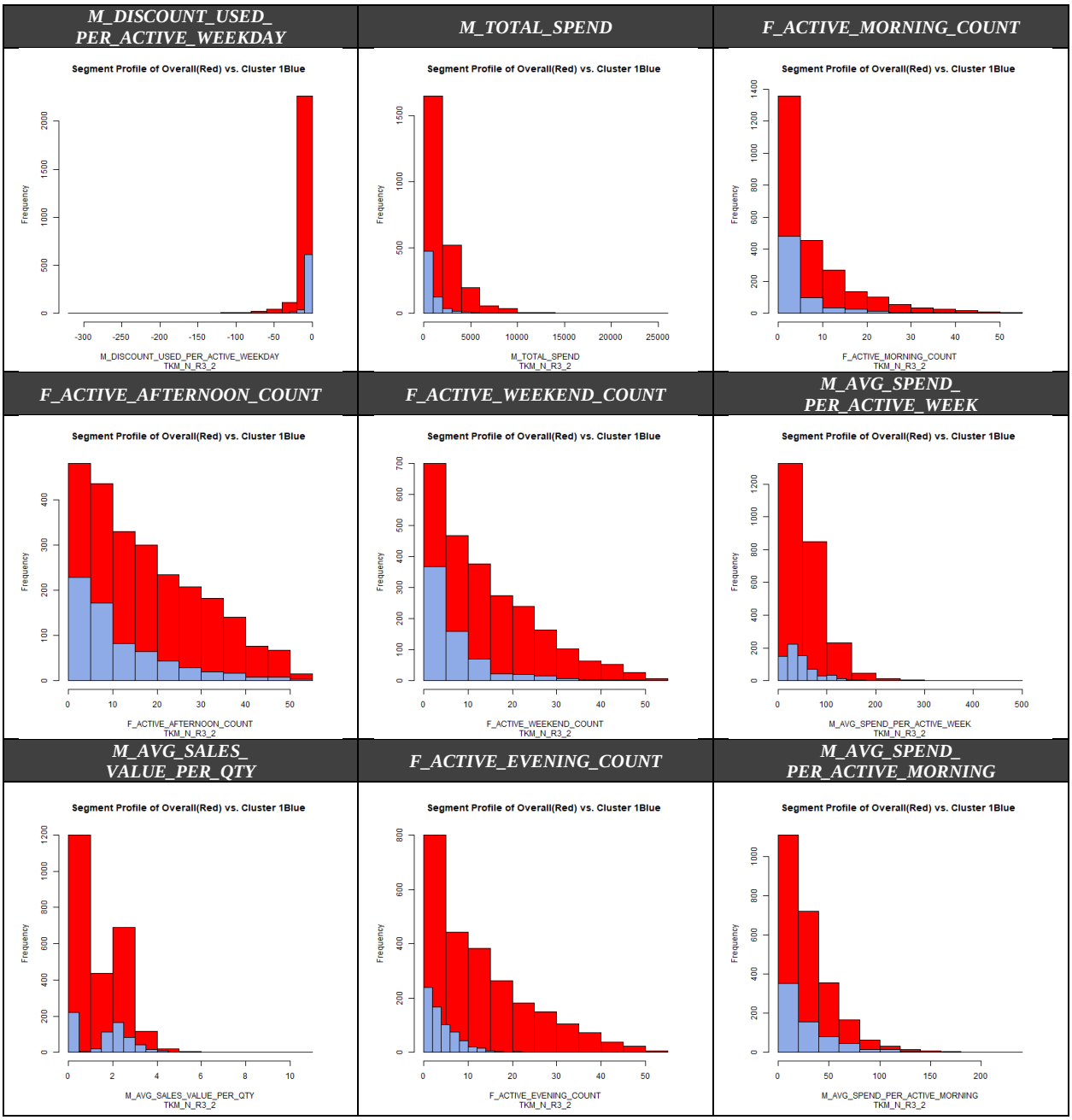


Figure 61: 9 Significant Attributes for Cluster 1 of TKM-N-R3

Figure 62 reveals TKM-N-R3's Cluster 2's significant attributes. Cluster 2 shoppers are highly infrequent shoppers with very little spending.

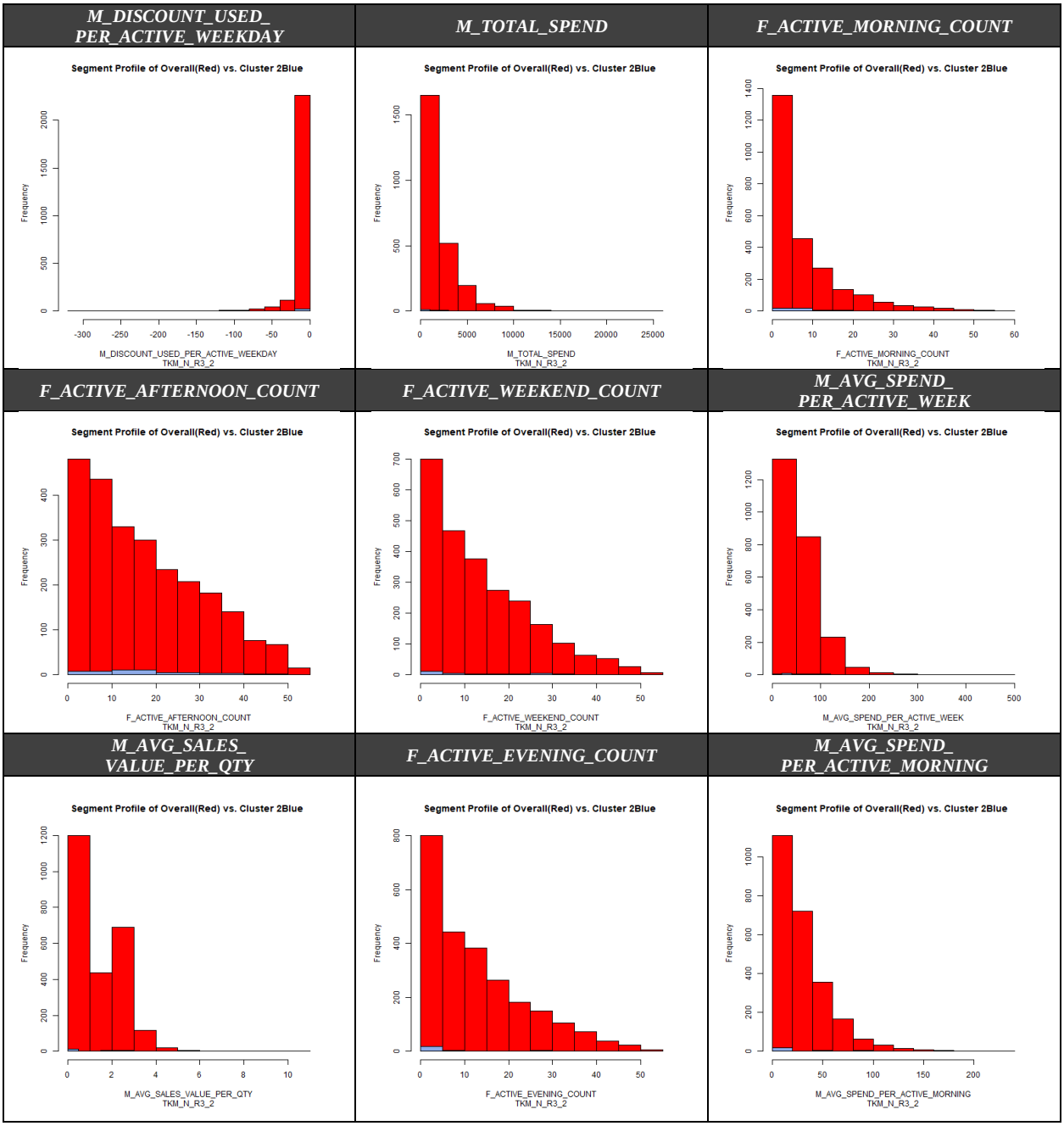


Figure 62: 9 Significant Attributes for Cluster 2 of TKM-N-R3

TKM-A-R3 (Traditional *k*-means clustering on as-is form of 3 reduct inputs)

Figure 63 reveals TKM-A-R3's Cluster 1's significant attributes. Cluster 1 shoppers are more infrequent shoppers with sizable spending. They prefer to shop in the afternoon, and to a lesser extent on weekends and evenings. They are not price sensitive, given the low discount usage and high sales value per quantity. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each item.

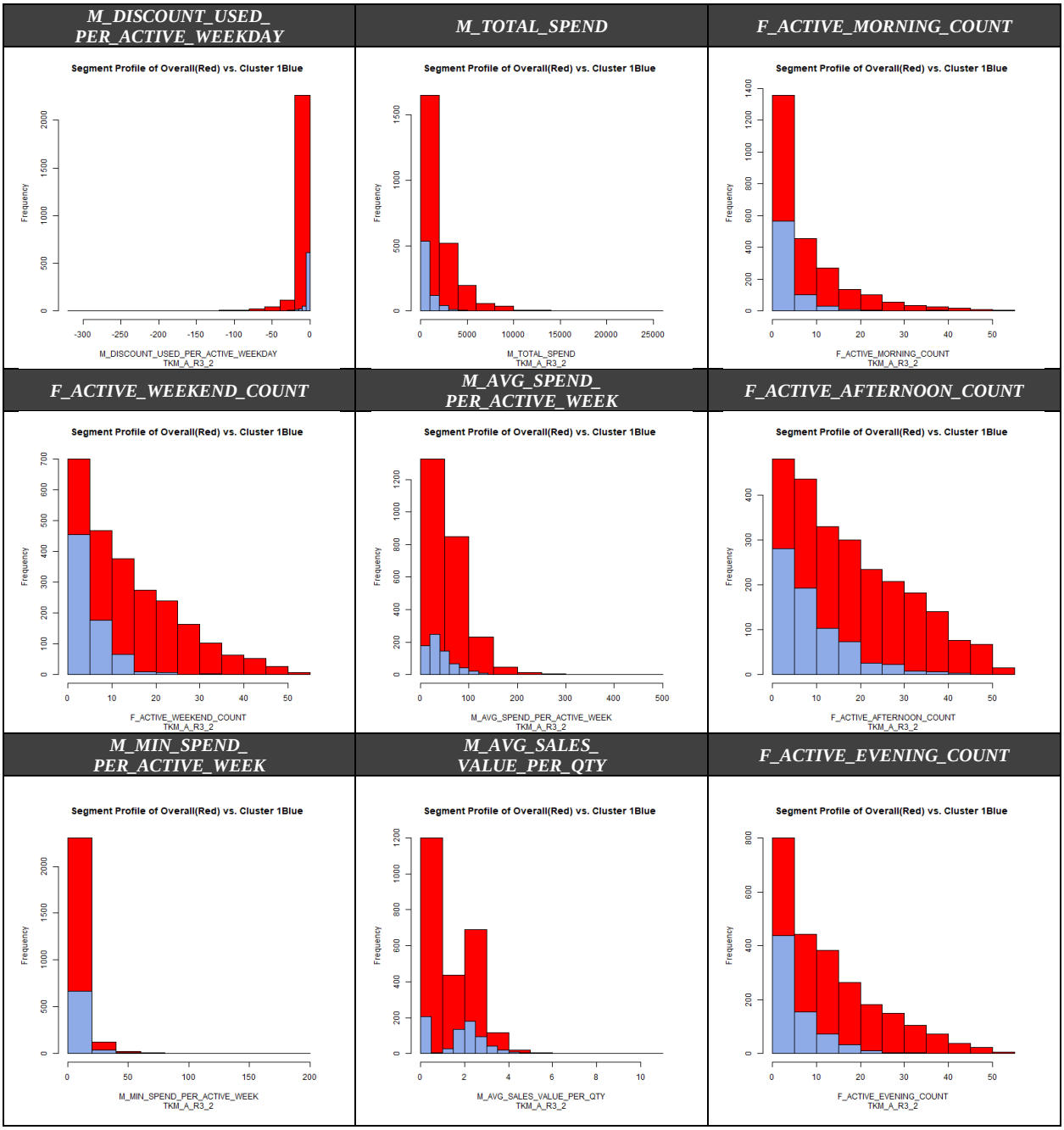


Figure 63: 9 Significant Attributes for Cluster 1 of TKM-A-R3

Figure 64 reveals TKM-A-R3's Cluster 2's significant attributes. Cluster 2 shoppers are highly infrequent shoppers and are price conscious. They have a propensity not to use any discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.

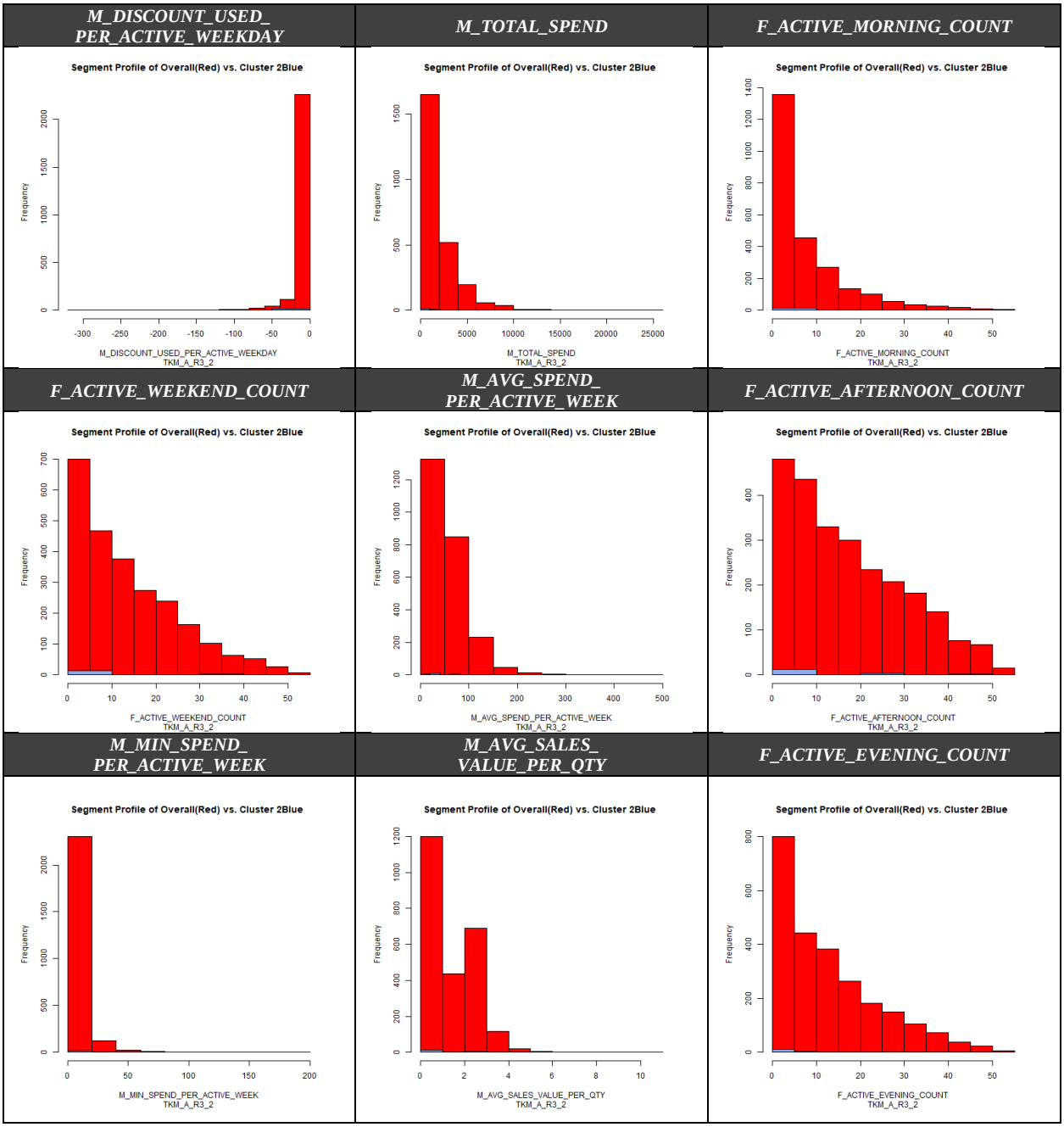


Figure 64: 9 Significant Attributes for Cluster 2 of TKM-A-R3

FKM-R3 (Fuzzy *k*-means clustering on the as-is form of 3 reduct inputs)

Figure 65 reveals FKM-R3's Cluster 1's significant attributes. Cluster 1 shoppers are a small group with highly infrequent shoppers and don't spend much. When they do shopping, it's usually in the afternoons.

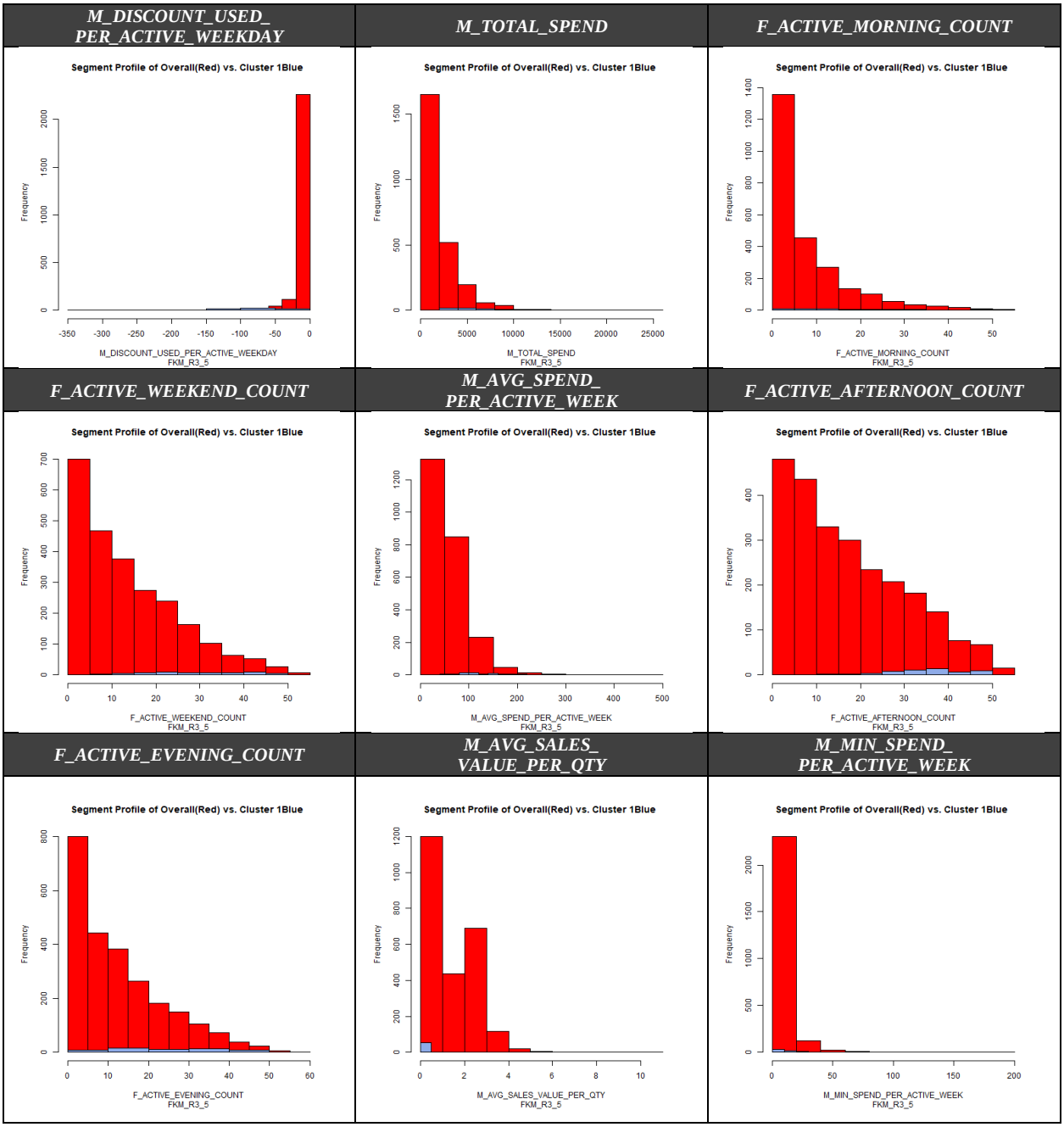


Figure 65: 9 Significant Attributes for Cluster 1 of FKM-R3

Figure 66 reveals FKM-R3's Cluster 2's significant attributes. Cluster 2 shoppers are a good mix of frequent and infrequent shoppers. They use little to no discounts on a weekday, and they tend to spend less per quantity.

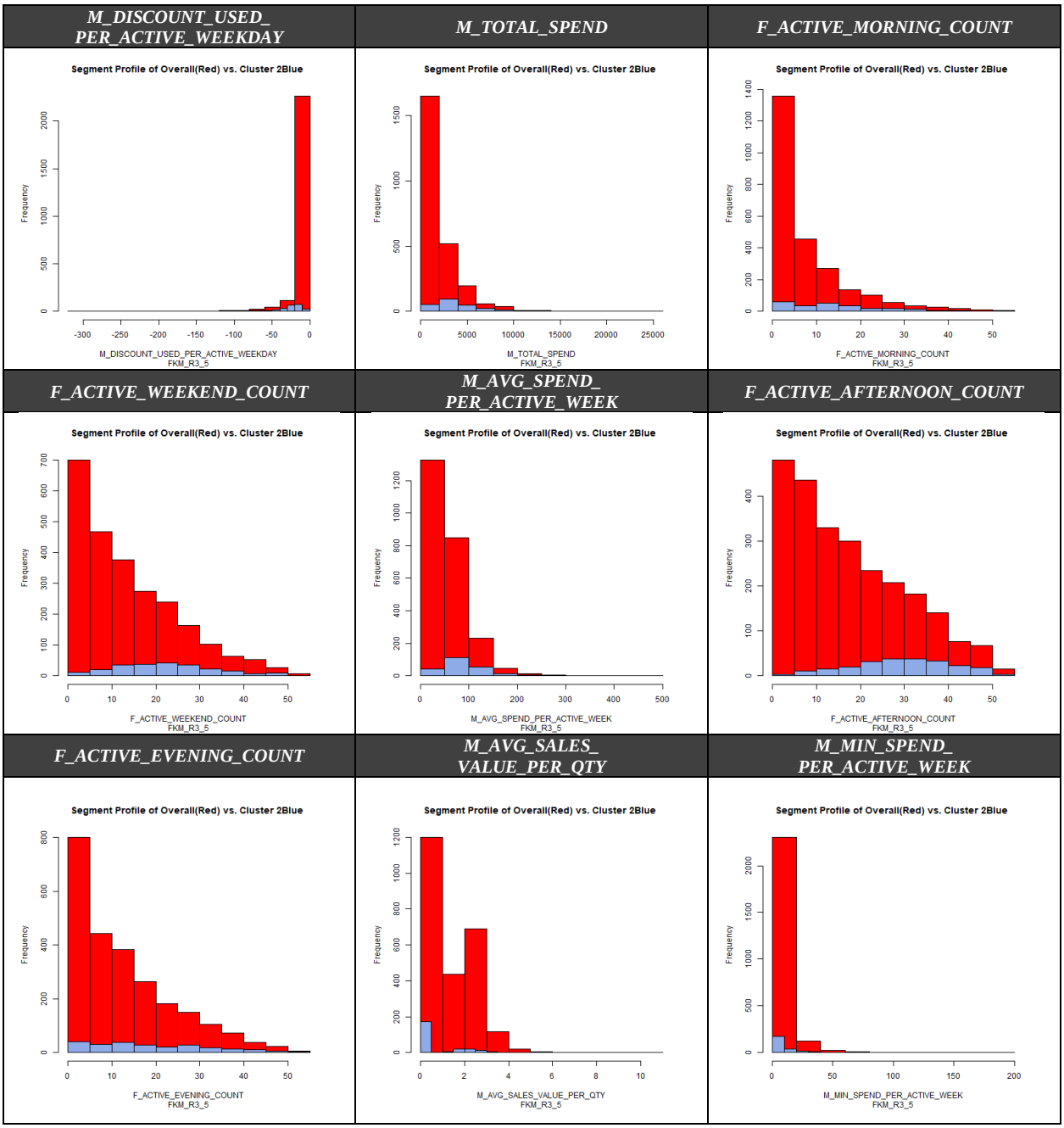


Figure 66: 9 Significant Attributes for Cluster 2 of FKM-R3

Figure 67 reveals FKM-R3's Cluster 3's significant attributes. Cluster 3 shoppers are a small group, with a good mix of frequent and infrequent shoppers.

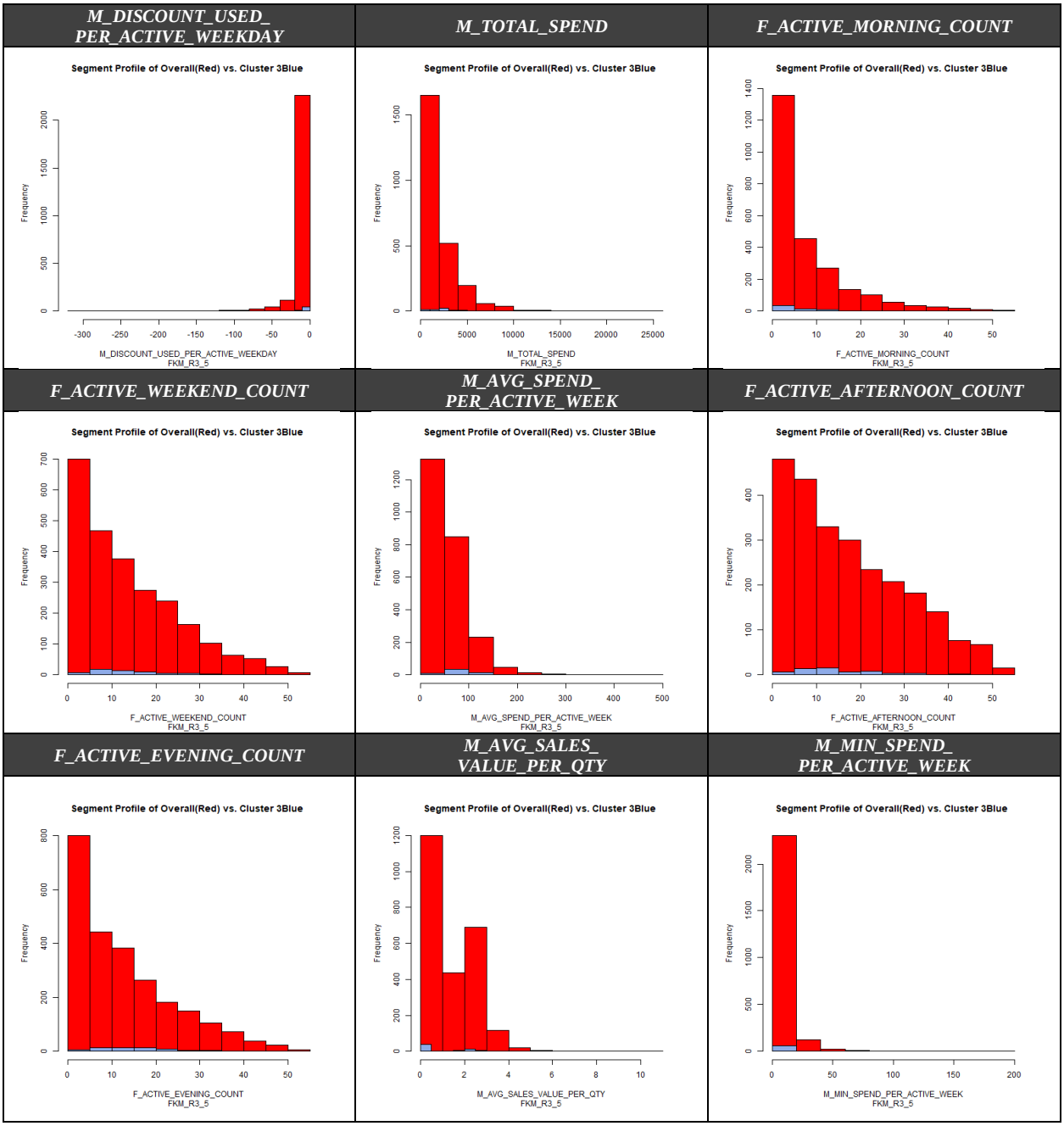


Figure 67: 9 Significant Attributes for Cluster 3 of FKM-R3

Figure 68 reveals FKM-R3's Cluster 4's significant attributes. Cluster 4 shoppers are highly frequent shoppers with deep pockets. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each item.

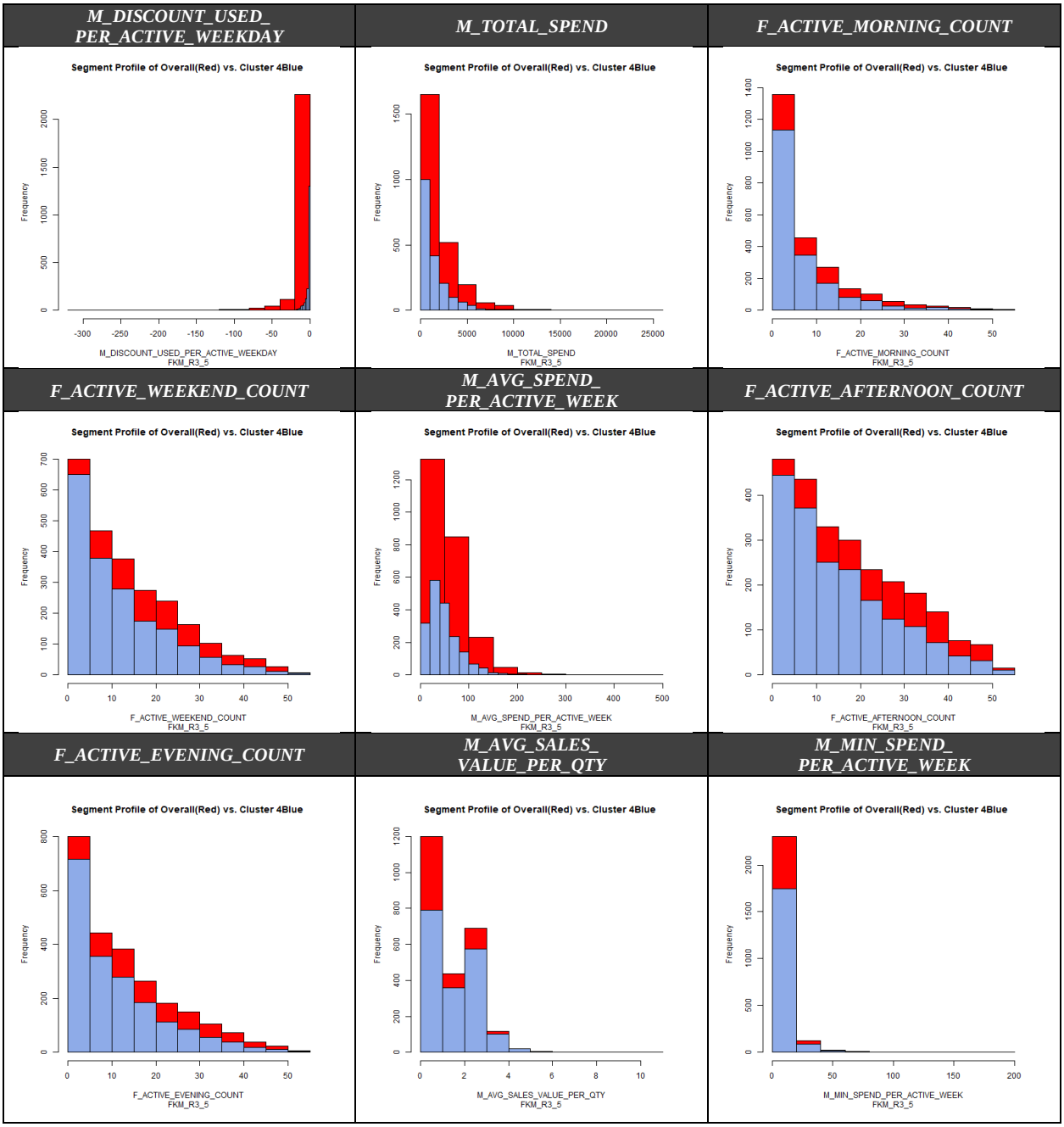


Figure 68: 9 Significant Attributes for Cluster 4 of FKM-R3

Figure 69 reveals FKM-R3's Cluster 5's significant attributes. Cluster 5 shoppers are a good mix of frequent and infrequent shoppers with deep pockets. They use little to no discounts on a weekday. They regularly spend a sizable amount each week. Even with their frequency, they spend more for each item.

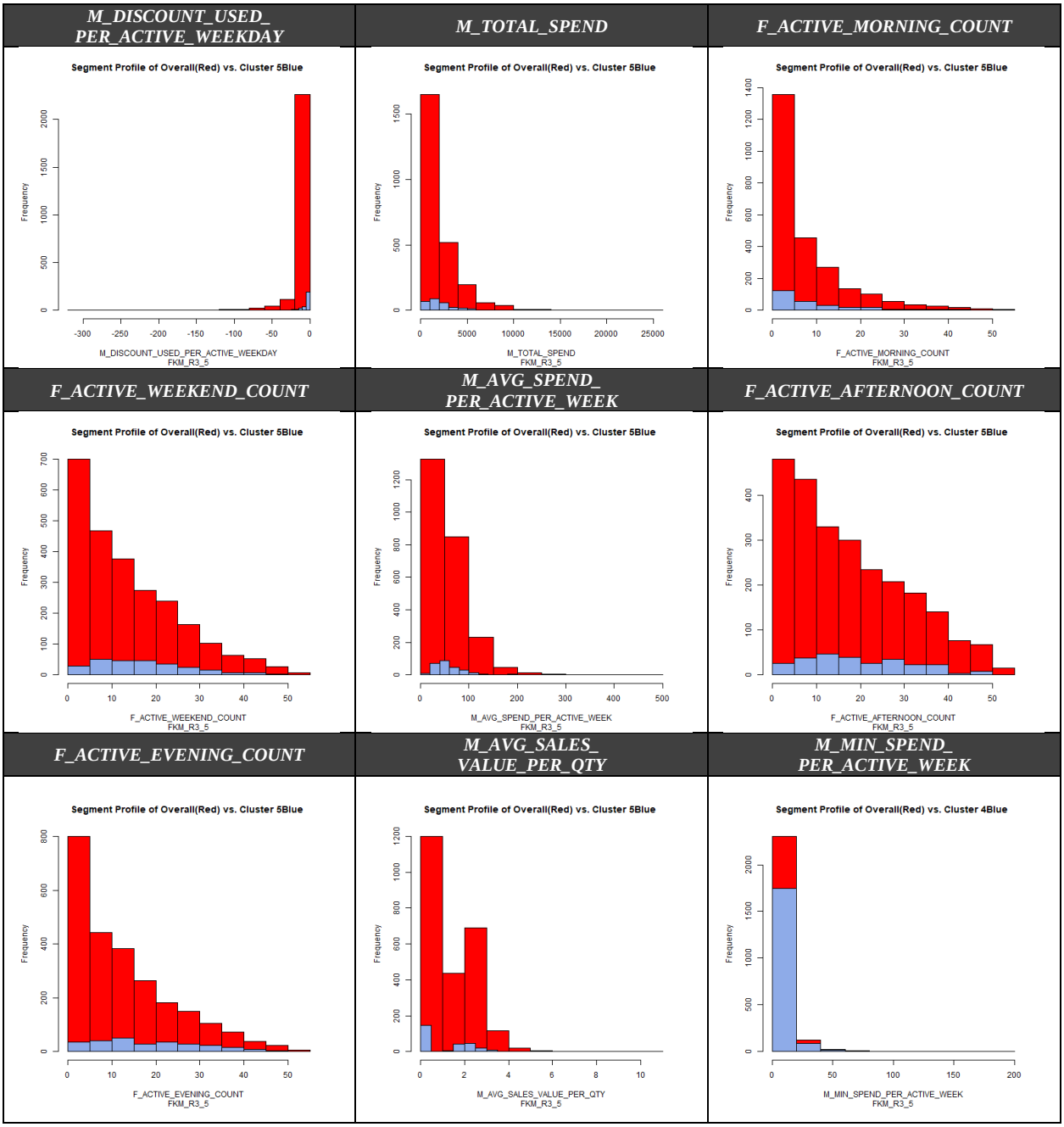


Figure 69: 9 Significant Attributes for Cluster 5 of FKM-R3

GMM-R3 (Gaussian Mixture Model on the as-is form of 3 reduct inputs)

Figure 70 reveals GMM-R3's Cluster 1's significant attributes. Cluster 1 shoppers are a small group of a mix of frequent and infrequent shoppers. They tend to prefer afternoon weekends. They do not use discounts at all, and they don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.

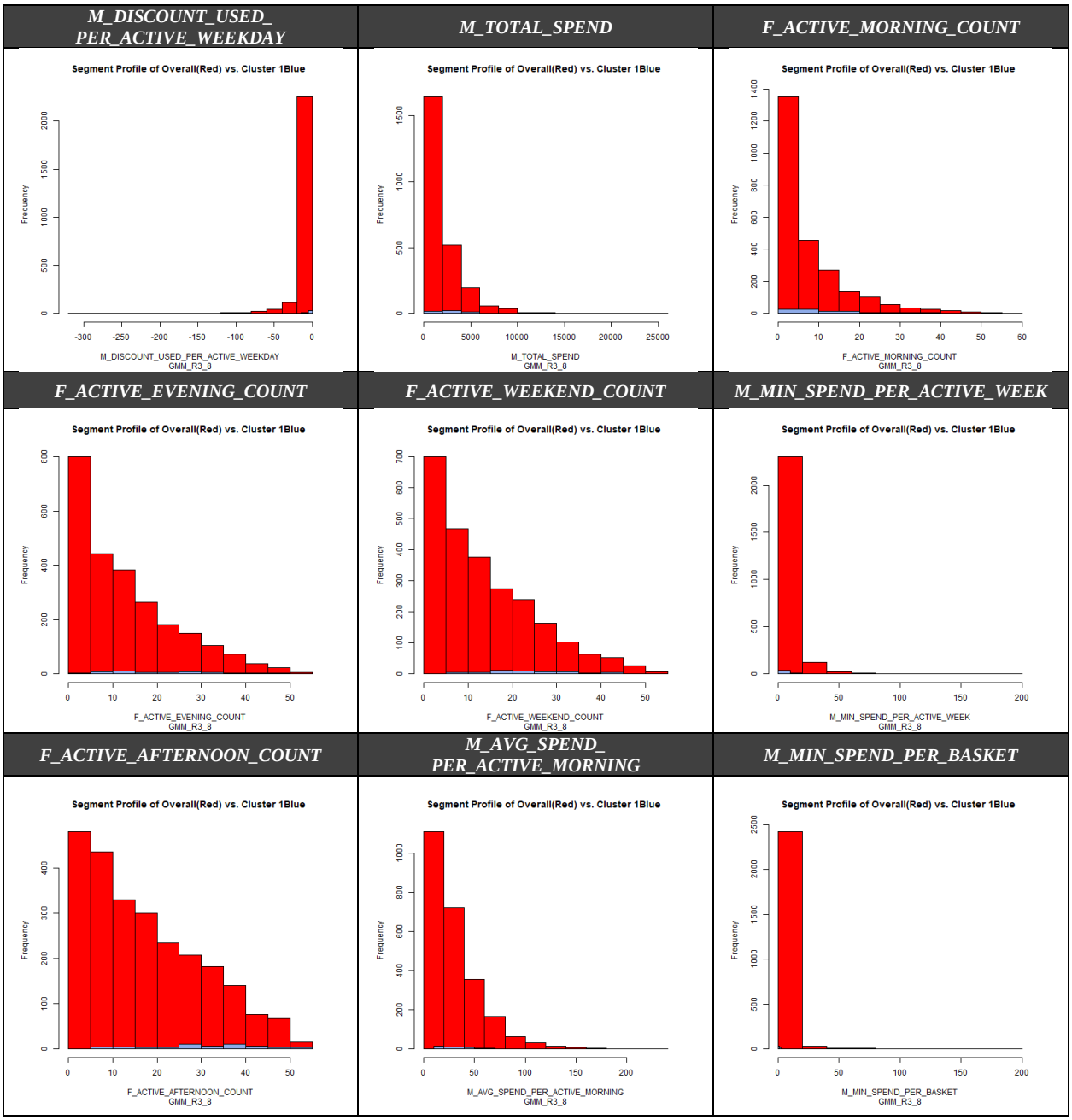


Figure 70: 9 Significant Attributes for Cluster 1 of GMM-R3

Figure 71 reveals GMM-R3's Cluster 2's significant attributes. Cluster 2 shoppers are very frequent shoppers, and active throughout the week. They spend a sizable amount and make full use of the discounts provided.

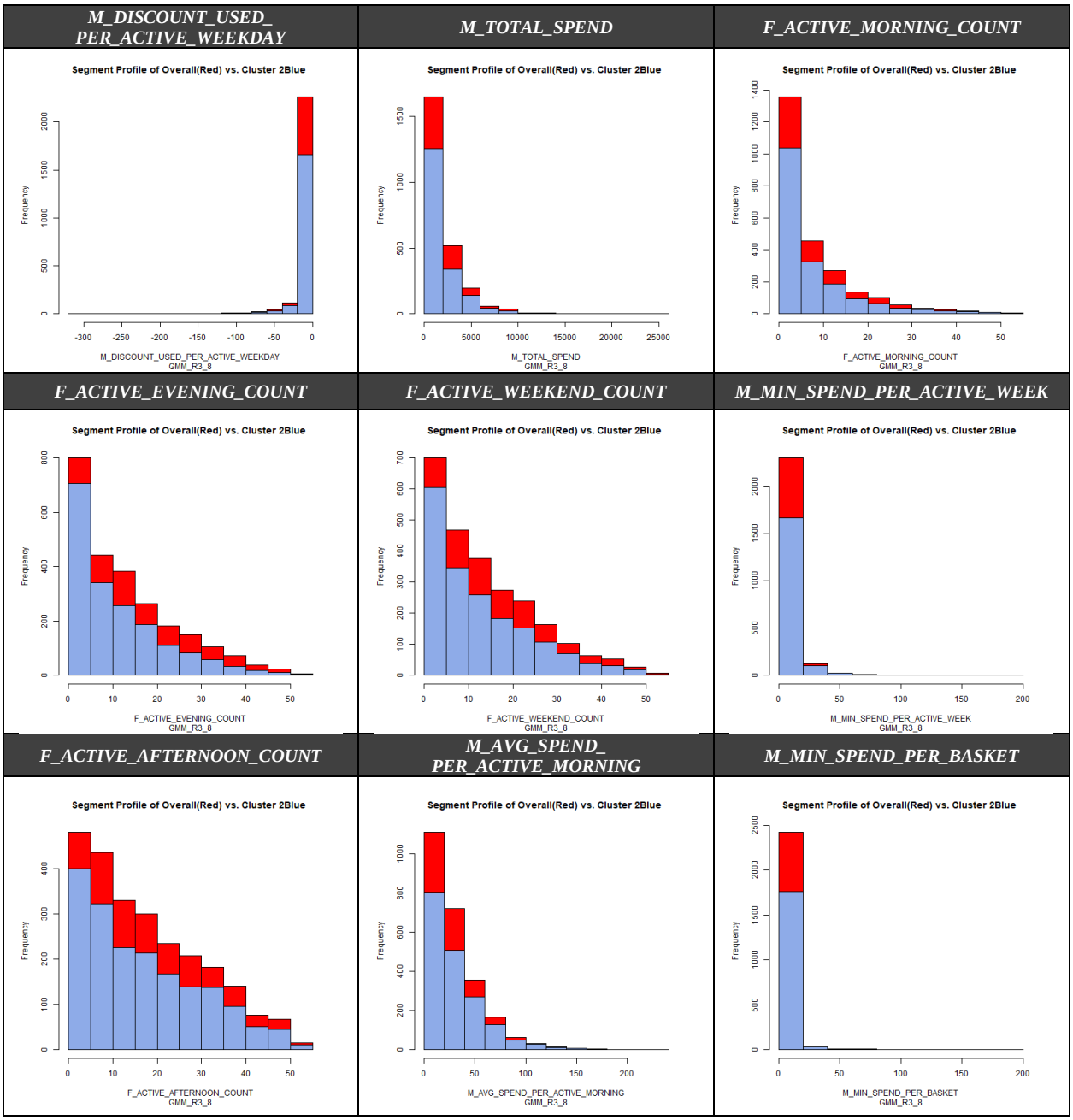


Figure 71: 9 Significant Attributes for Cluster 2 of GMM-R3

Figure 72 reveals GMM-R3's Cluster 3's significant attributes. Similar to Cluster 1, Cluster 3 shoppers are somewhat infrequent shoppers. They hardly spend during the week.

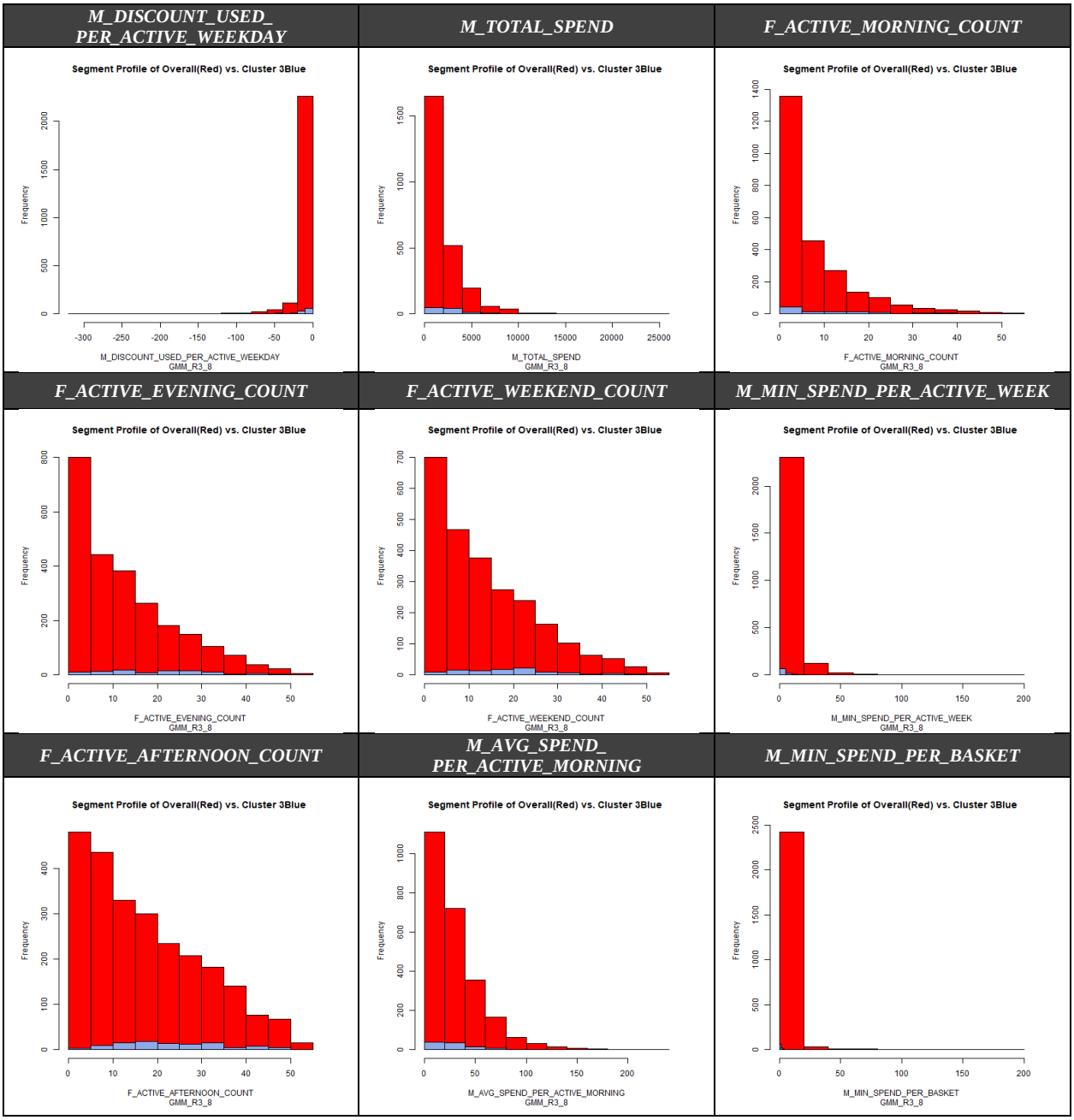


Figure 72: 9 Significant Attributes for Cluster 3 of GMM-R3

Figure 73 reveals GMM-R3's Cluster 4's significant attributes. Cluster 4 shoppers are highly infrequent shoppers.

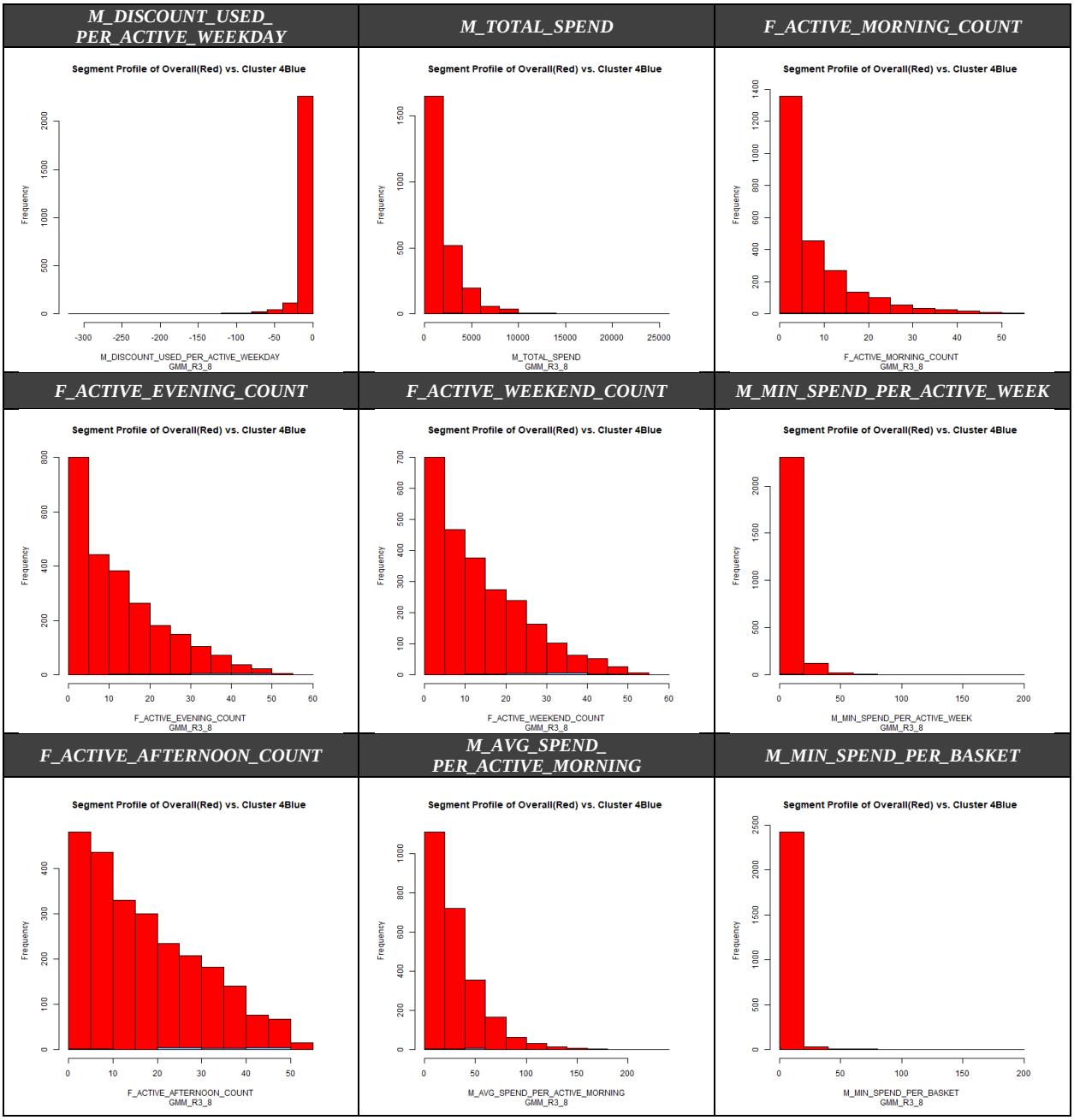


Figure 73: 9 Significant Attributes for Cluster 4 of GMM-R3

Figure 74 reveals GMM-R3's Cluster 5's significant attributes. Cluster 5 shoppers are highly similar to Cluster 4, in that they are highly infrequent shoppers.

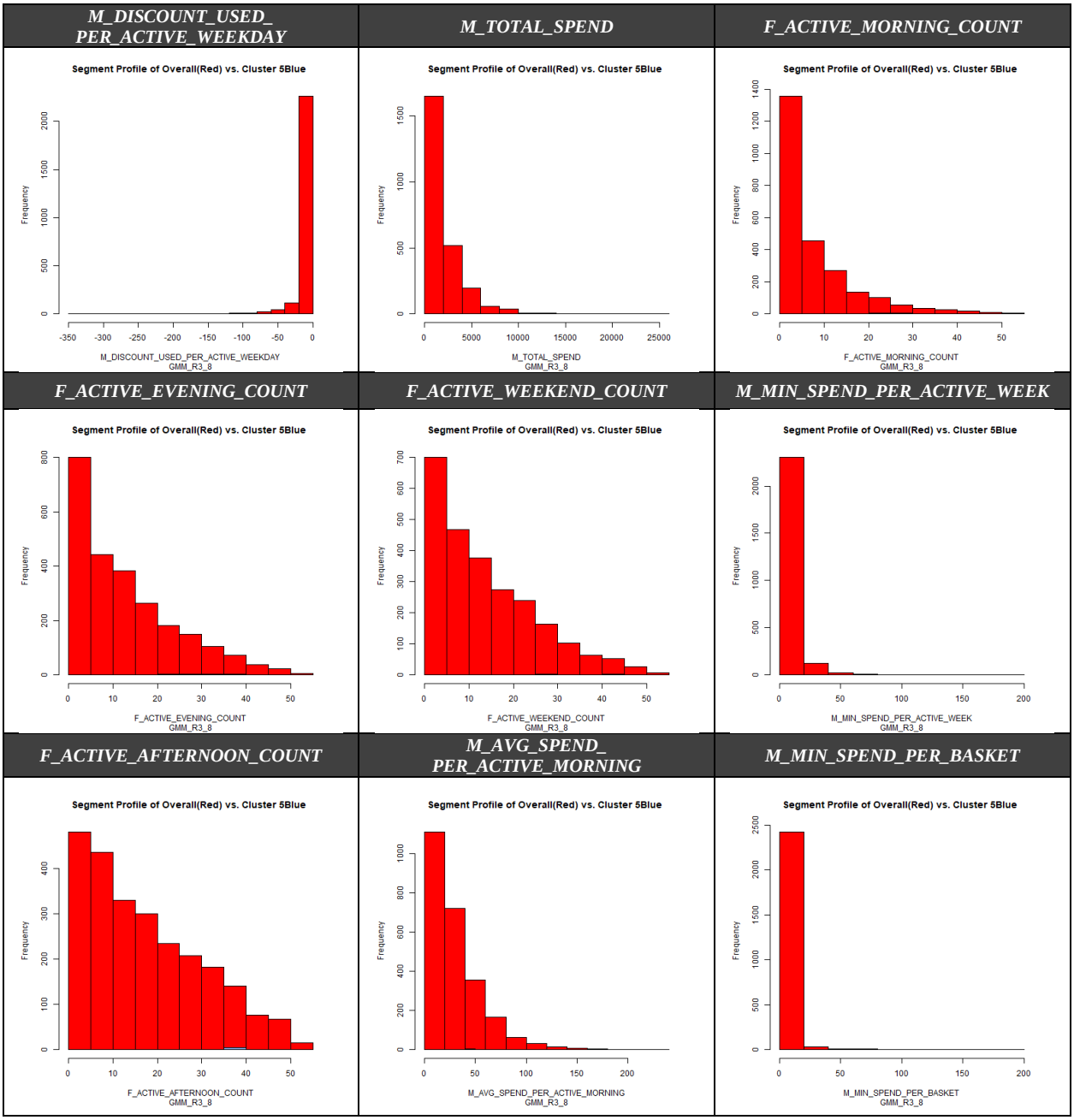


Figure 74: 9 Significant Attributes for Cluster 5 of GMM-R3

Figure 75 reveals GMM-R3's Cluster 6's significant attributes. Cluster 6 shoppers are very similar to Cluster 4 and 5, with highly infrequent shoppers.

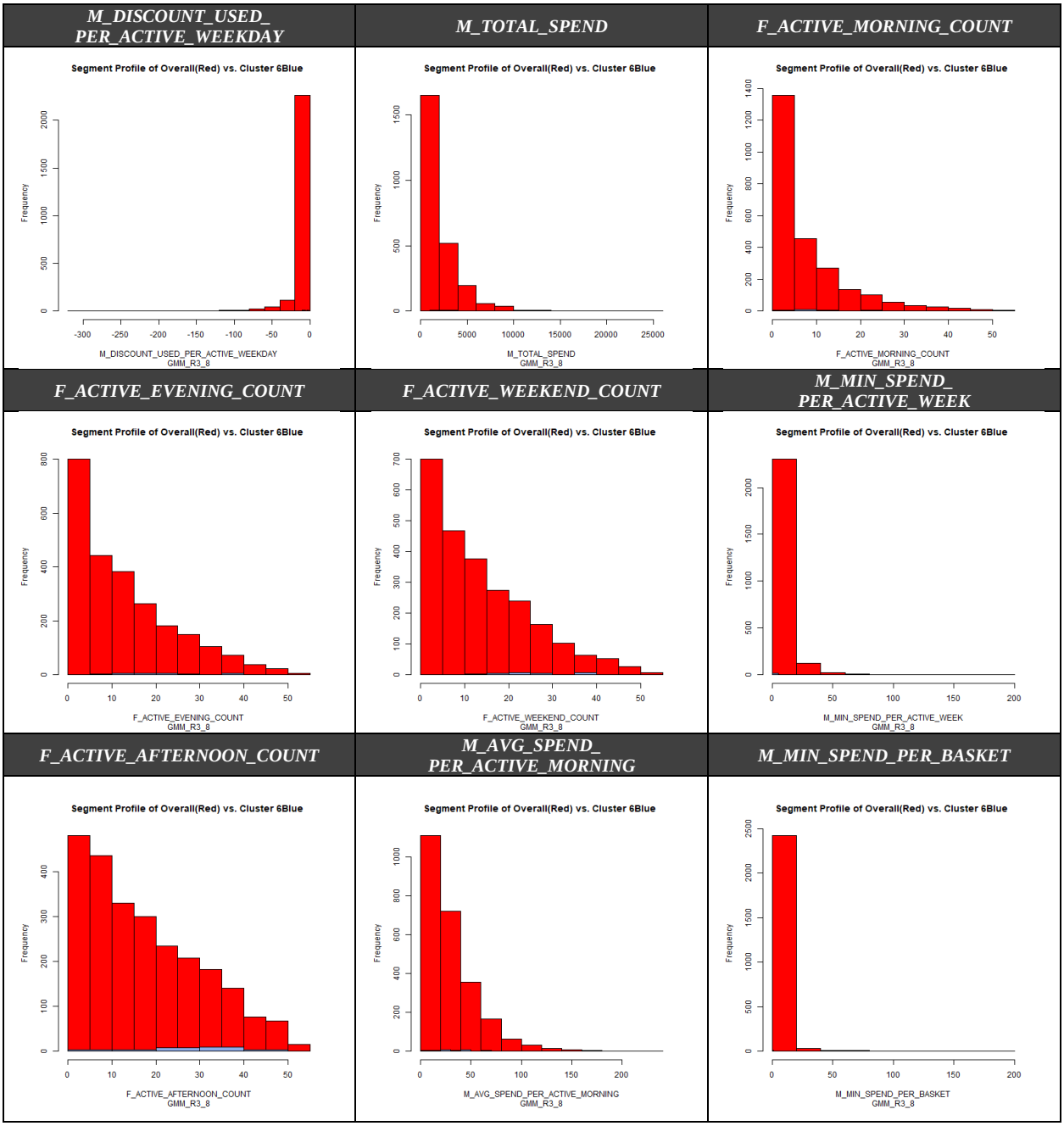


Figure 75: 9 Significant Attributes for Cluster 6 of GMM-R3

Figure 76 reveals GMM-R3's Cluster 7's significant attributes. Cluster 7 shoppers are much more active during the week, and at various times of the day. Their overall spending is lower, and they control their spending each time.

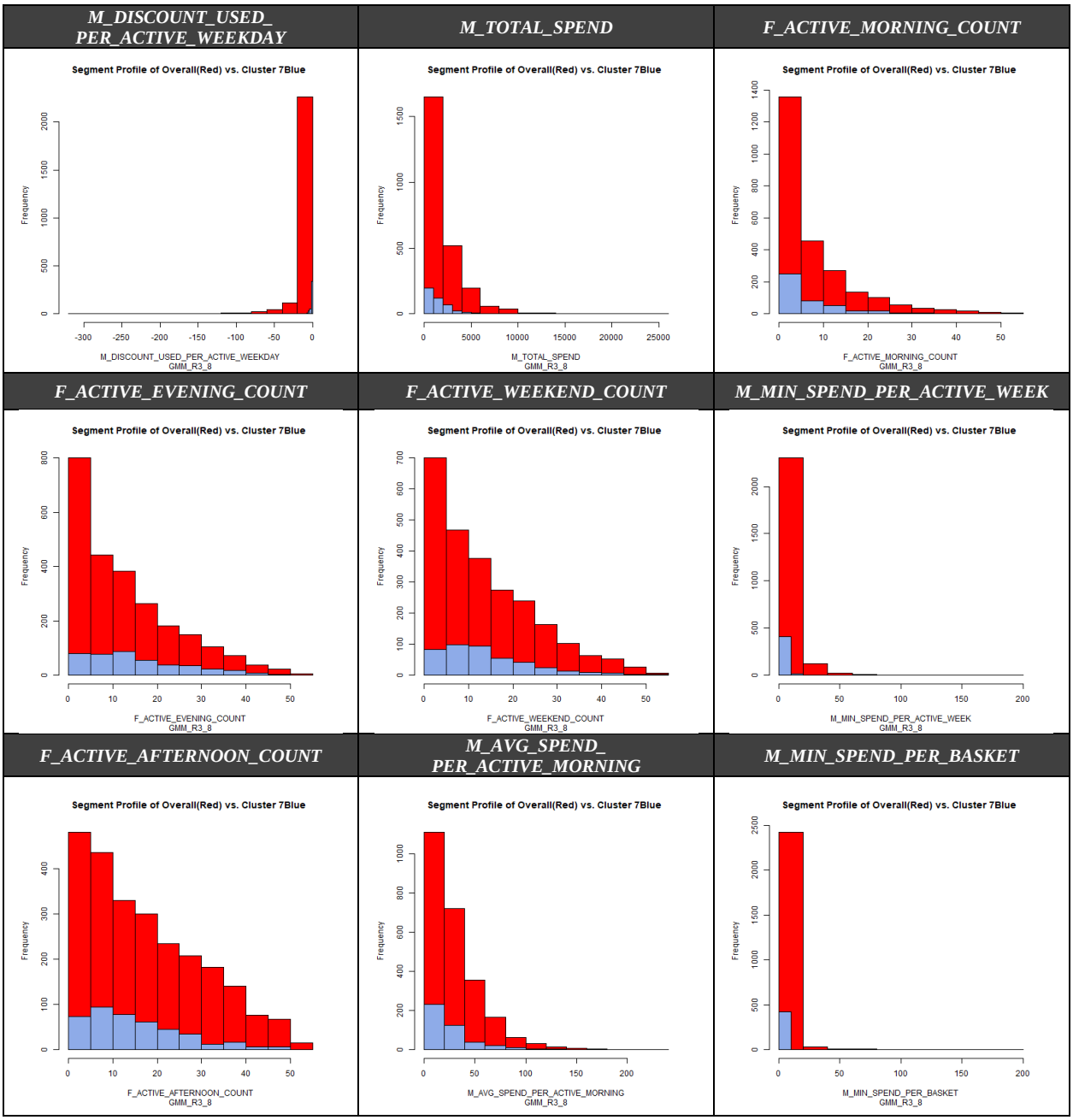


Figure 76: 9 Significant Attributes for Cluster 7 of GMM-R3

Figure 77 reveals GMM-R3's Cluster 8's significant attributes. Cluster 8 shoppers are very similar to Clusters 4, 5 and 6, in that this group has very few members.

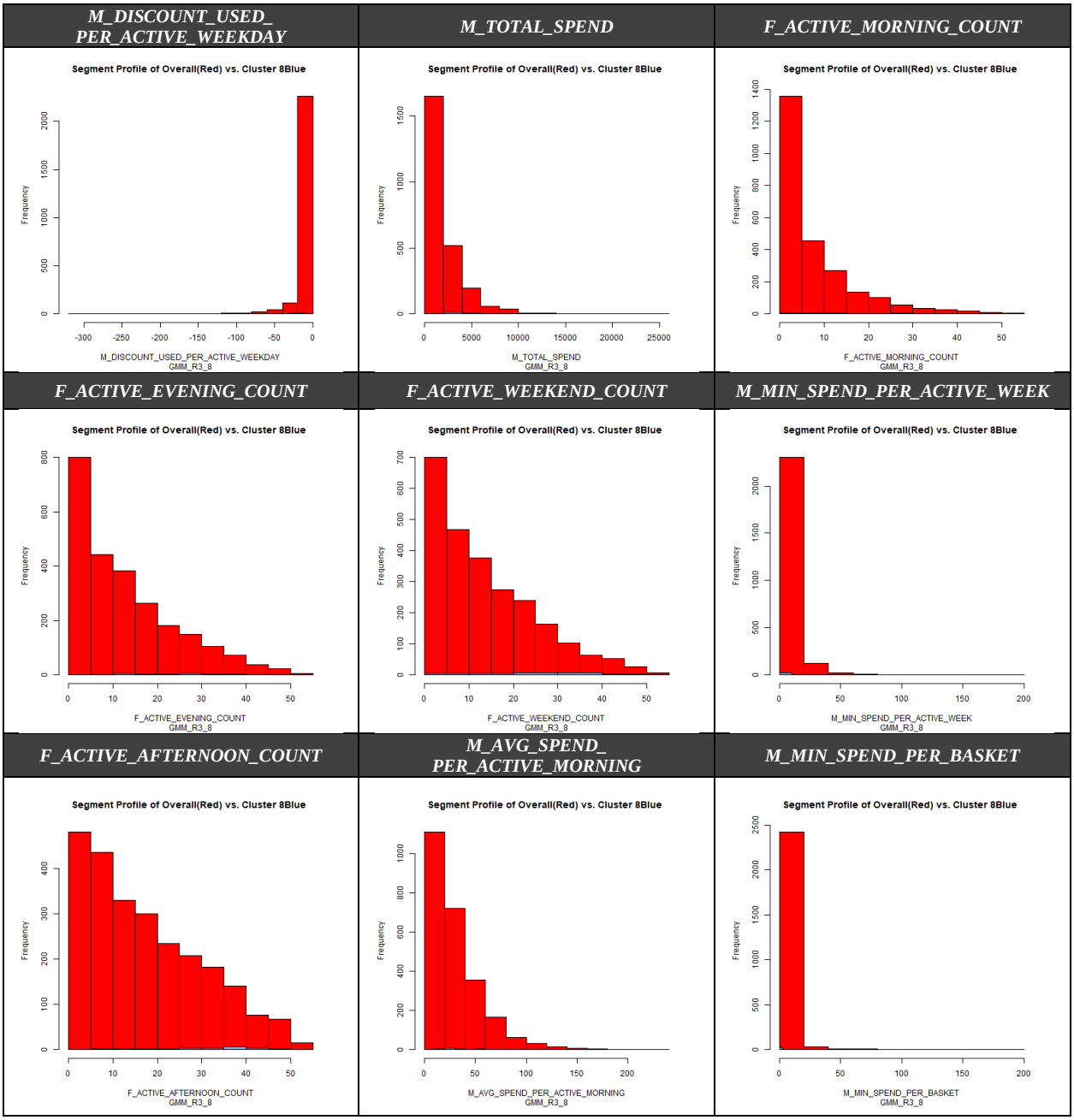


Figure 77: 9 Significant Attributes for Cluster 8 of GMM-R

6.3 Tying Back Clustering Output Results To Household Data

Now that all the clustering outputs have been completed and its characteristics distilled, the next step to understand how these cluster characteristics relate to its household data, as per Figure 78. The household data consists of 801 rows. This will not be a full merge since the cluster data has 2,470 rows. Post the merger, any rows without any accompanying household data will be removed.

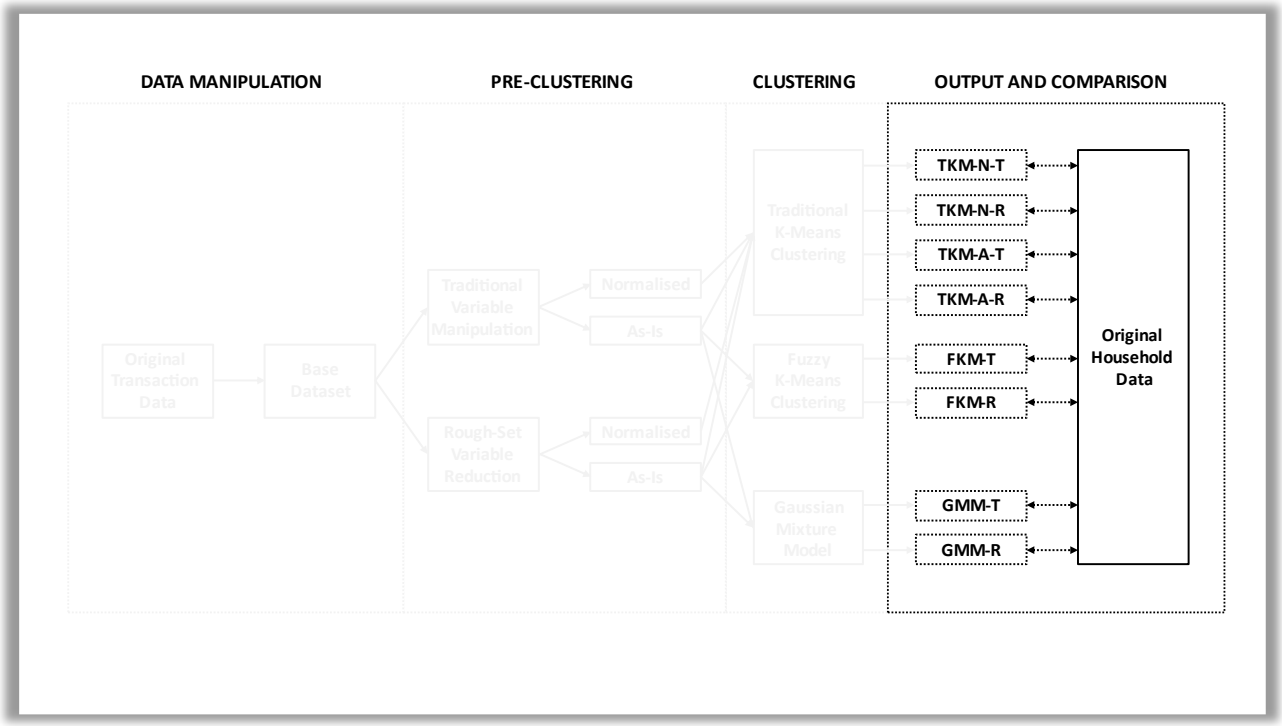


Figure 78: Capstone Workflow with emphasis on current step

Before merging with our cluster file, below is a summary of the household data. It covers age, income level, homeowner, household composition, household size and kids' category.

Age	Range	Income	Range	Homeowner	Range	HH_Composition	Range	HH_Size	Range	Kids_Category	Range
65+	9%	Above 250K	1%	Homeowner	63%	2 Adults Kids	23%	1	32%	None/Unknown	70%
55-64	7%	200 to 249K	1%	Probable Owner	1%	2 Adults No Kids	32%	2	40%	1	14%
45-54	36%	150 to 199K	5%	Renter	5%	1 Adult Kids	6%	3	14%	2	7%
35-44	24%	100 to 149K	9%	Probable Renter	1%	Single Female	18%	4	7%	3+	9%
25-34	18%	50 to 99K	36%	Unknown	29%	Single Male	12%	5+	8%		
19-24	6%	Under T50K	48%			Unknown	9%				

Table 8: Summary Statistics of Household Data

This household demographic data is then merged with a dataset with all generated clusters, using EM's Merge Node. The common identifier is the household ID. This merged data is then pushed to an OSI Node to generate plots.

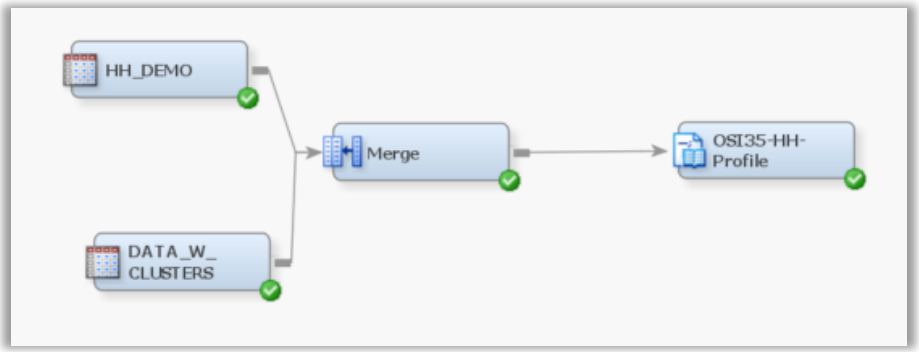
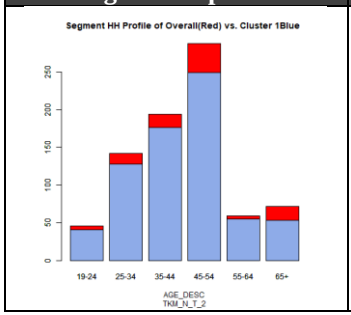
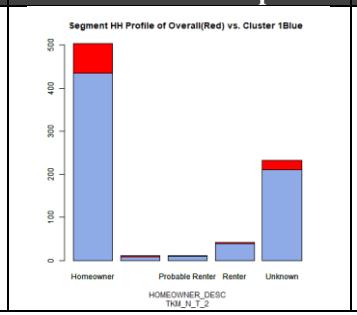
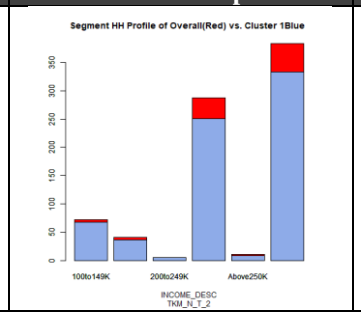
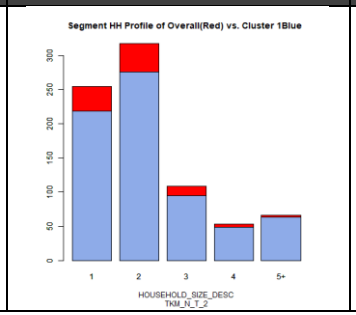
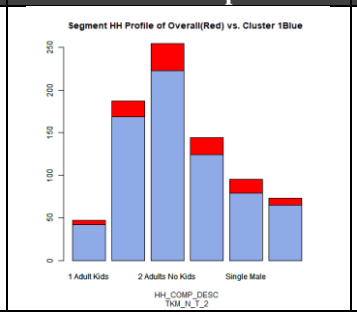
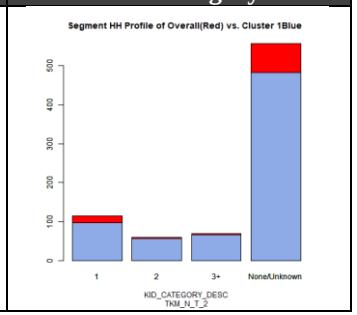


Figure 79: HH Demo merging with Cluster Data Before Being Ingested Into An OSI Node

6.4 TKM Clustering Output Household Characteristics

TKM-N-T (Traditional *k*-means clustering on normalized form of 16 non-collinear inputs)

TKM-N-T Cluster 1 Consumer Traits: Cluster 1 shoppers are a good mixture of frequent and infrequent shoppers. They are price sensitive, since they use many discounts and sales per quantity are generally on the lower side. Their total spending is sizable.
Tying Consumer Traits Back To Household Traits: Sizable spending with the use of large discounts ties nicely with a large homeownership.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

TKM-N-T Cluster 2 Consumer Traits: Cluster 2 shoppers are somewhat infrequent shoppers. They are not as price sensitive as they tend to use less discounts on weekdays. Their total spending is much lower than Cluster 1 shoppers. Given their infrequency, they tend to spend slightly more for each basket.
Tying Consumer Traits Back To Household Traits: Low spending may relate to a lower income bracket and smaller household size.

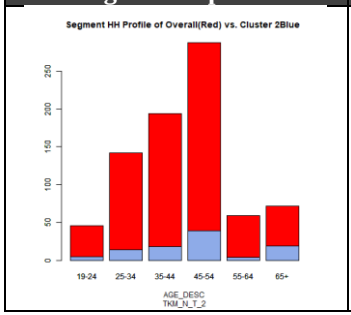
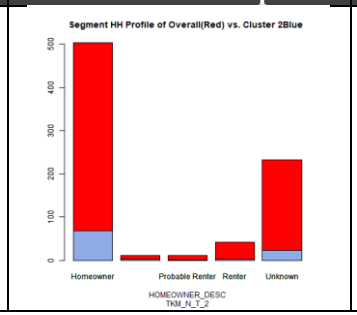
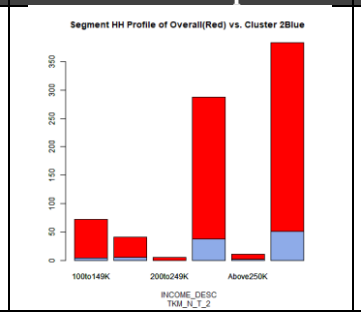
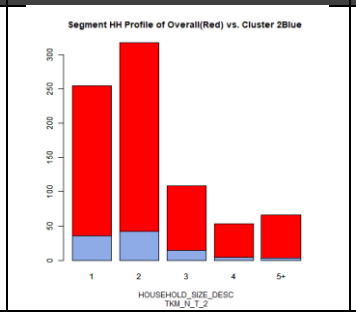
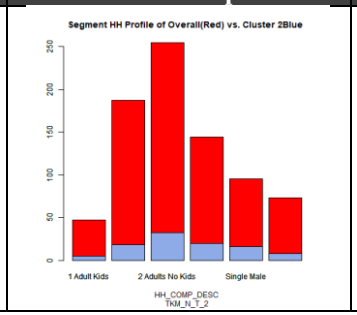
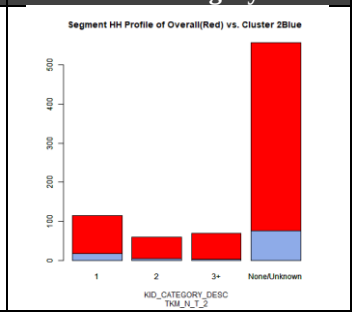
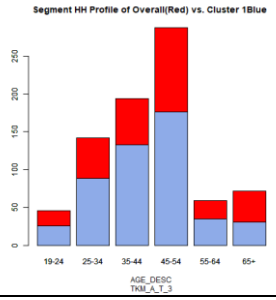
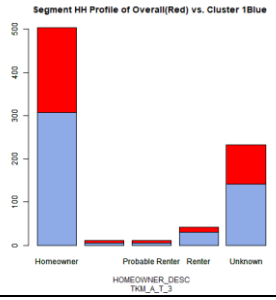
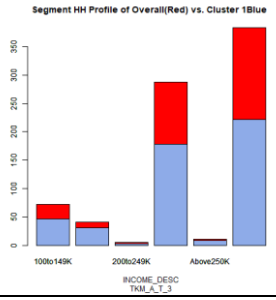
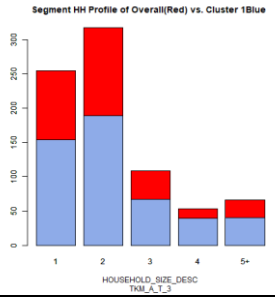
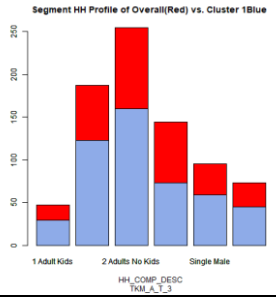
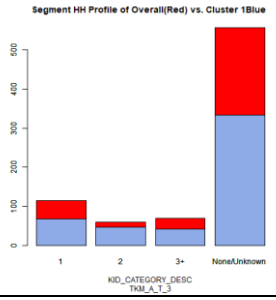
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

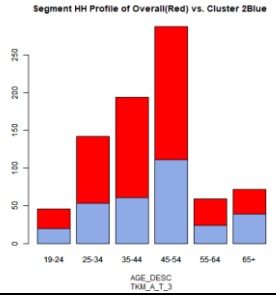
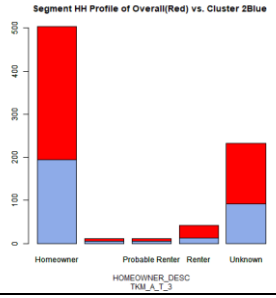
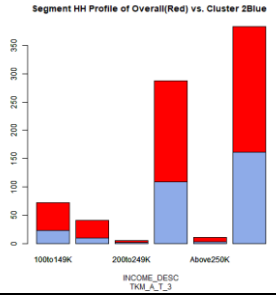
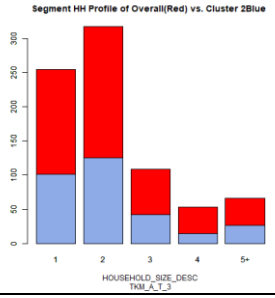
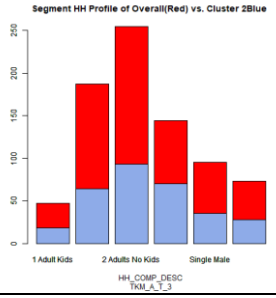
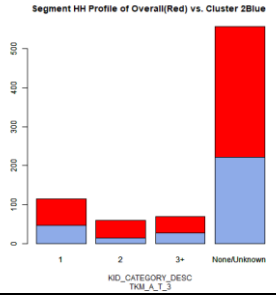
Figure 80: TKM-N-T Clustering Output Household Characteristics

TKM-A-T (Traditional *k*-means clustering on as-is form of 16 non-collinear inputs)

TKM-A-T Cluster 1 Consumer Traits: Cluster 1 shoppers tend to be a more frequent shopper with a preference for weekday shopping. They spend more overall, and yet are price conscious due to their use of discounts. Given their frequency, they spend less for each basket.
Tying Consumer Traits Back To Household Traits: Large spending may be due to homeownership, and higher income brackets.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

TKM-A-T Cluster 2 Consumer Traits: Cluster 2 shoppers are somewhat infrequent shoppers. They spend less each time, and hardly make use of discounts on weekdays. Overall spending is also on the lower side
Tying Consumer Traits Back To Household Traits: Low spending can be explained by lower income brackets.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

TKM-A-T Cluster 3 Consumer Traits: Cluster 3 shoppers are highly infrequent shoppers. They tend to use little to no discounts on weekdays and spend less overall than Cluster 2 shoppers. They are much less active through the week. Given their frequency, they tend to spend much more for each basket.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category

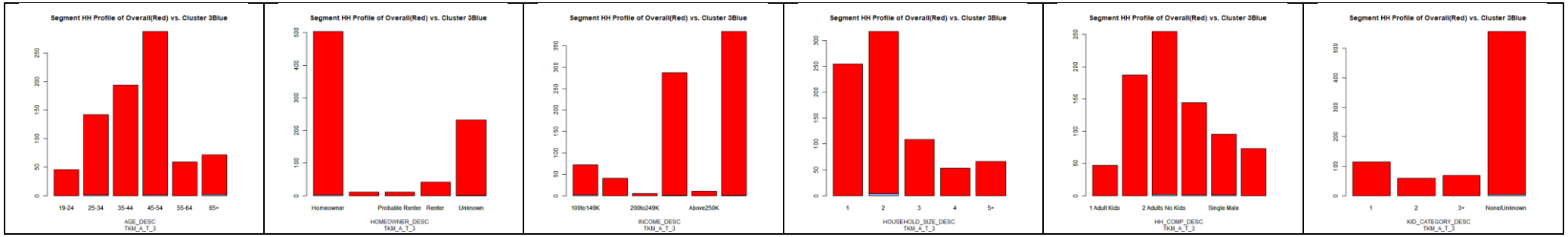
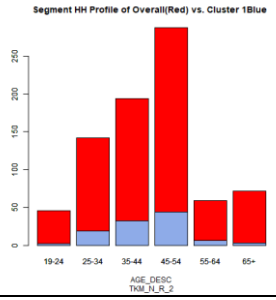
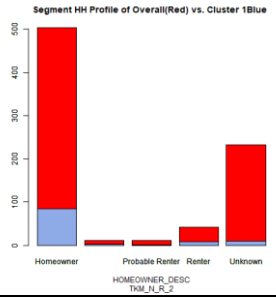
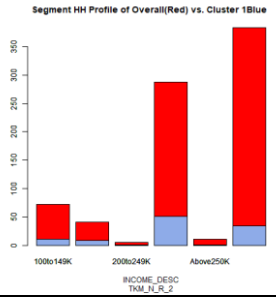
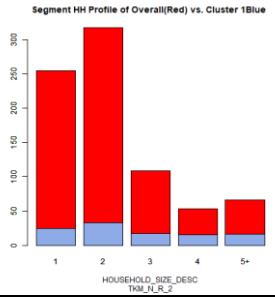
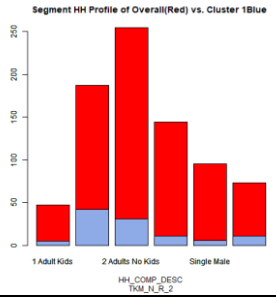
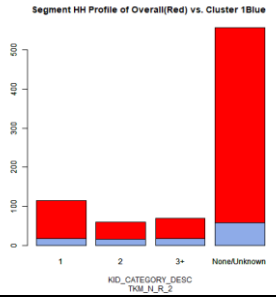


Figure 81: TKM-A-T Clustering Output Household Characteristics

TKM-N-R (Traditional *k*-means clustering on normalized form of 2 reduct inputs)

TKM-N-R Cluster 1 Consumer Traits: Cluster 1 shoppers are a good spread of frequent and infrequent shoppers. They are price conscious and have a propensity to use more discounts. They don't spend much overall. Given their frequency, they spend less for each basket.
Tying Consumer Traits Back To Household Traits: Price conscious due to older age bracket and homeownership.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

TKM-N-R Cluster 2 Consumer Traits: Cluster 2 shoppers are highly frequent shoppers with deeper pockets. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each basket.
Tying Consumer Traits Back To Household Traits: Price insensitivity due to larger income bracket.

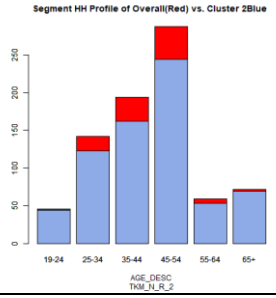
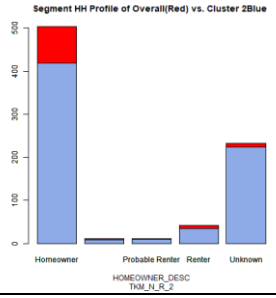
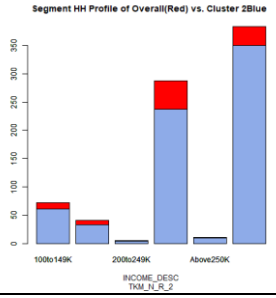
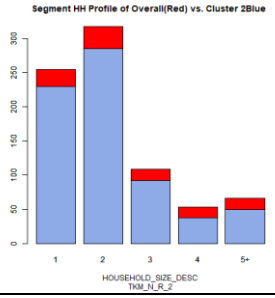
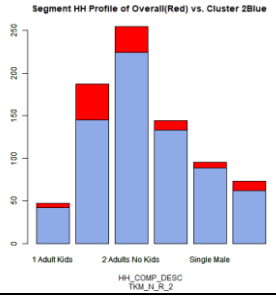
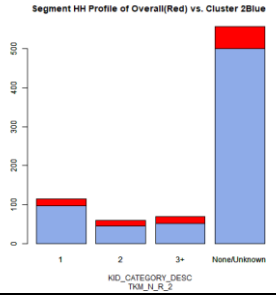
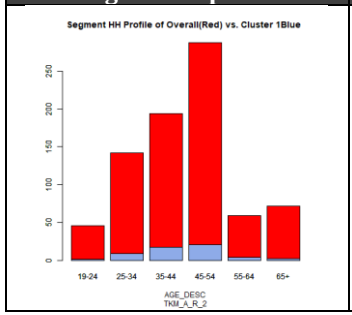
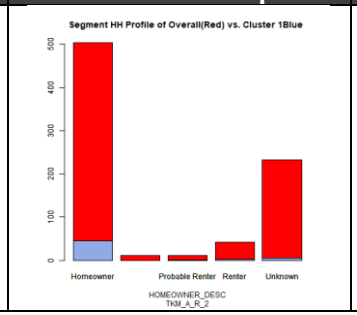
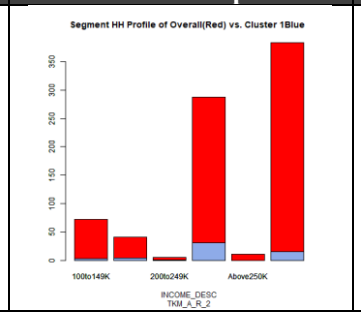
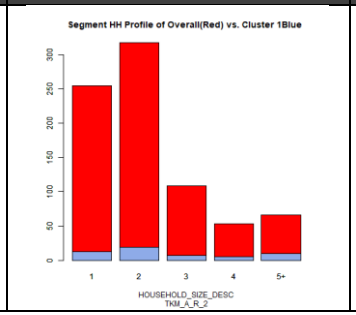
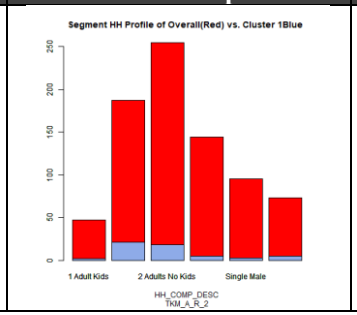
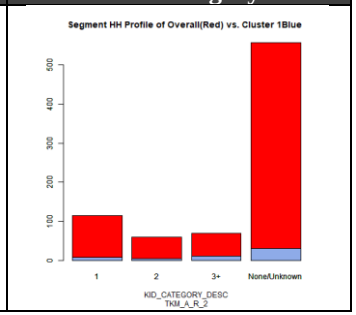
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

Figure 82: TKM-N-R Clustering Output Household Characteristics

TKM-A-R (Traditional *k*-means clustering on as-is form of 2 reduct inputs)

TKM-A-R Cluster 1 Consumer Traits: Cluster 1 shoppers are highly infrequent shoppers and are price conscious (with low average sales per quantity). They have a propensity not to use any discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.

Tying Consumer Traits Back To Household Traits: Price conscious due to greater proportion with lower income bracket.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

TKM-A-R Cluster 2 Consumer Traits: Cluster 2 shoppers are highly frequent shoppers with deeper pockets. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each item.

Tying Consumer Traits Back To Household Traits: Price insensitivity due to larger income bracket.

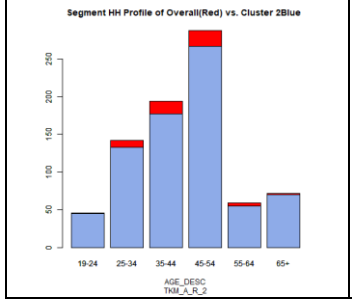
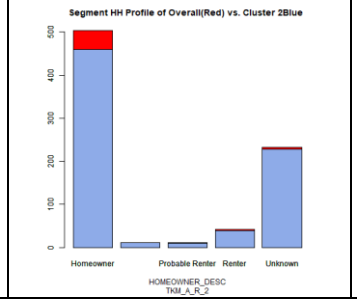
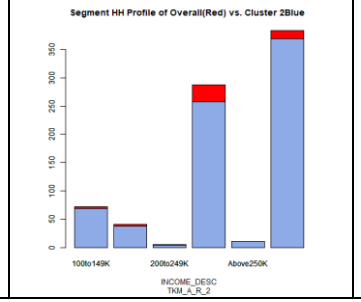
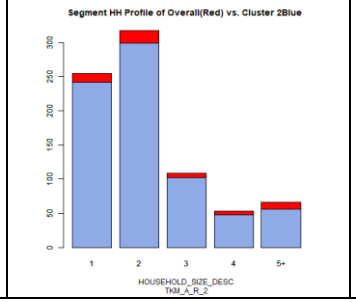
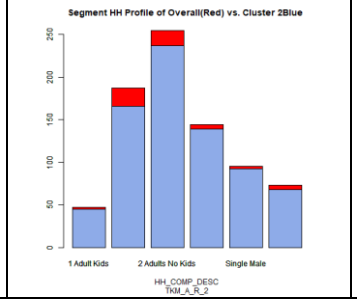
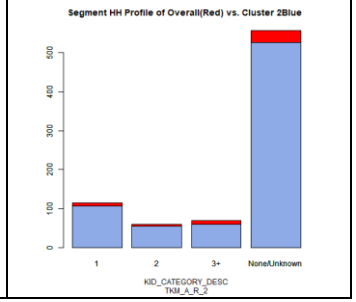
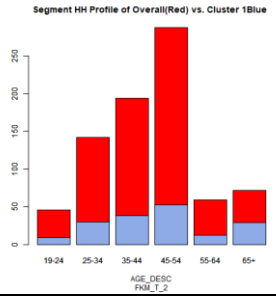
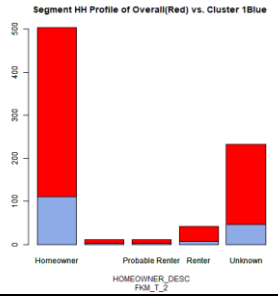
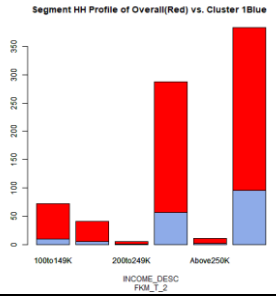
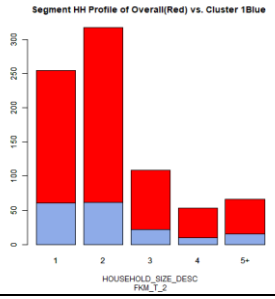
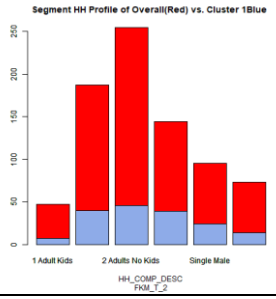
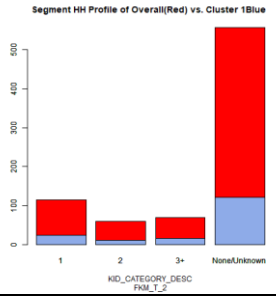
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

Figure 83: TKM-A-R Clustering Output Household Characteristics

6.5 FKM Clustering Output Household Characteristics

FKM-T (Fuzzy *k*-means clustering on the as-is form of traditional variable manipulation)

FKM-T Cluster 1 Consumer Traits: Cluster 1 shoppers are somewhat infrequent shoppers but are not price conscious. They have a propensity not to use any discounts. They don't spend much overall, but when they do, they don't mind paying more for each product. They somewhat shop all week around.
Tying Consumer Traits Back To Household Traits: Lower income bracket should technically lead to more price sensitivity but it's not the case here.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

FKM-T Cluster 2 Consumer Traits: Cluster 2 shoppers are a good mix of frequent and infrequent shoppers. They spend a lot and have no qualms about using discounts. This leads to a lower sales value per quantity. They actively shop all week around.
Tying Consumer Traits Back To Household Traits: Large income bracket should technically lead to less price sensitivity but it's not the case here.

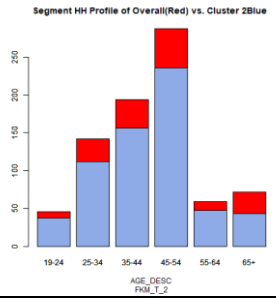
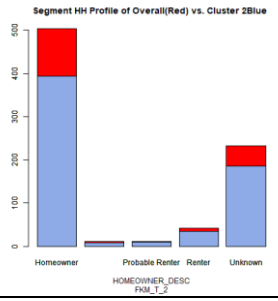
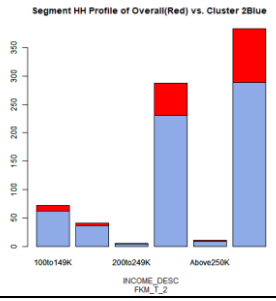
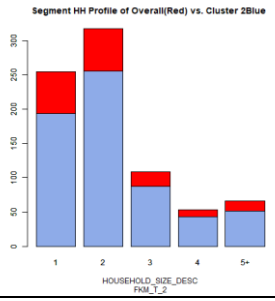
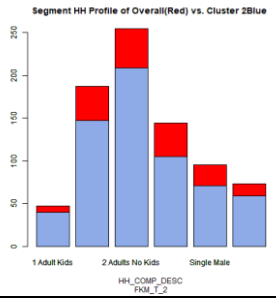
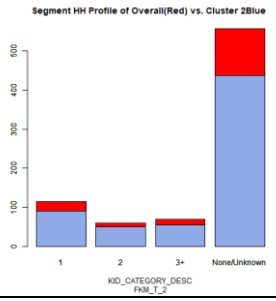
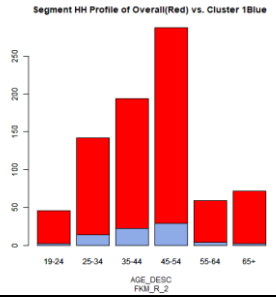
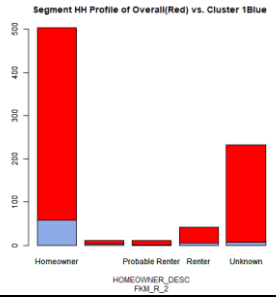
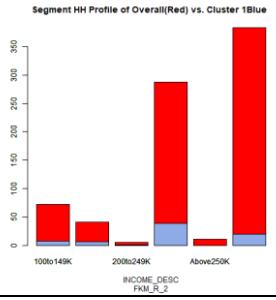
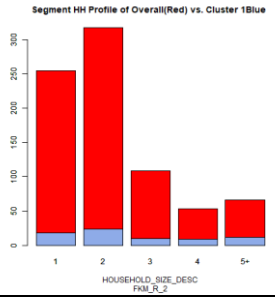
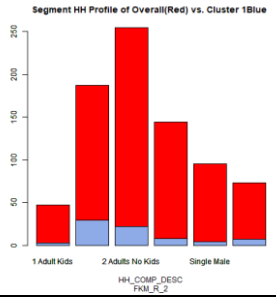
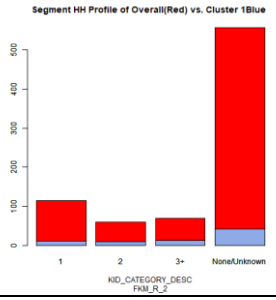
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

Figure 84: FKM-T Clustering Output Household Characteristics

FKM-R (Fuzzy *k*-means clustering on the as-is form of 2 reduct inputs)

FKM-R Cluster 1 Consumer Traits: Cluster 1 shoppers are a good mix of frequent and infrequent shoppers and don't spend much. When they do shopping, it's usually in the afternoons.

Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

FKM-R Cluster 2 Consumer Traits: Cluster 2 shoppers are highly frequent shoppers with deep pockets. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each item.

Tying Consumer Traits Back To Household Traits: Price insensitivity due to larger income bracket.

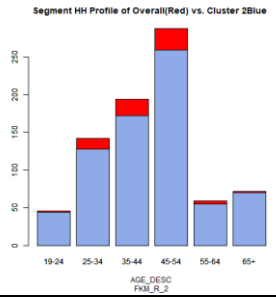
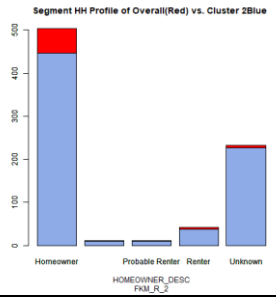
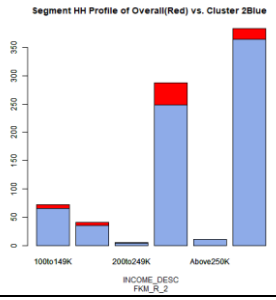
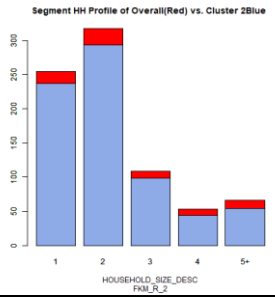
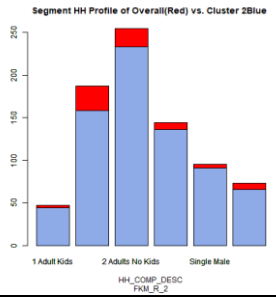
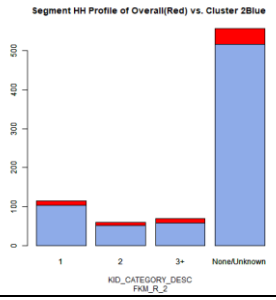
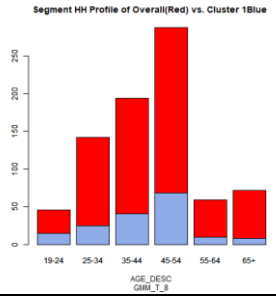
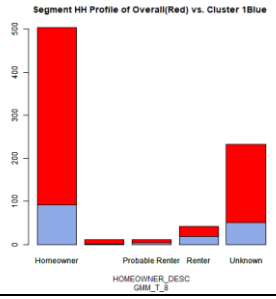
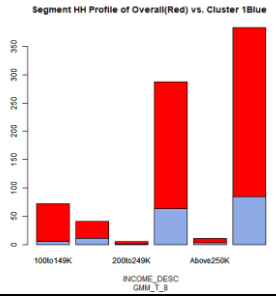
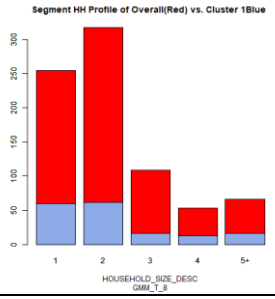
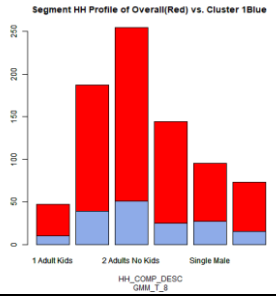
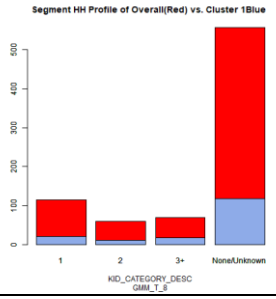
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

Figure 85: FKM-R Clustering Output Household Characteristics

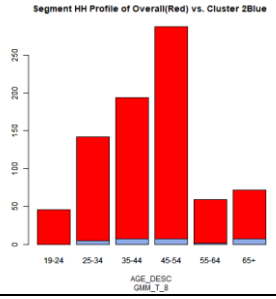
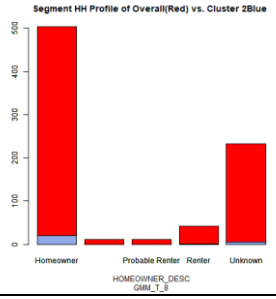
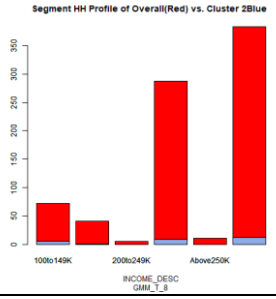
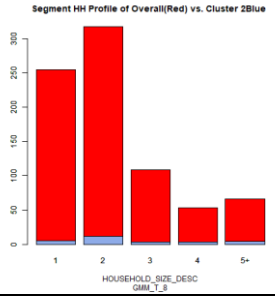
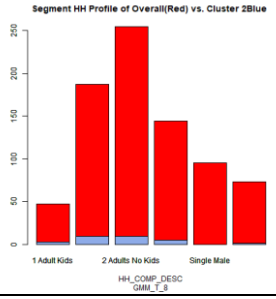
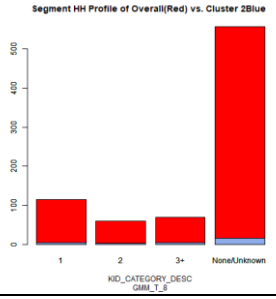
6.6 GMM Clustering Output Household Characteristics

GMM-T (Gaussian Mixture Model on the as-is form of traditional variable manipulation)

GMM-T Cluster 1 Consumer Traits: Cluster 1 shoppers are highly infrequent shoppers and are price conscious. They have a propensity to use plenty of discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.
Tying Consumer Traits Back To Household Traits: Low income brackets leading to greater discount usage.

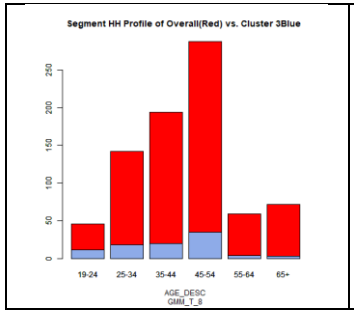
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

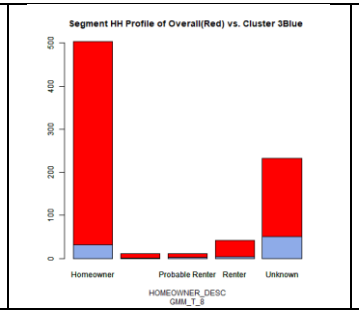
GMM-T Cluster 2 Consumer Traits: Cluster 2 shoppers are somewhat frequent shoppers and are price conscious. They have a propensity not to use any discounts. They don't spend much overall. They prefer afternoons or evenings to shop.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

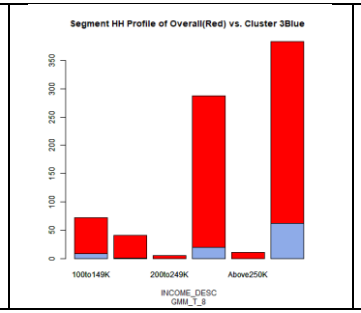
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

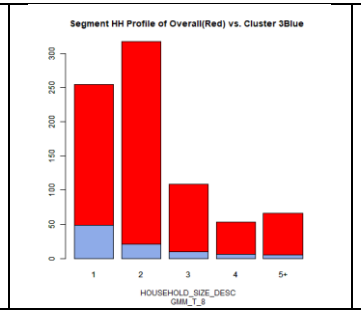
GMM-T Cluster 3 Consumer Traits: Cluster 3 shoppers have a mixed spread of frequent and infrequent shoppers. They prefer not to use any discounts and spend very little overall.
Tying Consumer Traits Back To Household Traits: Low spending due to lower income.

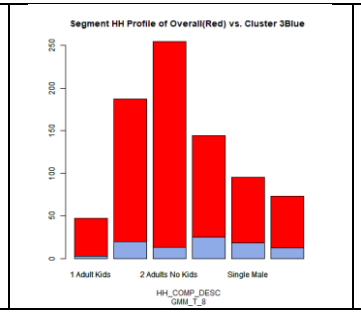
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category

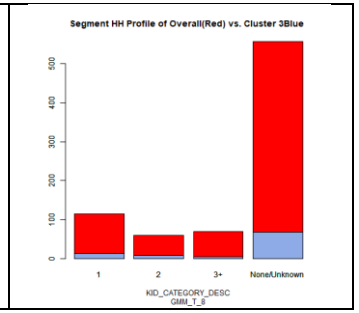




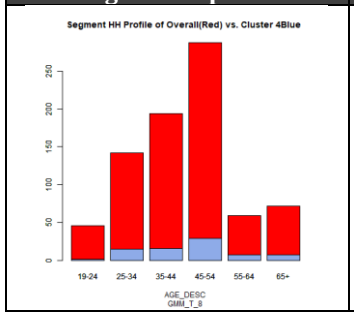
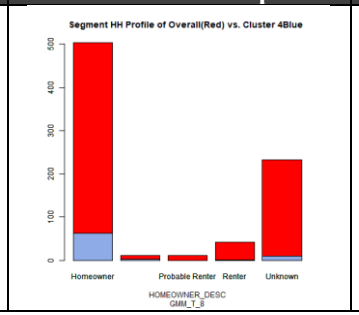
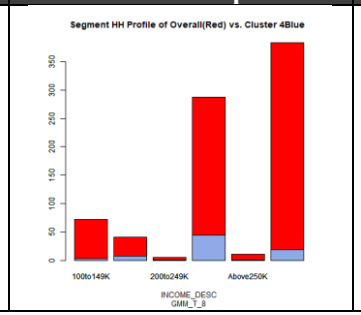
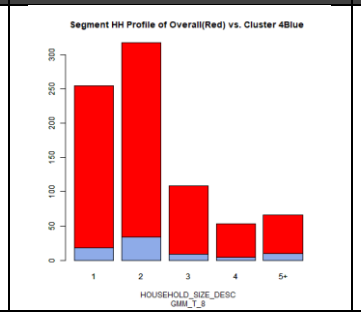
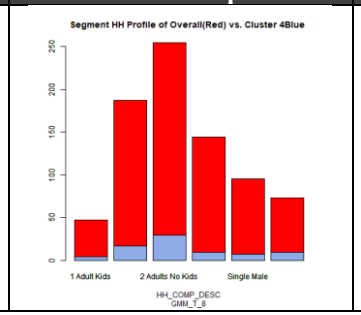
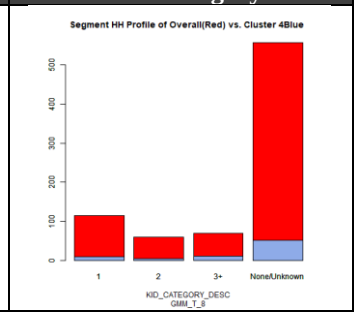






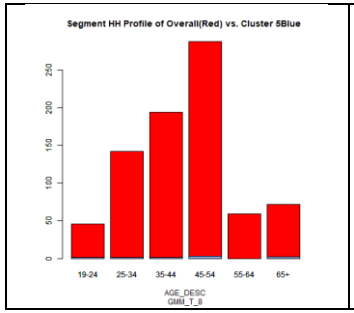


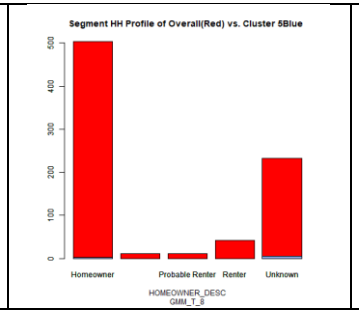
GMM-T Cluster 4 Consumer Traits: Cluster 4 shoppers is a small group with a mix of frequent and infrequent shoppers. They have a tendency not to use any discounts.
Tying Consumer Traits Back To Household Traits: Low spending due to lower income.

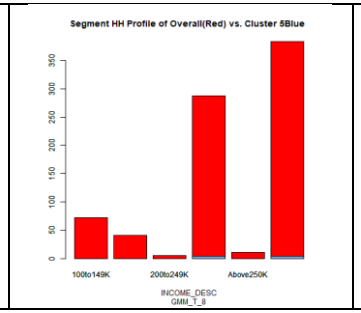
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

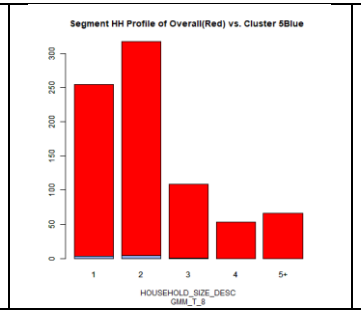
GMM-T Cluster 5 Consumer Traits: Cluster 5 shoppers are highly infrequent shoppers and do not use any discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.
Tying Consumer Traits Back To Household Traits: Low spending and infrequency due to older base, with unknown homeownership and low income.

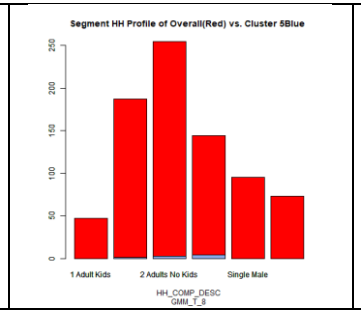
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category

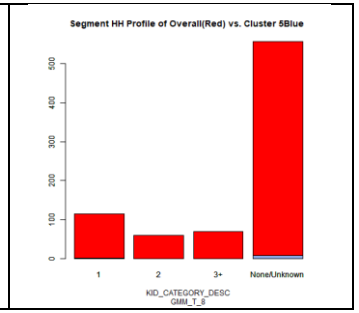




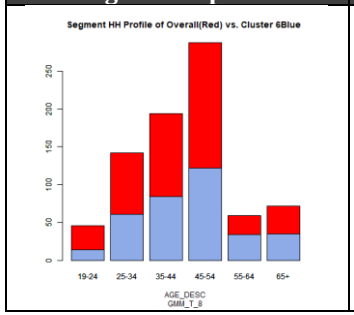
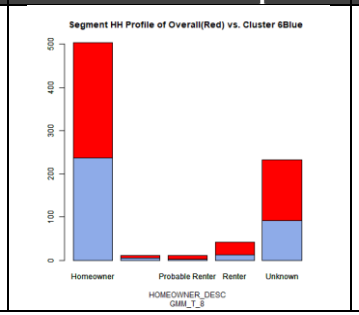
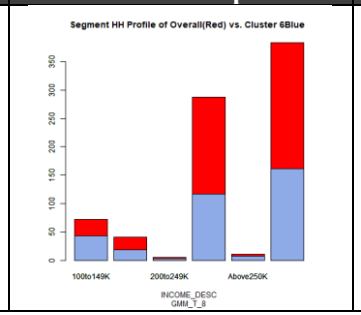
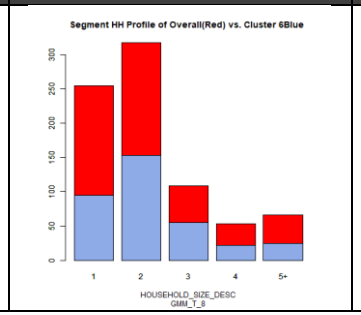
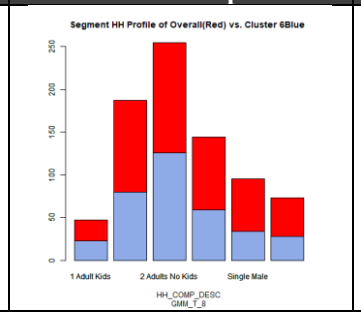
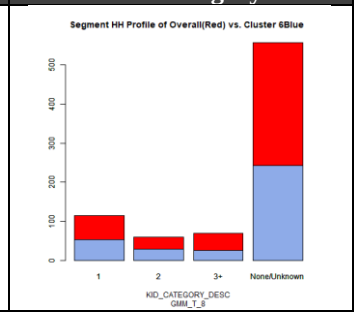




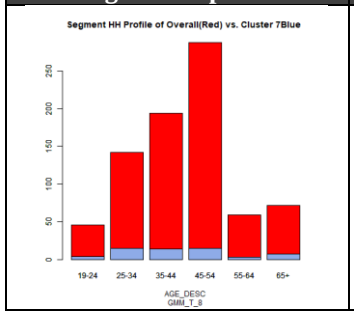
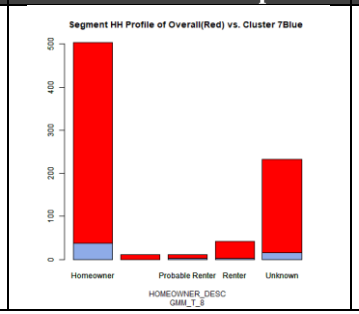
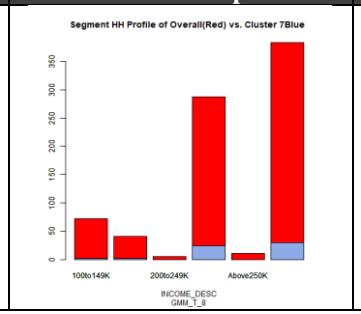
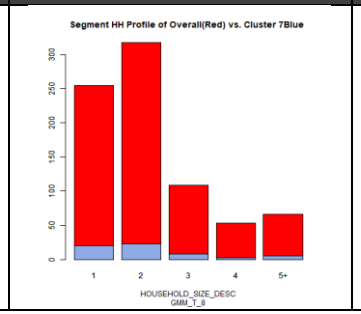
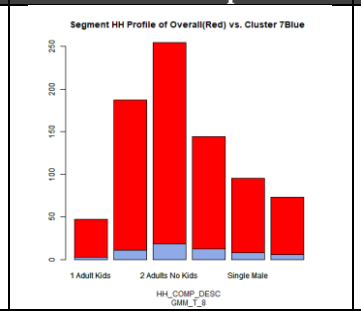
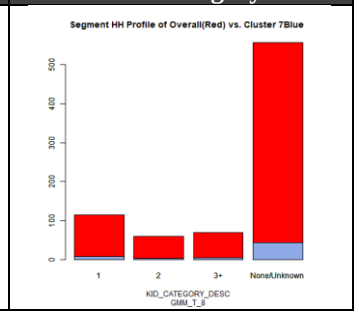




GMM-T Cluster 6 Consumer Traits: Cluster 6 shoppers are highly frequent shoppers, who much prefer weekend shopping. They also prefer to shop either in the morning, afternoon, or evening. They don't spend much and hardly use discounts.
Tying Consumer Traits Back To Household Traits: Good balance of income, age and household composition may explain frequent shopping.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

GMM-T Cluster 7 Consumer Traits: Cluster 7 shoppers are very similar to Cluster 6, but less frequency in shopping and spends less as a result.
Tying Consumer Traits Back To Household Traits: An older base with lower overall income leading to lesser spending.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

GMM-T Cluster 8 Consumer Traits: Cluster 8 shoppers are a small group of shoppers with a good spread in terms of preferred visit frequency. They have very little spending. When they do spend, they use discounts.

Tying Consumer Traits Back To Household Traits: Low spending with propensity for discounts ties nicely with an older base with lower income.

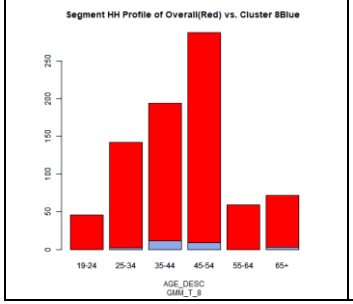
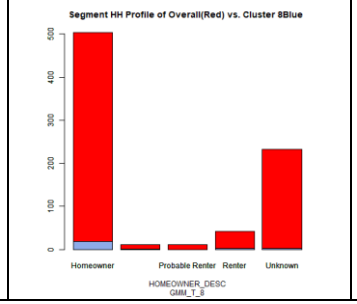
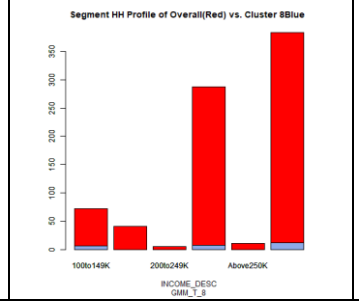
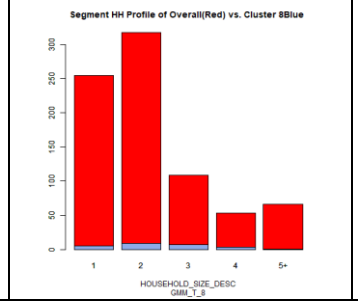
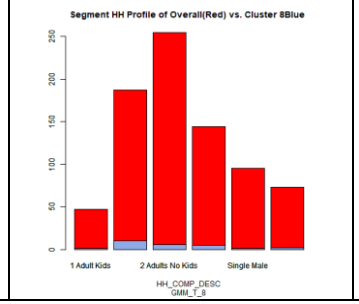
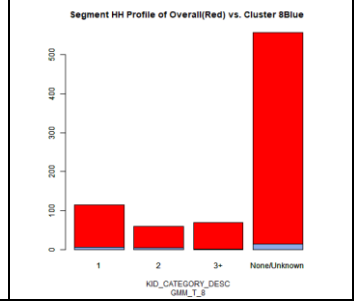
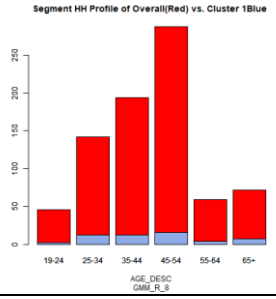
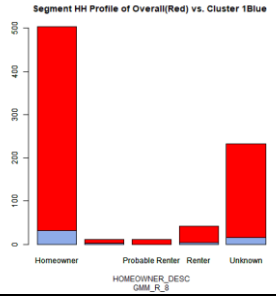
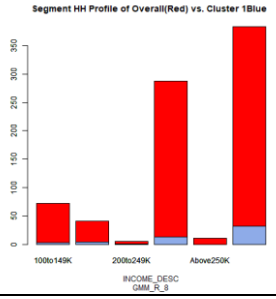
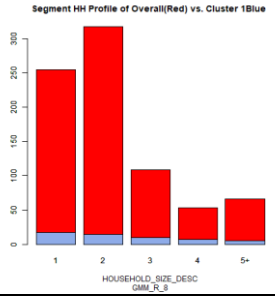
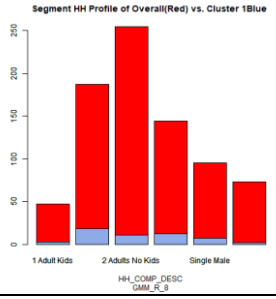
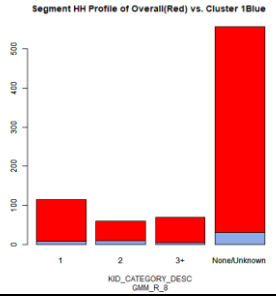
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

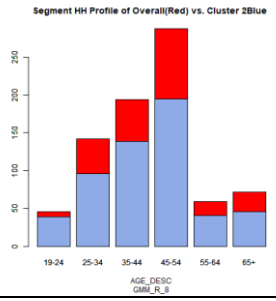
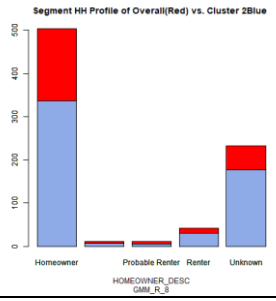
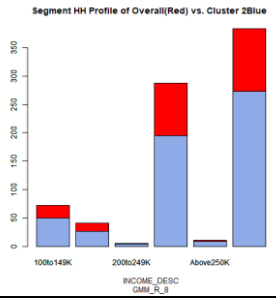
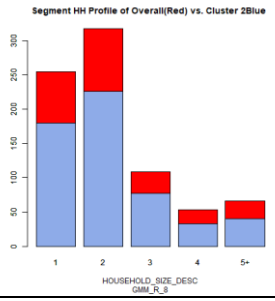
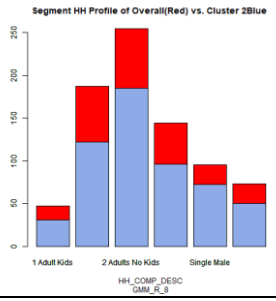
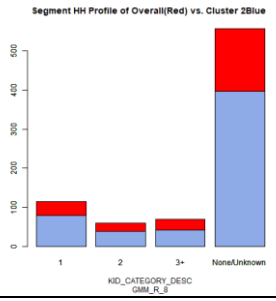
Figure 86: GMM-T Clustering Output Household Characteristics

GMM-R (Gaussian Mixture Model on the as-is form of 2 reduct inputs)

GMM-R Cluster 1 Consumer Traits: Cluster 1 shoppers are a good mix of frequent and infrequent shoppers. They are price conscious and have a propensity to use plenty of discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.
Tying Consumer Traits Back To Household Traits: Lower income bracket with an older base may explain the lesser spending.

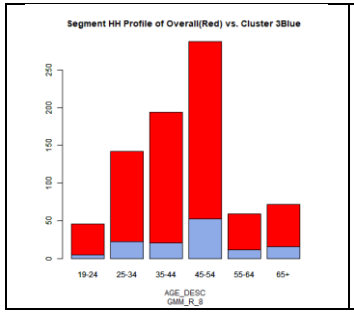
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

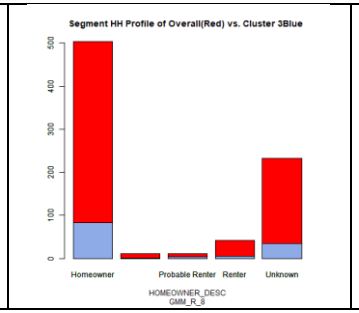
GMM-R Cluster 2 Consumer Traits: Cluster 2 shoppers are very active throughout the week. They use very few discounts and spend quite a sizable amount.
Tying Consumer Traits Back To Household Traits: Larger overall income perhaps explains few discounts used; a younger base may explain high frequency.

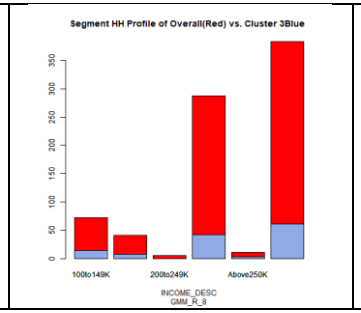
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

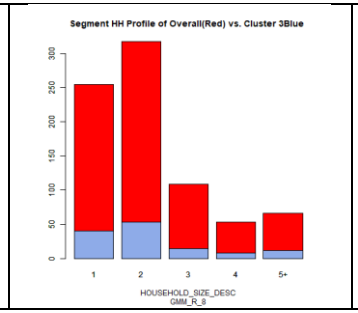
GMM-R Cluster 3 Consumer Traits: Cluster 3 is a much bigger group than Cluster 1. Nonetheless, they shop thru the week and spend not as much overall. No other noteworthy differences seen.
Tying Consumer Traits Back To Household Traits: An older base with lower overall income may explain the low overall spending.

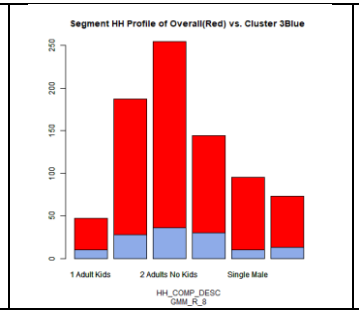
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category

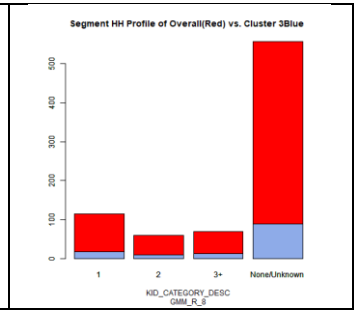




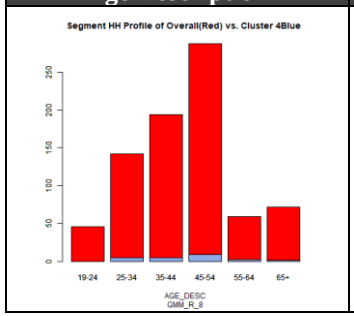
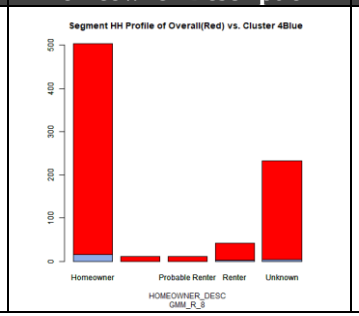
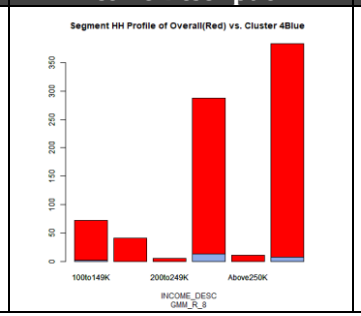
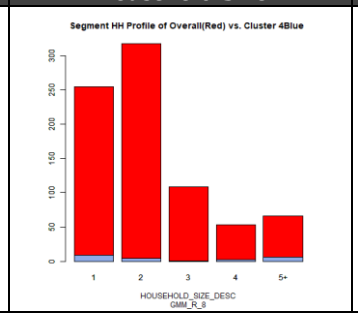
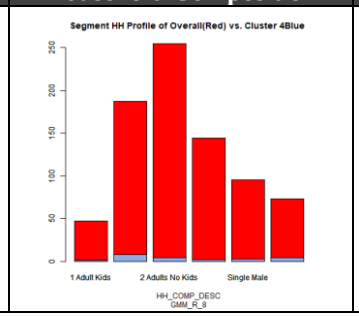
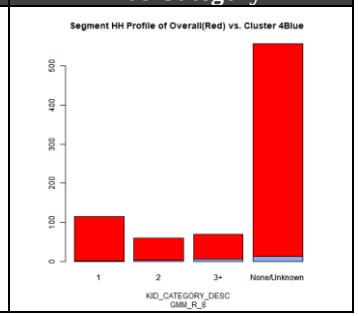




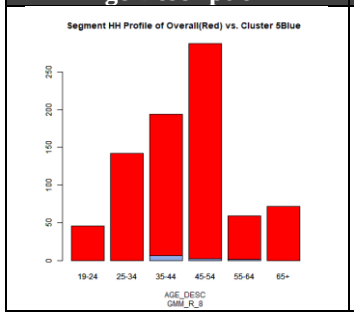
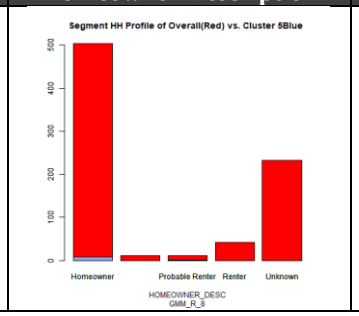
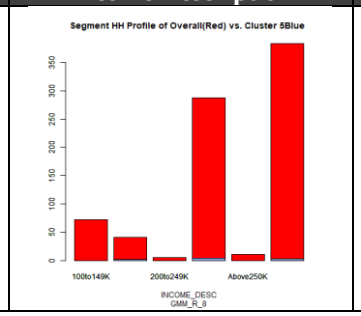
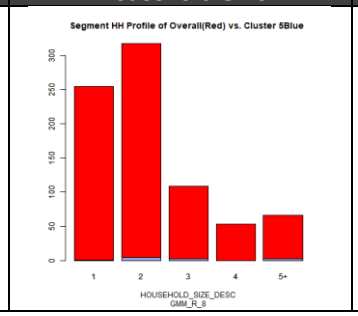
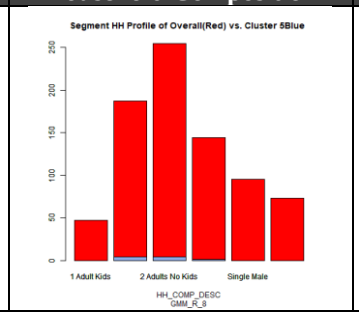
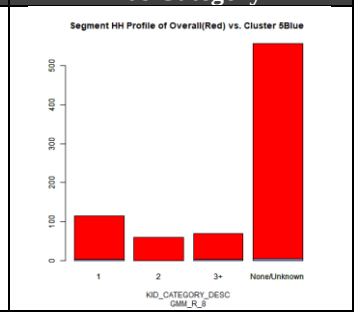




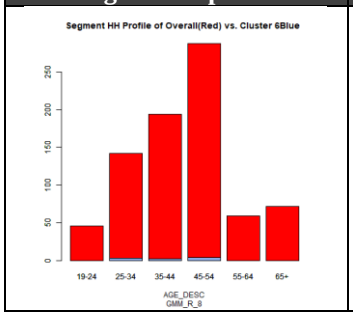
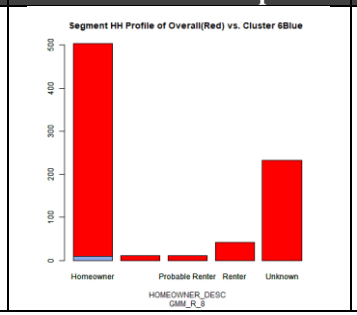
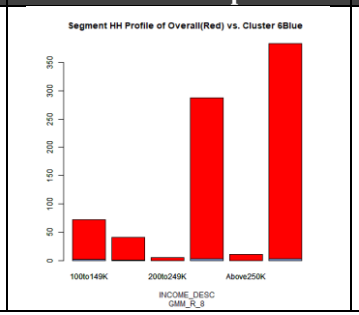
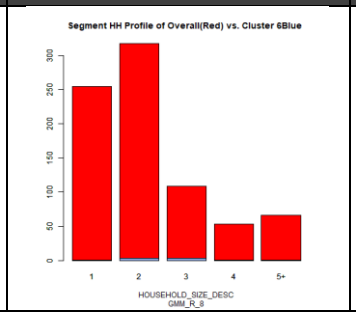
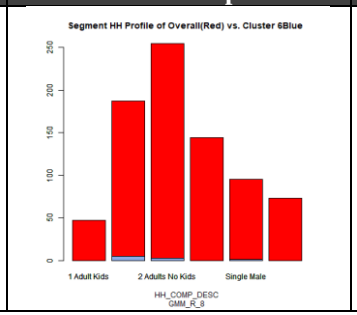
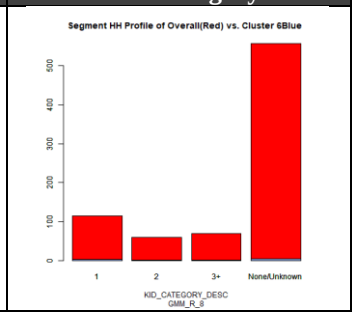
GMM-R Cluster 4 Consumer Traits: Cluster 4 shoppers are a much smaller group than Cluster 1. Very little traits can be drawn from the figure below.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

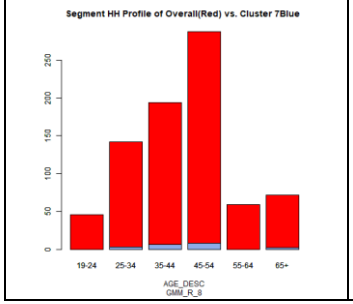
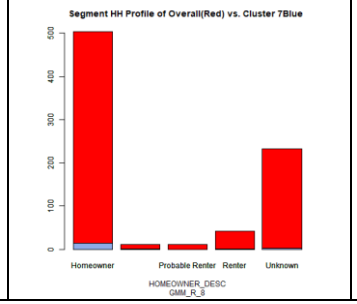
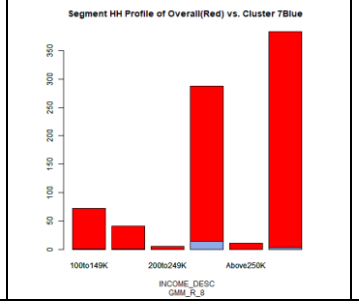
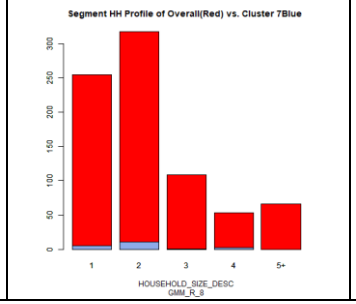
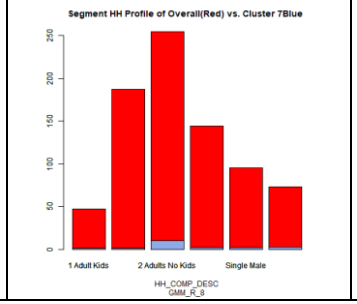
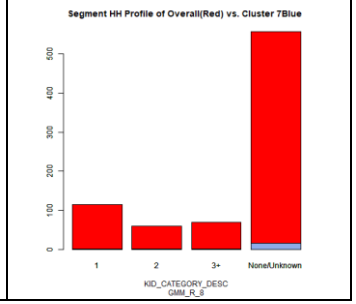
GMM-R Cluster 5 Consumer Traits: Though highly similar to Cluster 4, Cluster 5 shoppers shop slightly more frequently.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

GMM-R Cluster 6 Consumer Traits: Though highly similar to Cluster 4, Cluster 6 shoppers shop even less.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

GMM-R Cluster 7 Consumer Traits: Both clusters 6 and 7 are highly similar.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

GMM-R Cluster 8 Consumer Traits: Both clusters 7 and 8 are highly similar, as well, in terms of its low activity and spending.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category

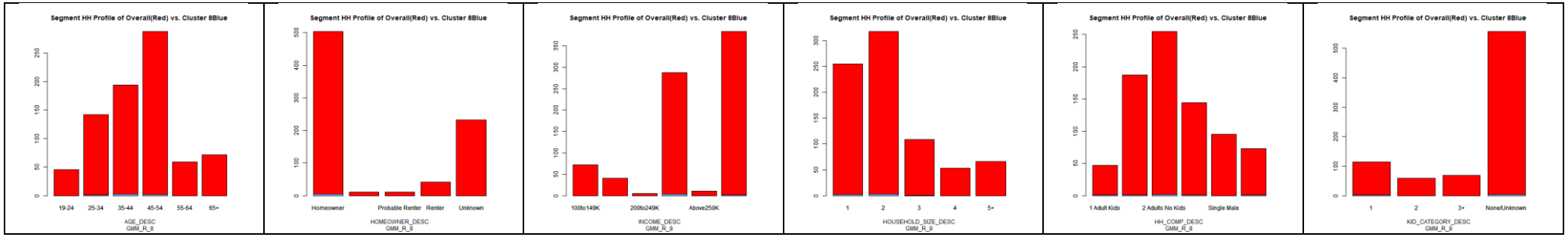
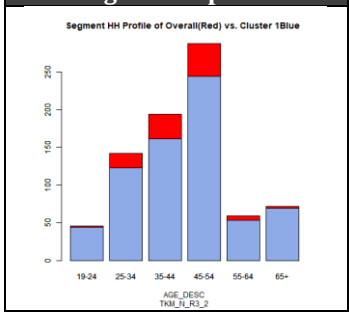
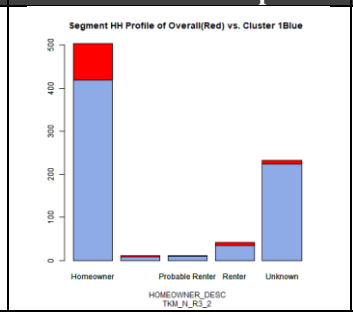
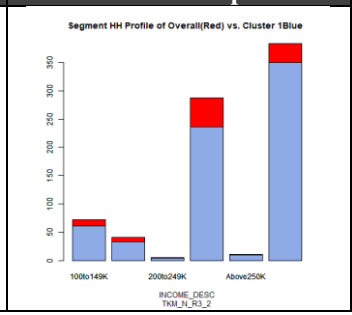
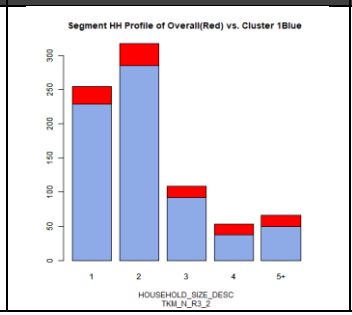
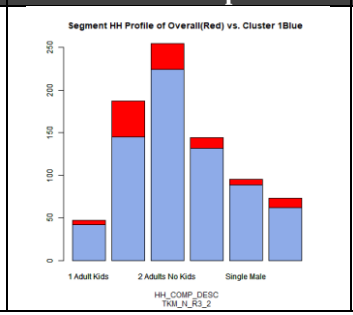
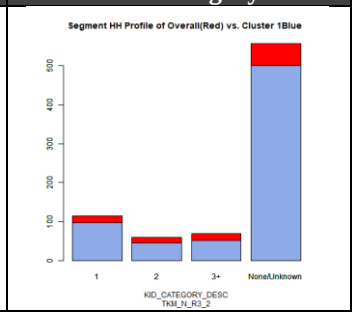


Figure 87: GMM-R Clustering Output Household Characteristics

6.7 3 Reduct Variables Clustering Output Household Characteristics

TKM-N-R (Traditional *k*-means clustering on normalized form of 3 reduct inputs)

TKM-N-R3 Cluster 1 Consumer Traits: Cluster 1 shoppers are highly frequent shoppers with deep pockets. They use some discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more per item.
Tying Consumer Traits Back To Household Traits: Higher overall income bracket with a younger base.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

TKM-N-R3 Cluster 2 Consumer Traits: Cluster 2 shoppers are highly infrequent shoppers with very little spending.
Tying Consumer Traits Back To Household Traits: Older base with lower income bracket.

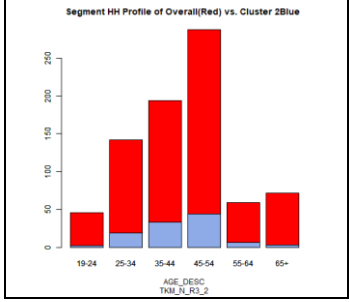
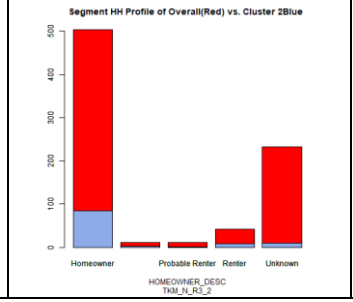
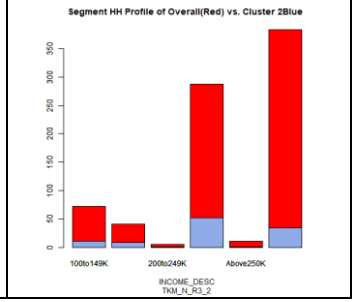
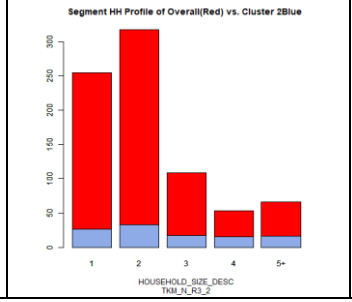
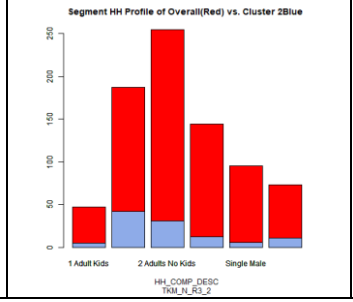
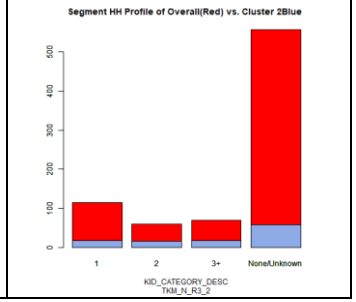
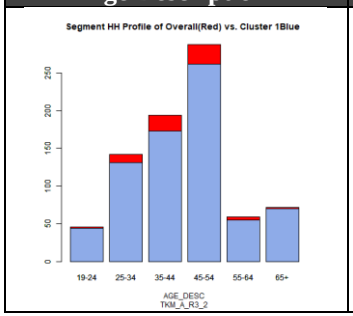
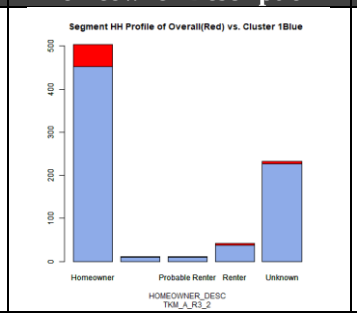
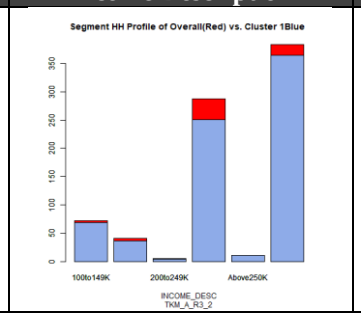
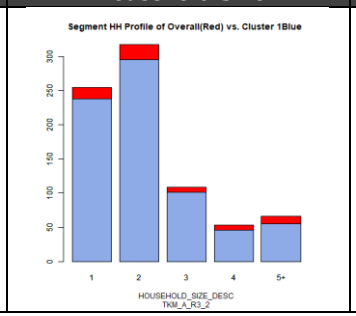
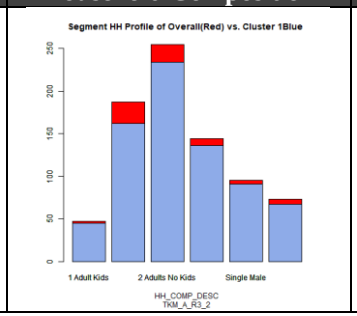
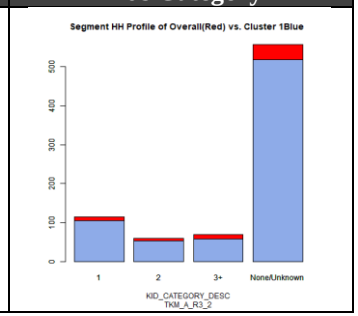
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

Figure 88: TKM-N-R3 Clustering Output Household Characteristics

TKM-A-R (Traditional *k*-means clustering on as-is form of 3 reduct inputs)

TKM-A-R3 Cluster 1 Consumer Traits: Cluster 2 shoppers are more infrequent shoppers with sizable spending. They prefer to shop in the afternoon, and to a lesser extent on weekends and evenings. They are not price sensitive, given the low discount usage and high sales value per quantity. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each item.
Tying Consumer Traits Back To Household Traits: Higher overall income bracket with a younger base.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

TKM-A-R3 Cluster 2 Consumer Traits: Cluster 2 shoppers are highly infrequent shoppers and are price conscious. They have a propensity not to use any discounts. They don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.
Tying Consumer Traits Back To Household Traits: Older base with lower income bracket.

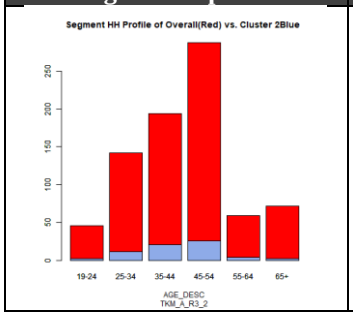
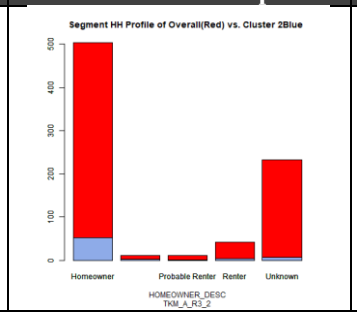
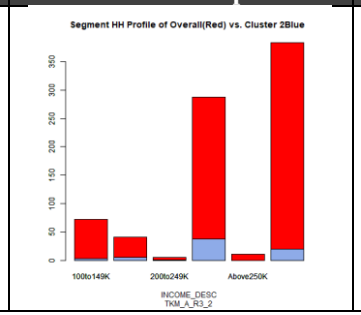
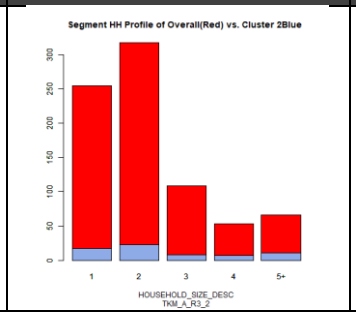
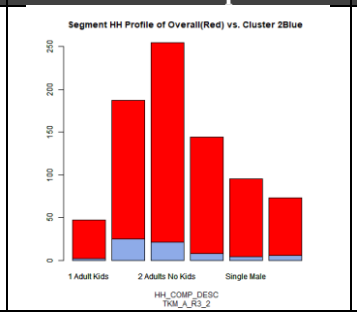
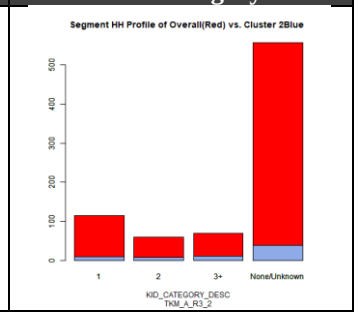
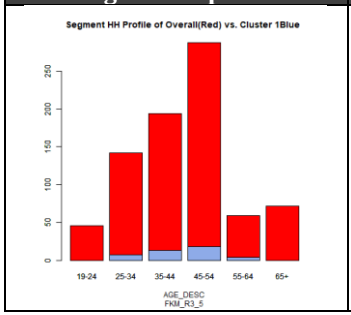
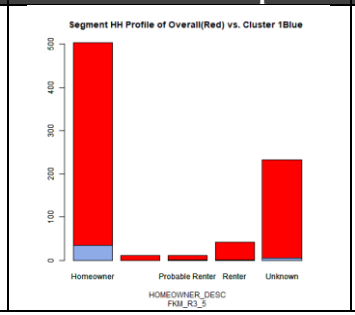
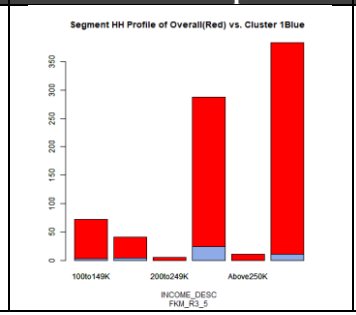
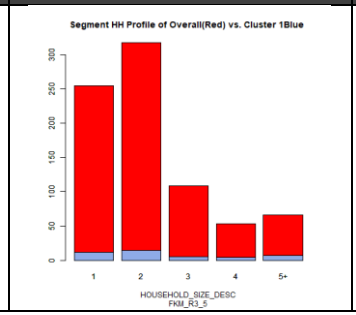
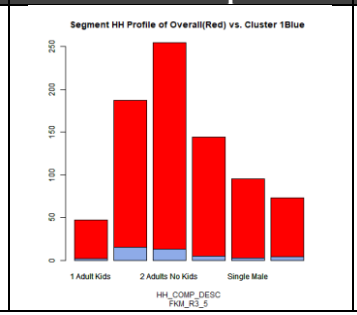
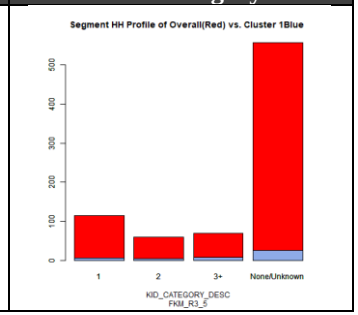
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

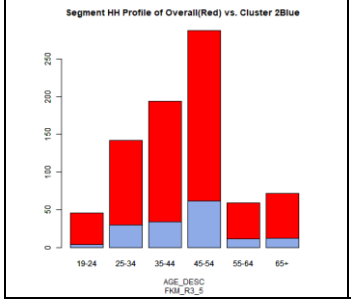
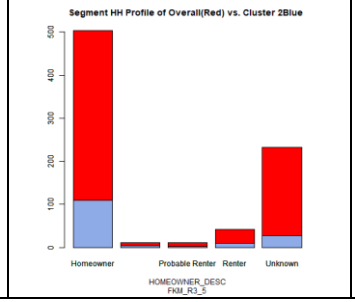
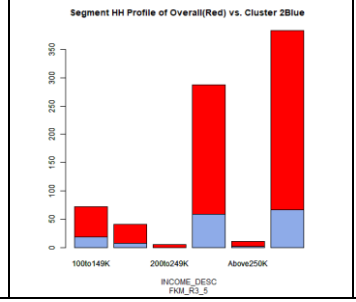
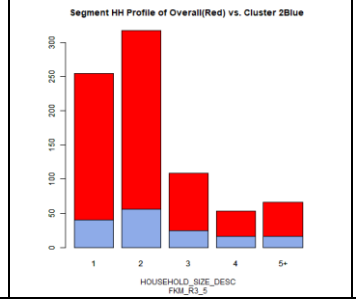
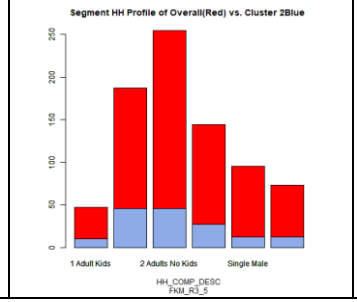
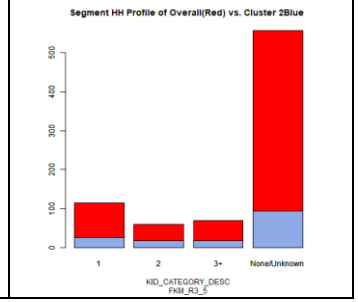
Figure 89: TKM-A-R3 Clustering Output Household Characteristics

FKM-R (Fuzzy *k*-means clustering on the as-is form of 3 reduct inputs)

FKM -R3 Cluster 1 Consumer Traits: Cluster 1 shoppers are a small group with highly infrequent shoppers and don't spend much. When they do shopping, it's usually in the afternoons.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

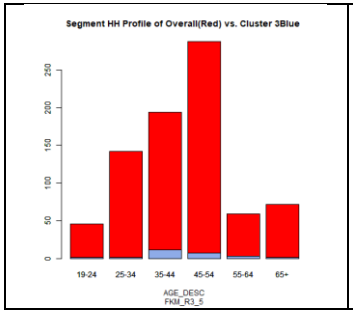
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

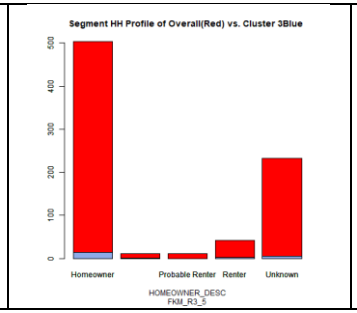
FKM -R3 Cluster 2 Consumer Traits: Cluster 2 shoppers are a good mix of frequent and infrequent shoppers. They use little to no discounts on a weekday, and they tend to spend less per quantity.
Tying Consumer Traits Back To Household Traits: Older base with lower income bracket.

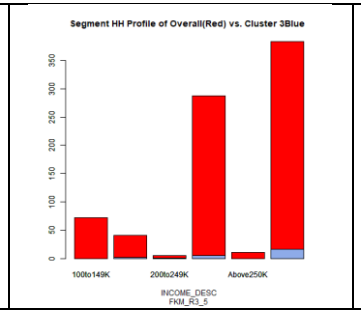
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

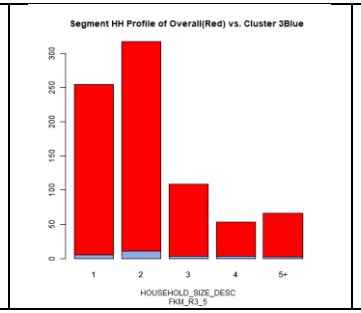
FKM -R3 Cluster 3 Consumer Traits: Cluster 3 shoppers are a small group, with a good mix of frequent and infrequent shoppers.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

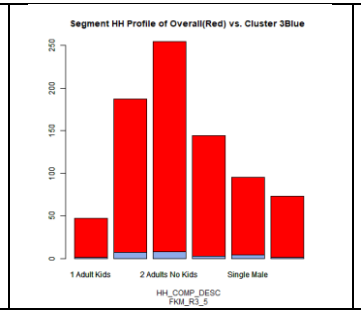
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category

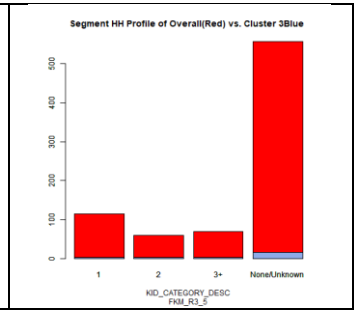




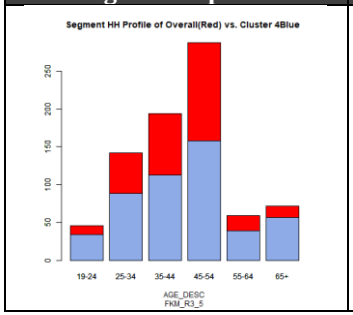
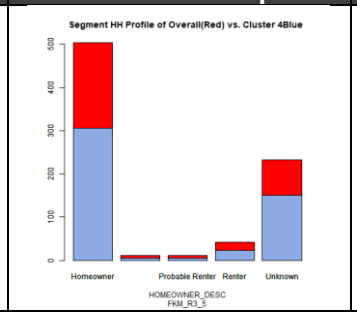
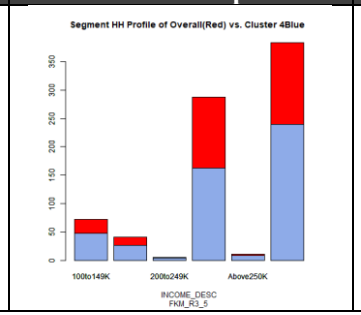
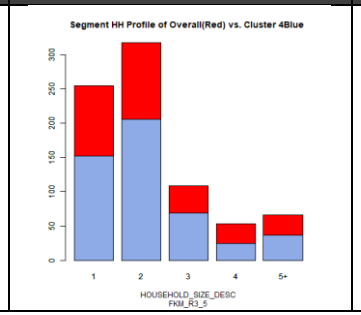
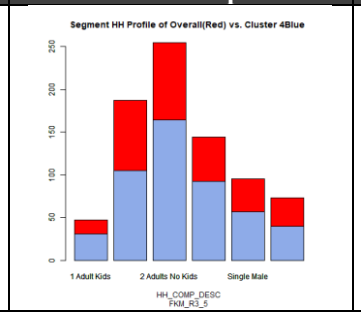
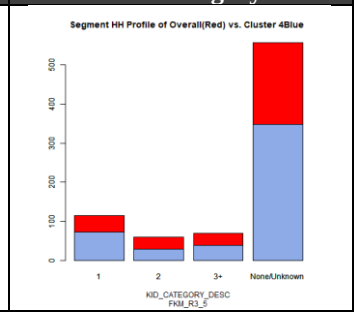








FKM -R3 Cluster 4 Consumer Traits: Cluster 4 shoppers are highly frequent shoppers with deep pockets. They use little to no discounts on a weekday. They spend more overall. They actively shop all week around. Even with their frequency, they spend more for each item.
Tying Consumer Traits Back To Household Traits: Higher overall income bracket with a younger base.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

FKM -R3 Cluster 5 Consumer Traits: Cluster 5 shoppers are a good mix of frequent and infrequent shoppers with deep pockets. They use little to no discounts on a weekday. They regularly spend a sizable amount each week. Even with their frequency, they spend more for each item.
Tying Consumer Traits Back To Household Traits: Lower income bracket should not be spending a sizable amount each week.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category

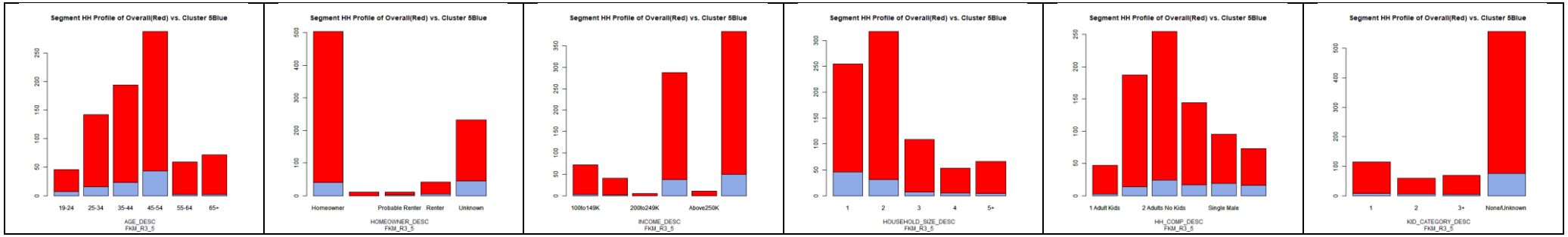
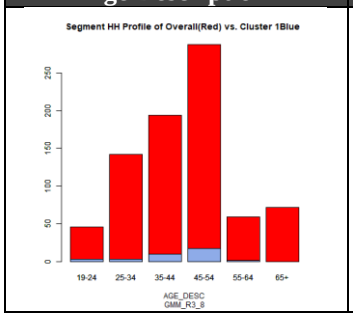
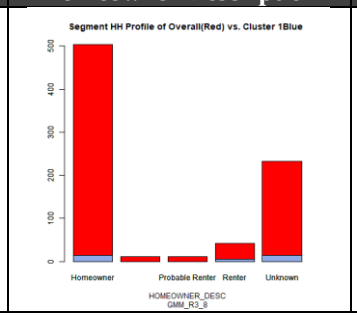
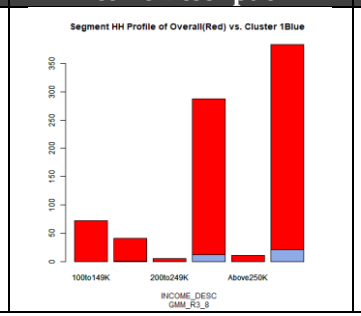
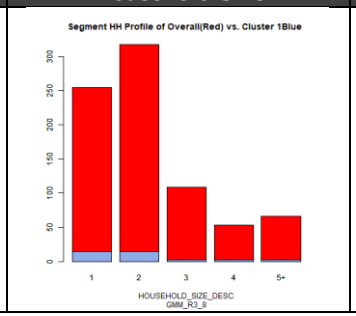
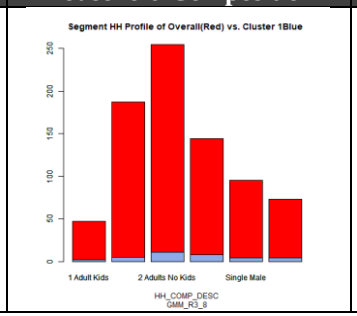
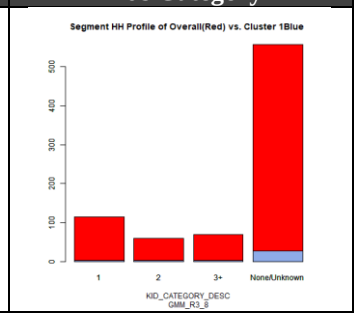


Figure 90: FKM-R3 Clustering Output Household Characteristics

GMM-R (Gaussian Mixture Model on the as-is form of 3 reduct inputs)

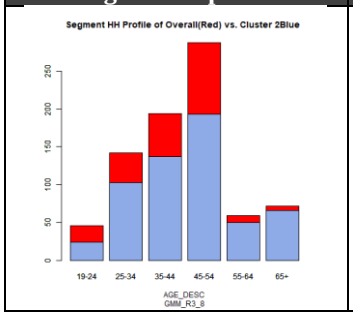
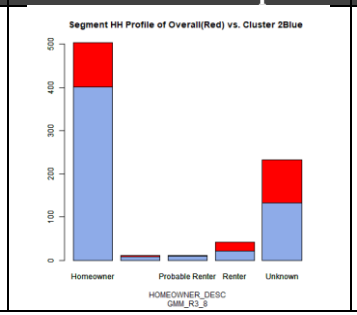
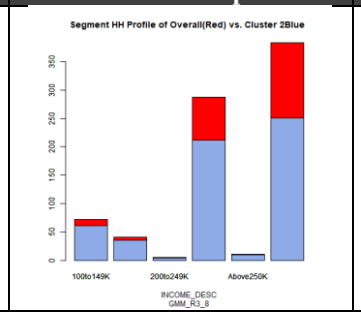
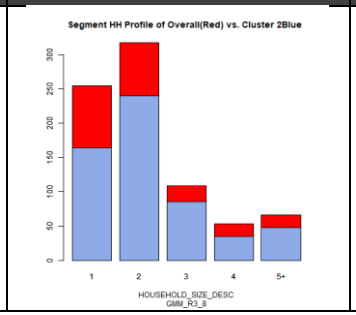
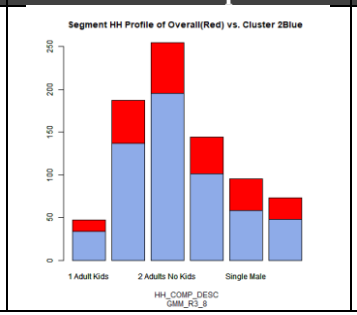
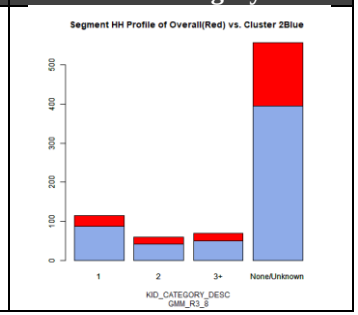
GMM-R3 Cluster 1 Consumer Traits: Cluster 1 shoppers are a small group of a mix of frequent and infrequent shoppers. They tend to prefer afternoon weekends. They do not use discounts at all, and they don't spend much overall. They hardly shop all week around. Given their frequency, they spend less on each item.

Tying Consumer Traits Back To Household Traits: Lower income bracket with an older base.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

GMM-R3 Cluster 2 Consumer Traits: Cluster 2 shoppers are very frequent shoppers, and active throughout the week. They spend a sizable amount and make full use of the discounts provided.

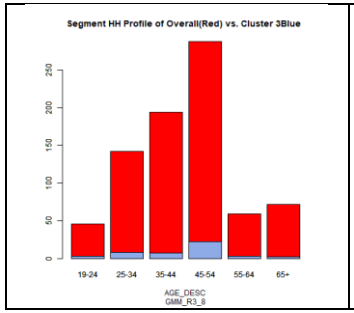
Tying Consumer Traits Back To Household Traits: A younger base with higher overall income.

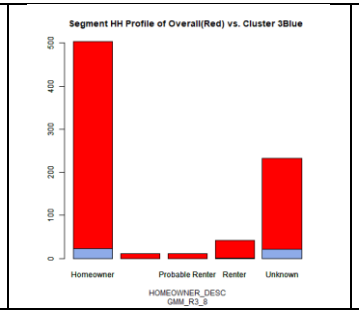
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

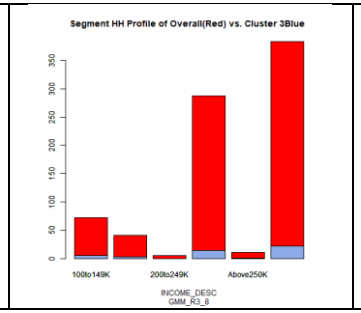
GMM-R3 Cluster 3 Consumer Traits: Like Cluster 1, Cluster 3 shoppers are somewhat infrequent shoppers. They hardly spend during the week.

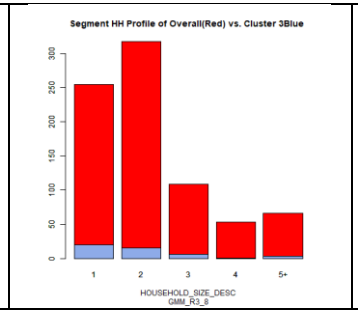
Tying Consumer Traits Back To Household Traits: Older singles with lower income base.

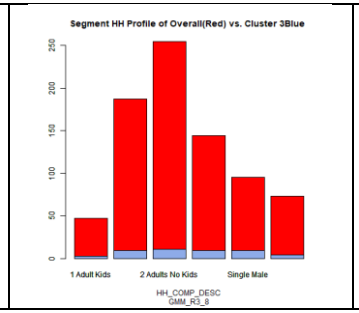
Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category

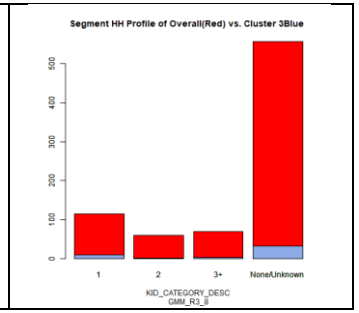




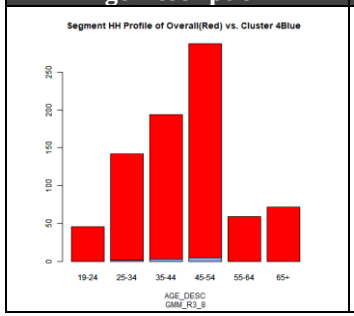
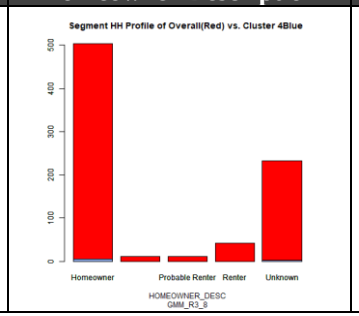
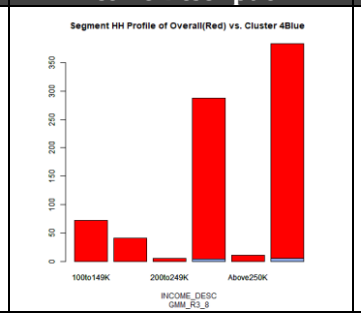
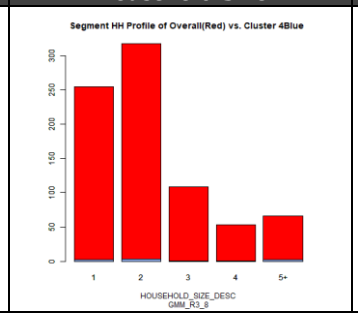
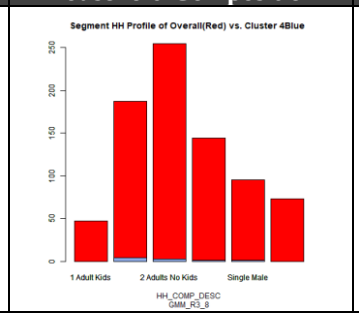
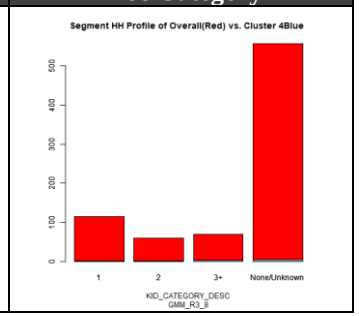




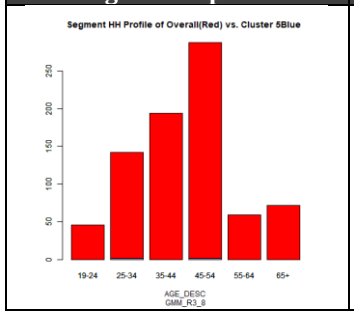
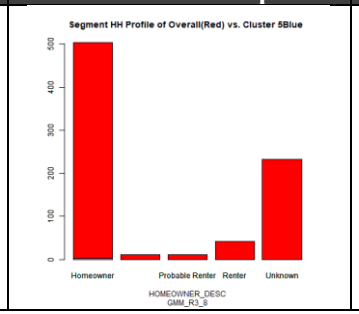
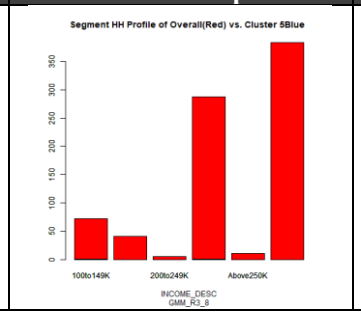
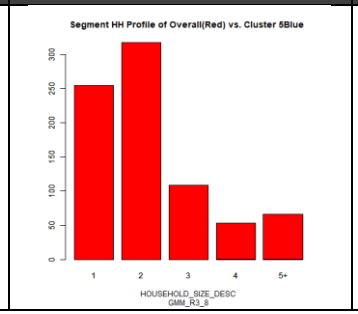
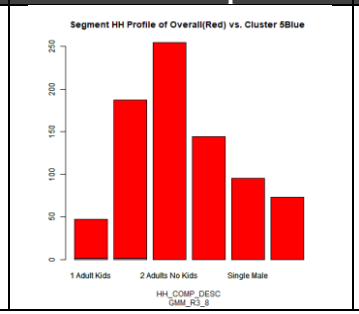
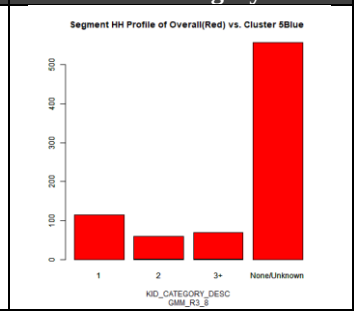




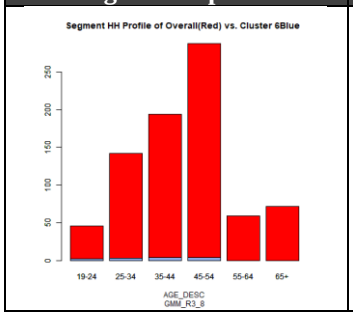
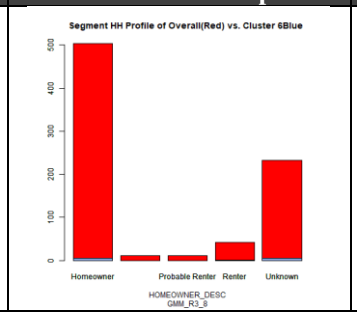
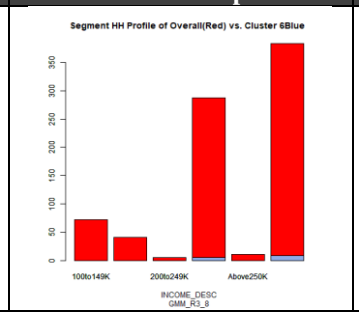
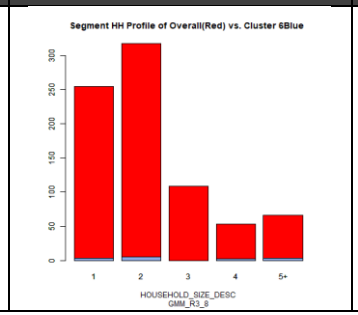
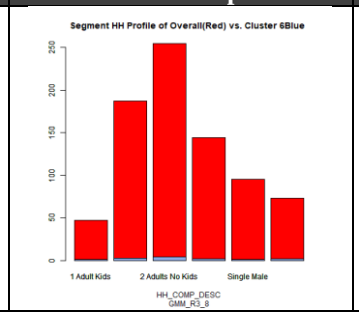
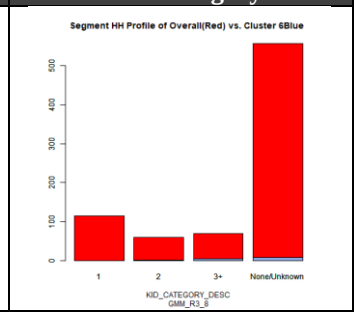
GMM-R3 Cluster 4 Consumer Traits: Cluster 4 shoppers are highly infrequent shoppers.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

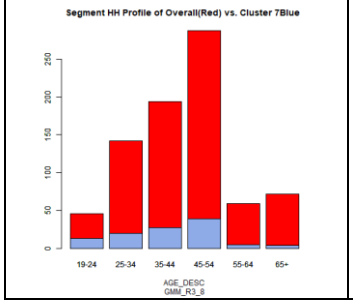
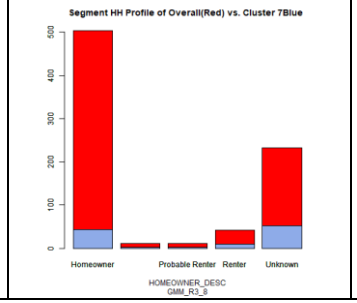
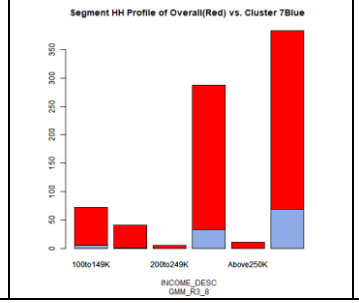
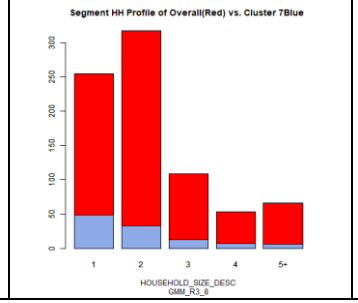
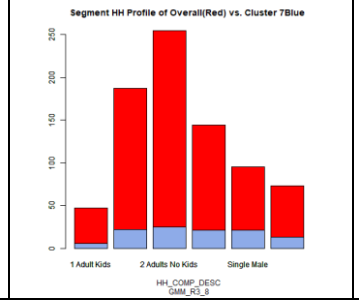
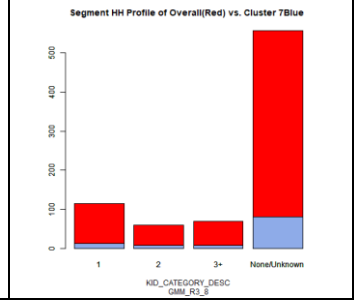
GMM-R3 Cluster 5 Consumer Traits: Cluster 5 shoppers are highly similar to Cluster 4, in that they are highly infrequent shoppers.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

GMM-R3 Cluster 6 Consumer Traits: Cluster 6 shoppers are very similar to Cluster 4 and 5, with highly infrequent shoppers.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

GMM-R3 Cluster 7 Consumer Traits: Cluster 7 shoppers are much more active during the week, and at various times of the day. Their overall spending is lower, and they control their spending each time.
Tying Consumer Traits Back To Household Traits: Younger base with lower income explain frequency and limited spending.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category
					

GMM-R3 Cluster 8 Consumer Traits: Cluster 8 shoppers are very similar to Clusters 4, 5 and 6, in that this group has very few members.
Tying Consumer Traits Back To Household Traits: Very little to tie it back, given small sample size.

Age Description	Homeowner Description	Income Description	Household Size	Household Composition	Kids Category

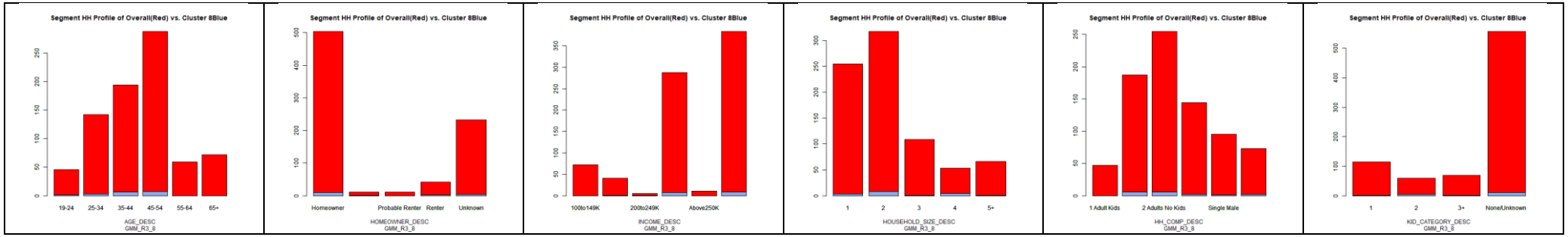


Figure 91: GMM-R3 Clustering Output Household Characteristics

From the above analysis and results, the following points can be noted. TKM and FKM outputs were somewhat similar. Their optimal cluster counts were between 2 or 3. Conversely, GMM had optimal cluster counts of 8 and 19 respectively. Outputs, relating to reduct datasets, had uneven household or object distribution across its clusters. TKM and FKM clusters displayed unique characteristics, while the reduct dataset of both these outputs showed the most distinctive characteristics despite its uneven distribution.

Based on the silhouette scores between TKM and FKM, FKM-R scored the highest. Exploring alternative cluster counts with similarly high silhouette scores yielded cluster counts of 7 and 9. Unfortunately, the distribution across both these cluster counts were equally uneven.

GMM-T's 8 cluster counts had overlapping characteristics, or its clusters were similar to one another. GMM-R's optimal cluster count of 19 were extremely uneven, where one of its clusters had only 7 objects. Using GMM-T's optimal cluster count of 8 onto GMM-R yielded sub-optimal clusters with similar characteristics.

Expanding the reduct dataset from 2 to 3 variables yielded similar silhouette or BIC scores. While TKM-N-R3 and TKM-A-R3 had the same optimal cluster count of 2 as before, FKM-R3 and GMM-R3 yield 5 and 8 cluster counts respectively. Uneven distributions remain throughout these four R3 outputs. While TKM-N-R3 and TKM-A-R3 showed unique cluster characteristics, helped by their low cluster counts, GMM-R3 and FKM-R3 had clusters with similar characteristics.

When superimposing the household variables onto the cluster outputs, household attributes in individual clusters largely match its transactional attributes, except in outputs with large cluster counts. Clusters using reduct dataset did not show a similar clear split on household characteristics. Thus, this wraps up our clustering outputs and comparisons with its household demographic information.

7.0 Discussion

The clustering work done during this capstone serves to underscore the importance of alternative clustering algorithms. Clusters generated by fuzzy *k*-means and GMM, on the back of a reduct dataset, have shown distinctive characteristics unique to each cluster. This can be said to be more distinctive than those outputs generated by the traditional *k*-means clustering. This might possibly be due to the nature of the dataset that makes it less ideal for traditional *k*-means clustering. Nonetheless, through the course of this capstone, several key points stood out from the use of these alternative clustering algorithms, namely reduct outcomes, processing speed and inaccuracy in higher cluster counts. Each of these will be contextualized and addressed below to serve as a reminder in terms of the tradeoffs between benefits and limitations, beginning first with the reduct methodology.

Outputs that have relied on the 2 reduct dataset have shown distinct individual cluster attributes. This is more distinctive than outputs from the as-is dataset. This is a peculiar find, considering the basis behind Rough Set's reduct. It is a feature reduction methodology that reduces a dataset down to its key attributes without losing the main essence of the original dataset. Hence a reduct dataset should work as well as an as-is dataset. On reflection, this study used multiple reduct iterations to obtain more variables for the main purpose of clustering. The efficiency of the reduct methodology in stripping away a whole dataset down to a single variable was both unexpected and not ideal for a clustering study. Clustering algorithms require a minimum of two or more variables to generate meaningful outcomes. A key learning here would be that the outcome of a Rough Set's reduct

methodology can neither be known beforehand nor controlled to select a set number of variables a priori. Hence it was highly fortunate on this capstone's part that two reduct iterations subsequently yielded highly distinctive clustering outputs.

Moving away from reduct and to processing speed, this capstone already had an expectation that the use of large cluster ranges (between 2 to 50) to find the optimal number of clusters would invariably take up time. This capstone was surprised to find that it took much longer, specifically for fuzzy *k*-means clustering. An attempt to find an optimal cluster count between this said range took almost a whole day using fuzzy *k*-means, and even then, it was not done processing. On reflection, this can be explained by its use of soft clustering ie. an object *i* has membership levels to all clusters and may belong to a handful or even all of them at the same time. It is this soft clustering approach, that though yields greater flexibility towards messy data, it is also very computationally expensive. Larger datasets would invariably take much longer. Add to that, the need to calculate the fuzzy silhouette score would make the overall processing time even longer. In the table below, fuzzy *k*-means processing ballooned to a mean of 38 seconds, compared to only 3 seconds without the fuzzy silhouette score. A caveat here would be the use of a lower R version of *fclust*. In the newer version of *fclust*, the fuzzy silhouette index is calculated together within the FKM function call. In the older version that was used by this capstone, the FKM function call lacked the in-built silhouette index and thus had to be called separately, resulting perhaps in a longer processing time. While reflections on the use of a lower R version would be covered in a later section, ultimately it is the decision of the analyst to balance the needs for greater clustering flexibility at the expense of (much) longer processing time.

Output	**Min**	**Lower Q**	**Mean**	**Median**	**Upper Q**	**Max**
TKMNT *(no silhouette)*	0.12	0.12	0.13	0.13	0.13	0.21
TKMNT *(with silhouette)*	0.87	0.89	0.94	0.94	0.97	1.05
TKMAT *(no silhouette)*	0.12	0.12	0.14	0.13	0.14	0.24
TKMAT *(with silhouette)*	0.87	0.89	0.94	0.96	0.98	1.07
TKMNR *(no silhouette)*	0.02	0.02	0.03	0.03	0.03	0.03
TKMNR *(with silhouette)*	0.14	0.15	0.22	0.25	0.25	0.27
TKMAR *(no silhouette)*	0.02	0.02	0.03	0.02	0.03	0.03
TKMAR *(with silhouette)*	0.15	0.16	0.22	0.25	0.25	0.27
FKMT *(no silhouette)*	2.41	2.43	2.50	2.49	2.54	2.70
FKMT *(with silhouette)*	37.45	37.72	38.05	37.96	38.29	39.27
FKMR *(no silhouette)*	3.13	3.18	3.21	3.21	3.22	3.37
FKMR *(with silhouette)*	37.13	37.47	37.82	37.73	38.20	38.68
GMMT	3.05	3.10	3.16	3.17	3.21	3.27
GMMR	2.43	2.48	2.58	2.57	2.66	2.78

Table 9. Execution time (secs) of R output with cluster k = 2 with 20 iterations (Using microbenchmark R package)

Lastly, in several of the optimal cluster outputs, $k = 2$ seemed to have yielded either the highest silhouette or fuzzy silhouette index. These 2 eventual clusters yielded distinctive qualities. This capstone had also explored higher cluster counts through the use of these indexes, such as exploring the cluster counts above the 0.85 fuzzy silhouette mark. It was hoped that these greater cluster counts, exhibiting high (but not the highest) fuzzy silhouette index would yield a more even spread across the various clusters. This capstone has found that greater cluster counts did not show any meaningful outputs. On reflection, the greater cluster counts should have yielded something meaningful. After all, these greater cluster counts were not too far away from $k = 2$ itself. This would perhaps be an addition to a future scope of work, where selected cluster counts along the silhouette index with high values would be tested to see how much its clustering output degrades as it goes further down the line.

8.0 Conclusions and Contributions

In conclusion, this capstone exercise has underscored the importance of adding three different methodologies ie. fuzzy *k*-means, GMM and reduct to a data analyst's ever burgeoning toolbox. While a traditional *k*-means clustering is an ever-trusted clustering method for a dataset with continuous variables, both fuzzy *k*-means and GMM provide worthy justification for usage as well. This capstone has demonstrated that using traditional *k*-means clustering on a simple retail behaviour dataset may not necessarily yield deep insights. By incorporating other clustering methods, a data analyst can compare the usefulness of each method and decide which would be most suitable for the job at hand.

Having said this, through the course of this study, this capstone had also identified future work opportunities. These opportunities focus on the Rough Set theory. One of them is on the Rough Set reduct method. The specific reduct method used in this capstone specifically focused on global discernibility. As such, a possible avenue would be to explore other discernibility methods such as local, quick reduct.rst, and quick reduct.frst.

Another future work opportunity would be to build on the work by Upadhyaya, Arora, & Jain (2008) and Singh & Mandal (2017). They applied *rough set*'s *reduct* algorithm on the attributes within each cluster, *post* clustering, to reduce each cluster's insignificant attributes and retain its significant attributes. Earlier, this capstone intentionally took the position to examine the effects of incorporating a reduct algorithm *pre*-clustering, rather than incorporate the reduct algorithm *post* clustering. Hence, the future work opportunity lies on exploring the reduct algorithm done at both the *pre* and *post* clustering stage. The belief here is that reduct at the *pre* clustering stage helps to improve the accuracy and uniqueness of the clustering output, whereas reduct at the *post* clustering stage helps to identify the significant variables that best explain each cluster. It will be an additional benefit, if this Rough Set comparison is done on a retail customer segmentation. This is one field where Rough Set rarely appears in, since it is more often used on more complex dataset.

9.0 Milestones and Reflections

This capstone project has fully fleshed out several key learning points that are not necessarily specific to the technicalities of the project. The first would be, one cannot underestimate the rigour

of a capstone project. While taking additional courses to fulfill MITB course requirements would have sufficed, a capstone project provides a holistic learning experience to a post-graduate. It provides an opportunity to do deep research into a highly technical field, traverse known (and possibly incomprehensible) academic work and look to apply it in a real-world setting. Of course, it would have been all the more daunting, had it not been for the decisive guidance by Prof Kam. It is to Prof Kam, that the author of this capstone's work extends his sincerest gratitude.

From early on, having an idea and a rough direction on where the research intended to head to, did not placate the mind on the task at hand. At the initial stage of the proposal, clarity on the capstone project was at its fuzziest (pun intended). After multiple rounds of consultation and idea exchanges, the process felt clearer at the progress report stage. Unfortunately, it did not give a sufficient fair warning on the final leg. As such, the best advice for future students keen on completing their own capstone projects would be to start early, read a multitude of journals to understand the work already done in that field and have regular conversations with your supervisor. While there is plenty of work that will need to be completed by the student themselves, they would need to trust the process. On reflection, there are indeed challenges at every stage from the need for clarity and direction, to gaining momentum and to finally pull through the finishing line through sheer grit.

While the SAS scholarship has kindly subsidised my postgraduate education, it has also given me an opportunity to explore the use of their statistical software. The use of their software would not have been possible within my work environment. Their software has been invaluable in helping to manipulate and analyse this capstone study. Little did I know beforehand that their software allows the extension of an open-source integration of R/Python within their own SAS environment. As such, some advice to future SAS scholars would be to take full advantage of SAS's latest software on offer for the purposes of their capstone and incorporate learnings from your other modules such as R and/or Python into your SAS capstone project.

A last note is specifically on the SAS software (EM and Viya inclusive) and its wonderful allowance for open-source integrations. This allows for the use of algorithms that are not otherwise available within the SAS environment, such as fuzzy *k*-means, GMM and Rough Set's reduct. To be able to be fully compatible with SAS Enterprise Miner 14.1 for this project, the R version would need to be between 2.13.0 to 3.2.5. Any other R versions outside of this range would not work on SAS EM 14.1 At the time of this report, the latest R version is 4.2.2. Having to downgrade would mean finding a compatible older version. Unfortunately, two shortfalls to an older package version would be reduced functionalities or uninstallable dependencies. Hence the need to incorporate a more updated R version on the SAS EM or other related software becomes crucial for sustained use. It is hoped that this capstone would shed important light on the need for continued and updated integration of R within SAS-related software.

10.0 References

Bezdek, J.C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York. http://dx.doi.org/10.1007/978-1-4757-0450-1

Campello, & Hruschka, E. R. (2006). A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems*, 157(21), 2858–2875. https://doi.org/10.1016/j.fss.2006.07.006

Chaudhuri, & Bhowmik, P. (1998). An approach of clustering data with noisy or imprecise feature measurement. *Pattern Recognition Letters*, 19(14), 1307–1317. https://doi.org/10.1016/S0167-8655(98)00112-3

Chelly Dagdia, Zarges, C., Beck, G., & Lebbah, M. (2020). A scalable and effective rough set theory-based approach for big data pre-processing. Knowledge and Information Systems, 62(8), 3321–3386. https://doi.org/10.1007/s10115-020-01467-y

Cormack. (1971). A Review of Classification. *Journal of the Royal Statistical Society*. Series A. General, 134(3), 321–367. https://doi.org/10.2307/2344237

Dunnhumby (2014). The Complete Journey. [Data file]. Retrieved from one of the data sets used in Singapore Management University's Data Analytics Lab module.

Fraley, & Raftery, A. E. (2002). Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association*, 97(458), 611–631. https://doi.org/10.1198/016214502760047131

Garla, S., Chakraborty, G., & Gaeth, G. (2012). Comparison of K-means, normal mixtures, and probabilistic-D clustering for b2b segmentation using customers' perceptions. In *SAS Global Forum 2012-Data Mining and Text Analytics*.

Melchior, & Goulding, A.. (2018). Filling the gaps: Gaussian mixture models from noisy, truncated, or incomplete samples. *Astronomy and Computing*, 25, 183–194. https://doi.org/10.1016/j.ascom.2018.09.013

Nayak, R. K., Mishra, D., Shaw, K., & Mishra, S. (2012). *Rough set-based attribute clustering for sample classification of Gene Expression Data*. Procedia Engineering, 38, 1788–1792. https://doi.org/10.1016/j.proeng.2012.06.219

Patel, E., & Kushwaha, D. S. (2020). Clustering cloud workloads: K-means vs gaussian mixture model. *Procedia computer science*, 171, 158-167. https://doi.org/10.1016/j.procs.2020.04.017

Pawlak. (1982). Rough sets. *International Journal of Computer & Information Sciences*, 11(5), 341–356. https://doi.org/10.1007/BF01001956

Pięta, & Szmuc, T. (2021). Applications of rough sets in big data analysis: An overview. *International Journal of Applied Mathematics and Computer Science*, 31(4), 659–683. https://doi.org/10.34768/amcs-2021-0046

Raykov, Boukouvalas, A., Baig, F., & Little, M. A. (2016). What to Do When K-Means Clustering Fails: A Simple yet Principled Alternative Algorithm. PloS One, 11(9), e0162259–e0162259. https://doi.org/10.1371/journal.pone.0162259

Riza, Janusz, A., Bergmeir, C., Cornelis, C., Herrera, F., Śle¸zak, D., & Benítez, J. M. (2014). Implementing algorithms of rough set theory and fuzzy rough set theory in the R package "RoughSets." *Information Sciences*, 287, 68–89. https://doi.org/10.1016/j.ins.2014.07.029

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53-65. https://doi.org/10.1016%2F0377-0427%2887%2990125-7

Schwarz. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2), 461–464. https://doi.org/10.1214/aos/1176344136

Scrucca, Fop, M., Murphy, T. B., & Raftery, A. E. (2016). Mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. The R Journal, 8(1), 289–317. https://doi.org/10.32614/rj-2016-021

Singh, G. K., & Mandal, S. (2017). Cluster analysis using rough set theory. *Journal of Informatics and Mathematical Sciences*, 9(3), 509-520.

Steinley. (2006). K-means clustering: A half-century synthesis. *British Journal of Mathematical & Statistical Psychology*, 59(1), 1–34. https://doi.org/10.1348/000711005X48266

Upadhyaya, S., Arora, A., & Jain, R. (2008). Deriving Cluster Knowledge Using Rough Set Theory. *Journal of Theoretical and Applied Information Technology*. 2008; 4:8: 688-696

Zadeh, L.A. (1965); Fuzzy Sets, *Informations and Control*, 8: 338-353. http://dx.doi.org/10.1016/S0019-9958(65)90241-X

Zhang, Q., Xie, Q., & Wang, G, (2016); *A survey on rough set theory and its applications*, 1(4), 323-333. https://doi.org/10.1016/j.trit.2016.11.001

11.01 Appendix 1: Clustering Comparisons

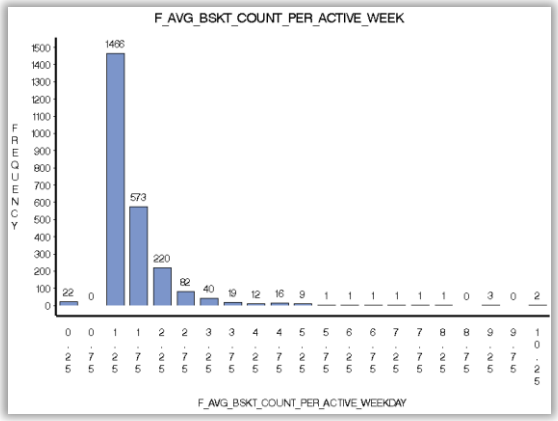
Clustering Algorithms	***k*-MEANS**	**FUZZY *k*-MEANS**	**GAUSSIAN MIXTURE MODEL (GMM)**
Founder, Year	MacQueen, 1967	Dunn, 1973 and Bezdek, 1981	Duda and Hart, 1973
Methodology	Objects are partitioned into k clusters, where each object belongs to the cluster with the nearest mean distance	Similar to *k*-MEANS. Only that objects have a membership level to all clusters	Assumes that all the objects within a dataset are part of a single or multiple normal distributions. Each distribution identified is considered as a cluster.
Approach	Unsupervised	Unsupervised	Unsupervised
Clustering Type	Hard clustering only	Soft and hard clustering	Soft and hard clustering
Clustering Speed	Fast	Somewhat fast	Somewhat fast
Solution Type	Tendency towards local solution	Less tendency towards local solution	Tendency towards local solution
Optimisation Type	NP-Hard	NP-Hard	NP-Hard
Strengths	• Simple and fast • Computationally efficient in dealing with large datasets	• Soft and hard clustering flexibility	• Soft and hard clustering flexibility • Allows for more flexible cluster shapes
Limitations	• Number of *k* clusters need to be known *a priori* • Fundamental assumptions of sphericity, equal variance (sensitive to outliers) • Hard clustering (each object can only be assigned to one set) • Tendency towards local optimum. Difficulty to obtain global optimum unless through brute force • Tends to produce clusters of similar sizes	• Distance to centroid defines membership. Unable to find some 'natural' clusters	• Sensitive to outliers • Datasets may not fit a normal distribution • Instability due to components not being properly separated • Tendency towards local optimum
Method to Obtain Optimal Cluster Count	• Elbow Curve Method • Silhouette Index	• Fuzzy Silhouette Index	• Akaike Information Criterion (AIC) • Bayesian Information Criterion (BIC)

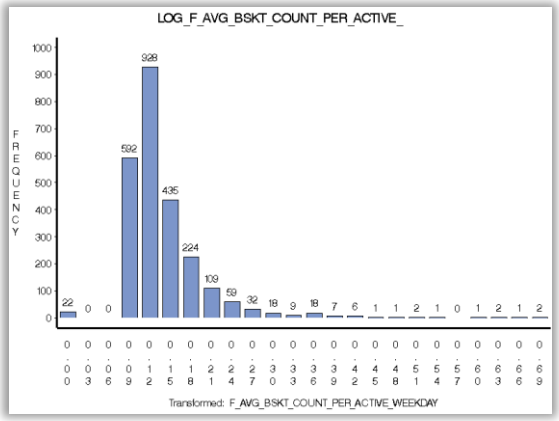
11.02 Appendix 2: Summary of Clustering Methodology

Pre-Clustering				Clustering					
	Dataset Modified?	*Eventual Dataset Variable Count*	*Normalised?*	*Clustering Algorithm*	*Chosen Method to Determine Optimal Clusters*	*Initial Cluster Range*	*Optimal Cluster Count*	*Mean Time Taken (s) to Run 2 Clusters Over 20 Iterations*	*Clustering Output*
1.	Removed Highly Collinear Variables	16 Inputs + 1 ID	Yes	*k*-means	Silhouette Index	2:50	2	0.13 (Without silhouette calculation), 0.94 (With silhouette calculation)	TKM-N-T
2.	Rough Set Reduct	2 Inputs + 1 ID	Yes	*k*-means	Silhouette Index	2:50	2	0.03 (Without silhouette calculation), 0.22 (With silhouette calculation)	TKM-N-R
3.	Removed Highly Collinear Variables	16 Inputs + 1 ID	-	*k*-means	Silhouette Index	2:50	3	0.14 (Without silhouette calculation), 0.94 (With silhouette calculation)	TKM-A-T
4.	Rough Set Reduct	2 Inputs + 1 ID	-	*k*-means	Silhouette Index	2:50	2	0.03 (Without silhouette calculation), 0.22 (With silhouette calculation)	TKM-A-R
5.	-	28 Inputs + 1 ID	-	fuzzy *k*-means	Fuzzy Silhouette	2:20	2	2.50 (Without fuzzy silhouette calculation), 38.05 (With fuzzy silhouette calculation)	FKM-T
6.	Rough Set Reduct	2 Inputs + 1 ID	-	fuzzy *k*-means	Fuzzy Silhouette	2:20	2	3.21 (Without fuzzy silhouette calculation), 37.82 (With fuzzy silhouette calculation)	FKM-R
7.	-	28 Inputs + 1 ID	-	Gaussian Mixture Model	Bayesian Information Criterion (BIC)	2:20	8	3.16	GMM-T
8.	Rough Set Reduct	2 Inputs + 1 ID	-	Gaussian Mixture Model	Bayesian Information Criterion (BIC)	2:20	19	2.58	GMM-R

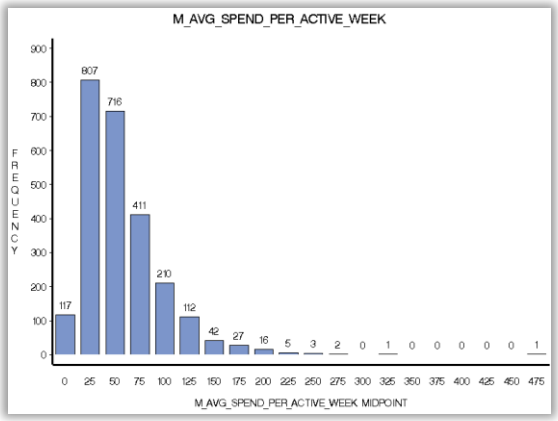
**All eight outputs were scaled and generated from EM's Open-Source Integration Node.*

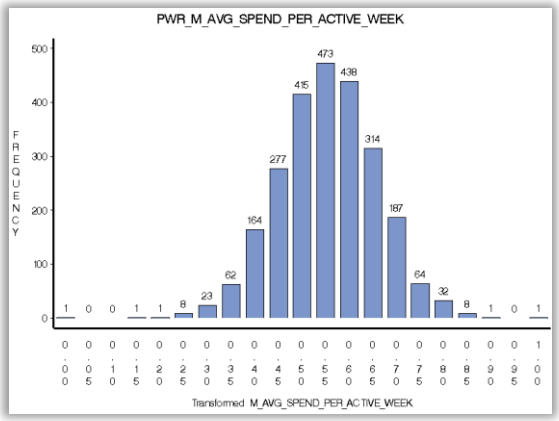
11.03 Appendix 3: Remaining TVM Variables



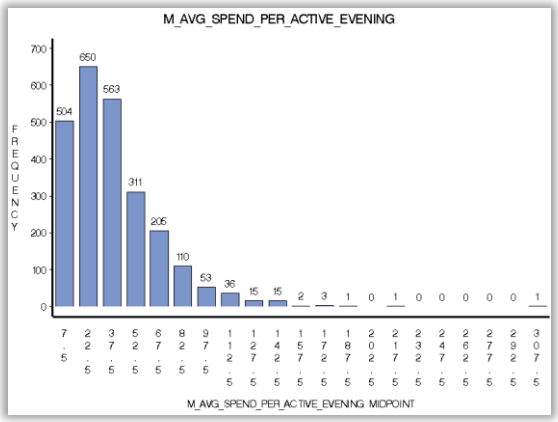


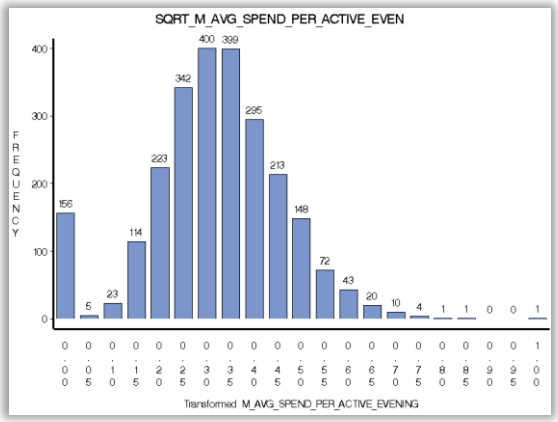
Comparison of F_AVG_BSKT_COUNT_PER_ACTIVE_WEEKDAY from Original to Transformed (log)



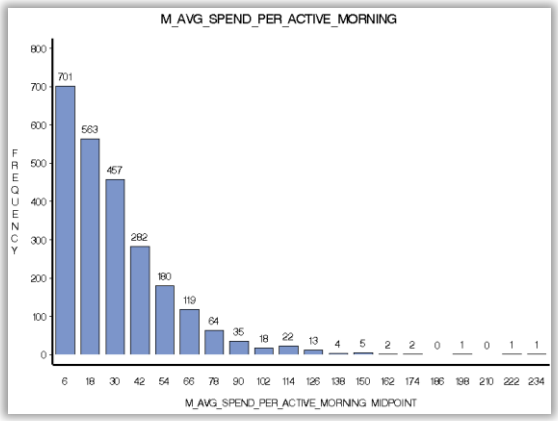


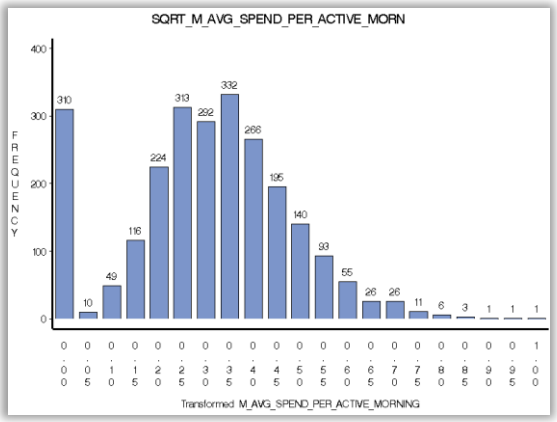
Comparison of M_AVG_SPEND_PER_ACTIVE_WEEK from Original to Transformed (Power)



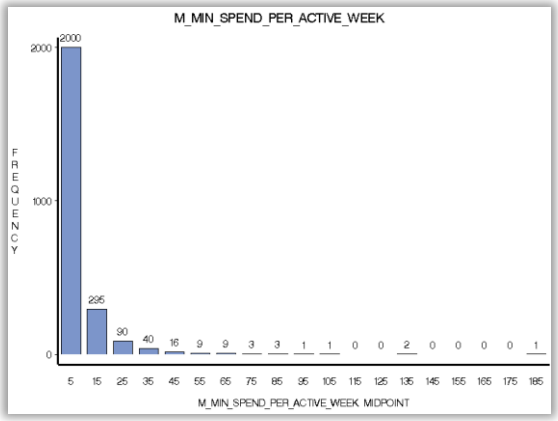


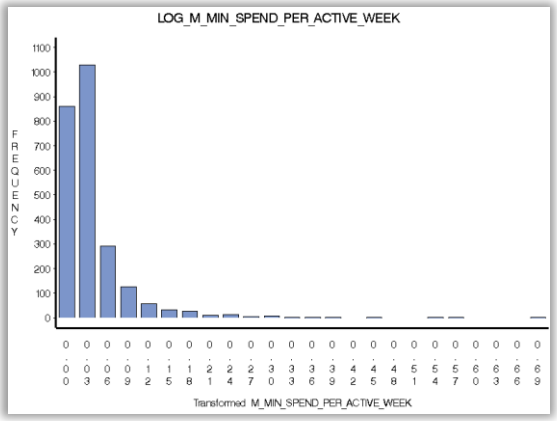
Comparison of M_AVG_SPEND_PER_ACTIVE_EVENING from Original to Transformed (Square Root)



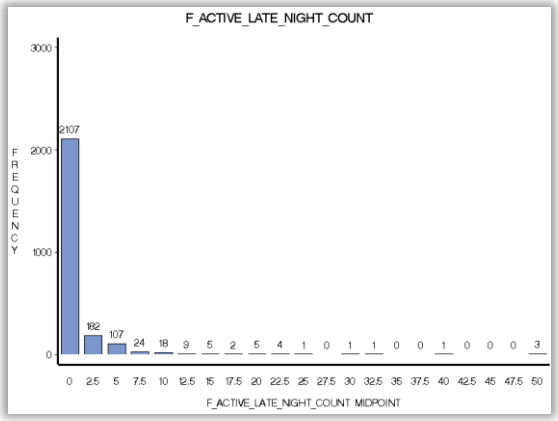


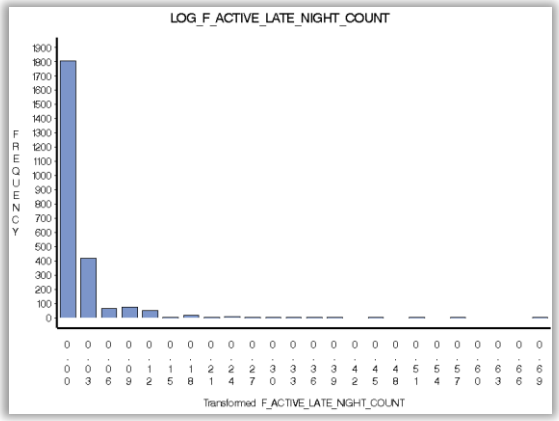
Comparison of M_AVG_SPEND_PER_ACTIVE_MORNING
from Original to Transformed (Square Root)



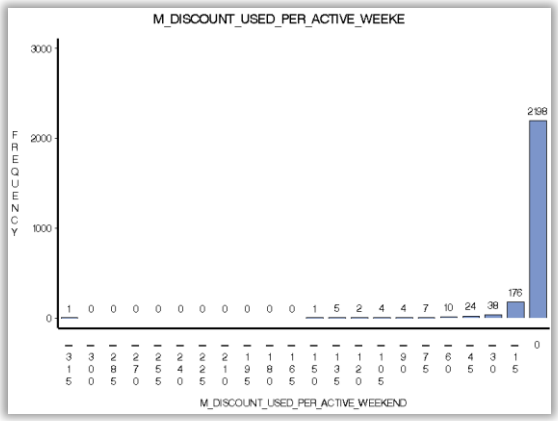


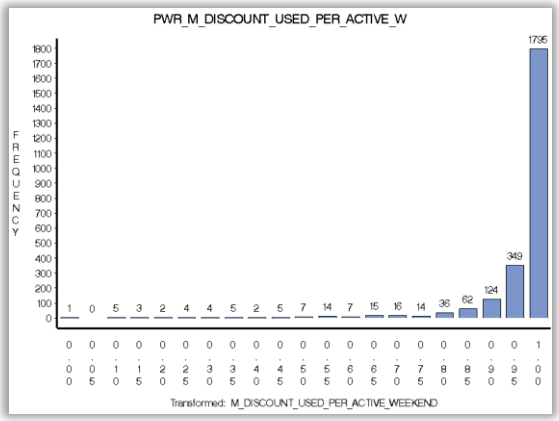
Comparison of M_MIN_SPEND_PER_ACTIVE_WEEK
from Original to Transformed (log)



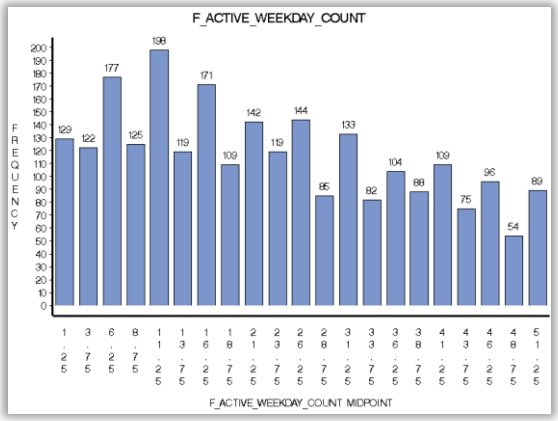


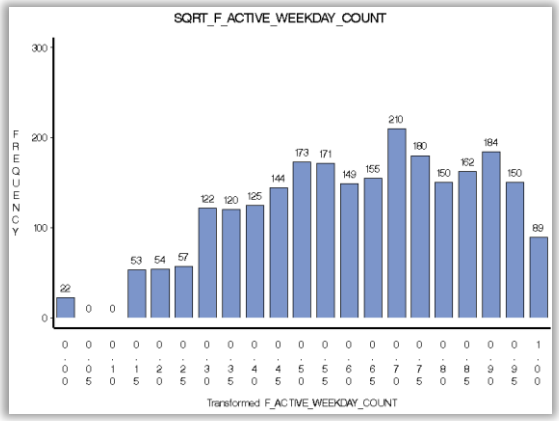
Comparison of F_ACTIVE_LATE_NIGHT_COUNT
from Original to Transformed (log)



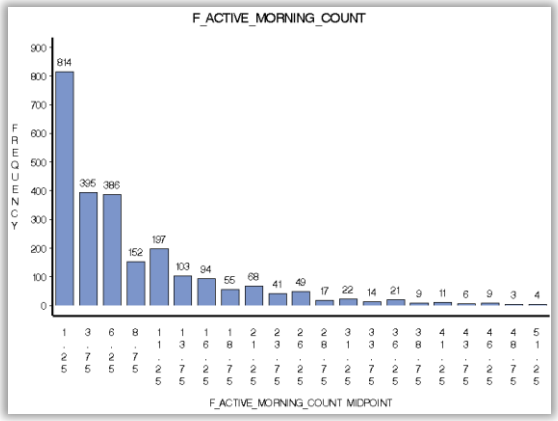


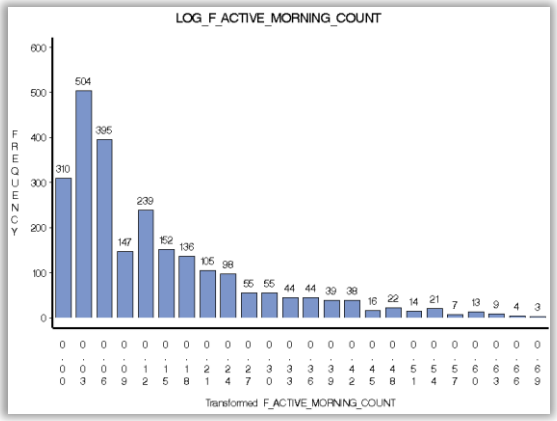
Comparison of M_DISCOUNT_USED_PER_ACTIVE_WEEKEND from Original to Transformed (Power)



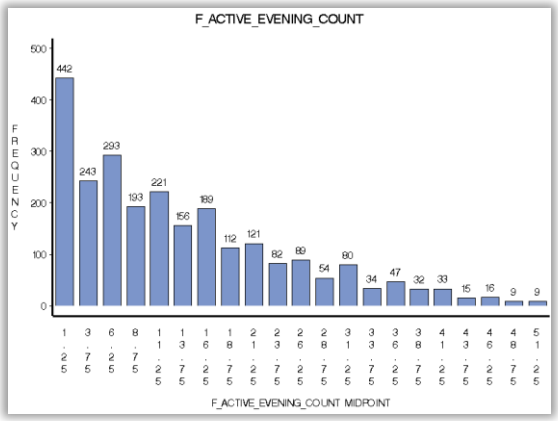


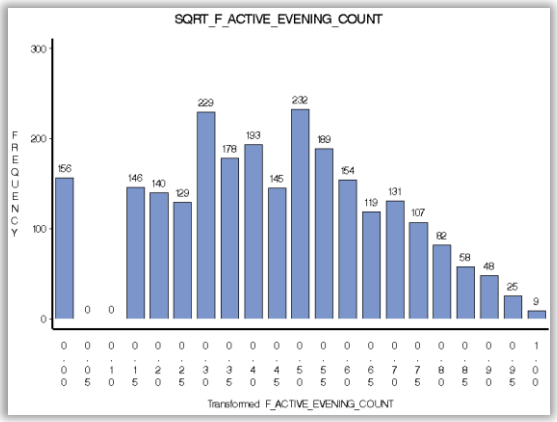
Comparison of F_ACTIVE_WEEKDAY_COUNT from Original to Transformed (Square Root)



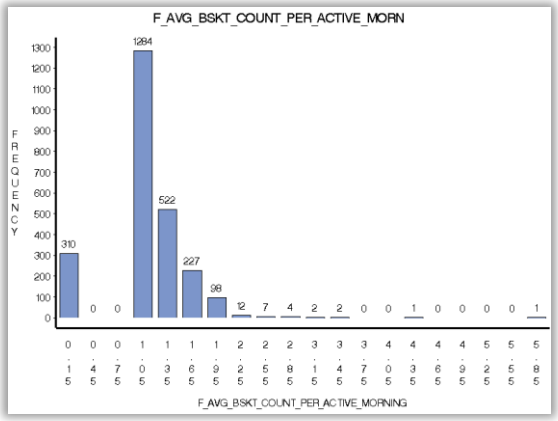


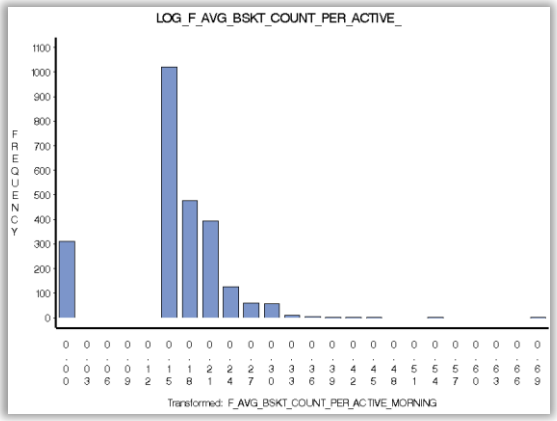
Comparison of F_ACTIVE_MORNING_COUNT from Original to Transformed (log)



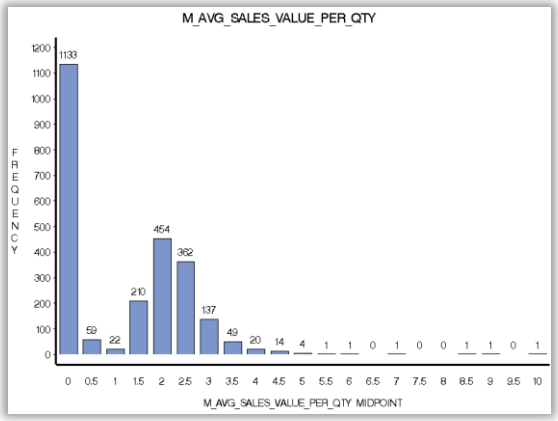


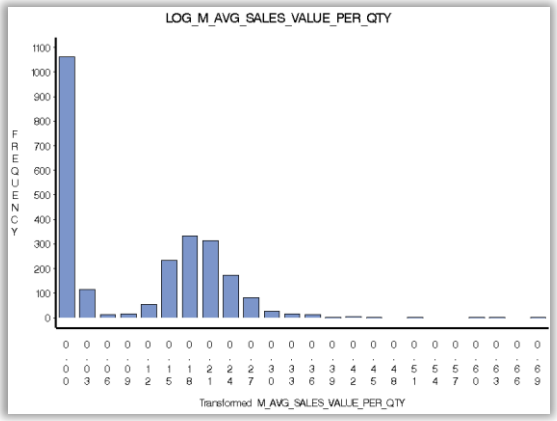
Comparison of F_ACTIVE_EVENING_COUNT
from Original to Transformed (Square Root)



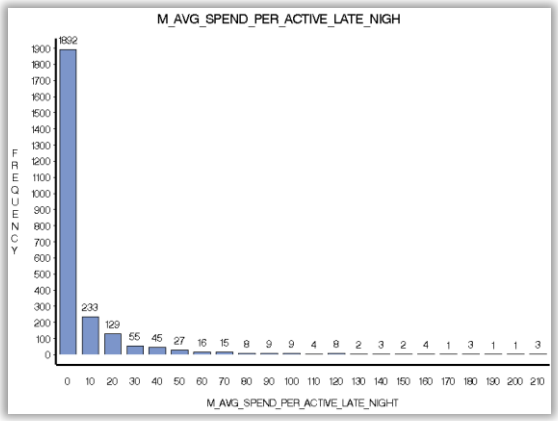


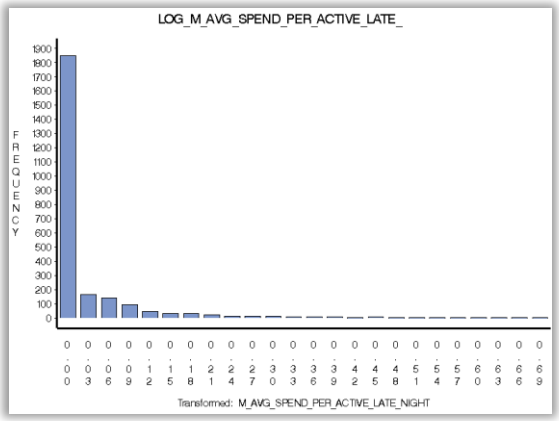
Comparison of F_AVG_BSKT_COUNT_PER_ACTIVE_MORNING
from Original to Transformed (log)



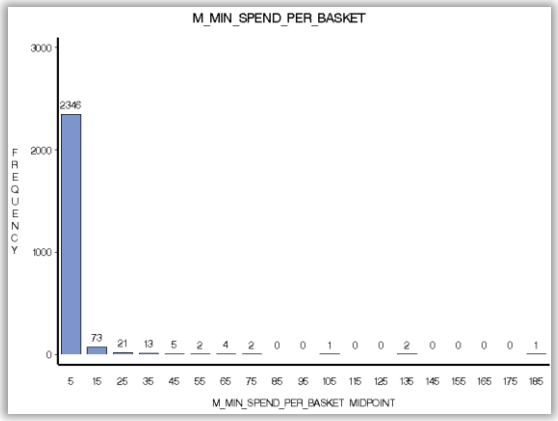


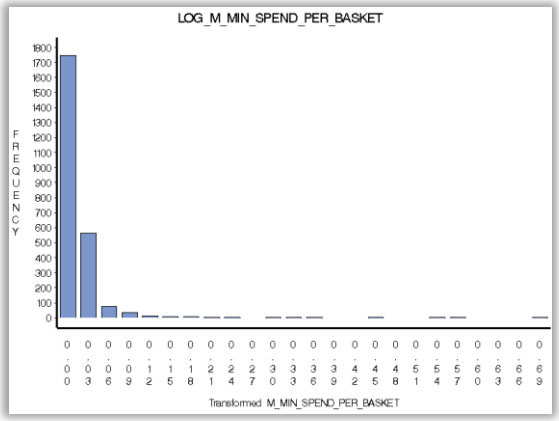
Comparison of M_AVG_SALES_VALUE_PER_QTY
from Original to Transformed (log)



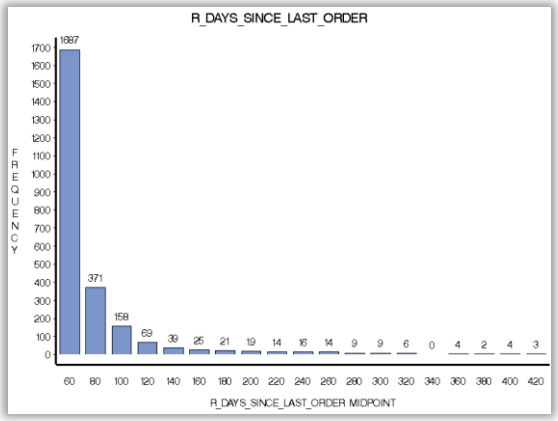


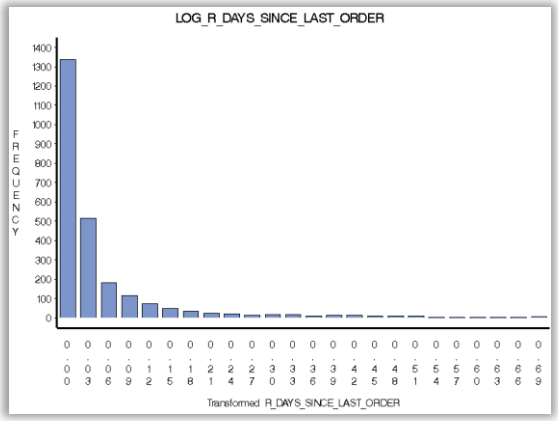
Comparison of M_AVG_SPEND_PER_ACTIVE_LATE_NIGHT
from Original to Transformed (log)



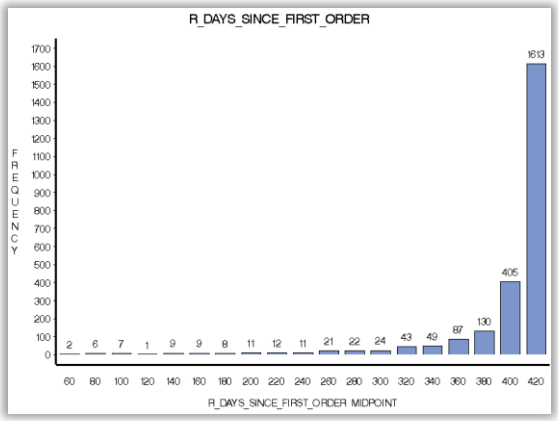


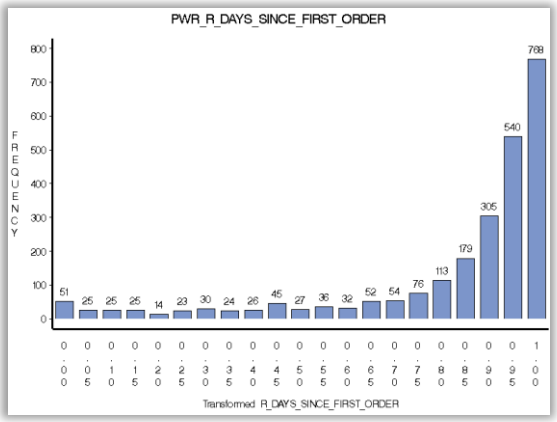
Comparison of M_MIN_SPEND_PER_BASKET
from Original to Transformed (log)





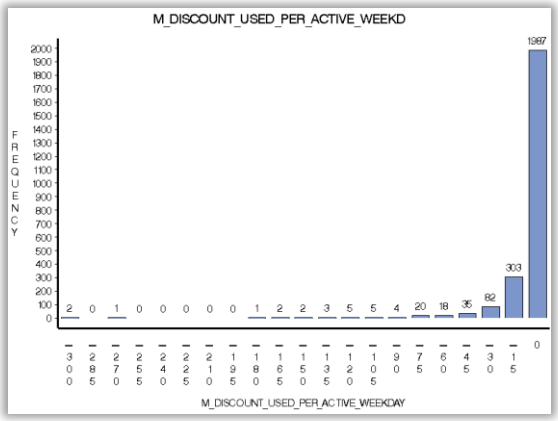
Comparison of R_DAYS_SINCE_LAST_ORDER
from Original to Transformed (log)

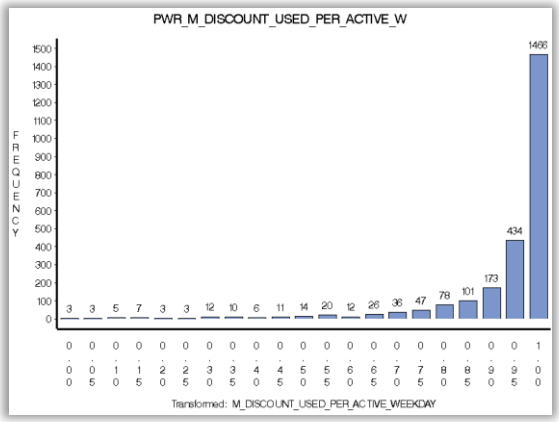




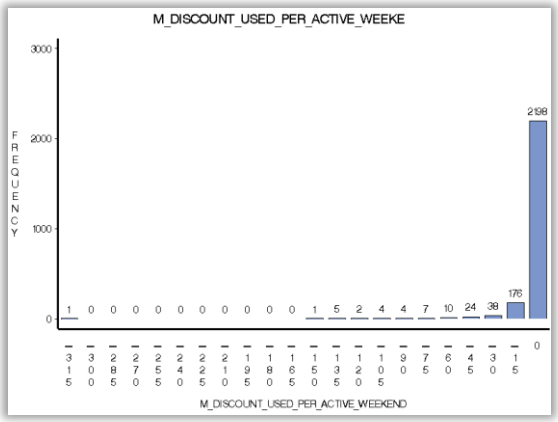
Comparison of R_DAYS_SINCE_FIRST_ORDER
from Original to Transformed (Power)

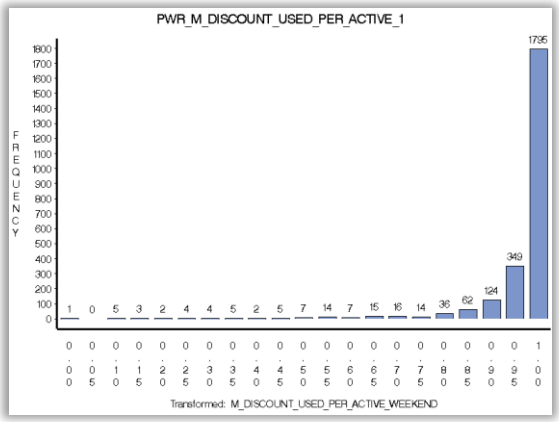
11.04 Appendix 4: Remaining Rough Set Reduct Variables





Comparison of M_DISCOUNT_USED_PER_ACTIVE_WEEKDAY
from Original to Transformed (Power)





Comparison of M_DISCOUNT_USED_PER_ACTIVE_WEEKEND
from Original to Transformed (Power)

11.05 Appendix 5: Rough Set Reduct R Code – 1st Run

```
library(RoughSets)

rs_retail_data <- SF.asDecisionTable(&EMR_IMPORT_DATA, decision.attr = 1, indx.nominal = 1);
rs_d_retail_data <- D.discretization.RST(rs_retail_data, type.method =
"global.discernibility");
new_data <- SF.applyDecTable(rs_retail_data, rs_d_retail_data);
greedy_rst <- FS.feature.subset.computation(new_data, method =
"greedy.heuristic.superreduct");
print(greedy_rst)
```

11.06 Appendix 6: Rough Set Reduct R Code – 2nd Run

```
library(RoughSets)

data <- &EMR_IMPORT_DATA;
new_data <- data[, -29];

rs_retail_data <- SF.asDecisionTable(new_data, decision.attr = 1, indx.nominal = 1);
rs_d_retail_data <- D.discretization.RST(rs_retail_data, type.method =
"global.discernibility");
new_data <- SF.applyDecTable(rs_retail_data, rs_d_retail_data);
greedy_rst <- FS.feature.subset.computation(new_data, method =
"greedy.heuristic.superreduct");
print(greedy_rst)
```

11.07 Appendix 7: Rough Set Reduct R Code – 3rd Run

```
library(RoughSets)

data <- &EMR_IMPORT_DATA;
new_data <- data[, -29];
new_data <- new_data[, -28];

rs_retail_data <- SF.asDecisionTable(new_data, decision.attr = 1, indx.nominal = 1);
rs_d_retail_data <- D.discretization.RST(rs_retail_data, type.method =
"global.discernibility");
new_data <- SF.applyDecTable(rs_retail_data, rs_d_retail_data);
greedy_rst <- FS.feature.subset.computation(new_data, method =
"greedy.heuristic.superreduct");
print(greedy_rst)
```

11.08 Appendix 8: TKM-N-T R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_16_norm_scaled <- scale(data_all[,18:33])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_16_norm_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_16_norm_scaled))
  mean(ss[, 3])
}

k.values <- 2:50

library(purrr)
avg_sil_values <- map_dbl(k.values, avg_sil)

png("TKM_N_T_Optimal_Cluster_Analysis.png")
plot(k.values, avg_sil_values,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
dev.off()
```

11.09 Appendix 9: TKM-N-T R Code – 2 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_16_norm_scaled <- scale(data_all[,18:33])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_16_norm_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_16_norm_scaled))
  mean(ss[, 3])
}

# Choosing k=2 clusters for TKM_N_T
km.res <- kmeans(data_16_norm_scaled, centers = 2, nstart = 25)
TKMNT_cluster_id <- as.data.frame(km.res$cluster)
colnames(TKMNT_cluster_id) <- "TKM_N_T"
data_16_TKMNT <- cbind(data_all[,1:17],
                       TKMNT_cluster_id)

write.csv(data_16_TKMNT, "data_16_TKMNT.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          " _",
          cluster_count,
          " _",
          var_name,
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_16_TKMNT)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))

for (i in 1:nrow(loop_main)){
  segment_profile(data_16_TKMNT,
                  "TKM_N_T",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.10 Appendix 10: TKM-A-T R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_16_scaled <- scale(data_all[,2:17])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_16_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_16_scaled))
  mean(ss[, 3])
}
```

```
k.values <- 2:50

library(purrr)
avg_sil_values <- map_dbl(k.values, avg_sil)

png("TKM_A_T_Optimal_Cluster_Analysis.png")
plot(k.values, avg_sil_values,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
dev.off()
```

11.11 Appendix 11: TKM-A-T R Code – 3 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_16_scaled <- scale(data_all[,2:17])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_16_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_16_scaled))
  mean(ss[, 3])
}

# Choosing k=3 clusters for TKM_A_T
km.res <- kmeans(data_16_scaled, centers = 3, nstart = 25)
TKMAT_cluster_id <- as.data.frame(km.res$cluster)
colnames(TKMAT_cluster_id) <- "TKM_A_T"
data_16_TKMAT <- cbind(data_all,
                       TKMAT_cluster_id)

write.csv(data_16_TKMAT, "data_16_TKMAT.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          " _ ",
          cluster_count,
          " _ ",
          var_name,
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_16_TKMAT)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:3), times = length(loop_table)),
                        X2 = rep(loop_table, each = 3))

for (i in 1:nrow(loop_main)){
  segment_profile(data_16_TKMAT,
                  "TKM_A_T",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.12 Appendix 12: TKM-N-R R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_2_norm_scaled <- scale(data_all[,4:5])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_2_norm_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_2_norm_scaled))
  mean(ss[, 3])
}

k.values <- 2:50

library(purrr)
avg_sil_values <- map_dbl(k.values, avg_sil)

png("TKM_N_R_Optimal_Cluster_Analysis.png")
plot(k.values, avg_sil_values,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
dev.off()
```

11.13 Appendix 13: TKM-N-R R Code – 2 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_norm_scaled <- scale(data_all[,4:5])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_2_norm_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_2_norm_scaled))
  mean(ss[, 3])
}

# Choosing k=2 clusters for TKM_N_T
km.res <- kmeans(data_2_norm_scaled, centers = 2, nstart = 25)
TKMNR_cluster_id <- as.data.frame(km.res$cluster)
colnames(TKMNR_cluster_id) <- "TKM_N_R"
data_2_TKMNR <- cbind(data_all[,1:3],
                      TKMNR_cluster_id)

write.csv(data_2_TKMNR, "data_2_TKMNR.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          " _ ",
          cluster_count,
          " _ ",
          var_name,
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_2_TKMNR)
```

```
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))

for (i in 1:nrow(loop_main)){
  segment_profile(data_2_TKMNR,
                  "TKM_N_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.14 Appendix 14: TKM-A-R R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:3])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_2_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_2_scaled))
  mean(ss[, 3])
}

k.values <- 2:50

library(purrr)
avg_sil_values <- map_dbl(k.values, avg_sil)

png("TKM_A_R_Optimal_Cluster_Analysis.png")
plot(k.values, avg_sil_values,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
dev.off()
```

11.15 Appendix 15: TKM-A-R R Code – 2 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:3])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_2_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_2_scaled))
  mean(ss[, 3])
}

# Choosing k=2 clusters for TKM_A_R
km.res <- kmeans(data_2_scaled, centers = 2, nstart = 25)
TKMAR_cluster_id <- as.data.frame(km.res$cluster)
colnames(TKMAR_cluster_id) <- "TKM_A_R"
data_2_TKMAR <- cbind(data_all,
                      TKMAR_cluster_id)

write.csv(data_2_TKMAR, "data_2_TKMAR.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          " _ ",
          cluster_count,
          " _ ",
          var_name,
          ".png",
```

```
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_2_TKMAR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))


for (i in 1:nrow(loop_main)){
  segment_profile(data_2_TKMAR,
                  "TKM_A_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.16 Appendix 16: FKM-T R Code – Exploration

```
library(fclust);

data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:29])

library(fclust)
avg_sil.f <- function(k) {
  fkm_t <- FKM(data_2_scaled,
               m = 2,
               RS = 1,
               k = k)
  sil.f <- SIL.F(fkm_t$Xca, fkm_t$U)
}

k.values <- 2:20

library(purrr)
avg_sil_values <- map_dbl(k.values, avg_sil.f)

png("FKM_T_Optimal_Cluster_Analysis.png")
plot(k.values, avg_sil_values,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
dev.off()
```

11.17 Appendix 17: FKM-T R Code – 2 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_all_inputs_scaled <- scale(data_all[,2:29])

library(fclust)
# FKM-R (Fuzzzy K-Means on Traditional As-Is Data)
fkm_t <- FKM(data_all_inputs_scaled,
             m = 2,     # Fuzziness parameter
             RS = 1,
             k = 2)

FKMT_cluster_id <- as.data.frame(fkm_t$clus[,1])
colnames(FKMT_cluster_id) <- "FKM_T"
data_all_FKMT <- cbind(data_all,
                       FKMT_cluster_id)

write.csv(data_all_FKMT, "data_all_FKMT.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
```

```
                              var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
           "_",
                     var_name,
          "_",
          cluster_count,
          "_",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_all_FKMT)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))

for (i in 1:nrow(loop_main)){
  segment_profile(data_all_FKMT,
                  "FKM_T",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.18 Appendix 18: FKM-R R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:3])

library(fclust)
avg_sil.f <- function(k) {
  fkm_t <- FKM(data_2_scaled,
               m = 2,
               RS = 1,
               k = k)
  sil.f <- SIL.F(fkm_t$Xca, fkm_t$U)
}

k.values <- 2:20

library(purrr)
avg_sil_values <- map_dbl(k.values, avg_sil.f)

png("FKM_R_Optimal_Cluster_Analysis.png")
plot(k.values, avg_sil_values,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
dev.off()
```

11.19 Appendix 19: FKM-R R Code – 2 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:3])

library(fclust)
# FKM-R (Fuzzzy K-Means on Reduct Data)
fkm_r <- FKM(data_2_scaled,
             m = 2,     # Fuzziness parameter
             RS = 1,
             k = 2)
```

```
FKMR_cluster_id <- as.data.frame(fkm_r$clus[,1])
colnames(FKMR_cluster_id) <- "FKM_R"
data_all_FKMR <- cbind(data_all,
                       FKMR_cluster_id)

write.csv(data_all_FKMR, "data_all_FKMR.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          "_",
                    var_name,
          "_",
          cluster_count,
          "_",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_all_FKMR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))

for (i in 1:nrow(loop_main)){
  segment_profile(data_all_FKMR,
                  "FKM_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.20 Appendix 20: FKM-R R Code – 6 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:3])

library(fclust)
# FKM-R (Fuzzzy K-Means on Reduct Data)
fkm_r <- FKM(data_2_scaled,
             m = 2,     # Fuzziness parameter
             RS = 1,
             k = 6)

FKMR_cluster_id <- as.data.frame(fkm_r$clus[,1])
colnames(FKMR_cluster_id) <- "FKM_R"
data_all_FKMR <- cbind(data_all,
                       FKMR_cluster_id)

write.csv(data_all_FKMR, "data_all_FKMR_6.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
```

```
png(paste(cluster_column_name,
           " - ",
                      var_name,
          " - ",
          cluster_count,
          " - ",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_all_FKMR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))

for (i in 1:nrow(loop_main)){

  segment_profile(data_all_FKMR,
                  "FKM_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.21 Appendix 21: FKM-R R Code – 7 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:3])

library(fclust)
# FKM-R (Fuzzzy K-Means on Reduct Data)
fkm_r <- FKM(data_2_scaled,
             m = 2,     # Fuzziness parameter
             RS = 1,
             k = 7)

FKMR_cluster_id <- as.data.frame(fkm_r$clus[,1])
colnames(FKMR_cluster_id) <- "FKM_R"
data_all_FKMR <- cbind(data_all,
                       FKMR_cluster_id)

write.csv(data_all_FKMR, "data_all_FKMR_7.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
           " - ",
                      var_name,
          " - ",
          cluster_count,
          " - ",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
```

```
}

loop_table <- colnames(data_all_FKMR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))

for (i in 1:nrow(loop_main)){

  segment_profile(data_all_FKMR,
                  "FKM_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.22 Appendix 22: FKM-R R Code – 9 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:3])

library(fclust)
# FKM-R (Fuzzzy K-Means on Reduct Data)
fkm_r <- FKM(data_2_scaled,
             m = 2,     # Fuzziness parameter
             RS = 1,
             k = 9)

FKMR_cluster_id <- as.data.frame(fkm_r$clus[,1])
colnames(FKMR_cluster_id) <- "FKM_R"
data_all_FKMR <- cbind(data_all,
                       FKMR_cluster_id)

write.csv(data_all_FKMR, "data_all_FKMR_9.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          "_",
                     var_name,
          "_",
          cluster_count,
          "_",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_all_FKMR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))

for (i in 1:nrow(loop_main)){

  segment_profile(data_all_FKMR,
                  "FKM_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.23 Appendix 23: GMM-T R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_all_inputs_scaled <- scale(data_all[,2:3])

library(mclust)
gmm_t <- Mclust(data_all_inputs_scaled,
                G = 2:20)

png("GMM_T_Optimal_Cluster_Analysis.png")
plot(gmm_t,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K")
dev.off()
```

11.24 Appendix 24: GMM-T R Code – 8 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_all_inputs_scaled <- scale(data_all[,2:29])

library(mclust)
gmm_t <- Mclust(data_all_inputs_scaled,
                G = 8)
GMMT_cluster_id <- as.data.frame(gmm_t$classification)
colnames(GMMT_cluster_id) <- "GMM_T"
data_all_GMMT <- cbind(data_all,
                       GMMT_cluster_id)

write.csv(data_all_GMMT, "data_all_GMMT.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          "_",
                    var_name,
          "_",
          cluster_count,
          "_",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_all_GMMT)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:8), times = length(loop_table)),
                        X2 = rep(loop_table, each = 8))

for (i in 1:nrow(loop_main)){
  segment_profile(data_all_GMMT,
                  "GMM_T",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.25 Appendix 25: GMM-R R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:3])

library(mclust)
gmm_r <- Mclust(data_2_scaled,
                G = 2:20)

png("GMM_R_Optimal_Cluster_Analysis.png")
plot(gmm_r,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K")
dev.off()
```

11.26 Appendix 26: GMM-R R Code – 19 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_inputs_scaled <- scale(data_all[,2:3])

library(mclust)
gmm_r <- Mclust(data_2_inputs_scaled,
                G = 19)
GMMR_cluster_id <- as.data.frame(gmm_r$classification)
colnames(GMMR_cluster_id) <- "GMM_R"
data_all_GMMR <- cbind(data_all,
                       GMMR_cluster_id)

write.csv(data_all_GMMR, "data_all_GMMR.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          "_",
                    var_name,
          "_",
          cluster_count,
          "_",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_all_GMMR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:19), times = length(loop_table)),
                        X2 = rep(loop_table, each = 19))

for (i in 1:nrow(loop_main)){
  segment_profile(data_all_GMMR,
                  "GMM_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.27 Appendix 27: GMM-R R Code – 8 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_inputs_scaled <- scale(data_all[,2:3])

library(mclust)
gmm_r <- Mclust(data_2_inputs_scaled,
                G = 8)
GMMR_cluster_id <- as.data.frame(gmm_r$classification)
colnames(GMMR_cluster_id) <- "GMM_R"
data_all_GMMR <- cbind(data_all,
                       GMMR_cluster_id)

write.csv(data_all_GMMR, "data_all_GMMR_8.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          "_",
                    var_name,
          "_",
          cluster_count,
          "_",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_all_GMMR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:19), times = length(loop_table)),
                        X2 = rep(loop_table, each = 19))

for (i in 1:nrow(loop_main)){

  segment_profile(data_all_GMMR,
                  "GMM_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.28 Appendix 28: TKM-N-R3 R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_2_norm_scaled <- scale(data_all[,5:7])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_2_norm_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_2_norm_scaled))
  mean(ss[, 3])
}

k.values <- 2:50

library(purrr)
avg_sil_values <- map_dbl(k.values, avg_sil)
```

```
png("TKM_N_R3_Optimal_Cluster_Analysis.png")
plot(k.values, avg_sil_values,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
dev.off()
```

11.29 Appendix 29: TKM-N-R3 R Code – 2 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_norm_scaled <- scale(data_all[,5:7])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_2_norm_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_2_norm_scaled))
  mean(ss[, 3])
}

# Choosing k=2 clusters for TKM_N_T
km.res <- kmeans(data_2_norm_scaled, centers = 2, nstart = 25)
TKMNR_cluster_id <- as.data.frame(km.res$cluster)
colnames(TKMNR_cluster_id) <- "TKM_N_R"
data_2_TKMNR <- cbind(data_all[,1:3],
                      TKMNR_cluster_id)

data_2_TKMNR_csv <- cbind(data_all,
                      TKMNR_cluster_id)
write.csv(data_2_TKMNR_csv, "data_2_TKMNR_R3_2.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          "_",
                      var_name,
          "_",
          cluster_count,
          "_",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_2_TKMNR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))

for (i in 1:nrow(loop_main)){

  segment_profile(data_2_TKMNR,
                  "TKM_N_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.30 Appendix 30: TKM-A-R3 R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:4])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_2_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_2_scaled))
  mean(ss[, 3])
}

k.values <- 2:50

library(purrr)
avg_sil_values <- map_dbl(k.values, avg_sil)

png("TKM_A_R3_Optimal_Cluster_Analysis.png")
plot(k.values, avg_sil_values,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
dev.off()
```

11.31 Appendix 31: TKM-A-R3 R Code – 2 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:4])

set.seed(1234)

library(cluster)
avg_sil <- function(k) {
  km.res <- kmeans(data_2_scaled, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data_2_scaled))
  mean(ss[, 3])
}

# Choosing k=2 clusters for TKM_A_R
km.res <- kmeans(data_2_scaled, centers = 2, nstart = 25)
TKMAR_cluster_id <- as.data.frame(km.res$cluster)
colnames(TKMAR_cluster_id) <- "TKM_A_R"
data_2_TKMAR <- cbind(data_all,
                      TKMAR_cluster_id)

write.csv(data_2_TKMAR, "data_2_TKMAR_R3.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
           " – ",
                     var_name,
          " _",
          cluster_count,
          " _",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}
```

```
loop_table <- colnames(data_2_TKMAR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))


for (i in 1:nrow(loop_main)){

  segment_profile(data_2_TKMAR,
                  "TKM_A_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.32 Appendix 32: GMM-R3 R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_3_scaled <- scale(data_all[,2:4])

library(mclust)
gmm_r <- Mclust(data_3_scaled,
                G = 2:20)

png("GMM_R3_Optimal_Cluster_Analysis.png")
plot(gmm_r$BIC,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K")
dev.off()
```

11.33 Appendix 33: GMM-R3 R Code – 8 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_inputs_scaled <- scale(data_all[,2:4])

library(mclust)
gmm_r <- Mclust(data_2_inputs_scaled,
                G = 8)
GMMR_cluster_id <- as.data.frame(gmm_r$classification)
colnames(GMMR_cluster_id) <- "GMM_R"
data_all_GMMR <- cbind(data_all,
                       GMMR_cluster_id)

write.csv(data_all_GMMR, "data_all_GMMR_R3_8.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
          " – ",
                    var_name,
          " ",
          – ,
          cluster_count,
          " – ",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_all_GMMR)
loop_table <- loop_table[-1]
```

```
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:19), times = length(loop_table)),
                        X2 = rep(loop_table, each = 19))

for (i in 1:nrow(loop_main)){

  segment_profile(data_all_GMMR,
                  "GMM_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.34 Appendix 34: FKM-R3 R Code – Exploration

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:4])

library(fclust)
avg_sil.f <- function(k) {
  fkm_t <- FKM(data_2_scaled,
               m = 2,
               RS = 1,
               k = k)
  sil.f <- SIL.F(fkm_t$Xca, fkm_t$U)
}

k.values <- 2:20

library(purrr)
avg_sil_values <- map_dbl(k.values, avg_sil.f)

png("FKM_R3_Optimal_Cluster_Analysis.png")
plot(k.values, avg_sil_values,
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
dev.off()
```

11.35 Appendix 35: FKM-R3 R Code – 5 Optimised Clusters

```
data_all <- &EMR_IMPORT_DATA
data_2_scaled <- scale(data_all[,2:4])

library(fclust)
# FKM-R (Fuzzzy K-Means on Reduct Data)
fkm_r <- FKM(data_2_scaled,
             m = 2,     # Fuzziness parameter
             RS = 1,
             k = 5)

FKMR_cluster_id <- as.data.frame(fkm_r$clus[,1])
colnames(FKMR_cluster_id) <- "FKM_R"
data_all_FKMR <- cbind(data_all,
                       FKMR_cluster_id)

write.csv(data_all_FKMR, "data_all_FKMR_R3.csv")

segment_profile <- function(data,
                            cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data[which(data[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))
png(paste(cluster_column_name,
           " _ ",
                       var_name,
          " _",
          cluster_count,
          " _",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
```

```
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue",
                  sep = ""),
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table <- colnames(data_all_FKMR)
loop_table <- loop_table[-1]
loop_table <- loop_table[-length(loop_table)]

loop_main <- data.frame(X1 = rep((1:2), times = length(loop_table)),
                        X2 = rep(loop_table, each = 2))

for (i in 1:nrow(loop_main)){

  segment_profile(data_all_FKMR,
                  "FKM_R",
                  loop_main[i,1],
                  loop_main[i,2])
}
```

11.36 Appendix 36: Segment Profile R Code

```
data_all <- &EMR_IMPORT_DATA

segment_profile <- function(cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data_all[which(data_all[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)
hgA <- hist(data_all[[var_name]], plot = FALSE)
hgB <- hist(data_temp[[var_name]], plot = FALSE)
r_length <- range(c(hgA$breaks, hgB$breaks))

png(paste(cluster_column_name,
          " ",
          var_name,
          " ",
          cluster_count,
          " ",
          ".png",
          sep = ""))
plot(hgA,
     col = c_red,
     xlim = c(r_length[1], r_length[2]),
     main = paste("Segment Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue ",
                  sep = ""),
     sub = cluster_column_name,
     xlab = var_name)
plot(hgB, col = c_blue, add = TRUE)
dev.off()
}

loop_table_clusters <- colnames(data_all[,30:44])
loop_table_cluster_counts <- data.frame(c(2,
                                          2,
                                          3,
                                          2,
                                          2,
                                          2,
                                          8,
                                          19,
                                          8,
                                          2,
                                          2,
                                          8,
                                          5,
                                          7,
                                          9))
colnames(loop_table_cluster_counts) <- "Counts"

loop_table_clusters <- cbind(loop_table_clusters,
                             loop_table_cluster_counts)
```

```
loop_table_clusters$loop_table_clusters <-
as.character(loop_table_clusters$loop_table_clusters)

loop_main <- data.frame()
for (i in 1:nrow(loop_table_clusters)){

  loop_temp <- data.frame(X1 = rep((1:as.numeric(loop_table_clusters[i,2])),
                                   times = 1),
                          X2 = rep(loop_table_clusters[i,1],
                                   each = as.numeric(loop_table_clusters[i,2])))

  loop_main <- rbind(loop_main,
                     loop_temp)

}

loop_table_vars2 <- data.frame()
for (i in 1:81){

  loop_table_vars2_temp <- loop_main[rep(i,28),]
  loop_table_vars2 <- rbind(loop_table_vars2,
                            loop_table_vars2_temp)

}

loop_table_vars2$X2 <- as.character(loop_table_vars2$X2)

loop_main$X2 <- as.character(loop_main$X2)
loop_table_vars <- as.character(colnames(data_all[,2:29]))
loop_table_vars <- data.frame(X3 = loop_table_vars)
loop_table_vars$X3 <- as.character(loop_table_vars$X3)

loop_table_vars1 <- data.frame()
for (i in 1:81){
loop_table_vars1 <- rbind(loop_table_vars1,
                          loop_table_vars)
}

loop_main2 <- cbind(loop_table_vars1,
                    loop_table_vars2)

for (i in 1:nrow(loop_main2)){
  segment_profile(loop_main2[i,3],
                  loop_main2[i,2],
                  loop_main2[i,1])
}
```

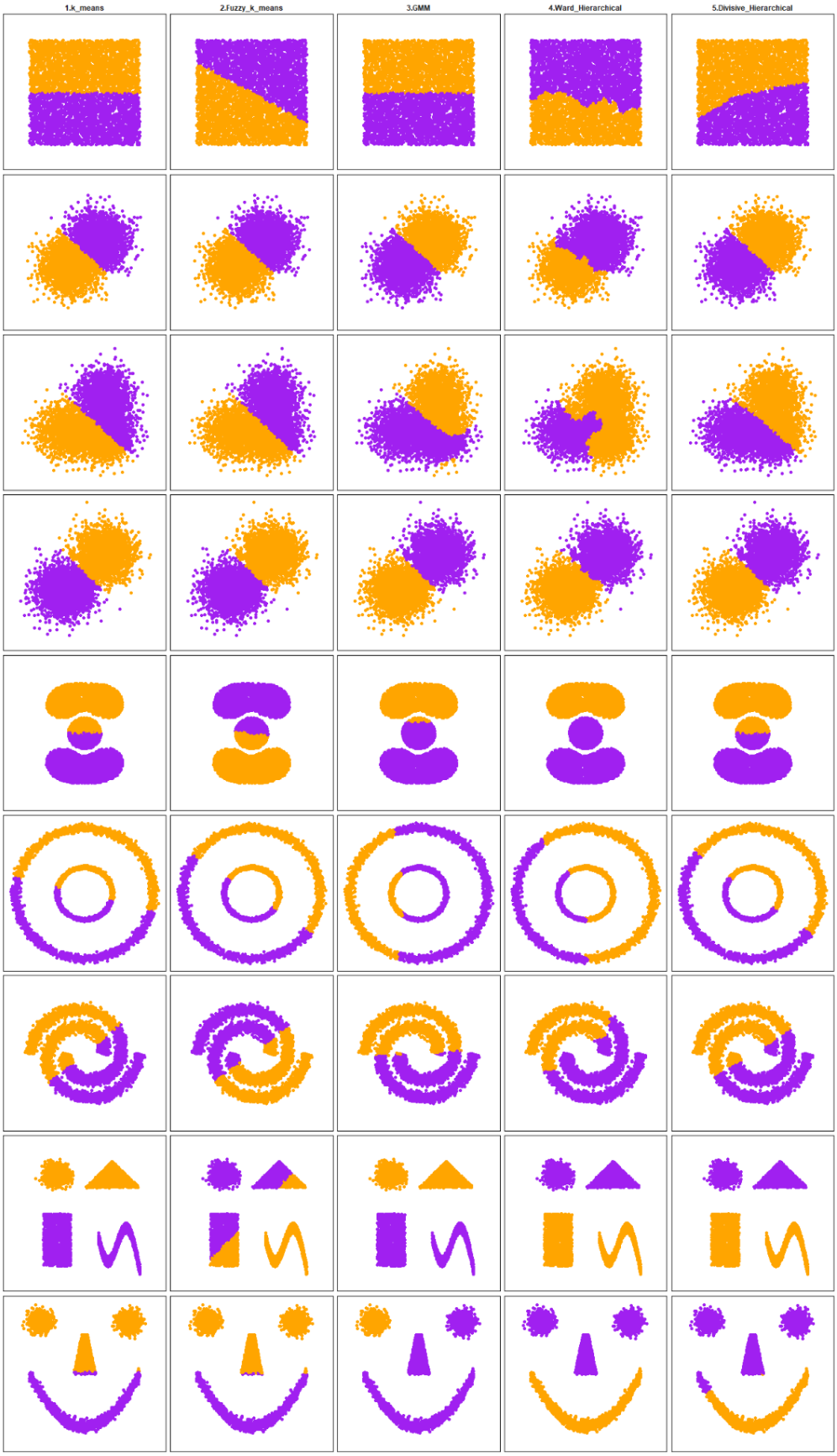
11.37 Appendix 37: Household Profile R Code

```
data_all <- &EMR_IMPORT_DATA
data_all <- data_all[, c(2:7,37:51)]
data_all <- data_all[!is.na(data_all$KID_CATEGORY_DESC),]

segment_profile <- function(cluster_column_name,
                            cluster_count,
                            var_name){
data_temp <- data_all[which(data_all[[cluster_column_name]] == cluster_count),]
c_blue <- rgb(142, 171, 232, max = 255)
c_red <- rgb(255, 0, 0, max = 255)

png(paste(cluster_column_name,
          " - ",
          var_name,
          " - ",
          cluster_count,
          " - ",
          ".png",
          sep = ""))
plot(data_all[[var_name]],
     col = c_red,
     main = paste("Segment HH Profile of Overall(Red) vs. Cluster ",
                  cluster_count,
                  "Blue ",
                  sep = ""),
     sub = cluster_column_name,
     xlab = var_name)
plot(data_temp[[var_name]], col = c_blue, add = TRUE)

dev.off()
}

loop_table_clusters <- colnames(data_all[,7:21])
```

```
loop_table_cluster_counts <- data.frame(c(2,
                                          2,
                                          3,
                                          2,
                                          2,
                                          2,
                                          8,
                                          19,
                                          8,
                                          2,
                                          2,
                                          8,
                                          5,
                                          7,
                                          9))
colnames(loop_table_cluster_counts) <- "Counts"

loop_table_clusters <- cbind(loop_table_clusters,
                             loop_table_cluster_counts)

loop_table_clusters$loop_table_clusters <-
as.character(loop_table_clusters$loop_table_clusters)

loop_main <- data.frame()
for (i in 1:nrow(loop_table_clusters)){

  loop_temp <- data.frame(X1 = rep((1:as.numeric(loop_table_clusters[i,2])),
                                   times = 1),
                          X2 = rep(loop_table_clusters[i,1],
                                   each = as.numeric(loop_table_clusters[i,2])))

  loop_main <- rbind(loop_main,
                     loop_temp)

}


loop_table_vars2 <- data.frame()
for (i in 1:81){

  loop_table_vars2_temp <- loop_main[rep(i,6),]
  loop_table_vars2 <- rbind(loop_table_vars2,
                            loop_table_vars2_temp)

}

loop_table_vars2$X2 <- as.character(loop_table_vars2$X2)

loop_main$X2 <- as.character(loop_main$X2)
loop_table_vars <- as.character(colnames(data_all[,1:6]))
loop_table_vars <- data.frame(X3 = loop_table_vars)
loop_table_vars$X3 <- as.character(loop_table_vars$X3)

loop_table_vars1 <- data.frame()
for (i in 1:81){

loop_table_vars1 <- rbind(loop_table_vars1,
                          loop_table_vars)
}

loop_main2 <- cbind(loop_table_vars1,
                    loop_table_vars2)

for (i in 1:nrow(loop_main2)){

  segment_profile(loop_main2[i,3],
                  loop_main2[i,2],
                  loop_main2[i,1])
}
```

11.38 Appendix 38: Cluster Comparison on Randomised Data

Inspired by https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html



11.39 Appendix 39: Reproducible R Code for Randomised Data

```
n <- 5000 # Number of samples

# Creating Rectangle plot
data_rectangle <- mlbench.xor(n,
                              d = 2)

data_rectangle <- as.tibble(data_rectangle$x) %>%
  rename("X" = "V1",
         "Y" = "V2") %>%
  mutate("Shape" = "Rectangle") %>%
  mutate("Number" = 1)

data_rectangle$X <- data_rectangle$X*4 # Enlarge Factor
data_rectangle$Y <- data_rectangle$Y*4 # Enlarge Factor

ggplot(data_rectangle,
       aes(X, Y)) +
  geom_point()

# Creating Blob
data_2dnorm <- mlbench.2dnormals(n,
                                 cl = 2,
                                 sd = 1)

data_2dnorm <- as.tibble(data_2dnorm$x) %>%
  rename("X" = "V1",
         "Y" = "V2") %>%
  mutate("Shape" = "2dnorm") %>%
  mutate("Number" = 2)

ggplot(data_2dnorm,
       aes(X, Y)) +
  geom_point()

# Creating Right-Sided Blob
data_blob <- mlbench.threenorm(n,
                               d = 2)

data_blob <- as.tibble(data_blob$x) %>%
  rename("X" = "V1",
         "Y" = "V2") %>%
  mutate("Shape" = "Blob1") %>%
  mutate("Number" = 3)

ggplot(data_blob,
       aes(X, Y)) +
  geom_point()

# Creating Twonorm (Two Blobs)
data_twonorm <- mlbench.twonorm(n,
                                d = 2)

data_twonorm <- as.tibble(data_twonorm$x) %>%
  rename("X" = "V1",
         "Y" = "V2") %>%
  mutate("Shape" = "Blob2") %>%
  mutate("Number" = 4)

ggplot(data_twonorm,
       aes(X, Y)) +
  geom_point()

# Creating Cassini
data_cassini <- mlbench.cassini(n)

data_cassini <- as.tibble(data_cassini$x) %>%
  rename("X" = "V1",
         "Y" = "V2") %>%
  mutate("Shape" = "Cassini") %>%
  mutate("Number" = 5)

data_cassini$X <- data_cassini$X*2 # Enlarge Factor
data_cassini$Y <- data_cassini$Y*2 # Enlarge Factor

ggplot(data_cassini,
       aes(X, Y)) +
  geom_point()

# Creating Two Circles (One within another)
theta <- runif(n/2, 0, 2*pi)
x <- cos(theta) + rnorm(n/2, 0, 0.03)
y <- sin(theta) + rnorm(n/2, 0, 0.03)
```

```
data_circles1 <- tibble(X = x*5,
                        Y = y*5) %>%
  mutate("Shape" = "Circles") %>%
  mutate("Number" = 6)

data_circles2 <- tibble(X = x*2,
                        Y = y*2) %>%
  mutate("Shape" = "Circles") %>%
  mutate("Number" = 6)

data_circles <- bind_rows(data_circles1,
                          data_circles2)

ggplot(data_circles,
       aes(X, Y)) +
  geom_point()

# Creating Spirals
data_spiral <- mlbench.spirals(n,
                               cycles = 1,
                               sd = 0.05)

data_spiral <- as.tibble(data_spiral$x) %>%
  rename("X" = "V1",
         "Y" = "V2") %>%
  mutate("Shape" = "Spiral") %>%
  mutate("Number" = 7)

data_spiral$X <- data_spiral$X*4 # Enlarge Factor
data_spiral$Y <- data_spiral$Y*4 # Enlarge Factor

ggplot(data_spiral,
       aes(X, Y)) +
  geom_point()

# Creating Multiple Shapes
data_shapes <- mlbench.shapes(n)

data_shapes <- as.tibble(data_shapes$x) %>%
  rename("X" = "x4",
         "Y" = "V2") %>%
  mutate("Shape" = "Multi") %>%
  mutate("Number" = 8)

data_shapes$X <- data_shapes$X*2 # Enlarge Factor
data_shapes$Y <- data_shapes$Y*2 # Enlarge Factor

ggplot(data_shapes,
       aes(X, Y)) +
  geom_point()

# Creating Smileys
data_smiley <- mlbench.smiley(n,
                              sd1 = 0.1,
                              sd2 = 0.05)

data_smiley <- as.tibble(data_smiley$x) %>%
  rename("X" = "x4",
         "Y" = "V2") %>%
  mutate("Shape" = "Smiley") %>%
  mutate("Number" = 9)

data_smiley$X <- data_smiley$X*4 # Enlarge Factor
data_smiley$Y <- data_smiley$Y*4 # Enlarge Factor

ggplot(data_smiley,
       aes(X, Y)) +
  geom_point()

data_consol <- bind_rows(data_rectangle,
                         data_2dnorm,
                         data_blob,
                         data_twonorm,
                         data_cassini,
                         data_circles,
                         data_spiral,
                         data_shapes,
                         data_smiley)

rm(data_rectangle,
   data_2dnorm,
   data_blob,
   data_twonorm,
   data_cassini,
   data_circles,
   data_circles1,
   data_circles2,
```

```
   data_spiral,
   data_shapes,
   data_smiley)
```

11.40 Appendix 40: Reproducible R Code for Clustering Example

```
seed <- 1234
set.seed(1234)
cluster_count <- 2

# Calculate k-means
data_consol_kmeans <- tibble()
#i <- 1

for (i in 1:max(data_consol$Number)){

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    subset(select = c("X", "Y"))

  data_cluster <- kmeans(data_temp,
                         centers = cluster_count)

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    mutate("Cluster_Type" = "1.k_means") %>%
    bind_cols(data_cluster$cluster) %>%
    rename("Cluster" = "...6")

  data_consol_kmeans <- bind_rows(data_consol_kmeans,
                                  data_temp)

  print(paste(i,
              "of",
              max(data_consol$Number),
              "k-means data",
              sep = " "))

}

# Calculate GMM (Gaussian Mixture Model)
data_consol_GMM <- tibble()
#i <- 1

for (i in 1:max(data_consol$Number)){

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    subset(select = c("X", "Y"))

  data_cluster <- Mclust(data_temp,
                         G = cluster_count)

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    mutate("Cluster_Type" = "3.GMM") %>%
    bind_cols(data_cluster$classification) %>%
    rename("Cluster" = "...6")

  data_consol_GMM <- bind_rows(data_consol_GMM,
                               data_temp)

  print(paste(i,
              "of",
              max(data_consol$Number),
              "GMM data",
              sep = " "))

}

# Calculate Fuzzy k-means
data_consol_fuzzy <- tibble()
#i <- 1

for (i in 1:max(data_consol$Number)){

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    subset(select = c("X", "Y"))

  data_cluster <- FKM(data_temp,
                      m = 2,     # Fuzziness parameter
```

```
                      RS = 1,
                      stand = 1,
                      index = "SIL.F",
                      seed = seed,
                      k = cluster_count)

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    mutate("Cluster_Type" = "2.Fuzzy_k_means") %>%
    bind_cols(data_cluster$clus) %>%
    select(-7)

  data_consol_fuzzy <- bind_rows(data_consol_fuzzy,
                                 data_temp)

  print(paste(i,
              "of",
              max(data_consol$Number),
              "Fuzzy k-means data",
              sep = " "))

}

# Calculate Hierarchical Ward Agglomerative Clustering
data_consol_ward <- tibble()
#i <- 1

for (i in 1:max(data_consol$Number)){

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    subset(select = c("X", "Y"))

  data_cluster <- dist(data_temp,
                       method = "euclidean")

  data_cluster <- hclust(data_cluster,
                         method = "ward.D")

  groups <- cutree(data_cluster,
                   k = cluster_count) %>%
    tibble() %>%
    rename("Cluster" = ".")

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    mutate("Cluster_Type" = "4.Ward_Hierarchical") %>%
    bind_cols(groups)

  data_consol_ward <- bind_rows(data_consol_ward,
                                data_temp)

  print(paste(i,
              "of",
              max(data_consol$Number),
              "Ward Hierarchical data",
              sep = " "))

}

# Calculate Hierarchical Divisive Clustering
data_consol_div <- tibble()
#i <- 1

for (i in 1:max(data_consol$Number)){

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    subset(select = c("X", "Y"))

  data_cluster <- diana(data_temp)

  groups <- cutree(data_cluster,
                   k = cluster_count) %>%
    tibble() %>%
    rename("Cluster" = ".")

  data_temp <- data_consol %>%
    filter(Number == i) %>%
    mutate("Cluster_Type" = "5.Divisive_Hierarchical") %>%
    bind_cols(groups)

  data_consol_div <- bind_rows(data_consol_div,
                               data_temp)

  print(paste(i,
              "of",
```

```
               max(data_consol$Number),
               "Divisive Hierarchical data",
               sep = " "))

}

rm(data_temp,
   data_cluster,
   groups)

# Putting It All Together

data_consol_all <- bind_rows(data_consol_kmeans,
                             data_consol_GMM,
                             data_consol_fuzzy,
                             data_consol_ward,
                             data_consol_div)

ggplot(data_consol_all,
       aes(x = X,
           y = Y,
           color = Cluster)) +
  geom_point() +
  facet_grid(Number ~ Cluster_Type) +
  labs(x = NULL,
       y = NULL) +
  scale_color_gradient(low = "purple", high = "orange") +
  theme(axis.text = element_blank(),
        axis.ticks = element_blank(),
        legend.position = "none",
        panel.background = element_blank(),
        panel.border = element_rect(colour = "black",
                                    fill = NA),
        panel.grid = element_blank(),
        strip.background = element_blank(),
        strip.text.x = element_text(size = 9,
                                    face = "bold"),
        strip.text.y = element_blank())
```