CARS Notebook

Code

Importing all required libraries

Hide

```
library(caret)
```

```
Loading required package: lattice
Loading required package: ggplot2
Registered S3 method overwritten by 'dplyr':
  method           from
  print.rowwise_df
```

Hide

```
library(caTools)
library(class)
library(e1071)
library(DataExplorer)
```

```
Registered S3 method overwritten by 'htmlwidgets':
  method           from
  print.htmlwidget tools:rstudio
```

Hide

```
library(rpivotTable)
library(ggplot2)
library(corrplot)
```

```
corrplot 0.84 loaded
```

Hide

```
library(ggthemes)
library(pROC)
```

```
Type 'citation("pROC")' for a citation.

Attaching package: 惚牠pROC惚牸

The following objects are masked from 惚牠package:stats惚牸:

    cov, smooth, var
```

Hide

```
library(PerformanceAnalytics)
```

```
Loading required package: xts
Loading required package: zoo

Attaching package: 惏栬zoo惏犵

The following objects are masked from 惏栬package:base惏犵:

    as.Date, as.Date.numeric

Registered S3 method overwritten by 'xts':
  method     from
  as.zoo.xts zoo

Attaching package: 惏栬PerformanceAnalytics惏犵

The following objects are masked from 惏栬package:e1071惏犵:

    kurtosis, skewness

The following object is masked from 惏栬package:gplots惏犵:

    textplot

The following object is masked from 惏栬package:graphics惏犵:

    legend
```

Hide

```
library(ipred)
library(rpart)
library(ROCR)
library(data.table)
```

```
data.table 1.12.2 using 4 threads (see ?getDTthreads).  Latest news: r-datatable.com

Attaching package: 惏栬data.table惏犵

The following objects are masked from 惏栬package:xts惏犵:

    first, last
```

Hide

```
library(mltools)
```

```
Attaching package: 惻䏆mltools惻䏇

The following object is masked from 惻䏆package:PerformanceAnalytics惻䏇:

    skewness

The following object is masked from 惻䏆package:e1071惻䏇:

    skewness
```

Hide

```
library(xgboost)
library(caret)
library(rms)
```

```
Loading required package: Hmisc
Loading required package: survival

Attaching package: 惻䏆survival惻䏇

The following object is masked from 惻䏆package:caret惻䏇:

    cluster

Loading required package: Formula

Attaching package: 惻䏆Hmisc惻䏇

The following object is masked from 惻䏆package:e1071惻䏇:

    impute

The following objects are masked from 惻䏆package:base惻䏇:

    format.pval, units

Loading required package: SparseM

Attaching package: 惻䏆SparseM惻䏇

The following object is masked from 惻䏆package:base惻䏇:

    backsolve
```

Hide

```
library(DMwR)
```

```
Loading required package: grid
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

Setting up working directory and reading the dataset.

Hide

```
setwd('D:/Smitayan/PGP BABI')
cars.study = read.csv('Cars case study-dataset.csv',header = T)
```

Hide

```
names(cars.study)
```

```
[1] "Age"       "Gender"    "Engineer"  "MBA"       "Work.Exp"  "Salary"
[7] "Distance"  "license"   "Transport"
```

Exploratory Data Analysis:

Hide

```
summary(cars.study)
```

```
      Age           Gender       Engineer           MBA            Work.Exp
 Min.   :18.00   Female:121   Min.   :0.0000   Min.   :0.0000   Min.   : 0.000
 1st Qu.:25.00   Male  :297   1st Qu.:0.2500   1st Qu.:0.0000   1st Qu.: 3.000
 Median :27.00                Median :1.0000   Median :0.0000   Median : 5.000
 Mean   :27.33                Mean   :0.7488   Mean   :0.2614   Mean   : 5.873
 3rd Qu.:29.00                3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 8.000
 Max.   :43.00                Max.   :1.0000   Max.   :1.0000   Max.   :24.000
                                               NA's   :1
     Salary          Distance        license                   Transport
 Min.   : 6.500   Min.   : 3.20   Min.   :0.0000   2Wheeler        : 83
 1st Qu.: 9.625   1st Qu.: 8.60   1st Qu.:0.0000   Car             : 35
 Median :13.000   Median :10.90   Median :0.0000   Public Transport:300
 Mean   :15.418   Mean   :11.29   Mean   :0.2033
 3rd Qu.:14.900   3rd Qu.:13.57   3rd Qu.:0.0000
 Max.   :57.000   Max.   :23.40   Max.   :1.0000
```

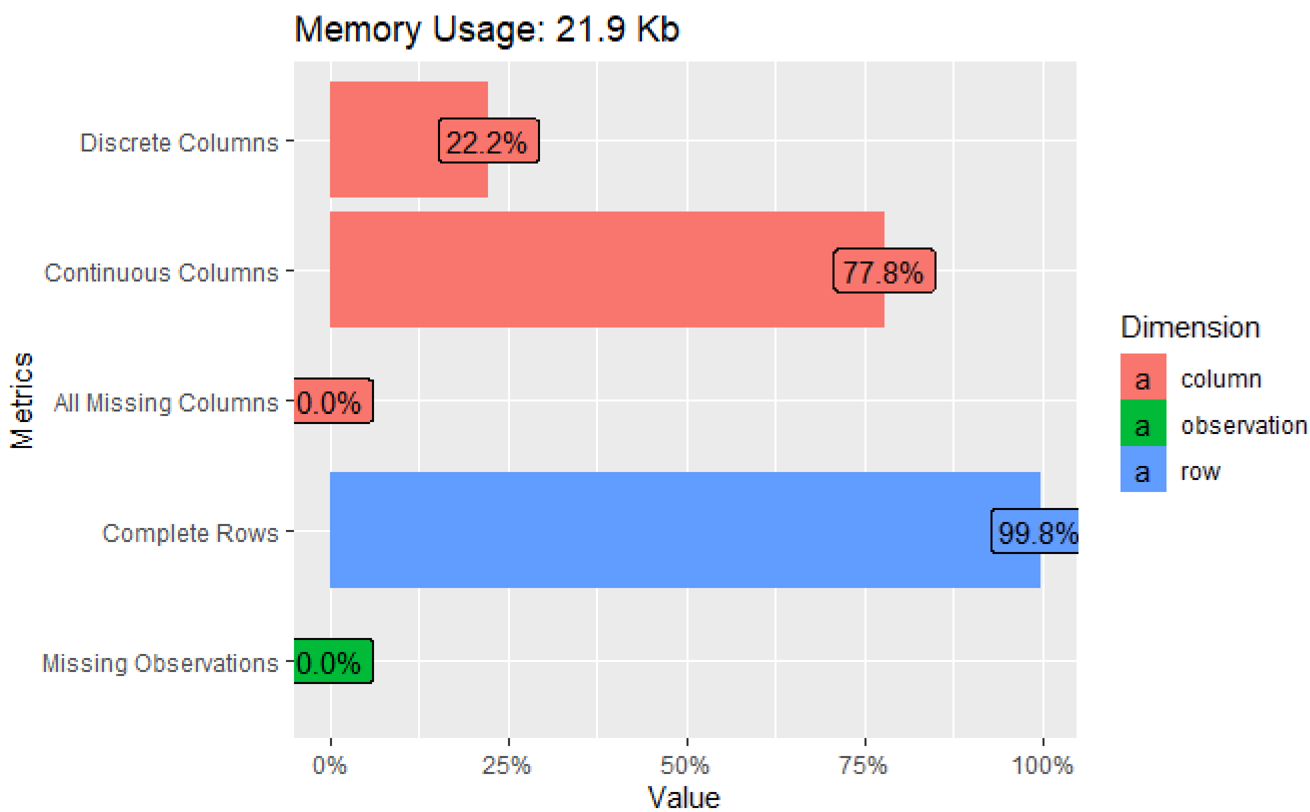
Hide

```
str(cars.study)
```

```
'data.frame':	418 obs. of  9 variables:
 $ Age      : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer : int  1 1 1 0 0 0 1 0 1 1 ...
 $ MBA      : int  0 0 0 0 0 0 1 0 0 0 ...
 $ Work.Exp : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 1 1 1 1 1 1 1 1 1 1 ...
```

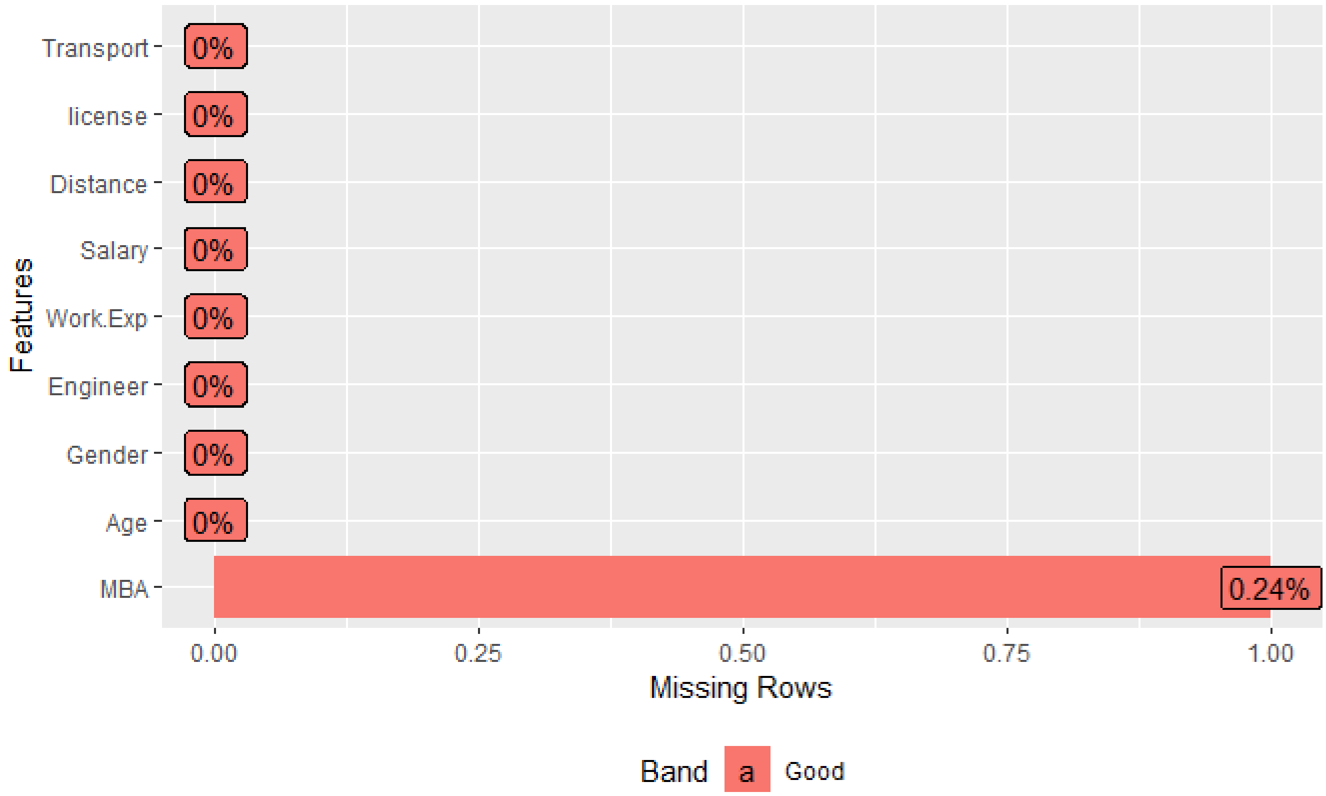
Hide

```
plot_intro(cars.study)
```



Hide

```
plot_missing(cars.study)
```



Hide

```
sapply(cars.study, function(x) sum(is.na(x)))
```

```
      Age     Gender   Engineer        MBA   Work.Exp     Salary   Distance    license  Transport 
        0          0          0          1          0          0          0          0          0 
```

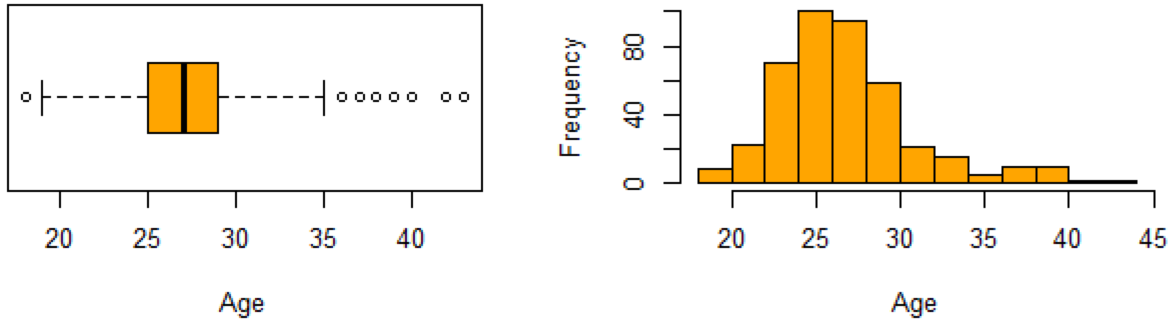
Univariate data analysis using data visualization techniques.

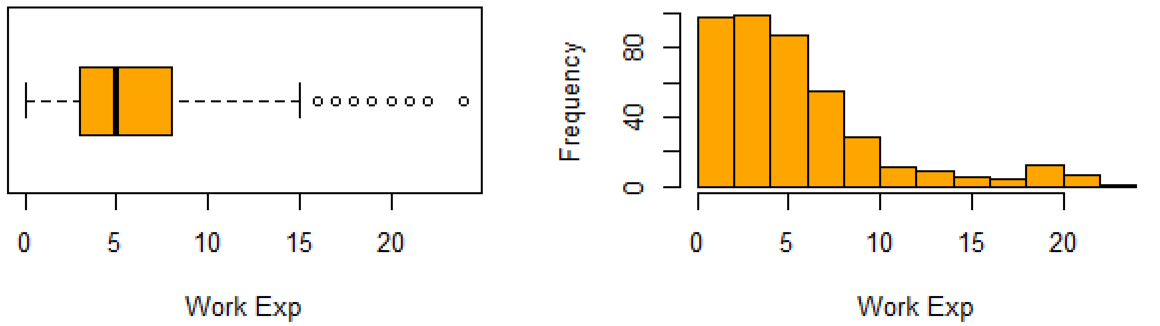
Hide

```
par(mfrow=c(2,2))
boxplot(cars.study$Age,xlab='Age',horizontal = T,col='orange')
hist(cars.study$Age,xlab = 'Age',main='',col = 'orange')
```

Hide

```
boxplot(cars.study$Work.Exp,xlab='Work Exp',horizontal = T,col = 'orange')
hist(cars.study$Work.Exp,xlab = 'Work Exp',main = '',col = 'orange')
```



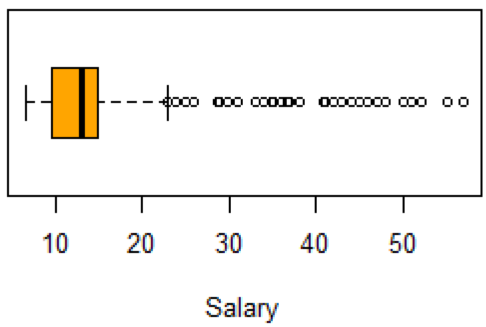


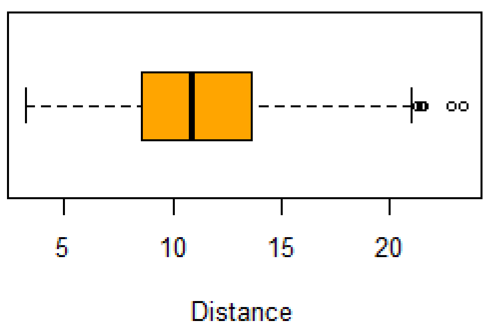
Hide

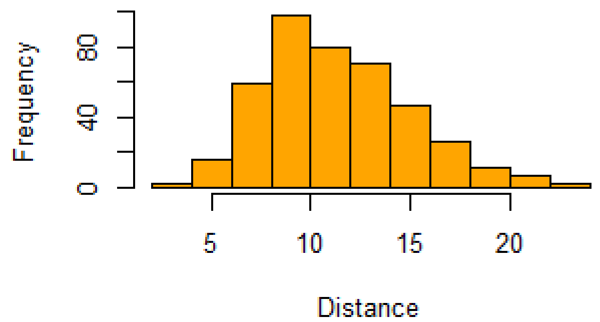
```
boxplot(cars.study$Salary,xlab='Salary',horizontal = T,col = 'orange')
hist(cars.study$Salary,xlab = 'Salary',main = '',col = 'orange')
```

Hide

```
boxplot(cars.study$Distance,xlab='Distance',horizontal = T,col = 'orange')
hist(cars.study$Distance,xlab = 'Distance',main = '',col = 'orange')
```







Converting certain attributes to factor

Hide

```
cars.study$Engineer = as.factor(cars.study$Engineer)
cars.study$MBA = as.factor(cars.study$MBA)
cars.study$license = as.factor(cars.study$license)
```

Barplots of factor variables

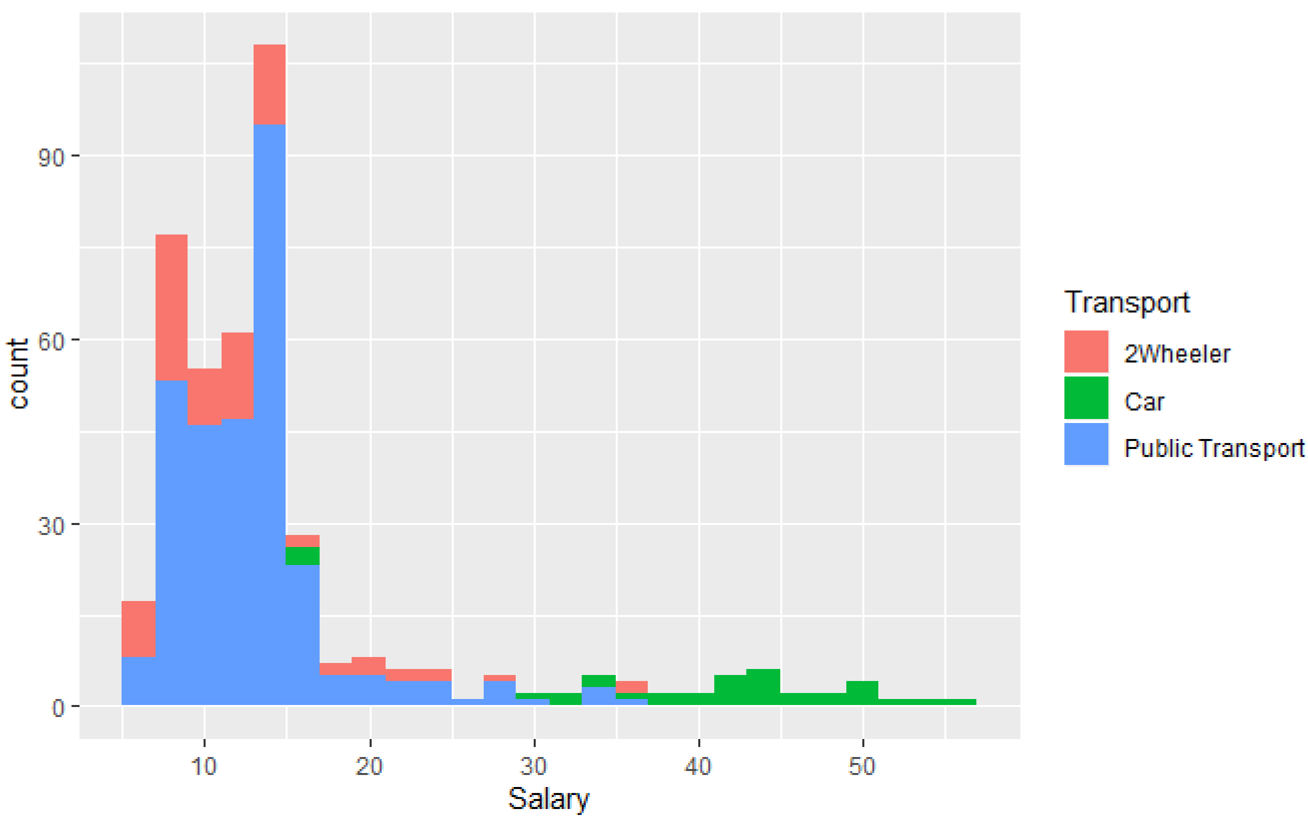
Hide

```
?plot_bar
plot_bar(cars.study)
```

Bivariate data analysis using ggplot

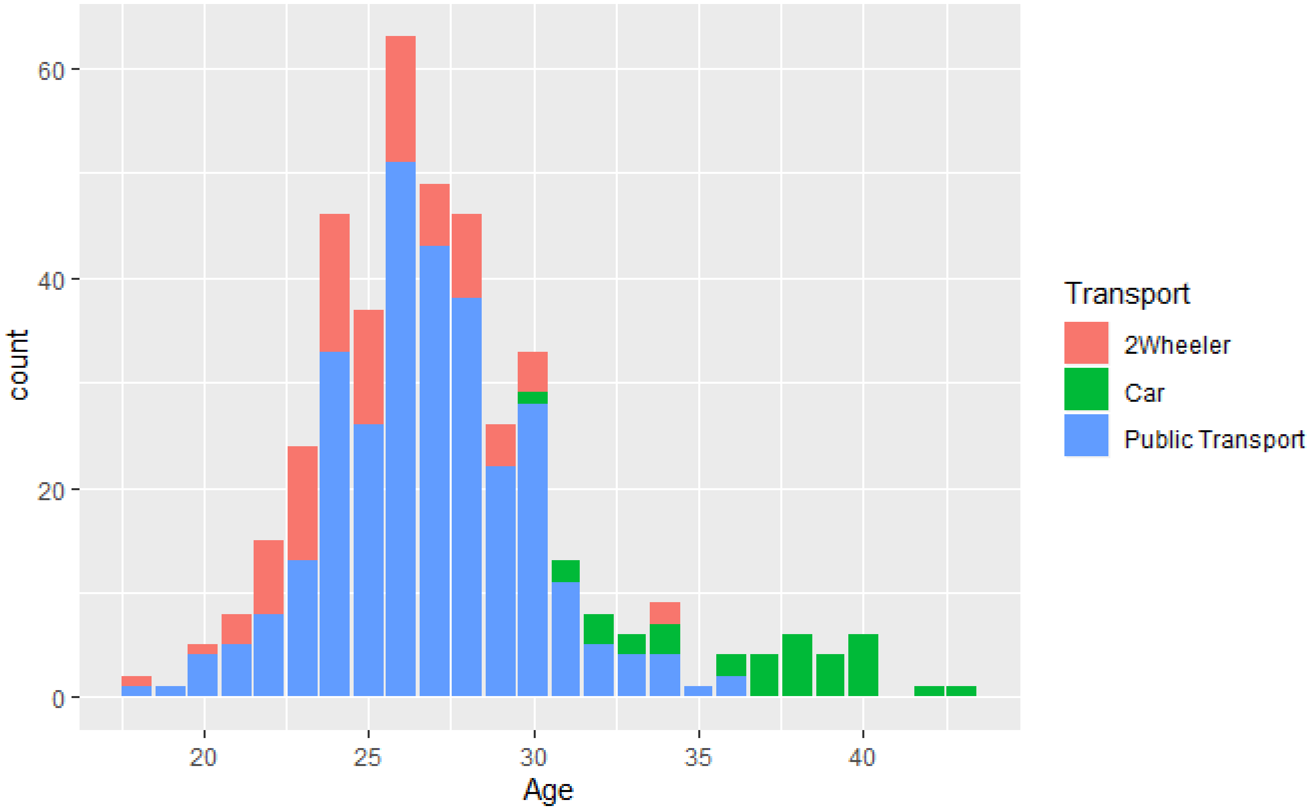
Hide

```
ggplot(data = cars.study) + aes(x=Salary,fill=Transport)+geom_histogram(binwidth = 2)
```



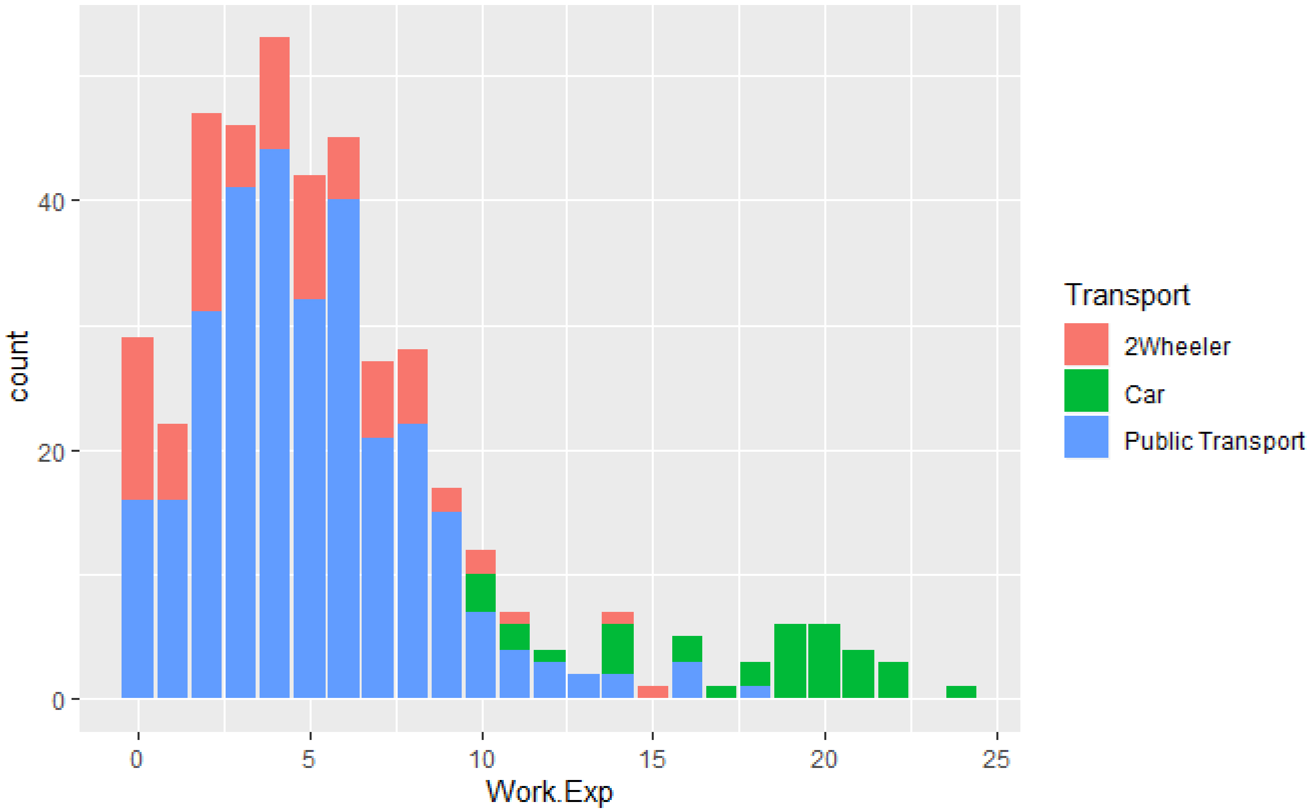
Hide

```
?geom_bar
ggplot(data = cars.study) + aes(x=Age,fill=Transport)+geom_bar(position = 'stack')
```



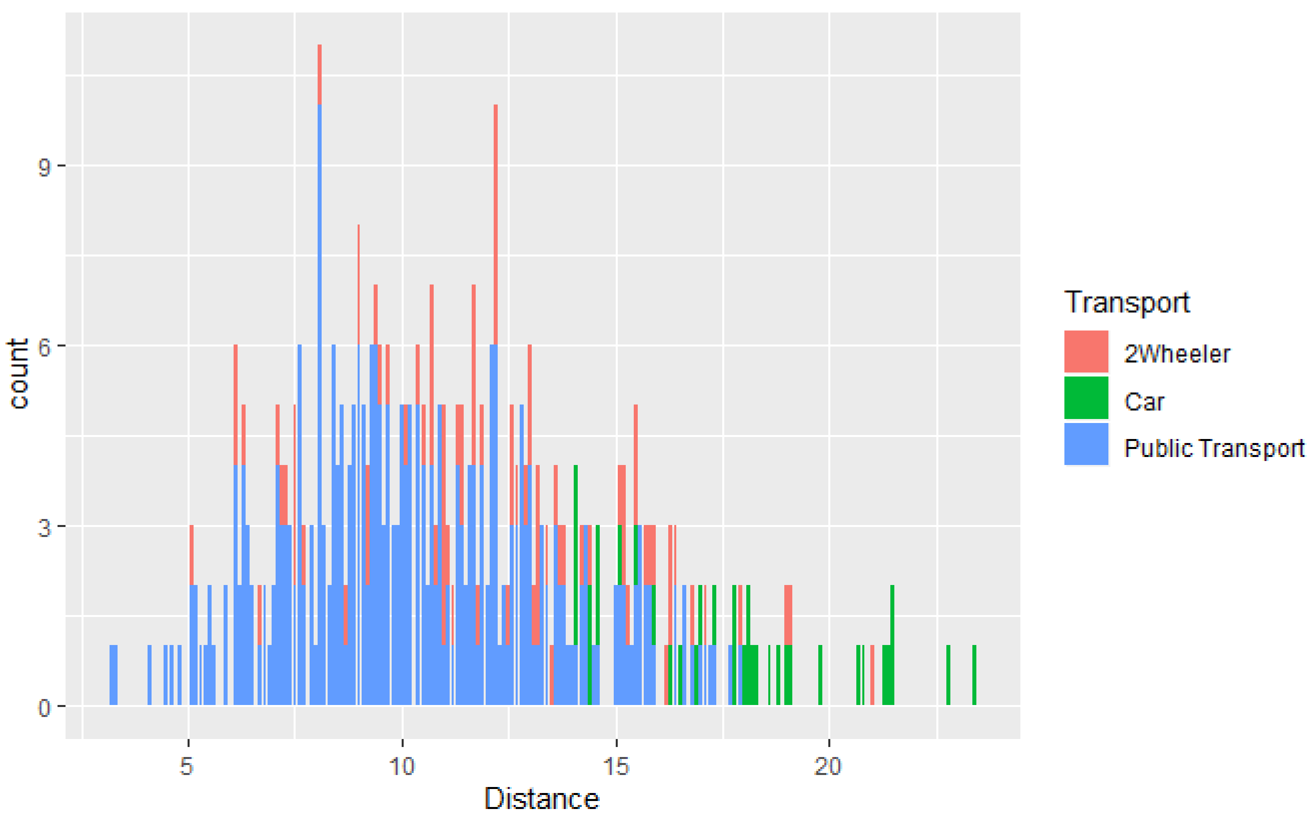
Hide

```
ggplot(data = cars.study) + aes(x=Work.Exp,fill=Transport)+geom_bar(position = 'stack')
```



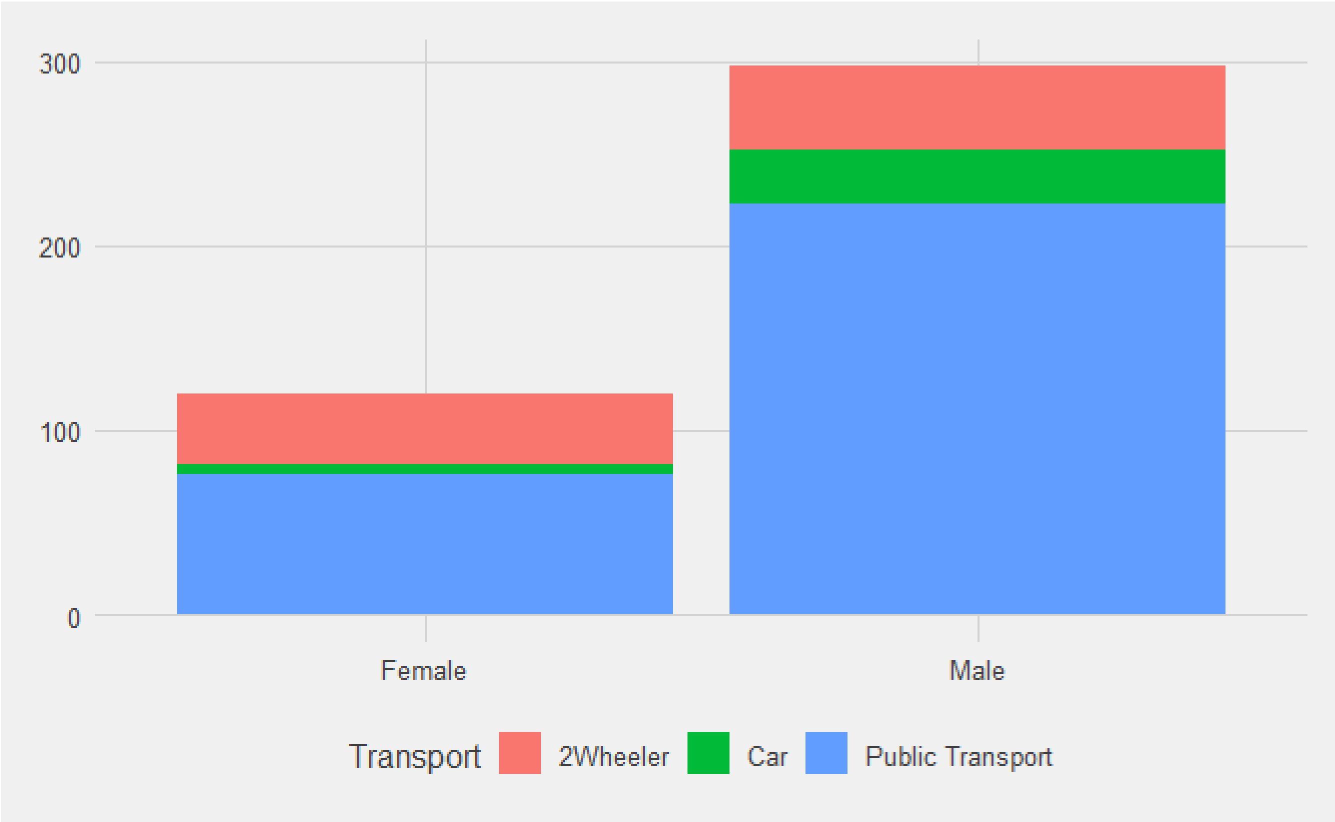
Hide

```
ggplot(data = cars.study) + aes(x=Distance,fill=Transport)+geom_bar(position = 'stack')
```



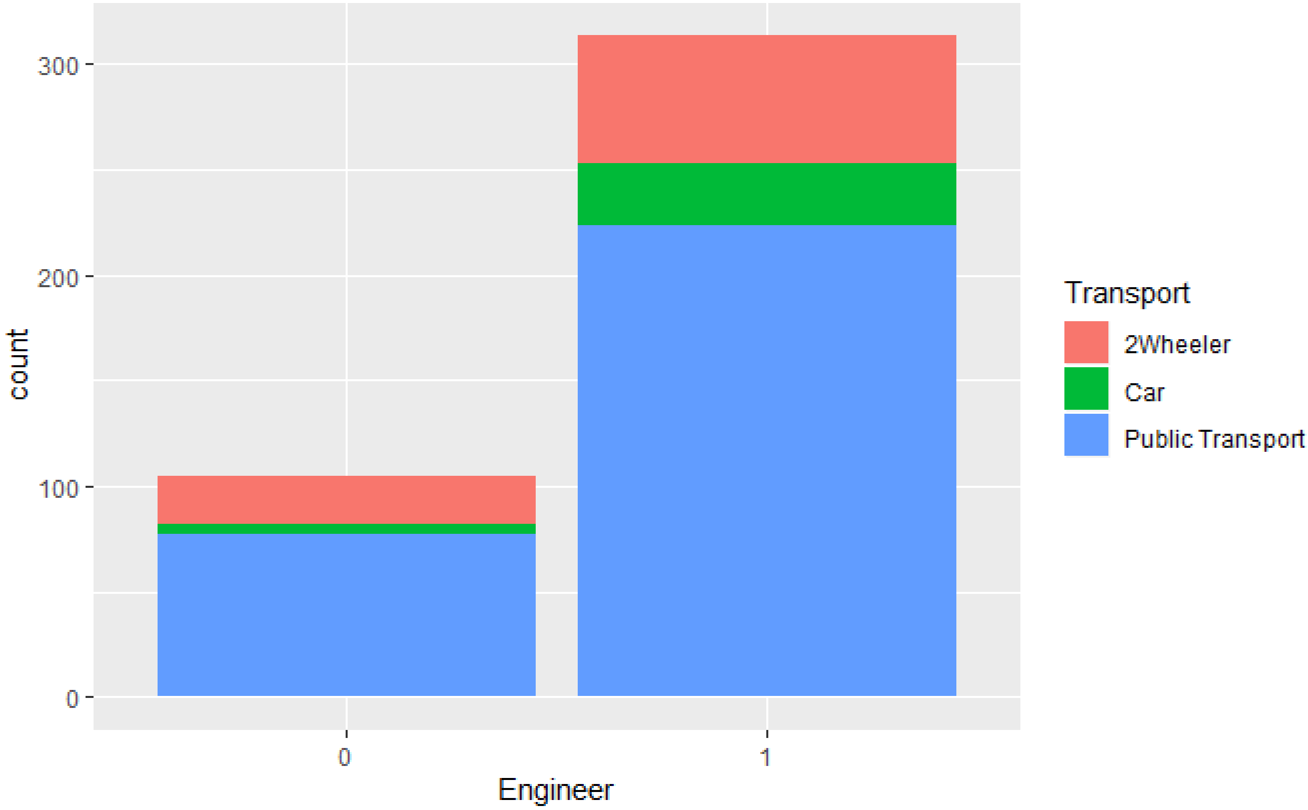
Hide

```
ggplot(data = cars.study) + aes(x=Gender,fill=Transport)+geom_bar(position = 'stack')+theme_five
thirtyeight()
```



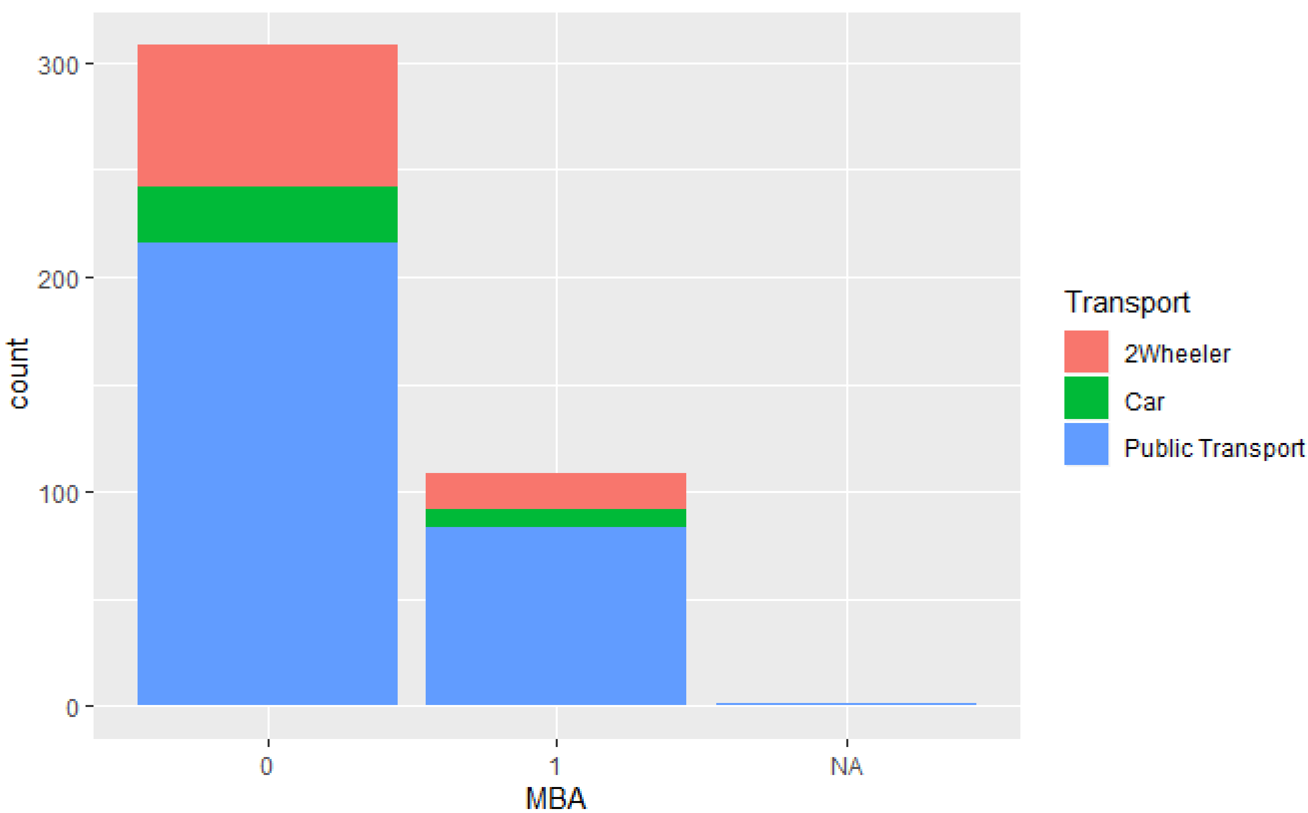
Hide

```
ggplot(data = cars.study) + aes(x=Engineer,fill=Transport)+geom_bar(position = 'stack')
```



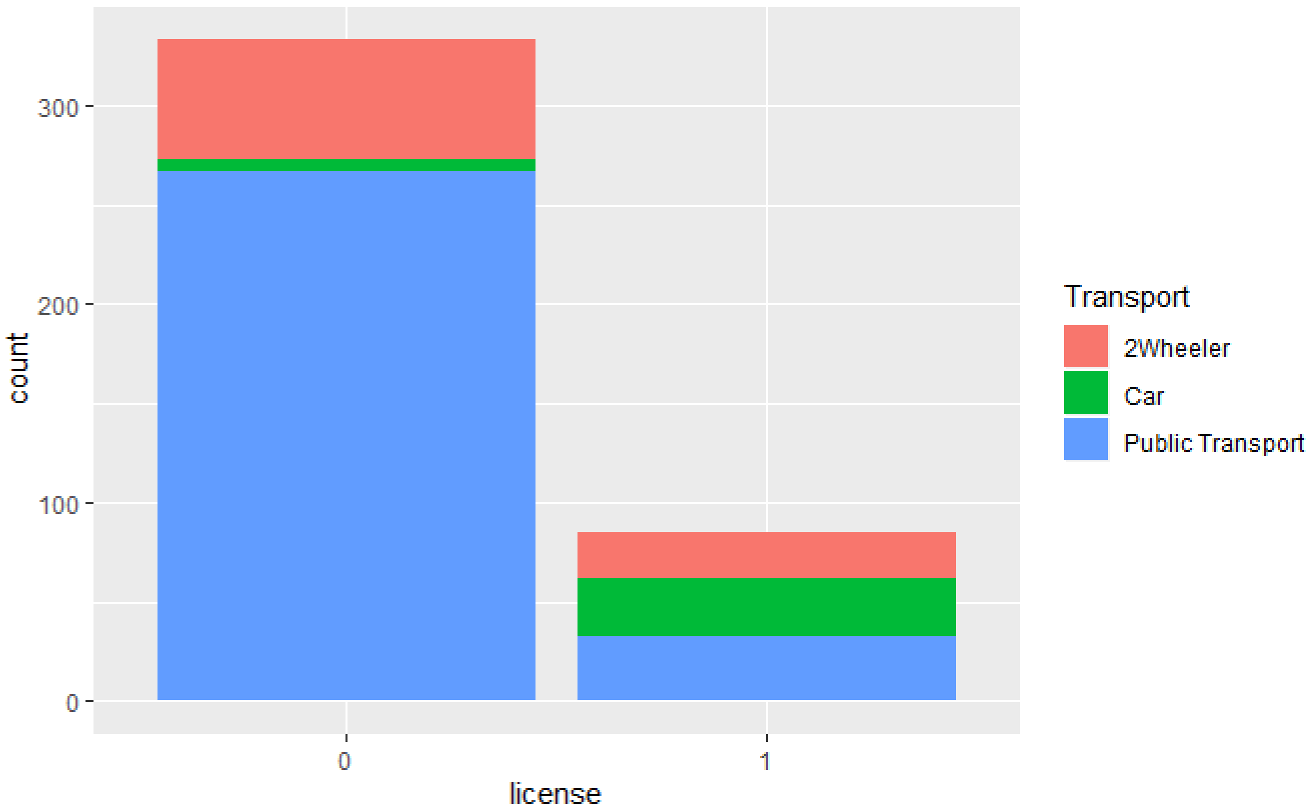
Hide

```
ggplot(data = cars.study) + aes(x=MBA,fill=Transport)+geom_bar(position = 'stack')
```



Hide

```
ggplot(data = cars.study) + aes(x=license,fill=Transport)+geom_bar(position = 'stack')
```



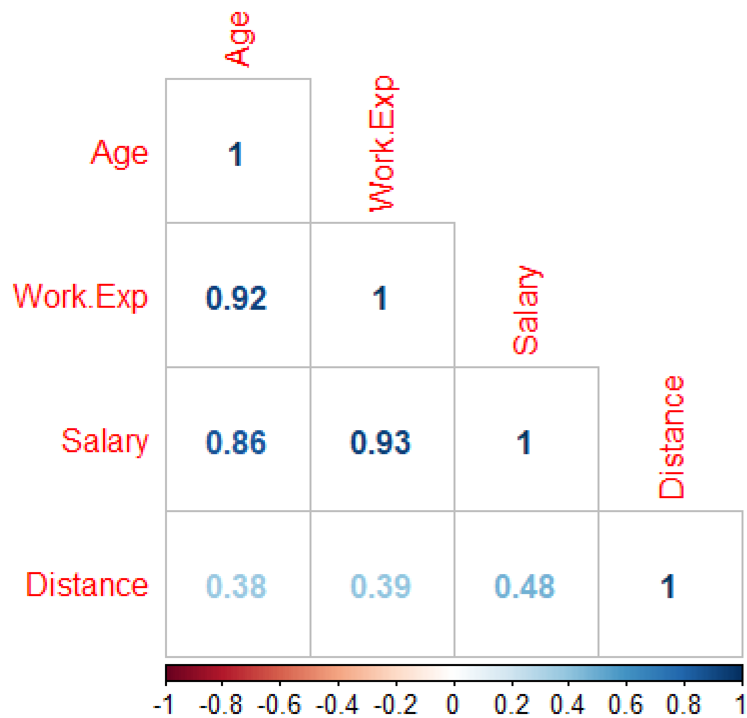
Hide

```
str(cars.study)
```

```
'data.frame':	418 obs. of  9 variables:
 $ Age      : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 1 2 2 ...
 $ MBA      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...
 $ Work.Exp : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
 $ Transport: Factor w/ 2 levels "Car","Others": 2 2 2 2 2 2 2 2 2 2 ...
```

Hide

```
corrplot(cor(cars.study[c(1,5,6,7)]),method="number",type = 'lower')
```



Hide

```
?chart.Correlation
chart.Correlation(cars.study[c(1,5,6,7)])
```

Chisquare test to determine significance of factor variables on dependent variable

Hide

```
chisq.test(cars.study$Gender,cars.study$Transport)$p.value
```

```
[1] 0.0003958196
```

Hide

```
chisq.test(cars.study$Engineer,cars.study$Transport)$p.value
```

```
[1] 0.2866151
```

Hide

```
chisq.test(cars.study$MBA,cars.study$Transport)$p.value
```

```
[1] 0.409505
```

Hide

```
chisq.test(cars.study$license,cars.study$Transport)$p.value
```

```
[1] 4.271117e-23
```

Hide

```
table(cars.study$Transport)
```

```

       2Wheeler              Car Public Transport 
             83               35              300 
```

Hide

```
35/nrow(cars.study)
```

```
[1] 0.08373206
```

Converting levels ‘2wheeler’ and ‘Public Transport’ to ‘Others’, so that we have two levels in the variable Transport and hence we can do binary classificaton.

Hide

```
cars.study$Transport = as.character(cars.study$Transport)
cars.study$Transport = ifelse(cars.study$Transport !='Car','Others','Car')
cars.study$Transport = as.factor(cars.study$Transport)
```

Hide

```
table(cars.study$Transport)
```

```

   Car Others 
    35    383 
```

Splitting dataset into Training and Test

Hide

```
set.seed(123)
trainidx = sample(nrow(cars.study),.7*nrow(cars.study),replace = F)
cars.training = cars.study[trainidx,]
cars.test = cars.study[-trainidx,]
```

Hide

```
table(cars.training$Transport)
```

```

   Car Others 
    26    266 
```

Hide

```
colnames(cars.study)
```

```
[1] "Age"       "Gender"    "Engineer"  "MBA"       "Work.Exp"  "Salary"    "Distance"  "licens
e"   "Transport"
```

#Logistic Regression

Hide

```
cars.logistic = glm(Transport~.,data = cars.training,family = 'binomial')
```

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

Hide

```
summary(cars.logistic)
```

```
Call:
glm(formula = Transport ~ ., family = "binomial", data = cars.training)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.11902   0.00012   0.00108   0.00854   1.53910

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  75.3575    44.2195   1.704   0.0883 .
Age          -2.0188     1.4301  -1.412   0.1581
GenderMale    1.2982     1.7540   0.740   0.4592
Engineer1    -0.4323     1.7672  -0.245   0.8068
MBA1          1.8562     2.1357   0.869   0.3848
Work.Exp      0.8418     1.0654   0.790   0.4294
Salary       -0.1456     0.2038  -0.715   0.4748
Distance     -1.0086     0.4477  -2.253   0.0243 *
license1     -2.8730     2.6916  -1.067   0.2858
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 175.196  on 290  degrees of freedom
Residual deviance:  16.317  on 282  degrees of freedom
  (1 observation deleted due to missingness)
AIC: 34.317

Number of Fisher Scoring iterations: 11
```

Hide

```
vif(cars.logistic)
```

```
       Age GenderMale  Engineer1       MBA1   Work.Exp     Salary   Distance
 24.392514   1.893063   1.212887   2.709972  29.764629   9.211644   4.030199
  license1
  4.490734
```

Hide

```
cars.logistic = glm(Transport~Age+Gender+Distance+license,data = cars.training,family = 'binomia
l')
```

Hide

```
summary(cars.logistic)
```

```
Call:
glm(formula = Transport ~ Age + Gender + Distance + license,
    family = "binomial", data = cars.training)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.17122   0.00044   0.00277   0.01182   1.33292

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  51.5286    18.7068   2.755  0.00588 **
Age          -1.1272     0.4356  -2.588  0.00966 **
GenderMale    0.5148     1.3267   0.388  0.69802
Distance     -0.9042     0.3364  -2.688  0.00718 **
license1     -1.2989     1.3244  -0.981  0.32673
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 175.38  on 291  degrees of freedom
Residual deviance:  18.29  on 287  degrees of freedom
AIC: 28.29

Number of Fisher Scoring iterations: 11
```

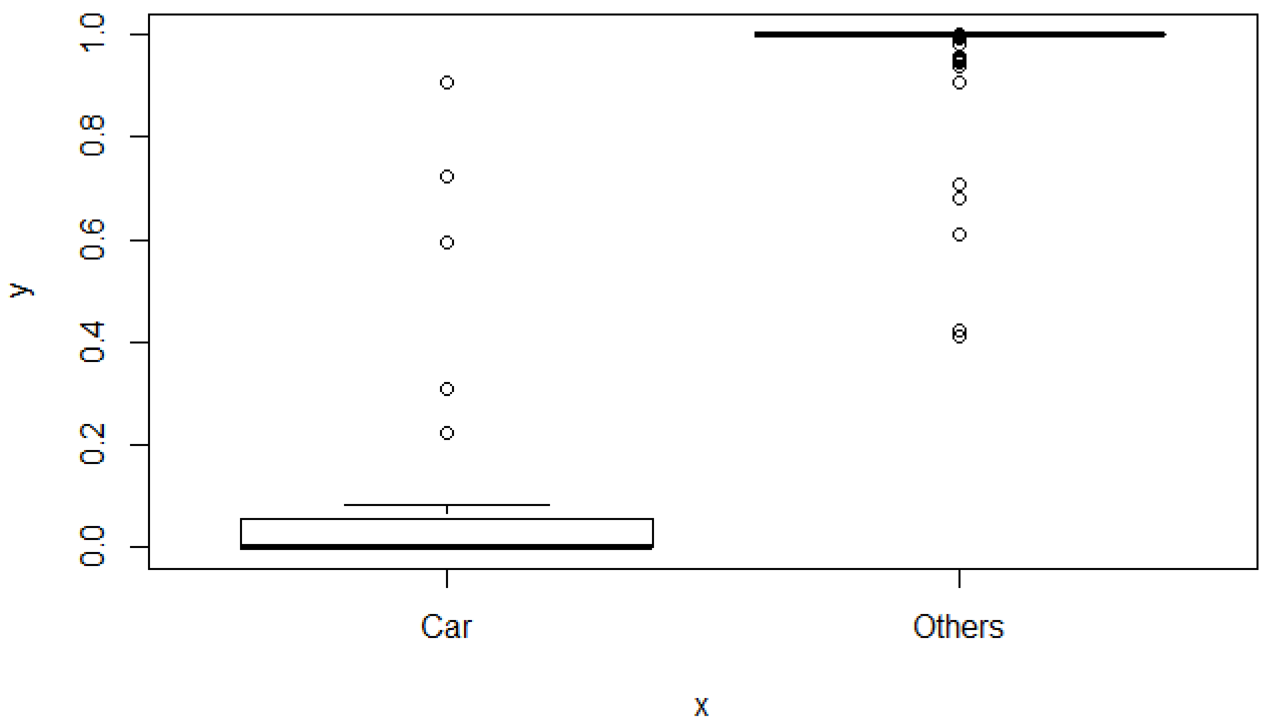
Hide

```
vif(cars.logistic)
```

```
       Age GenderMale   Distance   license1
  3.252862   1.179232   3.031925   1.204285
```

Hide

```
plot(cars.training$Transport,cars.logistic$fitted.values)
```



Hide

```
predicted.transport = ifelse(cars.logistic$fitted.values<0.92,'Car','Others')
```

Hide

```
table(cars.training$Transport,predicted.transport)
```

```
        predicted.transport
         Car Others
  Car     26      0
  Others   7    259
```

Hide

```
accuracy = sum(diag(table(cars.training$Transport,predicted.transport)))/nrow(cars.training)
accuracy
```

```
[1] 0.9760274
```

Hide

```
TNR = 259/266
TNR
```

```
[1] 0.9736842
```

Hide

```
roc(cars.training$Transport,cars.logistic$fitted.values)
```

```
Setting levels: control = Car, case = Others
Setting direction: controls < cases
```

```


Call:
roc.default(response = cars.training$Transport, predictor = cars.logistic$fitted.values)

Data: cars.logistic$fitted.values in 26 controls (cars.training$Transport Car) < 266 cases (car
s.training$Transport Others).
Area under the curve: 0.998
```

Hide

```
predicted.probs = predict.glm(cars.logistic,newdata = cars.test,type = 'response')
predicted.transport = ifelse(predicted.probs<0.92,'Car','Others')
#confusion matrix
table(cars.test$Transport,predicted.transport)
```

```
        predicted.transport
         Car Others
  Car      8      1
  Others   1    116
```

Hide

```
#accuracy
accuracy = sum(diag(table(cars.test$Transport,predicted.transport)))/nrow(cars.test)
accuracy
```

```
[1] 0.984127
```

Hide

```
#TPR
table(cars.test$Transport,predicted.transport)[1,1]/sum(table(cars.test$Transport,predicted.tran
sport)[1,])
```

```
[1] 0.8888889
```

Hide

```
#TNR
table(cars.test$Transport,predicted.transport)[2,2]/sum(table(cars.test$Transport,predicted.tran
sport)[2,])
```

```
[1] 0.991453
```

Hide

```
#AUC
roc(cars.test$Transport,predicted.probs)
```

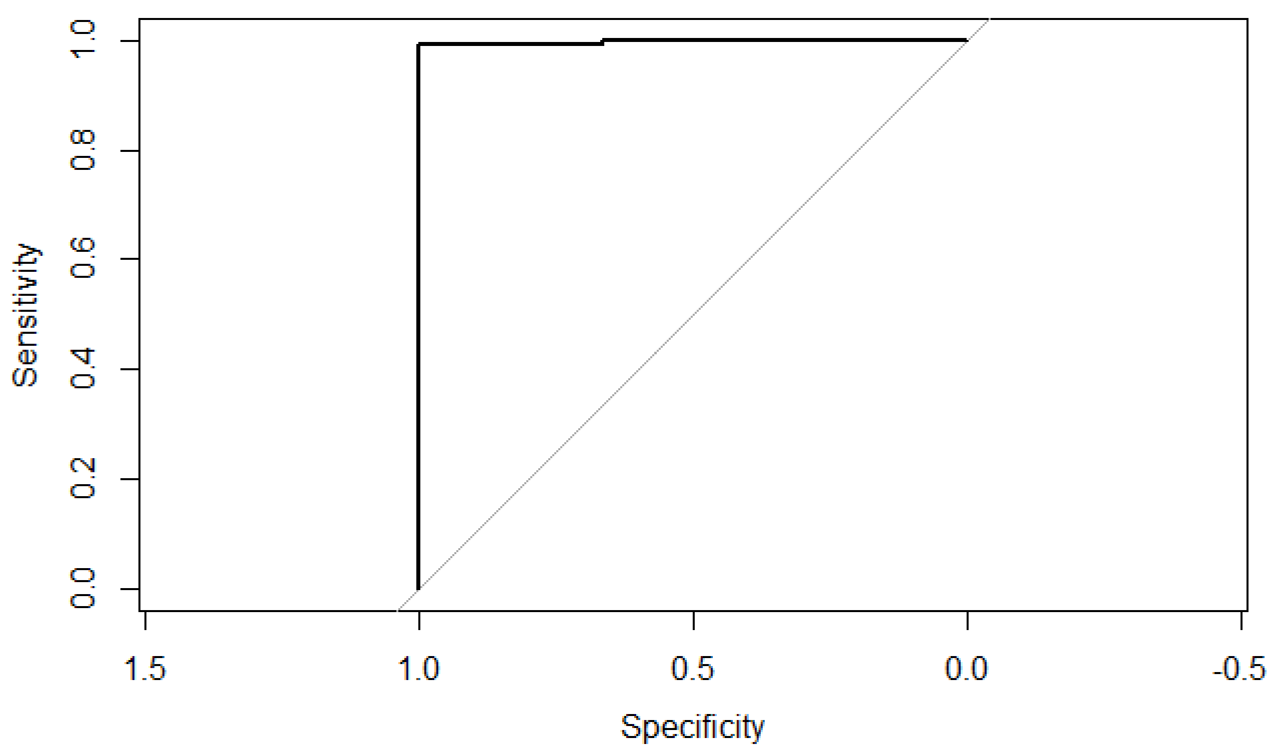
```
Setting levels: control = Car, case = Others
Setting direction: controls < cases
```

```


Call:
roc.default(response = cars.test$Transport, predictor = predicted.probs)

Data: predicted.probs in 9 controls (cars.test$Transport Car) < 117 cases (cars.test$Transport O
thers).
Area under the curve: 0.9972
```

Hide

```
plot(roc(cars.test$Transport,predicted.probs))
```

```
Setting levels: control = Car, case = Others
Setting direction: controls < cases
```



Naive Bayes

Hide

```
colnames(cars.study)
```

```
[1] "Age"       "Gender"    "Engineer"  "MBA"       "Work.Exp"  "Salary"    "Distance"  "licens
e"   "Transport"
```

Hide

```
str(cars.training)
```

```
'data.frame':	292 obs. of  9 variables:
 $ Age      : int  25 29 24 27 26 39 30 28 25 27 ...
 $ Gender   : Factor w/ 2 levels "Female","Male": 2 1 2 1 2 2 2 1 2 1 ...
 $ Engineer : Factor w/ 2 levels "0","1": 2 1 2 1 1 2 1 2 2 1 ...
 $ MBA      : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 1 1 1 ...
 $ Work.Exp : int  3 7 6 4 5 21 8 5 1 4 ...
 $ Salary   : num  9.9 14.6 12.7 13.6 12.6 50 14.6 14.6 8.6 13.9 ...
 $ Distance : num  17.2 7.7 8.7 8.2 11.1 23.4 10.9 9 9.4 17.3 ...
 $ license  : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 1 1 ...
 $ Transport: Factor w/ 2 levels "Car","Others": 2 2 2 2 2 1 2 2 2 2 ...
```

Hide

```
str(cars.test)
```

```
'data.frame':	126 obs. of  9 variables:
 $ Age      : int  28 27 21 23 21 27 23 29 29 28 ...
 $ Gender   : Factor w/ 2 levels "Female","Male": 2 1 2 2 2 2 2 1 2 2 ...
 $ Engineer : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 2 1 2 2 ...
 $ MBA      : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 1 1 2 ...
 $ Work.Exp : int  5 9 3 0 3 8 2 7 9 5 ...
 $ Salary   : num  14.4 15.5 9.5 6.5 10.6 15.6 8.8 14.6 23.8 14.8 ...
 $ Distance : num  5.1 6.1 7.1 7.3 7.7 9 9.2 9.2 9.4 10.8 ...
 $ license  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 2 ...
 $ Transport: Factor w/ 2 levels "Car","Others": 2 2 2 2 2 2 2 2 2 2 ...
```

Hide

```
cars.nb = naiveBayes(Transport~Age+Gender+Work.Exp+Salary+Distance+license,data=cars.training,la
place = T)
```

Hide

```
cars.nb$tables
```

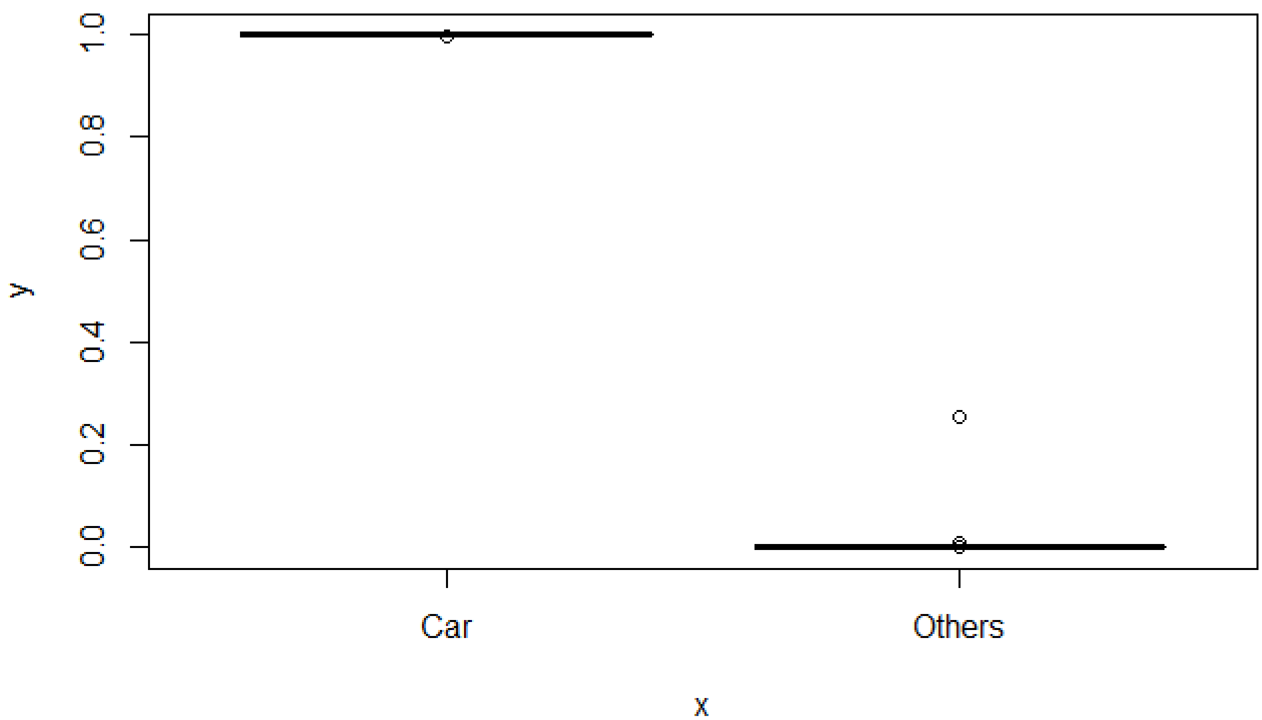
```
$Age
        Age
Y             [,1]     [,2]
  Car    36.88462 3.115470
  Others 26.49811 3.003943

$Gender
        Gender
Y           Female      Male
  Car    0.1785714 0.8214286
  Others 0.2958801 0.7041199

$Work.Exp
        Work.Exp
Y              [,1]     [,2]
  Car    17.692308 4.067129
  Others  4.728302 3.245298

$Salary
        Salary
Y             [,1]      [,2]
  Car    41.13462 10.388838
  Others 13.03057  5.386889

$Distance
        Distance
Y             [,1]     [,2]
  Car    17.95769 2.914951
  Others 10.86717 3.073349

$license
        license
Y                0         1
  Car    0.2142857 0.7857143
  Others 0.8426966 0.1573034
```

Hide

```
predicted.probs = predict(cars.nb,newdata = cars.test,type = 'raw')
```

Hide

```
plot(cars.test$Transport,predicted.probs[,1])
```



Hide

```
predicted.transport = ifelse(predicted.probs[,1]>.92,'Car','Others')
```

Hide

```
table(cars.test$Transport,predicted.transport)
```

```
        predicted.transport
         Car Others
  Car      9      0
  Others   0    117
```

Hide

```
roc(cars.test$Transport,predicted.probs[,1])
```

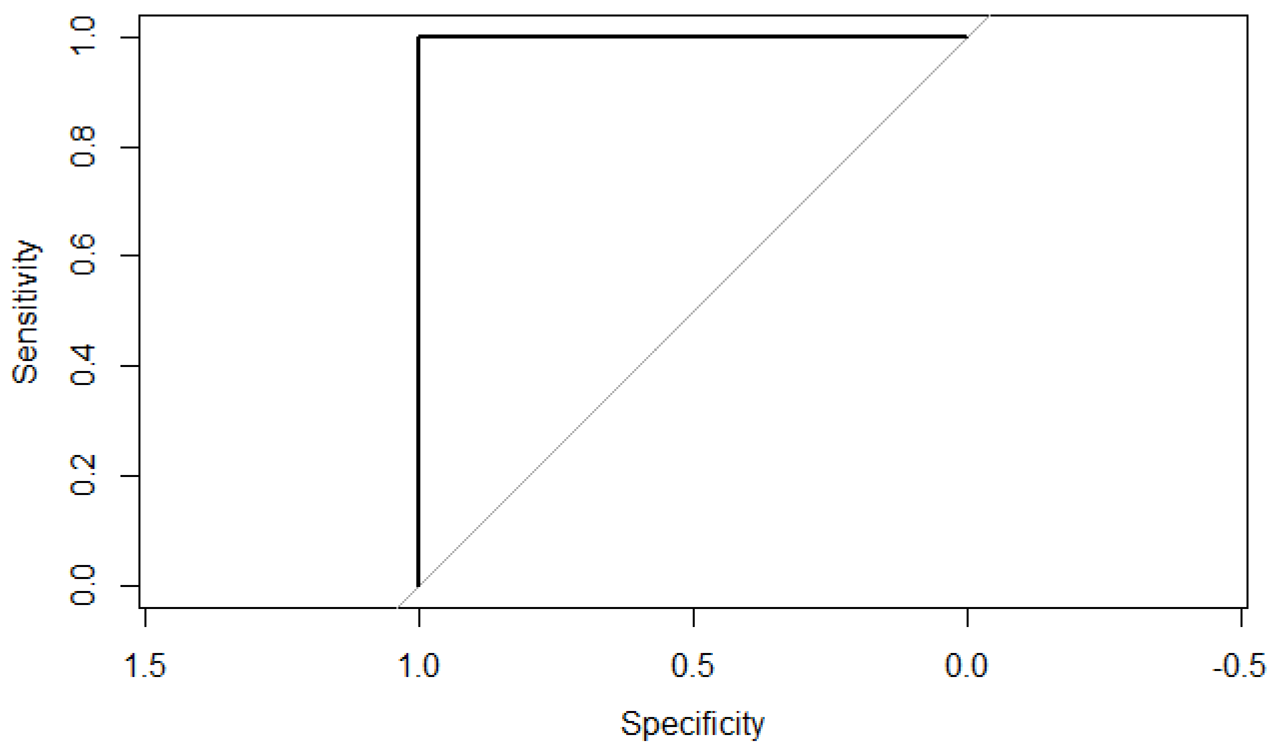
```
Setting levels: control = Car, case = Others
Setting direction: controls > cases
```

```


Call:
roc.default(response = cars.test$Transport, predictor = predicted.probs[,     1])

Data: predicted.probs[, 1] in 9 controls (cars.test$Transport Car) > 117 cases (cars.test$Transp
ort Others).
Area under the curve: 1
```

Hide

```
plot(roc(cars.test$Transport,predicted.probs[,1]))
```

```
Setting levels: control = Car, case = Others
Setting direction: controls > cases
```



#KNN

Hide

```
cars.study[c(1,5,6,7)]
```

Age <int>	Work.Exp <int>	Salary <dbl>	Distance <dbl>
28	5	14.4	5.1
24	6	10.6	6.1
27	9	15.5	6.1
25	1	7.6	6.3
25	3	9.6	6.7
21	3	9.5	7.1
23	3	11.7	7.2
23	0	6.5	7.3

Age	Work.Exp	Salary	Distance
<int>	<int>	<dbl>	<dbl>
24	4	8.5	7.5
28	6	13.7	7.5

1-10 of 418 rows Previous 1 2 3 4 5 6 ... 42 Next

Hide

```
set.seed(10)
TPR = c()
accuracy = c()
for (i in 1:10){
   cars.knn = knn(scale(cars.training[,c(1,5,6,7)]),scale(cars.test[,c(1,5,6,7)]),cars.training
[,c(9)],k=i)
   conf.matrix = table(cars.test$Transport,cars.knn)
   TPR[i] = diag(conf.matrix)[1]/sum(conf.matrix[1,])
   accuracy[i] = sum(diag(conf.matrix))/nrow(cars.test)
}
TPR
```

```
 [1] 0.8888889 0.8888889 1.0000000 1.0000000 0.8888889 1.0000000 0.8888889
 [8] 0.8888889 0.8888889 0.8888889
```

Hide

```
accuracy
```

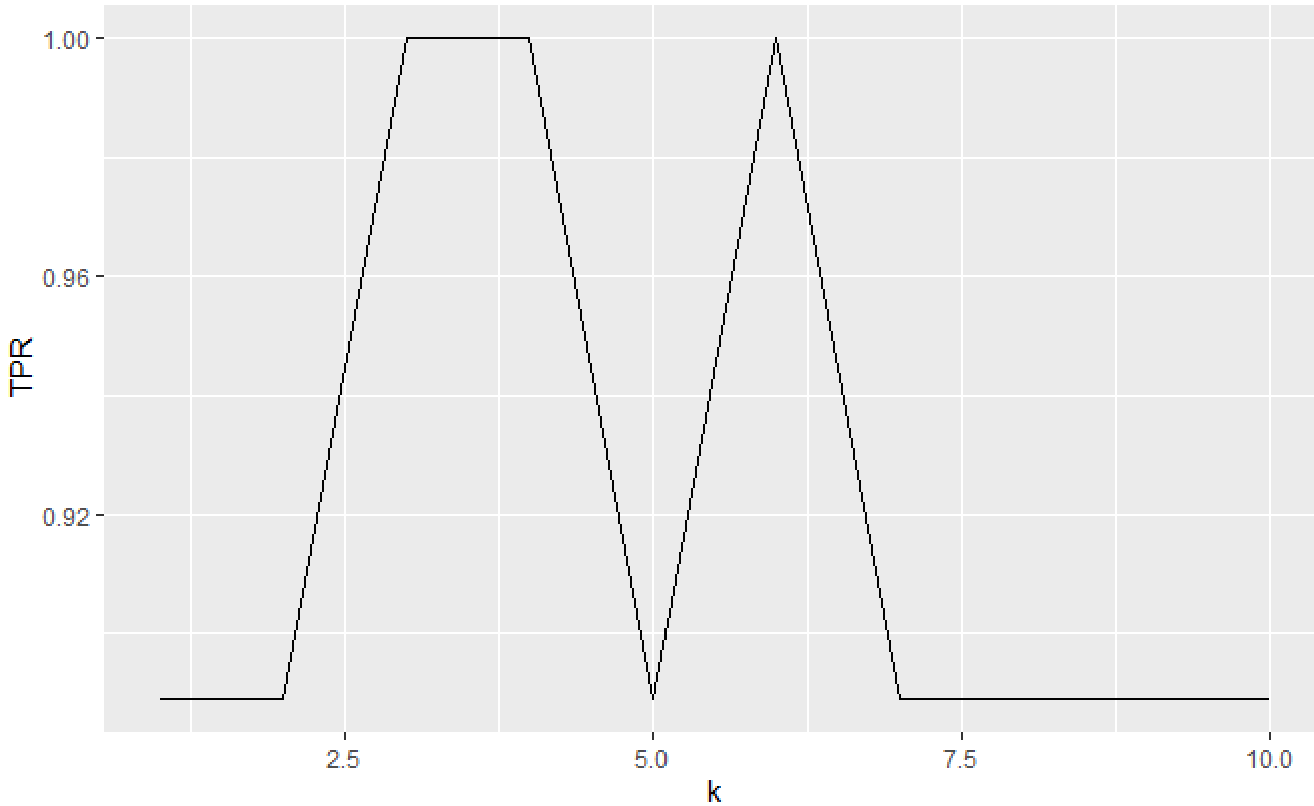
```
 [1] 0.9841270 0.9761905 0.9841270 0.9920635 0.9841270 0.9920635 0.9920635
 [8] 0.9841270 0.9920635 0.9920635
```

Hide

```
TPR = as.data.frame(cbind(k=c(1:10),TPR))
```

Hide

```
qplot(x=k,y=TPR,data = TPR,geom = 'line')
```

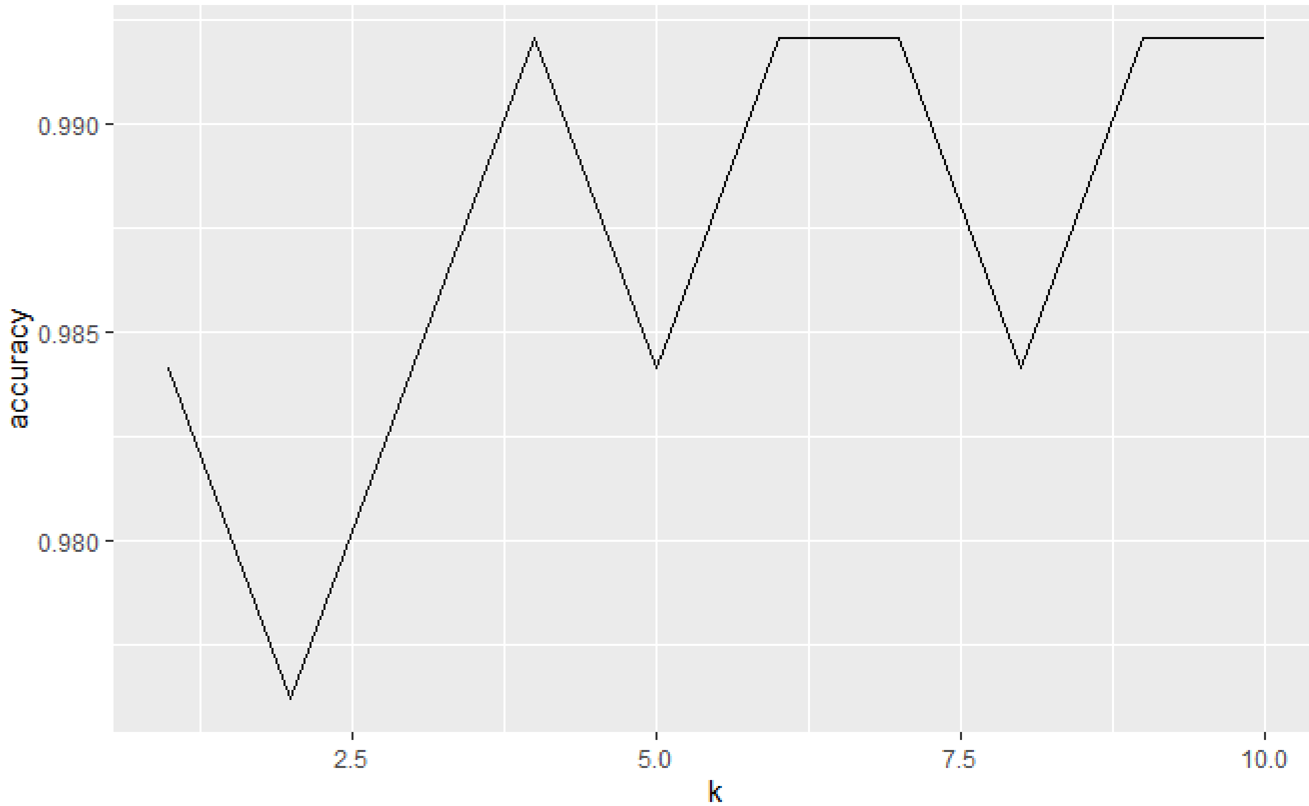


Hide

```
accuracy = as.data.frame(cbind(k=c(1:10),accuracy))
```

Hide

```
qplot(x=k,y=accuracy,data = accuracy,geom = 'line')
```



Hide

```
cars.knn = knn(scale(cars.training[,c(1,5,6,7)]),scale(cars.test[,c(1,5,6,7)]),cars.training[,c(
9)],k=6)
```

Hide

```
table(cars.test$Transport,cars.knn)
```

```
        cars.knn
         Car Others
  Car      9      0
  Others   0    117
```

Hide

```
#TPR
table(cars.test$Transport,cars.knn)[1,1]/sum(table(cars.test$Transport,cars.knn)[1,])
```

```
[1] 1
```

Hide

```
#TNR
table(cars.test$Transport,cars.knn)[2,2]/sum(table(cars.test$Transport,cars.knn)[2,])
```

```
[1] 1
```

Hide

```
accuracy = (9+117)/nrow(cars.test)
accuracy
```

```
[1] 1
```

#Bagging

Hide

```
?bagging
cars.bagging = bagging(Transport ~.,data = cars.training,control = rpart.control(minbucket = 5,c
p=0,xval = 10),na.action=na.rpart)
```

Hide

```
varImp(cars.bagging)
```

	Overall <dbl>
Age	35.27835714
Distance	28.34621662
Engineer	0.00745758
Gender	0.36577802
license	10.52281186
MBA	0.39299695
Salary	39.59553236
Work.Exp	34.75128614

8 rows

Hide

```
set.seed(1)
cars.bagging = bagging(Transport ~ Age+Distance+license+Salary+Work.Exp,data = cars.training,con
trol = rpart.control(minbucket = 5, cp = 0, xval = 10))
```

Hide

```
table(cars.training$Transport,cars.bagging$y)
```

```
         Car Others
  Car     26      0
  Others   0    266
```

Hide

```
predicted.transport = predict(cars.bagging,cars.test)
predicted.probs = predict(cars.bagging,cars.test,'prob')
```

Hide

```
table(cars.test$Transport,predicted.transport)
```

```
        predicted.transport
         Car Others
  Car      6      3
  Others   0    117
```

Hide

```
#TPR
table(cars.test$Transport,predicted.transport)[1,1]/sum(table(cars.test$Transport,predicted.tran
sport)[1,])
```

```
[1] 0.6666667
```

Hide

```
#TNR
table(cars.test$Transport,predicted.transport)[2,2]/sum(table(cars.test$Transport,predicted.tran
sport)[2,])
```

```
[1] 1
```

Hide

```
accuracy = sum(diag(table(cars.test$Transport,predicted.transport)))/nrow(cars.test)
accuracy
```

```
[1] 0.9761905
```

Hide

```
roc(cars.test$Transport,predicted.probs[,1])
```

Hide

```
plot(roc(cars.test$Transport,predicted.probs[,1]))
```

#boosting

Hide

```
str(cars.training)
#cars.training.1h = model.matrix(~0+cars.training[trainidx,'Gender'])
cars.training.gender = one_hot(as.data.table(cars.training$Gender))
names(cars.training.gender) = c('Female','Male')
```

Hide

```
cars.test.gender = one_hot(as.data.table(cars.test$Gender))
```

Hide

```
names(cars.test.gender) = c('Female','Male')
```

Hide

```
cars.training.1h = cbind(cars.training[-c(2)],cars.training.gender)
cars.test.1h = cbind(cars.test[-c(2)],cars.test.gender)
```

Hide

```
cars.training.1h$Transport = ifelse(cars.training.1h$Transport=='Car','1','0')
```

Hide

```
cars.test.1h$Transport = ifelse(cars.test.1h$Transport=='Car','1','0')
```

Hide

```
str(cars.training.1h)
cars.training.1h$Engineer = as.integer(cars.training.1h$Engineer)
cars.training.1h$MBA = as.integer(cars.training.1h$MBA)
cars.training.1h$license = as.integer(cars.training.1h$license)
cars.training.1h$Transport = as.integer(cars.training.1h$Transport)
```

Hide

```
head(cars.training.1h)
```

	... <int>	Engineer <int>	... <int>	Work.Exp <int>	Salary <dbl>	Distance <dbl>	license <int>	Transport <int>	Female <int>
416	27	1	1	4	13.9	17.3	1	0	1
179	29	1	1	7	14.6	7.7	1	0	1
14	24	2	1	6	12.7	8.7	1	0	0
195	27	1	2	4	13.6	8.2	1	0	1
307	29	2	1	5	14.9	11.2	1	0	0
118	39	2	2	21	50.0	23.4	2	1	0

6 rows | 1-10 of 10 columns

Hide

```
str(cars.training.1h)
```

```
'data.frame':   291 obs. of  10 variables:
 $ Age      : int  27 29 24 27 29 39 33 28 25 23 ...
 $ Engineer : int  1 1 2 1 2 2 2 2 2 1 ...
 $ MBA      : int  1 1 1 2 1 2 2 1 1 1 ...
 $ Work.Exp : int  4 7 6 4 5 21 14 5 1 3 ...
 $ Salary   : num  13.9 14.6 12.7 13.6 14.9 50 34.9 14.6 8.6 9.9 ...
 $ Distance : num  17.3 7.7 8.7 8.2 11.2 23.4 10.9 9 9.4 17.9 ...
 $ license  : int  1 1 1 1 1 2 1 1 1 1 ...
 $ Transport: int  0 0 0 0 0 1 0 0 0 0 ...
 $ Female   : int  1 1 0 1 0 0 0 1 1 0 ...
 $ Male     : int  0 0 1 0 1 1 1 0 0 1 ...
```

Hide

```
cars.xgb.fit = xgboost(
  data = as.matrix(cars.training.1h[,-c(8)]),
  label = as.matrix(cars.training.1h[,c(8)]),
  eta = 0.3,#this is like shrinkage in the previous algorithm
  max_depth = 5,#Larger the depth, more complex the model; higher chances of overfitting. There
 is no standard                      value for max_depth. Larger data sets require deep trees to
learn the rules from data.
  min_child_weight = 5,#it blocks the potential feature interactions to prevent overfitting
  nrounds = 100,#controls the maximum number of iterations. For classification, it is similar to
the number of                       trees to grow.
  nfold = 5,
  objective = "binary:logistic",  # for regression models
  verbose = 0,               # silent,
  early_stopping_rounds = 10 # stop if no improvement for 10 consecutive trees
)
```

Hide

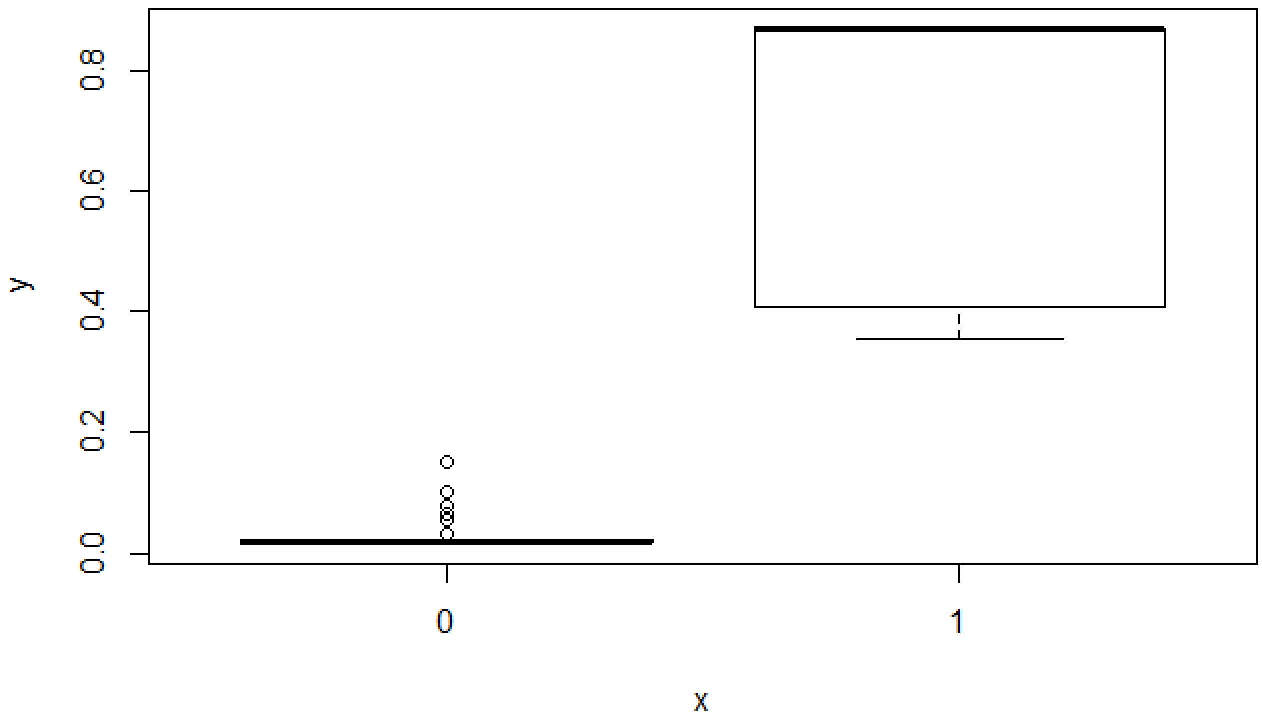
```
cars.test.1h$Engineer = as.integer(cars.test.1h$Engineer)
cars.test.1h$MBA = as.integer(cars.test.1h$MBA)
cars.test.1h$license = as.integer(cars.test.1h$license)
cars.test.1h$Transport = as.integer(cars.test.1h$Transport)
```

Hide

```
predicted.probs = predict(cars.xgb.fit,as.matrix(cars.test.1h[,-c(8)]))
```

Hide

```
plot(as.factor(cars.test.1h$Transport),predicted.probs)
```



Hide

```
predicted.transport = ifelse(predicted.probs>0.3,'1','0')
table(as.factor(cars.test.1h$Transport),predicted.transport)
```

```
   predicted.transport
      0   1
  0 117   0
  1   0   9
```

Hide

```
roc(as.factor(cars.test.1h$Transport),predicted.probs)
```

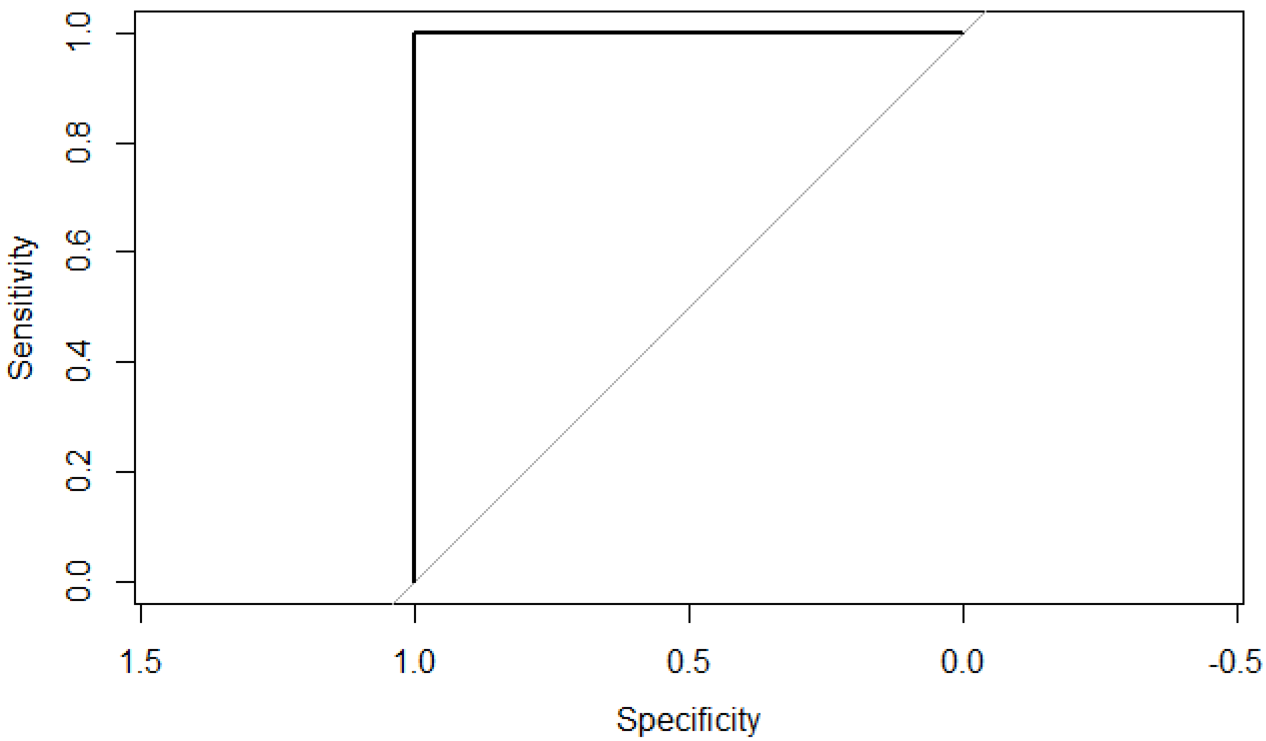
```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```


Call:
roc.default(response = as.factor(cars.test.1h$Transport), predictor = predicted.probs)

Data: predicted.probs in 117 controls (as.factor(cars.test.1h$Transport) 0) < 9 cases (as.factor
(cars.test.1h$Transport) 1).
Area under the curve: 1
```

Hide

```
plot(roc(as.factor(cars.test.1h$Transport),predicted.probs))
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```



#SMOTE

Hide

```
table(cars.training$Transport)
```

```
   Car Others 
    26    266 
```

Hide

```
cars.training.smote = SMOTE(Transport~Age+Gender+Distance+license,data = cars.training,perc.over
= 200,k=5,perc.under = 500)
```

Hide

```
table(cars.training.smote$Transport)
```

```
   Car Others 
    78    260 
```

#Logistic Regression after SMOTE

Hide

```
cars.logistic.smote = glm(Transport~Age+Gender+Distance+license,data = cars.training.smote,famil
y = 'binomial')
```

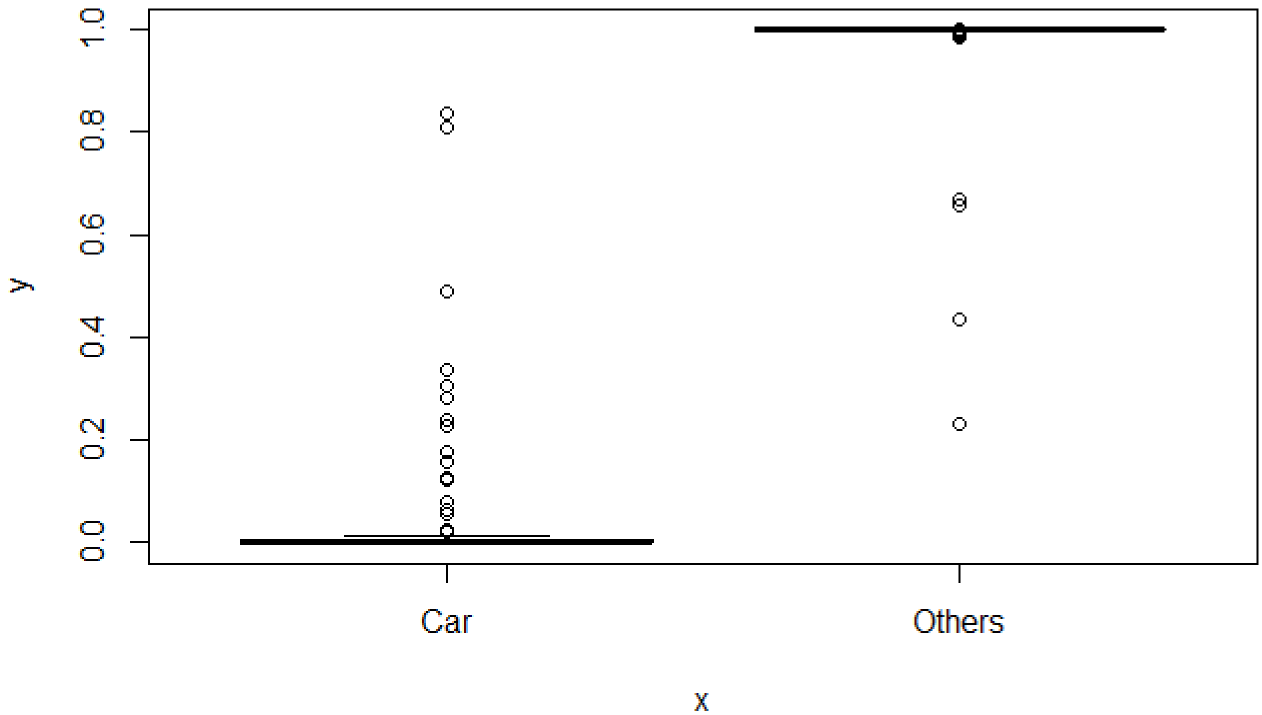
Hide

```
summary(cars.logistic.smote)
```

```
Call:
glm(formula = Transport ~ Age + Gender + Distance + license, 
    family = "binomial", data = cars.training.smote)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.90690   0.00004   0.00126   0.01148   1.71020  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)   
(Intercept)  54.9142    16.9123   3.247  0.00117 **
Age          -1.2288     0.4151  -2.960  0.00307 **
GenderMale    0.2488     1.0936   0.227  0.82005   
Distance     -0.9556     0.3433  -2.784  0.00538 **
license1     -2.7925     1.3328  -2.095  0.03615 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 365.18  on 337  degrees of freedom
Residual deviance:  27.15  on 333  degrees of freedom
AIC: 37.15

Number of Fisher Scoring iterations: 10
```

Hide

```
plot(cars.training.smote$Transport,cars.logistic.smote$fitted.values)
```



Hide

```
predicted.transport = ifelse(cars.logistic.smote$fitted.values<.90,'Car','Others')
```

Hide

```
table(cars.training.smote$Transport,predicted.transport)
```

```
        predicted.transport
         Car  Others
  Car     78       0
  Others   8     252
```

Hide

```
predicted.probs = predict.glm(cars.logistic.smote,newdata = cars.test,type = 'response')
```

Hide

```
predicted.transport = ifelse(predicted.probs<.90,'Car','Others')
```

Hide

```
table(cars.test$Transport,predicted.transport)
```

```
       predicted.transport
        Car Others
  Car      9      0
  Others   3    114
```

Hide

```
accuracy = sum(diag(table(cars.test$Transport,predicted.transport)))/nrow(cars.test)
accuracy
```

```
[1] 0.9761905
```

Hide

```
#TPR
table(cars.test$Transport,predicted.transport)[1,1]/sum(table(cars.test$Transport,predicted.tran
sport)[1,])
```

```
[1] 1
```

Hide

```
#TNR
table(cars.test$Transport,predicted.transport)[2,2]/sum(table(cars.test$Transport,predicted.tran
sport)[2,])
```

```
[1] 0.974359
```

Hide

```
#AUC
roc(cars.test$Transport,predicted.probs)
```

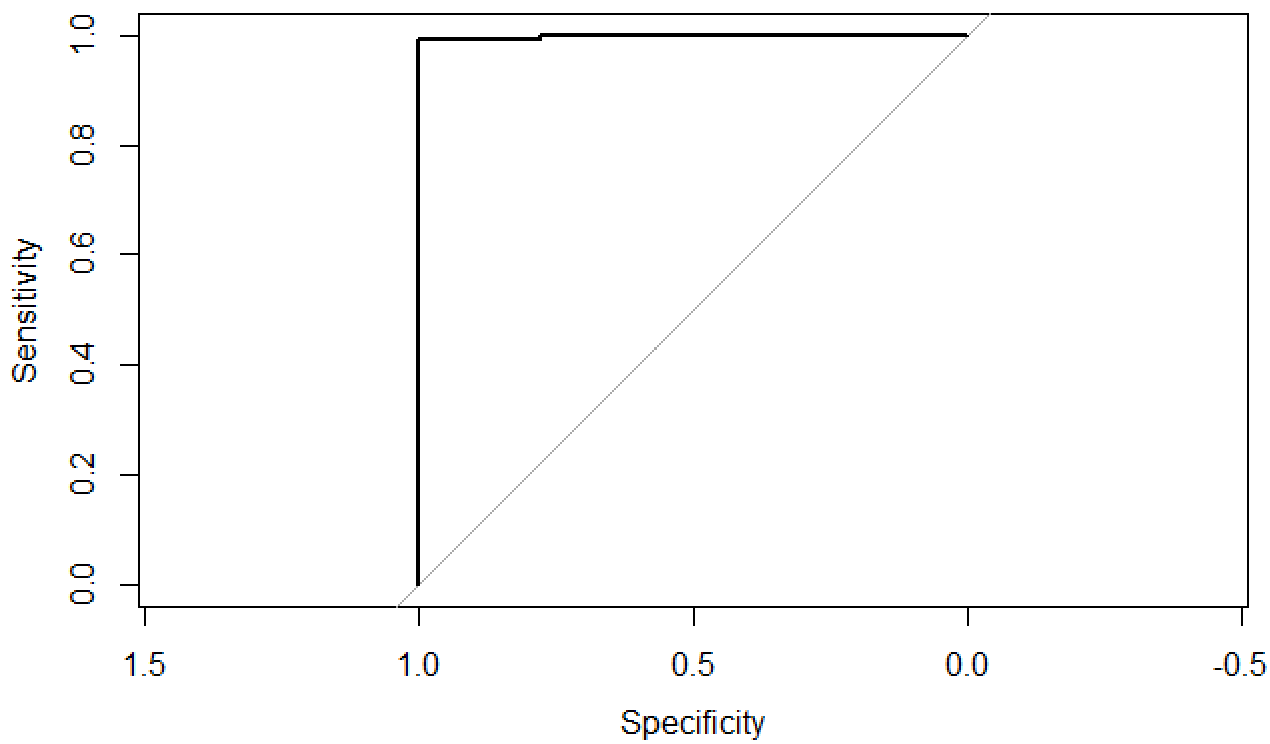
```
Setting levels: control = Car, case = Others
Setting direction: controls < cases
```

```


Call:
roc.default(response = cars.test$Transport, predictor = predicted.probs)

Data: predicted.probs in 9 controls (cars.test$Transport Car) < 117 cases (cars.test$Transport O
thers).
Area under the curve: 0.9981
```

Hide

```
plot(roc(cars.test$Transport,predicted.probs))
```

```
Setting levels: control = Car, case = Others
Setting direction: controls < cases
```



#Bagging after SMOTE

Hide

```
cars.training.smote = SMOTE(Transport ~ .,data = cars.training,perc.over = 200,k=5,perc.under =
600)
```

Hide

```
table(cars.training.smote$Transport)
```

```

   Car Others 
    78    312 
```

Hide

```
set.seed(2)
cars.bagging.smote = bagging(Transport ~ Age+Distance+license+Salary+Work.Exp,data = cars.traini
ng.smote,control = rpart.control(minbucket = 5, cp = 0, xval = 10,na.action=na.rpart))
```

Hide

```
varImp(cars.bagging.smote)
```

	Overall <dbl>
Age	108.30912
Distance	81.10129
license	39.34404
Salary	111.57718
Work.Exp	109.55873

5 rows

Hide

```
length(cars.bagging.smote$y)
```

```
[1] 390
```

Hide

```
table(cars.training.smote$Transport,cars.bagging.smote$y)
```

```
        Car Others
  Car     78      0
  Others   0    312
```

Hide

```
predicted.transport = predict(cars.bagging.smote,newdata = cars.test)
predicted.probs = predict(cars.bagging.smote,newdata = cars.test,'prob')
```

Hide

```
table(cars.test$Transport,predicted.transport)
```

```
       predicted.transport
        Car Others
  Car      9      0
  Others   1    116
```

Hide

```
roc(cars.test$Transport,predicted.probs[,1])
```

```
Setting levels: control = Car, case = Others
Setting direction: controls > cases
```

```
Call:
roc.default(response = cars.test$Transport, predictor = predicted.probs[,     1])

Data: predicted.probs[, 1] in 9 controls (cars.test$Transport Car) > 117 cases (cars.test$Transp
ort Others).
Area under the curve: 0.9995
```

Hide

```
plot.roc(roc(cars.test$Transport,predicted.probs[,1]))
```

```
Setting levels: control = Car, case = Others
Setting direction: controls > cases
```