```
#################################################
#####     ASSOCIATION ANALYSIS EXERCISE      #####
#################################################

## This is a "scripted exercise". Some of the less-
## intuitive statements have been provided to you. The
## others, you should provide (where indicated):

## SURVEY DATA
## Data Preparation

# In this example we use the 'survey' dataset
# provided to you in your daily folder.

# Use the read.csv() function to read in the
# data and assign it to an object 'survey' in
# your workspace.

# Then review the dataset. How many rows and
# variables are there in the data set?

# <fill in code to read in data>
# <fill in code to answer question rows and vars>

# What are the dimensions of the dataframe?
# <fill in code to cite dimensions>

# What is the structure of the dataframe?
# <fill in code>

# Run the summary() function to get an idea
```

```
# of the "spread" of the numerics.
# <fill in code>

# Look at the first five rows to
# get some idea of the data.
# <fill in code.

# The dataset contains a mixture of categoric
# and numeric variables while the apriori
# algorithm works just with categoric variables
# (or factors). We note that the variable
# 'fnlwgt' is a calculated value and not of
# interest to us so remove it from the dataset.
# The variable 'Education.Num' is redundant
# since is it simply a numeric mapping of
# Education. Remove that variable as well.

survey$fnlwgt <- NULL
survey$Education.Num <- NULL

# This still leaves Age, Capital.Gain, Capital.Loss,
# and Hours.Per.Week. We will partition Age and
# Hours.Per.Week into fours segments each with
# the following code:

survey$Age <- ordered(cut(survey$Age, c(15, 25, 45, 65, 100)),
                      labels = c("Young", "Middle-aged", "Senior", "Old"))

survey$Hours.Per.Week <- ordered(cut(survey$Hours.Per.Week,c(0, 25, 40, 60, 168)),
                                 labels = c("Part-time", "Full-time", "Over-time",
"Workaholic"))
```

```
# Then map Capital.Gain and Capital.Loss to None,
# and Low and High according to the median with this code:

survey$Capital.Gain <- ordered(cut(survey$Capital.Gain,
                                   c(-Inf, 0,
median(survey$Capital.Gain[survey$Capital.Gain >0]), 1e+06)),
                                 labels = c("None", "Low", "High"))

survey$Capital.Loss <- ordered(cut(survey$Capital.Loss,
                                     c(-Inf, 0,
median(survey$Capital.Loss[survey$Capital.Loss >0]), 1e+06)),
                                  labels = c("None", "Low", "High"))

# Now we are finished with the preparation of the
# data for the apriori() function. Take another look
# at the first five records:
# <fill in code>

# Make sure you have the arules package loaded.
# The apriori() function will coerce the data into the
# transactions data type, and this can also be done
# prior to calling apriori using the as function to
# view the data as a transaction dataset:

# <fill in code to load arules package>
survey.transactions <-  as(survey, "transactions")
survey.transactions

# This illustrates how the transactions data type
# represents variables in a binary form, one binary
```

```
# variable for each level of each categoric variable.
# There are 115 distinct levels (values for the categoric
# variables) across all 13 of the categoric variables.

# Use the summary function to extract more details:

# < fill in code to apply summary against 'survey.transactions'

# The summary begins with a description of the dataset sizes.
# This is followed by a list of the most frequent items
# occurring in the dataset. A Capital.Loss of None is the
# single most frequent item, occurring 31,042 times
# (i.e., pretty much no transaction has any capital loss
# recorded). The length distribution of the transactions
# is then given, indicating that some transactions have NA's
# for some of the variables. Looking at the summary of the
# original dataset you'll see that the variables Workclass,
# Occupation, and Native.Country have NA's, and so the
# distribution ranges from 10 to 13 items in a transaction.

# The final piece of information in the summary output indicates
# the mapping that has been used to map the categoric variables
# to the binary variables, so that Age = Young is one binary
# variable, and Age = Middle-aged is another.

# Now it is time to find all association rules using apriori.
# Assign them to the variable "survey.rules"
# Use a support of 0.05 and a confidence of 0.95.

# <fill in code to use apriori function with these parameters>
```

```
# This leaves us with a set of 4,236 rules.
survey.rules

# Run the summary function on the "survey.rules" object:
# <fill in code to run summary function>

# Inspect the first five rules:
# <fill in code to inspect first five rules>

# Subset, and then inspect, the first five rules
# with a lift greater than 2.5:
# <fill in code to subset first five rules with lift > 2.5

# <fill in code to inspect same>
```