INFX 573 Problem Set 8 - Classification

Charudatta Deshpande

Due: Thursday, December 7, 2017

Introduction

Collaborators: Charles Hemstreet, Robert Hinshaw, Ram Ganesan, Manjiri Kharkar

Instructions:

2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. List all collaborators on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps7.Rmd`, knit a PDF and submit the PDF file on Canvas.

Data

You will be using credit card application data (on canvas). This originates from a confidential source, and all variable names are removed. The only variable you have to know is A16: approval (+) or refusal (-). The data is downloaded from UCI Machine Learning Repo, more information is in the meta file.

Task

Your task is to predict the approval or disapproval using logistic regression and decision trees, and compare the performance of these methods.

```
# Load some helpful libraries
library(tidyverse)
library(data.table)
library(mosaic)
library(rpart)
credit <- read.csv("credit_card_applications.csv")
#transform 'A16' - make it '1' for approval and '0' for refusal.
credit$A16 <-ifelse(credit$A16=="+", 1, 0)
#transform 'A16' to categorical
credit$A16 <- factor(credit$A16)
```

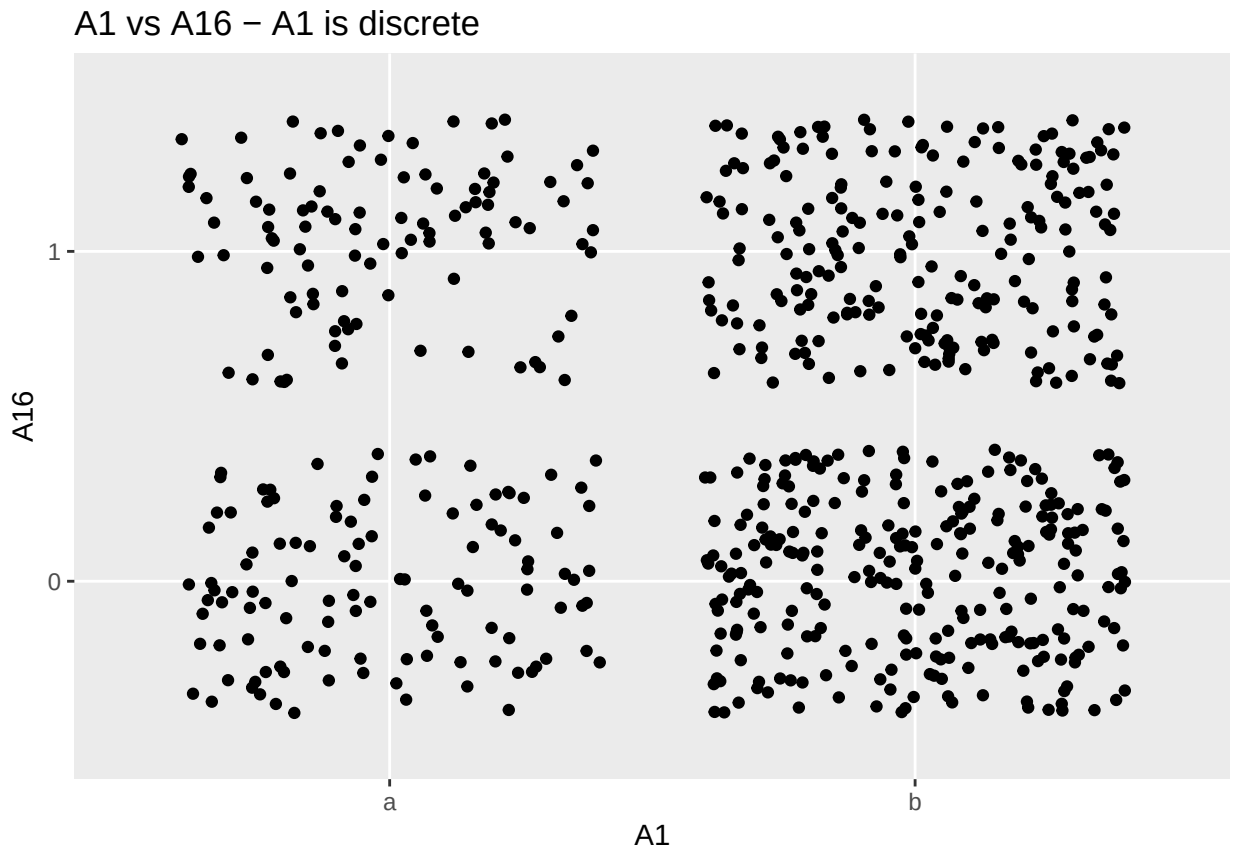
1. Select variables

Select some variables. As we don't know the meaning of the variables, you have just to use cross-tables, scatter plots, trial-and-error to find good predictors of A16.

Answer -

We will plot some variables against A16 to check what variables could be used as good predictors of approval/refusal. Based on metadata file, use an appropriate plotting method based on if predictor variable is discrete or continuous. Filter '?' values. Also we will calculate chi square value for columns with A16 and try to determine best fit for analysis.

```
# A1 Plot
credit %>% filter(A1 != "?") %>%
ggplot(aes(A1, A16)) +
  geom_jitter() +
  labs(title = "A1 vs A16 - A1 is discrete")
```

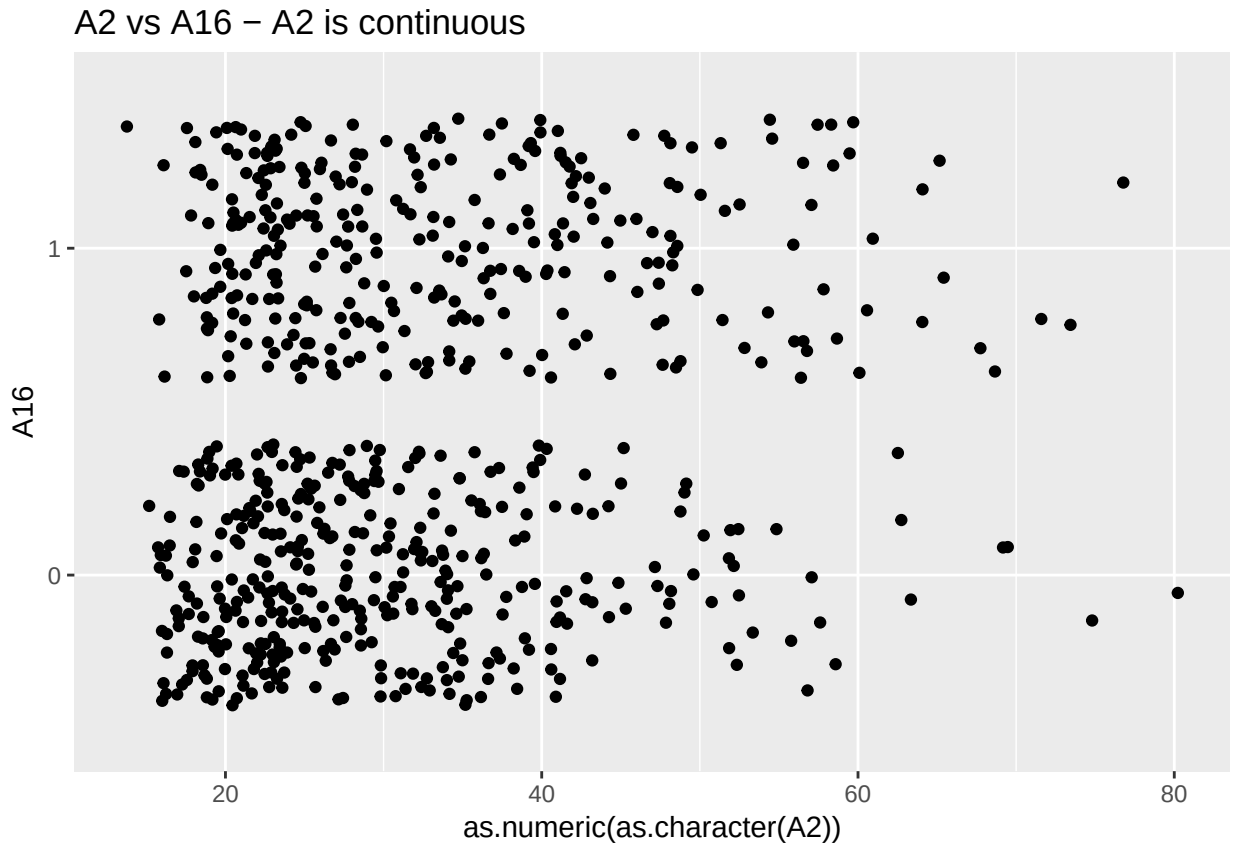



```
# A1 chi-square test
chisq.test(credit$A1, credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit$A1 and credit$A16
## X-squared = 2.291, df = 2, p-value = 0.3181
```

```
# A2 - since A2 is continuous, we will use geom_jitter and convert A2 to
# numeric. That will enable us to see A2 as continuous.
```

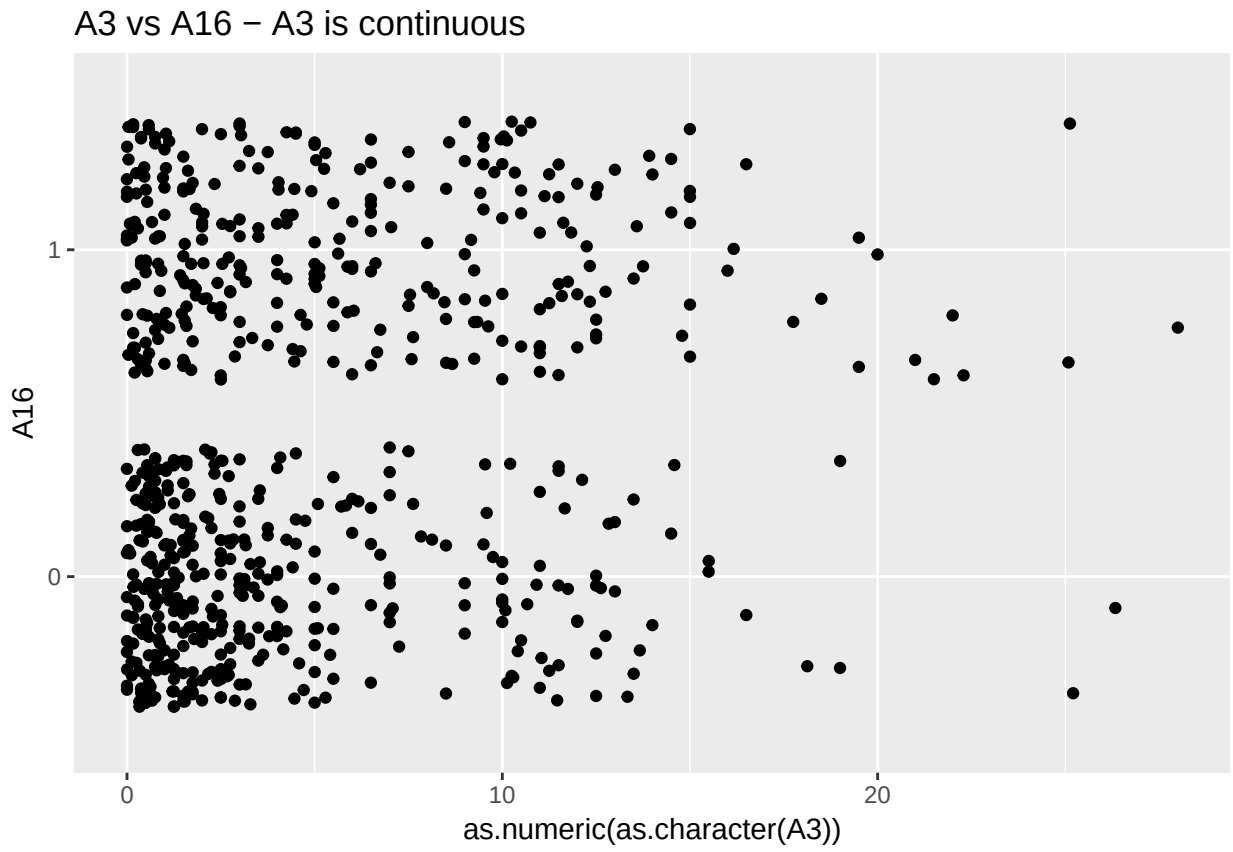
```
credit %>% filter(A2 != "?") %>%
ggplot(aes(as.numeric(as.character(A2)), A16)) +
      geom_jitter() +
      labs(title = "A2 vs A16 - A2 is continuous")
```




```
# A2 chi-square test
chisq.test(as.numeric(as.character(credit$A2)), credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  as.numeric(as.character(credit$A2)) and credit$A16
## X-squared = 375.88, df = 348, p-value = 0.1457
```

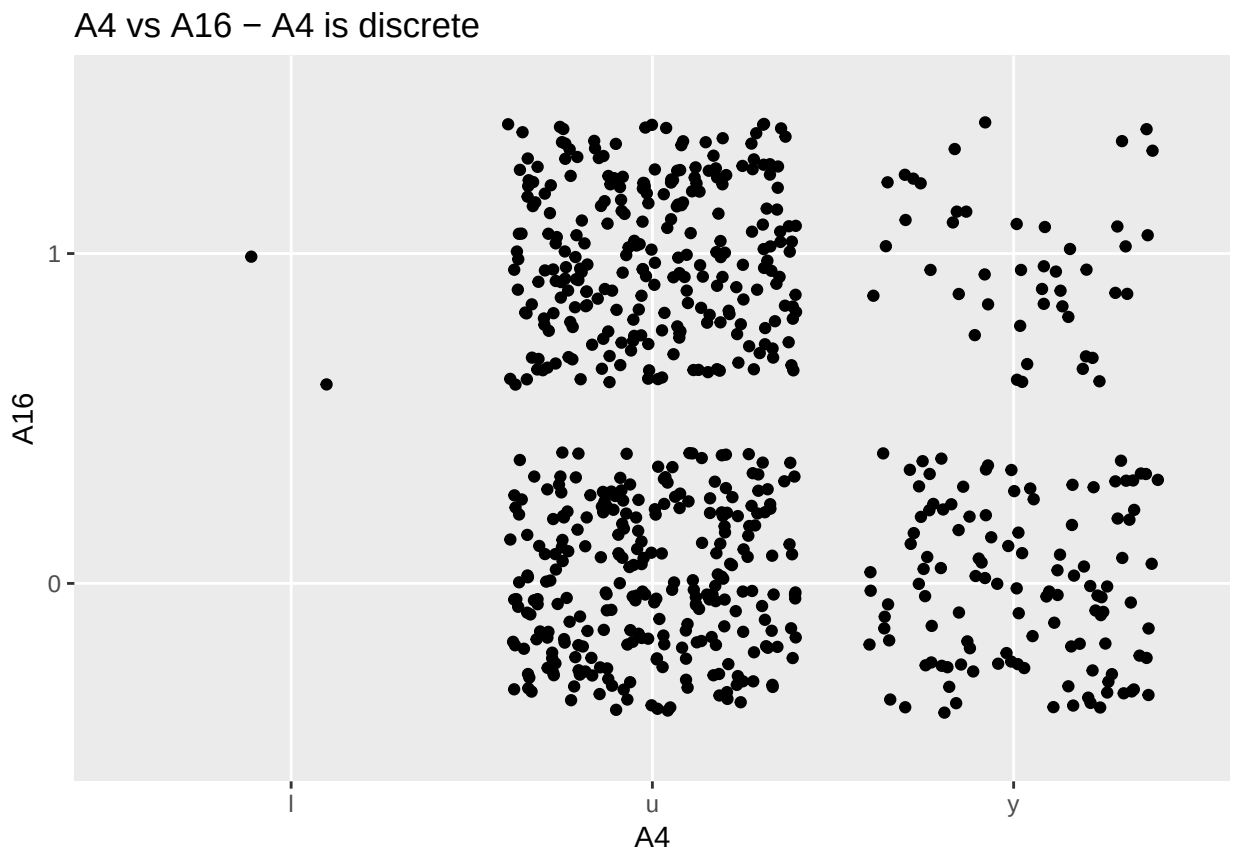
```
# A3 - since A3 is continuous, we will use geom_jitter and convert A3 to
# numeric. That will enable us to see A3 as continuous.
credit %>% filter(A3 != "?") %>%
ggplot(aes(as.numeric(as.character(A3)), A16)) +
      geom_jitter() +
      labs(title = "A3 vs A16 - A3 is continuous")
```




```
# A3 chi-square test
chisq.test(as.numeric(as.character(credit$A3)), credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  as.numeric(as.character(credit$A3)) and credit$A16
## X-squared = 239.05, df = 214, p-value = 0.1154
```

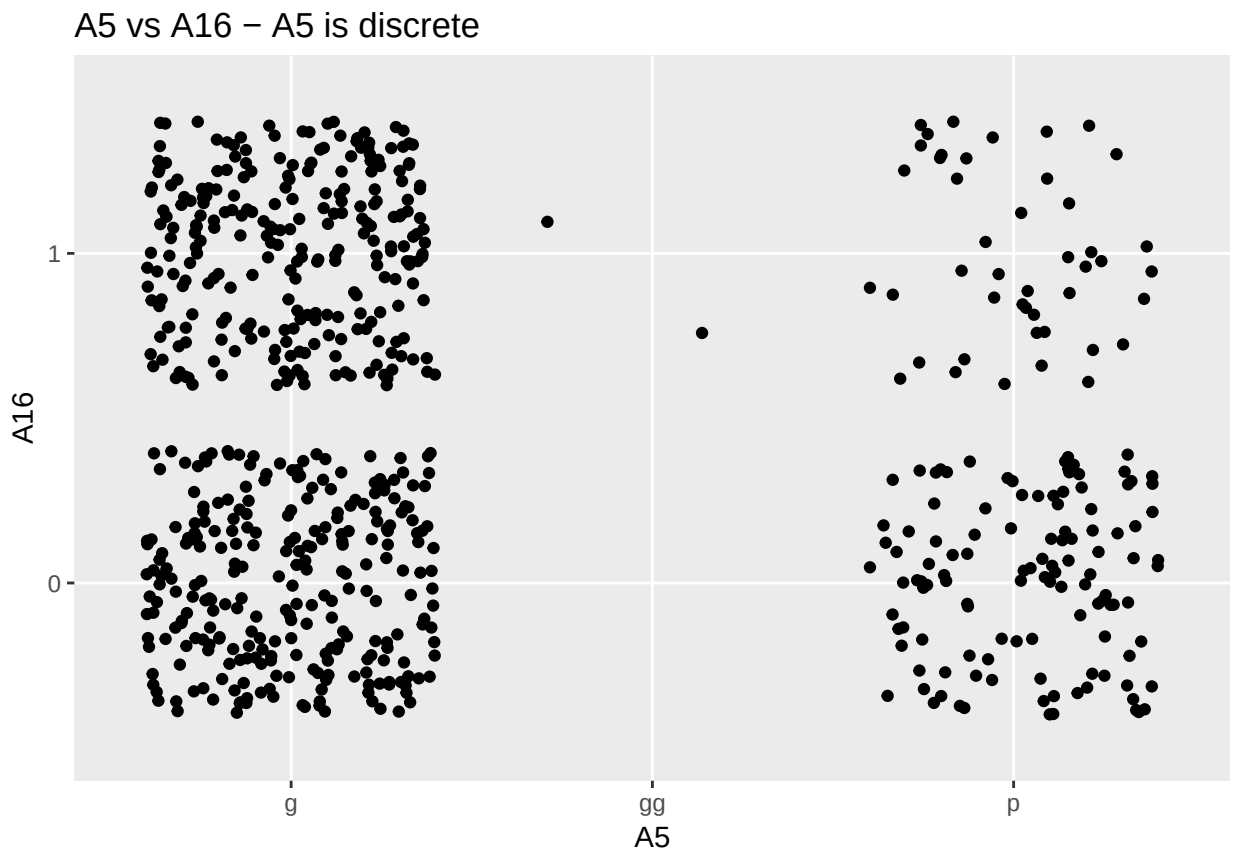
```
# A4 Plot
credit %>% filter(A4 != "?") %>%
ggplot(aes(A4, A16)) +
  geom_jitter() +
  labs(title = "A4 vs A16 - A4 is discrete")
```




```
# A4 chi-square test
chisq.test(credit$A4, credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit$A4 and credit$A16
## X-squared = 27.416, df = 3, p-value = 4.816e-06
```

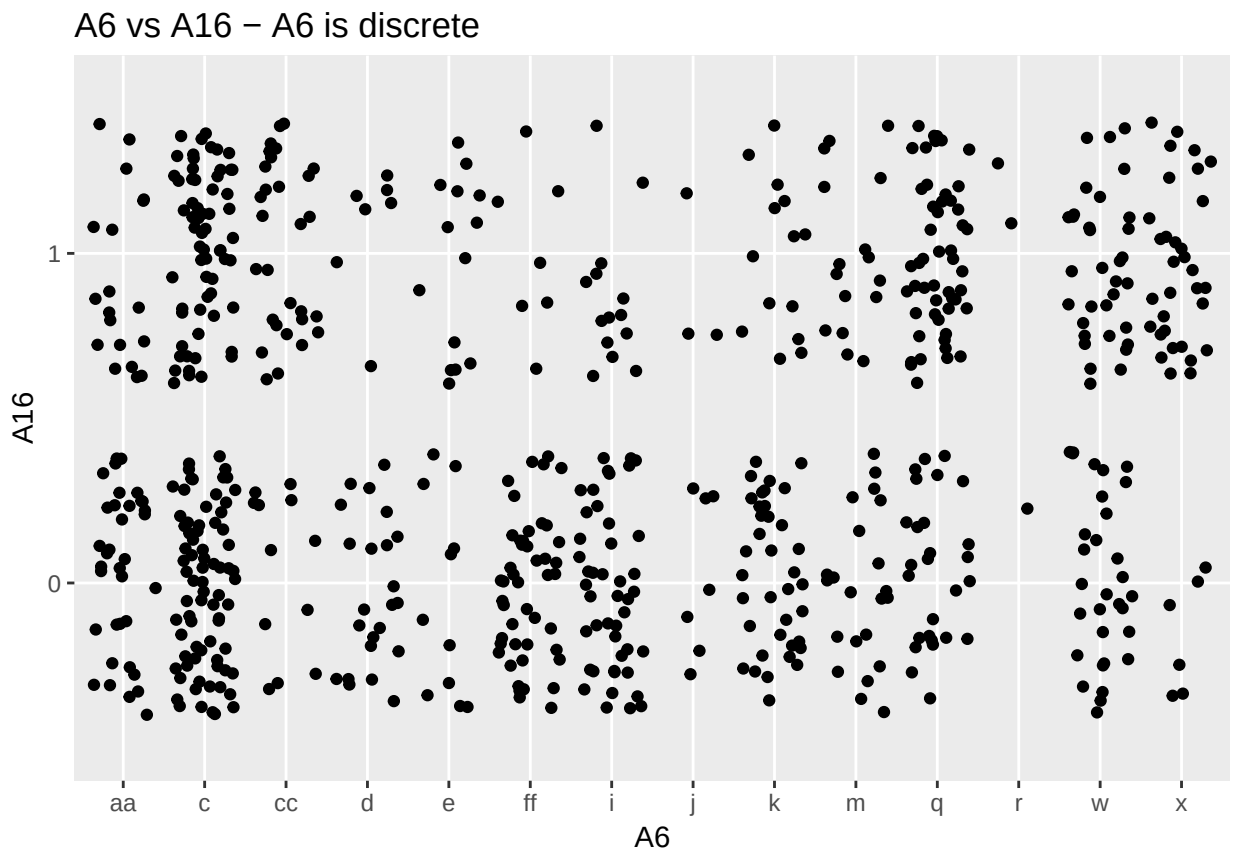
```
# A5 Plot
credit %>% filter(A5 != "?") %>%
ggplot(aes(A5, A16)) +
  geom_jitter() +
  labs(title = "A5 vs A16 - A5 is discrete")
```




```
# A5 chi-square test
chisq.test(credit$A5, credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit$A5 and credit$A16
## X-squared = 27.416, df = 3, p-value = 4.816e-06
```

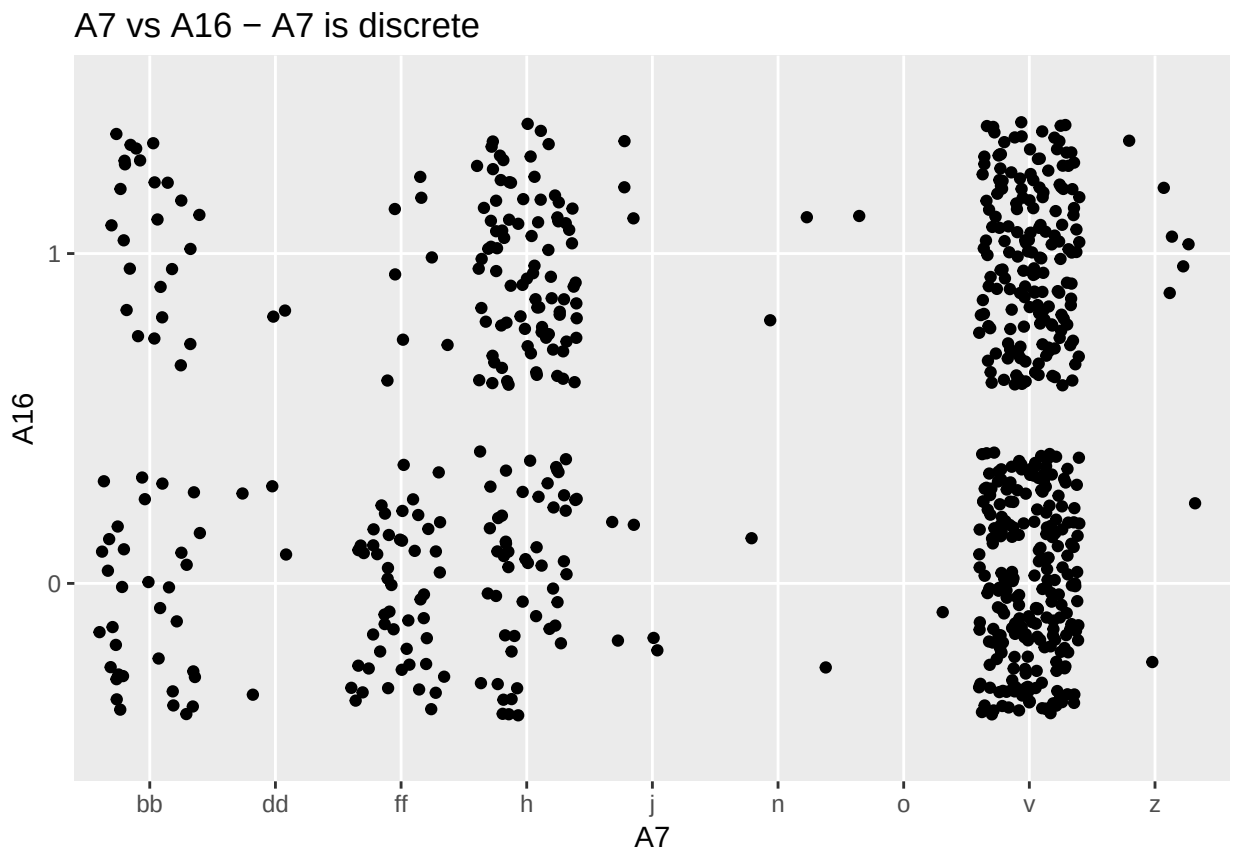
```
# A6 Plot
credit %>% filter(A6 != "?") %>%
ggplot(aes(A6, A16)) +
  geom_jitter() +
  labs(title = "A6 vs A16 - A6 is discrete")
```




```
# A6 chi-square test
chisq.test(credit$A6, credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit$A6 and credit$A16
## X-squared = 98.325, df = 14, p-value = 9.921e-15
```

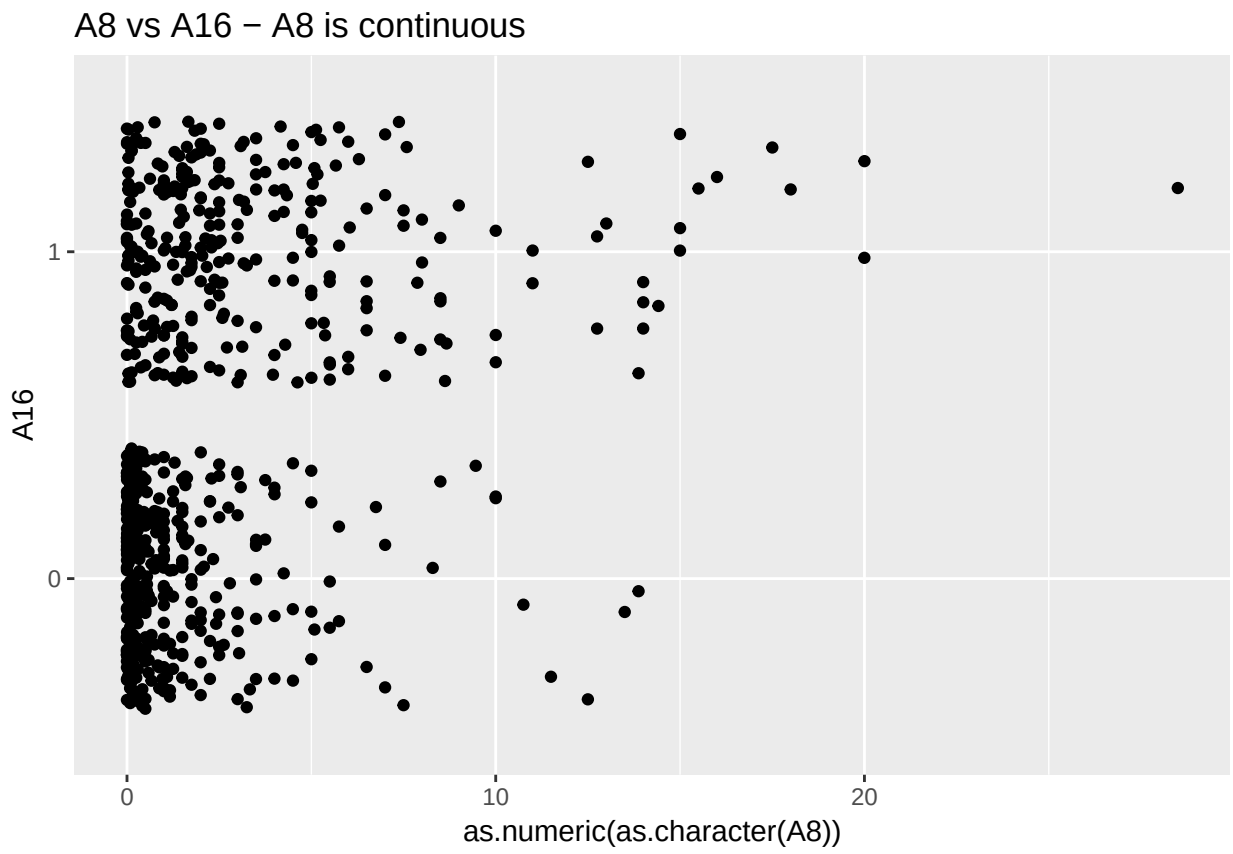
```
# A7 Plot
credit %>% filter(A7 != "?") %>%
ggplot(aes(A7, A16)) +
  geom_jitter() +
  labs(title = "A7 vs A16 - A7 is discrete")
```




```
# A7 chi-square test
chisq.test(credit$A7, credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit$A7 and credit$A16
## X-squared = 45.034, df = 9, p-value = 9.093e-07
```

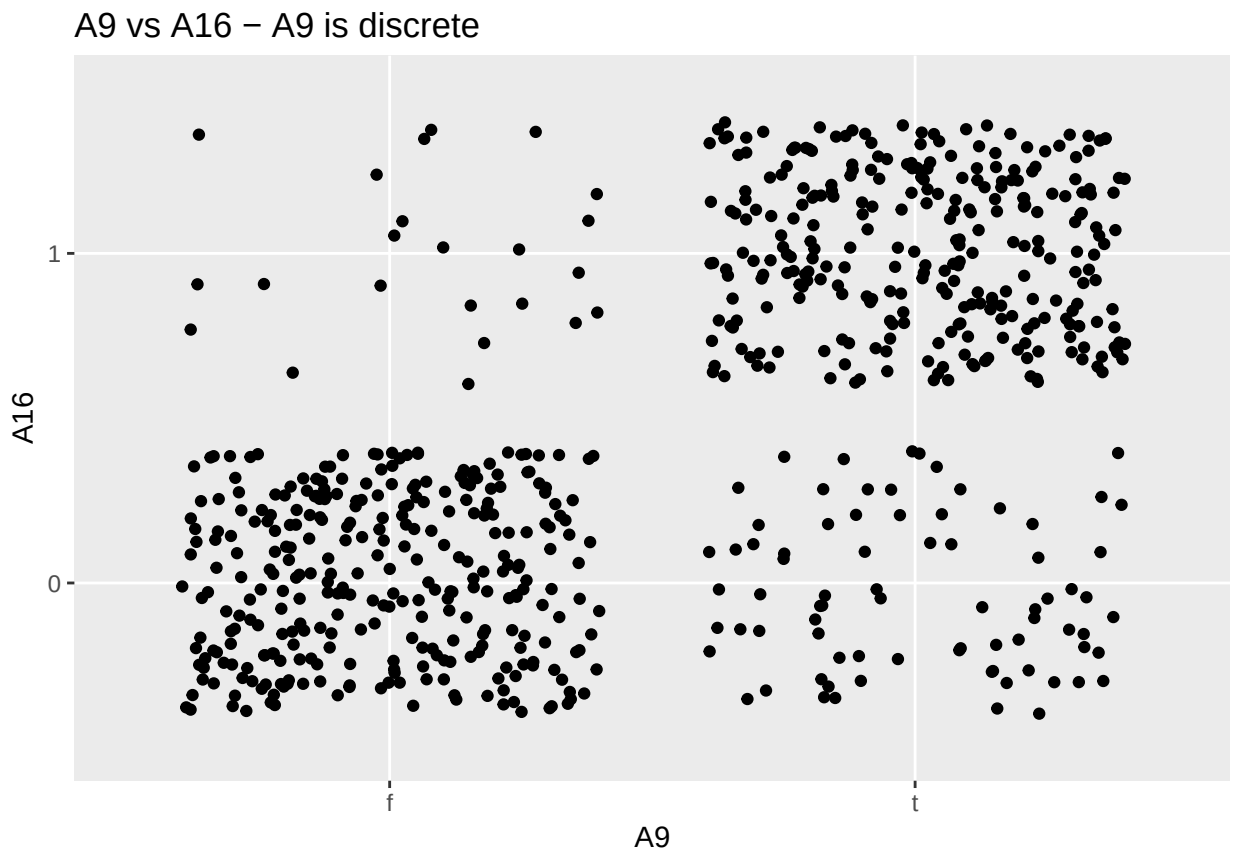
```
# A8 - since A8 is continuous, we will use geom_jitter and convert A8 to
# numeric. That will enable us to see A8 as continuous.
credit %>% filter(A8 != "?") %>%
ggplot(aes(as.numeric(as.character(A8)), A16)) +
      geom_jitter() +
      labs(title = "A8 vs A16 - A8 is continuous")
```




```
# A8 chi-square test
chisq.test(as.numeric(as.character(credit$A8)), credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  as.numeric(as.character(credit$A8)) and credit$A16
## X-squared = 214.15, df = 131, p-value = 6.17e-06
```

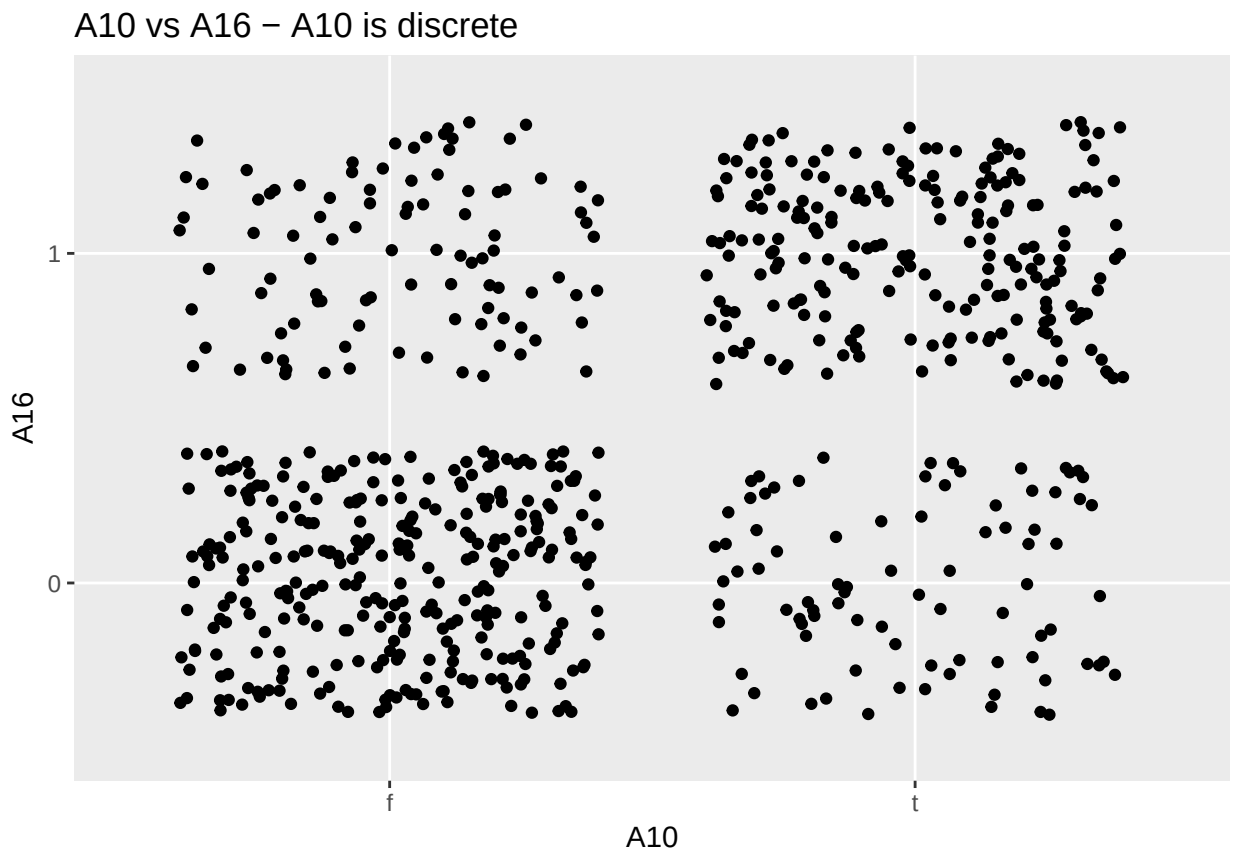
```
# A9 Plot
credit %>% filter(A9 != "?") %>%
ggplot(aes(A9, A16)) +
  geom_jitter() +
  labs(title = "A9 vs A16 - A9 is discrete")
```




```
# A9 chi-square test
chisq.test(credit$A9, credit$A16)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit$A9 and credit$A16
## X-squared = 355.2, df = 1, p-value < 2.2e-16
```

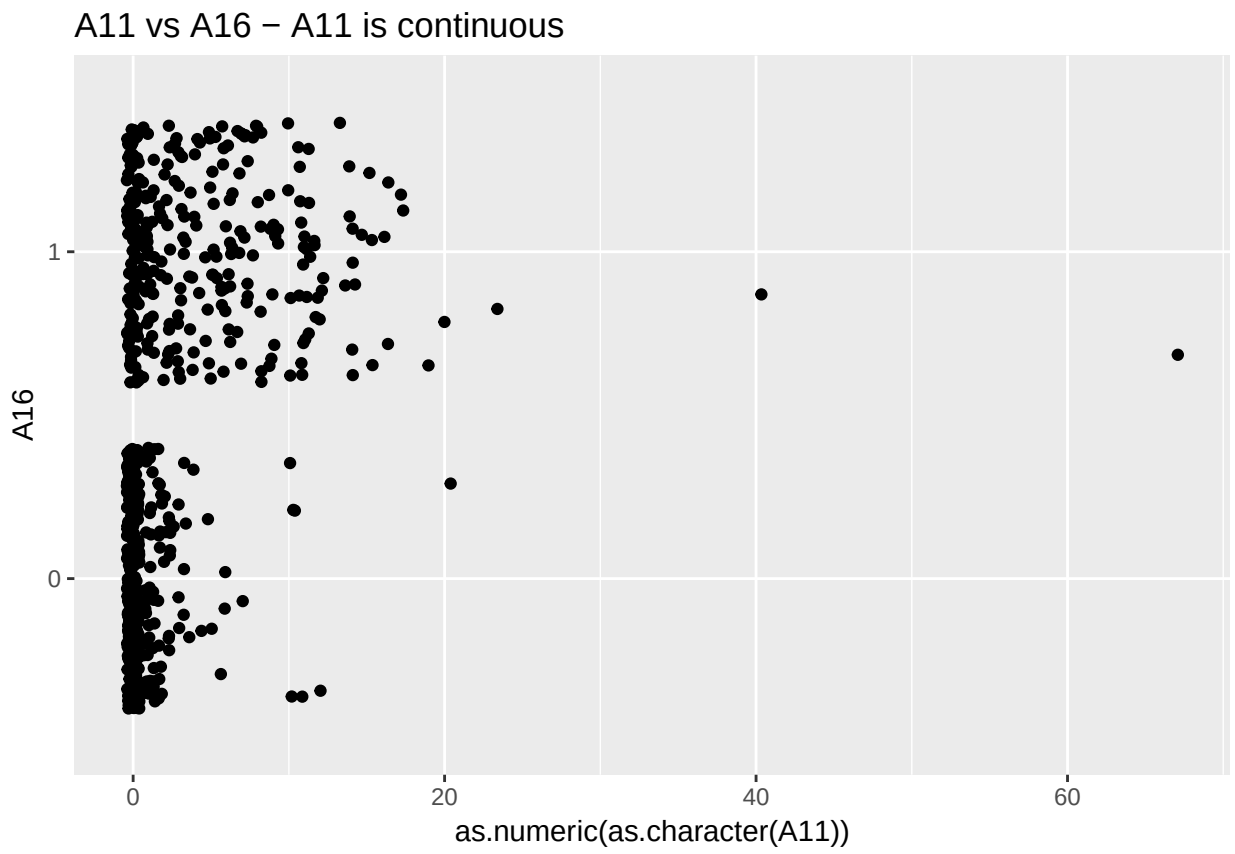
```
# A10 Plot
credit %>% filter(A10 != "?") %>%
ggplot(aes(A10, A16)) +
  geom_jitter() +
  labs(title = "A10 vs A16 - A10 is discrete")
```




```
# A10 chi-square test
chisq.test(credit$A10, credit$A16)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit$A10 and credit$A16
## X-squared = 143.07, df = 1, p-value < 2.2e-16
```

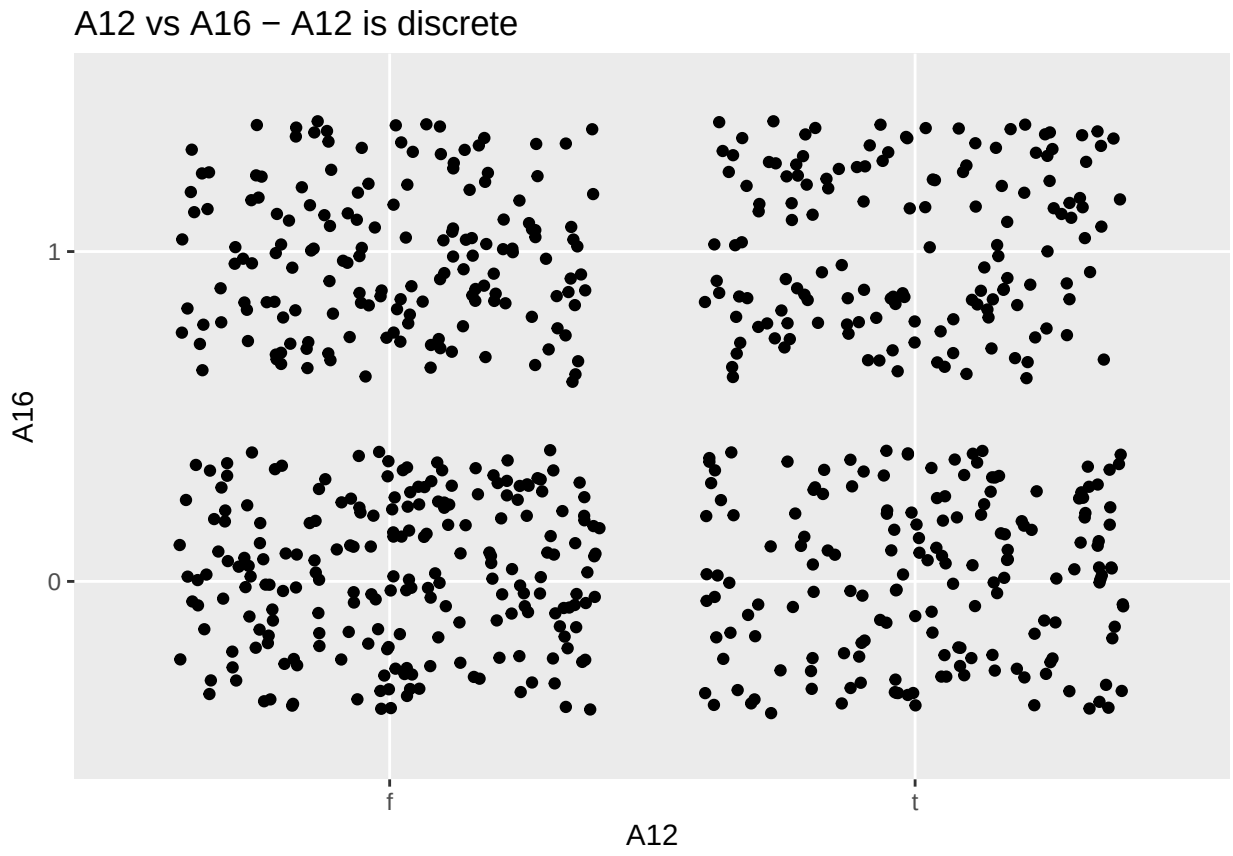
```
# A11 - since A11 is continuous, we will use geom_jitter and convert A11 to
# numeric. That will enable us to see A11 as continuous.
credit %>% filter(A11 != "?") %>%
ggplot(aes(as.numeric(as.character(A11)), A16)) +
      geom_jitter() +
      labs(title = "A11 vs A16 - A11 is continuous")
```




```
# A11 chi-square test
chisq.test(as.numeric(as.character(credit$A11)), credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  as.numeric(as.character(credit$A11)) and credit$A16
## X-squared = 203.41, df = 22, p-value < 2.2e-16
```

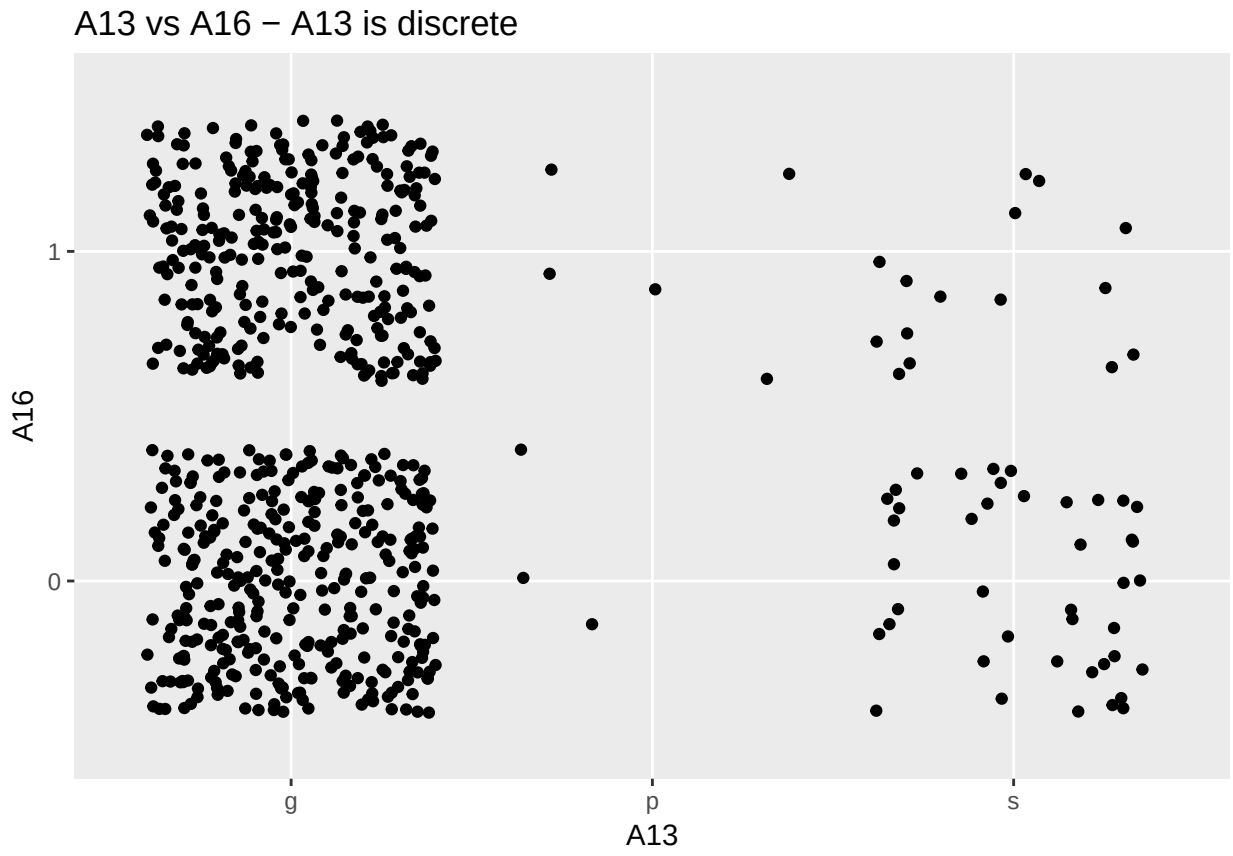
```
# A12 Plot
credit %>% filter(A12 != "?") %>%
ggplot(aes(A12, A16)) +
  geom_jitter() +
  labs(title = "A12 vs A16 - A12 is discrete")
```




```
# A12 chi-square test
chisq.test(credit$A12, credit$A16)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit$A12 and credit$A16
## X-squared = 0.56827, df = 1, p-value = 0.4509
```

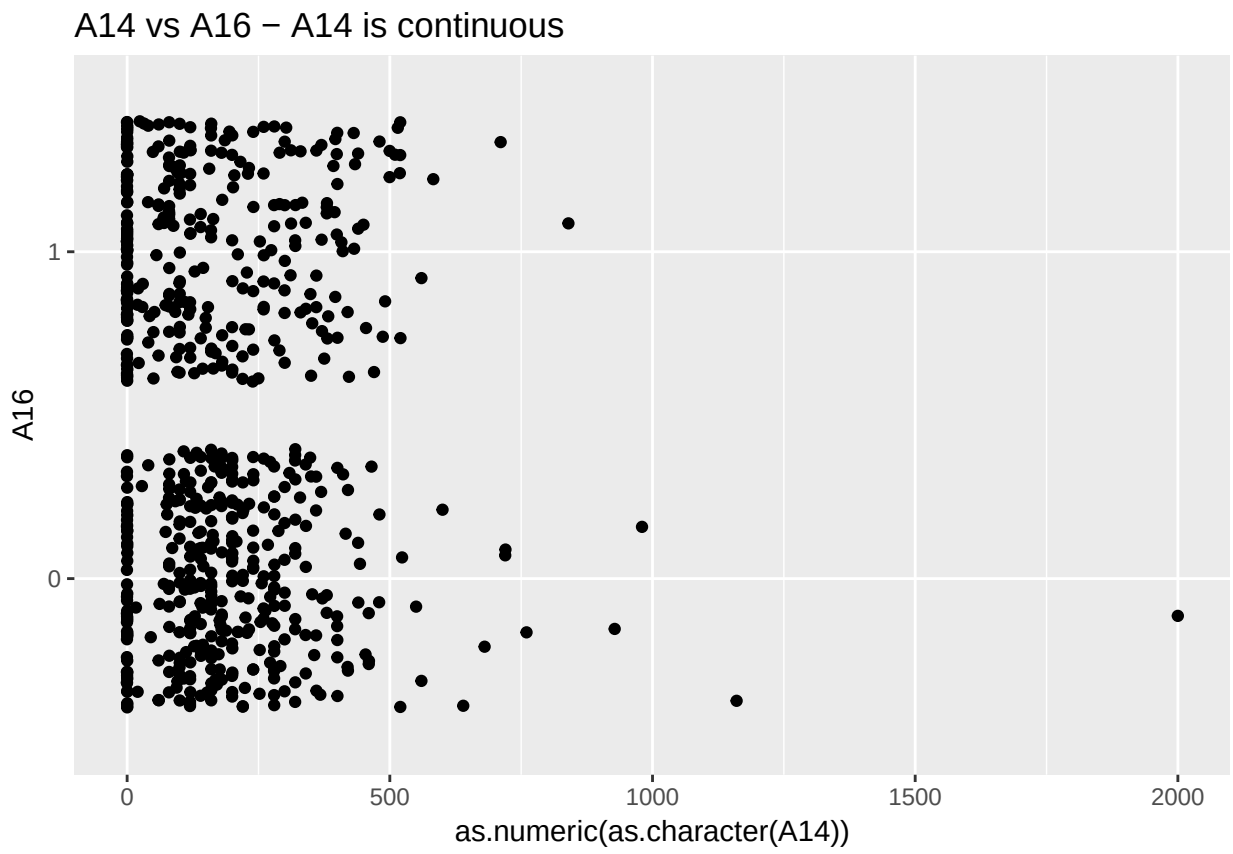
```
# A13 Plot
credit %>% filter(A13 != "?") %>%
ggplot(aes(A13, A16)) +
  geom_jitter() +
  labs(title = "A13 vs A16 - A13 is discrete")
```




```
# A13 chi-square test
chisq.test(credit$A13, credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit$A13 and credit$A16
## X-squared = 9.1916, df = 2, p-value = 0.01009
```

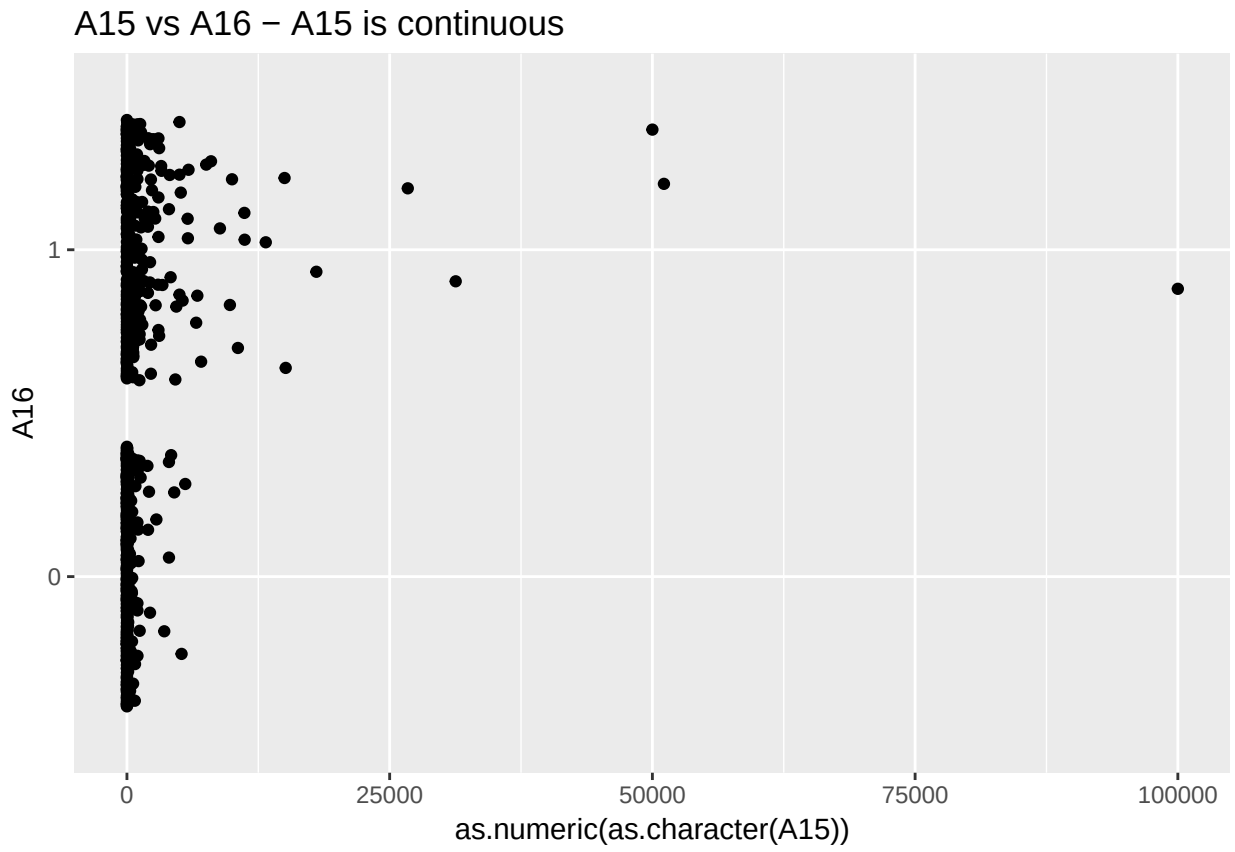
```
# A14 - since A14 is continuous, we will use geom_jitter and convert A14 to
# numeric. That will enable us to see A14 as continuous.
credit %>% filter(A14 != "?") %>%
ggplot(aes(as.numeric(as.character(A14)), A16)) +
      geom_jitter() +
      labs(title = "A14 vs A16 - A14 is continuous")
```




```
# A14 chi-square test
chisq.test(as.numeric(as.character(credit$A14)), credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  as.numeric(as.character(credit$A14)) and credit$A16
## X-squared = 219.39, df = 169, p-value = 0.005489
```

```
# A15 - since A15 is continuous, we will use geom_jitter and convert A15 to
# numeric. That will enable us to see A15 as continuous.
credit %>% filter(A15 != "?") %>%
ggplot(aes(as.numeric(as.character(A15)), A16)) +
      geom_jitter() +
      labs(title = "A15 vs A16 - A15 is continuous")
```




```
# A15 chi-square test
chisq.test(as.numeric(as.character(credit$A15)), credit$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  as.numeric(as.character(credit$A15)) and credit$A16
## X-squared = 314.94, df = 239, p-value = 0.0007081
```

Based on these results, we will note that following set of variables could be good predictors for A16 (approval or refusal) -

A4 - p-value = 4.816e-06
A5 - p-value = 4.816e-06 - seems to provide exact same results as A4
A6 - p-value = 9.921e-15
A7 - p-value = 9.093e-07
A8 - p-value = 6.17e-06
A9 - p-value < 2.2e-16
A10 - p-value < 2.2e-16
A11 - p-value < 2.2e-16
A13 - p-value = 0.01009
A14 - p-value = 0.005489
A15 - p-value = 0.0007081

Based on these results, we will choose **A9, A10 and A11** as our first analysis set, since these seem to indicate the highest correlation with A16, and are likely the best candidate for predicting A16.

2. Estimate logistic regression

Use these variables to estimate logistic regression models. You may use the function `glm` in the base package, or any other implementation of logistic regression. Use this model to predict the outcome. Make a cross-table of actual/predicted outcomes. Which percentage did you get right?

```
# Logistic regression of A16 using A9, A10 and A11.
glm1 <- glm(A16 ~ A9 + A10 + A11, data=credit,
          family=binomial(link="logit"))
# Predict the outcomes using model
glm.result1 <- predict(glm1, type="response")
# A review of glm.result1 indicates that the output is non-binary.
# Making a cross table of actual and predicted values is not
# useful. We will use following condition to round the output to
# either 0 or 1. A value greater than or equal to 0.5 will round to 1.
# A value of less than 0.5 will round to 0.
glm.result1 <- ifelse(predict(glm1, type="response")>= 0.5, 1, 0)
print(table(credit$A16, glm.result1))
```

```
##    glm.result1
##       0   1
##   0 306  77
##   1  23 284
```

```
success.rate.glm1 <- table(credit$A16, glm.result1) %>% diag() %>% sum()/nrow(credit)
success.rate.glm1
```

```
## [1] 0.8550725
```

3. Estimate decision trees.

Use exactly the same variables to compute decision tree models. You may use function `rpart` in the `rpart` package, or any other decision tree implementations in R. As above, predict the result, make a cross-table, and find the correct percentage.

```
# Decision tree modeling of A16 using A9, A10 and A11.
tm1 <- rpart::rpart(A16 ~ A9 + A10 + A11, data=credit)
# Predict the outcomes using model
tree.result1 <- predict(tm1, type="class")
print(table(credit$A16, tree.result1))
```

```
##    tree.result1
##       0   1
##   0 306  77
##   1  23 284
```

```
success.rate.rpart1 <- table(credit$A16, tree.result1) %>% diag() %>% sum()/nrow(credit)
success.rate.rpart1
```

```
## [1] 0.8550725
```

As can be seen, the results from glm prediction and decision tree prediction are exactly same. The only difference is for glm, we had to round the outcome to either 0 or 1, but for decision tree model we got it by default.

4. Repeat the process

Repeat steps 1,2,3 with 3 different sets of variables. Feel free to do feature engineering.

Answer -

Step 1 has already been performed for all variables. Based on p-values calculated in Q1, we will choose next groups of variables for further analysis. We will repeat step 2 and 3 for following groups of variables -

Group 1 - A9, A10 and A11 (already done above)
Group 2 - A5, A6 and A7
Group 3 - A13, A14 and A15
Group 4 - A5, A6, A7, A9, A10 and A11 - combination of Group 1 and Group 2 above

Group 2 process -

```
# Group 2 Part 1 - Logistic Regression
# Logistic regression of A16 using A5, A6 and A7.
glm2 <- glm(A16 ~ A5 + A6 + A7, data=credit,
          family=binomial(link="logit"))
# Predict the outcomes using model
glm.result2 <- ifelse(predict(glm2, type="response")>= 0.5, 1, 0)
print(table(credit$A16, glm.result2))
```

```
##    glm.result2
##       0   1
##   0 299  84
##   1 131 176
```

```
success.rate.glm2 <- table(credit$A16, glm.result2) %>% diag() %>% sum()/nrow(credit)
success.rate.glm2
```

```
## [1] 0.6884058
```

```
# Group 2 Part 2 - Decision Tree
# Decision tree modeling of A16 using A5, A6 and A7.
tm2 <- rpart::rpart(A16 ~ A5 + A6 + A7, data=credit)
# Predict the outcomes using model
tree.result2 <- predict(tm2, type="class")
print(table(credit$A16, tree.result2))
```

```
##    tree.result2
##       0   1
##   0 275 108
##   1 115 192
```

```
success.rate.rpart2 <- table(credit$A16, tree.result2) %>% diag() %>% sum()/nrow(credit)
success.rate.rpart2
```

```
## [1] 0.6768116
```

Group 3 process -

```
# Group 3 Part 1 - Logistic Regression
# Logistic regression of A16 using A13, A14 and A15.
glm3 <- glm(A16 ~ A13 + A14 + A15, data=credit,
          family=binomial(link="logit"))
# Predict the outcomes using model
glm.result3 <- ifelse(predict(glm3, type="response")>= 0.5, 1, 0)
print(table(credit$A16, glm.result3))
```

```
##    glm.result3
##       0   1
##   0 302  81
##   1  89 218
```

```
success.rate.glm3 <- table(credit$A16, glm.result3) %>% diag() %>% sum()/nrow(credit)
success.rate.glm3
```

```
## [1] 0.7536232
```

```
# Group 3 Part 2 - Decision Tree
# Decision tree modeling of A16 using A13, A14 and A15.
tm3 <- rpart::rpart(A16 ~ A13 + A14 + A15, data=credit)
# Predict the outcomes using model
tree.result3 <- predict(tm3, type="class")
print(table(credit$A16, tree.result3))
```

```
##    tree.result3
##       0   1
##   0 319  64
##   1  69 238
```

```
success.rate.rpart3 <- table(credit$A16, tree.result3) %>% diag() %>% sum()/nrow(credit)
success.rate.rpart3
```

```
## [1] 0.8072464
```

Group 4 process -

```
# Group 4 Part 1 - Logistic Regression
# Logistic regression of A16 using A5, A6, A7, A9, A10 and A11.
glm4 <- glm(A16 ~ A5 + A6 + A7 + A9 + A10 + A11, data=credit,
          family=binomial(link="logit"))
# Predict the outcomes using model
glm.result4 <- ifelse(predict(glm4, type="response")>= 0.5, 1, 0)
print(table(credit$A16, glm.result4))
```

```
##    glm.result4
##       0   1
##   0 327  56
##   1  25 282
```

```
success.rate.glm4 <- table(credit$A16, glm.result4) %>% diag() %>% sum()/nrow(credit)
success.rate.glm4
```

```
## [1] 0.8826087
```

```
# Group 4 Part 2 - Decision Tree
# Decision tree modeling of A16 using A5, A6, A7, A9, A10 and A11.
tm4 <- rpart::rpart(A16 ~ A5 + A6 + A7 + A9 + A10 + A11, data=credit)
# Predict the outcomes using model
tree.result4 <- predict(tm4, type="class")
print(table(credit$A16, tree.result4))
```

```
##    tree.result4
##       0   1
##   0 341  42
##   1  38 269
```

```
success.rate.rpart4 <- table(credit$A16, tree.result4) %>% diag() %>% sum()/nrow(credit)
success.rate.rpart4
```

```
## [1] 0.884058
```

5. Compare the models

Which model performed best overall? Did logistic regression or decision trees perform better generally?

Answer -

Here are the success rate results from above test -

Group 1 logistic regression success rate - 0.8550725
Group 1 decision trees success rate - 0.8550725

Group 2 logistic regression success rate - 0.6884058
Group 2 decision trees success rate - 0.6768116

Group 3 logistic regression success rate - 0.7536232
Group 3 decision trees success rate - 0.8072464

Group 4 logistic regression success rate - 0.8826087
Group 4 decision trees success rate - 0.884058

Group 1 - Results are exactly same.

Group 2 - Logistic Regression is slightly better.

Group 3 - Decision Tree is a lot better.

Group 4 - Decision Tree is very slightly better.

Based on these results, I am concluding that though results of both, logistic regression and decision trees are higly comparable, Decision Tree model worked slightly better than logistic regression considering all 4 results.

We can also derive this by calculating mean success rate for logistic regressions and decision trees.

```
#Mean success rate from logitsic regression
mean.success.rate.glm <- (success.rate.glm1 + success.rate.glm2 +
                          success.rate.glm3 + success.rate.glm4)/4
#Mean success rate from decision tree
mean.success.rate.rpart <- (success.rate.rpart1 + success.rate.rpart2 +
                            success.rate.rpart3 + success.rate.rpart4)/4
mean.success.rate.glm
```

```
## [1] 0.7949275
```

```
mean.success.rate.rpart
```

```
## [1] 0.8057971
```

As noted above, mean Decision Tree success rate is slightly more than that of logistic regression. This reinforces our conclusion that Decision Tree model worked slightly better than logistic regression considering all 4 results.