MVA_Ass_8.R

mihikagupta

2020-11-05

```
###### Assignment 8 ##############
## Applying Logistic Regression Analysis

#Getting working directory
getwd()
```

```
## [1] "/Users/mihikagupta/Desktop/SEM_2/MVA"
```

```
#Setting directory to load data set
setwd("/Users/mihikagupta/Desktop/SEM_2/MVA")

#Reading the data into a data frame
#df <- read.csv(file = 'US_Acc_June20.csv')
num <- read.csv(file = 'num.csv')
# Performing clustering on the first 500 records for now to achieve easy and quick results and test the
attach(num)
# Printing first few columns of data set for inference
#head(df)

## Setting random seed to shuffle data before splitting
set.seed(23)

#Checking number of rows
#rows<-sample(nrow(df))

#Shuffling the data
#mva<-df[rows,]

#Taking the required number of instances from the shuffled data to reduce any biases
#mva<-mva[950000:1000000,]

#Checking the structure of the data set
#str(mva)

# Checking the number of rows and columns in the current uncleaned dataset
#ncol(mva)
#nrow(mva)

# Printing all the column names to find and filter the relevant and irrelevant attributes
#names<-names(mva)
#names

## DATA CLEANING ##
```

```
#Dropping the surplus attributes which do not contribute to the analysis
#mva <- mva[-c(1:3,7:10,13,14,19,21:23,33,47:49)]

#Checking for any null values in the present data set
# is.na(mva[,])

#Checking which rows have all the values filled and complete
# complete.cases(mva)

#Making a new dataframe with only the rows that have complete information and all values filled
#Mva<-na.omit(mva)
#Mva<-Mva[!(is.na(Mva$Sunrise_Sunset) | Mva$Sunrise_Sunset==""), ]
#Mva<- Mva[complete.cases(Mva),]
#Verifying for missing values in the new dataframe
#complete.cases(Mva)
#unique(Mva$Sunrise_Sunset)

# Creating new dataframe with only the numerical attributes to perform statistical functions
#num<-Mva[,c(1,4,11:15,17,18)]
#write.csv(num,"/Users/mihikagupta/Desktop/SEM_2/MVA/num.csv", row.names = FALSE)

# Scaling the new data set for better accuracies
# num<-scale(num)

# Checking the dimensions of the data
nrow(num)
```

```
## [1] 18250
```

```
ncol(num)
```

```
## [1] 9
```

```
names(num)
```

```
## [1] "Severity"          "Distance.mi."      "Temperature.F."
## [4] "Wind_Chill.F."     "Humidity..."       "Pressure.in."
## [7] "Visibility.mi."    "Wind_Speed.mph."   "Precipitation.in."
```

```
names(num)[names(num) == "Distance.mi."] <- "dist"
names(num)[names(num) == "Temperature.F."] <- "temp"
names(num)[names(num) == "Wind_Chill.F."] <- "windchill"
names(num)[names(num) == "Humidity..."] <- "humidity"
names(num)[names(num) == "Pressure.in."] <- "pressure"
names(num)[names(num) == "Visibility.mi."] <- "visibility"
names(num)[names(num) == "Wind_Speed.mph."] <- "windspeed"
names(num)[names(num) == "Precipitation.in."] <- "precip"
names(num)
```

```
## [1] "Severity"   "dist"       "temp"       "windchill"  "humidity"
## [6] "pressure"   "visibility" "windspeed"  "precip"
```

```
# finding covariance,Covariance measures the linear relationship between two variables. ... The correla
cov(num)
```

```
##                Severity          dist          temp     windchill     humidity
## Severity    0.310236580  0.1670840167   -0.35171651   -0.47620943    0.6439822
## dist        0.167084017  2.4204304155   -0.46482491   -0.64706455    0.5613710
## temp       -0.351716514 -0.4648249146  354.39442593  397.07844765 -185.2055863
## windchill  -0.476209435 -0.6470645543  397.07844765  450.54026468 -200.8257533
## humidity    0.643982246  0.5613709540 -185.20558632 -200.82575330  530.2655286
## pressure   -0.005748438 -0.0567534649    1.01969027    1.16018376    5.2382330
## visibility -0.049013351 -0.1056278936   16.89140979   19.68781198  -27.8803082
## windspeed   0.143762668  0.1759889303   -1.21123799   -7.64665928  -18.3282076
## precip      0.001178471  0.0004767836   -0.05005218   -0.05860904    0.1685233
##                pressure   visibility    windspeed        precip
## Severity   -0.005748438  -0.04901335   0.14376267  0.0011784711
## dist       -0.056753465  -0.10562789   0.17598893  0.0004767836
## temp        1.019690266  16.89140979  -1.21123799 -0.0500521830
## windchill   1.160183765  19.68781198  -7.64665928 -0.0586090372
## humidity    5.238233012 -27.88030819 -18.32820757  0.1685233120
## pressure    1.319215237  -0.28537521  -0.29431416  0.0015056899
## visibility -0.285375212   8.02960442  -0.42228979 -0.0303542240
## windspeed  -0.294314160  -0.42228979  29.19965932  0.0161919228
## precip      0.001505690  -0.03035422   0.01619192  0.0070357365
```

```r
# here we find that the highest covariances with severity in either directions, positive or negative ar


############# TRYING IF LOGISTIC REGRESSION IS A GOOD CHOICE FOR OUR DATASET ################

# Let us first perform a simple multiple regression with some variables
fit<-lm(Severity~dist+temp+windchill+humidity,data = num)

# showing results
summary(fit)
```

```
##
## Call:
## lm(formula = Severity ~ dist + temp + windchill + humidity, data = num)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6912 -0.2745 -0.2413  0.5915  1.8145
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.0659506  0.0292263  70.688  < 2e-16 ***
## dist         0.0679376  0.0025946  26.184  < 2e-16 ***
## temp         0.0160049  0.0019505   8.205 2.45e-16 ***
## windchill   -0.0145144  0.0017157  -8.460  < 2e-16 ***
## humidity     0.0012356  0.0001955   6.319 2.70e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5449 on 18245 degrees of freedom
## Multiple R-squared:  0.04329,    Adjusted R-squared:  0.04308
## F-statistic: 206.4 on 4 and 18245 DF,  p-value: < 2.2e-16
```

```r
coefficients(fit)
```

```
##  (Intercept)         dist         temp    windchill     humidity
##  2.065950617  0.067937558  0.016004929 -0.014514415  0.001235566
```

```r
# Performing initial logistic regression on  dataset
# logistic_fit<-glm(Severity~dist+temp+windchill+humidity,data = num,family="binomial")
## When we apply this regression , we get the following error
# Error in eval(family$initialize) : y values must be 0 <= y <= 1

# Alternate Explaination
unique(num$Severity)
```

```
## [1] 2 3 1 4
```

```r
# the above result shows that our dependent variable that is "SEVERITY" is not binary , but
# has 4 categories, namesly "1","2","3","4", therefore we cannot use the simple
# binomial logistic regression on this dataset since it is only applicable for binary classification.

############## Computing multinomial logistic regression ###########
# therefore we now try the multinomial logistic regression using the "caret" and "nnet"libraries
library("nnet")
library("caret")
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

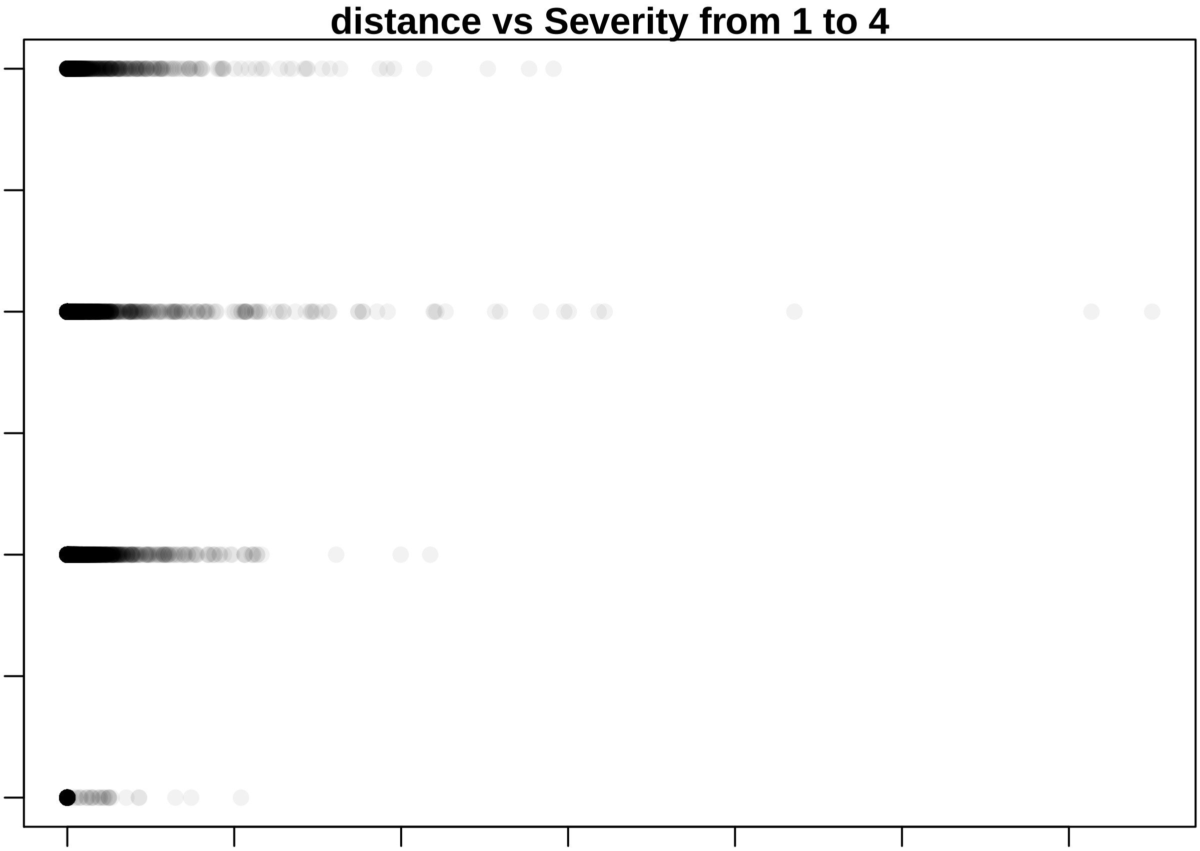
```r
library("magrittr")

# Now lets divide our dataset into 2 parts, the training and testing sets for checking our model accura
num1<-num[1:400,]
num2<-num[400:500,]

# applying the multinomial log regression to training set
model<-nnet::multinom(Severity~.,data=num1)
```

```
## # weights:  40 (27 variable)
## initial  value 554.517744
## iter  10 value 363.079681
## iter  20 value 316.027015
## iter  30 value 293.716007
## iter  40 value 292.639663
## iter  50 value 291.528928
## iter  60 value 291.448698
## iter  70 value 291.443695
## iter  80 value 291.313810
## iter  90 value 291.001479
## iter 100 value 290.178494
## final  value 290.178494
## stopped after 100 iterations
```

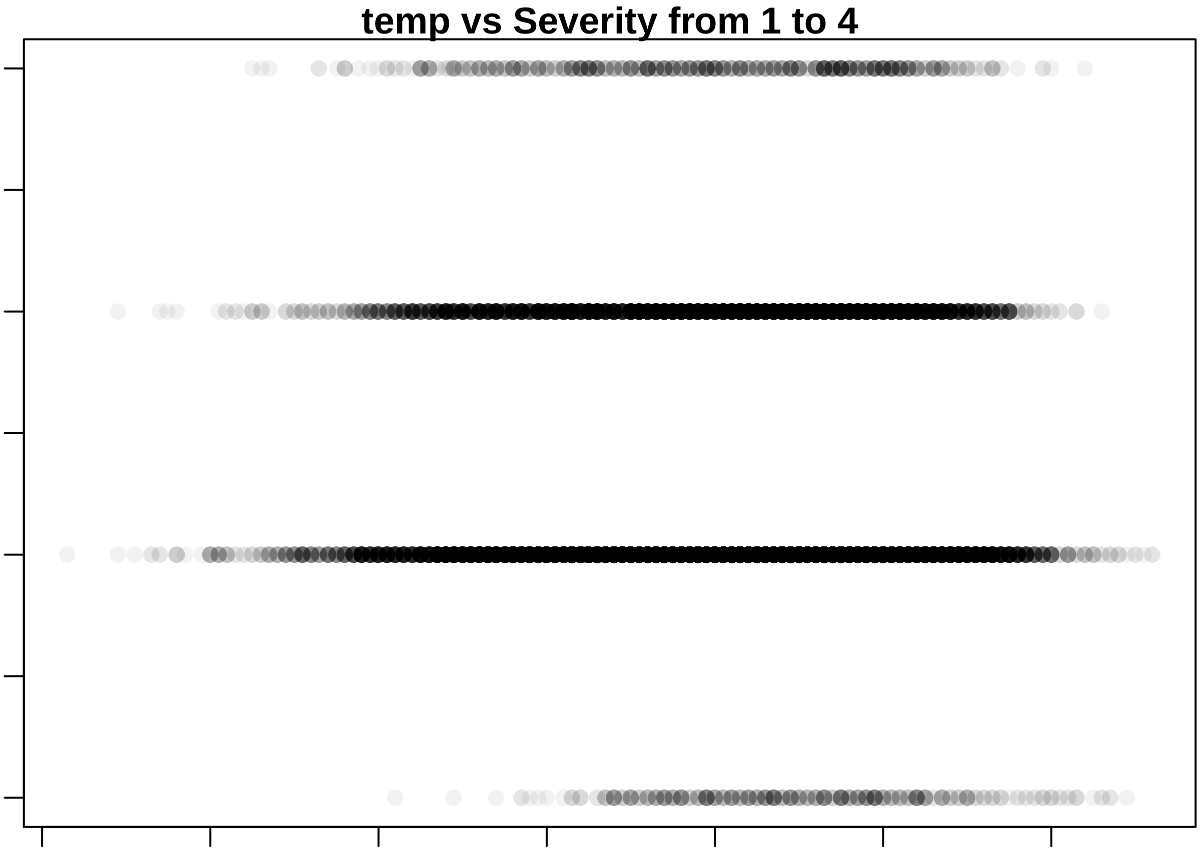
```r
#predicting class of outcome variable
p1<-predict(model,num2,type="class")

# predicting probability of outcome being true
p2<-predict(model,num2,type="probs")

# Rowsums
rowSums(p2)
```

```
## 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 500
##   1
```

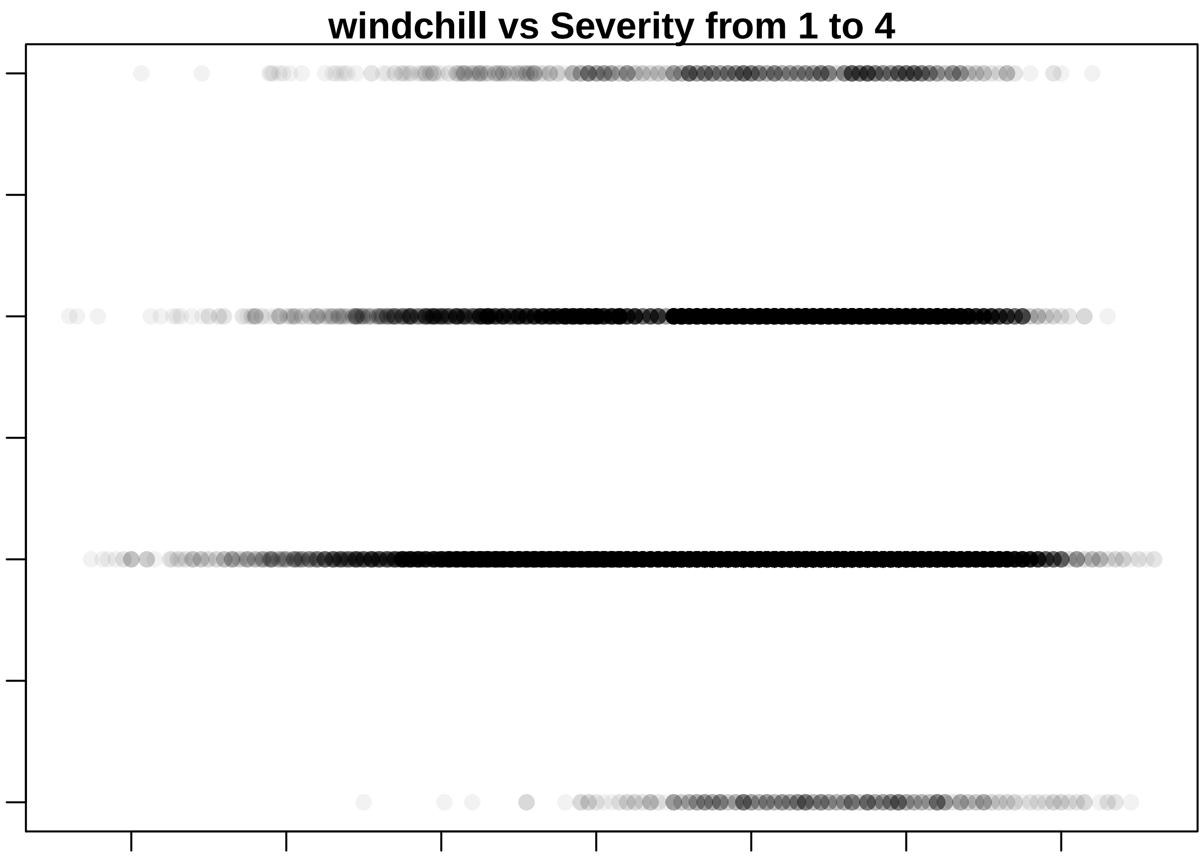
```r
par(mar=c(1,1,1,1))


plot(Severity ~ dist, col = rgb(0, 0, 0, 0.05), pch = 19,data = num,main="distance vs Severity from 1 t
```


distance vs Severity from 1 to 4

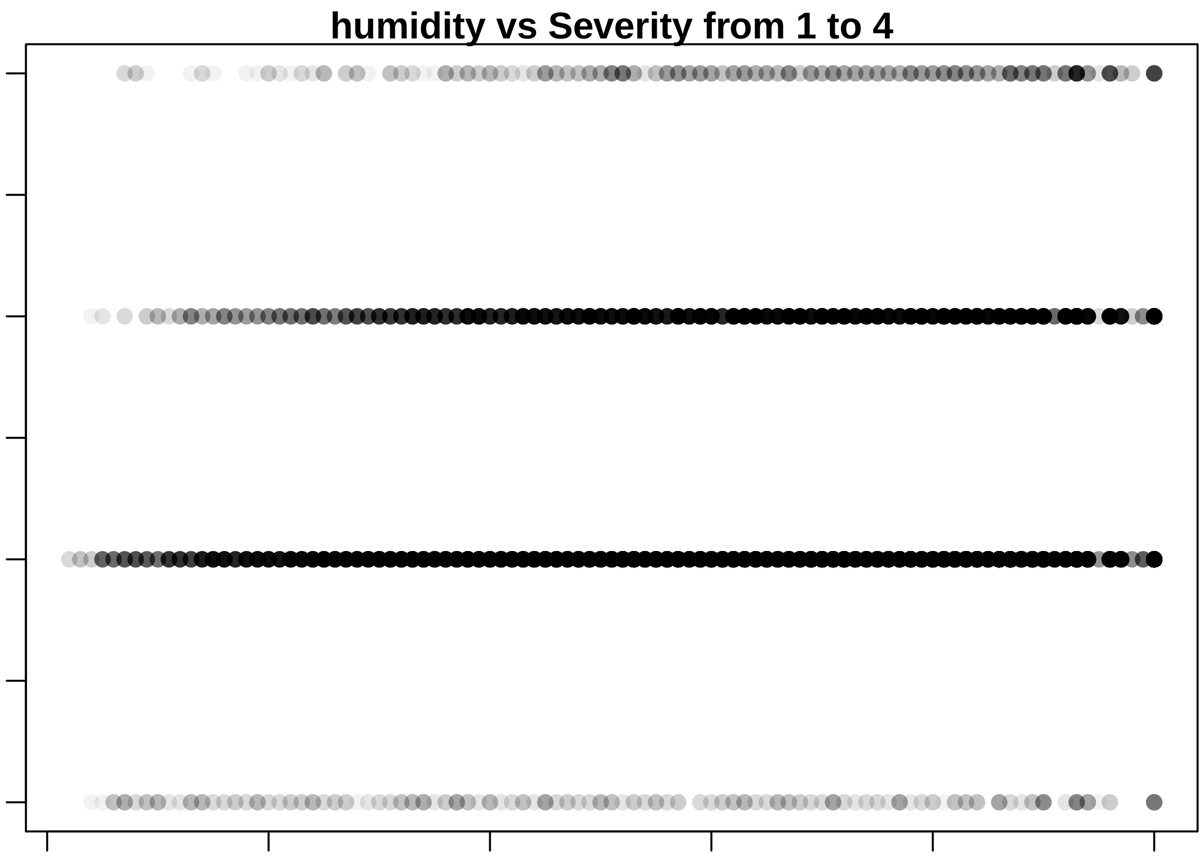
```r
plot(Severity ~ temp, col = rgb(0, 0, 0, 0.05), pch = 19,data = num,main="temp vs Severity from 1 to 4"
```



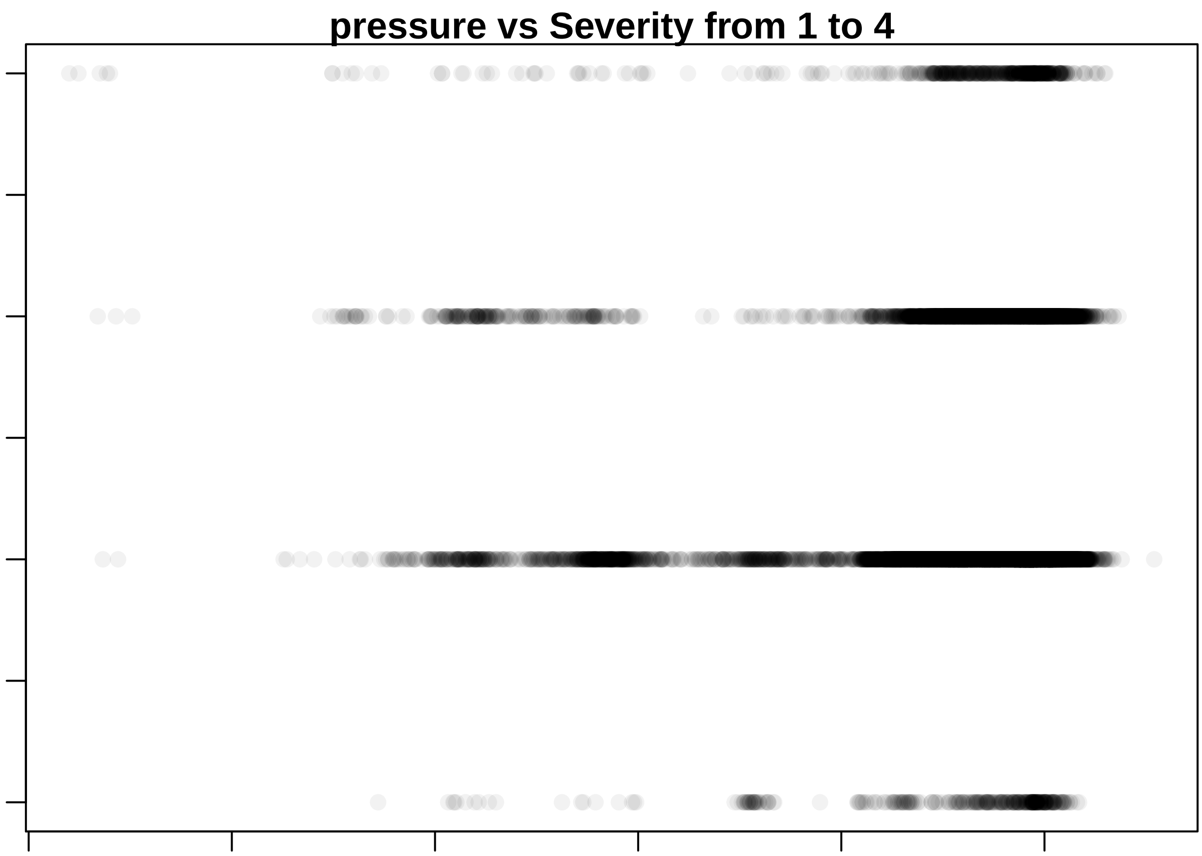
```
plot(Severity ~ windchill, col = rgb(0, 0, 0, 0.05), pch = 19,data = num,main="windchill vs Severity fr
```



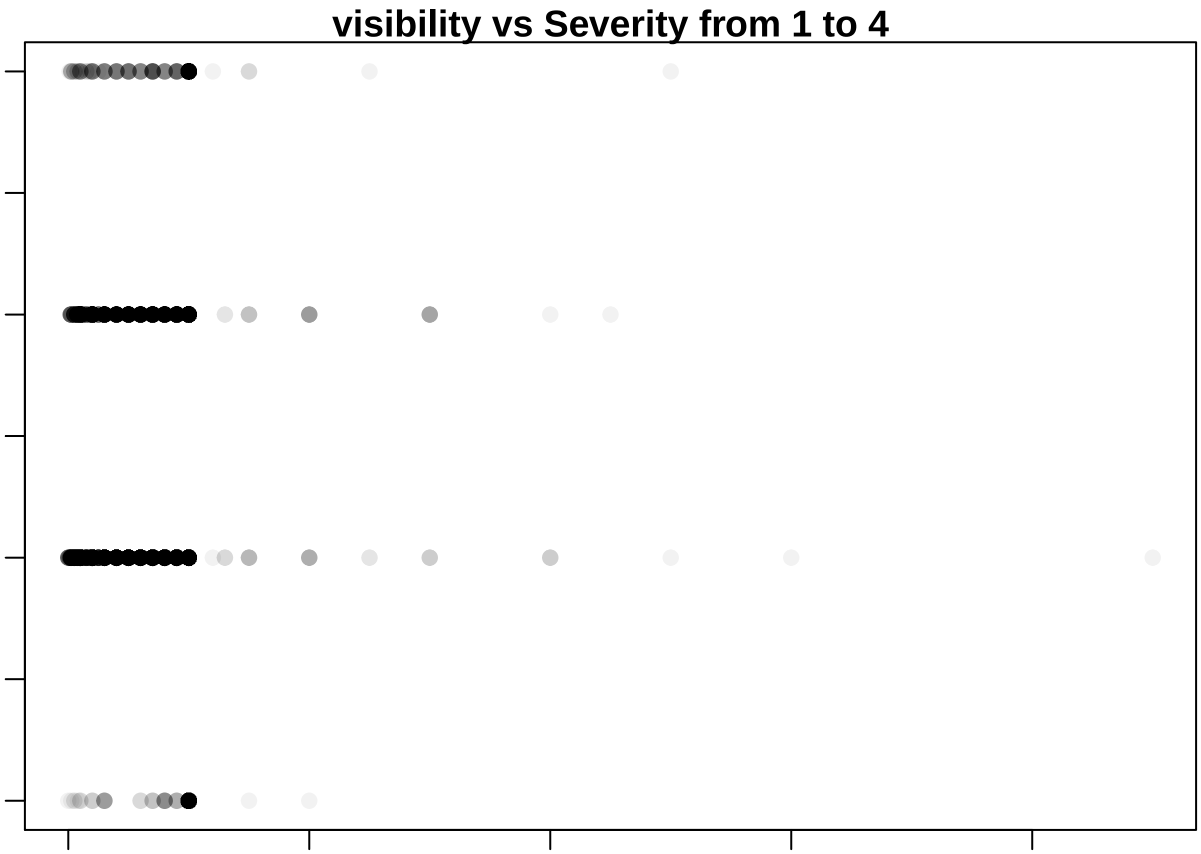
```
plot(Severity ~ humidity, col = rgb(0, 0, 0, 0.05), pch = 19,data = num,main="humidity vs Severity from
```



```
plot(Severity ~ pressure, col = rgb(0, 0, 0, 0.05), pch = 19,data = num,main="pressure vs Severity from
```

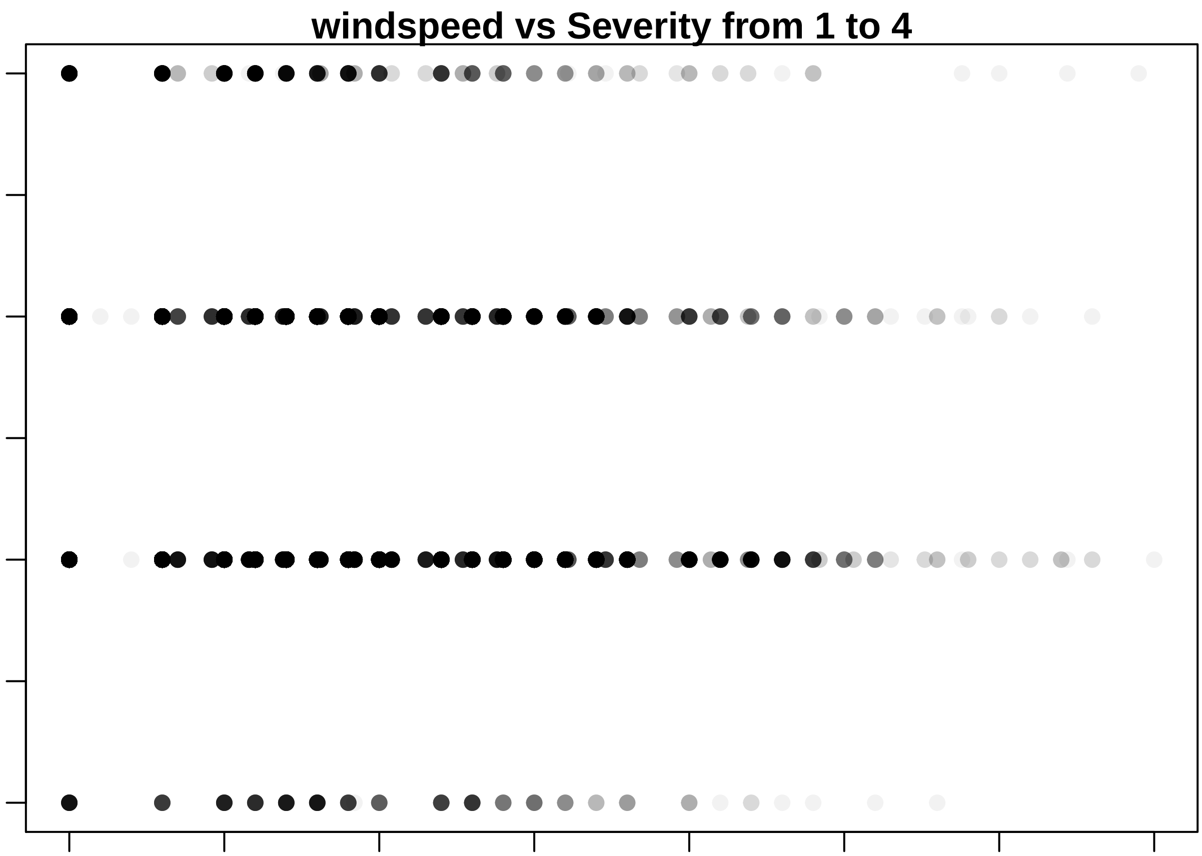


```
plot(Severity ~ visibility, col = rgb(0, 0, 0, 0.05), pch = 19,data = num,main="visibility vs Severity
```

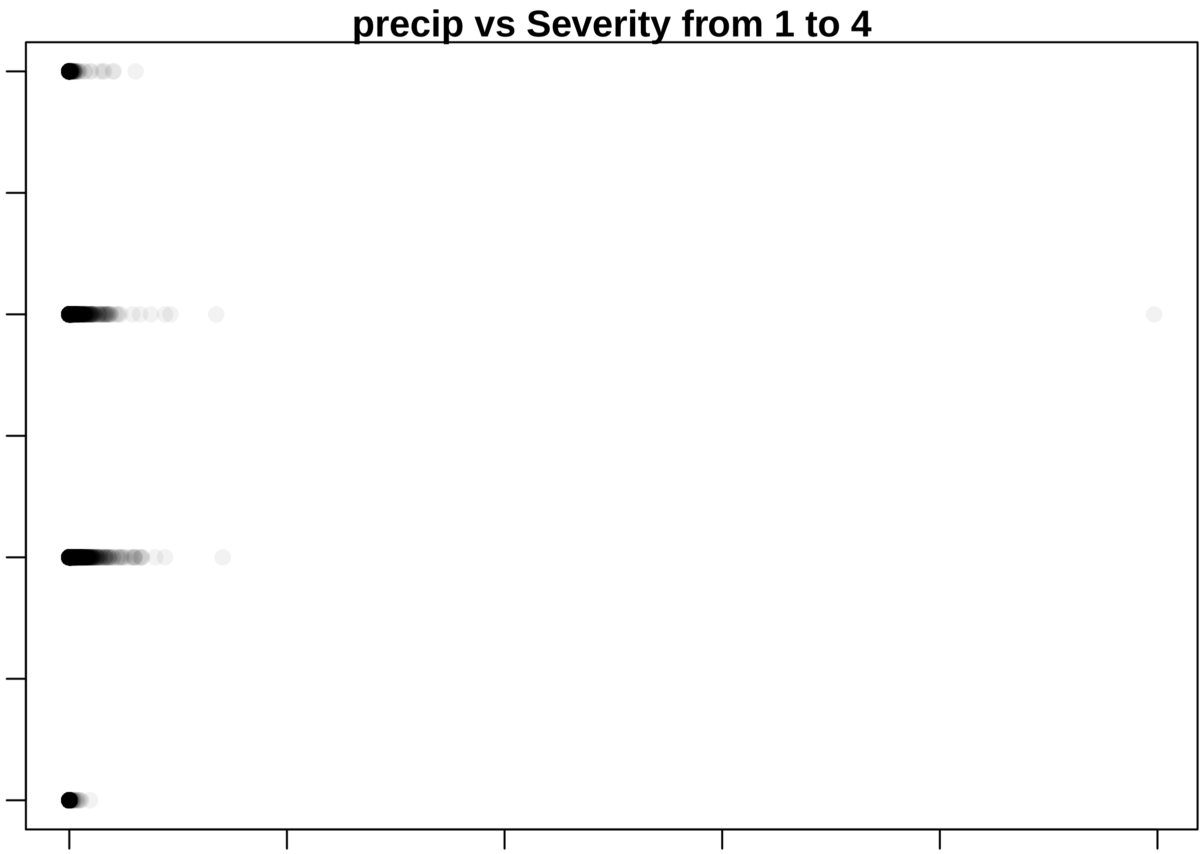


visibility vs Severity from 1 to 4

```
plot(Severity ~ windspeed, col = rgb(0, 0, 0, 0.05), pch = 19,data = num,main="windspeed vs Severity fr
```



```
plot(Severity ~ precip, col = rgb(0, 0, 0, 0.05), pch = 19,data = num,main="precip vs Severity from 1 t
```



```r
# Summarizing the model
summary(model)
```

```
## Call:
## nnet::multinom(formula = Severity ~ ., data = num1)
##
## Coefficients:
##   (Intercept)       dist     temp windchill   humidity  pressure visibility
## 2   -7.548126 -1.3921219 3.680148 -3.639778 0.04310664 0.1000599  0.5195707
## 3  -11.531331 -0.9592218 3.697775 -3.661394 0.04537197 0.2030625  0.5266955
## 4  -21.984556 -0.1579770 3.706935 -3.673514 0.04573332 0.4720234  0.6041697
##    windspeed   precip
## 2 -0.1765484 64.66716
## 3 -0.1676951 61.38921
## 4 -0.3802370 65.69752
##
## Std. Errors:
##   (Intercept)      dist       temp  windchill   humidity  pressure visibility
## 2  1.65793903 0.4236117 0.08833962 0.08702884 0.02216777 0.1171091  0.1767346
## 3  1.56734206 0.4078683 0.09033302 0.08790304 0.02250307 0.1172062  0.1776503
## 4  0.01411302 0.3473918 0.17542151 0.15531441 0.02918690 0.1426612  0.1876223
##    windspeed      precip
## 2 0.05275536 2.919630905
## 3 0.05467885 2.905830333
## 4 0.13637428 0.009116952
##
## Residual Deviance: 580.357
## AIC: 634.357
```

```r
# the above coefficients can now be used to infer how the indivisual variables contribution

# Cannot plot roc curve here since predicted values are not binary

# Making predictions
predicted.classes<-model %>% predict(num2)
head(predicted.classes)
```

```
## [1] 2 2 2 2 2 2
## Levels: 1 2 3 4
```

```r
#Checking model accuracy
mean(predicted.classes==num2$Severity)
```

```
## [1] 0.6732673
```

```r
# we observe that this model predictes the severity with a 67.32 % accuracy


#################### SECOND APPROACH ####################
library("foreign")
library("reshape2")
library(ggplot2)

# performing multinom logistic regression using multinom function
test<-multinom(Severity~.,data=num1)
```

```
## # weights:  40 (27 variable)
## initial  value 554.517744
## iter  10 value 363.079681
## iter  20 value 316.027015
## iter  30 value 293.716007
## iter  40 value 292.639663
## iter  50 value 291.528928
## iter  60 value 291.448698
## iter  70 value 291.443695
## iter  80 value 291.313810
## iter  90 value 291.001479
## iter 100 value 290.178494
## final  value 290.178494
## stopped after 100 iterations
```

```r
# printing model summary
summary(test)
```

```
## Call:
## multinom(formula = Severity ~ ., data = num1)
##
## Coefficients:
##   (Intercept)       dist     temp windchill   humidity  pressure visibility
## 2   -7.548126 -1.3921219 3.680148 -3.639778 0.04310664 0.1000599  0.5195707
## 3  -11.531331 -0.9592218 3.697775 -3.661394 0.04537197 0.2030625  0.5266955
## 4  -21.984556 -0.1579770 3.706935 -3.673514 0.04573332 0.4720234  0.6041697
##    windspeed   precip
## 2 -0.1765484 64.66716
## 3 -0.1676951 61.38921
## 4 -0.3802370 65.69752
##
## Std. Errors:
##   (Intercept)      dist       temp  windchill   humidity  pressure visibility
## 2  1.65793903 0.4236117 0.08833962 0.08702884 0.02216777 0.1171091  0.1767346
## 3  1.56734206 0.4078683 0.09033302 0.08790304 0.02250307 0.1172062  0.1776503
## 4  0.01411302 0.3473918 0.17542151 0.15531441 0.02918690 0.1426612  0.1876223
##    windspeed      precip
## 2 0.05275536 2.919630905
## 3 0.05467885 2.905830333
## 4 0.13637428 0.009116952
##
## Residual Deviance: 580.357
## AIC: 634.357
```

```
z<- summary(test)$coefficients/summary(test)$standard.errors

# performing two tailed z test
p<-(1-pnorm(abs(z),0,1))*2
p
```

```
##    (Intercept)       dist temp windchill   humidity     pressure  visibility
## 2 5.295773e-06 0.00101507    0         0 0.05182757 0.3928746375 0.003283860
## 3 1.876277e-13 0.01868317    0         0 0.04377307 0.0831804824 0.003028909
## 4 0.000000e+00 0.64928799    0         0 0.11713510 0.0009372966 0.001281290
##      windspeed precip
## 2 0.0008182447      0
## 3 0.0021628474      0
## 4 0.0053003912      0
```

```
## extract the coefficients from the model and exponentiate
exp(coef(test))
```

```
##    (Intercept)      dist     temp  windchill humidity pressure visibility
## 2 5.270972e-04 0.2485473 39.65228 0.02625817 1.044049 1.105237   1.681306
## 3 9.817623e-06 0.3831910 40.35741 0.02569668 1.046417 1.225149   1.693327
## 4 2.832882e-10 0.8538695 40.72879 0.02538710 1.046795 1.603235   1.829732
##   windspeed       precip
## 2 0.8381582 1.215038e+28
## 3 0.8456117 4.581351e+26
## 4 0.6836993 3.404630e+28
```

```
# calculating predicted probabilities for outcome levels using the fitted function
head(pp<-fitted(test))
```

```
##              1         2         3           4
## 1 1.192976e-02 0.7168351 0.2685252 0.002709911
## 2 7.277186e-03 0.7486696 0.2374313 0.006621870
## 3 2.342787e-03 0.7188632 0.2670833 0.011710766
## 4 8.587124e-04 0.7360087 0.2522244 0.010908184
## 5 2.850881e-02 0.6764755 0.2915150 0.003500643
## 6 1.197328e-12 0.6697014 0.3265845 0.003714080
```