Part 2a.

The Apriori algorithm was designed to provide predictive capabilities through the analysis of k-itemsets, looking for items frequented together to formulate predictions within specific parameters. This method iteratively scans a given set of data and creates k number of itemsets, depending on what is specified. While this method could become computationally expensive depending on the size of the dataset, there are techniques employed to reduce the number of transactions being scanned. For example, the Apriori property will ensure a specified minimum threshold is met by a given itemset, otherwise it will not be scanned. Additionally, antimonotonicity will ensure all supersets of that itemset are not scanned either. Once frequent itemsets are collected, rules can be created based on a few different metrics, namely support, confidence, and lift. The support percentage threshold highlights the frequency of the antecedent itemset. The confidence percentage threshold tells us the frequency percentage of the antecedent and consequent itemsets together. Lift tells us the correlation between both itemsets. If the lift is exactly 1, there is no correlation. If the lift is above 1, there is a positive correlation, and below 1 means a negative correlation (Han, Kamber, & Pei, 2011).

Part2ci

In the above command, we ran the Apriori algorithm on our credit data with the default parameters. The default parameters, in this instance, are a confidence level of 0.8, support of 0.1, minimum length of 1, maximum length of 10, and rules is the target. In the case of our credit data, 690 transactions were processed producing 14,152 rules.

Part 2cii

An example of one rule which was returned, if the antecedent were blank, then the consequent is predicted to that A13 will equal "g" with a support of approximately 0.9, confidence of 0.9, lift of 1.0 with a count of 625. This highlights a need to modify the default settings to retrieve more interesting results.

Part 2di

In the above command, I have changed the minimum length to 2, support to 0.2, and confidence to 0.9. The purpose of this was to remove the rules with a blank antecedent and to narrow the resulting rules to those with higher levels of support and confidence in hopes of discovering more interesting rules. This command generated 1,281 rules.

Part 2di\_2

Look at the first 10 rules in our new rules set, we can see one the strongest rules are rules [4] and [5]. In rule [4] we see that the antecedent of value "A4=y" is predicting a value of "A5=p" in the consequent. The support of 0.23, confidence of 1.0, lift of 4.23 and count of 163 are the highest across the board in these first 10 rules. This tells us there is a high support of both "A4=y" and "A5=p" in the transactions, and there is a high correlation between the two because of the high lift. Rule [5] simple has the antecedent and consequent reversed.

Part 2dii

In the following command we leave the minimum length, confidence, and support values the same as before and now adjust the appearance setting to only show values in the consequent where A5 is equal to "p" and A9 is equal to "t" to determine if there are more interesting rules in the narrower set. In this case, we have generated 44 rules.

Part2dii\_2

In the above results, we can see the same rule from the previous section is still the strongest here, with the antecedent being "A4=y" and then consequent being "A5=p", now showing as rule [1]. Rule [3] is also showing potentially interesting results, with the antecedent being "A4=y,A15=[0,1.91e+03]" and the consequent being equal to "A5=p". The support is 0.22, confidence of 1.0, and lift of 4.23. Rule [3], however, might be considered redundant since the confidence of rule [1] is 100% and these two rules are very similar.

Part 2diii

From the above commands, we can see changing the confidence, support, and minimum length settings can cause a significant change in the resulting rules. For example, if we adjust the minimum length to 2, versus the default of 1, we can immediately remove the uninteresting rules with blank item sets. Adjusting the support and confidence values will help narrow the rules to a more usable set. For instance, we might want to make sure our rules only showing items having a support of at least 50%, so we would need to adjust this value to 0.5. The confidence of a set is also an important attribute for finding interesting predictions, so we would want to adjust it accordingly to remove any rules that are not interesting. For example, we might only want to see rules that have 100% confidence, so we would need to set the value to 1.000.

Part 2div

Confidence, support, and lift are different metrics we can use within the Apriori algorithm to narrow our parameters and discover more interesting predictive patterns in the datasets. The support threshold is a metric for the antecedent itemset, which is also known as the left-hand side. In this case, the metric is telling us the percentage of how often k-itemset is seen in the entire dataset. Confidence, on the other hand, tells us how often the antecedent itemset and consequent itemset are seen together. While these can be useful metrics, they do not always tell the whole story. If there happens to just be a lot of the consequent itemset, there is not necessarily a correlation. The lift metric takes into consideration the support metric for both itemsets and determines if there is a positive, negative, or zero correlation between the specified itemsets (Ng, 2016).

Part 2ei

In the above command, I adjusted the appearance setting of the apriori command, including a list for the right-hand itemsets to only include transactions containing a class attribute value of "+" or "-". Leaving the default support, confidence, and lift we found 1,394 rules. Rule [2], only looking at the first 10, had the highest positive correlation between the itemset {A7=h,A9=t} and itemset {class=+}.

Part 2eii

Examining the first 10 rules created in the previous step provides interesting results. Although the attributes are masked, we can still infer some meaning. As mentioned in the previous post, rule [2] has the strongest positive correlation between the itemset {A7=h,A9=t} and itemset {class=+}. The support was 0.12, confidence at 0.86, and the lift was 1.94. After examining A7 and A9, it was noted there are only two possible values for A9, either "f" or "t", and A7 has 10 possible values. For this argument, we could assume A7 is relating to some type of credit code assigned to individuals based on their credit rating, A9 might refer to the person's gender, and class might be the result of whether the credit application was approved or denied. Looking at the rule [2]. we might deduce there is approximately 12% support that the applicants have a code "h", these applicants are male, and there is an 86% confidence level these applicants have an approved credit application. The lift value of 1.94 indicates there is a positive correlation between males with a code h having approved credit applications in this example. Likewise, rule [1] states an antecedent itemset {A9=f} relates to a consequent itemset of {class=-} with 0.44 support, 0.93 confidence, and 1.67 lift. Using the same example previously, we might say female applicants have a 44% support level within the dataset, and along with that a 93% confidence level of being associated with a disapproved credit application. There is a positive correlation between these itemsets as well.

Part 2fi

Pruning is an important function for efficient utilization of the Apriori algorithm. This is referring to the process of removing transactions that are not interesting for any association rules. The Apriori property, mentioned in a previous answer, plays a part in this ensuring our minimum thresholds are maintained. Other actions taken to properly prune the dataset includes item merging, sub-itemset pruning, and item skipping. Item merging will form frequent closed itemsets if there is no superset of the consequent itemset. Sub-itemset pruning works by removing descendants of already discovered supersets. Item skipping involves pruning itemsets that have equivalent support at varying levels of different header tables (Han, Kamber, & Pei, 2011).

Part 2fii

I had to run the following code block to get this section to work properly:

rules.sorted<-sort(rules, by="lift")

inspect(rules.sorted)

redundant <- is.redundant(rules.sorted)

redundant

which(redundant)

When running the "subset.matrix" commands I would get the following error:

"Warning message:

In `[<-`(`\*tmp\*`, as.vector(i), value = NA) :

x[.] <- val: x is “ngTMatrix”, val not in {TRUE, FALSE} is coerced; NA |--> TRUE."

It turns out this is an antiquated form of finding these redundant rules, and the "is.redundant(rules.sorted)" command is an upgraded form of finding the redundant rules. Of note, our original rule set had over 1,394 rules, and the rules pruned was narrowed down to 240 rules. This highlights the need for this step, there are many redundant rules that do not provide any additional interesting points and can be removed.

Part 2fiii

Now that we have sorted our rules by the lift value, and since we have pruned redundant rules we can now determine more interesting results about the credit dataset. For example, the rule with the highest lift value, rule [1], tells us the antecedent itemset {A4=u,A9=t,A10=t,A14=[0,80.6)} has approximately 11% support in the dataset, along with 97% confidence of being matched up with the consequent value of {class=+}, and an overall positive correlation of 2.19.

Part 2g

The following plot diagram is a basic visual representation of our confidence, support, and lift metrics. While we cannot use this visualization for specific details, it does provide a broad overview of where most of our pruned rules stand. This type of plot would be a useful first step in determining, by visual means, whether our rule set will be effective in providing interesting results. For example, if many of the points are in the upper right corner, indicating high values for all three metrics, then we know we will have many interesting results. However, if many of the points are in the lower left corner, we might need to revisit our rule set and modify accordingly.

Part 2hi

While every metric provides a critical piece of information, it is rare to say one metric is good enough to provide strong rules. For this, let us consider the support, confidence, and lift metrics discussed thoroughly in this exercise. The support metric is, without doubt, a necessary value. A support percentage tells us how often the antecedent itemset is seen in the dataset. However, this value alone does not provide any reliable prediction data we are looking for. At a minimum, we would also need the confidence metric. The confidence percentage will tell us how often the consequent value was seen with the antecedent. These two values will provide enough data to begin making predictive assumptions. However, for more accurate results we would want to include the lift metric. This will provide us the positive or negative correlation between the different itemsets, or lack thereof (Ng, 2016).

Part 2hii

The most challenging part of this exercise was comprehending the different formulas used in determining the support, confidence, and lift values.

Part 2hiii

The tutorial listed several alternative metrics, such as chiQuare, conviction, and cosine. The purpose of these different metrics is no different than the already discussed metrics, and that is to provide additional data metrics to further provide interesting results and validate these results. For example, the conviction is comparable to confidence, except it goes further by including the lack of a consequent value in the calculation (Hahsler, 2015).

Hahsler, M. (2015). *A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules*. Retrieved from http://michael.hahsler.net/research/association\_rules/measures.html#conviction

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques.* Elsevier.

Ng, A. (2016, April). *Association Rules and the Apriori Algorithm: A Tutorial*. Retrieved from KDnuggets: https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html