BST Project

Dustin Lo

Sunday, May 10, 2015

```r
setwd("C:/Users/Dustin K. Lo/Desktop/BST test")
d <- read.csv("dataCSV.csv", header = T, stringsAsFactors = F)
d[d == "#NULL!"] <- NA
d <- na.omit(d)

for(i in c(63,65,66,68:71)) { #changes columns into numeric
  d[,i] <- as.numeric(d[,i])
}

d <- d[, -2]

for(i in c(4,5,10,11,12, 13:17)) { #changes columns into factor
  d[,i] <- factor(d[,i])
}
```

Order table by Project ID

```r
dav <- d
dav <- dav[-c(1:nrow(dav)), ]

n <- sort(unique(d$ProjectID))
for(i in 1:length(sort(unique(d$ProjectID)))) {
  sub <- d[which(d$ProjectID == n[i]), ]
  dav <- rbind(dav, sub)
}
# write.csv(dav, "bst.csv")
```

```r
########################################################
dav <- dav[which(dav$Industry_Groups == "Chemicals"), ]
#########################################################

#subsetting the dataset 75/25
#the 75% is for modeling and the 25% is for predictive analysis
keep <- function(x, seed) {
  set.seed(seed)
  k <- sort(sample(1:nrow(x), size = round(nrow(x)*.80), replace = F, prob = NULL))
  return(k)
}
in.index<- keep(dav, 10261991)
din <- dav[in.index,]
dout <- dav[-in.index,]
```

```r
#choosing stronger variables with our injury binary variable
cors <- cor(din[, sapply(din, is.numeric)], method = "pearson")
cors <- cors[-c(51,53:56), -c(51,53:56)]
strong <- which(abs(cors[,51]) > 0.05)
```

```
cors <- cors[strong, strong]

#ProcedureHotRiskRate_perFTE is bad!!!
# subsetting the table to match our variables chosen in our correlation table
use <- din[, match(row.names(cors), colnames(din))]
use <- use[, -c(2,3,29,30)]
use$ProcedureHotRiskRate_perFTE <- NULL
#running a linear model in all variables chosen from correlation table
m <- glm(InjuryYN_Lag1 ~ . , data = use, family = "binomial")
#summary(m)
```

```
#library(MASS)
#stepAIC(m, direction = "both", k = 10)
```

```
#sqrt transform SumRisks
m1 <- glm(formula = InjuryYN_Lag1 ~ ObsRate_perFTE + sqrt(SumRisks) + ContractorEERate_perObs,
          family = "binomial", data = use)
summary(m1)
```

```
##
## Call:
## glm(formula = InjuryYN_Lag1 ~ ObsRate_perFTE + sqrt(SumRisks) +
##     ContractorEERate_perObs, family = "binomial", data = use)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6309  -0.8238  -0.5972   1.0535   3.1344
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.37263    0.23327  -5.884 3.99e-09 ***
## ObsRate_perFTE          -0.90380    0.24723  -3.656 0.000257 ***
## sqrt(SumRisks)           0.15696    0.03031   5.178 2.25e-07 ***
## ContractorEERate_perObs -1.77073    0.60728  -2.916 0.003547 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 692.63  on 597  degrees of freedom
## Residual deviance: 632.25  on 594  degrees of freedom
## AIC: 640.25
##
## Number of Fisher Scoring iterations: 5
```

```
#Ho = model is a good fit for data
#Ha = model is bad fit for data
pchisq(632.25,  594)
```

```
## [1] 0.8656417
```

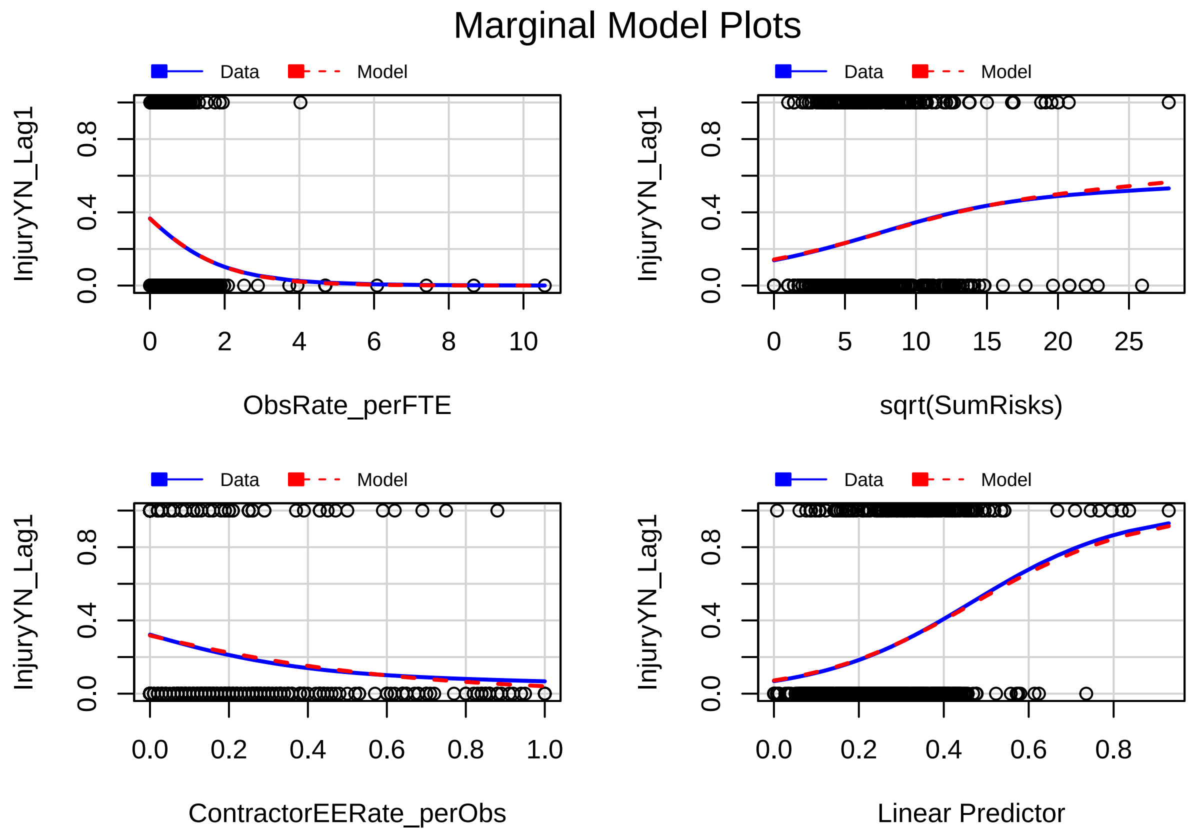
```r
# 0.865 so we reject the null

#checking for the fit of our model
library(alr3)
```

```
## Warning: package 'alr3' was built under R version 3.1.3
```

```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 3.1.3
```

```r
mmps(m1)
```



```r
#good fit
```

```r
#testing the numerical model out
attach(din)
input1 <- data.frame(SumRisks = rep(round(mean(SumRisks)), 50), ObsRate_perFTE = seq(from = 0, to = 3, l
                    ContractorEERate_perObs = rep(0.1327, 50))
input1 <- cbind(input1, Prob = predict(m1, input1, type = "response", se = TRUE))
input1$Prob.residual.scale <- NULL
detach(din)
upper <- round(input1$Prob.fit + 1.96 * input1$Prob.se.fit, 4)
```

```r
lower <- round(input1$Prob.fit - 1.96 * input1$Prob.se.fit, 4)
output1 <- cbind(input1, lower, upper)

exp(coef(m1))
```

```
##             (Intercept)          ObsRate_perFTE          sqrt(SumRisks)
##               0.2534387               0.4050276               1.1699455
## ContractorEERate_perObs
##               0.1702086
```

```r
# Change in Odds per change in input1 variables
#(Intercept)          ObsRate_perFTE          sqrt(SumRisks) ContractorEERate_perObs
#0.2534387               0.4050276               1.1699455               0.1702086

confint(m1)
```

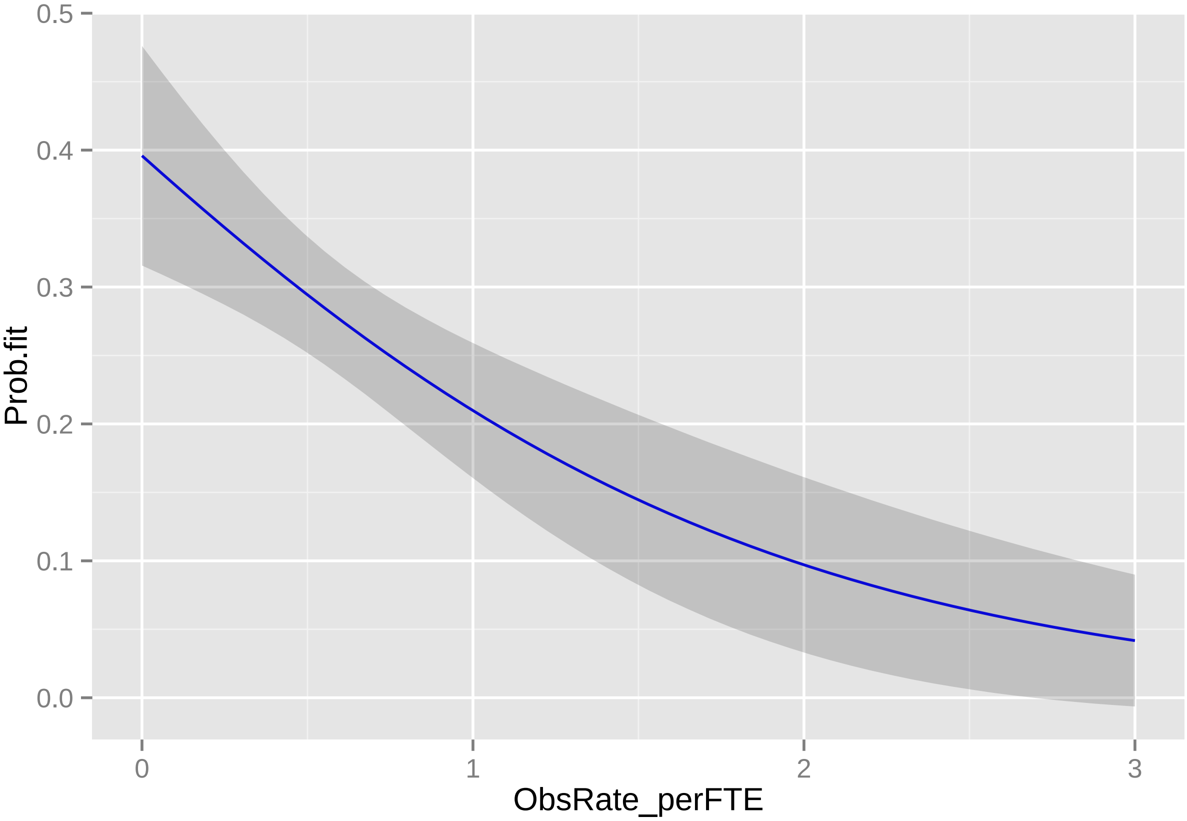
```
## Waiting for profiling to be done...
```

```
##                               2.5 %     97.5 %
## (Intercept)             -1.83284758 -0.9179411
## ObsRate_perFTE          -1.40852093 -0.4402804
## sqrt(SumRisks)           0.09888288  0.2179332
## ContractorEERate_perObs -3.03623340 -0.6422850
```

```r
#                            2.5 %     97.5 %
#  (Intercept)           -1.83284758 -0.9179411
#ObsRate_perFTE          -1.40852093 -0.4402804
#sqrt(SumRisks)           0.09888288  0.2179332
#ContractorEERate_perObs -3.03623340 -0.6422850

head(output1[,-6])
```

```
##   SumRisks ObsRate_perFTE ContractorEERate_perObs  Prob.fit Prob.se.fit
## 1       57     0.00000000                  0.1327 0.3958926  0.04090228
## 2       57     0.06122449                  0.1327 0.3827378  0.03762001
## 3       57     0.12244898                  0.1327 0.3697526  0.03450417
## 4       57     0.18367347                  0.1327 0.3569532  0.03159561
## 5       57     0.24489796                  0.1327 0.3443547  0.02893949
## 6       57     0.30612245                  0.1327 0.3319713  0.02658475
##    upper
## 1 0.4761
## 2 0.4565
## 3 0.4374
## 4 0.4189
## 5 0.4011
## 6 0.3841
```

```r
library(ggplot2)
ggplot(output1, aes(x = ObsRate_perFTE, y = Prob.fit)) + geom_line(col = "blue") +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2)
```



```
with(m1, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
```

```
## [1] 4.865744e-13
```

```
# 4.865744e-13
# very low p-value, shows that our model fits better than an empty model
```

```
# checking for interaction terms
# InjuryYN_Lag1   ObsRate_perFTE   sqrt(SumRisks)   ContractorEERate_perObs
dcat <- din[, !sapply(din, is.numeric)]
dcat <- dcat[, -c(3,4,5,14)]
dcat <- dcat[, -c(10,9,8,2,3,4)]
str(dcat)
```

```
## 'data.frame':    598 obs. of  4 variables:
##  $ CompanyGroup              : chr  "4" "4" "4" "4" ...
##  $ RD_WorldRegionGroups      : Factor w/ 3 levels "Americas","Asia Pacific",..: 3 3 3 3 3 3 3 3 3 1
##  $ Employee_TypeContractor_YN: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
##  $ CoachedObs_YN             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
```

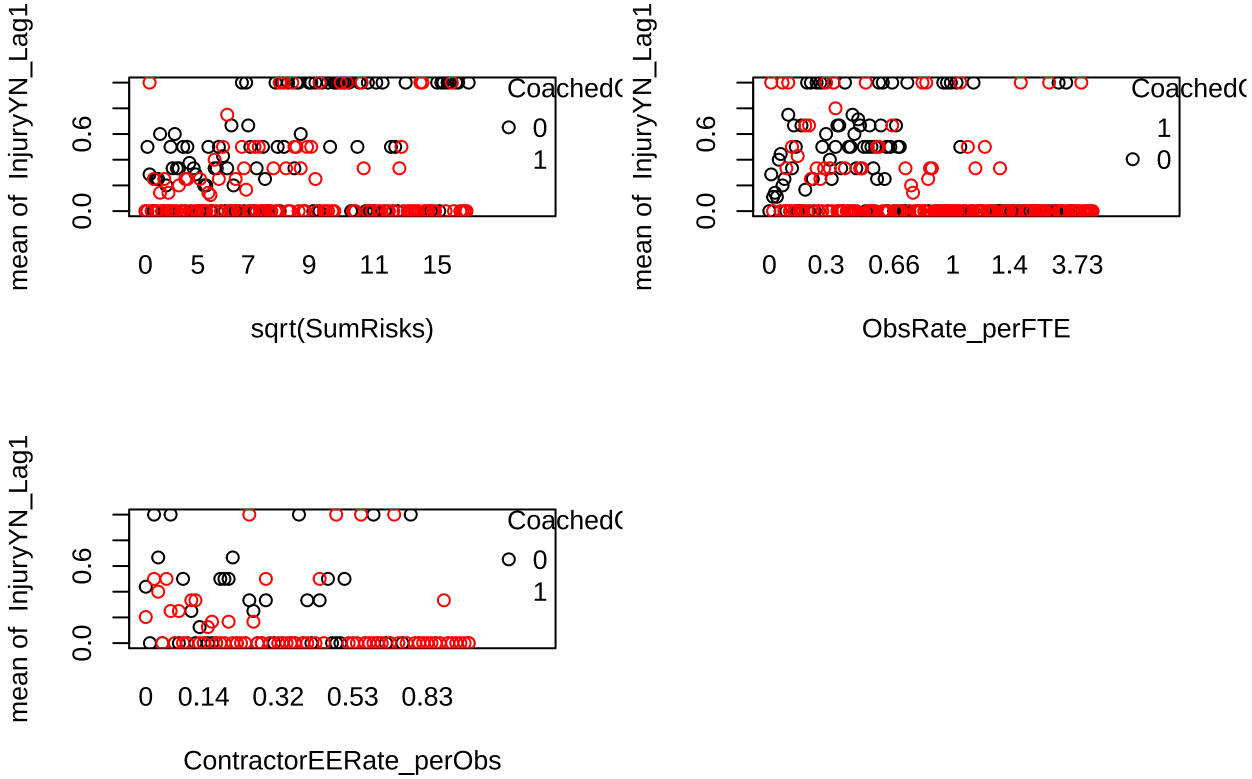
```
attach(din)
table(InjuryYN_Lag1, CoachedObs_YN)
```

```
##              CoachedObs_YN
## InjuryYN_Lag1   0   1
##             0 193 246
##             1 109  50
```

```
table(InjuryYN_Lag1,Employee_TypeContractor_YN)
```

```
##              Employee_TypeContractor_YN
## InjuryYN_Lag1   0   1
##             0 220 219
##             1 114  45
```

```
# CoachedObs_YN looks like a better factor variable than Employee_TypeContractor_YN
par(mfrow = c(2,2))
interaction.plot(sqrt(SumRisks), CoachedObs_YN, InjuryYN_Lag1, type = "p",
                 pch = 1, col = c(1,2))
#some interaction between sqrt(SumRisks), CoachedObs_YN
interaction.plot(ObsRate_perFTE, CoachedObs_YN, InjuryYN_Lag1, type = "p",
                 pch = 1, col = c(1,2))
#some interaction between ObsRate_perFTE, CoachedObs_YN
interaction.plot(ContractorEERate_perObs, CoachedObs_YN, InjuryYN_Lag1, type = "p",
                 pch = 1, col = c(1,2))
# little to no interaction
par(mfrow = c(1,1))
```



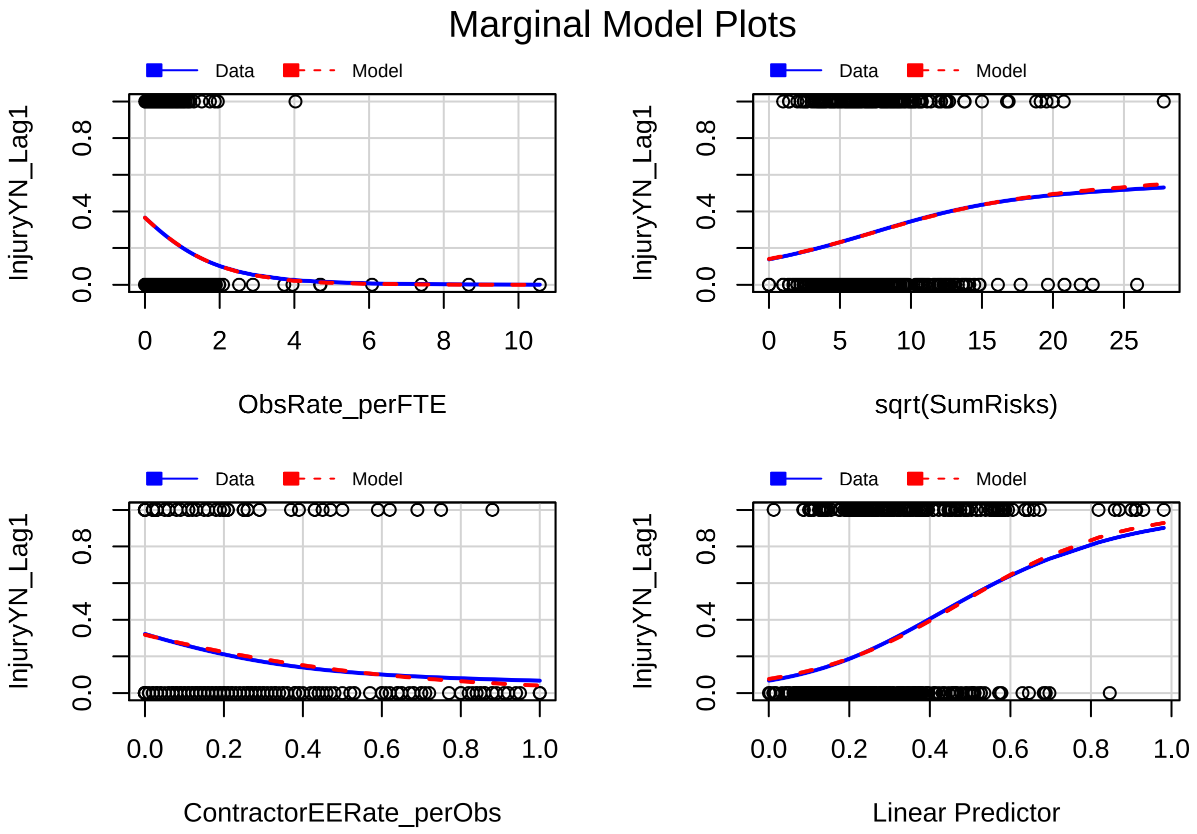
```r
detach(din)
```

```r
#our final model
#modeling with interaction terms
final <- glm(formula = InjuryYN_Lag1 ~ ObsRate_perFTE + sqrt(SumRisks) +
               ContractorEERate_perObs + sqrt(SumRisks):CoachedObs_YN, family = "binomial", data = din)
summary(final)
```

```
##
## Call:
## glm(formula = InjuryYN_Lag1 ~ ObsRate_perFTE + sqrt(SumRisks) +
##     ContractorEERate_perObs + sqrt(SumRisks):CoachedObs_YN, family = "binomial",
##     data = din)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9369  -0.7746  -0.5865   0.9084   2.9664
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.53136    0.24809  -6.173 6.72e-10 ***
## ObsRate_perFTE                -0.63938    0.24757  -2.583 0.009805 **
## sqrt(SumRisks)                 0.20694    0.03516   5.886 3.97e-09 ***
## ContractorEERate_perObs       -1.48961    0.61234  -2.433 0.014990 *
## sqrt(SumRisks):CoachedObs_YN1 -0.10991    0.02852  -3.853 0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 692.63  on 597  degrees of freedom
## Residual deviance: 616.85  on 593  degrees of freedom
## AIC: 626.85
##
## Number of Fisher Scoring iterations: 5
```

```r
mmps(final)
```

```
## Warning in mmps(final): Interactions and/or factors skipped
```



```
#Ho: model is good fit for data
#Ha: model is bad fit for data
pchisq(616.85,  593)
```

```
## [1] 0.7588989
```

```
# 0.758898, reject null, so good fit

with(final, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
```

```
## [1] 1.36039e-15
```

```
#low p-value, so our model is better than an empty model
```

testing our final model out with ObsRate_perFTE

```
attach(din)
input2 <- data.frame(SumRisks = rep(round(mean(SumRisks)), 100),
                     ObsRate_perFTE = rep(seq(from = 0, to = 3, length.out = 50), times = 2),
                     ContractorEERate_perObs = rep(0.1327, 100),
                     CoachedObs_YN = factor(rep(c(0,1), times = 1, each = 50)))
input2 <- cbind(input2, Prob = predict(final, input2, type = "response", se = TRUE))
input2$Prob.residual.scale <- NULL
detach(din)
```

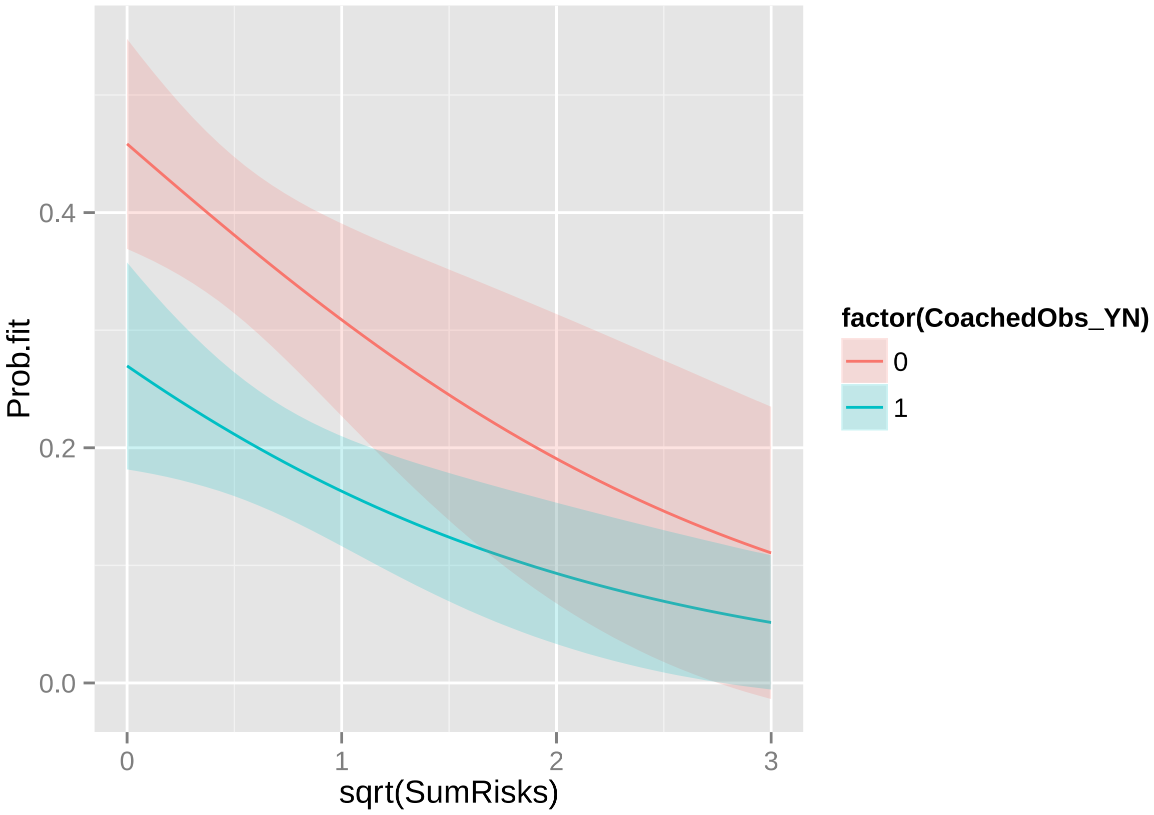
```r
upper <- round(input2$Prob.fit + 1.96 * input2$Prob.se.fit, 4)
lower <- round(input2$Prob.fit - 1.96 * input2$Prob.se.fit, 4)
output2 <- cbind(input2, lower, upper)

head(output2[1:50,-6])
```

```
##   SumRisks ObsRate_perFTE ContractorEERate_perObs CoachedObs_YN  Prob.fit
## 1       57     0.00000000                  0.1327             0 0.4584333
## 2       57     0.06122449                  0.1327             0 0.4487315
## 3       57     0.12244898                  0.1327             0 0.4390685
## 4       57     0.18367347                  0.1327             0 0.4294515
## 5       57     0.24489796                  0.1327             0 0.4198875
## 6       57     0.30612245                  0.1327             0 0.4103832
##    lower  upper
## 1 0.3691 0.5477
## 2 0.3641 0.5333
## 3 0.3588 0.5193
## 4 0.3530 0.5059
## 5 0.3467 0.4930
## 6 0.3399 0.4808
```

```r
head(output2[50:100,-6])
```

```
##    SumRisks ObsRate_perFTE ContractorEERate_perObs CoachedObs_YN  Prob.fit
## 50       57     3.00000000                  0.1327             0 0.1105826
## 51       57     0.00000000                  0.1327             1 0.2696361
## 52       57     0.06122449                  0.1327             1 0.2619968
## 53       57     0.12244898                  0.1327             1 0.2544986
## 54       57     0.18367347                  0.1327             1 0.2471431
## 55       57     0.24489796                  0.1327             1 0.2399317
##      lower  upper
## 50 -0.0137 0.2348
## 51  0.1816 0.3576
## 52  0.1797 0.3443
## 53  0.1776 0.3314
## 54  0.1753 0.3190
## 55  0.1727 0.3072
```

```r
ggplot(output2, aes(x = ObsRate_perFTE, y = Prob.fit)) + geom_line(aes(color = factor(CoachedObs_YN)))
  geom_ribbon(aes(fill = factor(CoachedObs_YN), ymin = lower, ymax = upper), alpha = 0.2) +
  labs(x = "sqrt(SumRisks)")
```



testing our final model out with SumRisks

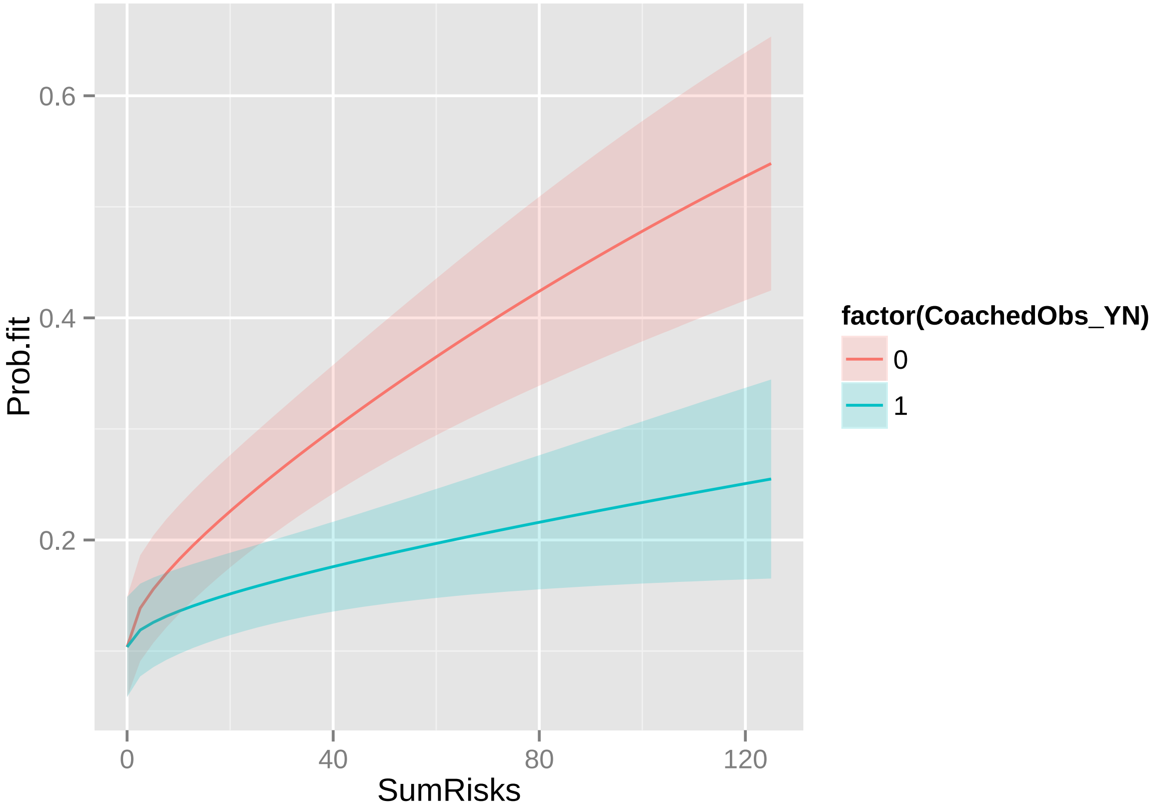
```r
attach(din)
input3 <- data.frame(SumRisks = rep(seq(from = 0, to = 125, length.out = 50), times = 2),
                     ObsRate_perFTE = rep(mean(ObsRate_perFTE), 100),
                     ContractorEERate_perObs = rep(0.1327, 100),
                     CoachedObs_YN = factor(rep(c(0,1), times = 1, each = 50)))
input3 <- cbind(input3, Prob = predict(final, input3, type = "response", se = TRUE))
input3$Prob.residual.scale <- NULL
detach(din)
upper <- round(input3$Prob.fit + 1.96 * input3$Prob.se.fit, 4)
lower <- round(input3$Prob.fit - 1.96 * input3$Prob.se.fit, 4)
output3 <- cbind(input3, lower, upper)

head(output3[1:50,-6])
```

```
##    SumRisks ObsRate_perFTE ContractorEERate_perObs CoachedObs_YN  Prob.fit
## 1  0.000000      0.6698495                  0.1327             0 0.1036488
## 2  2.551020      0.6698495                  0.1327             0 0.1386202
## 3  5.102041      0.6698495                  0.1327             0 0.1557905
## 4  7.653061      0.6698495                  0.1327             0 0.1701118
## 5 10.204082      0.6698495                  0.1327             0 0.1829824
## 6 12.755102      0.6698495                  0.1327             0 0.1949370
##    lower  upper
## 1 0.0585 0.1488
## 2 0.0909 0.1863
## 3 0.1074 0.2041
## 4 0.1214 0.2189
## 5 0.1339 0.2320
## 6 0.1456 0.2443
```

```
head(output3[50:100,-6])
```

```
##      SumRisks ObsRate_perFTE ContractorEERate_perObs CoachedObs_YN
## 50 125.000000      0.6698495                  0.1327             0
## 51   0.000000      0.6698495                  0.1327             1
## 52   2.551020      0.6698495                  0.1327             1
## 53   5.102041      0.6698495                  0.1327             1
## 54   7.653061      0.6698495                  0.1327             1
## 55  10.204082      0.6698495                  0.1327             1
##     Prob.fit  lower  upper
## 50 0.5390114 0.4247 0.6533
## 51 0.1036488 0.0585 0.1488
## 52 0.1189566 0.0772 0.1607
## 53 0.1258507 0.0855 0.1662
## 54 0.1313702 0.0921 0.1706
## 55 0.1361817 0.0977 0.1746
```

```
ggplot(output3, aes(x = SumRisks, y = Prob.fit)) + geom_line(aes(color = factor(CoachedObs_YN))) +
  geom_ribbon(aes(fill = factor(CoachedObs_YN), ymin = lower, ymax = upper), alpha = 0.2)
```



testing our final model out with ContractorEERate_perObs

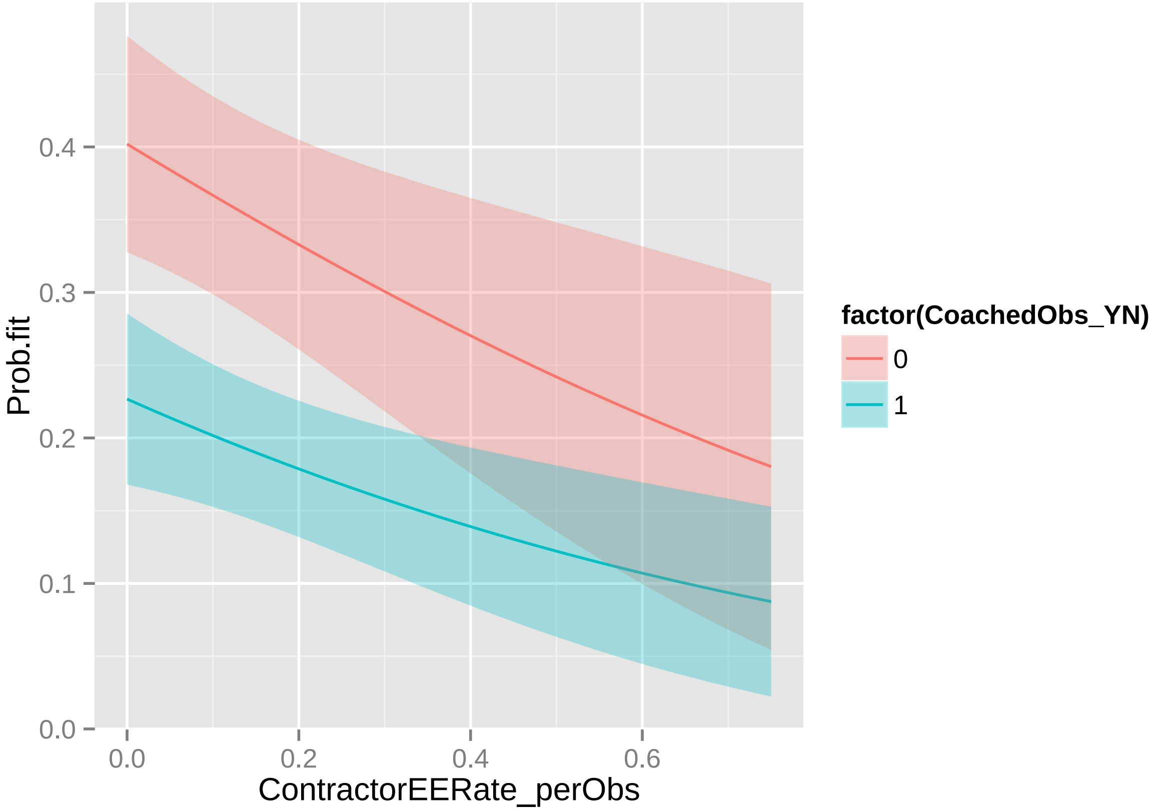
```
attach(din)
input4 <- data.frame(SumRisks = rep(round(mean(SumRisks)), 100),
                     ObsRate_perFTE = rep(mean(ObsRate_perFTE), 100),
                     ContractorEERate_perObs = rep(seq(from = 0, to = .75, length.out = 50), times = 2)
                     CoachedObs_YN = factor(rep(c(0,1), times = 1, each = 50)))
input4 <- cbind(input4, Prob = predict(final, input4, type = "response", se = TRUE))
input4$Prob.residual.scale <- NULL
detach(din)
upper <- round(input4$Prob.fit + 1.96 * input4$Prob.se.fit, 4)
lower <- round(input4$Prob.fit - 1.96 * input4$Prob.se.fit, 4)
output4 <- cbind(input4, lower, upper)
head(output4[1:50,-6])
```

```
##   SumRisks ObsRate_perFTE ContractorEERate_perObs CoachedObs_YN  Prob.fit
## 1       57      0.6698495              0.00000000             0 0.4019680
## 2       57      0.6698495              0.01530612             0 0.3964996
## 3       57      0.6698495              0.03061224             0 0.3910569
## 4       57      0.6698495              0.04591837             0 0.3856411
## 5       57      0.6698495              0.06122449             0 0.3802536
## 6       57      0.6698495              0.07653061             0 0.3748953
##    lower  upper
## 1 0.3276 0.4763
## 2 0.3238 0.4692
## 3 0.3198 0.4624
## 4 0.3155 0.4558
## 5 0.3110 0.4495
## 6 0.3063 0.4435
```

```
head(output4[50:100,-6])
```

```
##    SumRisks ObsRate_perFTE ContractorEERate_perObs CoachedObs_YN  Prob.fit
## 50       57      0.6698495              0.75000000             0 0.1802761
## 51       57      0.6698495              0.00000000             1 0.2266911
## 52       57      0.6698495              0.01530612             1 0.2227192
## 53       57      0.6698495              0.03061224             1 0.2187971
## 54       57      0.6698495              0.04591837             1 0.2149249
## 55       57      0.6698495              0.06122449             1 0.2111028
##     lower  upper
## 50 0.0543 0.3062
## 51 0.1679 0.2855
## 52 0.1659 0.2795
## 53 0.1638 0.2738
## 54 0.1616 0.2683
## 55 0.1592 0.2630
```

```
ggplot(output4, aes(x = ContractorEERate_perObs, y = Prob.fit)) + geom_line(aes(color = factor(CoachedO
  geom_ribbon(aes(fill = factor(CoachedObs_YN), ymin = lower, ymax = upper), alpha = 0.3)
```


Prob.fit
0.0
0.1
0.2
0.3
0.4
0.0
0.2
0.4
0.6
ContractorEERate_perObs
factor(CoachedObs_YN)
0
1