hw3

shuangyu_zhao

2023-02-03

1.

a.

$$-6 + X_1 \times 0.05 + X_2 \times 1 = Y$$
$$p = \frac{1}{1 + e^{-(-6+40\times0.05+3.5*1)}} = 0.3775$$

So, the probability is 37.75%.

b.

$$p = 0.5 = \frac{1}{1 + e^{-(-6+X_1\times0.05+3.5*1)}}$$
$$-6 + X_1 \times 0.05 + 3.5 * 1 = 0$$
$$X_1 = 50h$$

This student should study 50h.

2. odd

a.

$$p/(1-p) = 0.37$$
$$p = 0.27$$

b.

$$p/(1-p) = 0.16/(1-0.16) = 0.19$$

3.

a.

```r
auto <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/Auto.csv")
head(auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
```

```
## 6  15         8          429        198   4341          10.0   70      1
##                         name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```r
median(auto$mpg)
```

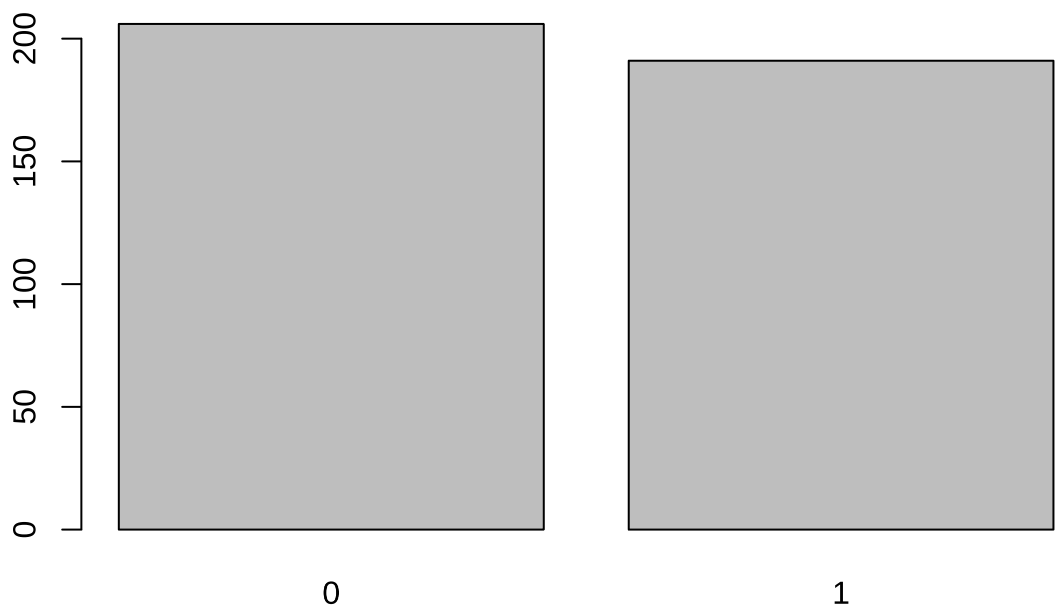
```
## [1] 23
```

```r
auto$mpg01 <- ifelse(auto$mpg > median(auto$mpg), 1, 0)
head(auto, 10)
```

```
##    mpg cylinders displacement horsepower weight acceleration year origin
## 1   18         8          307        130   3504         12.0   70      1
## 2   15         8          350        165   3693         11.5   70      1
## 3   18         8          318        150   3436         11.0   70      1
## 4   16         8          304        150   3433         12.0   70      1
## 5   17         8          302        140   3449         10.5   70      1
## 6   15         8          429        198   4341         10.0   70      1
## 7   14         8          454        220   4354          9.0   70      1
## 8   14         8          440        215   4312          8.5   70      1
## 9   14         8          455        225   4425         10.0   70      1
## 10  15         8          390        190   3850          8.5   70      1
##                          name mpg01
## 1  chevrolet chevelle malibu     0
## 2          buick skylark 320     0
## 3         plymouth satellite     0
## 4              amc rebel sst     0
## 5                ford torino     0
## 6           ford galaxie 500     0
## 7           chevrolet impala     0
## 8          plymouth fury iii     0
## 9           pontiac catalina     0
## 10         amc ambassador dpl     0
```

b.

```r
barplot(table(auto$mpg01))
```



the data are balanced.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
glimpse(auto)
```

```
## Rows: 397
## Columns: 10
## $ mpg          <dbl> 18, 15, 18, 16, 17, 15, 14, 14, 14, 15, 15, 14, 15, 14, 2~
## $ cylinders    <int> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 6, 6, 6, 4, ~
## $ displacement <dbl> 307, 350, 318, 304, 302, 429, 454, 440, 455, 390, 383, 34~
## $ horsepower   <chr> "130", "165", "150", "150", "140", "198", "220", "215", "~
## $ weight       <int> 3504, 3693, 3436, 3433, 3449, 4341, 4354, 4312, 4425, 385~
## $ acceleration <dbl> 12.0, 11.5, 11.0, 12.0, 10.5, 10.0, 9.0, 8.5, 10.0, 8.5, ~
## $ year         <int> 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 7~
## $ origin       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 3, ~
## $ name         <chr> "chevrolet chevelle malibu", "buick skylark 320", "plymou~
## $ mpg01        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, ~
```

numeric: mpg, displacement, horsepower, weight, acceleration, year categoric: cylinders, origin, name

```
auto$horsepower <- as.numeric(auto$horsepower)
```

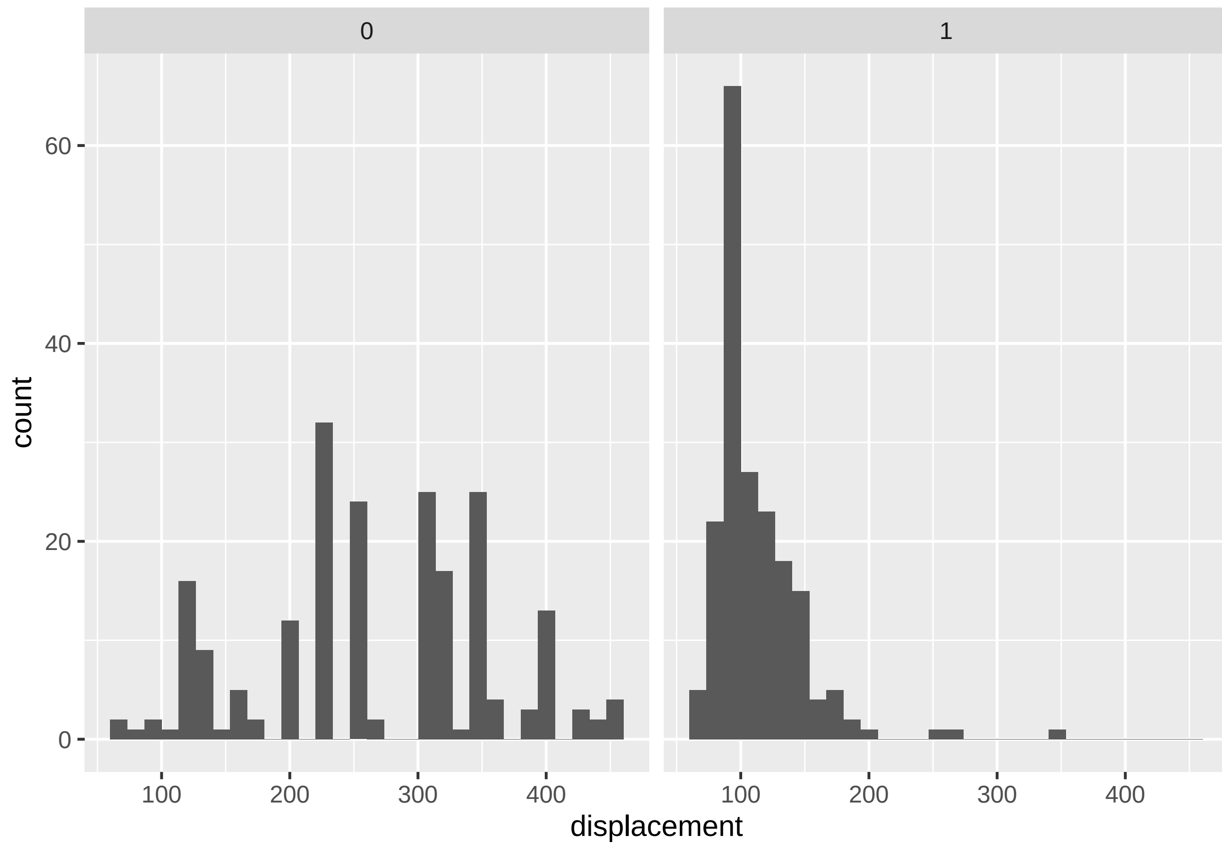
```
## Warning: NAs introduced by coercion
```

```
glimpse(auto)
```

```
## Rows: 397
## Columns: 10
## $ mpg          <dbl> 18, 15, 18, 16, 17, 15, 14, 14, 14, 15, 15, 14, 15, 14, 2~
## $ cylinders    <int> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 6, 6, 6, 4, ~
## $ displacement <dbl> 307, 350, 318, 304, 302, 429, 454, 440, 455, 390, 383, 34~
## $ horsepower   <dbl> 130, 165, 150, 150, 140, 198, 220, 215, 225, 190, 170, 16~
## $ weight       <int> 3504, 3693, 3436, 3433, 3449, 4341, 4354, 4312, 4425, 385~
## $ acceleration <dbl> 12.0, 11.5, 11.0, 12.0, 10.5, 10.0, 9.0, 8.5, 10.0, 8.5, ~
## $ year         <int> 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 7~
## $ origin       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 3, ~
## $ name         <chr> "chevrolet chevelle malibu", "buick skylark 320", "plymou~
## $ mpg01        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, ~
```

for numeric

```
# displacement and mpg01
library(ggplot2)
ggplot(data =auto, aes(x = displacement)) + geom_histogram(bins = 30) + facet_grid(.~mpg01)
```




```
quantile(filter(auto, mpg01 == 1)$displacement, seq(0,1, by=0.1))
```

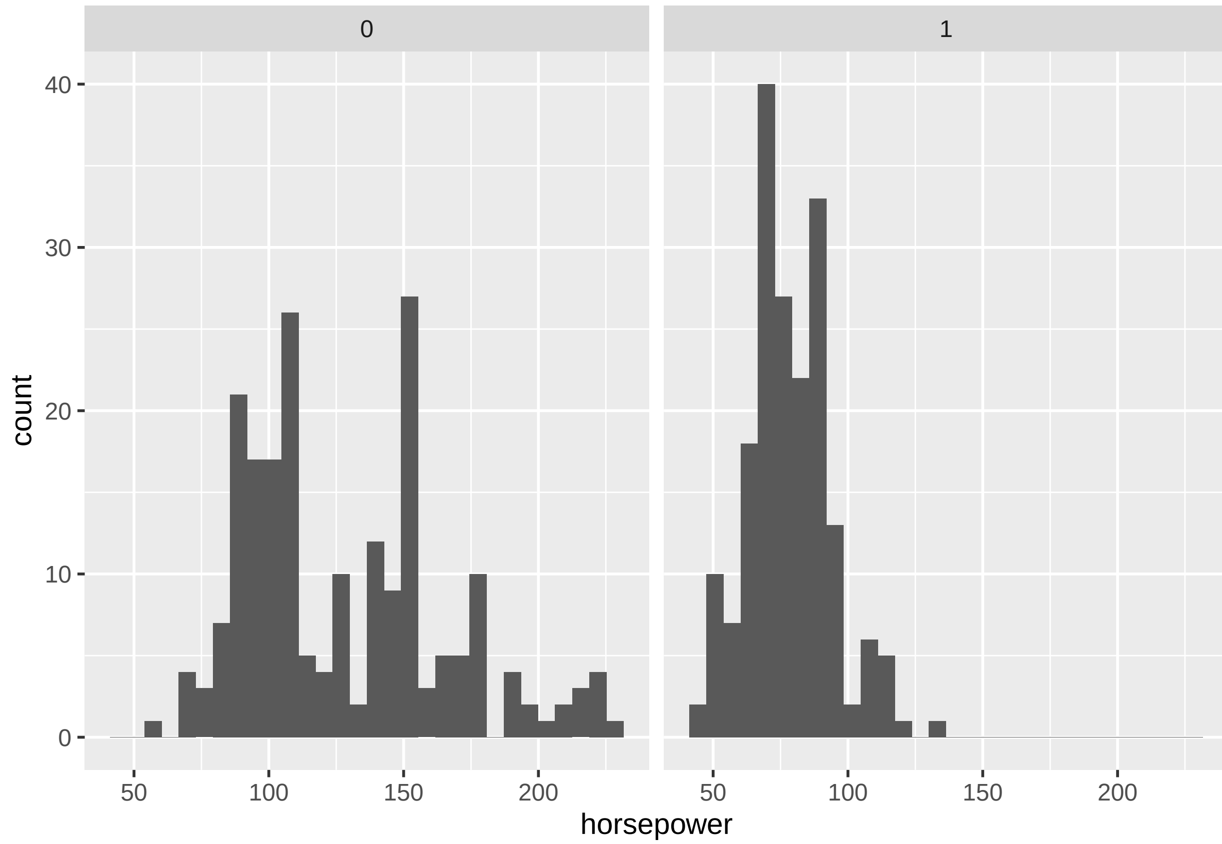
```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   68   85   90   97   98  105  112  120  140  151  350
```

```
quantile(filter(auto, mpg01 == 0)$displacement, seq(0,1, by=0.1))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   70  122  198  225  232  258  304  318  350  400  455
```

```
# horsepower and mpg01
ggplot(data =auto, aes(x = horsepower)) + geom_histogram(bins = 30) + facet_grid(.~mpg01)
```

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```



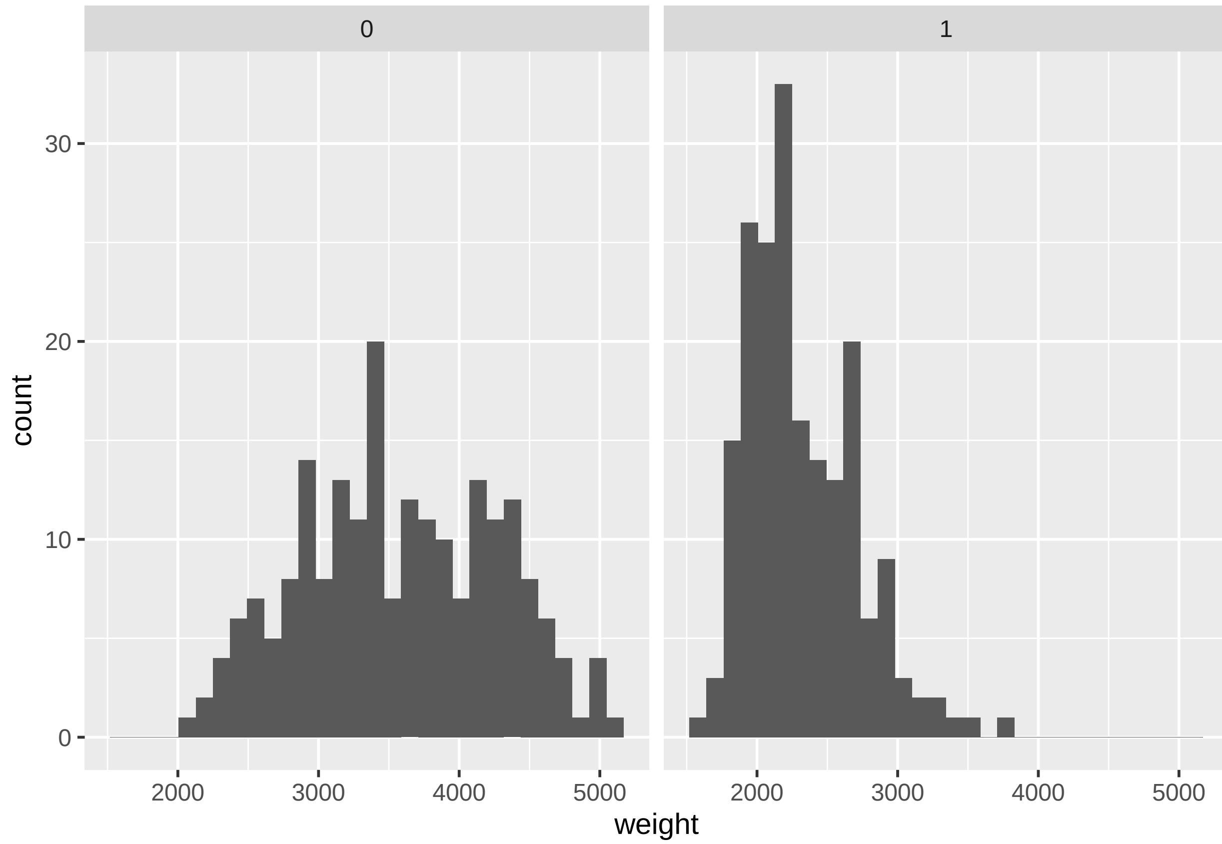

```
quantile(filter(auto, mpg01 == 1)$horsepower, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##  46.0  60.6  67.0  69.0  71.4  75.0  81.0  88.0  90.0  96.4 132.0
```

```
quantile(filter(auto, mpg01 == 0)$horsepower, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   54   88   95  100  110  120  140  150  155  180  230
```

```
# weight and mpg01
ggplot(data =auto, aes(x = weight)) + geom_histogram(bins = 30) + facet_grid(.~mpg01)
```



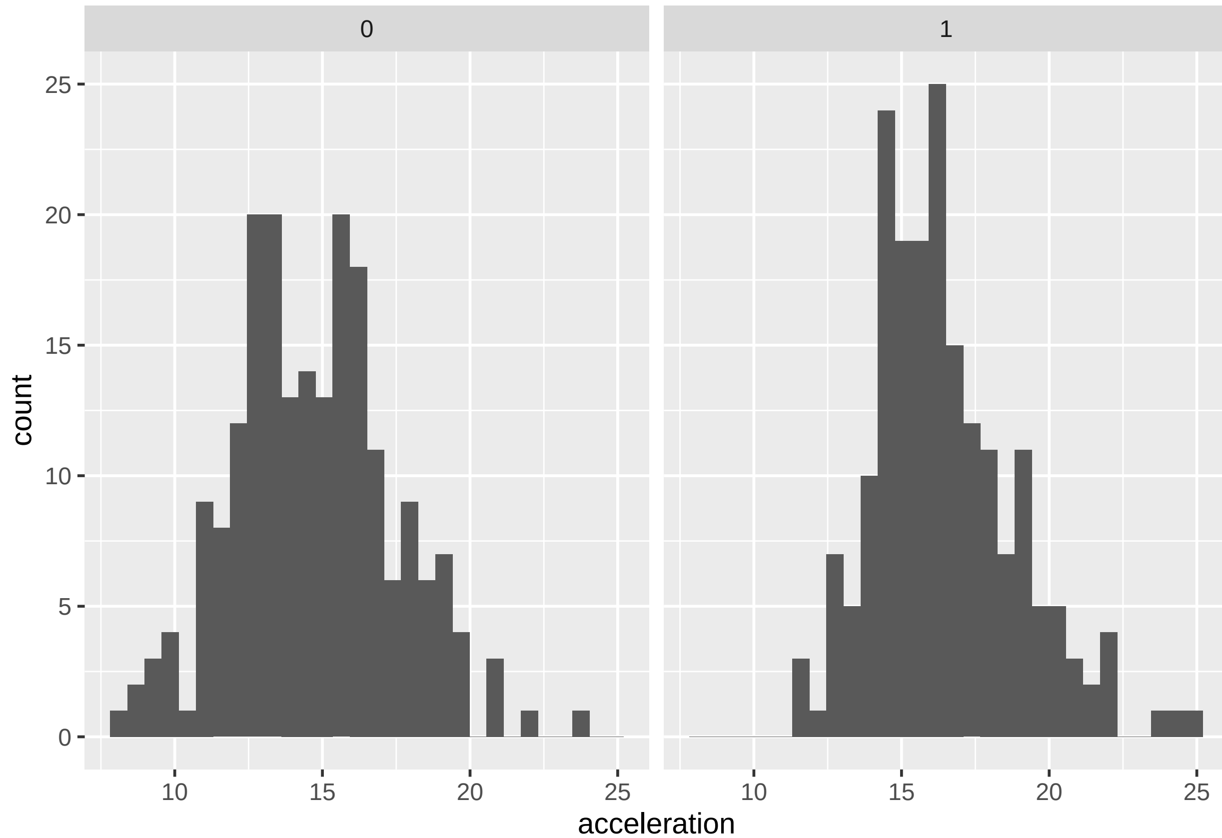

```
quantile(filter(auto, mpg01 == 1)$weight, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
## 1613 1915 1985 2074 2145 2219 2300 2500 2660 2855 3725
```

```
quantile(filter(auto, mpg01 == 0)$weight, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##     0%    10%    20%    30%    40%    50%    60%    70%    80%    90%   100%
## 2124.0 2636.5 2945.0 3163.5 3380.0 3549.0 3777.0 4054.5 4257.0 4460.5 5140.0
```

```
# acceleration and mpg01
ggplot(data =auto, aes(x = acceleration)) + geom_histogram(bins = 30) + facet_grid(.~mpg01)
```



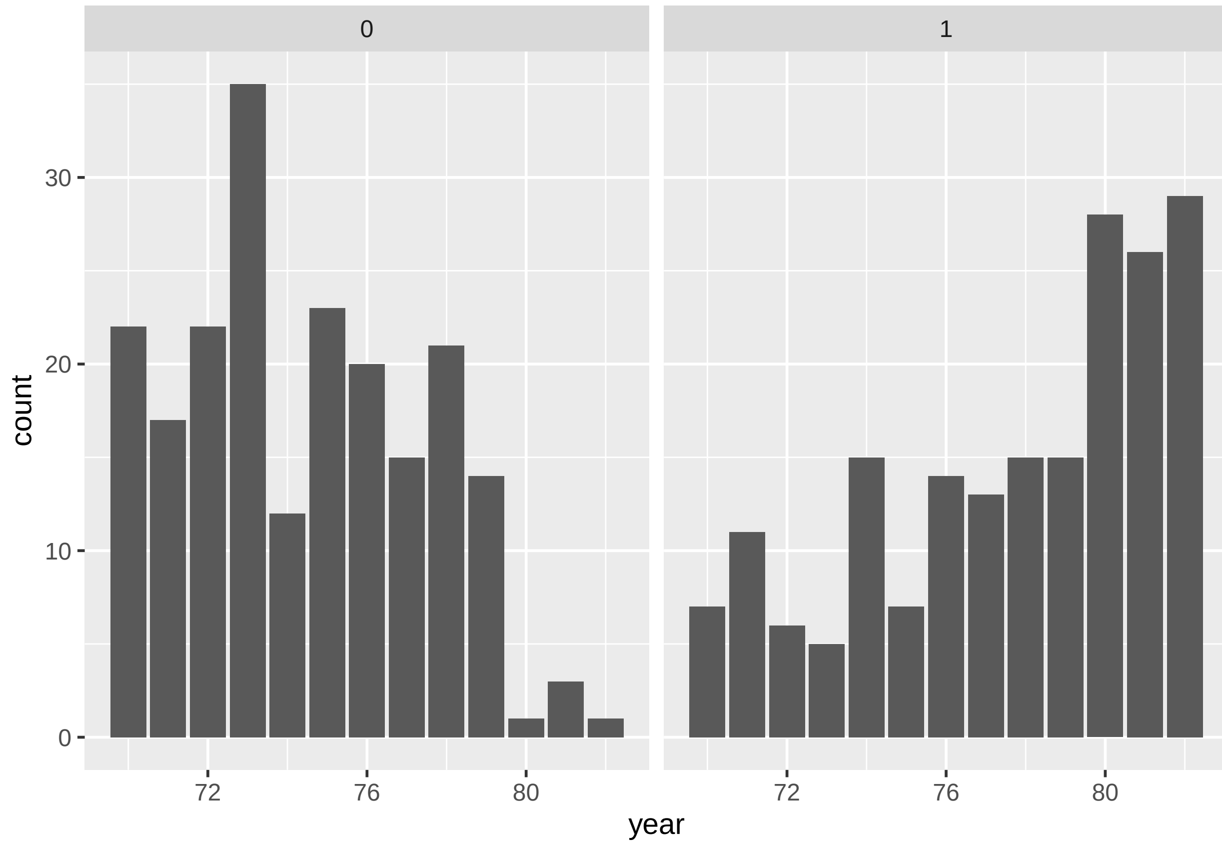

```
quantile(filter(auto, mpg01 == 1)$acceleration, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
## 11.3 14.0 14.5 15.0 15.5 16.2 16.7 17.5 18.5 19.6 24.8
```

```
quantile(filter(auto, mpg01 == 0)$acceleration, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##  8.00 11.45 12.50 13.20 13.90 14.50 15.50 16.00 17.00 18.50 23.50
```

```
ggplot(data = auto, aes(x = year)) + geom_bar() + facet_grid(.~mpg01)
```




```
quantile(filter(auto, mpg01 == 1)$year, seq(0,1, by=0.1), na.rm=TRUE)
```

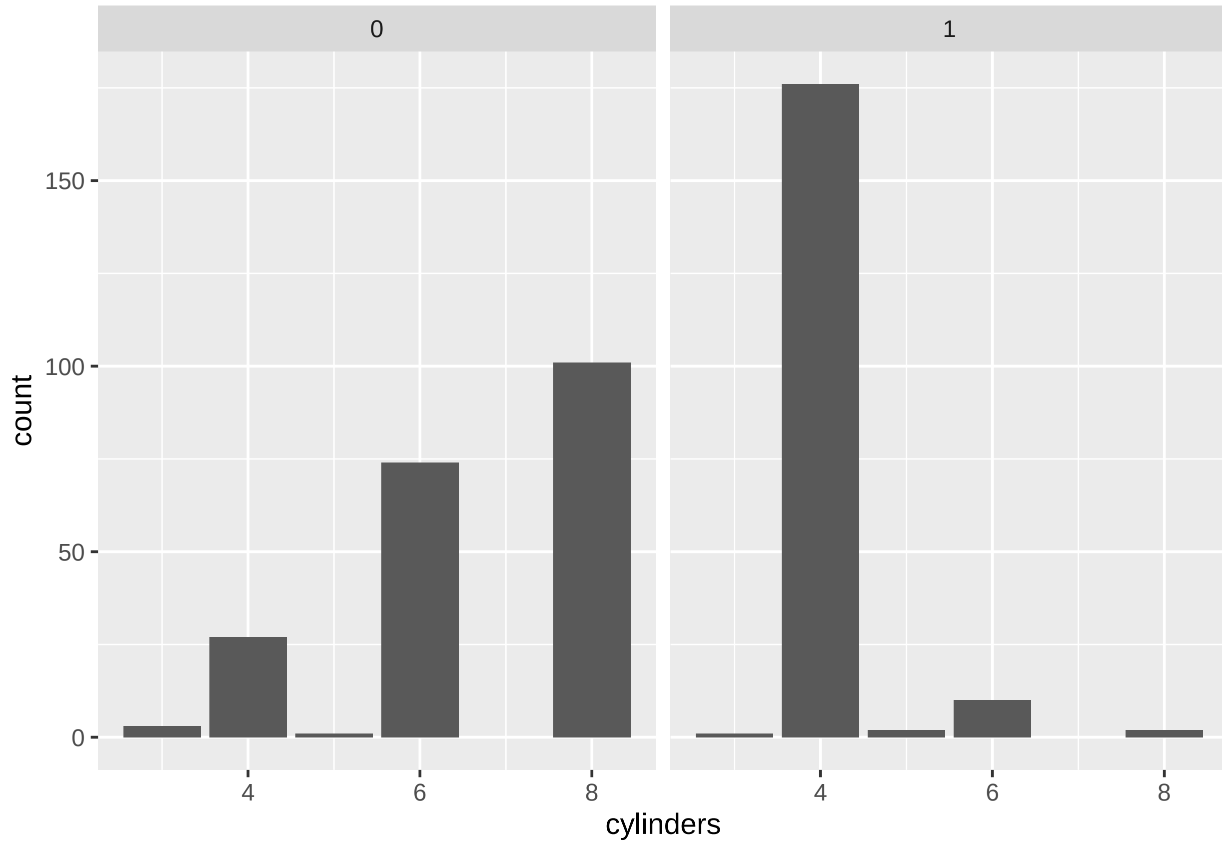
```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100% 
##   70   72   74   76   77   79   80   80   81   82   82
```

```
quantile(filter(auto, mpg01 == 0)$year, seq(0,1, by=0.1), na.rm=TRUE)
```

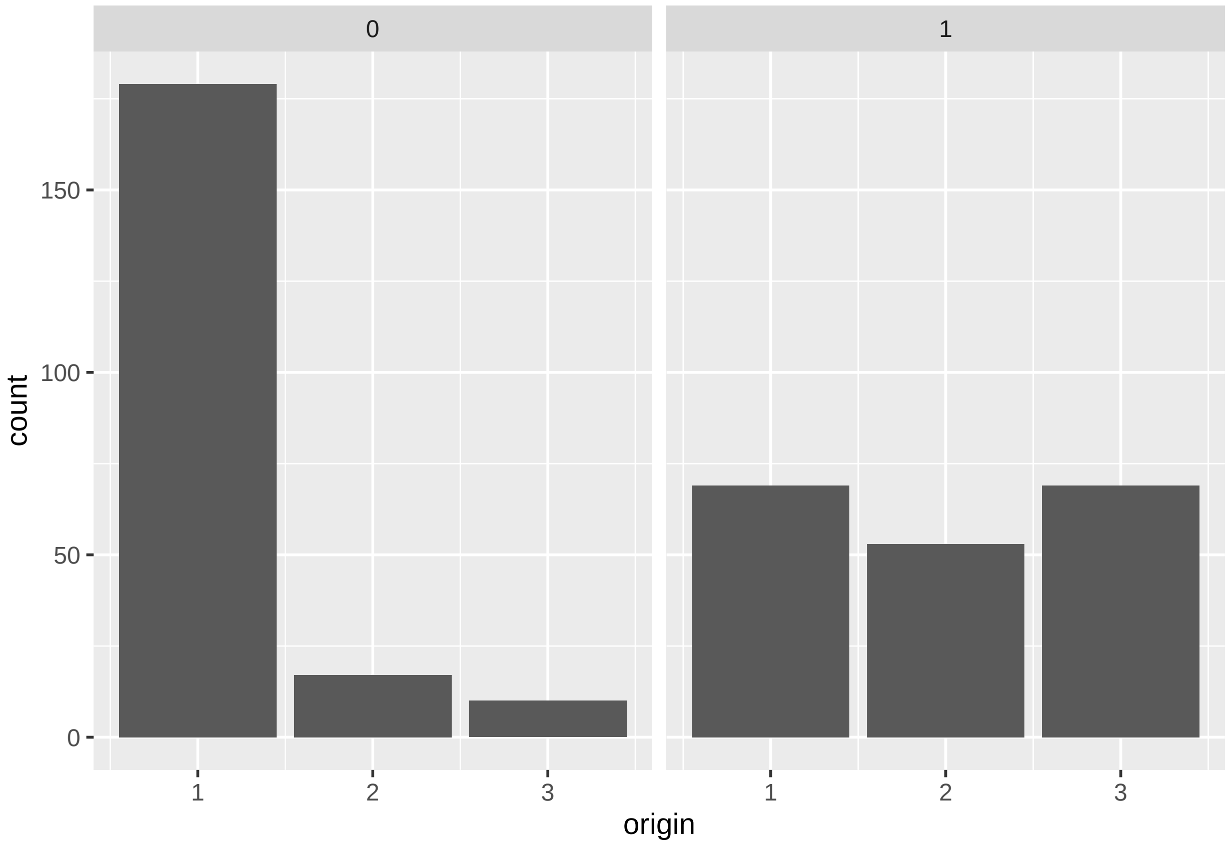
```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100% 
##   70   70   72   73   73   74   75   76   77   78   82
```

for categorical

```
# cylinders and mpg01
ggplot(data = auto, aes(x = cylinders)) + geom_bar() + facet_grid(.~mpg01)
```




```
ggplot(data = auto, aes(x = origin)) + geom_bar() + facet_grid(.~mpg01)
```



displacement, horsepower, weight, cylinder,origin are useful for prediction

c.

```
# train-test split

split_pro <- 0.75
n <- length(auto$mpg)*split_pro
row_samp <- sample(1:length(auto$mpg), n, replace = FALSE)
train <- auto[row_samp,]
test <- auto[-row_samp,]
```

d.

```
mod <- glm(data = train, mpg01 ~ displacement + horsepower + weight + acceleration + year+ cylinders + o
summary(mod)
```

```
##
## Call:
## glm(formula = mpg01 ~ displacement + horsepower + weight + acceleration +
##     year + cylinders + origin, family = binomial, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.77292  -0.11452  -0.00128   0.18824   2.14976
```

```
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -25.126486   7.458874  -3.369 0.000755 ***
## displacement   0.013676   0.015045   0.909 0.363336
## horsepower    -0.018483   0.028090  -0.658 0.510545
## weight        -0.005573   0.001523  -3.658 0.000254 ***
## acceleration   0.070887   0.169481   0.418 0.675757
## year           0.527222   0.099572   5.295 1.19e-07 ***
## cylinders     -0.506304   0.502847  -1.007 0.313994
## origin         0.913878   0.453226   2.016 0.043760 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 403.92  on 291  degrees of freedom
## Residual deviance: 109.45  on 284  degrees of freedom
##   (5 observations deleted due to missingness)
## AIC: 125.45
##
## Number of Fisher Scoring iterations: 8
```

```r
mod2 <- glm(data = train, mpg01 ~ weight + year , family = binomial)
summary(mod2)
```

```
##
## Call:
## glm(formula = mpg01 ~ weight + year, family = binomial, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.32239  -0.12854  -0.00144   0.20613   2.37195
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -23.396340   5.543890  -4.220 2.44e-05 ***
## weight       -0.005730   0.000793  -7.226 4.97e-13 ***
## year          0.518387   0.089844   5.770 7.94e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 411.16  on 296  degrees of freedom
## Residual deviance: 118.41  on 294  degrees of freedom
## AIC: 124.41
##
## Number of Fisher Scoring iterations: 7
```

```r
prediction <- predict(mod2, test, type = "response")
cofm <- confusionMatrix(data =as.factor(as.integer(2*prediction)), reference = as.factor(test$mpg01))
```

```r
test_error <- 1-cofm$overall["Accuracy"]
print(paste0("test error: ", test_error))
```

```
## [1] "test error: 0.09"
```

e.

```r
p <- 1/(1 + exp(-(mod2$coefficients[1] + mod2$coefficients[2]*test$weight + mod2$coefficients[3]*test$y
prediction_direct <- ifelse(p<0.5, 0, 1)
```

```r
prediction_direct
```

```
##   [1] 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0
##  [38] 0 0 1 1 0 1 1 0 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0
##  [75] 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```r
confusionMatrix(data = factor(prediction_direct), reference = factor(test$mpg01))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 43  1
##          1  8 48
##
##                Accuracy : 0.91
##                  95% CI : (0.836, 0.958)
##     No Information Rate : 0.51
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8204
##
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.8431
##             Specificity : 0.9796
##          Pos Pred Value : 0.9773
##          Neg Pred Value : 0.8571
##              Prevalence : 0.5100
##          Detection Rate : 0.4300
##    Detection Prevalence : 0.4400
##       Balanced Accuracy : 0.9114
##
##        'Positive' Class : 0
##
```

f. The accuracies of these two confusion matrix are similar. For train dataset, it is 0.9024, and for test dataset, it is 0.93, which is a little bit higher than 0.9024. That means the accuracies of predictions are similar.

```r
library(caret)
# train dataset
confusionMatrix(data = as.factor(as.integer(2*mod2$fitted.values)), reference = as.factor(train$mpg01))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 137  10
##          1  18 132
##
##                Accuracy : 0.9057
##                  95% CI : (0.8666, 0.9364)
##     No Information Rate : 0.5219
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8115
##
##  Mcnemar's Test P-Value : 0.1859
##
##             Sensitivity : 0.8839
##             Specificity : 0.9296
##          Pos Pred Value : 0.9320
##          Neg Pred Value : 0.8800
##              Prevalence : 0.5219
##          Detection Rate : 0.4613
##    Detection Prevalence : 0.4949
##       Balanced Accuracy : 0.9067
##
##        'Positive' Class : 0
##
```

```r
# test dataset
prediction <- predict(mod2, test, type = "response")
confusionMatrix(data = as.factor(as.integer(2*prediction)), reference = as.factor(test$mpg01))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 43  1
##          1  8 48
##
##                Accuracy : 0.91
##                  95% CI : (0.836, 0.958)
##     No Information Rate : 0.51
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8204
##
##  Mcnemar's Test P-Value : 0.0455
##
```

```
##             Sensitivity : 0.8431
##             Specificity : 0.9796
##          Pos Pred Value : 0.9773
##          Neg Pred Value : 0.8571
##              Prevalence : 0.5100
##          Detection Rate : 0.4300
##    Detection Prevalence : 0.4400
##       Balanced Accuracy : 0.9114
##
##        'Positive' Class : 0
##
```

g.

```r
sum_mod <- summary(mod2)
```

```r
sum_mod$coefficients
```

```
##                 Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept) -23.396340206 5.543890134 -4.220203 2.440827e-05
## weight       -0.005729974 0.000792961 -7.226048 4.972507e-13
## year          0.518386704 0.089844439  5.769825 7.935383e-09
```

z value = Estimate/Std.Error

```r
CI_intercept<-  sum_mod$coefficients[1,1] + sum_mod$coefficients[1,2] * qnorm(c(0.025, 0.975))
CI_weight <-  sum_mod$coefficients[2,1]  + sum_mod$coefficients[2,2] * qnorm(c(0.025, 0.975))
CI_year <- sum_mod$coefficients[3,1]  + sum_mod$coefficients[3,2] * qnorm(c(0.025, 0.975))
CI_intercept
```

```
## [1] -34.26217 -12.53052
```

```r
CI_weight
```

```
## [1] -0.007284149 -0.004175799
```

```r
CI_year
```

```
## [1] 0.3422948 0.6944786
```

h.

```r
coeff_inter <- rep(0, 1000)
coeff_wei <- rep(0, 1000)
coeff_yea <- rep(0, 1000)
n <- nrow(auto)
for(i in 1:1000){
  row_samp <- sample(1:n, replace = TRUE)
  auto_samp <- auto[row_samp,]
```

```
  temp_mod <- glm(data = auto_samp, mpg01 ~ weight + year, family = binomial)
  coeff_inter[i] <- temp_mod$coefficients[1]
  coeff_wei[i] <- temp_mod$coefficients[2]
  coeff_yea[i] <- temp_mod$coefficients[3]
}
quantile(coeff_inter, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## -38.02260 -15.79226
```

```
quantile(coeff_wei, c(0.025, 0.975))
```

```
##         2.5%        97.5%
## -0.007912136 -0.004910570
```

```
quantile(coeff_yea, c(0.025, 0.975))
```

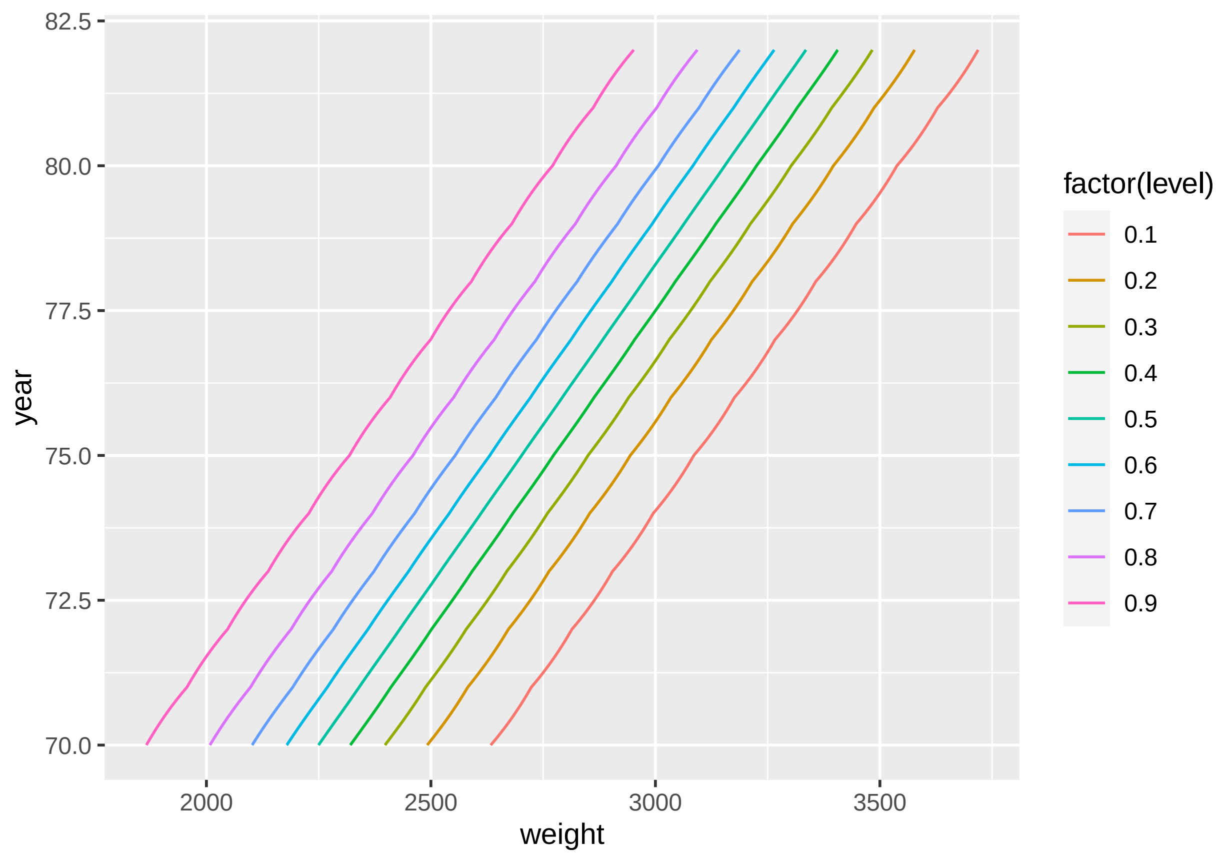
```
##      2.5%     97.5%
## 0.4052757 0.7628001
```

i.

```
contourdata <- data.frame("weight" = as.numeric(), "year" = as.integer())
for(i in min(auto$weight):max(auto$weight)){
  for(j in min(auto$year):max(auto$year)){
    contourdata[nrow(contourdata)+1,]$weight <- i
    contourdata[nrow(contourdata),]$year <- j

  }
}
contourdata$Predict <- predict(mod2, contourdata, type = "response")

ggplot(data = contourdata, aes(x = weight, y = year, z = Predict)) + geom_contour(aes(color = factor(..l
```



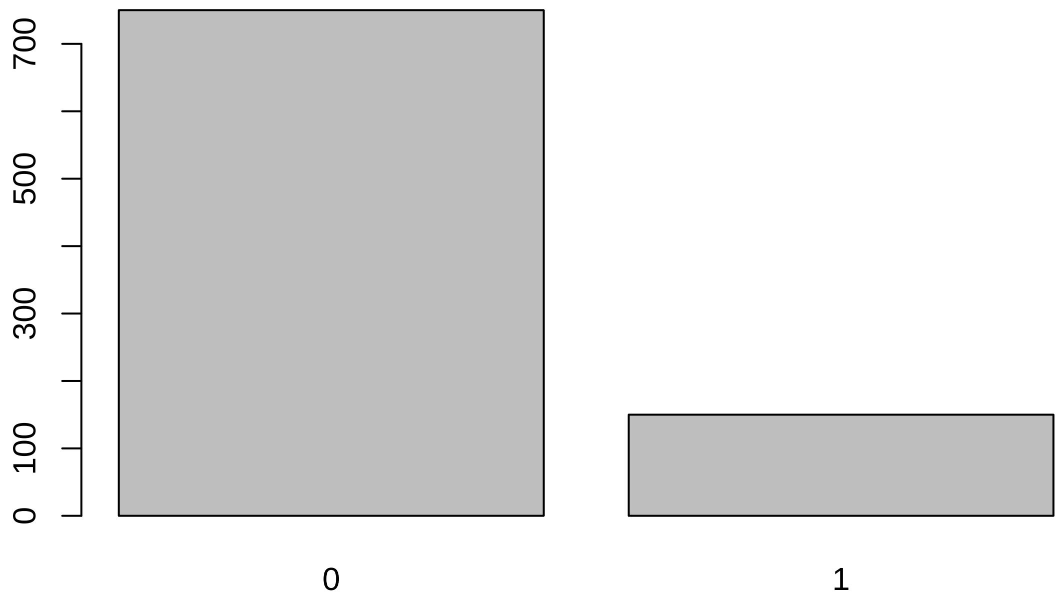
4.

```
churn<- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/customer_churn.csv")
head(churn)
```

```
##               Names Age Total_Purchase Account_Manager Years Num_Sites
## 1 Cameron Williams  42       11066.80               0  7.22         8
## 2    Kevin Mueller  41       11916.22               0  6.50        11
## 3      Eric Lozano  38       12884.75               0  6.67        12
## 4    Phillip White  42        8010.76               0  6.71        10
## 5   Cynthia Norton  37        9191.58               0  5.56         9
## 6 Jessica Williams  48       10356.02               0  5.12         8
##           Onboard_date                                         Location
## 1 2013-08-30 07:00:40      10265 Elizabeth Mission Barkerburgh, AK 89518
## 2 2013-08-13 00:38:46 6157 Frank Gardens Suite 019 Carloshaven, RI 17756
## 3 2016-06-29 06:20:07             1331 Keith Court Alyssahaven, DE 90114
## 4 2014-04-22 12:43:12       13120 Daniel Mount Angelabury, WY 30645-4695
## 5 2016-01-19 15:31:15                765 Tricia Row Karenshire, MH 71730
## 6 2009-03-03 23:13:37 6187 Olson Mountains East Vincentborough, PR 74359
##                        Company Churn
## 1                   Harvey LLC     1
## 2                   Wilson PLC     1
## 3 Miller, Johnson and Wallace     1
## 4                    Smith Inc     1
## 5                   Love-Jones     1
## 6                 Kelly-Warren     1
```

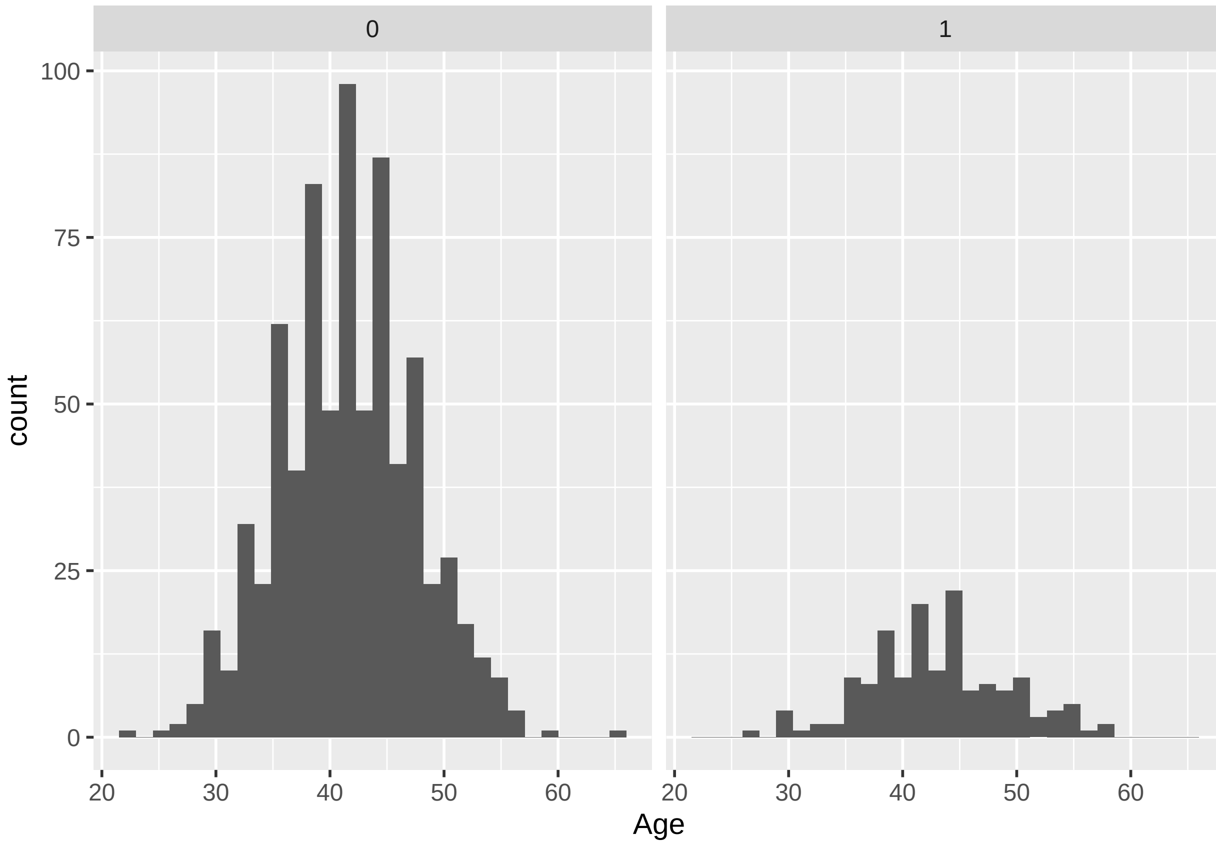
a.

```
barplot(table(churn$Churn))
```



for numeric

```
ggplot(data =churn, aes(x = Age)) + geom_histogram(bins = 30) + facet_grid(.~Churn)
```



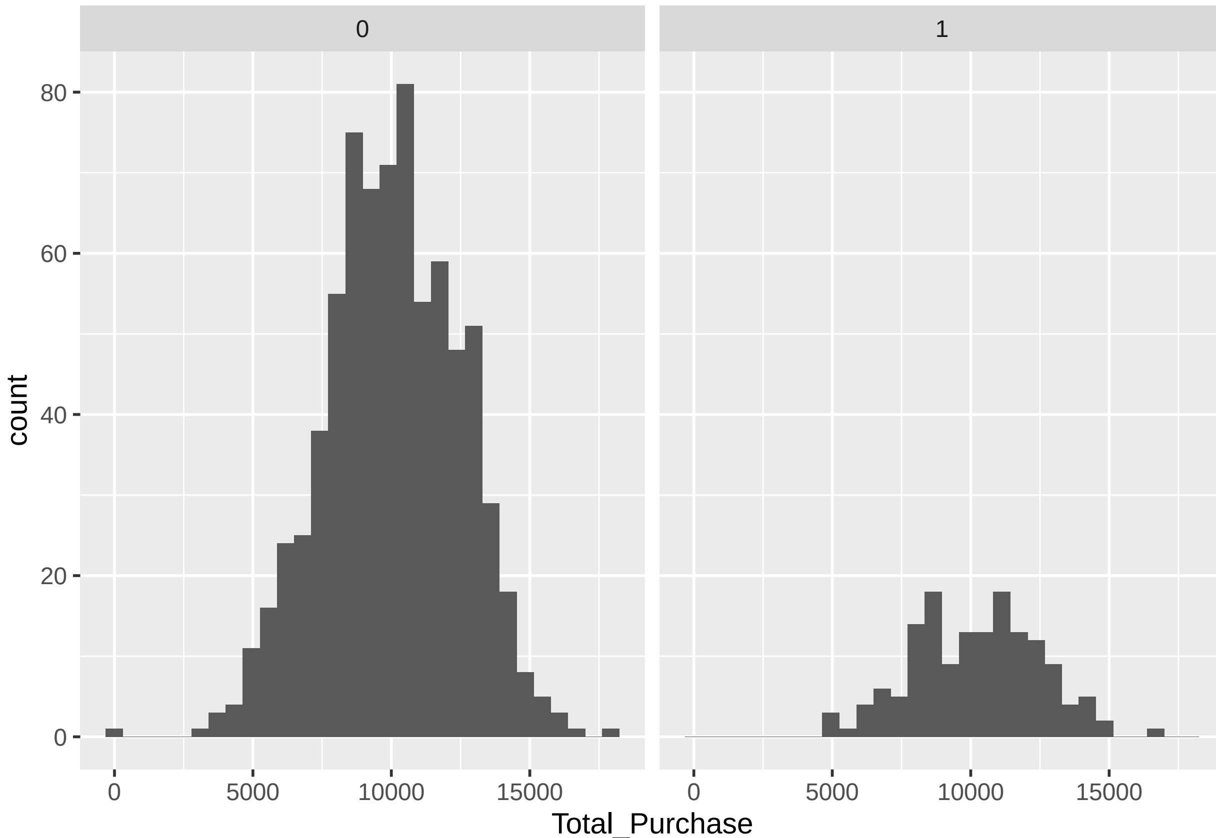

```
quantile(filter(churn, Churn == 1)$Age, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
## 26.0 36.0 38.0 40.0 41.0 43.0 44.0 46.0 49.0 51.1 58.0
```

```
quantile(filter(churn, Churn == 0)$Age, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   22   34   36   38   40   41   43   45   47   49   65
```

```
ggplot(data =churn, aes(x = Total_Purchase)) + geom_histogram(bins = 30) + facet_grid(.~Churn)
```



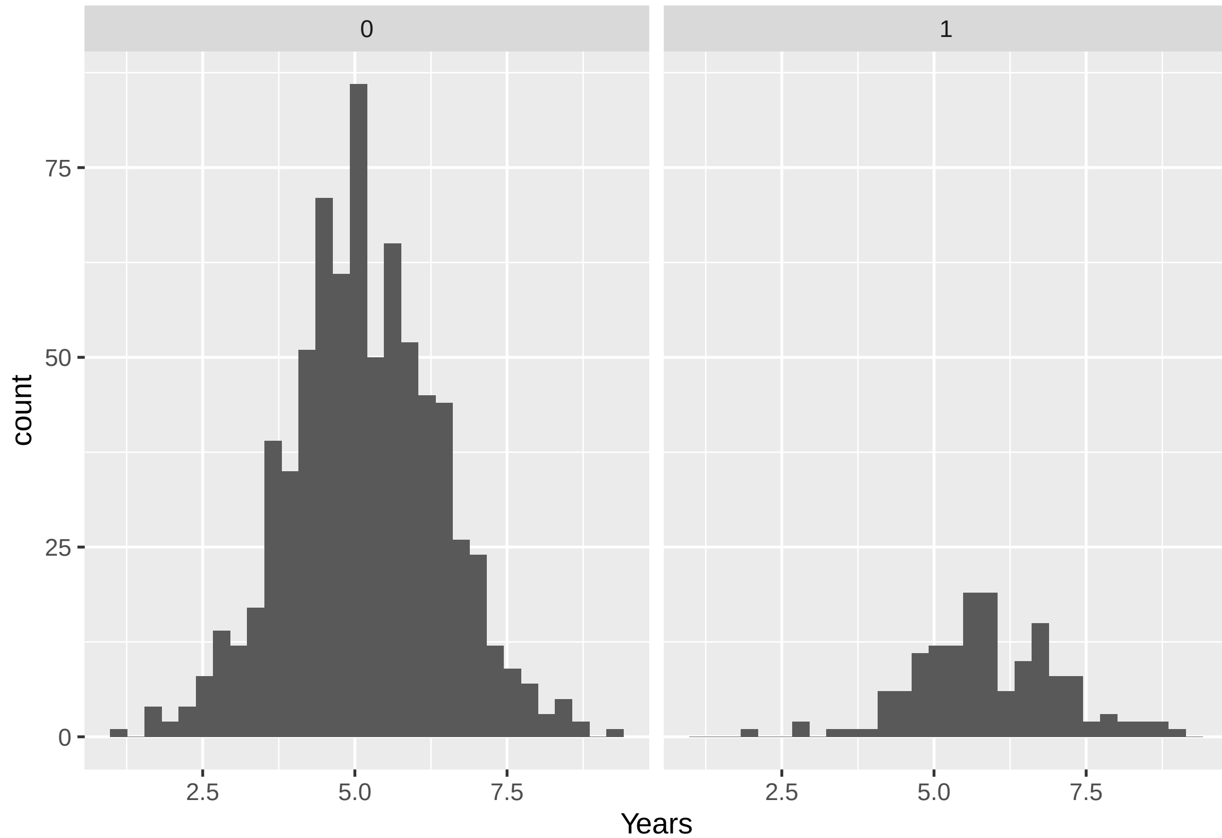

```
quantile(filter(churn, Churn == 1)$Total_Purchase, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##        0%       10%       20%       30%       40%       50%       60%       70%
##  4771.650  7281.048  8231.274  8721.693  9605.476 10273.760 11013.616 11557.814
##       80%       90%      100%
## 12137.814 12894.601 16838.940
```

```
quantile(filter(churn, Churn == 0)$Total_Purchase, seq(0,1, by=0.1), na.rm=TRUE)
```

```
##        0%       10%       20%       30%       40%       50%       60%       70%
##   100.000  6782.914  8038.418  8817.808  9373.312  9999.705 10623.032 11406.109
##       80%       90%      100%
## 12248.320 13137.442 18026.010
```

```
ggplot(data =churn, aes(x = Years)) + geom_histogram(bins = 30) + facet_grid(.~Churn)
```




```
quantile(filter(churn, Churn == 1)$Years, seq(0,1, by=0.1), na.rm=TRUE)
```

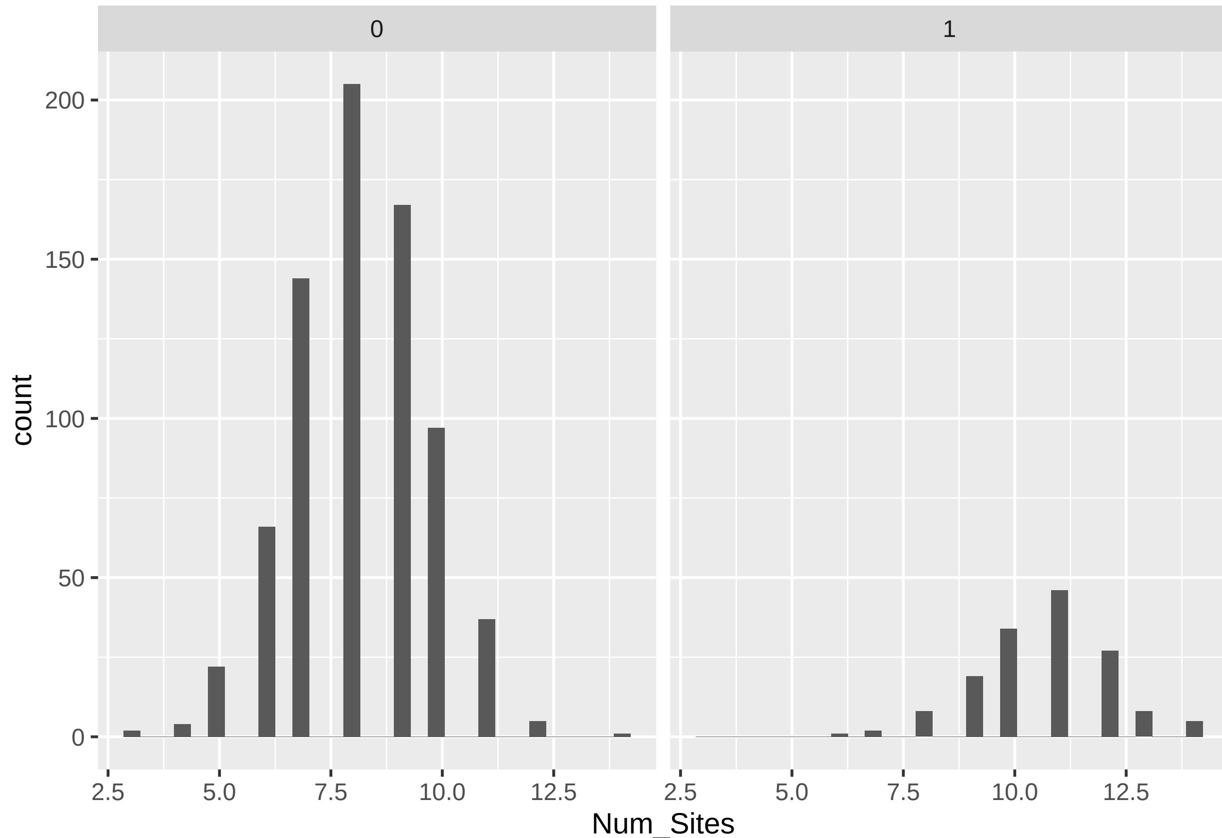
```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
## 2.050 4.557 4.920 5.290 5.582 5.800 6.010 6.509 6.832 7.353 8.970
```

```
quantile(filter(churn, Churn == 0)$Years, seq(0,1, by=0.1), na.rm=TRUE)
```

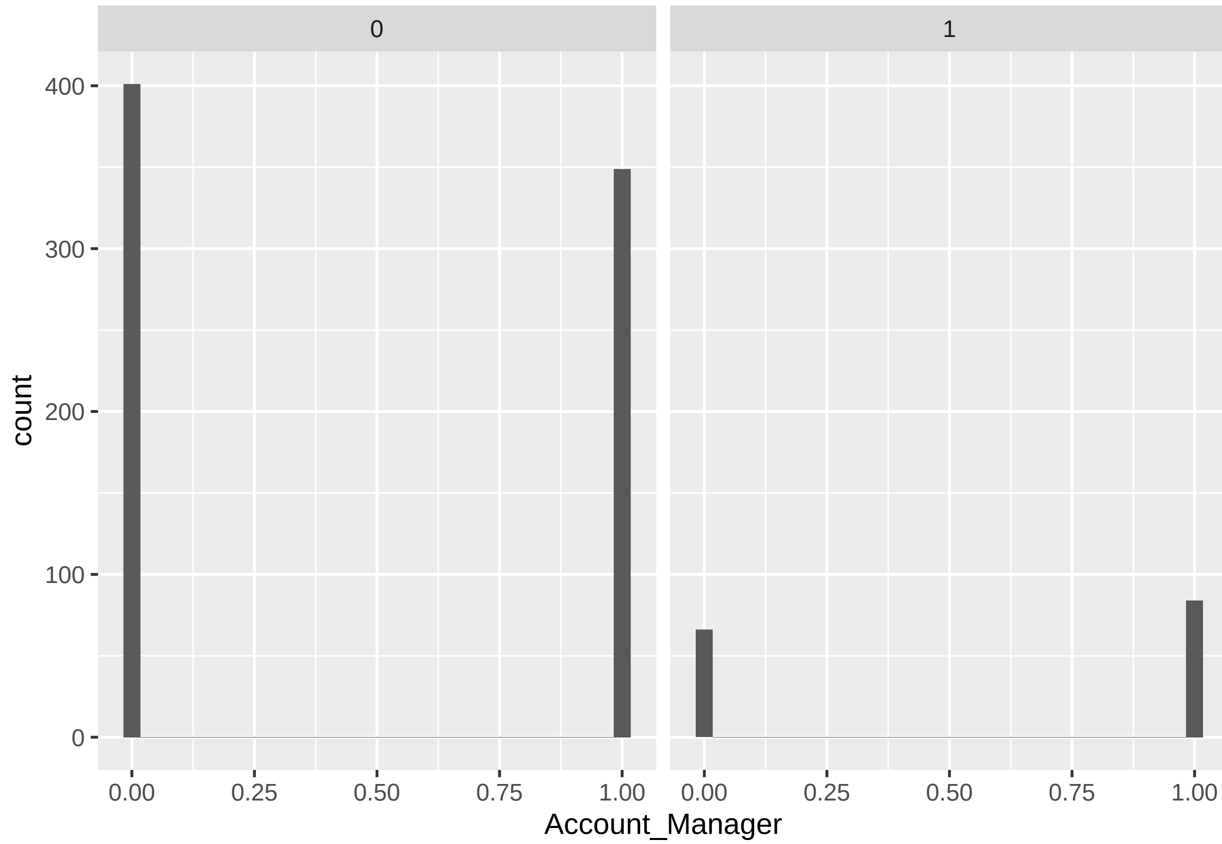
```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
## 1.000 3.620 4.138 4.520 4.840 5.080 5.454 5.800 6.222 6.742 9.150
```

for categorical

```
ggplot(data =churn, aes(x = Num_Sites)) + geom_histogram(bins = 30) + facet_grid(.~Churn)
```




```
ggplot(data =churn, aes(x = Account_Manager)) + geom_histogram(bins = 30) + facet_grid(.~Churn)
```



b.

```
# train-test split

split_pro <- 0.5
n <- length(churn$Names)*split_pro
row_samp <- sample(1:length(churn$Names), n, replace = FALSE)
train <- churn[row_samp,]
test <- churn[-row_samp,]
head(train)
```

```
##                  Names Age Total_Purchase Account_Manager Years Num_Sites
## 250     Tony Schneider  43       11197.42               1  3.48         9
## 807   Michael Anderson  40       11873.76               1  6.50         8
## 856    Jessica Morales  49       11227.48               0  5.10         9
## 199       Andrea Salas  42       11473.38               1  2.87        10
## 585 Elizabeth Kennedy  47       11335.97               0  6.84         6
## 52        Shawn Chavez  44       14036.28               1  7.25        10
##             Onboard_date                                              Location
## 250 2009-04-30 13:55:51     329 Pierce Place Apt. 176 North Tammybury, WV 17594
## 807 2011-08-22 14:22:42            92927 Chavez Fork Brownhaven, WV 20848-9320
## 856 2011-08-16 08:46:53              1384 Wendy Ferry West Ryanburgh, ID 88650
## 199 2015-03-19 22:32:48              308 Graham Corners Valeriehaven, SC 12062
## 585 2014-06-26 02:50:21       07770 Henry Ways Suite 523 Larsonchester, NE 05818
## 52  2009-01-30 01:58:56 42028 Hampton Flat Apt. 206 North Samuelburgh, ME 73072
```

```
##                           Company Churn
## 250                  Harper-Noble     0
## 807    Matthews, Burns and Miller     0
## 856              Barrera-Hamilton     0
## 199                 Blackwell PLC     0
## 585      Davis, Curry and Wallace     0
## 52  Carter, Murphy and Valenzuela     1
```

c.

```r
mod0 <- glm(data = train, Churn ~ Age + Total_Purchase + Account_Manager + Years + Num_Sites, family = b
summary(mod0)
```

```
##
## Call:
## glm(formula = Churn ~ Age + Total_Purchase + Account_Manager +
##     Years + Num_Sites, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9981  -0.4456  -0.2173  -0.0928   3.3284
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.654e+01  2.217e+00  -7.460 8.64e-14 ***
## Age              3.465e-02  2.780e-02   1.246 0.212629
## Total_Purchase  -6.301e-06  6.628e-05  -0.095 0.924253
## Account_Manager  2.554e-01  3.242e-01   0.788 0.430709
## Years            4.723e-01  1.297e-01   3.642 0.000271 ***
## Num_Sites        1.154e+00  1.354e-01   8.522  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 402.27  on 449  degrees of freedom
## Residual deviance: 250.22  on 444  degrees of freedom
## AIC: 262.22
##
## Number of Fisher Scoring iterations: 6
```

```r
mod1 <- glm(data = train, Churn ~ Years + Num_Sites, family = binomial)
summary(mod1)
```

```
##
## Call:
## glm(formula = Churn ~ Years + Num_Sites, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9026  -0.4527  -0.2271  -0.0956   3.2657
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -15.0487     1.6322  -9.220  < 2e-16 ***
## Years         0.4753     0.1294   3.674 0.000238 ***
## Num_Sites     1.1560     0.1351   8.557  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 402.27  on 449  degrees of freedom
## Residual deviance: 252.40  on 447  degrees of freedom
## AIC: 258.4
##
## Number of Fisher Scoring iterations: 6
```

```r
confusionMatrix(data = as.factor(as.integer(2*mod1$fitted.values)), reference = as.factor(train$Churn))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 364  38
##          1  12  36
##
##                Accuracy : 0.8889
##                  95% CI : (0.8561, 0.9164)
##     No Information Rate : 0.8356
##     P-Value [Acc > NIR] : 0.0008966
##
##                   Kappa : 0.5292
##
##  Mcnemar's Test P-Value : 0.0004070
##
##             Sensitivity : 0.9681
##             Specificity : 0.4865
##          Pos Pred Value : 0.9055
##          Neg Pred Value : 0.7500
##              Prevalence : 0.8356
##          Detection Rate : 0.8089
##    Detection Prevalence : 0.8933
##       Balanced Accuracy : 0.7273
##
##        'Positive' Class : 0
##
```

```r
prediction <- predict(mod1, test, type = "response")
confusionMatrix(data = as.factor(as.integer(2*prediction)), reference = as.factor(test$Churn))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
```

```
##          0 356  32
##          1  18  44
##
##                Accuracy : 0.8889
##                  95% CI : (0.8561, 0.9164)
##     No Information Rate : 0.8311
##     P-Value [Acc > NIR] : 0.000385
##
##                   Kappa : 0.5729
##
##  Mcnemar's Test P-Value : 0.065992
##
##             Sensitivity : 0.9519
##             Specificity : 0.5789
##          Pos Pred Value : 0.9175
##          Neg Pred Value : 0.7097
##              Prevalence : 0.8311
##          Detection Rate : 0.7911
##    Detection Prevalence : 0.8622
##       Balanced Accuracy : 0.7654
##
##        'Positive' Class : 0
##
```

The accuracy of model towards test datasets is higher than one towards train datasets, ane p-value is much higher, too. For sensitivity, specificity and so on, the values of test dataset is much better than train dataset.