INFX 573 Problem Set 8 - Classification

TAPASVI BANSAL

Due: Thursday, December 7, 2017

Introduction

Collaborators:

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.

```
#install.packages("tidyverse")
#install.packages("mfx")
#install.packages("GGally")
library(mfx)
```

```
## Loading required package: sandwich

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: MASS

## Loading required package: betareg
```

```r
library(rpart)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.2

## -- Attaching packages --------------------------------- tidyverse 1.2.1 --

## √ ggplot2 2.2.1     √ purrr   0.2.4
## √ tibble  1.3.4     √ dplyr   0.7.4
## √ tidyr   0.7.2     √ stringr 1.2.0
## √ readr   1.1.1     √ forcats 0.2.0

## Warning: package 'tidyr' was built under R version 3.4.2

## Warning: package 'purrr' was built under R version 3.4.2

## Warning: package 'dplyr' was built under R version 3.4.2

## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```r
library(GGally)
```

```
##
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':
##
##     nasa
```

```r
library(tidyverse)

CCdata <- read.csv("credit_card_applications.csv.bz2")
CCdata1 <- read.csv("credit_card_applications.csv.bz2")
#summary(CCdata)
```

Task

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

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.

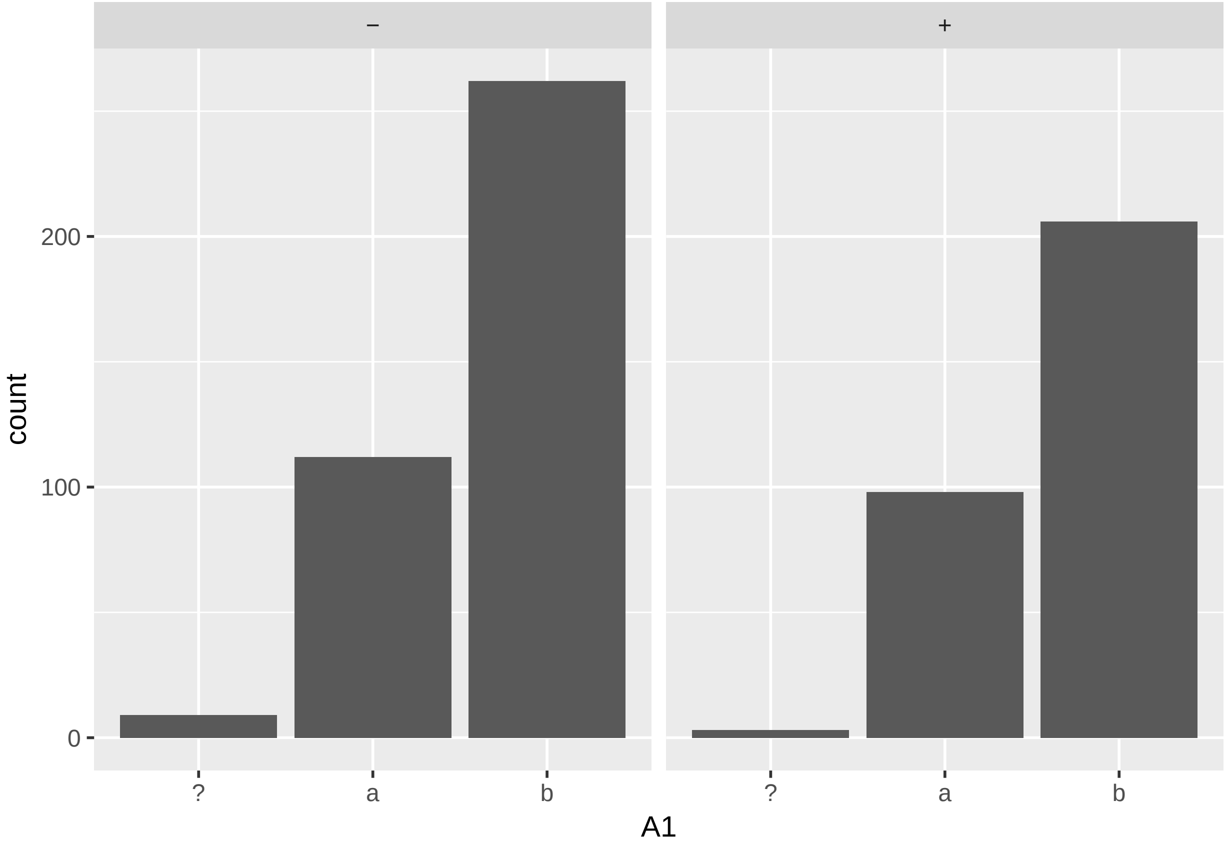
```r
CCdata$A2 <- as.numeric(as.character(CCdata$A2))
```

```
## Warning: NAs introduced by coercion
```

```r
CCdata$A14 <- as.numeric(as.character(CCdata$A14))
```

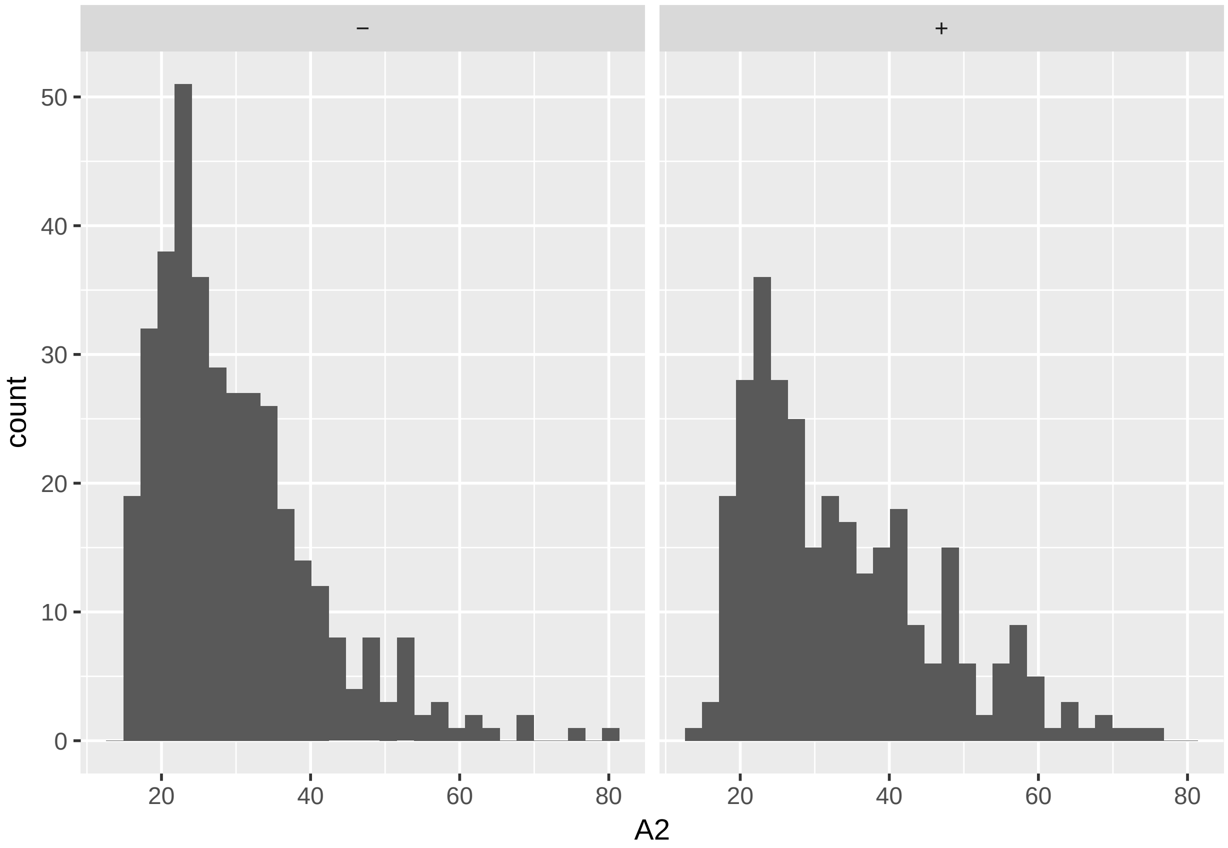
```
## Warning: NAs introduced by coercion
```

```r
ggplot(data=CCdata) + geom_bar(aes(x=A1)) + facet_wrap(~A16)
```



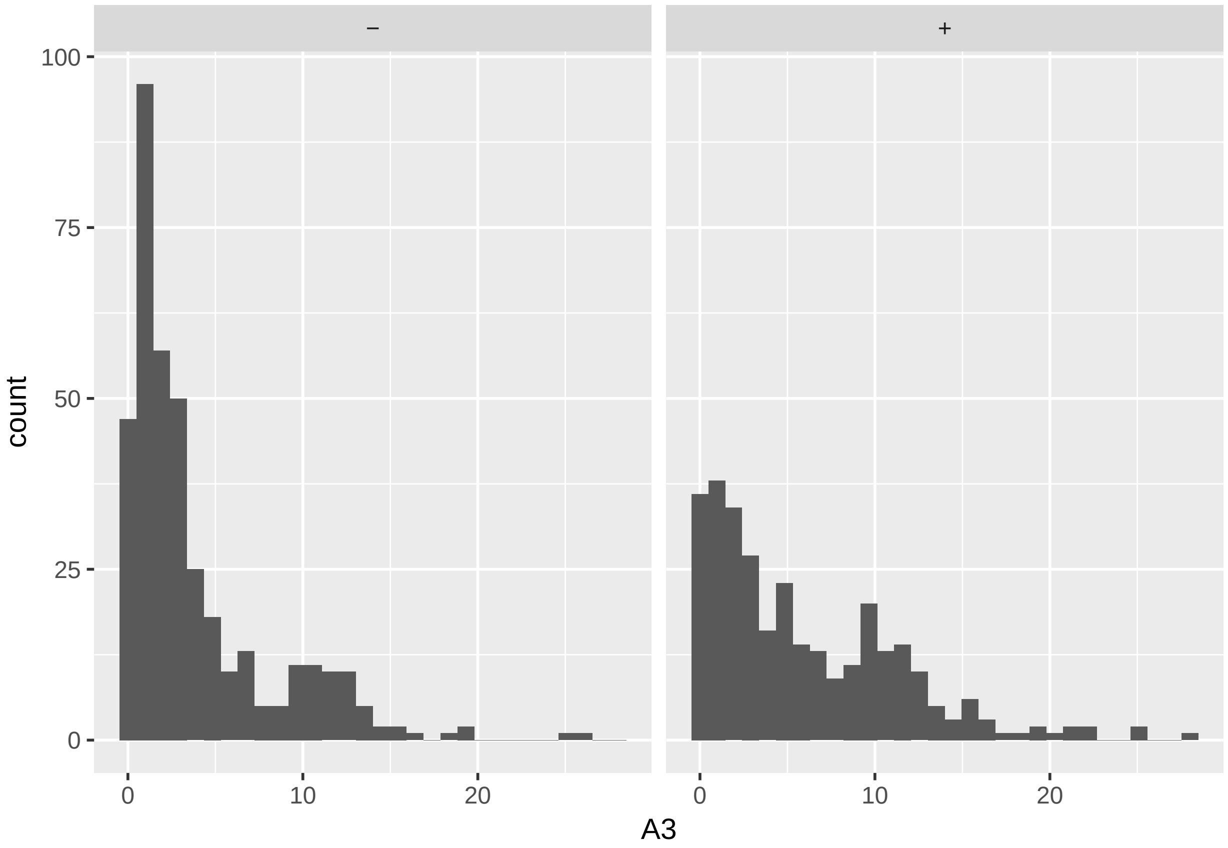
```
  # Observations in b for approved are less

ggplot(data=CCdata) + geom_histogram(aes(x=A2)) + facet_wrap(~A16)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 12 rows containing non-finite values (stat_bin).
```

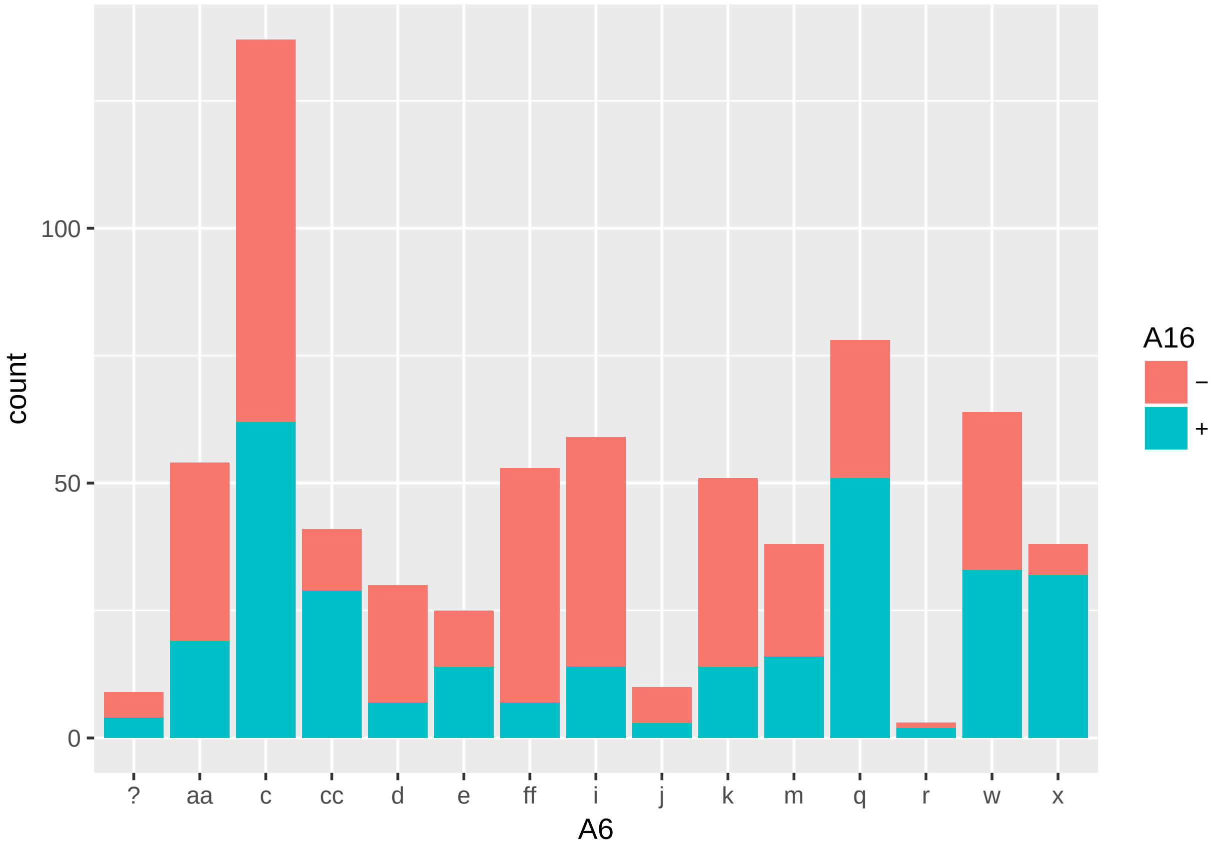


```
  # Values for A2 in rejected are low

ggplot(data=CCdata) + geom_histogram(aes(x=A3)) + facet_wrap(~A16)
```

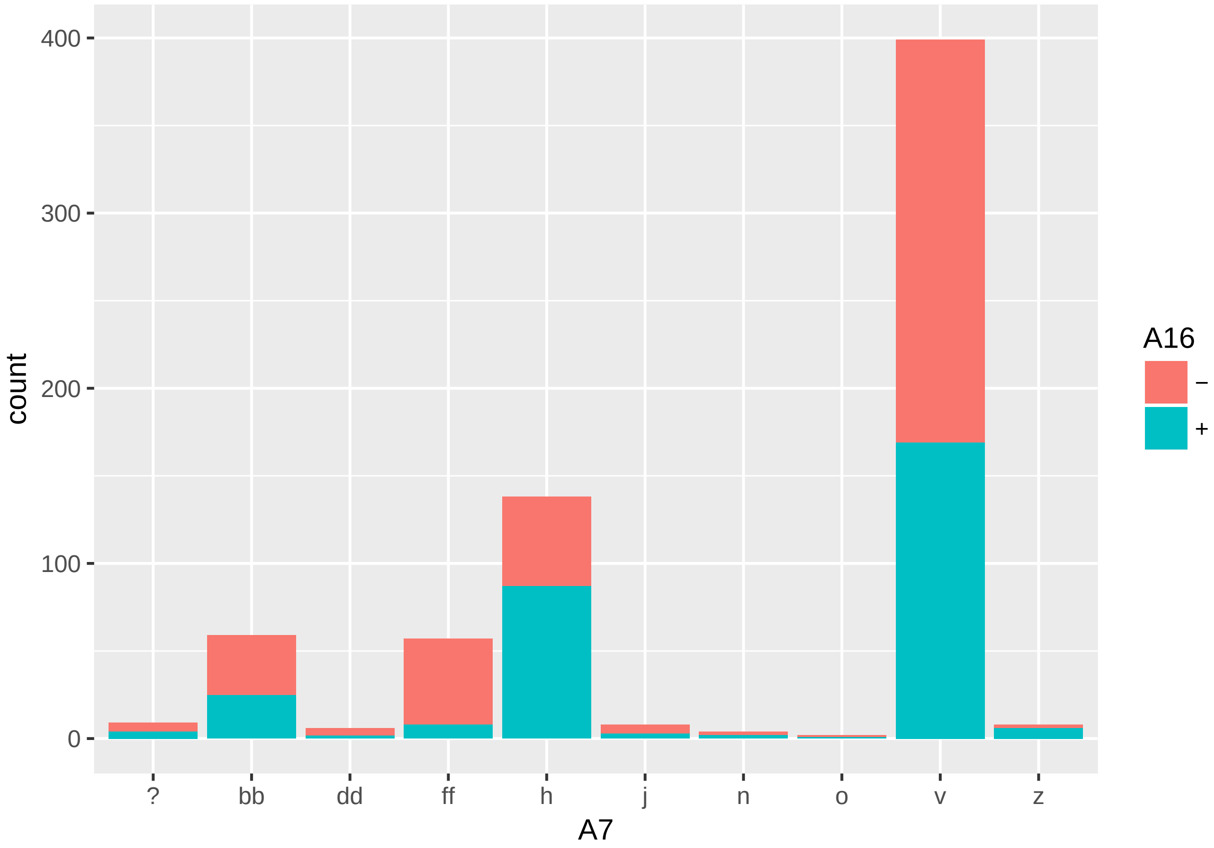
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
  # More values that are close to 0 for applications that are rejected in A3 - Second PRIORITY

#ggplot(data=CCdata) + geom_bar(aes(x=A4)) + facet_wrap(~A16)
  # Fewer observations in y for approved

#ggplot(data=CCdata) + geom_bar(aes(x=A5)) + facet_wrap(~A16)
  # Fewer observations in p for approved

ggplot(data=CCdata) + geom_bar(aes(x=A6,fill=A16))
```

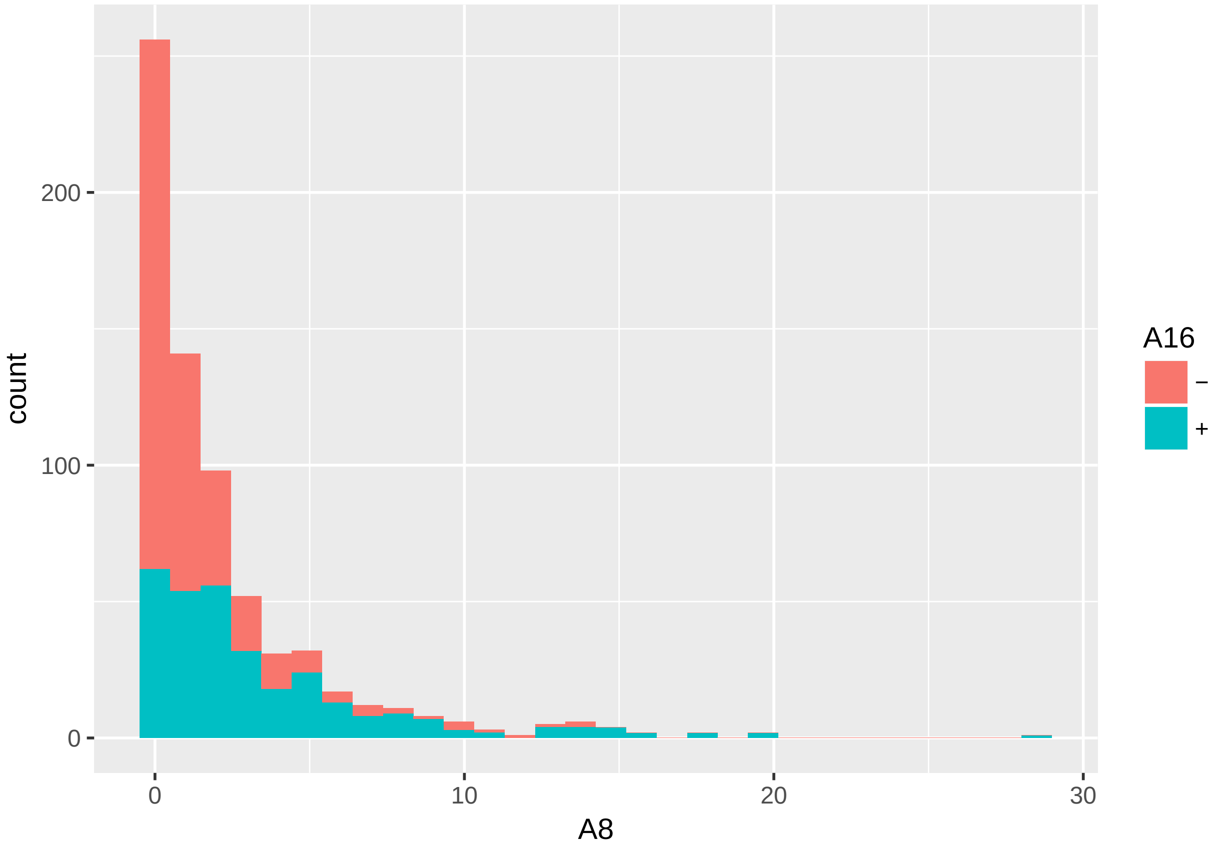


```
  # Observable differences in the attributes included

ggplot(data=CCdata) + geom_bar(aes(x=A7,fill=A16))
```

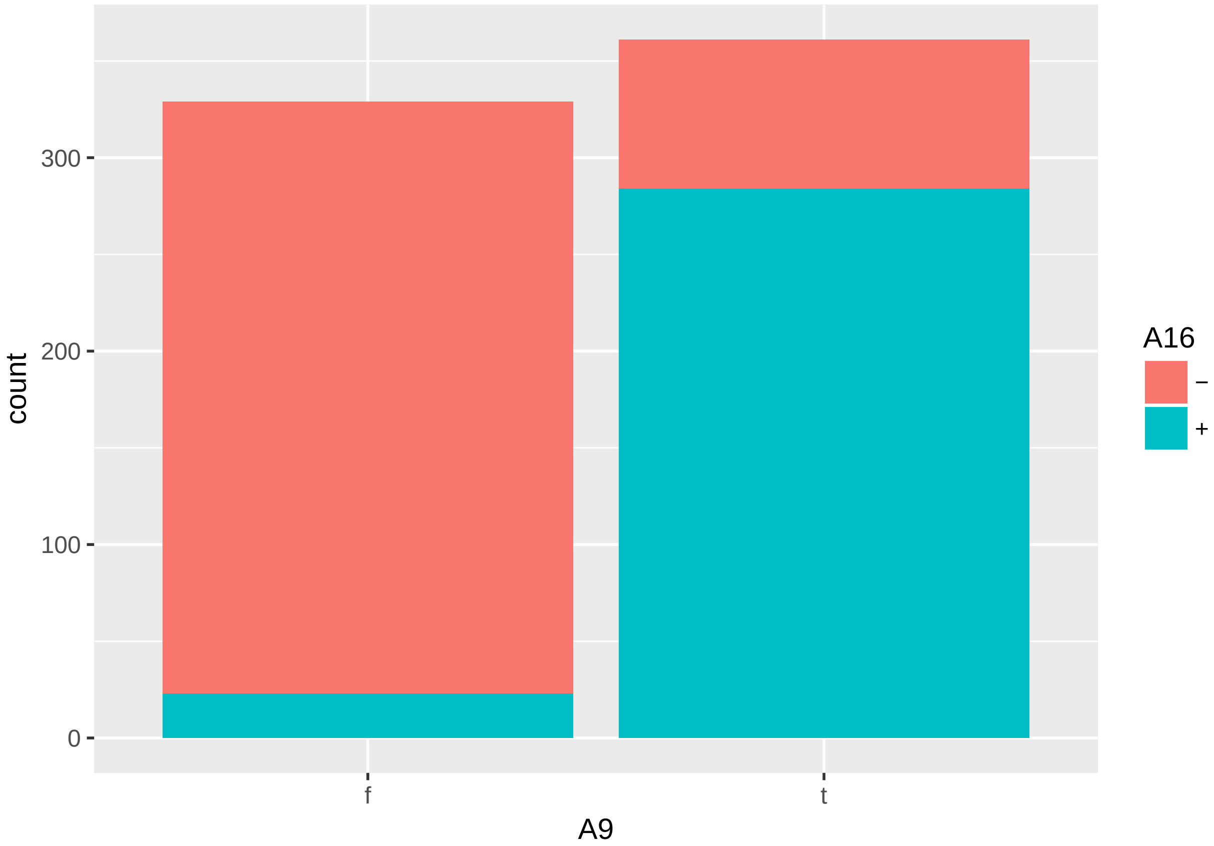


```
  # ff, h, and v attributes show observable differences b/w approvals & rejections

ggplot(data=CCdata) + geom_histogram(aes(x=A8,fill=A16))
```

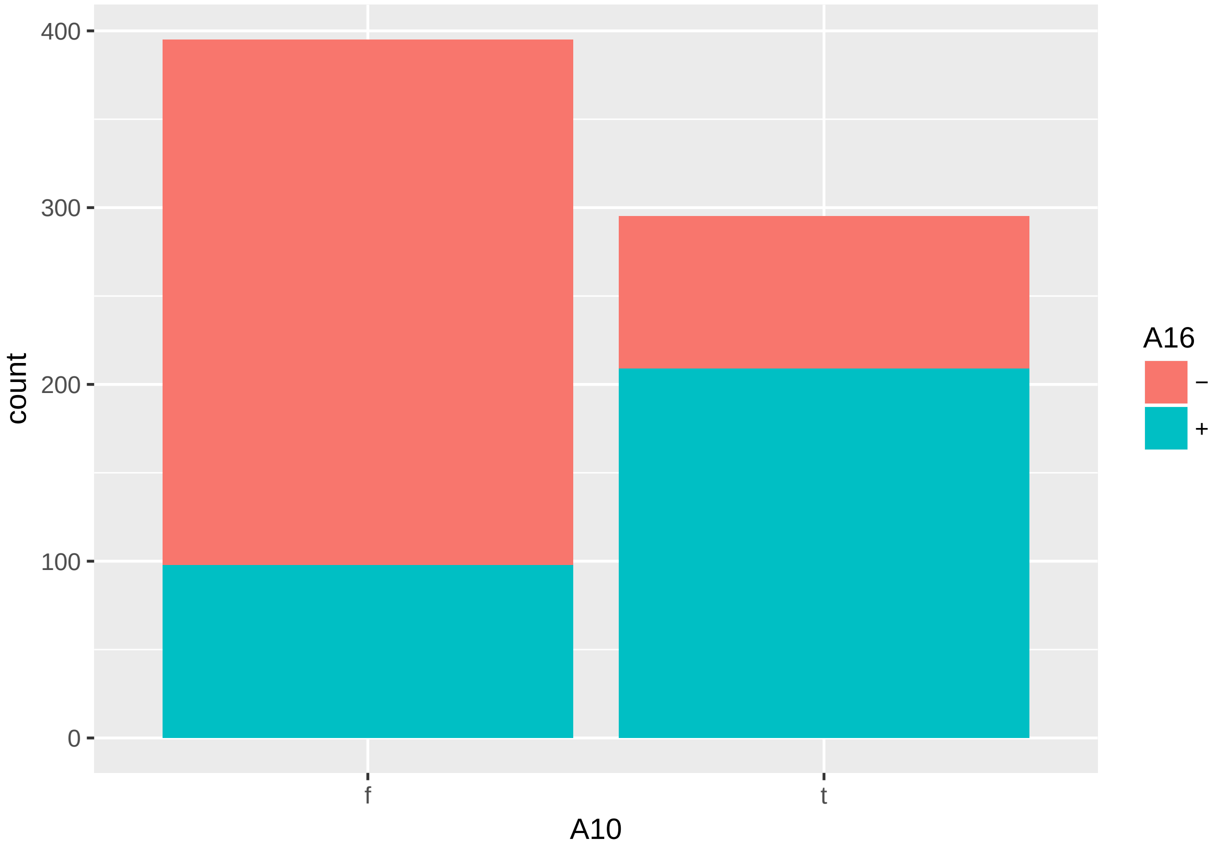
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
  # A8 has more values that are close to 0 for applications that are rejected - Second PRIORITY

ggplot(data=CCdata) + geom_bar(aes(x=A9,fill=A16))
```

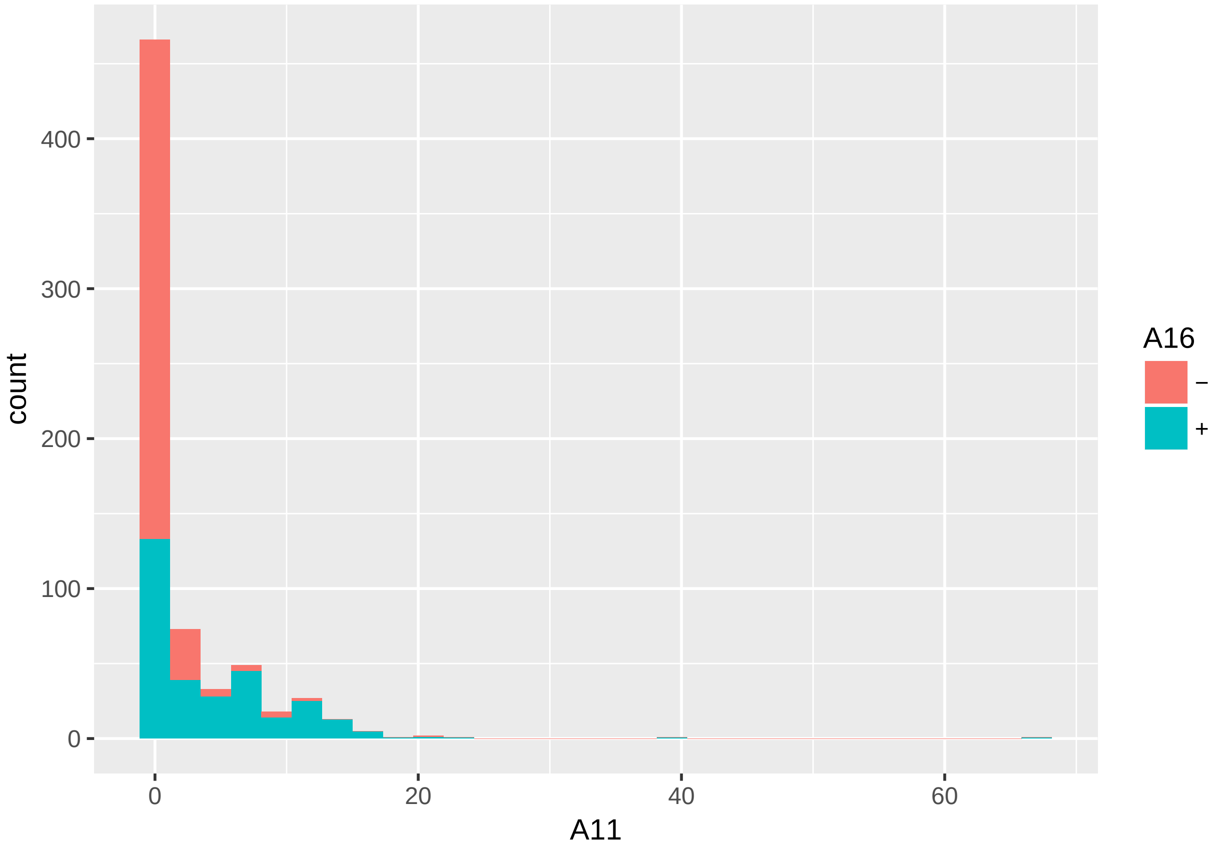


```
  # A9 is vastly of approval or rejection  - First PRIORITY

ggplot(data=CCdata) + geom_bar(aes(x=A10,fill=A16))
```

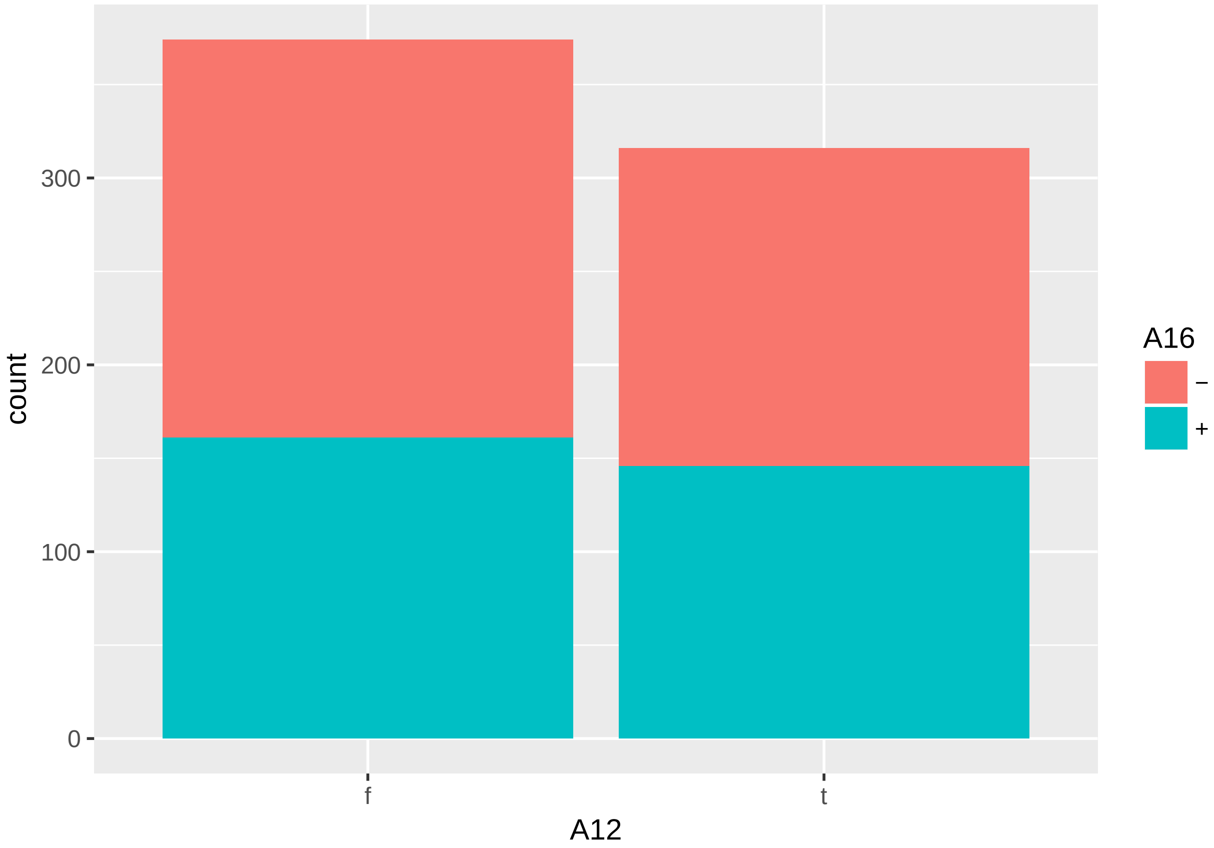


```
  # A10 is largely indicative of approval or rejection  - First PRIORITY

ggplot(data=CCdata) + geom_histogram(aes(x=A11,fill=A16))
```

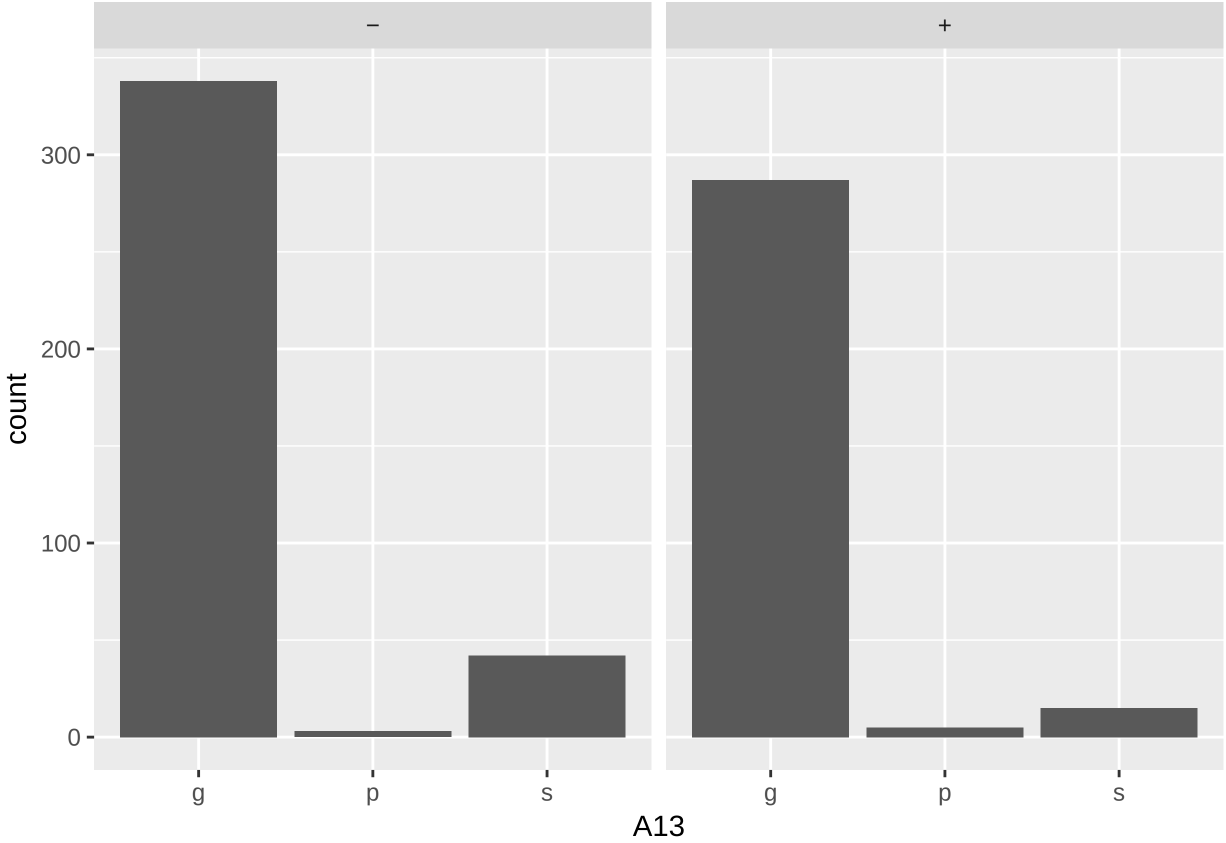
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
  # A11 has more values that are close to 0 for applications that are rejected - Second PRIORITY

ggplot(data=CCdata) + geom_bar(aes(x=A12,fill=A16))
```

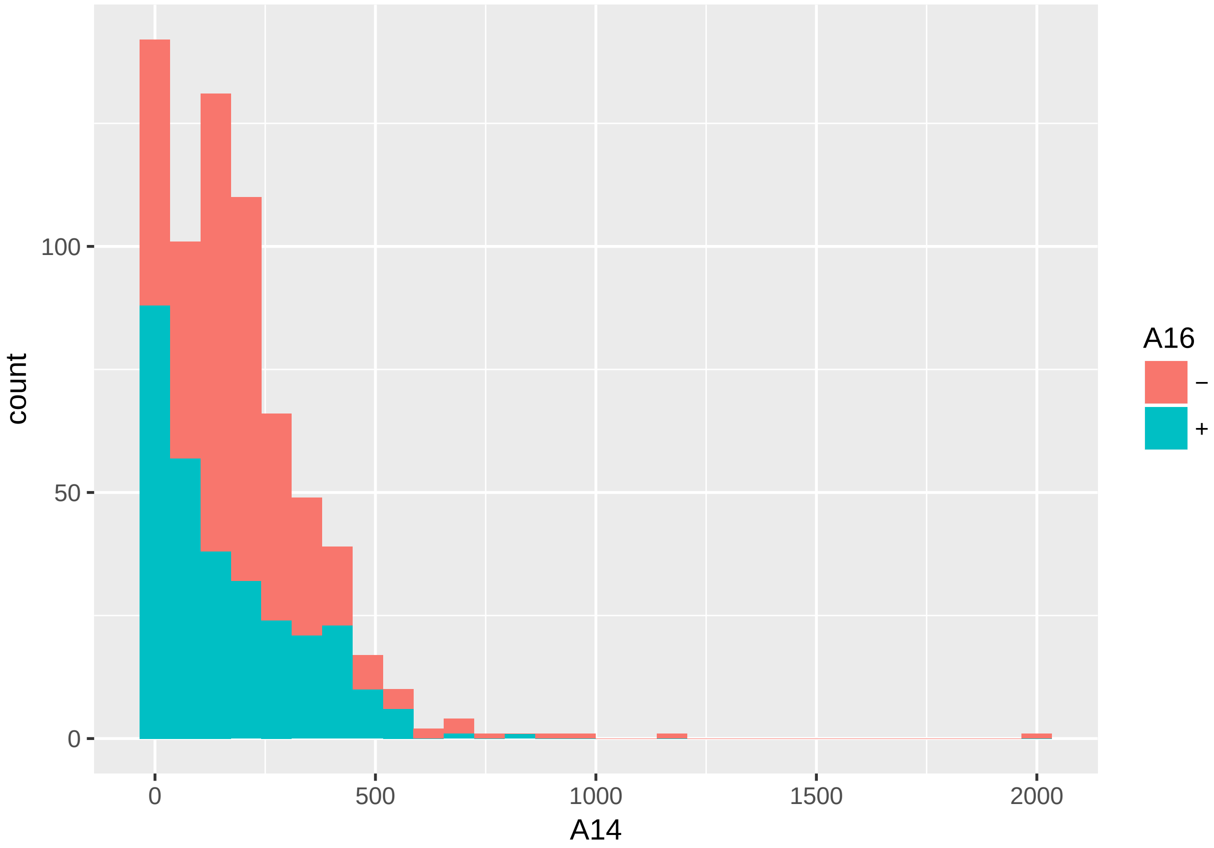


```
  # Not very different between approvals and rejections

ggplot(data=CCdata) + geom_bar(aes(x=A13)) + facet_wrap(~A16)
```



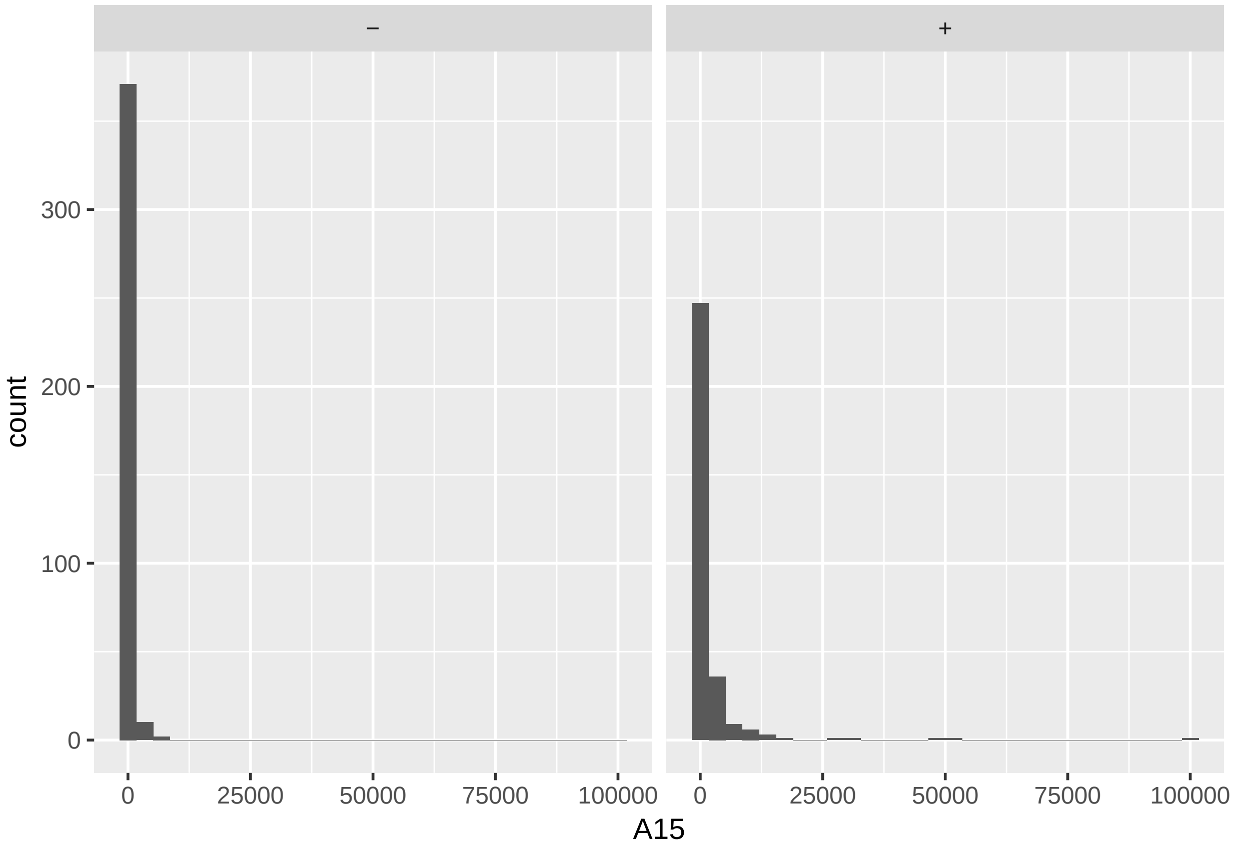
```
  # almost identical

ggplot(data=CCdata) + geom_histogram(aes(x=A14,fill=A16))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 13 rows containing non-finite values (stat_bin).
```



```
  # Rejected applications have higher values in A14 - Second PRIORITY

ggplot(data=CCdata) + geom_histogram(aes(x=A15)) + facet_wrap(~A16)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# A15 has more values that are close to 0 for applications that are rejected - Second PRIORITY
```

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?

```
#blrmodel <- mfx::logitmfx(A16 ~ A9, data= CCdata)
#blrmodel1 <- mfx::logitmfx(A16 ~ A10 , data= CCdata)


Pmodel <- glm(A16 ~ A15+A14+A11+A10+A9+A8+A3, data=CCdata1, family=binomial(link="logit"))
#summary(model)
lrp <- predict(Pmodel, type="response") > 0.5
crt <- table(CCdata1$A16,lrp)
diasum <- crt %>% diag() %>% sum()
PdtPercentage <- (crt[1]+crt[4])/sum(crt)
PdtPercentage
```

```
## [1] 0.9173913
```

1) The logistic regression model predicted the data with 91.73% accuracy.

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.

```
treemodel <- rpart::rpart(A16 ~ A15+A14+A11+A10+A9+A8+A3,data=CCdata1)
dtm <- predict(treemodel, type="class")

Tcrt <- table(CCdata1$A16,dtm)
Tcrt %>% diag() %>% sum()
```

```
## [1] 631
```

```
Tdiasum <- Tcrt %>% diag() %>% sum()
TPdtPercentage <- (Tdiasum)/sum(Tcrt)
TPdtPercentage
```

```
## [1] 0.9144928
```

1) The decision tree model predicted data with 91.44% accuracy.

4. Repeat the process

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

```
## Logistic regression, Almost All variables included (Set 1)

Tmodel1<- glm(A16 ~ A15+A14+A12+A11+A10+A9+A8+A7+A6+A5+A4+A3+A1, data=CCdata1, family=binomial(link="log
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
lrp1 <- predict(Tmodel1, type="response") > 0.5
crt1 <- table(CCdata1$A16,lrp1)
crt1 %>% diag() %>% sum()
```

```
## [1] 648
```

```
diasum1 <- crt1 %>% diag() %>% sum()
PdtPercentage1 <- (crt1[1]+crt1[4])/sum(crt1)
PdtPercentage1
```

```
## [1] 0.9391304
```

1) The logistic regression model predicted the data with 93.91% accuracy.

```
## Decision tree, All variables (Set 1)

Tmodel1 <- rpart::rpart(A16 ~ A15+A14+A12+A11+A10+A9+A8+A7+A6+A5+A4+A3+A1,data=CCdata1)
Tdtm1 <- predict(Tmodel1, type="class")
Tcrt1 <- table(CCdata1$A16,Tdtm1)
Tcrt1 %>% diag() %>% sum()
```

```
## [1] 642
```

```
Tdiasum1 <- Tcrt1 %>% diag() %>% sum()
TPdtPercentage1 <- (Tdiasum1)/sum(Tcrt1)
TPdtPercentage1
```

```
## [1] 0.9304348
```

1) The decision tree model predicted data with 93.04% accuracy.

```
## Logistic regression, Trimming down some variables (Set 2)

lmodel2 <- glm(A16 ~ A15+A14+A9+A2, data=CCdata1, family=binomial(link="logit"))
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
lrp2 <- predict(lmodel2, type="response") > 0.5
lcrt2 <- table(CCdata1$A16,lrp2)
lcrt2 %>% diag() %>% sum()
```

```
## [1] 689
```

```
ldiasum2 <- lcrt2 %>% diag() %>% sum()
PdtPercentage2 <- (lcrt2[1]+lcrt2[4])/sum(lcrt2)
PdtPercentage2
```

```
## [1] 0.9985507
```

1) The logistic regression model predicted the data with “99.85%” accuracy.

```
## Decision tree, Trimming down some variables (Set 2)

Tmodel2 <- rpart::rpart(A16 ~ A15+A14+A9+A2, data=CCdata1)
dtm2 <- predict(Tmodel2, type="class")
Tcrt2 <- table(CCdata1$A16,dtm2)
Tcrt2 %>% diag() %>% sum()
```

```
## [1] 662
```

```
Tdiasum2 <- Tcrt2 %>% diag() %>% sum()
TPdtPercentage2 <- (Tdiasum2)/sum(Tcrt2)
TPdtPercentage2
```

```
## [1] 0.9594203
```

1) The decision tree model predicted data with 95.94% accuracy.

```
# Logistic regression, Feature engineering (Set 3):
lmodel3 <- glm(A16 ~ A15^2+A14^2+A13+A12+sqrt(A11)+A10+A9+sqrt(A8)+A7+A6+A5+A4+A3, data=CCdata1, family
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
lrp3 <- predict(lmodel3, type="response") > 0.5
lcrt3 <- table(CCdata1$A16,lrp3)
lcrt3 %>% diag() %>% sum()
```

```
## [1] 646
```

```
Pdiasum3 <- lcrt3 %>% diag() %>% sum()
PdtPercentage3 <- (lcrt3[1]+lcrt3[4])/sum(lcrt3)
PdtPercentage3
```

```
## [1] 0.9362319
```

1) The logistic regression model predicted the data with 93.62% accuracy.

```
## Decision tree, Feature engineering (Set 3)

Tmodel3 <- rpart::rpart(A16 ~ A15^2+A14^2+A13+A12+sqrt(A11)+A10+A9+sqrt(A8)+A7+A6+A5+A4+A3, data=CCdata
dtm3 <- predict(Tmodel3, type="class")
Tcrt3 <- table(CCdata1$A16,dtm3)
Tcrt3 %>% diag() %>% sum()
```

```
## [1] 642
```

```
Tdiasum3 <- Tcrt3 %>% diag() %>% sum()
TPdtPercentage3 <- (Tdiasum3/sum(Tcrt3))
TPdtPercentage3
```

```
## [1] 0.9304348
```

1) The decision tree model predicted data with 93.04% accuracy.

5. Compare the models

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

On subjecting the given dataset to both logistic regression modelling and decision tree modelling, multiple times (by altering the input variables and also using feature engineering), logistic regression models performed better (higher percentage of accuracy) than all the decision tree models.