R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

```
library("tidyverse")

## ── Attaching packages ────────────────────────────────────────
tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.0     ✓ purrr   0.3.3
## ✓ tibble  3.0.0     ✓ dplyr   0.8.5
## ✓ tidyr   1.0.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0

## ── Conflicts ─────────────────────────────────────────────────
tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library("tidymodels")

## ── Attaching packages ────────────────────────────────────────
tidymodels 0.1.0 ──

## ✓ broom     0.5.5      ✓ rsample   0.0.6
## ✓ dials     0.0.5      ✓ tune      0.1.0
## ✓ infer     0.5.1      ✓ workflows 0.1.1
## ✓ parsnip   0.0.5      ✓ yardstick 0.0.6
## ✓ recipes   0.1.10
```

```
## ── Conflicts ──────────────────────────────────────────
tidymodels_conflicts() ──
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x dials::margin()   masks ggplot2::margin()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
```

```r
library("plotly")
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library("skimr")
library("caret")
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following objects are masked from 'package:yardstick':
##
##     precision, recall, sensitivity, specificity

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
setwd("/Users/vishaldodamani/Downloads/DataMining/Assignment5")
df<-read_csv("airlines.csv")
```

```
## Parsed with column specification:
## cols(
##   ID = col_double(),
##   Balance = col_double(),
##   Qual_miles = col_double(),
```

```
##   cc1_miles = col_double(),
##   cc2_miles = col_double(),
##   cc3_miles = col_double(),
##   Bonus_miles = col_double(),
##   Bonus_trans = col_double(),
##   Flight_miles_12mo = col_double(),
##   Flight_trans_12 = col_double(),
##   Days_since_enroll = col_double(),
##   Award = col_double()
## )
```

```
df<-df %>% mutate(Award=as.factor(Award))
```

```
str(df)
```

```
## tibble [3,999 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ID               : num [1:3999] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Balance          : num [1:3999] 28143 19244 41354 14776 97752 ...
##  $ Qual_miles       : num [1:3999] 0 0 0 0 0 0 0 0 0 0 ...
##  $ cc1_miles        : num [1:3999] 1 1 1 1 4 1 3 1 3 3 ...
##  $ cc2_miles        : num [1:3999] 1 1 1 1 1 1 1 1 2 1 ...
##  $ cc3_miles        : num [1:3999] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Bonus_miles      : num [1:3999] 174 215 4123 500 43300 ...
##  $ Bonus_trans      : num [1:3999] 1 2 4 1 26 0 25 4 43 28 ...
##  $ Flight_miles_12mo: num [1:3999] 0 0 0 0 2077 ...
##  $ Flight_trans_12  : num [1:3999] 0 0 0 0 4 0 0 1 12 3 ...
##  $ Days_since_enroll: num [1:3999] 7000 6968 7034 6952 6935 ...
##  $ Award            : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 2 2 ...
```

```
skim(df)
```

Data summary

Name	df
Number of rows	3999
Number of columns	12
_______________________	
Column type frequency:	
factor	1
numeric	11
________________________	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Award	0	1	FALSE	2	0: 2518, 1: 1481

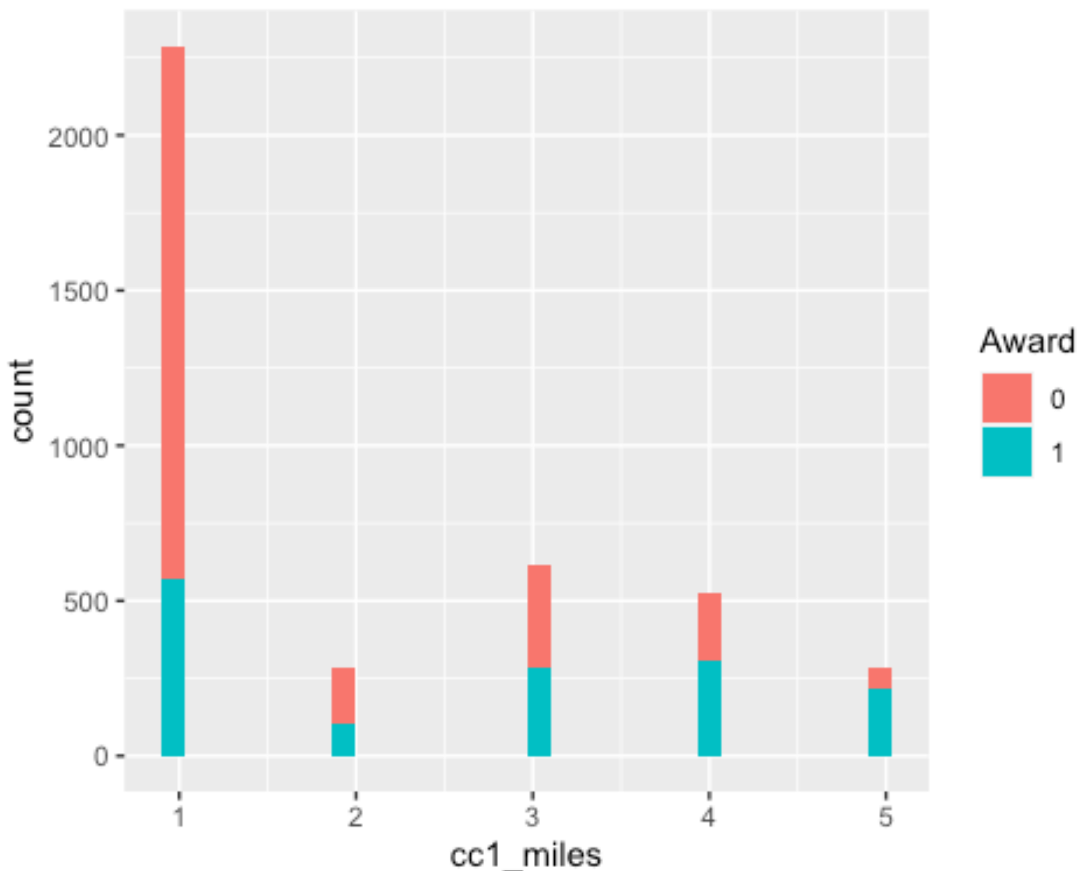
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ID	0	1	2014.82	1160.76	1	1010.5	2016	3020.5	4021	▇▇▇▇▇
Balance	0	1	73601.33	100775.66	0	18527.5	43097	92404.0	1704838	▇▁▁▁▁
Qual_miles	0	1	144.11	773.66	0	0.0	0	0.0	11148	▇▁▁▁▁
cc1_miles	0	1	2.06	1.38	1	1.0	1	3.0	5	▇▁▂▂▁
cc2_miles	0	1	1.01	0.15	1	1.0	1	1.0	3	▇▁▁▁▁
cc3_miles	0	1	1.01	0.20	1	1.0	1	1.0	5	▇▁▁▁▁
Bonus_miles	0	1	17144.85	24150.97	0	1250.0	7171	23800.5	263685	▇▁▁▁▁
Bonus_trans	0	1	11.60	9.60	0	3.0	12	17.0	86	▇▃▁▁▁
Flight_miles_12mo	0	1	460.06	1400.21	0	0.0	0	311.0	30817	▇▁▁▁▁
Flight_trans_12	0	1	1.37	3.79	0	0.0	0	1.0	53	▇▁▁▁▁
Days_since_enroll	0	1	4118.56	2065.13	2	2330.0	4096	5790.5	8296	▃▇▇▇▃

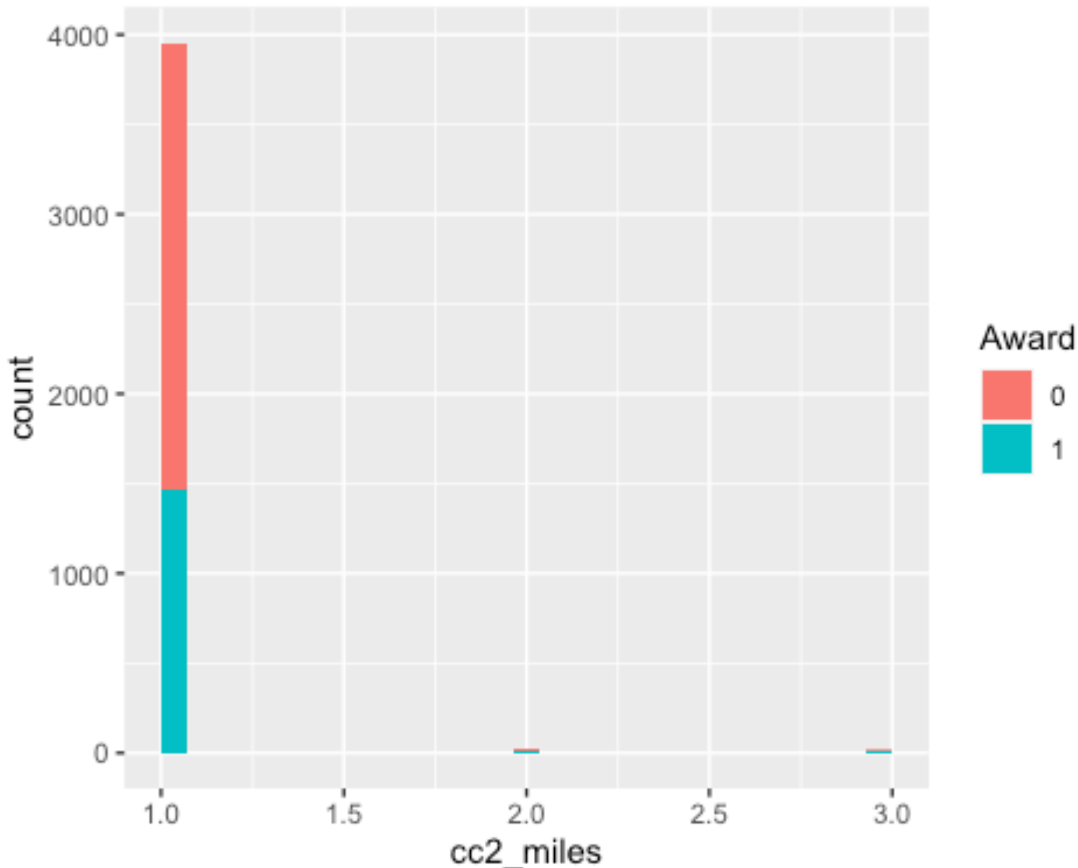
```
set.seed(123)

dftrain<-  df %>% sample_frac(0.7)
dftest<-dplyr::setdiff(df,dftrain)
dftest

## # A tibble: 1,200 x 12
##        ID Balance Qual_miles cc1_miles cc2_miles cc3_miles Bonus_miles
##     <dbl>   <dbl>      <dbl>     <dbl>     <dbl>     <dbl>       <dbl>
##  1      3   41354          0         1         1         1        4123
##  2      6   16420          0         1         1         1           0
##  3     14   43097          0         1         1         1        3258
##  4     22  185681       2024         1         1         1       13300
##  5     43   60313          0         1         1         1       10000
##  6     47   92336          0         2         1         1       11214
##  7     50   17051          0         1         1         1        1150
##  8     53  118531          0         4         1         1       44577
```

```
## 9     55    38348          0          1          1          1            0
## 10    57    75971          0          4          1          1        34339
## # … with 1,190 more rows, and 5 more variables: Bonus_trans <dbl>,
## #   Flight_miles_12mo <dbl>, Flight_trans_12 <dbl>, Days_since_enroll
<dbl>,
## #   Award <fct>
```

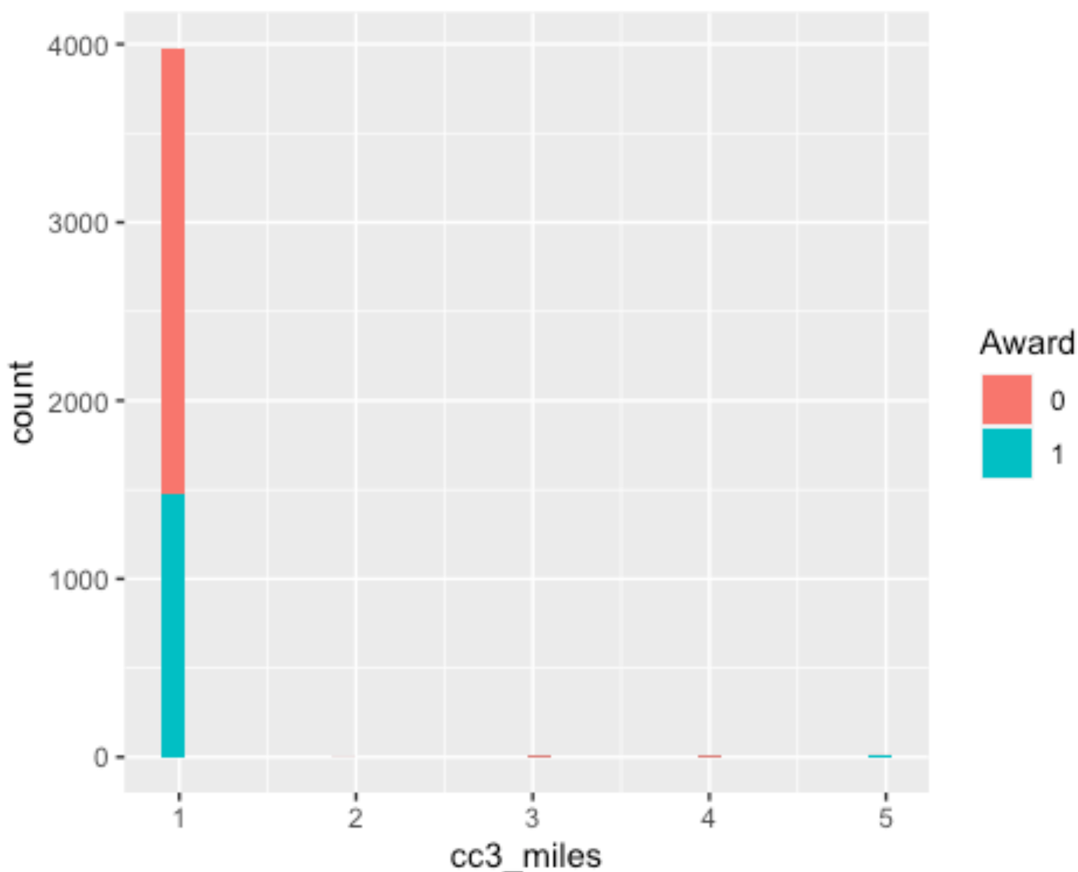
```r
HistAward<-df %>% ggplot(aes(x=cc1_miles,fill=Award))+geom_histogram()
HistAward

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



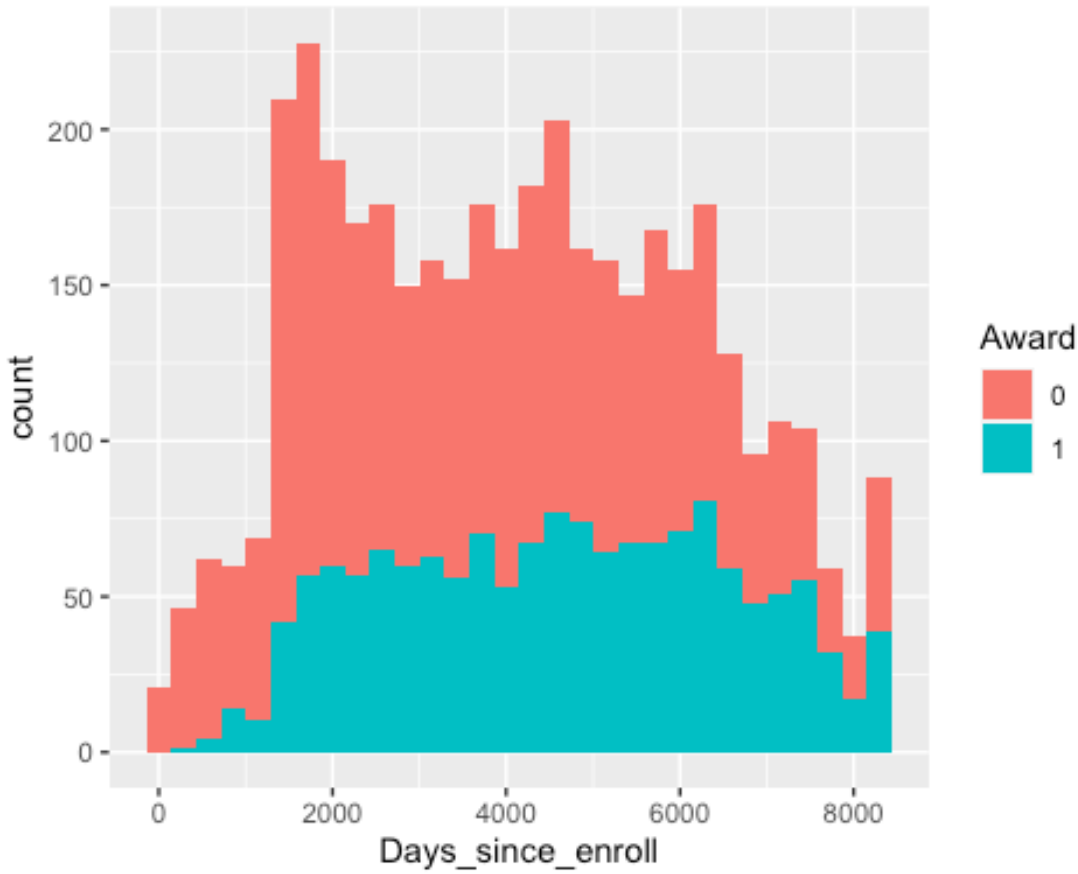

```r
HistAward2<-df %>% ggplot(aes(x=cc2_miles,fill=Award))+geom_histogram()
HistAward2

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



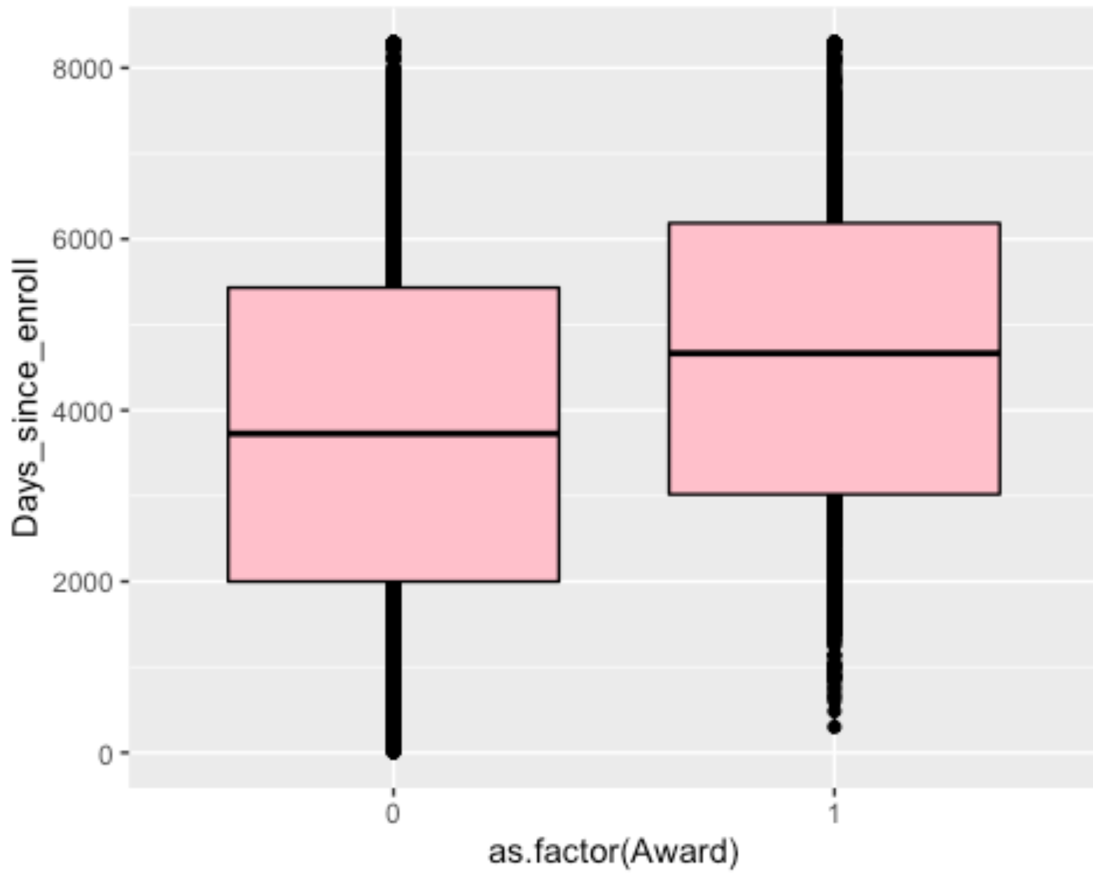

```
HistAward3<-df %>% ggplot(aes(x=cc3_miles,fill=Award))+geom_histogram()
HistAward3

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



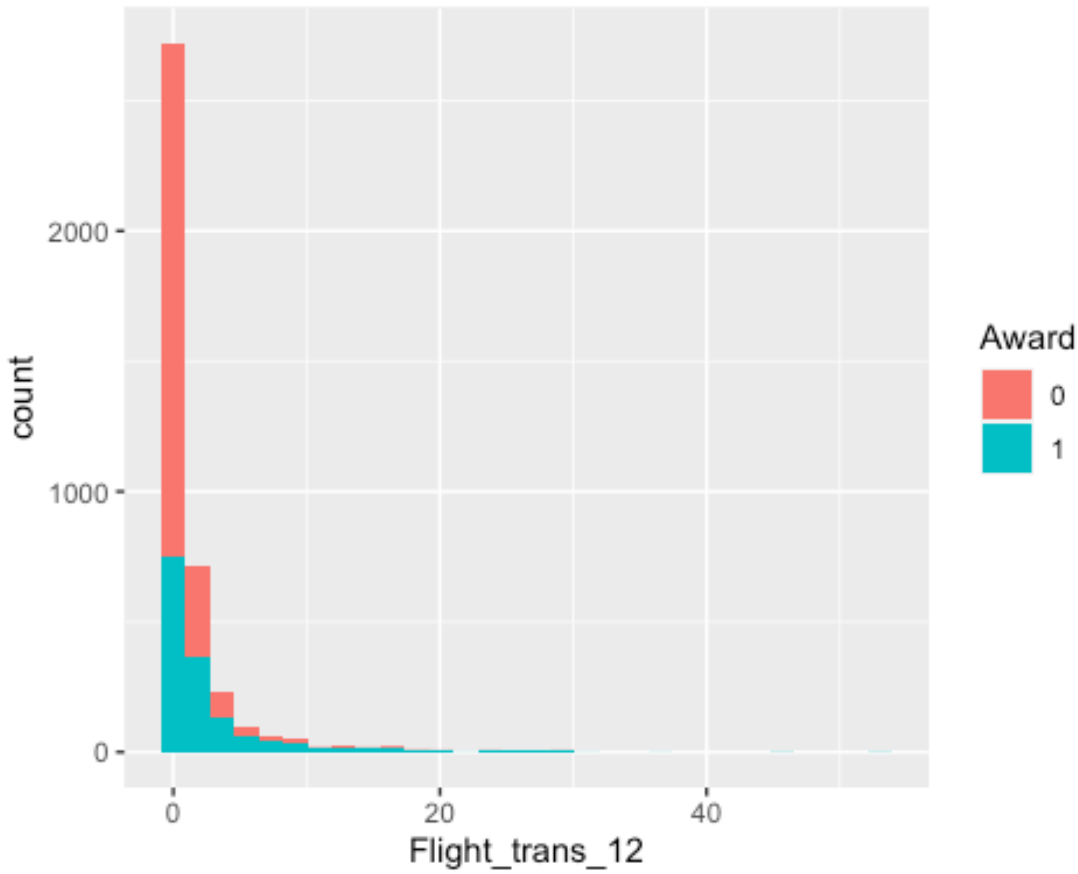

```
HistAward4<-df %>%
ggplot(aes(x=Days_since_enroll,fill=Award))+geom_histogram()
HistAward4

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



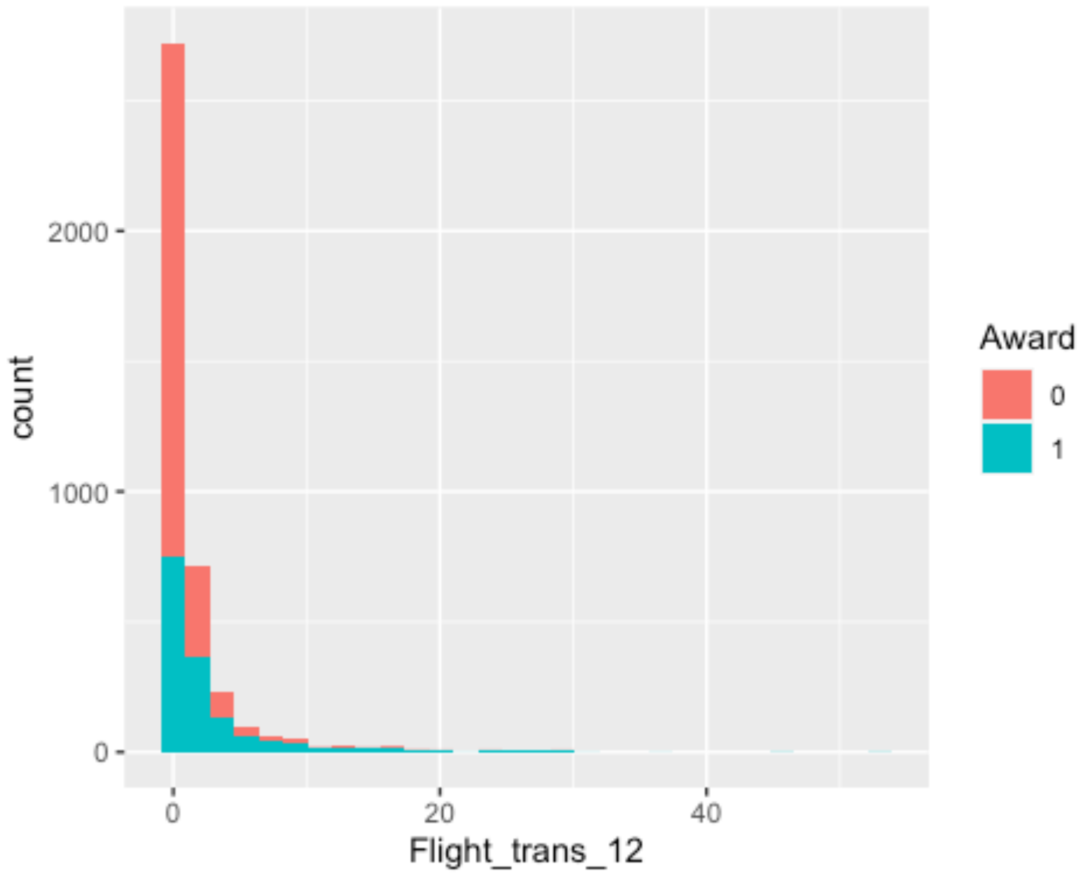

```
BoxPLotAward<-df %>%
ggplot(aes(x=as.factor(Award),y=Days_since_enroll))+geom_point()+geom_boxplot
(fill="pink", color="black")
BoxPLotAward
```



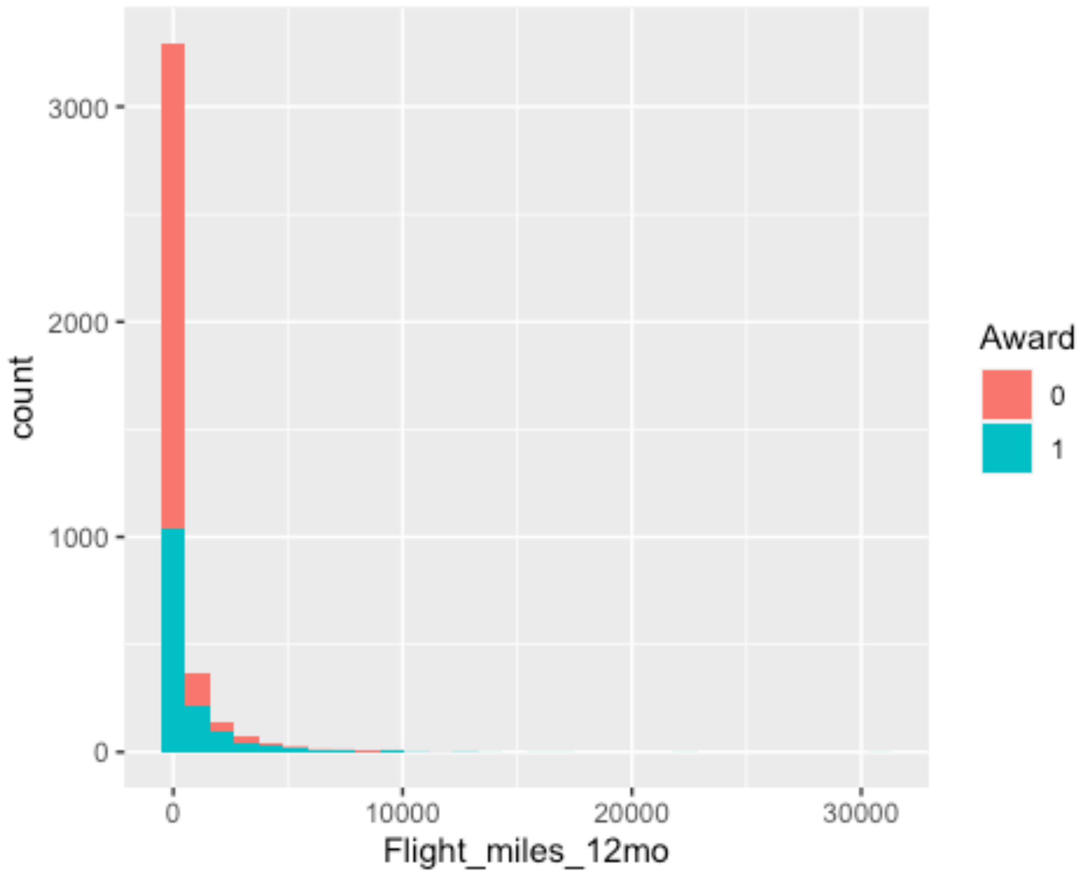

```
HistAward5<-df %>% ggplot(aes(x=Flight_trans_12,fill=Award))+geom_histogram()
HistAward5

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




```
HistAward6<-df %>%
ggplot(aes(x=as.factor(Award),y=Flight_miles_12mo))+geom_point()+geom_boxplot
(fill="pink", color="black")
HistAward5

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




```
BoxPLotAward2<-df
%>%ggplot(aes(x=Flight_miles_12mo,fill=Award))+geom_histogram()
BoxPLotAward2

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




```
df %>% group_by(cc3_miles,as.factor(Award)) %>% tally()

## # A tibble: 10 x 3
## # Groups:   cc3_miles [5]
##    cc3_miles `as.factor(Award)`     n
##        <dbl> <fct>              <int>
##  1         1 0                   2509
##  2         1 1                   1472
##  3         2 0                      2
##  4         2 1                      1
##  5         3 0                      2
##  6         3 1                      2
##  7         4 0                      4
##  8         4 1                      2
##  9         5 0                      1
## 10         5 1                      4

df %>% group_by(cc2_miles,as.factor(Award)) %>% tally()

## # A tibble: 6 x 3
## # Groups:   cc2_miles [3]
##   cc2_miles `as.factor(Award)`     n
##       <dbl> <fct>              <int>
## 1         1 0                   2492
```

```
## 2          1 1                      1464
## 3          2 0                        17
## 4          2 1                        11
## 5          3 0                         9
## 6          3 1                         6

result<-
   train(Award~.,family='binomial',method='glm',data=dftrain) %>%
  predict(dftest,type='raw') %>%
  bind_cols(dftest,Pred_Award=.)

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

result %>%
  xtabs(~Pred_Award+Award, .) %>%
  confusionMatrix(positive = '1')

## Confusion Matrix and Statistics
##
##             Award
## Pred_Award   0   1
##            0 674 260
##            1  66 200
##
##                 Accuracy : 0.7283
##                   95% CI : (0.7022, 0.7533)
##      No Information Rate : 0.6167
##      P-Value [Acc > NIR] : 2.421e-16
##
##                    Kappa : 0.3756
##
##   Mcnemar's Test P-Value : < 2.2e-16
##
##              Sensitivity : 0.4348
##              Specificity : 0.9108
##           Pos Pred Value : 0.7519
##           Neg Pred Value : 0.7216
##               Prevalence : 0.3833
##           Detection Rate : 0.1667
##     Detection Prevalence : 0.2217
##        Balanced Accuracy : 0.6728
##
##         'Positive' Class : 1
##

performance <-
  metric_set(rmse, mae)

performance(result, truth = as.numeric(Award), estimate =
as.numeric(Pred_Award))
```

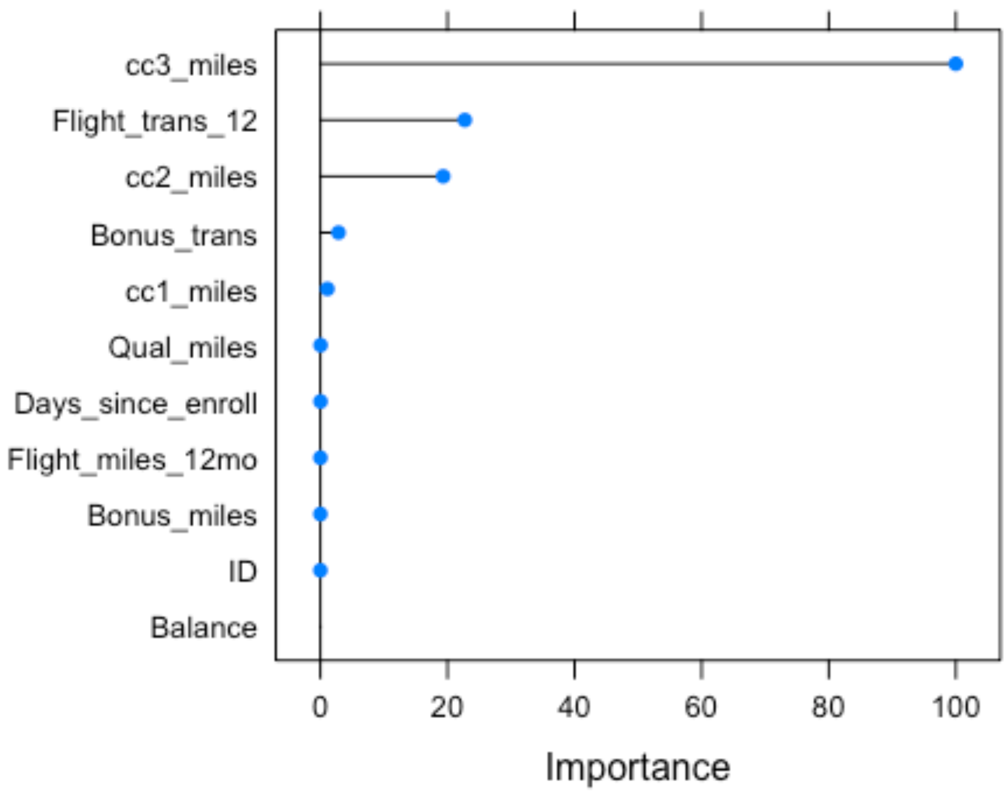
```
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.521
## 2 mae     standard       0.272
```

```
lambdaValues <- 10^seq(-3, 3, length = 100)

R1<-
  train(Award ~ ., family='binomial', data=dftrain, method='glmnet',
trControl=trainControl(method='cv', number=10), tuneLength=100)

resultsElasticNet <-
  R1 %>%
  predict(dftest, type='raw') %>%
  bind_cols(dftest, predictedAward=.)

resultsElasticNet %>%
  xtabs(~predictedAward+Award, .) %>%
  confusionMatrix(positive = '1')
```

```
## Confusion Matrix and Statistics
##
##               Award
## predictedAward   0   1
##              0 673 258
##              1  67 202
##
##                Accuracy : 0.7292
##                  95% CI : (0.7031, 0.7541)
##     No Information Rate : 0.6167
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3783
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4391
##             Specificity : 0.9095
##          Pos Pred Value : 0.7509
##          Neg Pred Value : 0.7229
##              Prevalence : 0.3833
##          Detection Rate : 0.1683
##    Detection Prevalence : 0.2242
##       Balanced Accuracy : 0.6743
##
##        'Positive' Class : 1
##
```

```
varImp(R1)$importance %>%
  rownames_to_column(var = "Variable") %>%
```

```
  mutate(Importance = scales::percent(Overall/100)) %>%
  arrange(desc(Overall)) %>%
  as_tibble()

## # A tibble: 11 x 3
##    Variable          Overall Importance
##    <chr>               <dbl> <chr>
##  1 cc3_miles       100       100.0000%
##  2 Flight_trans_12  22.7     22.7322%
##  3 cc2_miles        19.3     19.3252%
##  4 Bonus_trans       2.86    2.8560%
##  5 cc1_miles         1.11    1.1082%
##  6 Qual_miles        0.0330  0.0330%
##  7 Days_since_enroll 0.0243  0.0243%
##  8 Flight_miles_12mo 0.00766 0.0077%
##  9 Bonus_miles       0.00491 0.0049%
## 10 ID                0.00285 0.0028%
## 11 Balance           0       0.0000%

plot(varImp(R1))
```




```
performance(resultsElasticNet, truth = as.numeric(Award), estimate =
as.numeric(predictedAward))
```

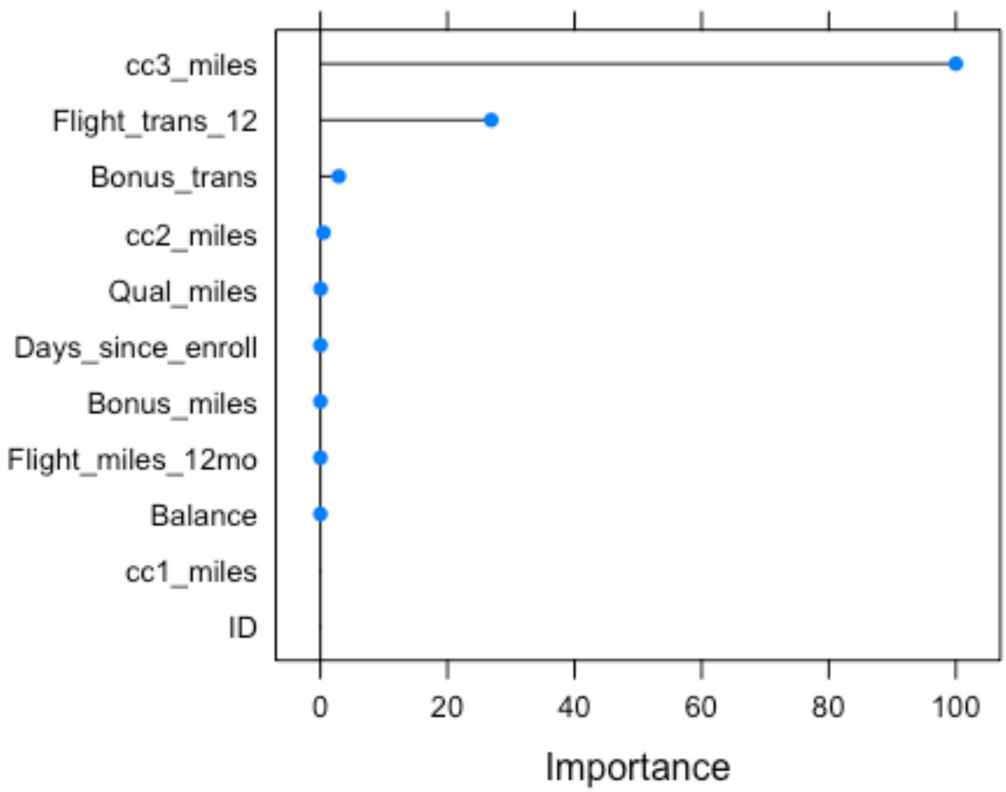
```
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.520
## 2 mae     standard       0.271
```

```
fitLasso <-
  train(Award ~ ., family='binomial', data=dftrain, method='glmnet',
trControl=trainControl(method='cv', number=10), tuneGrid =
expand.grid(alpha=1, lambda=lambdaValues))

resultsLasso <-
  fitLasso %>%
  predict(dftest, type='raw') %>%
  bind_cols(dftest, predictedAward=.)

resultsLasso %>%
  xtabs(~predictedAward+Award, .) %>%
  confusionMatrix(positive = '1')
```

```
## Confusion Matrix and Statistics
##
##               Award
## predictedAward   0   1
##              0 677 263
##              1  63 197
##
##                Accuracy : 0.7283
##                  95% CI : (0.7022, 0.7533)
##     No Information Rate : 0.6167
##     P-Value [Acc > NIR] : 2.421e-16
##
##                   Kappa : 0.3739
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4283
##             Specificity : 0.9149
##          Pos Pred Value : 0.7577
##          Neg Pred Value : 0.7202
##              Prevalence : 0.3833
##          Detection Rate : 0.1642
##    Detection Prevalence : 0.2167
##       Balanced Accuracy : 0.6716
##
##        'Positive' Class : 1
##
```

```
varImp(fitLasso)$importance %>%
  rownames_to_column(var = "Variable") %>%
```

```
  mutate(Importance = scales::percent(Overall/100)) %>%
  arrange(desc(Overall)) %>%
  as_tibble()

## # A tibble: 11 x 3
##    Variable              Overall Importance
##    <chr>                   <dbl> <chr>
##  1 cc3_miles          100        100%
##  2 Flight_trans_12     26.9      27%
##  3 Bonus_trans          2.90     3%
##  4 cc2_miles            0.486    0%
##  5 Qual_miles           0.0338   0%
##  6 Days_since_enroll    0.0244   0%
##  7 Bonus_miles          0.00589  0%
##  8 Flight_miles_12mo    0.00191  0%
##  9 Balance              0.000227 0%
## 10 ID                   0        0%
## 11 cc1_miles            0        0%

plot(varImp(fitLasso))
```




```
performance(resultsLasso, truth = as.numeric(Award), estimate =
as.numeric(predictedAward))
```

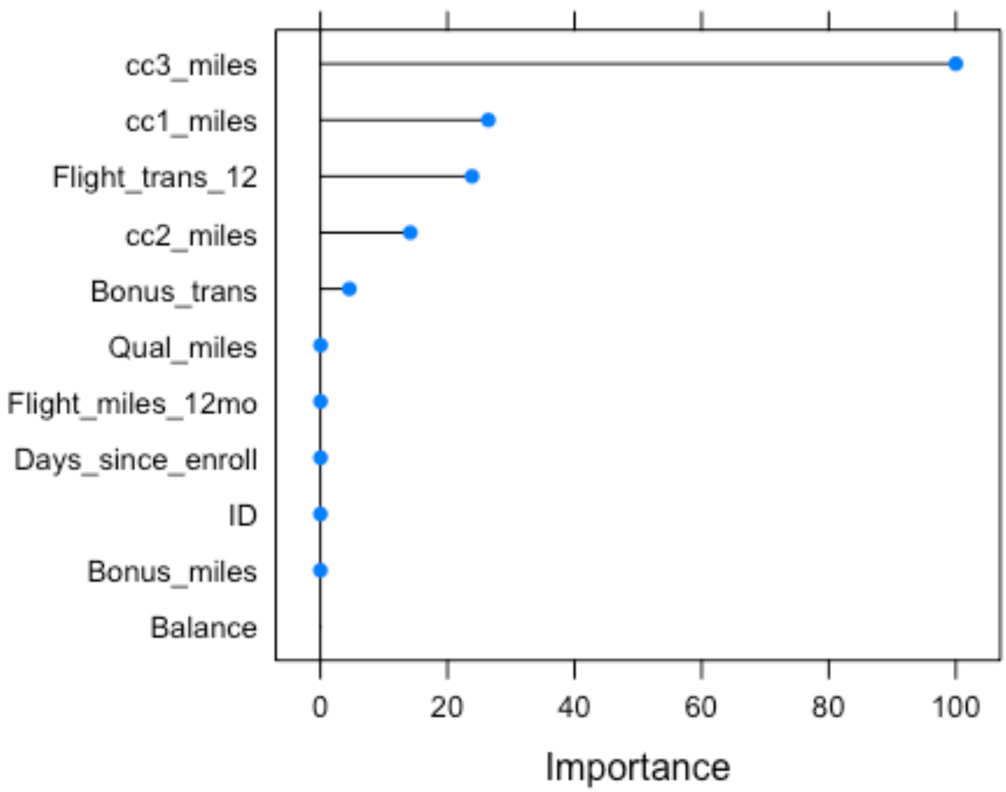
```
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.521
## 2 mae     standard       0.272
```

```
fitRidge <-
  train(Award ~ ., family='binomial', data=dftrain, method='glmnet',
trControl=trainControl(method='cv', number=10), tuneGrid =
expand.grid(alpha=0, lambda=lambdaValues))

resultsfitRidge <-
  fitRidge %>%
  predict(dftest, type='raw') %>%
  bind_cols(dftest, predictedAward=.)

resultsfitRidge %>%
  xtabs(~predictedAward+Award, .) %>%
  confusionMatrix(positive = '1')
```

```
## Confusion Matrix and Statistics
##
##               Award
## predictedAward   0   1
##              0 675 263
##              1  65 197
##
##                Accuracy : 0.7267
##                  95% CI : (0.7005, 0.7517)
##     No Information Rate : 0.6167
##     P-Value [Acc > NIR] : 6.737e-16
##
##                   Kappa : 0.3706
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4283
##             Specificity : 0.9122
##          Pos Pred Value : 0.7519
##          Neg Pred Value : 0.7196
##              Prevalence : 0.3833
##          Detection Rate : 0.1642
##    Detection Prevalence : 0.2183
##       Balanced Accuracy : 0.6702
##
##        'Positive' Class : 1
##
```

```
varImp(fitRidge)$importance %>%
  rownames_to_column(var = "Variable") %>%
```

```r
  mutate(Importance = scales::percent(Overall/100)) %>%
  arrange(desc(Overall)) %>%
  as_tibble()

## # A tibble: 11 x 3
##    Variable           Overall Importance
##    <chr>                <dbl> <chr>
##  1 cc3_miles        100       100.0000%
##  2 cc1_miles         26.4     26.4363%
##  3 Flight_trans_12   23.9     23.8719%
##  4 cc2_miles         14.2     14.1570%
##  5 Bonus_trans        4.57    4.5656%
##  6 Qual_miles         0.0462  0.0462%
##  7 Flight_miles_12mo  0.0280  0.0280%
##  8 Days_since_enroll  0.0247  0.0247%
##  9 ID                 0.0105  0.0105%
## 10 Bonus_miles        0.00484 0.0048%
## 11 Balance            0       0.0000%

plot(varImp(fitRidge))
```




```r
performance(resultsfitRidge, truth = as.numeric(Award), estimate =
as.numeric(predictedAward))
```

```
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.523
## 2 mae     standard       0.273
```

```
resultRandomForest <- train(Award ~ ., data=dftrain, method='ranger',
trControl=trainControl(method='cv', number=10)) %>%
  predict(dftest, type='raw') %>%
  bind_cols(dftest, predictAward=.)

resultRandomForest %>%
  xtabs(~predictAward+Award, .) %>%
  confusionMatrix(positive = '1')
```

```
## Confusion Matrix and Statistics
##
##             Award
## predictAward   0   1
##            0 673 198
##            1  67 262
##
##                Accuracy : 0.7792
##                  95% CI : (0.7546, 0.8023)
##     No Information Rate : 0.6167
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5063
##
##  Mcnemar's Test P-Value : 1.396e-15
##
##             Sensitivity : 0.5696
##             Specificity : 0.9095
##          Pos Pred Value : 0.7964
##          Neg Pred Value : 0.7727
##              Prevalence : 0.3833
##          Detection Rate : 0.2183
##    Detection Prevalence : 0.2742
##       Balanced Accuracy : 0.7395
##
##        'Positive' Class : 1
##
```

#4

```
fitLasso <-
  train(Award ~ cc3_miles+Flight_trans_12+cc2_miles+Bonus_trans,
family='binomial', data=dftrain, method='glmnet',
trControl=trainControl(method='cv', number=10), tuneGrid =
expand.grid(alpha=1, lambda=lambdaValues))
```

```r
resultsLasso <-
  fitLasso %>%
  predict(dftest, type='raw') %>%
  bind_cols(dftest, predictedAward=.)

resultsLasso %>%
  xtabs(~predictedAward+Award, .) %>%
  confusionMatrix(positive = '1')

## Confusion Matrix and Statistics
##
##               Award
## predictedAward   0   1
##              0 683 320
##              1  57 140
##
##                Accuracy : 0.6858
##                  95% CI : (0.6587, 0.712)
##     No Information Rate : 0.6167
##     P-Value [Acc > NIR] : 3.497e-07
##
##                   Kappa : 0.2549
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3043
##             Specificity : 0.9230
##          Pos Pred Value : 0.7107
##          Neg Pred Value : 0.6810
##              Prevalence : 0.3833
##          Detection Rate : 0.1167
##    Detection Prevalence : 0.1642
##       Balanced Accuracy : 0.6137
##
##        'Positive' Class : 1
##

fitRidge2 <-
  train(Award ~ cc3_miles+Flight_trans_12+cc2_miles+cc1_miles+Bonus_trans,
family='binomial', data=dftrain, method='glmnet',
trControl=trainControl(method='cv', number=10), tuneGrid =
expand.grid(alpha=0, lambda=lambdaValues))

resultsRidge2 <-
  fitLasso %>%
  predict(dftest, type='raw') %>%
  bind_cols(dftest, predictedAward=.)

resultsRidge2 %>%
```

```r
xtabs(~predictedAward+Award, .) %>%
confusionMatrix(positive = '1')
```

```
## Confusion Matrix and Statistics
##
##               Award
## predictedAward   0   1
##              0 683 320
##              1  57 140
##
##                Accuracy : 0.6858
##                  95% CI : (0.6587, 0.712)
##     No Information Rate : 0.6167
##     P-Value [Acc > NIR] : 3.497e-07
##
##                   Kappa : 0.2549
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3043
##             Specificity : 0.9230
##          Pos Pred Value : 0.7107
##          Neg Pred Value : 0.6810
##              Prevalence : 0.3833
##          Detection Rate : 0.1167
##    Detection Prevalence : 0.1642
##       Balanced Accuracy : 0.6137
##
##        'Positive' Class : 1
##
```