

MIS 545 Lab03 R Script and Rapid Miner

1. RScript:

```
# Fnu Geetika, Vihari Reddy Tummuru
# MIS 545 01
# Lab03Group02GeetikaTummuru
# Import csv files, assign data types, subset datasets, calculate summary
# statistics, manipulate tibbles and prepare histogram and boxplot visualizations

# Install the tidyverse package
install.packages("tidyverse")

# Load the tidyverse library
library(tidyverse)

# Set the working directory to your Lab03 folder
setwd("C:/Users/uai-laptop/Documents/Lab03")

# Read GroceryTransactions.csv into a tibble called groceryTransactions1
groceryTransactions1 <- read_csv(file = "GroceryTransactions.csv",
                                col_types = "iDffffiffffffin",
                                col_names = TRUE)

# Display groceryTransactions1 in the console
print(groceryTransactions1)

# Display the first 20 rows of groceryTransactions1 in the console
head(groceryTransactions1, n=20)
```

```
# Display the structure of groceryTransactions1 in the console
str(groceryTransactions1)

# Display the summary of groceryTransactions1 in the console
summary(groceryTransactions1)

# Use the dplyr summarize() function to display the following on the console
# Mean of revenue
print(summarize(.data = groceryTransactions1, mean(Revenue)))

#Median of units sold
print(summarize(.data = groceryTransactions1, median(UnitsSold)))

# Standard deviation of revenue
print(summarize(.data = groceryTransactions1, sd(Revenue)))

# Inter-quartile range of units sold
print(summarize(.data = groceryTransactions1, IQR(UnitsSold)))

# Minimum of revenue
print(summarize(.data = groceryTransactions1, min(Revenue)))

# Maximum of children
print(summarize(.data = groceryTransactions1, max(Children)))

# Create a new tibble called groceryTransactions2 that contains only the
# following columns
# PurchaseDate
# Homeowner
# Children
```

```

# AnnualIncome
# UnitsSold
# Revenue
groceryTransactions2 <- select(.data = groceryTransactions1,
                               PurchaseDate,
                               Homeowner,
                               Children,
                               AnnualIncome,
                               UnitsSold,
                               Revenue)

# Display all of the features in groceryTransactions2 for transactions made by
# non-homeowners with at least 4 children
print(filter(.data = groceryTransactions2,
             Homeowner == "N" & Children >= 4))

# Display all of the records and features in groceryTransactions2 that were
# either made by customers in the $150K + annual income category OR had more
# than 6units sold
print(filter(.data = groceryTransactions2,
             AnnualIncome == "$150K +" | UnitsSold > 6))

# Display the average transaction revenue grouped by annual income level
# Sort the results by average transaction revenue from largest to smallest
print(groceryTransactions2 %>%
      group_by(AnnualIncome) %>%
      summarize(AverageRevenue = mean(Revenue)) %>%
      arrange(desc(AverageRevenue)),
      n = Inf)

```

```

# Create a new tibble called groceryTransactions3 that contains all of the
# features in groceryTransactions2 along with a new calculated feature called
# AveragePricePerUnit calculated by dividing revenue by units sold
groceryTransactions3 <- groceryTransactions2 %>%
  mutate(AveragePricePerUnit = Revenue / UnitsSold)

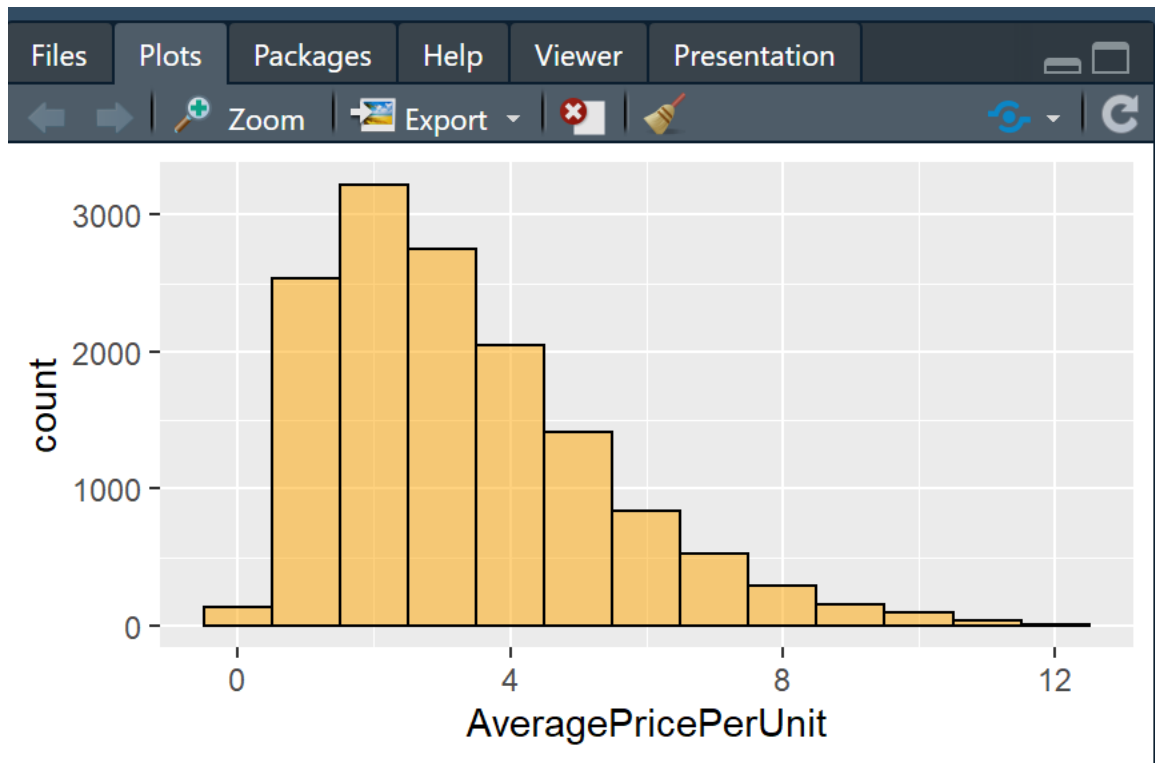
# Display the groceryTransactions3 tibble on the console
print(groceryTransactions3)

# Use ggplot() to create a histogram of AveragePricePerUnit with a bin width of
# 1, a bin outline of black, a bin fill of orange, and a bin transparency of 50%
histogram <- ggplot(data = groceryTransactions3,
  aes(x = AveragePricePerUnit))
histogram + geom_histogram(binwidth = 1,
  color = "black",
  fill = "orange",
  alpha = 0.5)

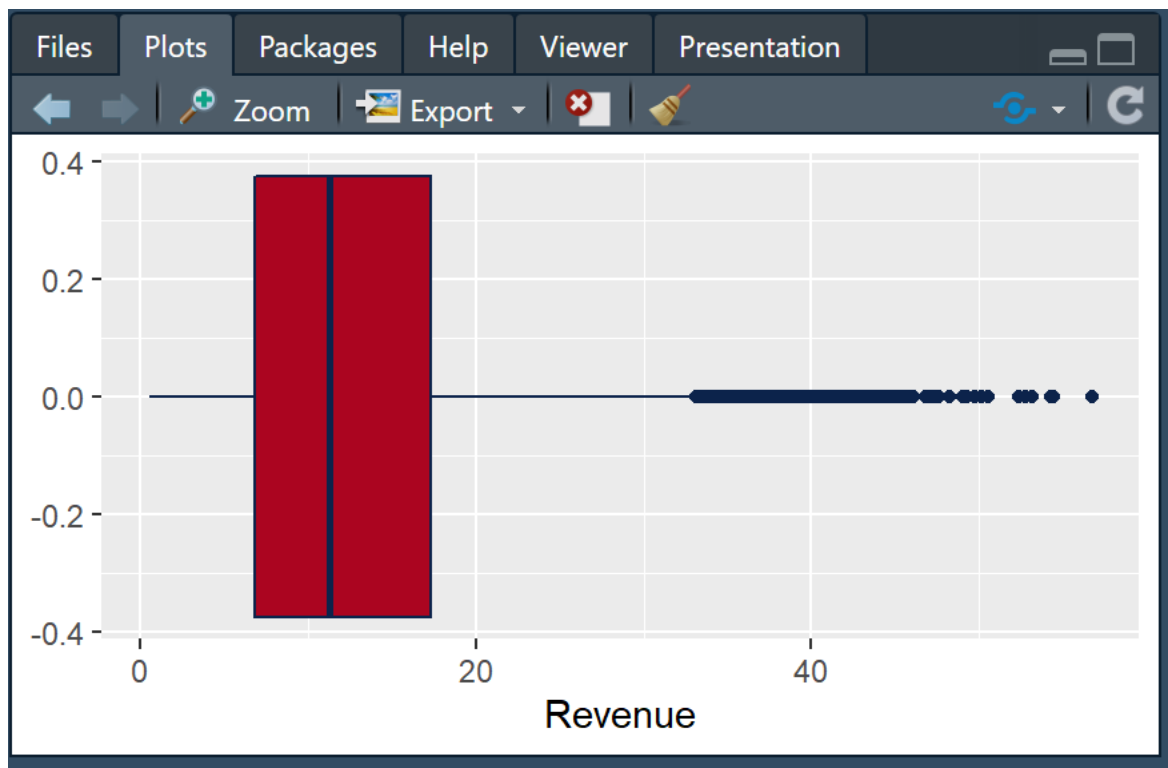
# Use ggplot() to create a boxplot of revenue with an outline color of Arizona
# Blue (#0C234B) and a fill color of Arizona Red (#AB0520)
boxplot <- ggplot(data = groceryTransactions3,
  aes(x = Revenue))
boxplot + geom_boxplot(color = "#0C234B",
  fill = "#AB0520")

```

➤ Histogram of AveragePricePerUnit:

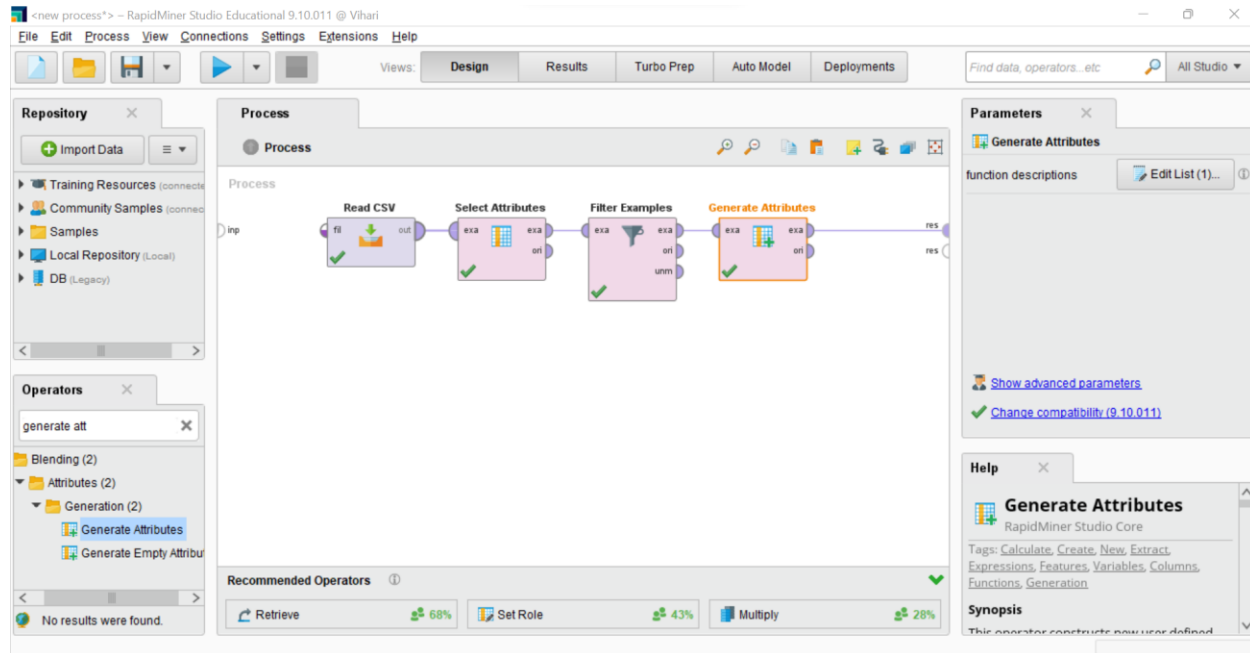


➤ Boxplot of revenue:



2. Rapid Miner:

➤ Process:



The screenshot shows the RapidMiner Studio interface in the 'Process' view. The main canvas displays a workflow with four operators: 'Read CSV', 'Select Attributes', 'Filter Examples', and 'Generate Attributes'. The 'Generate Attributes' operator is highlighted in orange. The left sidebar contains the 'Repository' and 'Operators' panels. The 'Repository' panel shows a tree structure with 'Training Resources', 'Community Samples', 'Samples', 'Local Repository (Local)', and 'DB (Legacy)'. The 'Operators' panel shows a search for 'generate att' with results for 'Blending (2)', 'Attributes (2)', and 'Generation (2)'. The 'Parameters' panel on the right shows the 'Generate Attributes' operator's parameters, including 'function descriptions' and 'Edit List (1)'. The 'Help' panel at the bottom right provides information about the 'Generate Attributes' operator, including tags and a synopsis.

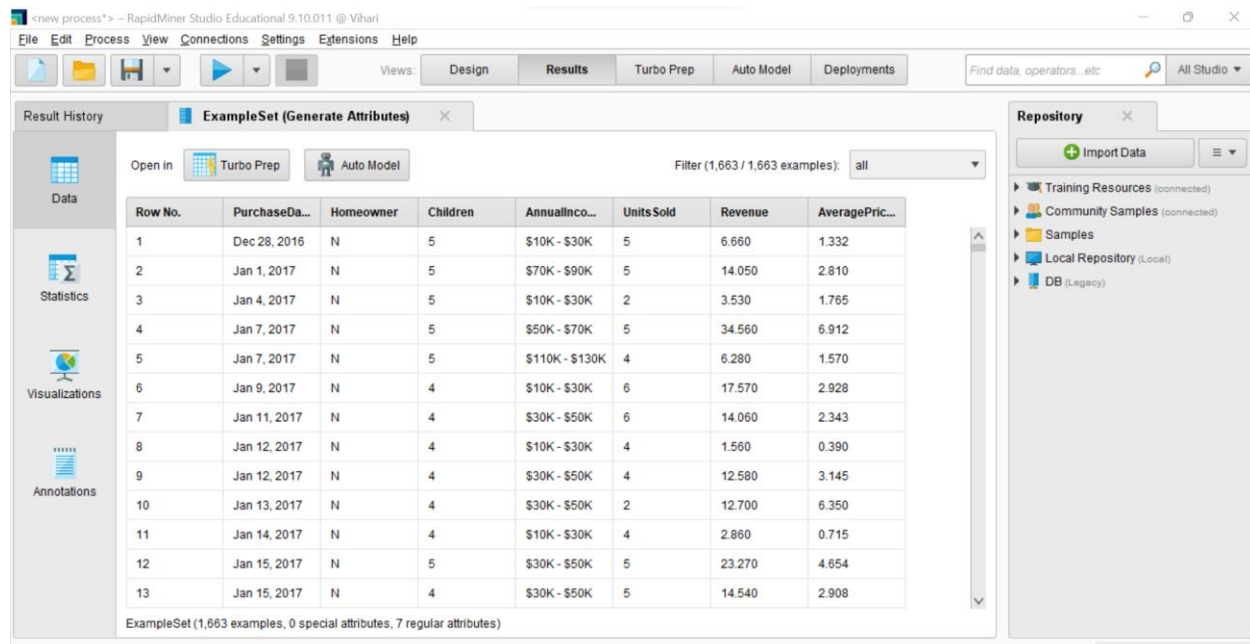
Process

Read CSV → Select Attributes → Filter Examples → Generate Attributes

Recommended Operators

- Retrieve (68%)
- Set Role (43%)
- Multiply (28%)

➤ Results:



The screenshot shows the RapidMiner Studio interface in the 'Results' view. The main canvas displays a table of results for the 'ExampleSet (Generate Attributes)' operator. The table has 13 rows and 8 columns: 'Row No.', 'PurchaseDa...', 'Homeowner', 'Children', 'AnnualInco...', 'Units Sold', 'Revenue', and 'AveragePric...'. The 'Filter' dropdown is set to 'all'. The left sidebar contains the 'Result History' panel with icons for 'Data', 'Statistics', 'Visualizations', and 'Annotations'. The 'Repository' panel on the right shows the same tree structure as in the process view.

Result History

ExampleSet (Generate Attributes)

Open in: Turbo Prep, Auto Model

Filter (1,663 / 1,663 examples): all

Row No.	PurchaseDa...	Homeowner	Children	AnnualInco...	Units Sold	Revenue	AveragePric...
1	Dec 28, 2016	N	5	\$10K - \$30K	5	6.660	1.332
2	Jan 1, 2017	N	5	\$70K - \$90K	5	14.050	2.810
3	Jan 4, 2017	N	5	\$10K - \$30K	2	3.530	1.765
4	Jan 7, 2017	N	5	\$50K - \$70K	5	34.560	6.912
5	Jan 7, 2017	N	5	\$110K - \$130K	4	6.280	1.570
6	Jan 9, 2017	N	4	\$10K - \$30K	6	17.570	2.928
7	Jan 11, 2017	N	4	\$30K - \$50K	6	14.060	2.343
8	Jan 12, 2017	N	4	\$10K - \$30K	4	1.560	0.390
9	Jan 12, 2017	N	4	\$30K - \$50K	4	12.580	3.145
10	Jan 13, 2017	N	4	\$30K - \$50K	2	12.700	6.350
11	Jan 14, 2017	N	4	\$10K - \$30K	4	2.860	0.715
12	Jan 15, 2017	N	5	\$30K - \$50K	5	23.270	4.654
13	Jan 15, 2017	N	4	\$30K - \$50K	5	14.540	2.908

ExampleSet (1,663 examples, 0 special attributes, 7 regular attributes)

➤ Statistical Histogram:

