DATA 622 Assignment 3

CUNY: Spring 2021

Philip Tanofsky

22 March 2021

```
# Import required R libraries
library(tidyverse)
library(vcd)
library(caret)
#library(MASS)
#library(ggplot2)
#library(mvtnorm)
#library(e1071)
#library(klaR)
#library(pROC)
#library(corrplot)
theme_set(theme_classic())
```

```
# Read in loan approval csv
data <- read.csv("https://raw.githubusercontent.com/ptanofsky/data622/main/assignment03/Loan_approval.c

data$Credit_History <- as.factor(data$Credit_History)

data$Total_Income <- data$ApplicantIncome + data$CoapplicantIncome

data$LoanAmt_Per_Month <- data$LoanAmount / data$Loan_Amount_Term

data$Income_To_LoanAmt <- data$Total_Income / data$LoanAmount

data$Income_To_LoanAmtMonth <- data$Total_Income / data$LoanAmt_Per_Month

summary(data)
```

```
##      Loan_ID       Gender    Married   Dependents        Education
##  LP001002:  1         : 13      :  3     : 15     Graduate    :480
##  LP001003:  1   Female:112   No :213   0 :345     Not Graduate:134
##  LP001005:  1   Male  :489   Yes:398   1 :102
##  LP001006:  1                          2 :101
##  LP001008:  1                          3+: 51
##  LP001011:  1
##  (Other) :608
##  Self_Employed ApplicantIncome CoapplicantIncome   LoanAmount
##      : 32       Min.   :  150   Min.   :    0     Min.   :  9.0
##  No :500       1st Qu.: 2878   1st Qu.:    0     1st Qu.:100.0
##  Yes: 82       Median : 3812   Median : 1188     Median :128.0
##                Mean   : 5403   Mean   : 1621     Mean   :146.4
##                3rd Qu.: 5795   3rd Qu.: 2297     3rd Qu.:168.0
##                Max.   :81000   Max.   :41667     Max.   :700.0
##                                                  NA's   :22
##  Loan_Amount_Term Credit_History   Property_Area Loan_Status  Total_Income
##  Min.   : 12      0   : 89       Rural    :179   N:192        Min.   : 1442
##  1st Qu.:360      1   :475       Semiurban:233   Y:422        1st Qu.: 4166
##  Median :360      NA's: 50       Urban    :202                Median : 5416
##  Mean   :342                                                  Mean   : 7025
##  3rd Qu.:360                                                  3rd Qu.: 7522
##  Max.   :480                                                  Max.   :81000
##  NA's   :14
##  LoanAmt_Per_Month Income_To_LoanAmt Income_To_LoanAmtMonth
##  Min.   :0.0250    Min.   : 12.09    Min.   :   808.5
##  1st Qu.:0.2861    1st Qu.: 35.53    1st Qu.: 12233.0
##  Median :0.3653    Median : 41.43    Median : 14469.3
##  Mean   :0.4803    Mean   : 51.23    Mean   : 17241.8
##  3rd Qu.:0.5139    3rd Qu.: 51.78    3rd Qu.: 17992.4
##  Max.   :9.2500    Max.   :396.37    Max.   :142692.0
##  NA's   :36        NA's   :22        NA's   :36
```

```r
dim(data)
```

```
## [1] 614  17
```

Dimensions: 614 observations

13 columns

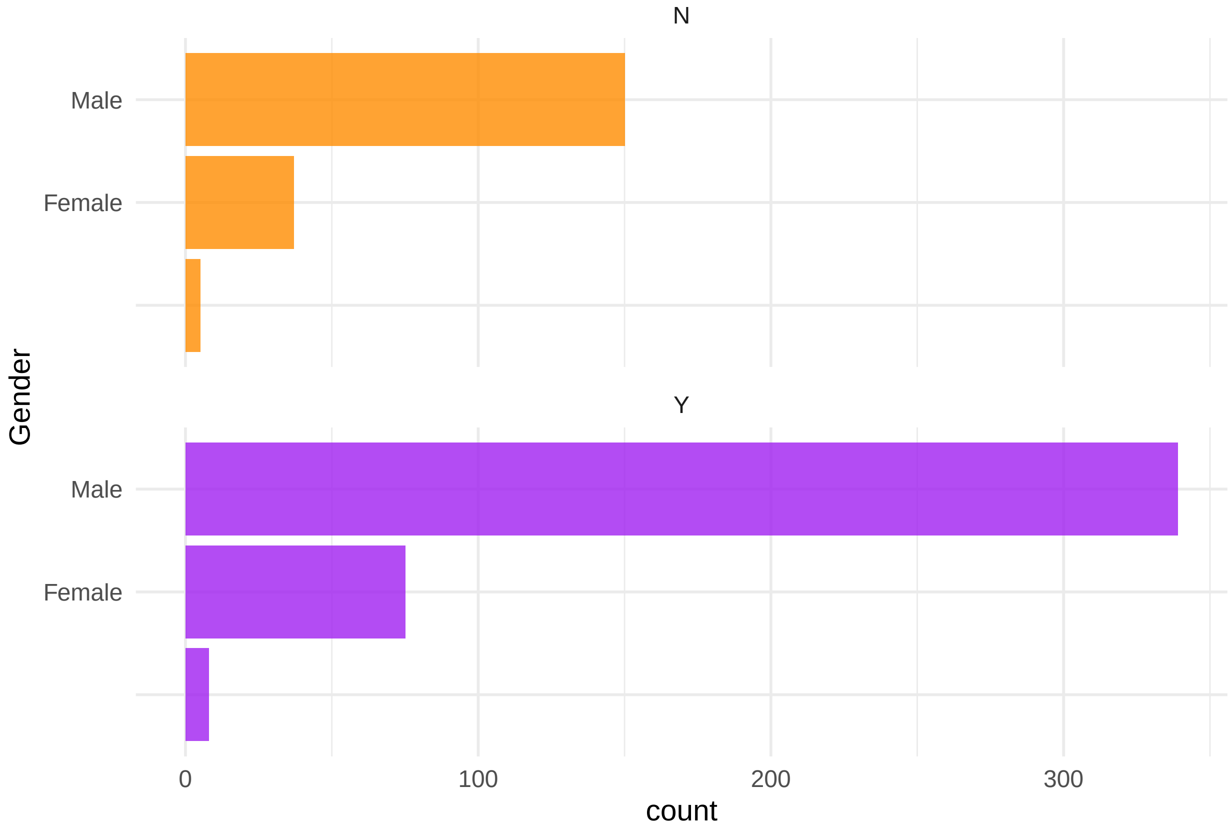
All columns factor except:

ApplicationIncome: int

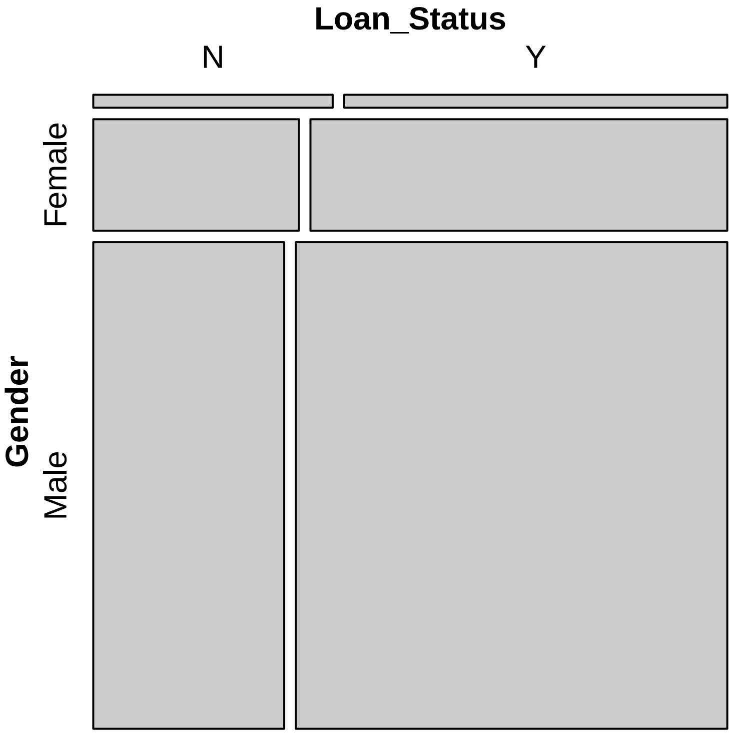
CoapplicantIncome: num LoanAmount: int Loan_Amount_Term: int Credit_History: int, should probably be factor

Loan_ID: Unique identifier Gender: Female|Male Married: No|Yes

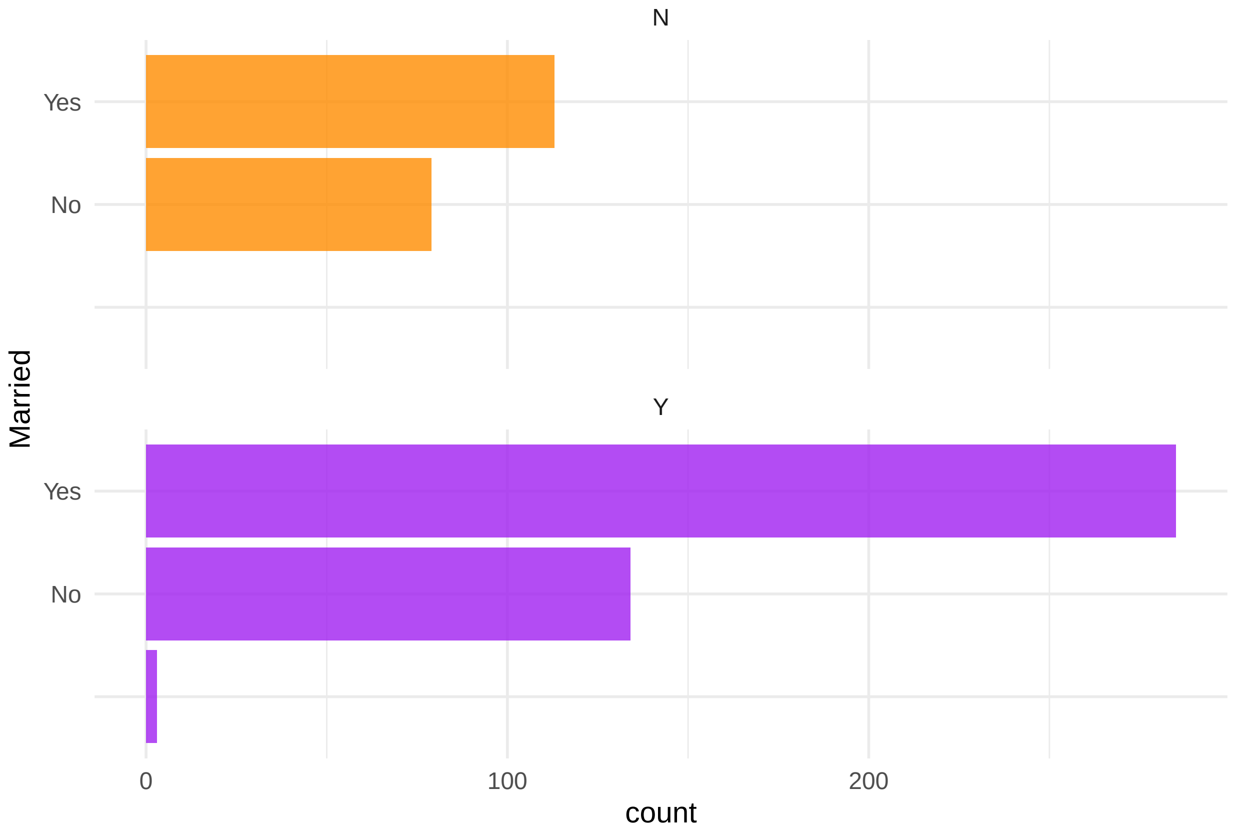
```r
# Count penguins for each loan status / gender
ggplot(data, aes(x = Gender, fill = Loan_Status)) +
  geom_bar(alpha = 0.8) +
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"),
                    guide = F) +
  theme_minimal() +
  facet_wrap(~Loan_Status, ncol = 1) +
  coord_flip()
```



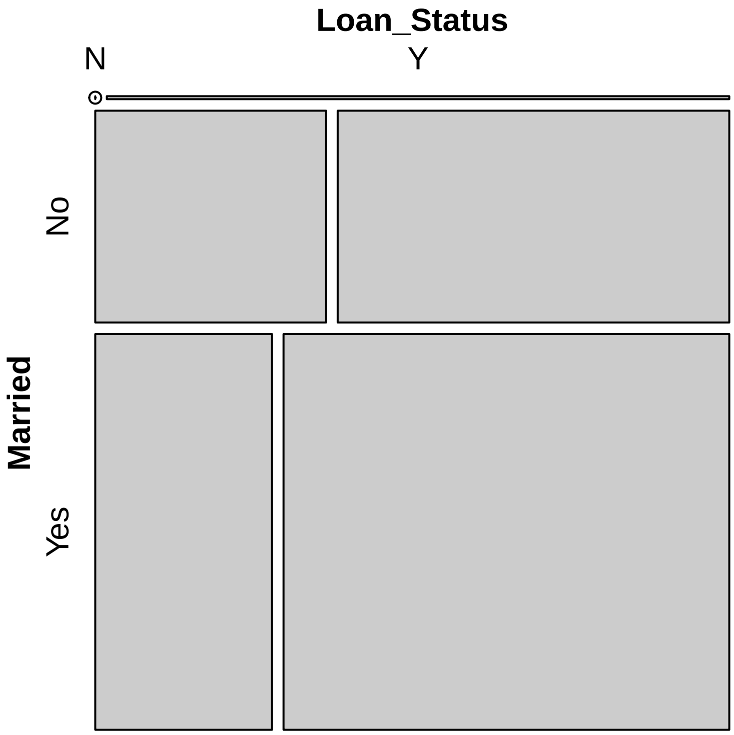
```
mosaic(~ Gender + Loan_Status, data = data)
```



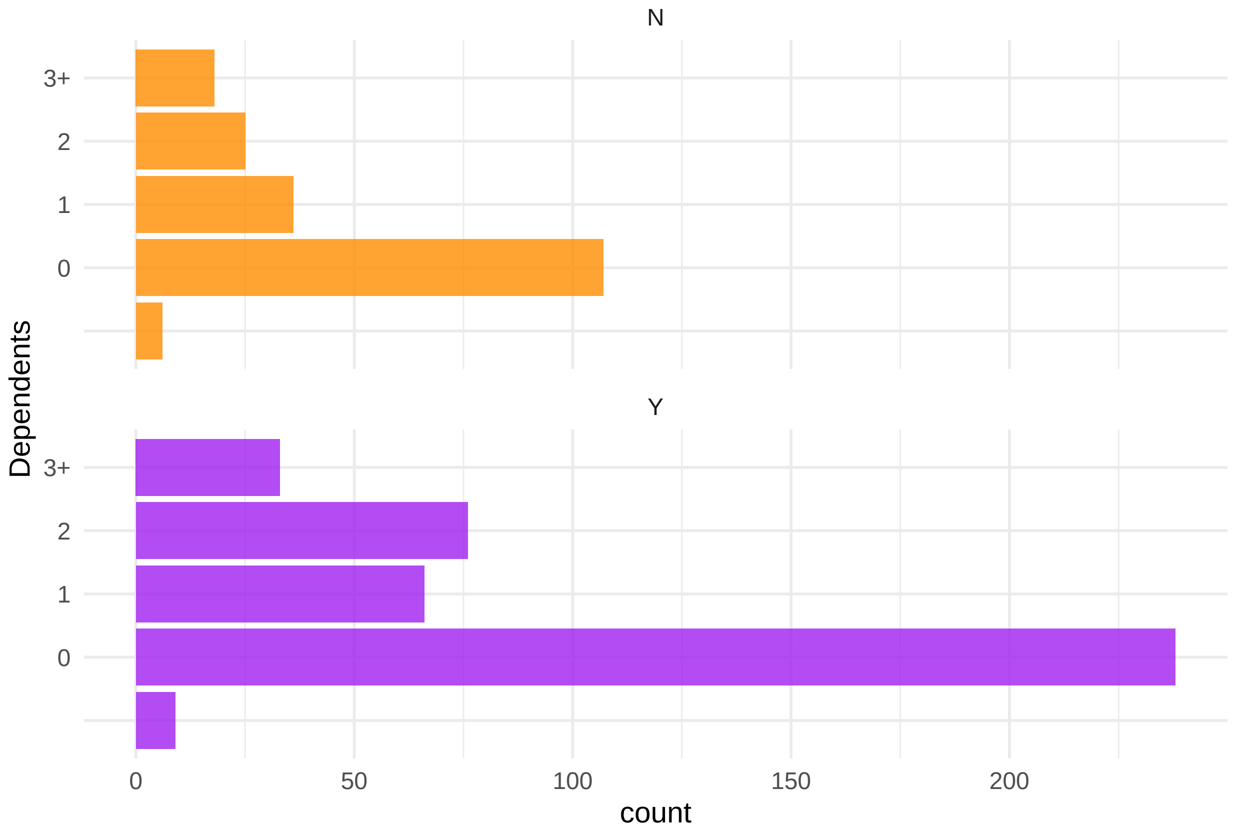
```r
# Count penguins for each loan status / married
ggplot(data, aes(x = Married, fill = Loan_Status)) +
  geom_bar(alpha = 0.8) +
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"),
                    guide = F) +
  theme_minimal() +
  facet_wrap(~Loan_Status, ncol = 1) +
  coord_flip()
```



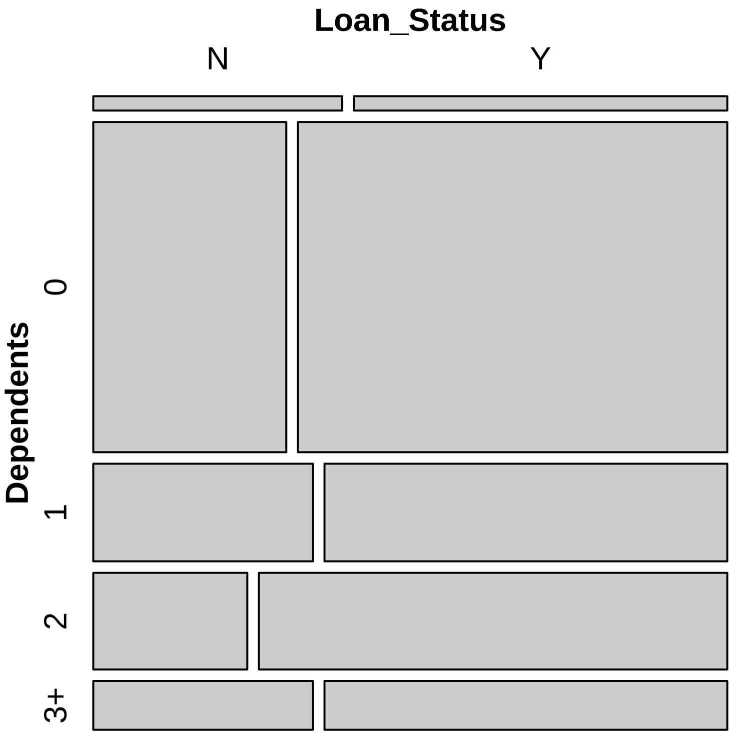
```r
mosaic(~ Married + Loan_Status, data = data)
```



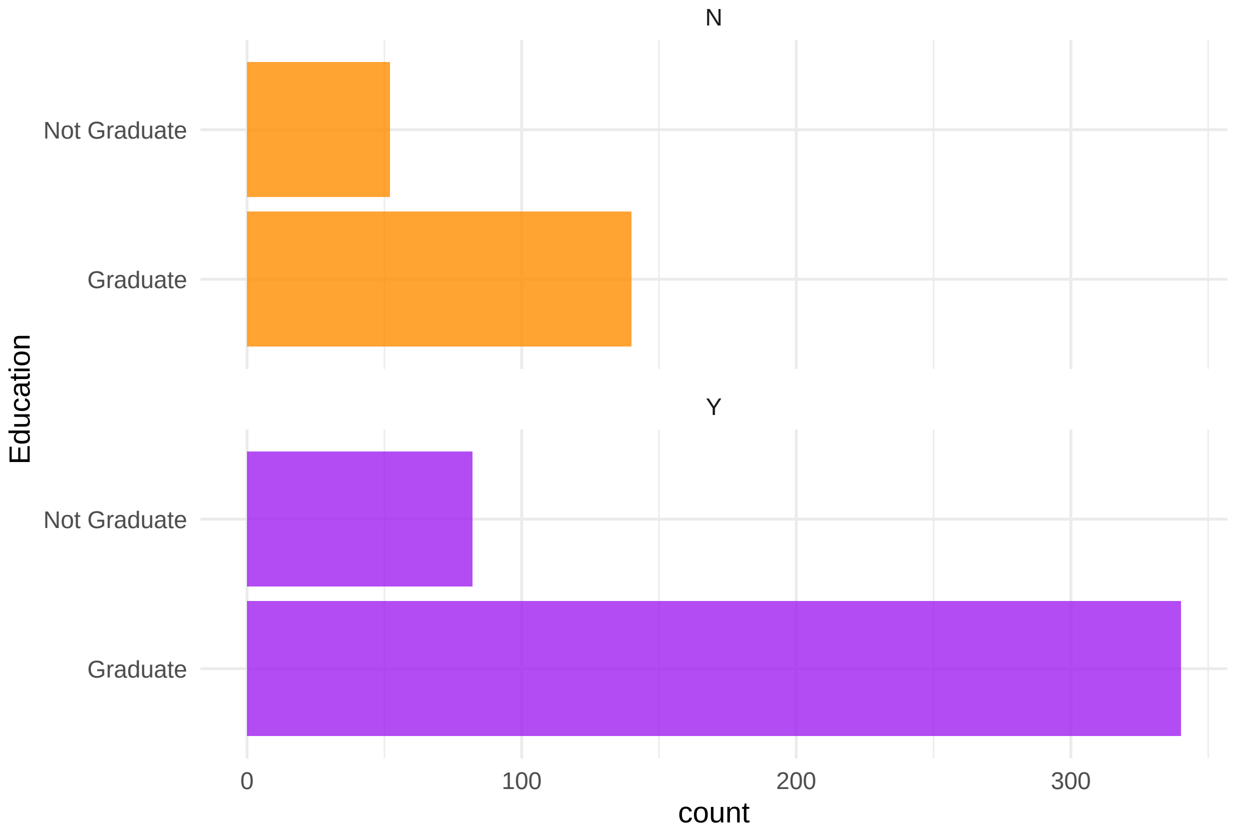
```r
# Count penguins for each loan status / dependents
ggplot(data, aes(x = Dependents, fill = Loan_Status)) +
  geom_bar(alpha = 0.8) +
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"),
                    guide = F) +
  theme_minimal() +
  facet_wrap(~Loan_Status, ncol = 1) +
  coord_flip()
```



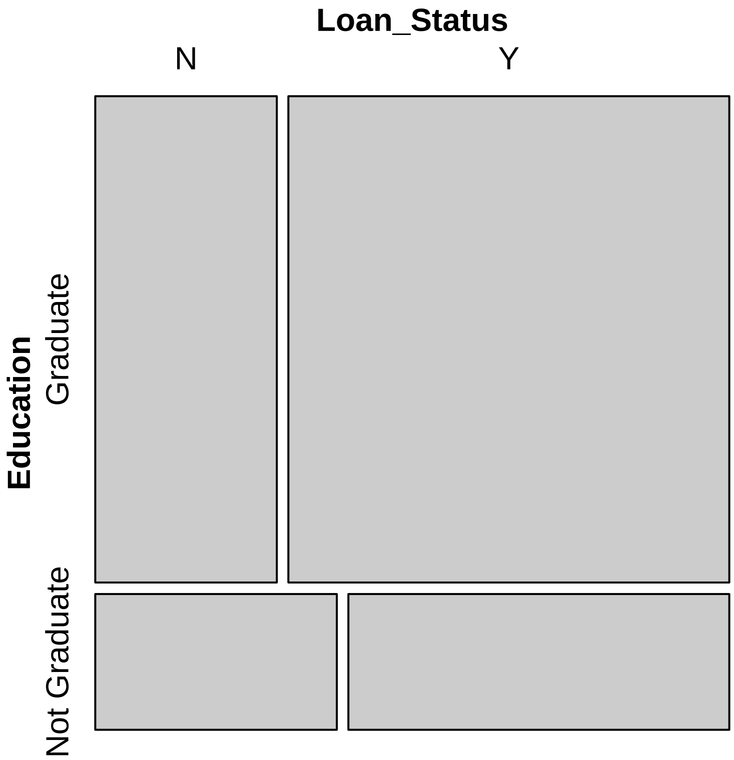
```r
mosaic(~ Dependents + Loan_Status, data = data)
```



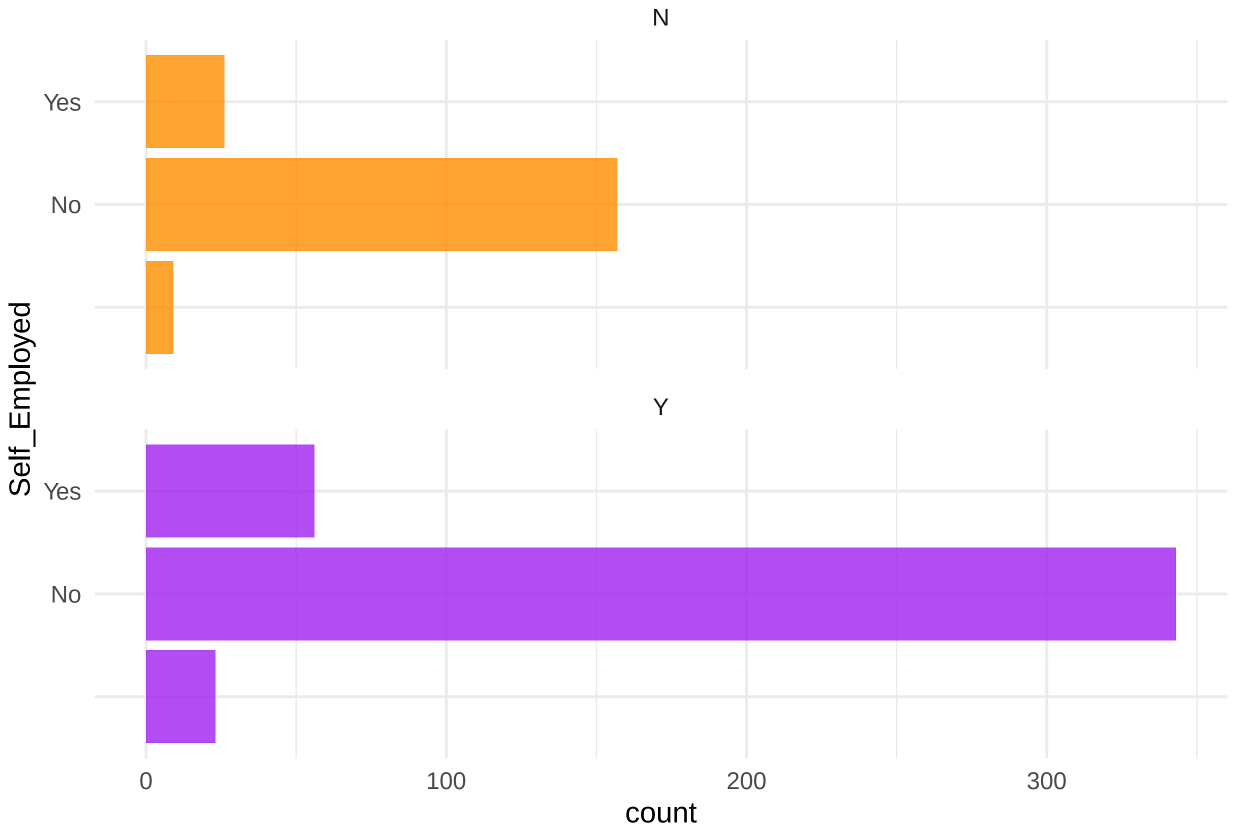
```
# Count penguins for each loan status / Education
ggplot(data, aes(x = Education, fill = Loan_Status)) +
  geom_bar(alpha = 0.8) +
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"),
                    guide = F) +
  theme_minimal() +
  facet_wrap(~Loan_Status, ncol = 1) +
  coord_flip()
```



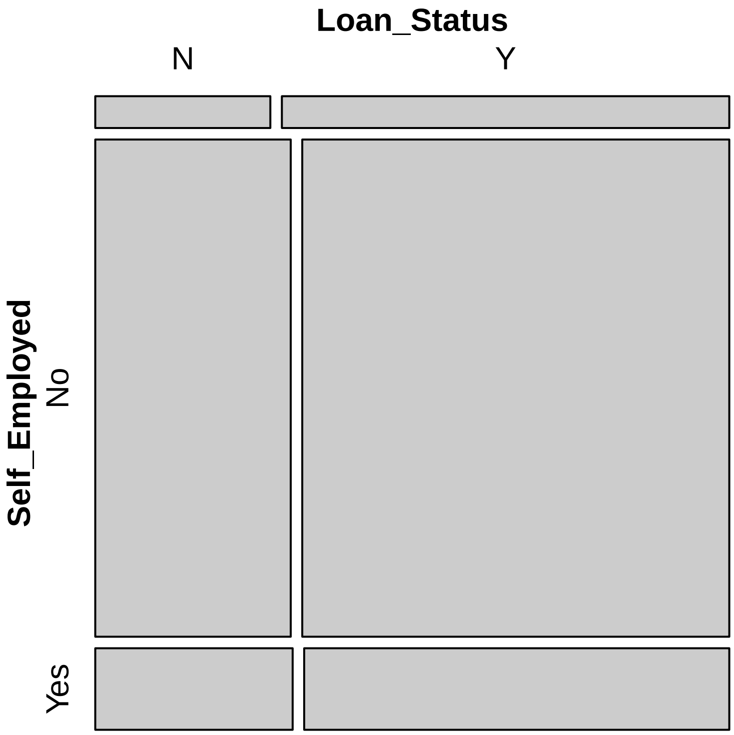
```
mosaic(~ Education + Loan_Status, data = data)
```



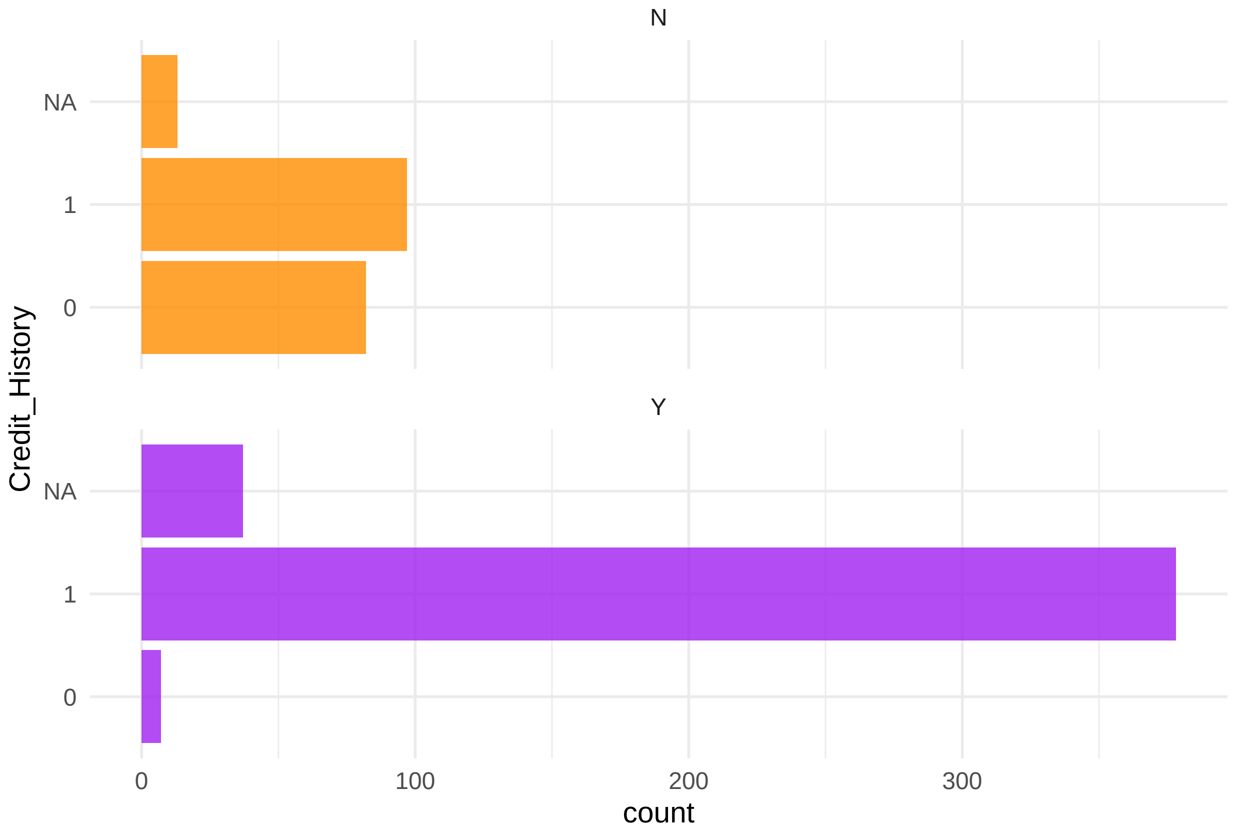
```
# Count penguins for each loan status / Self_Employed
ggplot(data, aes(x = Self_Employed, fill = Loan_Status)) +
  geom_bar(alpha = 0.8) +
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"),
                    guide = F) +
  theme_minimal() +
  facet_wrap(~Loan_Status, ncol = 1) +
  coord_flip()
```



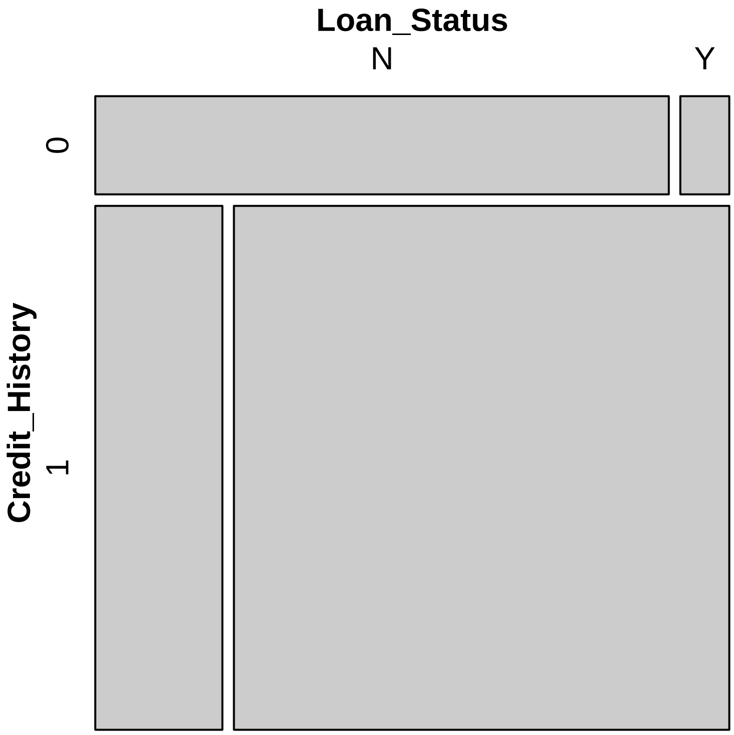
```r
mosaic(~ Self_Employed + Loan_Status, data = data)
```



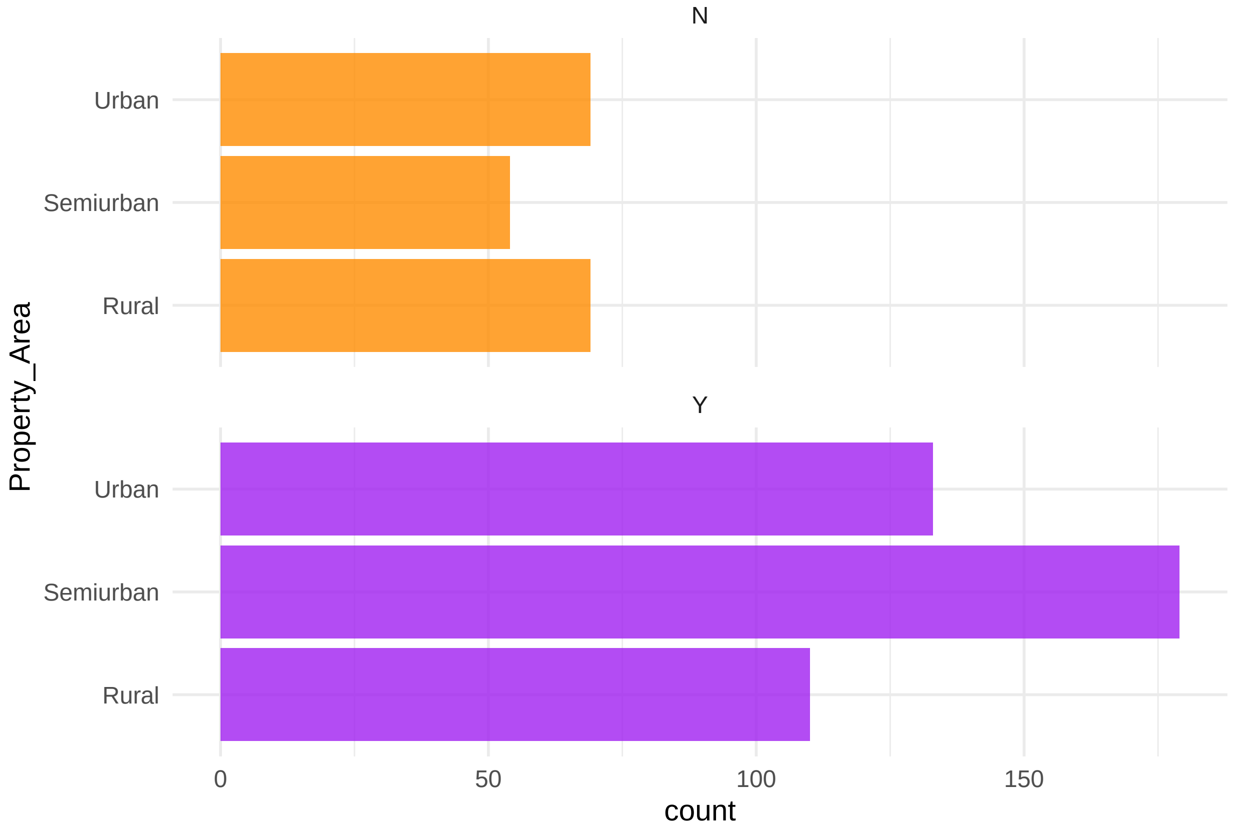
```r
# Count penguins for each loan status / Credit_History
ggplot(data, aes(x = Credit_History, fill = Loan_Status)) +
  geom_bar(alpha = 0.8) +
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"),
                    guide = F) +
  theme_minimal() +
  facet_wrap(~Loan_Status, ncol = 1) +
  coord_flip()
```



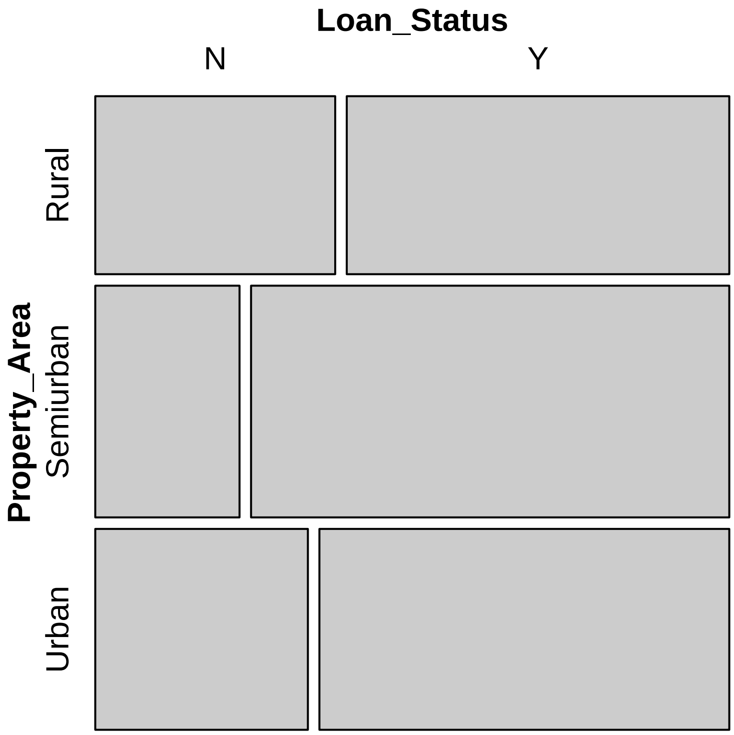
```
mosaic(~ Credit_History + Loan_Status, data = data)
```



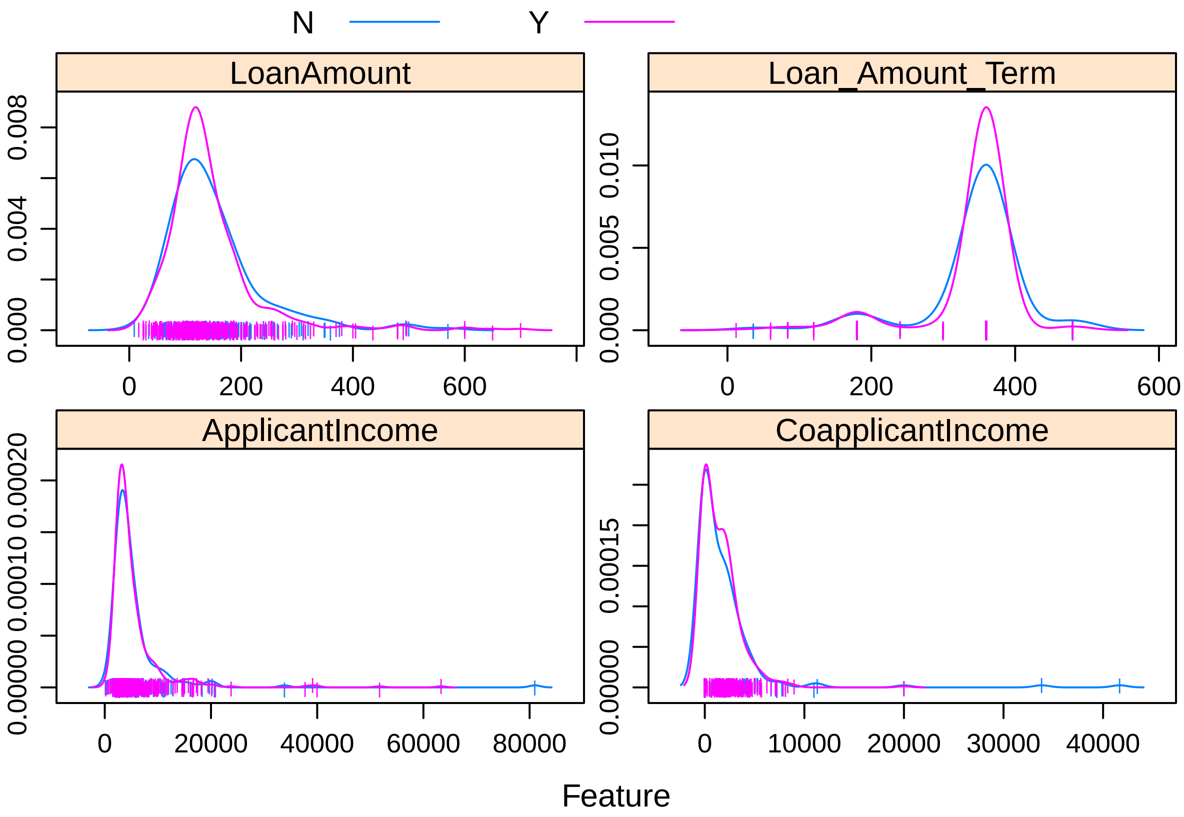
```
# Count penguins for each loan status / Property_Area
ggplot(data, aes(x = Property_Area, fill = Loan_Status)) +
  geom_bar(alpha = 0.8) +
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"),
                    guide = F) +
  theme_minimal() +
  facet_wrap(~Loan_Status, ncol = 1) +
  coord_flip()
```



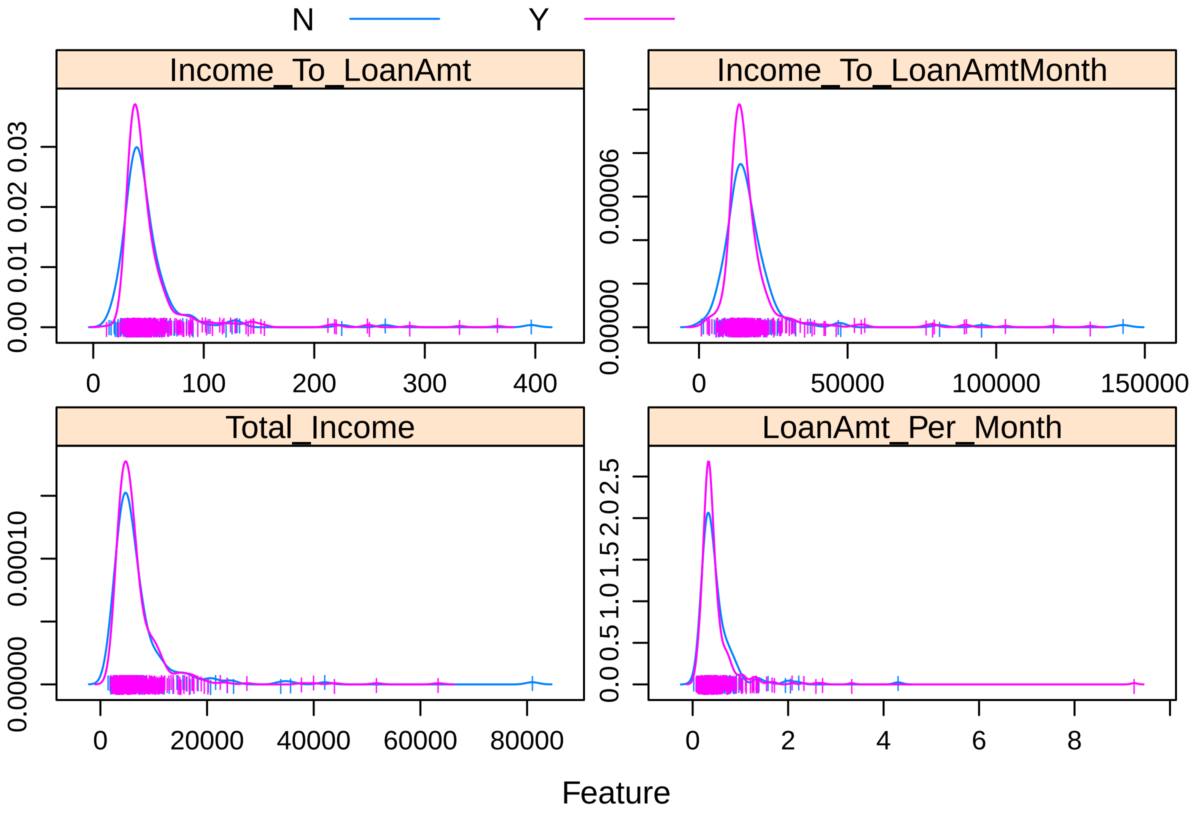
```
mosaic(~ Property_Area + Loan_Status, data = data)
```



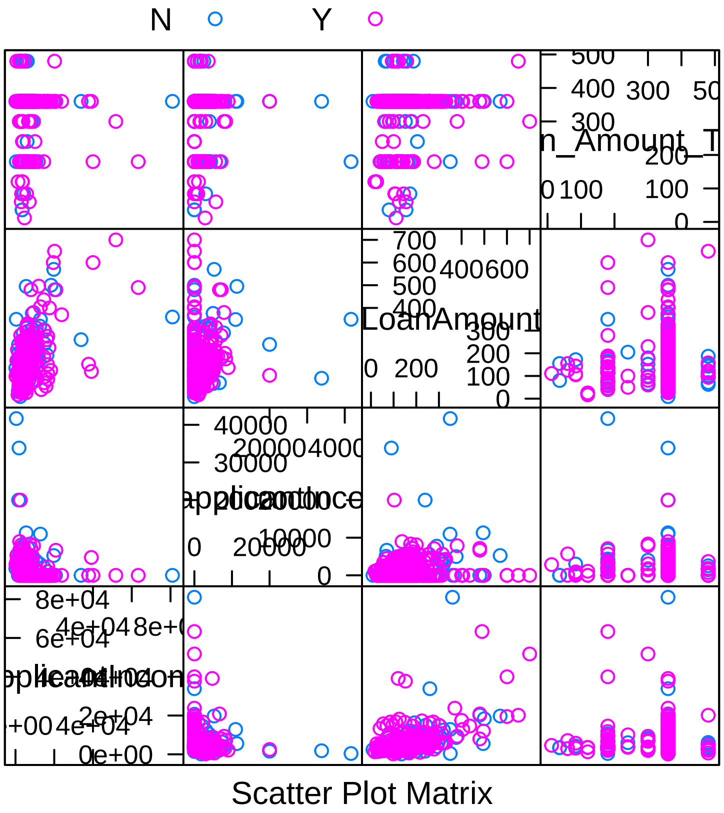
```
# Overlayed density plots
featurePlot(x = data[, 7:10],
            y = data$Loan_Status,
            plot = "density",
            # Pass in options to xyplot() to
            # make it prettier
            scales = list(x = list(relation="free"),
                          y = list(relation="free")),
            adjust = 1.5,
            pch = "|",
            layout = c(2, 2),
            auto.key = list(columns = 3))
```



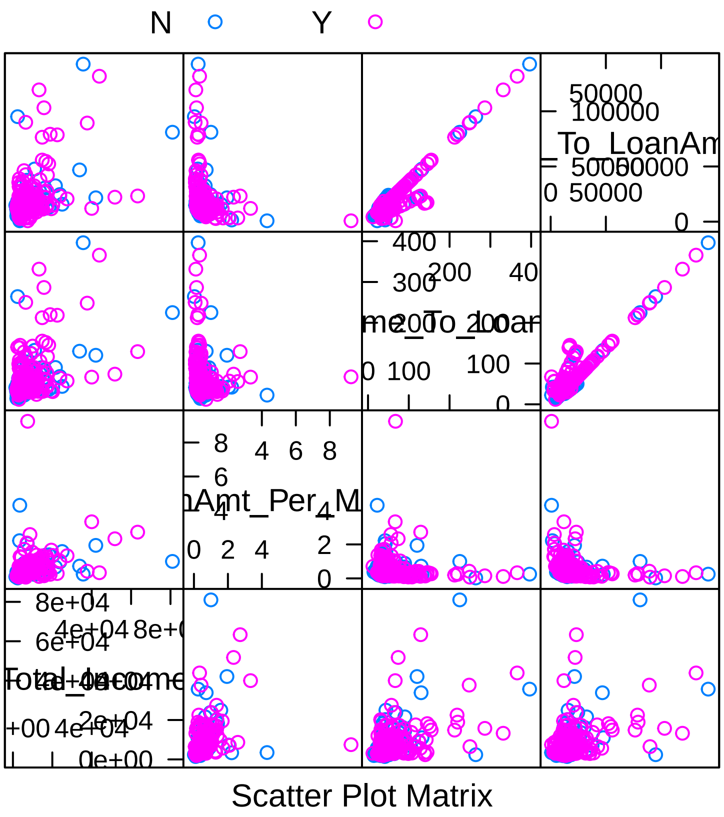
```r
# Overlayed density plots
featurePlot(x = data[, 14:17],
            y = data$Loan_Status,
            plot = "density",
            # Pass in options to xyplot() to
            # make it prettier
            scales = list(x = list(relation="free"),
                          y = list(relation="free")),
            adjust = 1.5,
            pch = "|",
            layout = c(2, 2),
            auto.key = list(columns = 3))
```



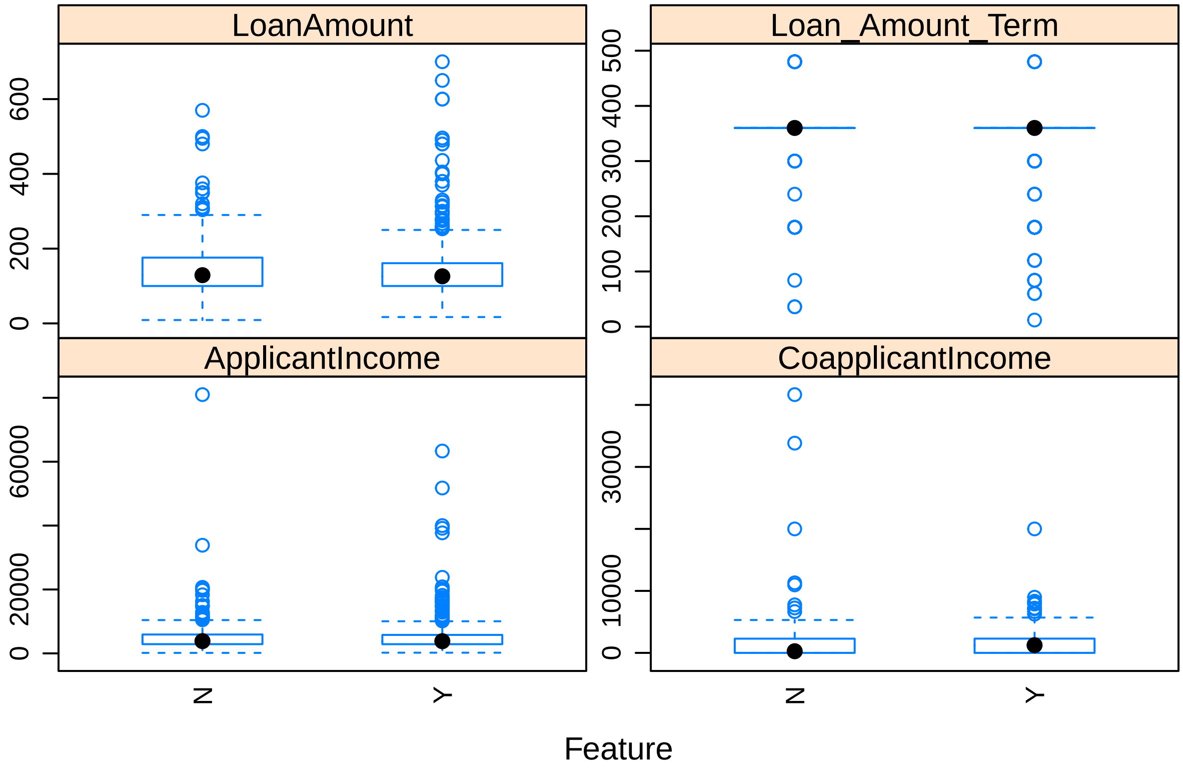
```r
# Use featurePlot
# https://topepo.github.io/caret/visualizations.html

# Scatterplot
featurePlot(x = data[, 7:10],
            y = data$Loan_Status,
            plot = "pairs",
            # Add a key at the top
            auto.key = list(columns = 3))
```



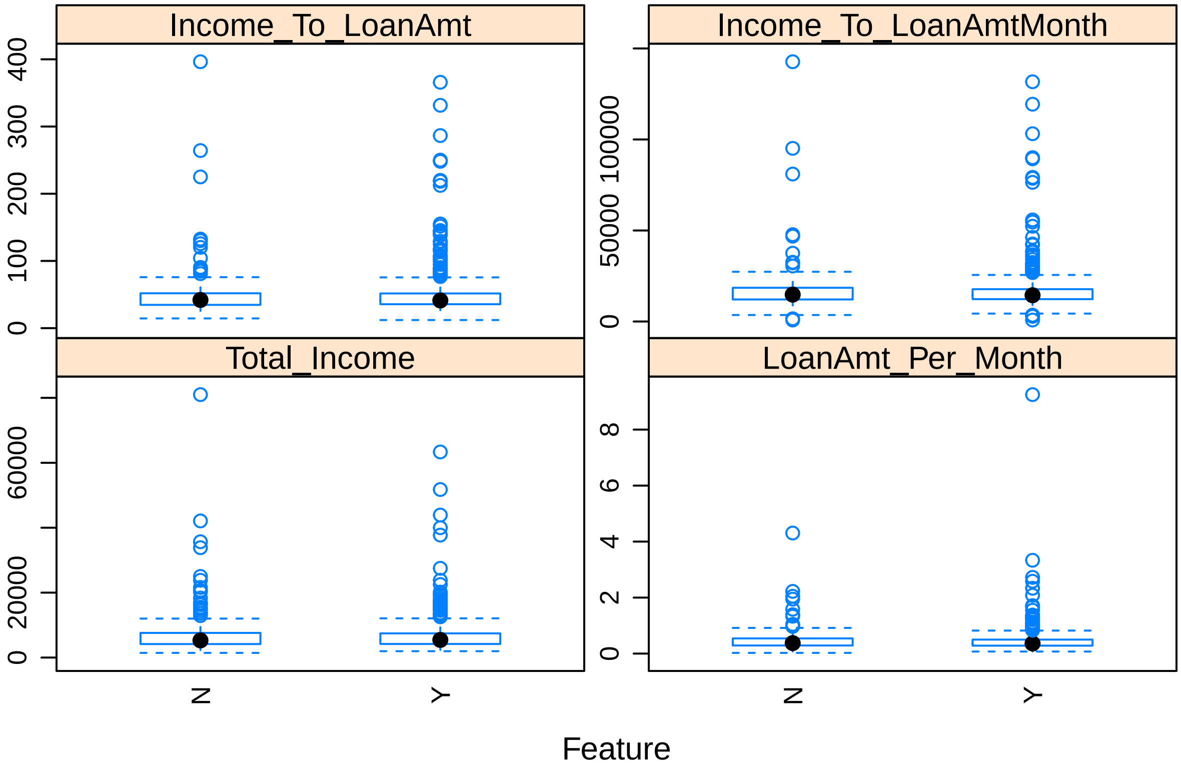
```
featurePlot(x = data[, 14:17],
            y = data$Loan_Status,
            plot = "pairs",
            # Add a key at the top
            auto.key = list(columns = 3))
```



```
featurePlot(x = data[, 7:10],
            y = data$Loan_Status,
            plot = "box",
            ## Pass in options to bwplot()
            scales = list(y = list(relation="free"),
                          x = list(rot = 90)),
            layout = c(2,2),
            auto.key = list(columns = 2))
```



```r
featurePlot(x = data[, 14:17],
            y = data$Loan_Status,
            plot = "box",
            ## Pass in options to bwplot()
            scales = list(y = list(relation="free"),
                          x = list(rot = 90)),
            layout = c(2,2),
            auto.key = list(columns = 2))
```


Income_To_LoanAmt
Income_To_LoanAmtMonth
Total_Income
LoanAmt_Per_Month
0
100
200
300
400
0
50000
100000
0
20000
60000
0
2
4
6
8
N
Y
N
Y
Feature