

Data_Cleaning

Chang Chen

11/14/2023

```
# Loading required packages
```

```
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(ggplot2)
```

```
library("vcdExtra")
```

```
## Loading required package: vcd
```

```
## Loading required package: grid
```

```
## Loading required package: gnm
```

```
##
```

```
## Attaching package: 'vcdExtra'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##   summarise
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##   as.Date, as.Date.numeric
```

```
library(MASS)
```

```
##
```

```
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##   select
```

```
library(pROC)
```

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

```
##
```

```
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      cov, smooth, var
```

```
library(patchwork)
```

```
##
```

```
## Attaching package: 'patchwork'
```

```
## The following object is masked from 'package:MASS':
```

```
##
```

```
##      area
```

```
library(lattice)
```

```
##
```

```
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:gnm':
```

```
##
```

```
##      barley
```

Step 1 and 2: Data Cleaning and Descriptive Statistics

```
# Load the CSV file
```

```
data_old <- read.csv("../Handout/final_cardiac_data.csv")
```

```
# Data Overview
```

```
head(data_old)
```

```
##      X   seqn event gender age ethnic1 educ sleep.hrs diabetes smoker  bmi
## 1 179 109441     0      2  20         3   4         9         2      2 18.0
## 2 323 109585     0      1  20         4   2         8         2      2 28.2
## 3 463 109726     0      1  20         4   4         5         2      2 26.2
## 4 546 109809     0      2  20         3   4         7         2      2 22.1
## 5 589 109852     0      2  20         2   2         6         2     NA 20.4
## 6 630 109893     0      2  20         4   3         8         2      1 30.3
```

```
# Remove identifier columns
```

```
data <- data_old[, !(names(data_old) %in% c("X", "seqn"))]
```

```
# Run summary statistics to get overview
```

```
summary(data)
```

```
##      event      gender      age      ethnic1
##  Min.   :0.0000  Min.   :1.000  Min.   :20.00  Min.   :1.000
## 1st Qu.:0.0000  1st Qu.:1.000  1st Qu.:36.00  1st Qu.:3.000
##  Median :0.0000  Median :2.000  Median :52.00  Median :3.000
##   Mean   :0.4246  Mean   :1.516  Mean   :50.96  Mean   :3.279
## 3rd Qu.:1.0000  3rd Qu.:2.000  3rd Qu.:64.00  3rd Qu.:4.000
##   Max.   :1.0000  Max.   :2.000  Max.   :80.00  Max.   :5.000
##
```

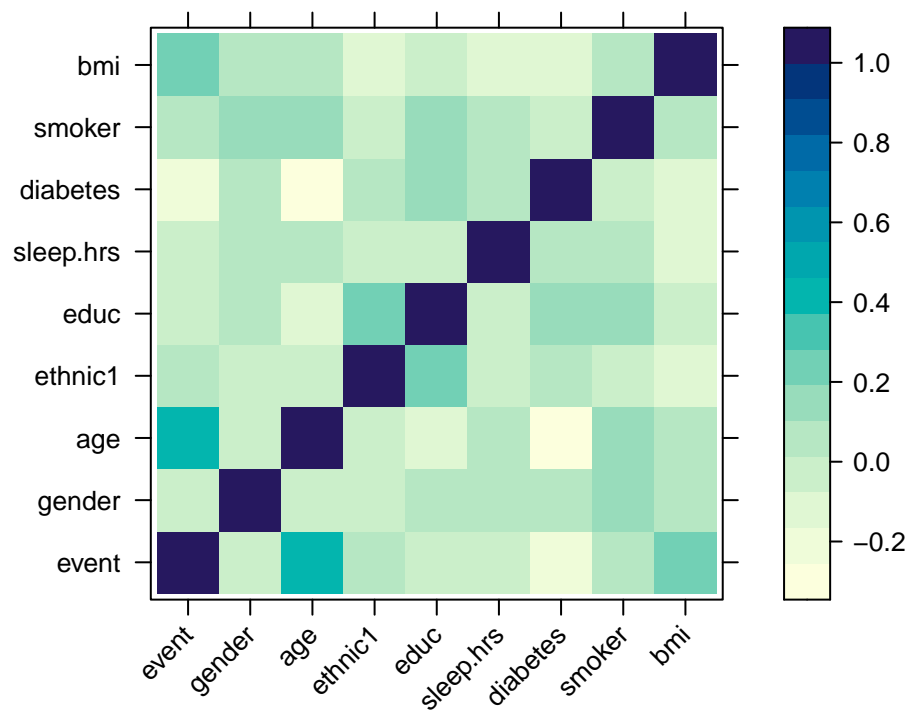
```
##      educ      sleep.hrs      diabetes      smoker
## Min.   :1.000   Min.    : 2.000   Min.    :1.000   Min.    :1.000
## 1st Qu.:3.000   1st Qu.: 6.500   1st Qu.:2.000   1st Qu.:2.000
## Median :4.000   Median : 7.500   Median :2.000   Median :2.000
## Mean   :3.557   Mean    : 7.551   Mean    :1.863   Mean    :1.763
## 3rd Qu.:4.000   3rd Qu.: 8.500   3rd Qu.:2.000   3rd Qu.:2.000
## Max.   :9.000   Max.    :14.000   Max.    :9.000   Max.    :2.000
##                NA's    :13      NA's    :72      NA's    :147
##      bmi
## Min.   :14.20
## 1st Qu.:24.98
## Median :29.10
## Mean   :30.48
## 3rd Qu.:34.50
## Max.   :86.20
## NA's   :46
```

```
# Find the correlation between each predictor
# Use it to check for collinearity when running regression models to predict missing values
```

```
cor_matrix <- cor(na.omit(data))
cor_matrix
```

```
##      event      gender      age      ethnic1      educ
## event      1.00000000 -0.03829681  0.45808113  0.05271390 -0.04336137
## gender     -0.03829681  1.00000000 -0.06239729  0.01084565  0.02054031
## age        0.45808113 -0.06239729  1.00000000 -0.02250016 -0.09388666
## ethnic1    0.05271390  0.01084565 -0.02250016  1.00000000  0.26144749
## educ       -0.04336137  0.02054031 -0.09388666  0.26144749  1.00000000
## sleep.hrs -0.01610149  0.06277700  0.05518124 -0.03185989 -0.02145965
## diabetes   -0.24450494  0.05232879 -0.25758949  0.04338853  0.12825910
## smoker     0.02144489  0.12497939  0.13249983 -0.06804697  0.13381708
## bmi        0.20399110  0.08309773  0.01496912 -0.07913569 -0.01939811
##      sleep.hrs      diabetes      smoker      bmi
## event     -0.01610149 -0.24450494  0.02144489  0.20399110
## gender     0.06277700  0.05232879  0.12497939  0.08309773
## age        0.05518124 -0.25758949  0.13249983  0.01496912
## ethnic1    -0.03185989  0.04338853 -0.06804697 -0.07913569
## educ       -0.02145965  0.12825910  0.13381708 -0.01939811
## sleep.hrs  1.00000000  0.03917174  0.05190982 -0.08127697
## diabetes   0.03917174  1.00000000 -0.02345722 -0.13925171
## smoker     0.05190982 -0.02345722  1.00000000  0.07292136
## bmi        -0.08127697 -0.13925171  0.07292136  1.00000000
```

```
# Use heat map to visualize correlation matrix
levelplot(cor_matrix,scale=list(x=list(rot=45)),xlab=NULL,ylab=NULL)
```



Event

```
table(data$event)
```

```
##
##      0      1
## 1099   811
```

```
data$event <- as.factor(data$event)
```

Gender

```
table(data$gender)
```

```
##
##      1      2
##  924   986
```

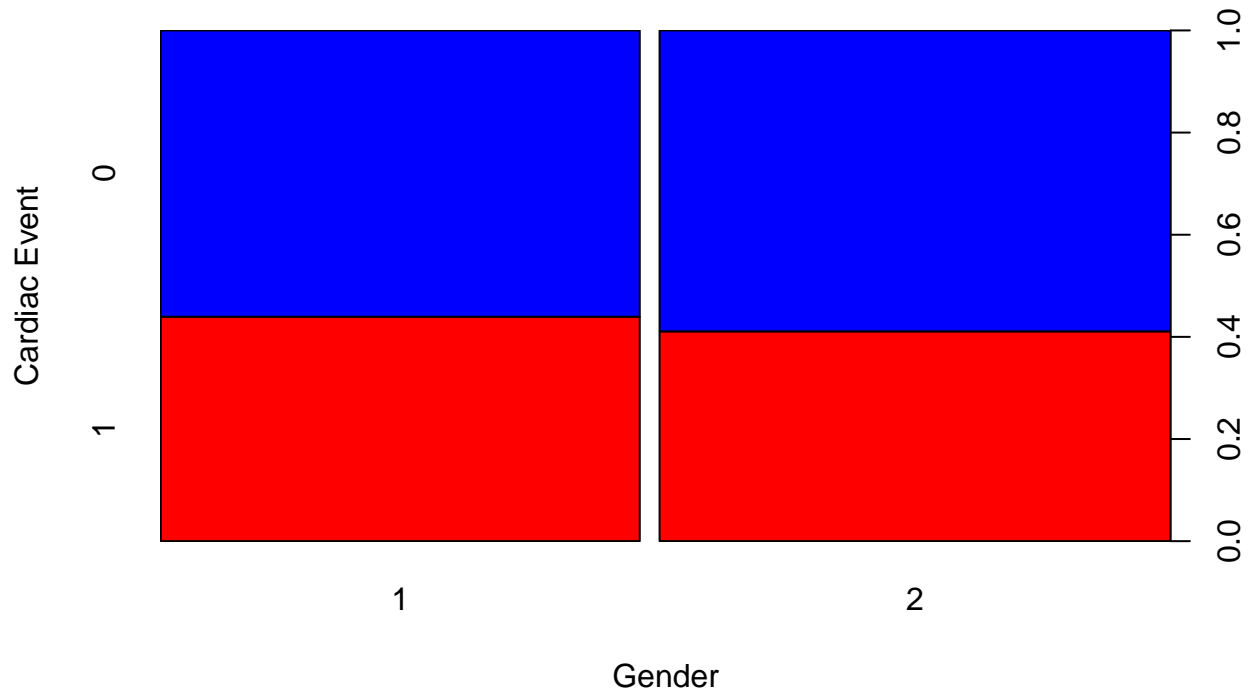
```
data$gender <- as.factor(data$gender)
```

```
gender_tab <- table(data$event, data$gender)
```

```
# Mosaic plot for gender
```

```
spineplot(t(gender_tab), main = "Gender and Cardiac Event",
          xlab="Gender", ylab="Cardiac Event", col=c("red", "blue"))
```

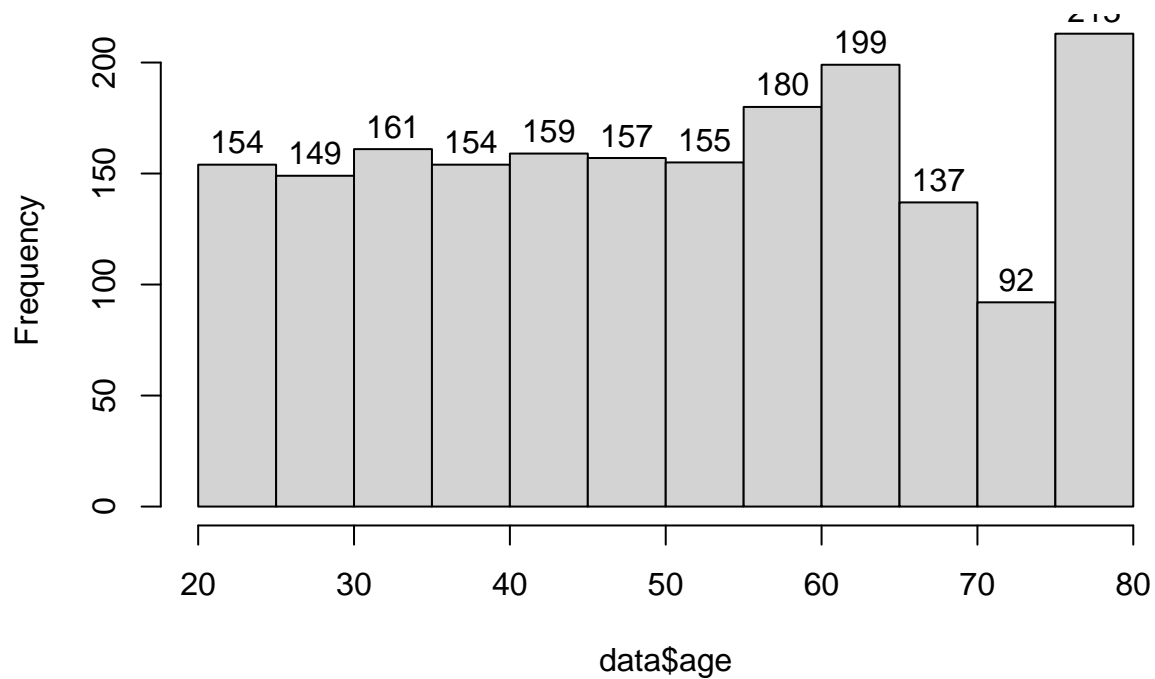
Gender and Cardiac Event



Age

```
hist(data$age, labels = T)
```

Histogram of data\$age



```
# Slicing plot for age
age.fac = factor(cut(data$age, breaks=15), labels=1:15)
```

```
table(age.fac)
```

```
## age.fac  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
## 121 112 137 122 126 125 130 110 140 146 171 107 98 72 193
```

```
# Empirical probs for each category
```

```
age.prob <- tapply(data$event, age.fac, mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:  
## returning NA
```

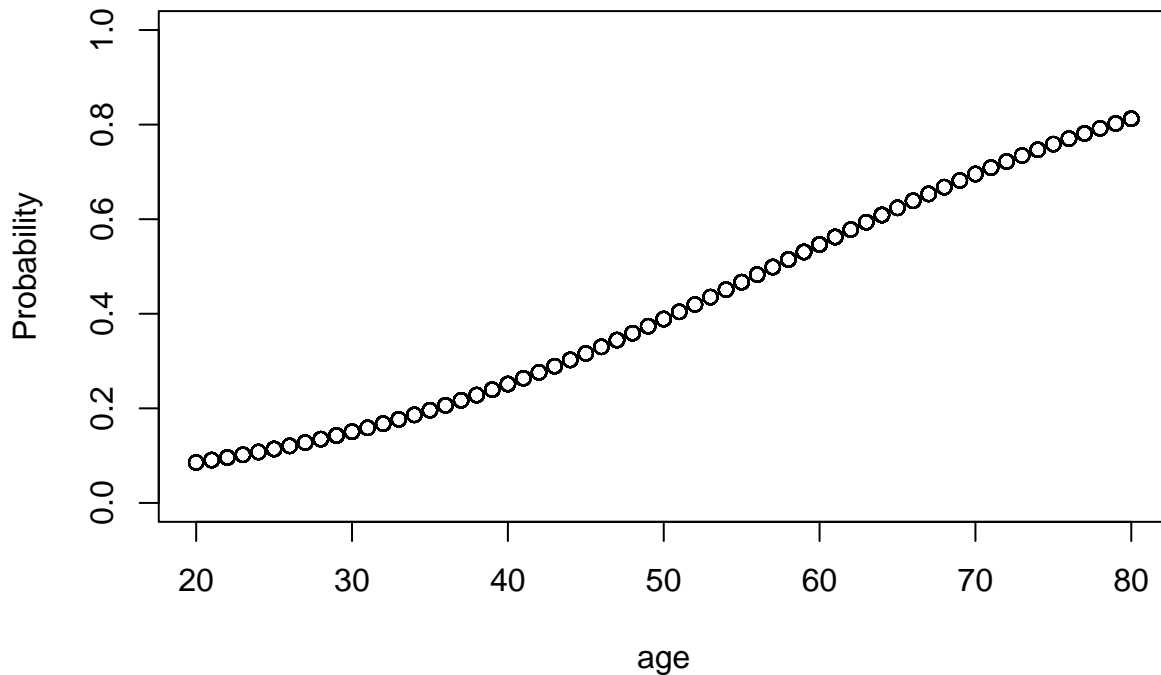
```

age.slice.avg <- tapply(data$age, age.fac, mean)
age.elogits <- log(age.prob/(1-age.prob))
# Run logistic regression on simulated data
age.out <- glm(event ~ age, data = data, family = 'binomial')

# Graph predicted and empirical probabilities
plot(data$age, age.out$fitted.values, ylab='Probability', ylim=c(0,1), xlab = 'age', main='Empirical Pro
points(age.slice.avg, age.prob, pch=16, col='blue')

```

Empirical Probability for age

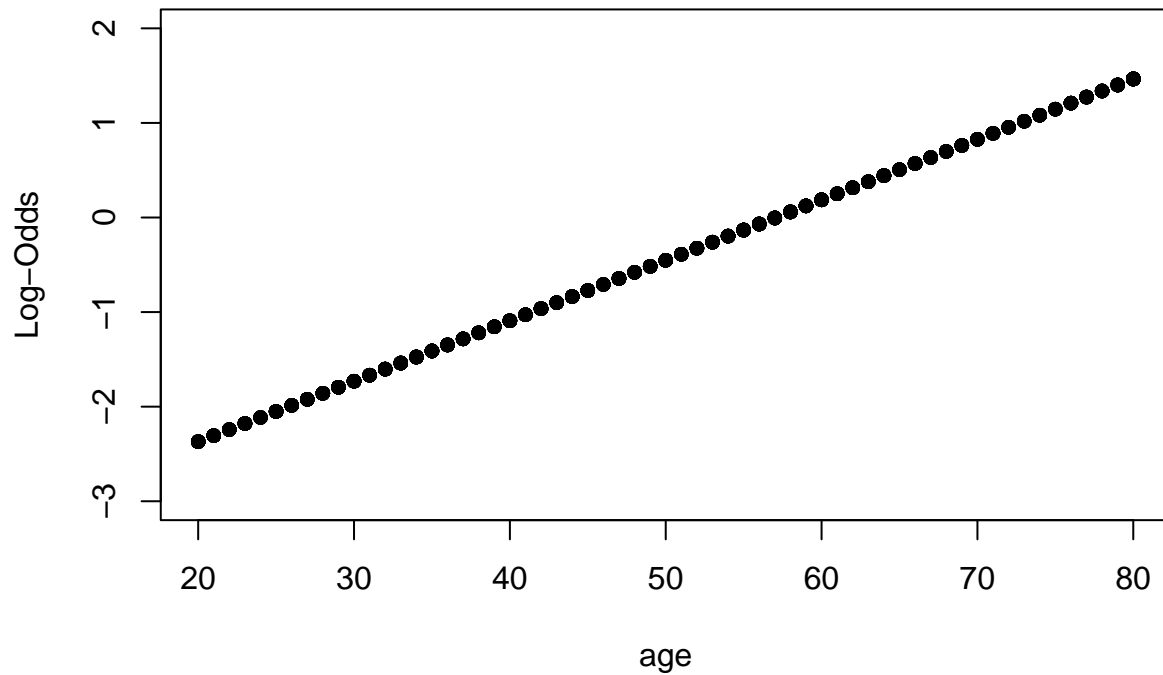


```

age.pred <- age.out$fitted.value
age.plogits <- log(age.pred/(1-age.pred))
plot(data$age, age.plogits, pch=16, ylab='Log-Odds', ylim=c(-3, 2), xlab = 'age', main='Empirical Logit
points(age.slice.avg, age.elogits, pch=16, col='blue')

```

Empirical Logit for age



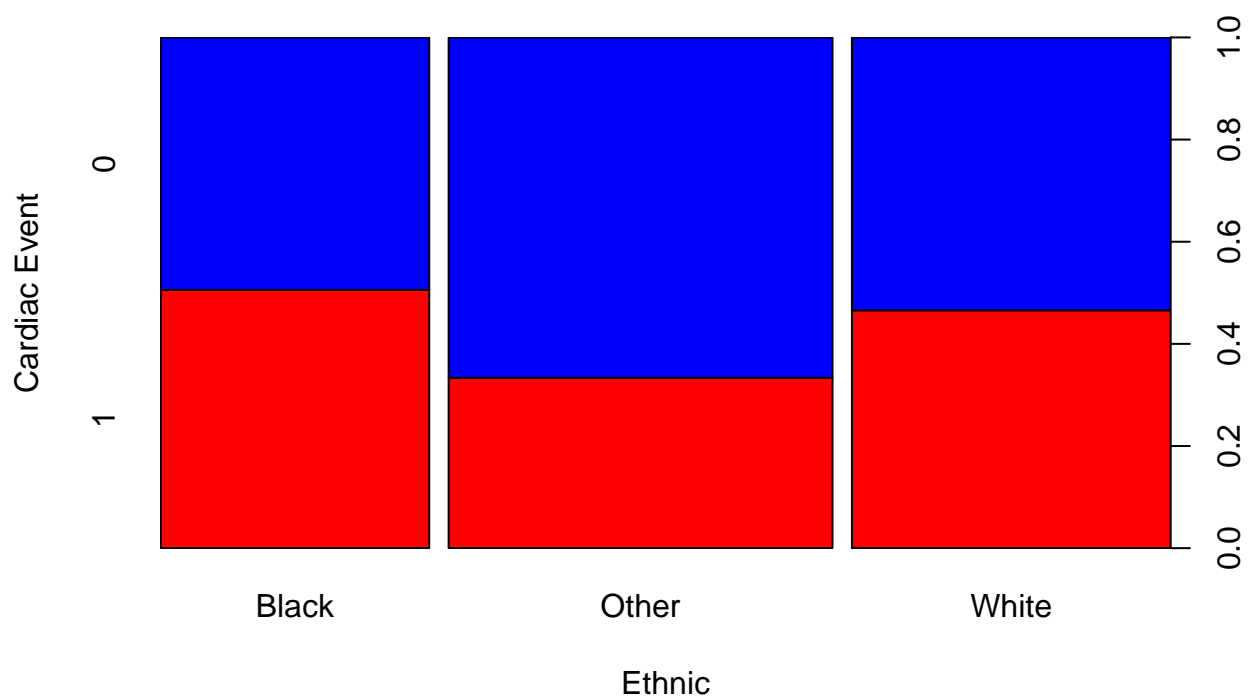
Ethnic

```
table(data$ethnic1)
```

```
##  
##  1  2  3  4  5  
## 230 195 627 528 330
```

```
data <- data %>%  
  mutate(ethnic1 = case_when(  
    ethnic1 == 1 ~ "Other",  
    ethnic1 == 2 ~ "Other",  
    ethnic1 == 3 ~ "White",  
    ethnic1 == 4 ~ "Black",  
    ethnic1 == 5 ~ "Other",  
    TRUE ~ as.character(ethnic1)  
  ))  
  
ethnic_tab <- table(data$event, data$ethnic1)  
spineplot(t(ethnic_tab), main = "Ethnic and Cardiac Event",  
  xlab="Ethnic", ylab="Cardiac Event", col=c("red","blue"))
```


Ethnic and Cardiac Event



```
data$ethnic1 <- as.factor(data$ethnic1)
```

Educ

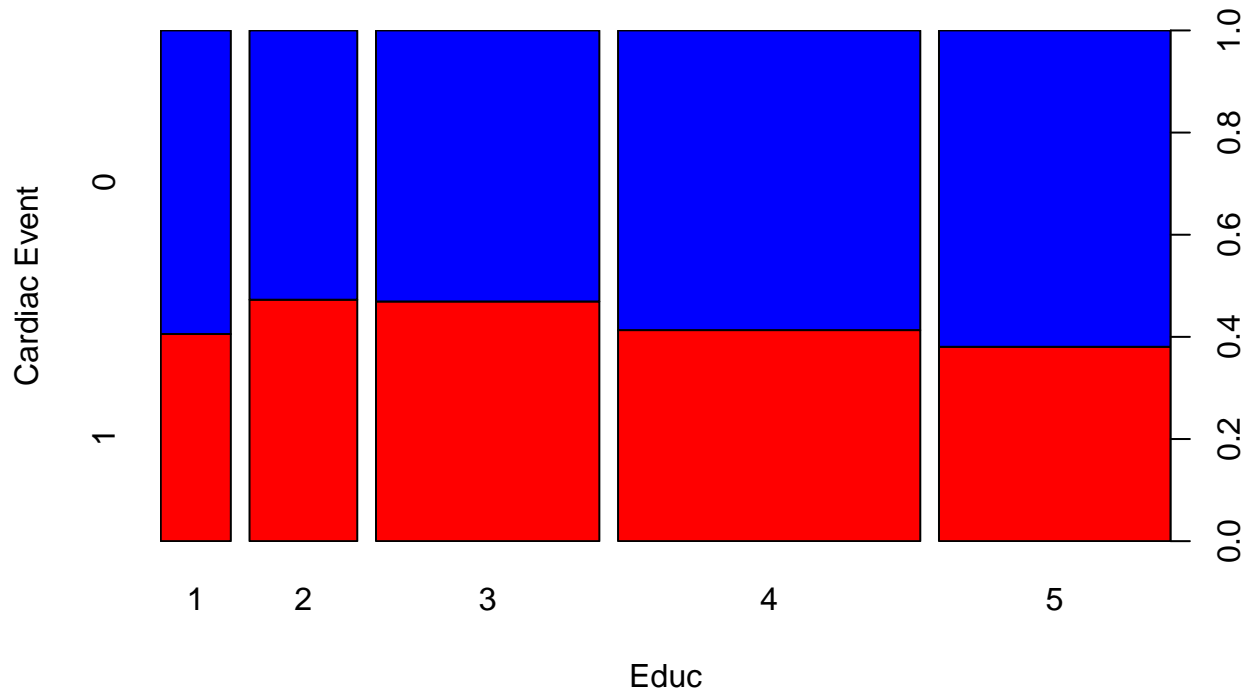
```
table(data$educ)
```

```
##
##  1  2  3  4  5  9
## 143 220 456 617 473  1
```

```
# Since educ = 9 means Don't know, it's hard to assign value and with only one record. Remove the record
data <- data[data$educ != 9, ]
```

```
educ_tab <- table(data$event, data$educ)
spineplot(t(educ_tab), main = "Educ and Cardiac Event",
          xlab="Educ", ylab="Cardiac Event", col=c("red","blue"))
```

Educ and Cardiac Event



```
CMHtest(educ_tab)
```

```
## Cochran-Mantel-Haenszel Statistics for by
##
##               AltHypothesis  Chisq Df    Prob
## cor             Nonzero correlation   4.3993  1 0.035954
## rmeans   Row mean scores differ   4.3993  1 0.035954
## cmeans   Col mean scores differ 10.0988  4 0.038796
## general      General association 10.0988  4 0.038796
```

```
data$educ <- as.factor(data$educ)
```

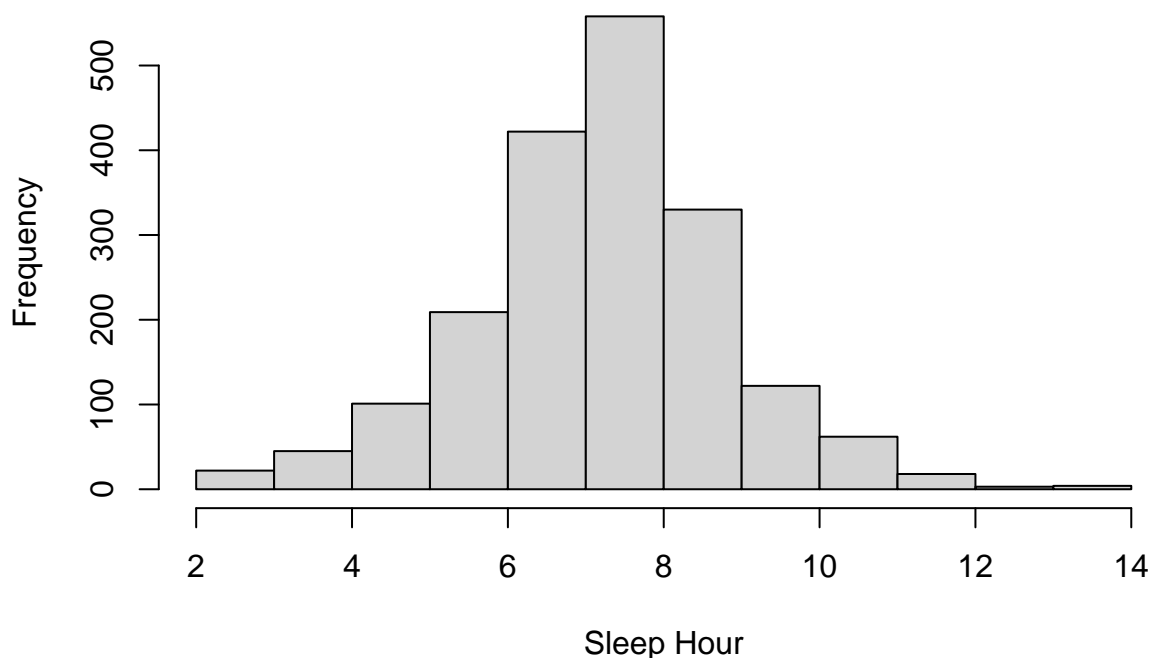
Processing Variables with significant amount of missing values

```
# Only keep records without N/A value to do the predict
data_no_na <- na.omit(data)
```

Sleep hrs

```
# Sleep distribution
hist(data$sleep.hrs, main="Histogram for Sleep Hour", xlab="Sleep Hour")
```

Histogram for Sleep Hour



```
# According to the correlation, use the highest possible predictors
# Use linear regression or logistic regression to predict the value of the NA
sleep_fit <- lm(sleep.hrs ~ gender+age+ethnic1+educ, data)
summary(sleep_fit)
```

```
##
## Call:
## lm(formula = sleep.hrs ~ gender + age + ethnic1 + educ, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.7567 -0.8773  0.0845  0.8807  6.8409
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.069670   0.213684  33.085 < 2e-16 ***
## gender2       0.224838   0.075323   2.985  0.00287 **
## age           0.004296   0.002233   1.924  0.05450 .
## ethnic10ther  0.231463   0.095843   2.415  0.01583 *
## ethnic1White  0.291135   0.098217   2.964  0.00307 **
## educ2        -0.033967   0.179713  -0.189  0.85011
## educ3        -0.048061   0.164188  -0.293  0.76977
## educ4        -0.066967   0.160005  -0.419  0.67561
## educ5        -0.010502   0.161354  -0.065  0.94811
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.635 on 1887 degrees of freedom
## (13 observations deleted due to missingness)
## Multiple R-squared:  0.01241,    Adjusted R-squared:  0.008222
```

```
## F-statistic: 2.964 on 8 and 1887 DF, p-value: 0.002674
# Based on first try, use the most significant predictors
sleep_fit1 <- lm(sleep.hrs ~ gender+ethnic1, data)
summary(sleep_fit1)

##
## Call:
## lm(formula = sleep.hrs ~ gender + ethnic1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.7789 -0.7789  0.0422  0.9369  6.7580
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.24203    0.08253  87.753 < 2e-16 ***
## gender2       0.21574    0.07518   2.870 0.004157 **
## ethnic1Other  0.23540    0.09321   2.525 0.011639 *
## ethnic1White  0.32110    0.09695   3.312 0.000944 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.635 on 1892 degrees of freedom
## (13 observations deleted due to missingness)
## Multiple R-squared:  0.01002, Adjusted R-squared:  0.008453
## F-statistic: 6.385 on 3 and 1892 DF, p-value: 0.0002651

# Evaluate Model Performance
# Predicted values from the model
predicted_sleep <- predict(sleep_fit1, newdata = data_no_na)

# R-squared ( $R^2$ ); low  $R^2$  values, so regression model does not perform well. Thus, we decide to replace
rsquared <- 1 - (sum((data_no_na$sleep.hrs - predicted_sleep)^2) / sum((data_no_na$sleep.hrs - mean(data_no_na$sleep.hrs))^2))

## [1] 0.01105339

# Replace the missing value using the predict model
# Find rows with missing values
sleep_rows_with_na <- is.na(data$sleep.hrs)
data$sleep.hrs[sleep_rows_with_na] <- median(data_no_na$sleep.hrs)

# Check Result
sum(is.na(data$sleep.hrs))

## [1] 0

summary(data$sleep.hrs)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   6.500   7.500   7.552   8.500  14.000

# Creating Slicing plot for sleeping hrs

# Table to show distribution
table(data$sleep.hrs)

##
```

```
##      2      3 3.5      4 4.5      5 5.5      6 6.5      7 7.5      8 8.5      9 9.5     10
##      6     16      6    39     19    82     45    164    106    316    193    378    131    199     45     77
## 10.5    11 11.5     12     13     14
##     24     38      6     12      3      4
```

```
# According to data distribution, we can find the numbers of observation having sleep hours less than 5
data$sleep.hrs <- cut(data$sleep.hrs, breaks=c(-Inf, 5, 6, 7, 9, Inf), labels=c("<=5 hrs", "5-6 hrs", "6-7 hrs", "7-9 hrs", ">9 hrs"))
table(data$sleep.hrs)
```

```
##
## <=5 hrs 5-6 hrs 6-7 hrs 7-9 hrs >9 hrs
##      168      209      422      901      209
```

```
sleep.fac = factor(data$sleep.hrs)
```

```
# Empirical probs for each category
sleep.prob <- tapply(data$event, sleep.fac, mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
sleep.slice.avg <- tapply(data$sleep.hrs, sleep.fac, mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
sleep.elogits <- log(sleep.prob/(1-sleep.prob))
```

```
# Run logistic regression on simulated data
```

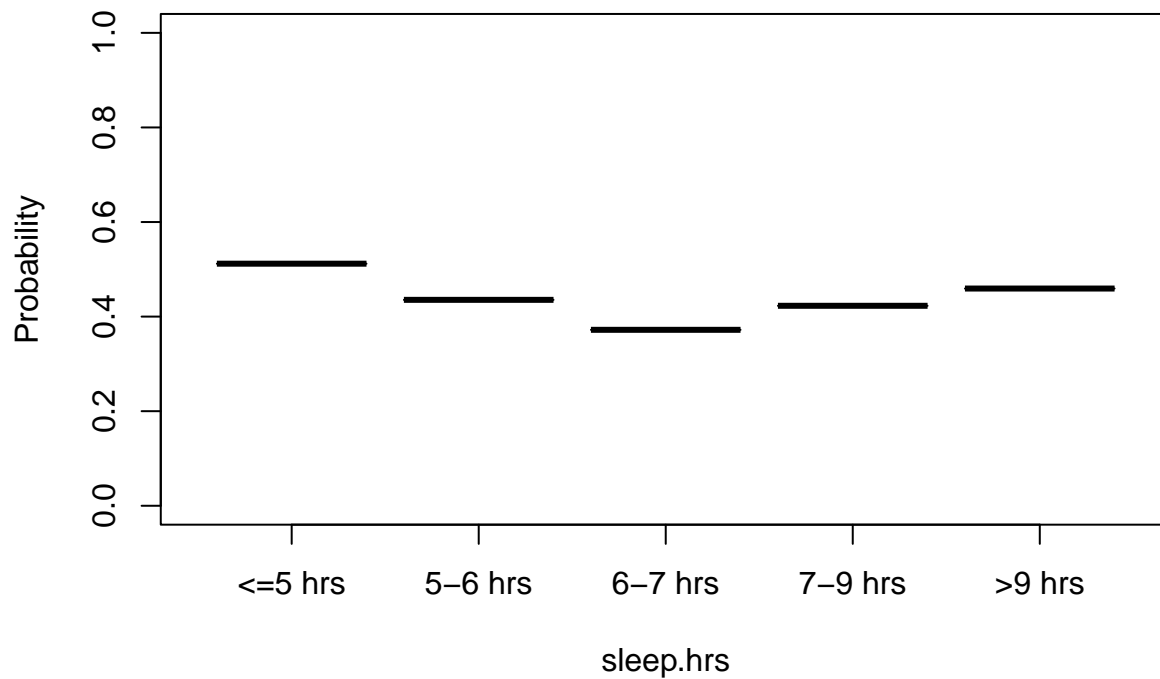
```
sleep.out <- glm(event ~ sleep.hrs, data = data, family = 'binomial')
```

```
# Graph predicted and empirical probabilities
```

```
plot(data$sleep.hrs, sleep.out$fitted.values, ylab='Probability', ylim=c(0,1), xlab = 'sleep.hrs', main='Logistic Regression Plot')
```

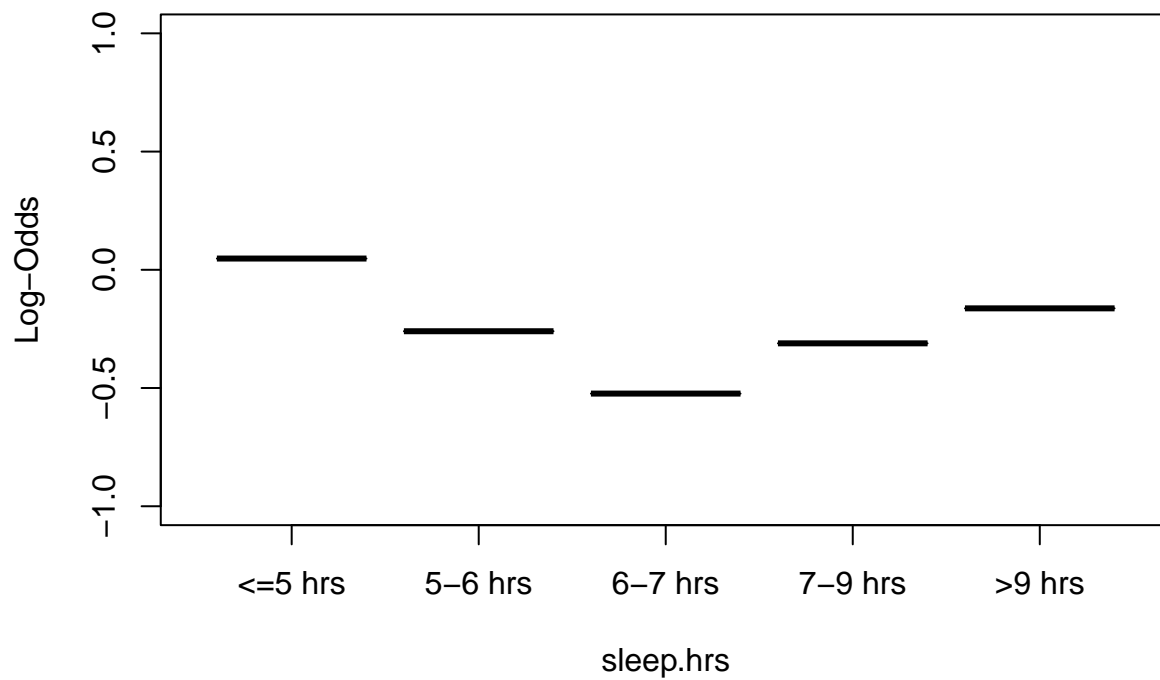
```
points(sleep.slice.avg, sleep.prob, pch=16, col='blue')
```

Empirical Probability for Sleep hours



```
slepp.pred <- sleep.out$fitted.value  
sleep.logits <- log(slepp.pred/(1-slepp.pred))  
plot(data$sleep.hrs, sleep.logits, pch=16, ylab='Log-Odds', ylim=c(-1, 1), xlab = 'sleep.hrs', main='Empirical Probability for Sleep hours')  
points(sleep.slice.avg, sleep.elogits, pch=16, col='blue')
```

Empirical Logits for Sleep hours

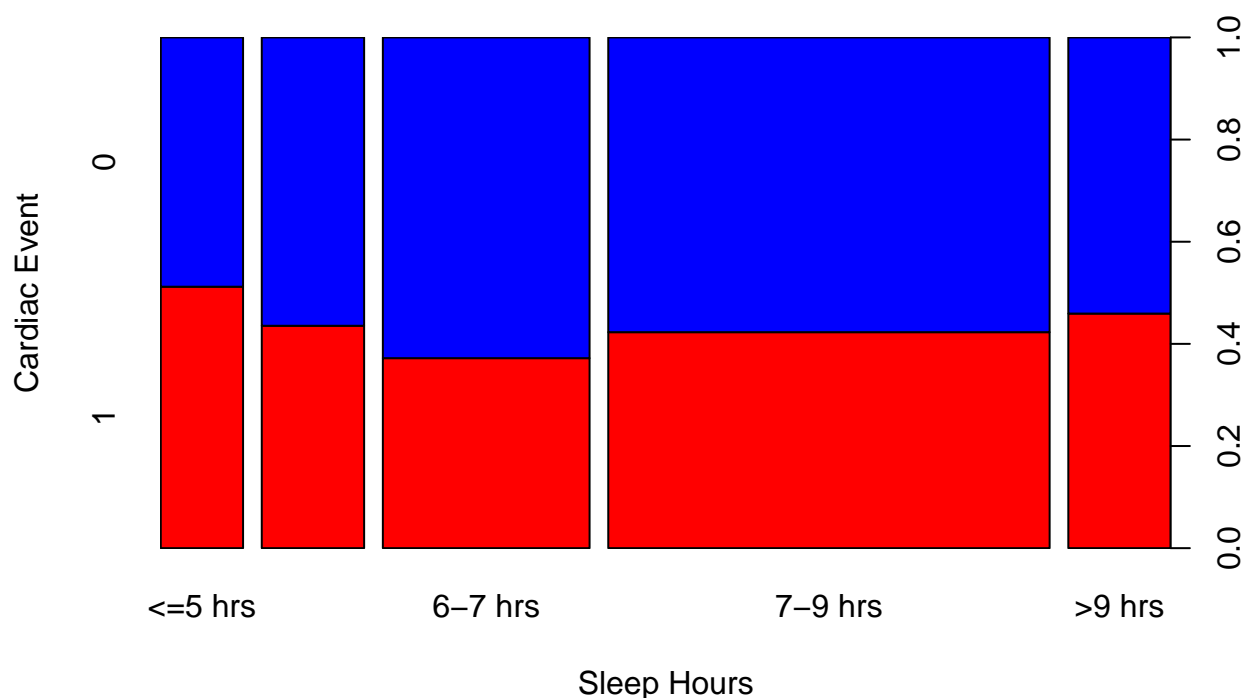


Slicing plot shows that different steps instead of straight line, so code sleep hrs as categorical variable

Mosaic plot for sleep hrs

```
sleep_tab <- table(data$event, data$sleep.hrs)
spineplot(t(sleep_tab), main = "Sleep Hours and Cardiac Event",
          xlab="Sleep Hours", ylab="Cardiac Event", col=c("red","blue"))
```

Sleep Hours and Cardiac Event



```
CMHtest(sleep_tab) # The results shows that sleep hrs and cardiac event have no linear association
```

```
## Cochran-Mantel-Haenszel Statistics for by
##
##               AltHypothesis  Chisq Df    Prob
## cor             Nonzero correlation  0.62864  1 0.427855
## rmeans   Row mean scores differ  0.62864  1 0.427855
## cmeans   Col mean scores differ 11.14836  4 0.024946
## general      General association 11.14836  4 0.024946
```

```
# Convert sleep to factor
```

```
data$sleep.hrs <- as.factor(data$sleep.hrs)
```

Diabetes

```
# Create a table to view the distribution of data
```

```
table(data$diabetes)
```

```
##
##      1      2      3      9
## 303 1489      44      1
```

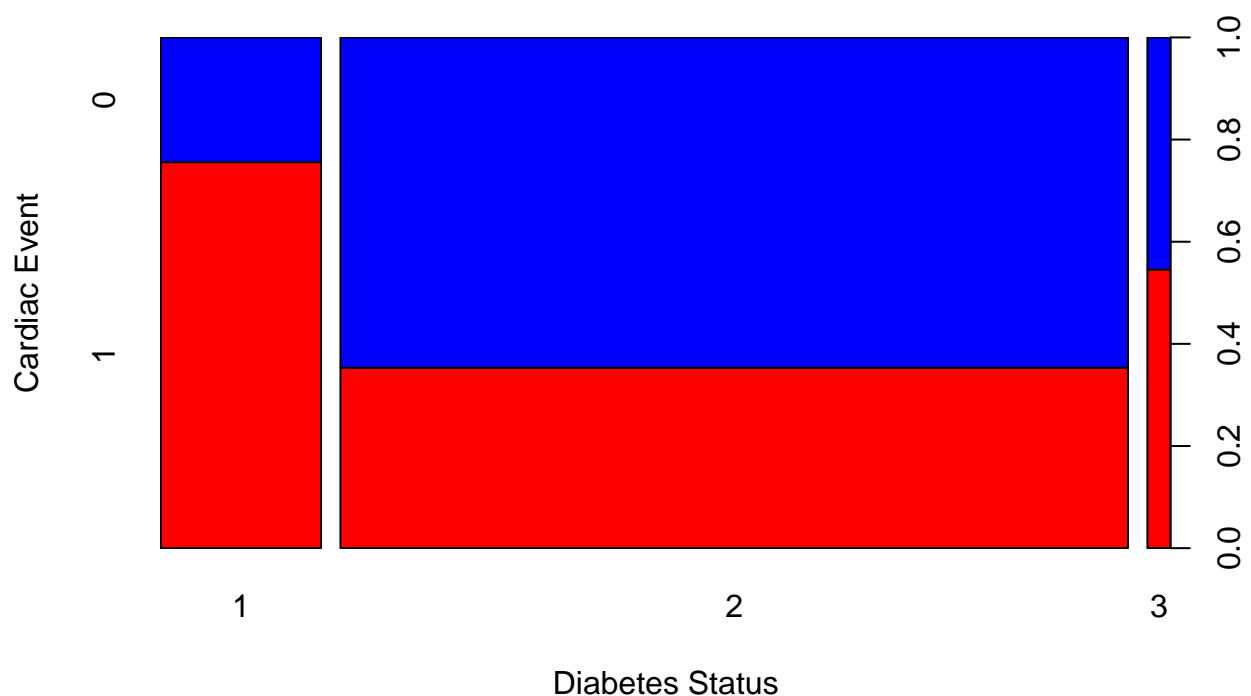
```
# Change Don't know as NaN, which will be processed later.
```

```
data$diabetes[data$diabetes == 9] <- NaN
```

```
diabete_tab <- table(data$event, data$diabetes)
```

```
spineplot(t(diabete_tab), main = "Diabetes Status and Cardiac Event",
          xlab="Diabetes Status", ylab="Cardiac Event", col=c("red", "blue"))
```


Diabetes Status and Cardiac Event



```
# Run trend data
CMHtest(diabete_tab)
```

```
## Cochran-Mantel-Haenszel Statistics for by
##
##               AltHypothesis  Chisq Df    Prob
## cor             Nonzero correlation 119.21  1 9.4070e-28
## rmeans   Row mean scores differ 119.21  1 9.4070e-28
## cmeans   Col mean scores differ 169.61  2 1.4765e-37
## general      General association 169.61  2 1.4765e-37
```

```
# The trend test shows that there is a linear association, so keep borderline category
```

```
# Since diabetes has three categories, it is challenging to run logistic regression to predict diabetes
# Thus, we removed all NA
```

```
data<-subset(data, !is.na(diabetes))
```

```
# Recode diabete as ordinal categorical varaibles
```

```
data$diabetes[data_no_na$diabetes == 2] <- 0
data$diabetes[data_no_na$diabetes == 1] <- 2
data$diabetes[data_no_na$diabetes == 3] <- 1
```

```
# Check if na exists
```

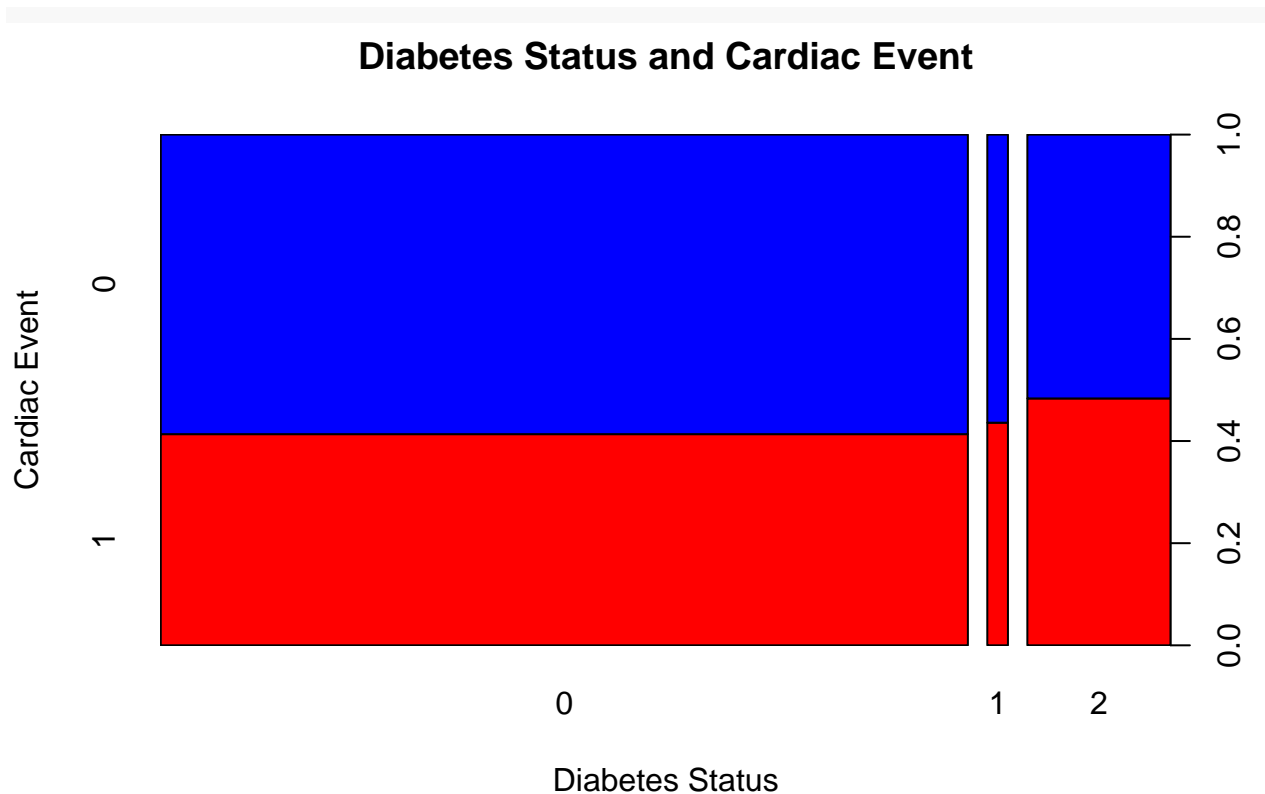
```
table(data$diabetes,useNA = "ifany")
```

```
##
```

```
##    0    1    2
## 1526  39 271
```

```
diabete_tab <- table(data$event, data$diabetes)
```

```
spineplot(t(diabete_tab), main = "Diabetes Status and Cardiac Event",
          xlab="Diabetes Status", ylab="Cardiac Event", col=c("red","blue"))
```



Smoker

```
# check distribution for smoker
table(data$smoker)
```

```
##
##      1      2
## 402 1292
```

```
smoker_tab <- table(data$event, data$smoker)
```

```
# For Categorical response, first Categorize the result to (0-No smoke; 1-smoke)
# turn 2 into 0, which not only turns data to correct format of data for logistic regression but also m
data$smoker[data$smoker == 2] <- 0
smoke_pr <- glm(smoker ~ gender + educ + age, data = data, family = binomial)
```

```
# Model evaluation
summary(smoke_pr)
```

```
##
## Call:
## glm(formula = smoker ~ gender + educ + age, family = binomial,
##      data = data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.096136   0.321853  -0.299   0.76517
## gender2      -0.664148   0.120470 -5.513 3.53e-08 ***
## educ2         0.880277   0.295409  2.980  0.00288 **
## educ3         0.577832   0.274894  2.102  0.03555 *
```

```

## educ4      0.322912  0.271656  1.189  0.23457
## educ5     -0.757818  0.298252 -2.541  0.01106 *
## age       -0.020030  0.003485 -5.748  9.03e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1856.5  on 1693  degrees of freedom
## Residual deviance: 1718.9  on 1687  degrees of freedom
## (142 observations deleted due to missingness)
## AIC: 1732.9
##
## Number of Fisher Scoring iterations: 4
data_no_na$smoker[data_no_na$smoker == 2] <- 0
predicted_smoker <- ifelse(predict(smoke_pr, newdata = data_no_na) >= 0.5, 1, 0)
mean(data_no_na$smoker == predicted_smoker)

## [1] 0.7619048
# Missing value replace
rows_with_na <- is.na(data$smoker)
predicted_probabilities <- predict(smoke_pr, newdata = data[rows_with_na, ], type = "response")
data$smoker[rows_with_na] <- ifelse(predicted_probabilities >= 0.5, 1, 0)
data$smoker<-as.factor(data$smoker)
# Check Result
sum(is.na(data$smoker))

## [1] 0
summary(data$smoker)

##      0      1
## 1430  406

BMI
# According to the correlation, use the highest possible predictors
bmi_model = lm(bmi ~ educ + age + gender + ethnic1, data)
summary(bmi_model)

##
## Call:
## lm(formula = bmi ~ educ + age + gender + ethnic1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.910  -5.259  -1.101   3.992  54.226
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.975054   1.052972  29.417 < 2e-16 ***
## educ2       -0.563561   0.896036  -0.629  0.52946
## educ3       -0.246001   0.815208  -0.302  0.76287
## educ4        0.768022   0.793910   0.967  0.33348
## educ5       -1.281855   0.801209  -1.600  0.10980
## age         0.006609   0.010992   0.601  0.54776

```

```
## gender2          1.082075    0.369375    2.929  0.00344 **
## ethnic1Other    -2.243788    0.468925   -4.785  1.85e-06 ***
## ethnic1White    -0.963150    0.479917   -2.007  0.04491 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.79 on 1784 degrees of freedom
## (43 observations deleted due to missingness)
## Multiple R-squared:  0.03134,    Adjusted R-squared:  0.027
## F-statistic: 7.215 on 8 and 1784 DF,  p-value: 1.882e-09
```

```
# Based on first try, use the most significant predictors
bmi_model1 = lm(bmi ~ gender + ethnic1, data)
summary(bmi_model1)
```

```
##
## Call:
## lm(formula = bmi ~ gender + ethnic1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.671  -5.371  -1.278   4.004  54.997
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   31.2027     0.4050  77.035 < 2e-16 ***
## gender2        1.0932     0.3699   2.955  0.00316 **
## ethnic1Other  -2.4248     0.4581  -5.293  1.35e-07 ***
## ethnic1White  -1.0177     0.4755  -2.140  0.03247 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.819 on 1789 degrees of freedom
## (43 observations deleted due to missingness)
## Multiple R-squared:  0.02148,    Adjusted R-squared:  0.01984
## F-statistic: 13.09 on 3 and 1789 DF,  p-value: 1.853e-08
```

```
# Evaluate Model Performance
# Predicted values from the model
```

```
# R-squared ( $R^2$ )
```

```
predicted_bmi <- predict(bmi_model1, newdata = data_no_na)
```

```
# R-squared ( $R^2$ ); low  $R^2$  values, so regression model does not perform well.
```

```
rsquared <- 1 - (sum((data_no_na$bmi - predicted_bmi)^2) / sum((data_no_na$bmi - mean(data_no_na$bmi))^2))
```

```
## [1] 0.02483995
```

```
# Have terrible result in model, so we decided to use the median value to replace the missing value
```

```
rows_with_na <- is.na(data$bmi)
```

```
data$bmi[rows_with_na] <- median(data_no_na$bmi)
```

```
# Check Result
```

```
sum(is.na(data$bmi))
```

```
## [1] 0
```

```

summary(data$bmi)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    14.20  25.10   29.20   30.45   34.20   86.20

# Slicing plot for BMI

bmi.fac = factor(cut(data$bmi, breaks=c(-Inf, 25, 30, 35, Inf)), labels=c("<=25, 25-30, 30-35, >35"))
table(bmi.fac)

## bmi.fac
## <=25, 25-30, 30-35, >351 <=25, 25-30, 30-35, >352 <=25, 25-30, 30-35, >353
##                                455                                585                                390
## <=25, 25-30, 30-35, >354
##                                406

# Empirical probs for each category
bmi.prob <- tapply(data$event, bmi.fac, mean)

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

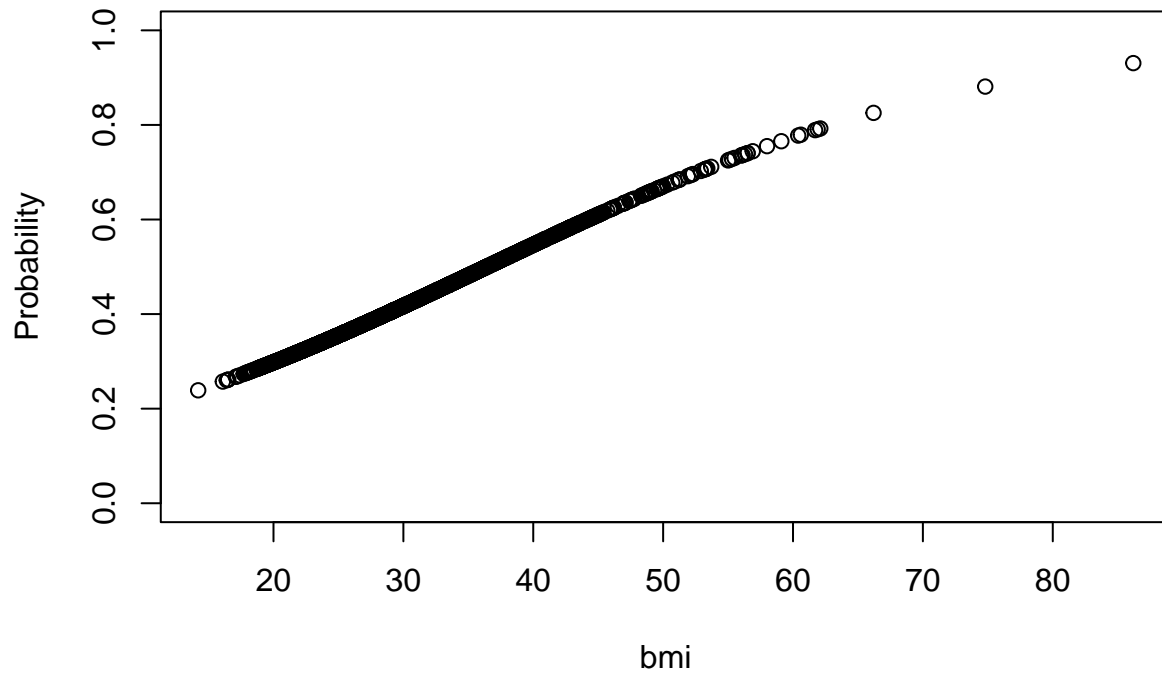
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

bmi.slice.avg <- tapply(data$bmi, bmi.fac, mean)
bmi.elogits <- log(bmi.prob/(1-bmi.prob))
# Run logistic regression on simulated data
bmi.out <- glm(event ~ bmi, data = data, family = 'binomial')

# Graph predicted and empirical probabilities
plot(data$bmi, bmi.out$fitted.values, ylab='Probability', ylim=c(0,1), xlab = 'bmi', main='Empirical Pro
points(bmi.slice.avg, bmi.prob, pch=16, col='blue')

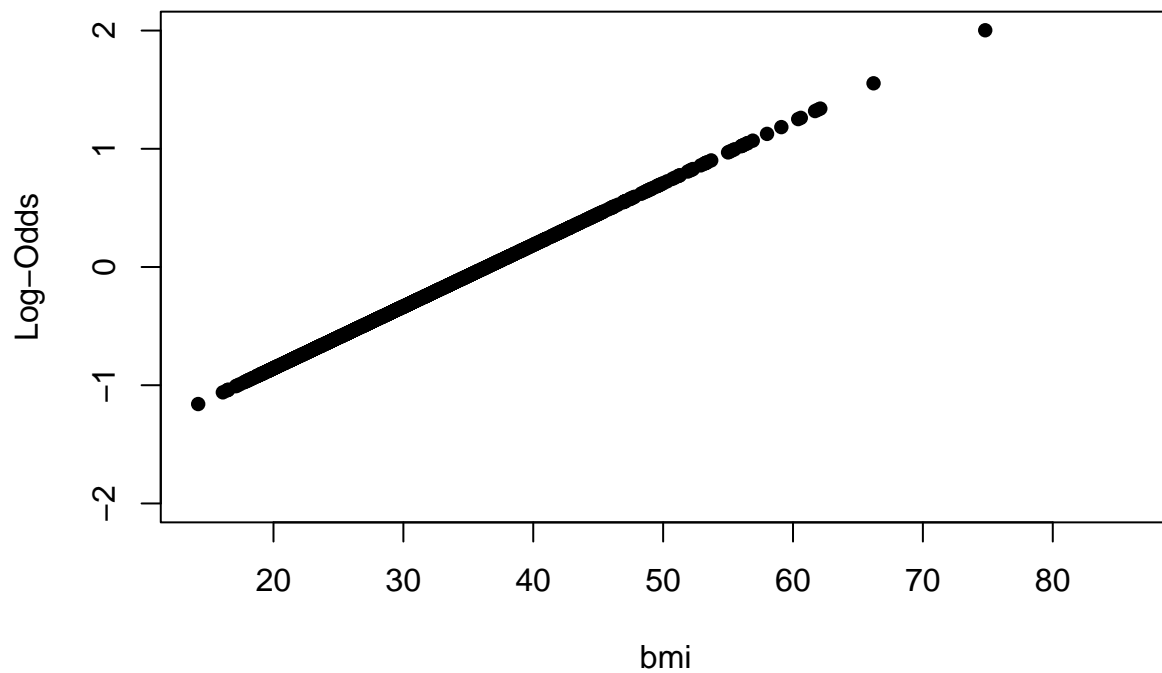
```

Empirical Probability for BMI



```
bmi.pred <- bmi.out$fitted.value
bmi.logits <- log(bmi.pred/(1-bmi.pred))
plot(data$bmi, bmi.logits, pch=16, ylab='Log-Odds', ylim=c(-2, 2), xlab = 'bmi', main='Empirical Logits')
points(bmi.slice.avg, bmi.elogits, pch=16, col='blue')
```

Empirical Logits for BMI



```
# According to the slicing plot, the log-odds of bmi is strictly numerical, so we treat it numerical.
```

```
# Confirm that there's no NA left  
anyNA(data)
```

```
## [1] FALSE
```

Step 2: Descriptive Statistics

```
summary(data)
```

```
## event      gender      age      ethnic1      educ      sleep.hrs  
## 0:1057    1:885    Min.      :20.00    Black:511    1:137    <=5 hrs:162  
## 1: 779    2:951    1st Qu.:36.00    Other:723    2:209    5-6 hrs:200  
##                      Median :52.00    White:602    3:438    6-7 hrs:407  
##                      Mean      :50.98              4:596    7-9 hrs:865  
##                      3rd Qu.:64.00              5:456    >9 hrs :202  
##                      Max.      :80.00  
##      diabetes      smoker      bmi  
## Min.      :0.0000    0:1430    Min.      :14.20  
## 1st Qu.:0.0000    1: 406    1st Qu.:25.10  
## Median :0.0000              Median :29.20  
## Mean      :0.3164              Mean      :30.45  
## 3rd Qu.:0.0000              3rd Qu.:34.20  
## Max.      :2.0000              Max.      :86.20
```

Step 3: Two-Way Table

```
# Chi-sq test for categorical predictors
```

```
# gender
```

```
gender_tab <- table(data$event, data$gender)  
chisq.test(gender_tab)
```

```
##  
## Pearson's Chi-squared test with Yates' continuity correction  
##  
## data:  gender_tab  
## X-squared = 2.0098, df = 1, p-value = 0.1563
```

```
# ethnic
```

```
ethnic_tab <- table(data$event, data$ethnic1)  
chisq.test(ethnic_tab)
```

```
##  
## Pearson's Chi-squared test  
##  
## data:  ethnic_tab  
## X-squared = 41.875, df = 2, p-value = 8.071e-10
```

```
# sleep
```

```
sleep_tab <- table(data$event, data$sleep.hrs)  
chisq.test(sleep_tab)
```

```
##  
## Pearson's Chi-squared test  
##  
## data:  sleep_tab
```

```
## X-squared = 8.6699, df = 4, p-value = 0.0699
# educ
educ_tab <- table(data$event, data$educ)
chisq.test(educ_tab)

##
## Pearson's Chi-squared test
##
## data: educ_tab
## X-squared = 8.9461, df = 4, p-value = 0.06246
# diabetes
diabetes_tab <- table(data$event, data$diabetes)
chisq.test(diabetes_tab)

##
## Pearson's Chi-squared test
##
## data: diabetes_tab
## X-squared = 4.6246, df = 2, p-value = 0.09903
# smoker
smoker_tab <- table(data$event, data$smoker)
chisq.test(smoker_tab)

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: smoker_tab
## X-squared = 0.7801, df = 1, p-value = 0.3771
# Only ethnic seems to be associated with cardiac event.

# LR test for numerical predictors

# age
glm.age.null <- glm(event ~ 1, data = data, family = binomial)
glm.age <- glm(event ~ age, data = data, family = binomial)
lrtest(glm.age.null, glm.age)

## Likelihood ratio test
##
## Model 1: event ~ 1
## Model 2: event ~ age
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1    1 -1251.5
## 2    2 -1028.5  1 445.98 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# bmi
glm.bmi.null <- glm(event ~ 1, data = data, family = binomial)
glm.bmi <- glm(event ~ bmi, data = data, family = binomial)
lrtest(glm.bmi.null, glm.bmi)

## Likelihood ratio test
##
```



```
## Model 1: event ~ 1
## Model 2: event ~ bmi
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1    1 -1251.5
## 2    2 -1216.4  1 70.186 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

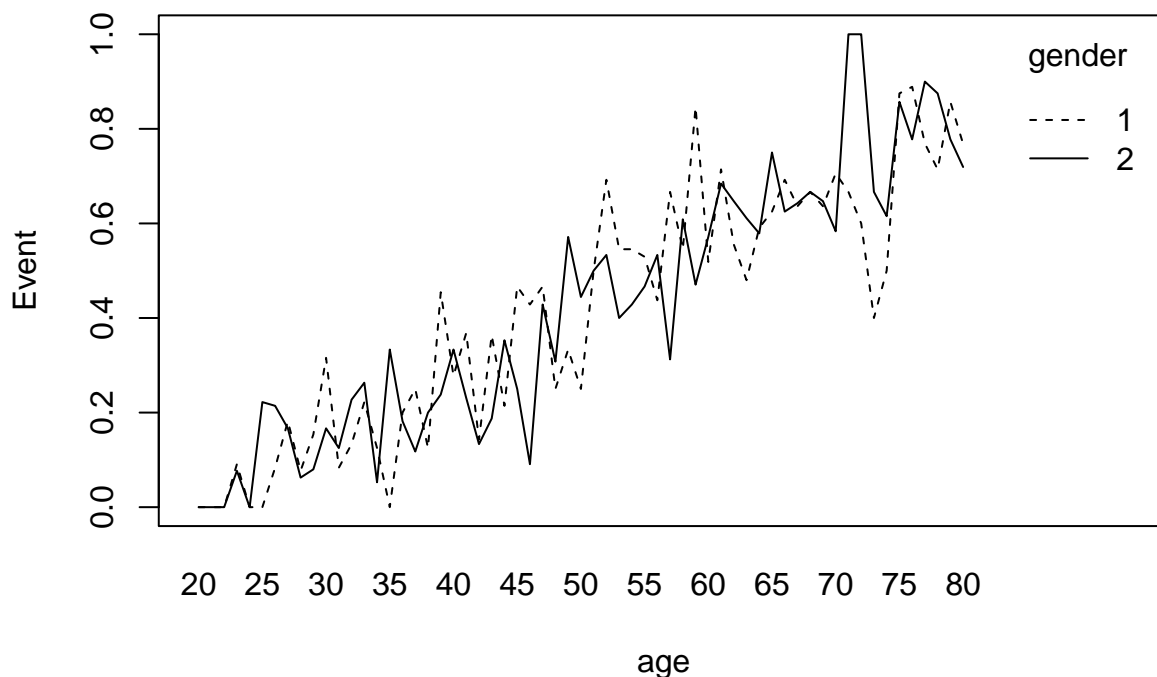
# Both age and bmi are good predictors of cardiac event.
```

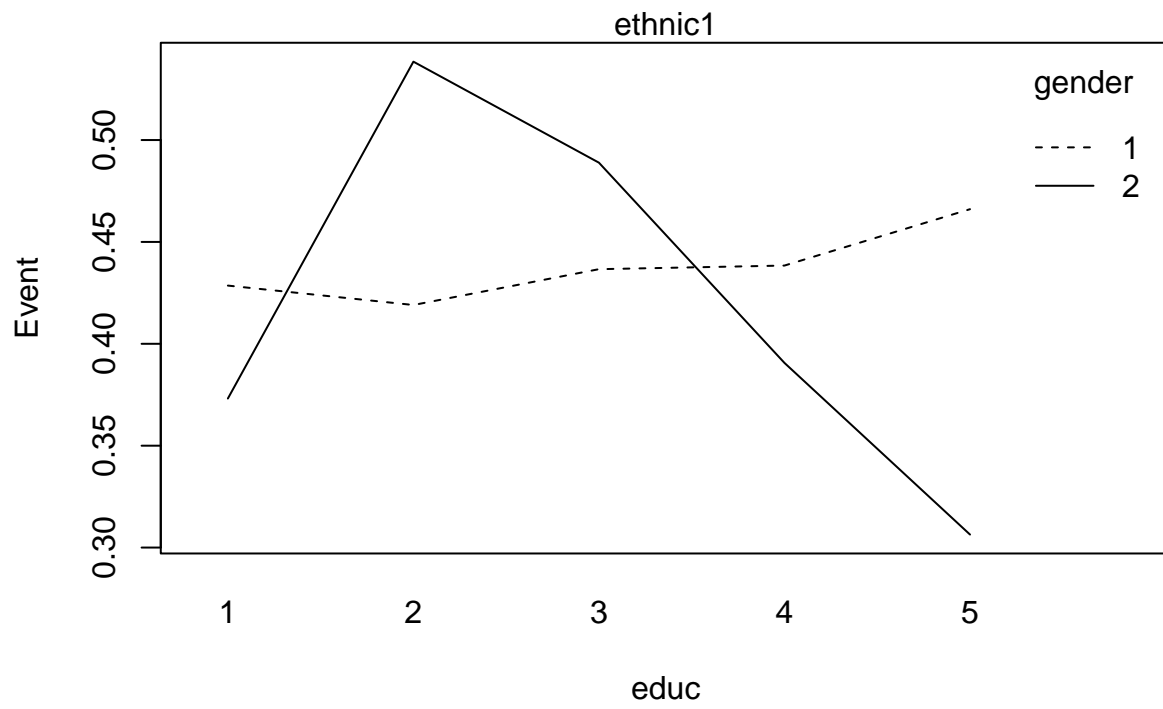
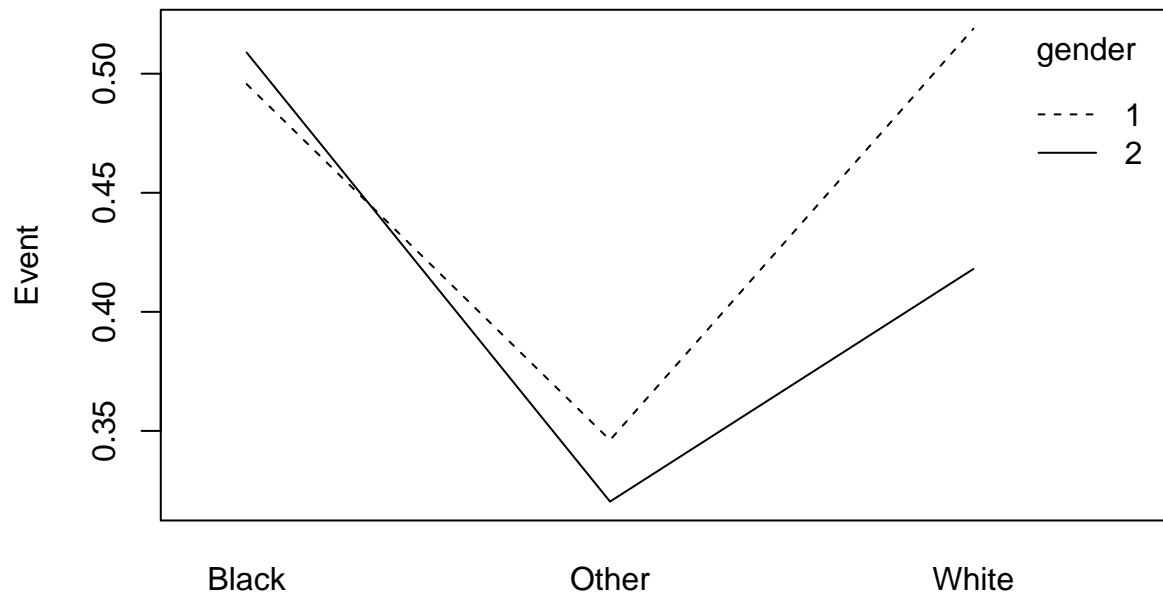
Step 4: Consider Model Option

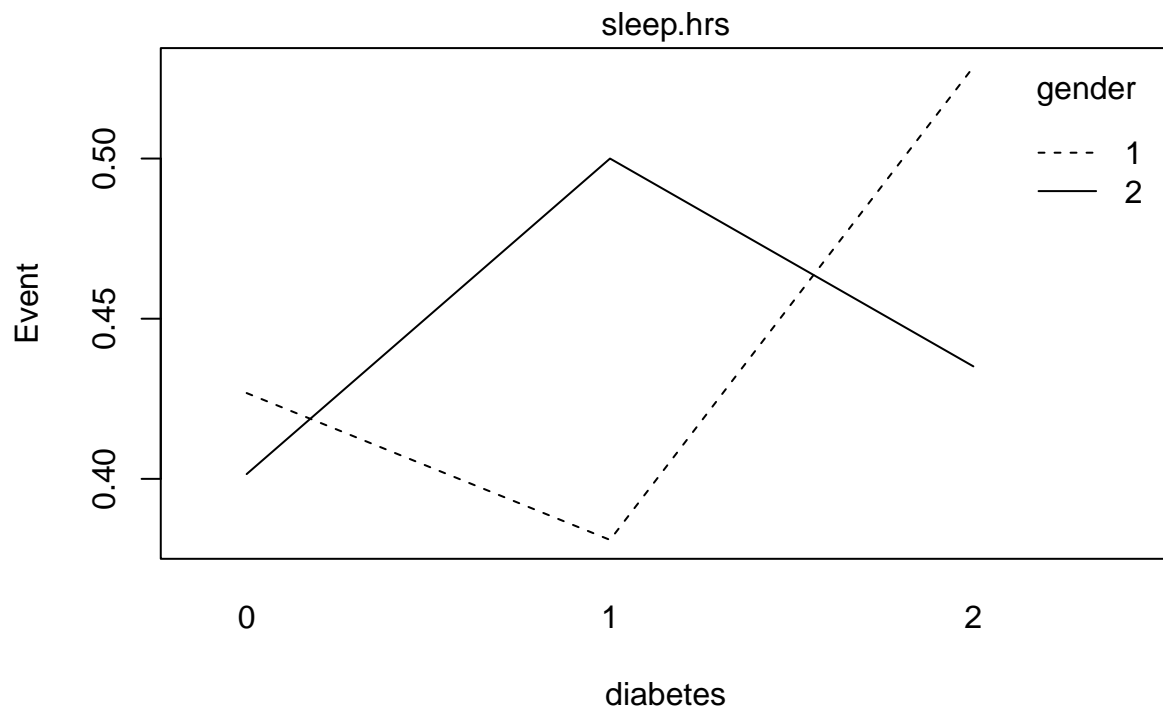
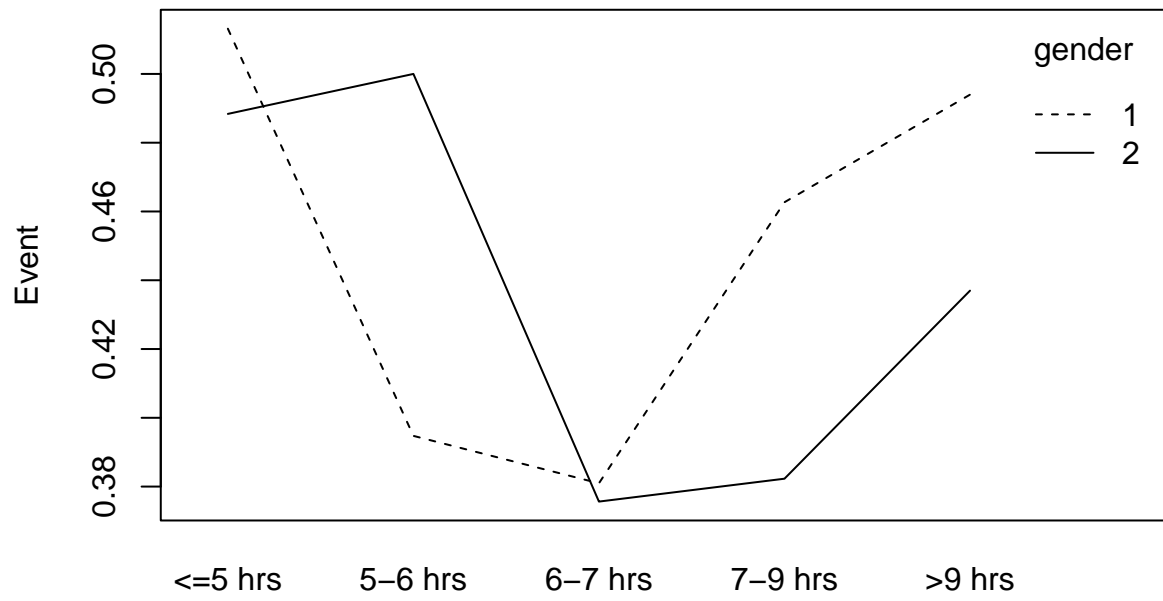
```
# Slicing plots from Step 2 shows that no transformation is needed
# We are focusing on exploring the interaction terms in this section.
```

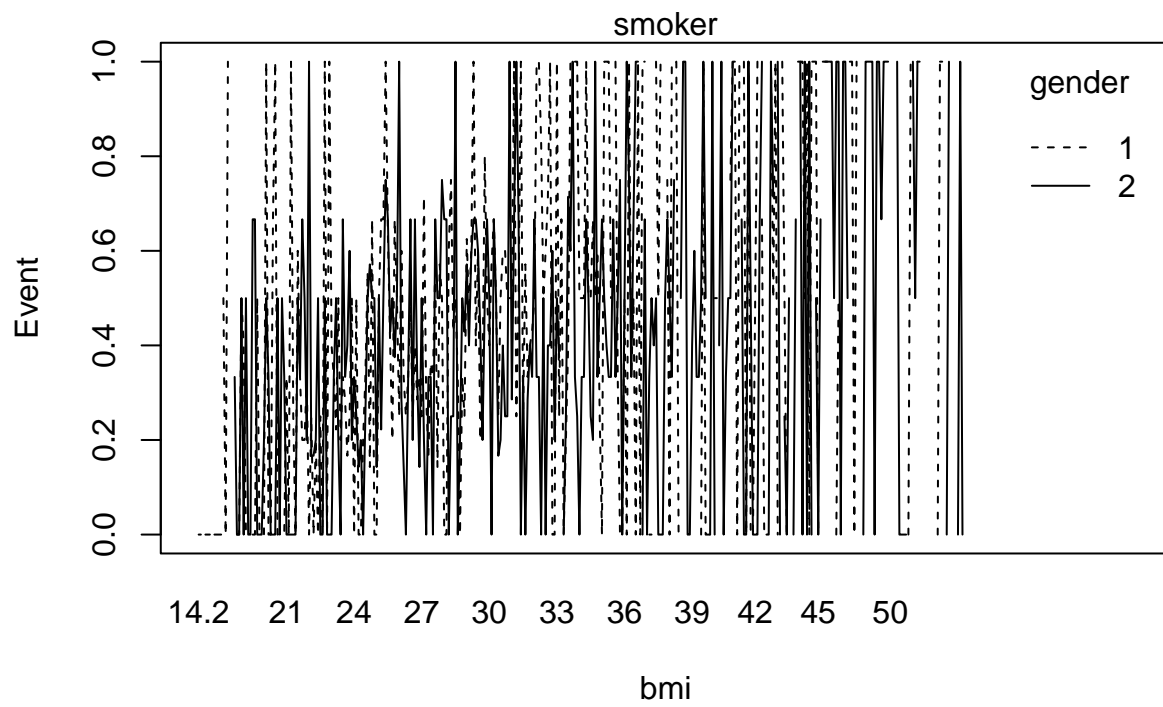
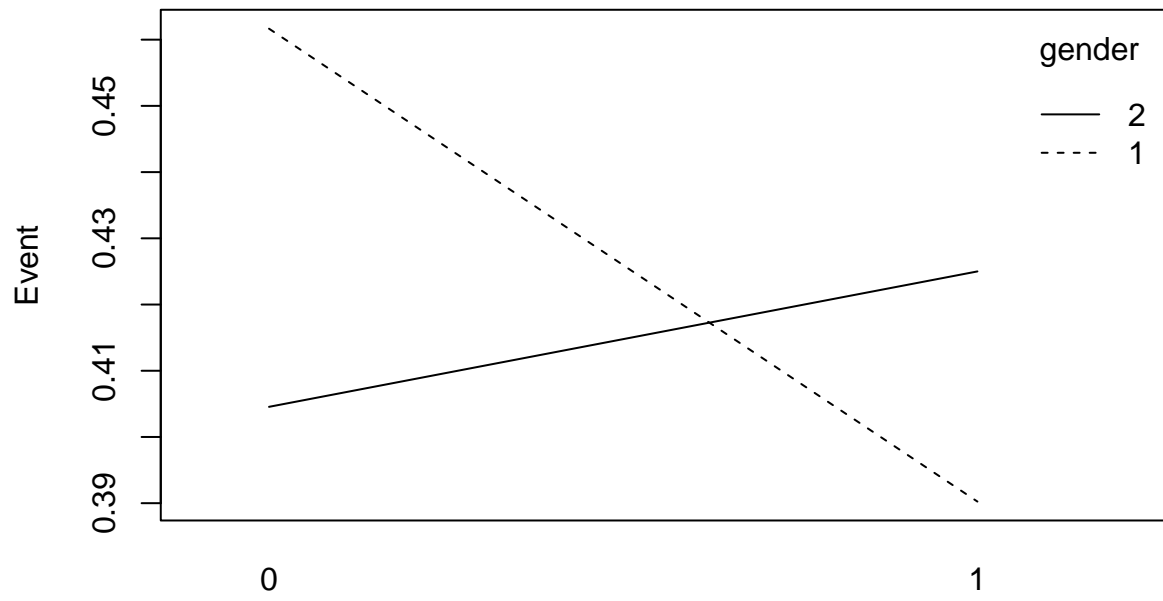
```
# interaction.plot(data$diabetes, data$smoker, response = as.numeric(data$event), trace.label = "smoker")
catv<-c("gender", "diabetes", "smoker", "ethnic1", "educ") # categorical
vars<-colnames(data)[!colnames(data)=="event"] # everything except response var.
```

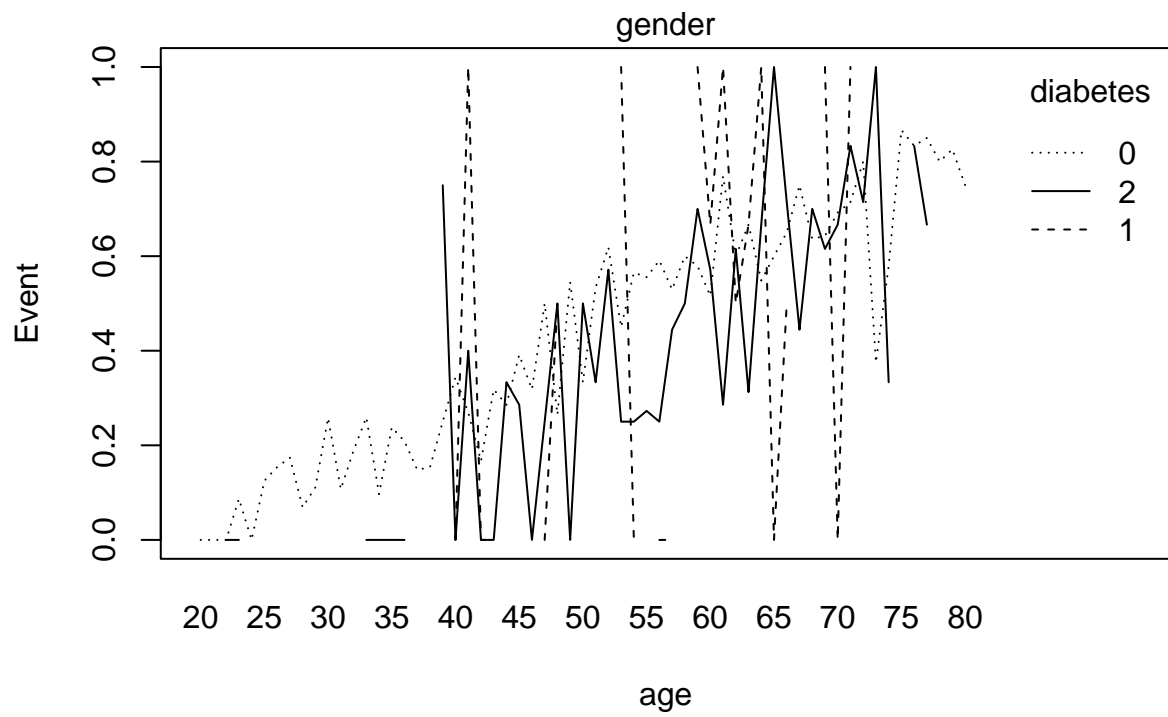
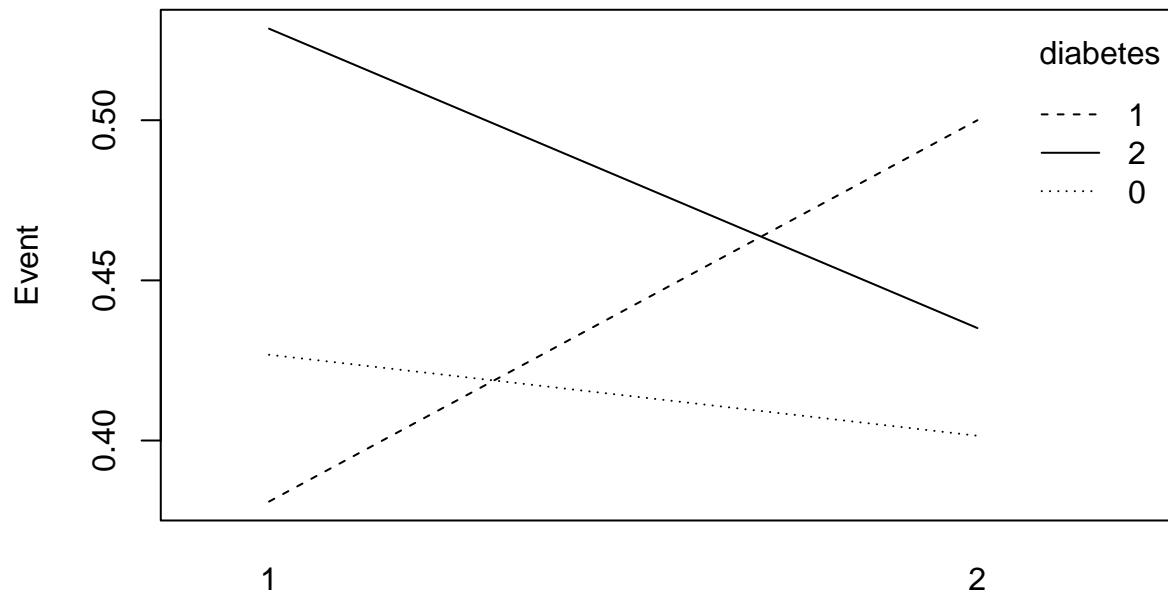
```
for (i in catv){
  for (j in vars){
    if (i!=j){
      interaction.plot(
        x.factor = as.factor(data[[j]]),
        trace.factor = as.factor(data[[i]]),
        response = as.numeric(as.character(data$event)),
        type = "l", legend = TRUE,
        xlab=j,
        ylab="Event",
        trace.label = i
      )
    }
  }
}
```

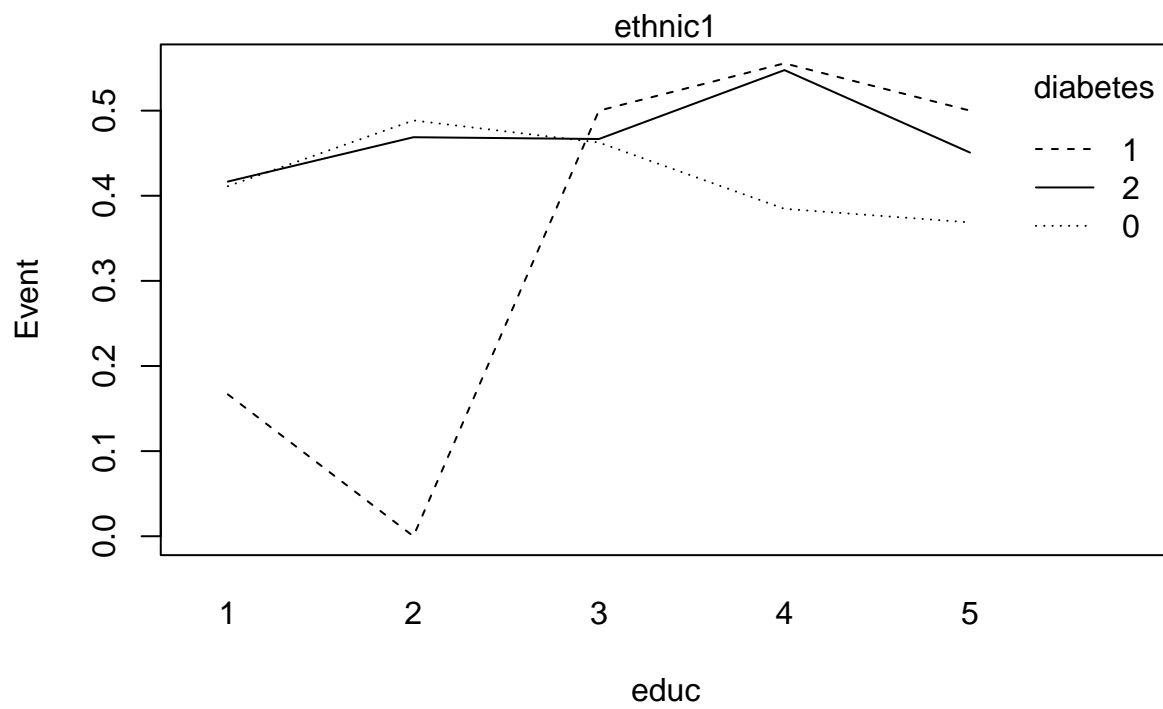
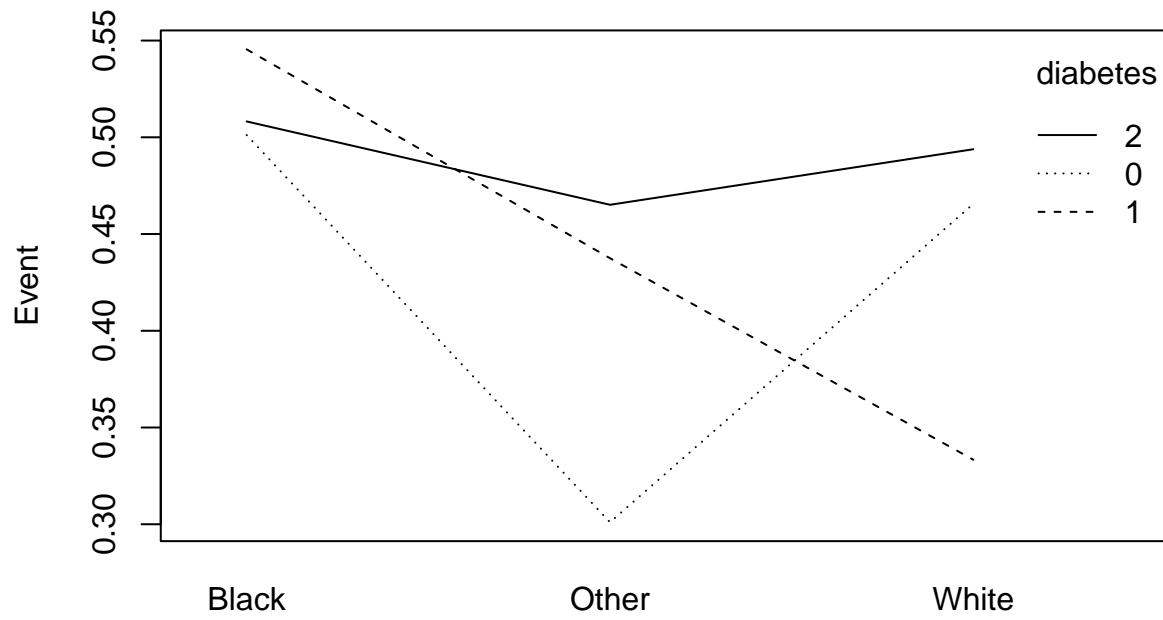


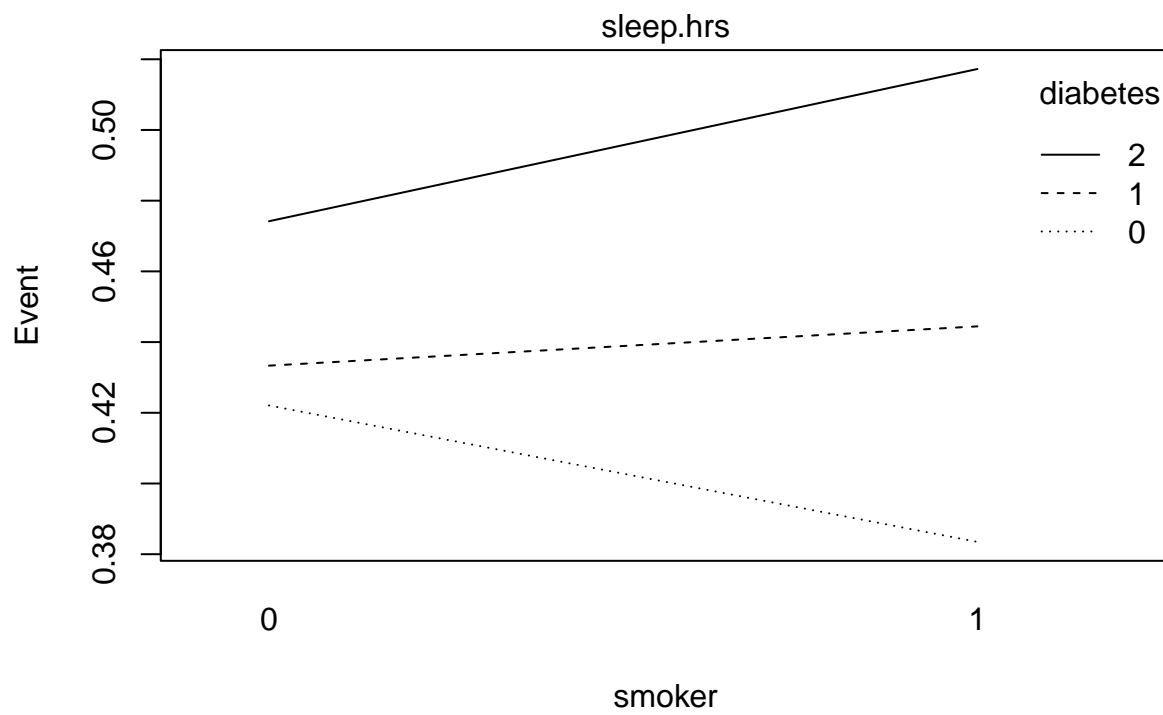
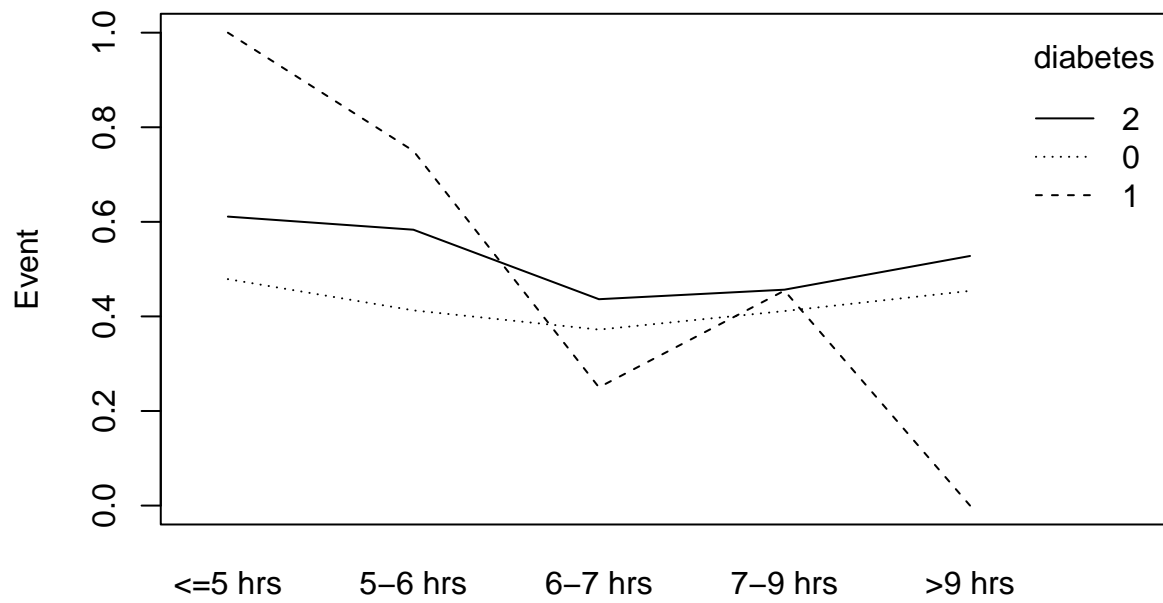


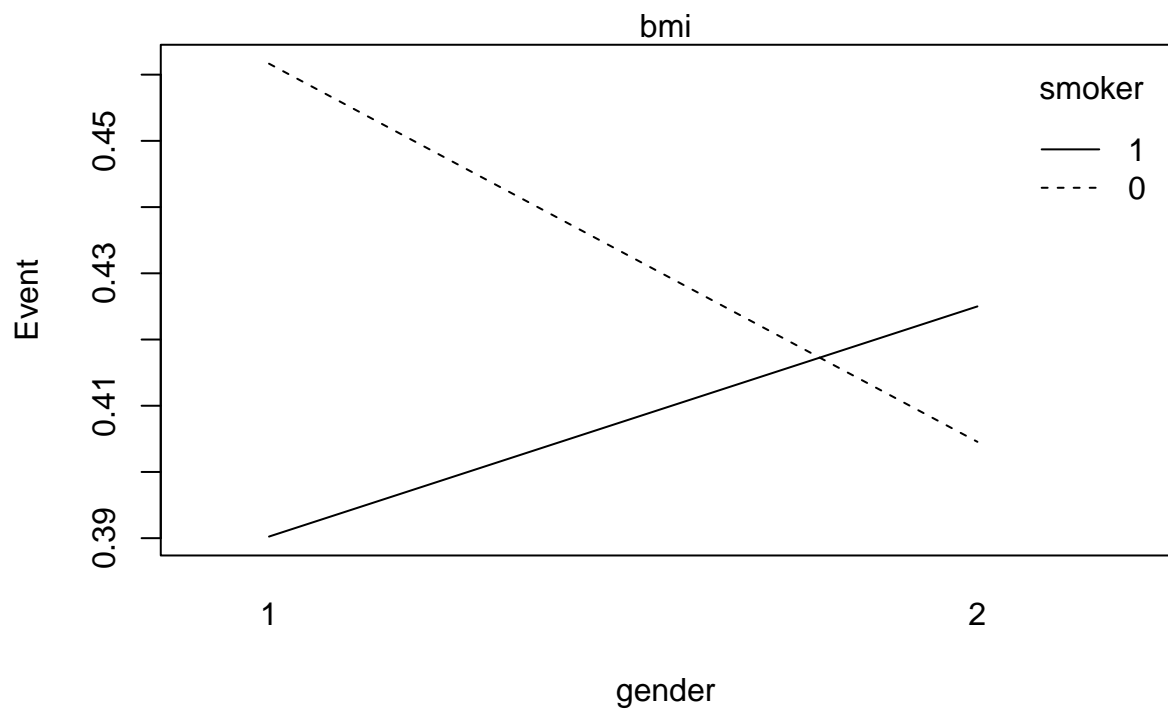
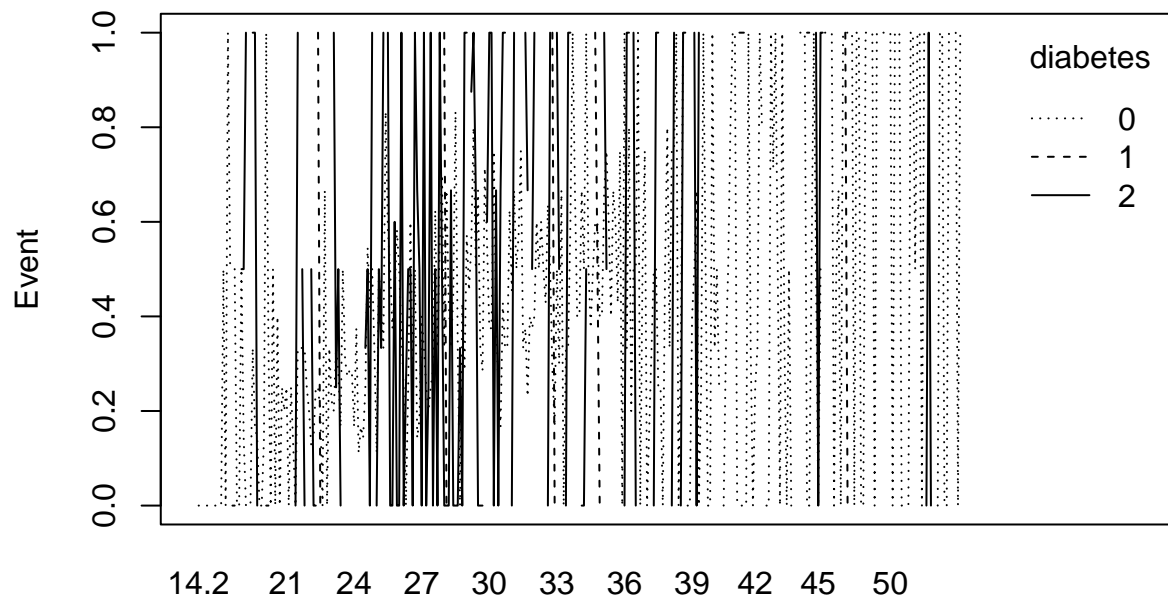


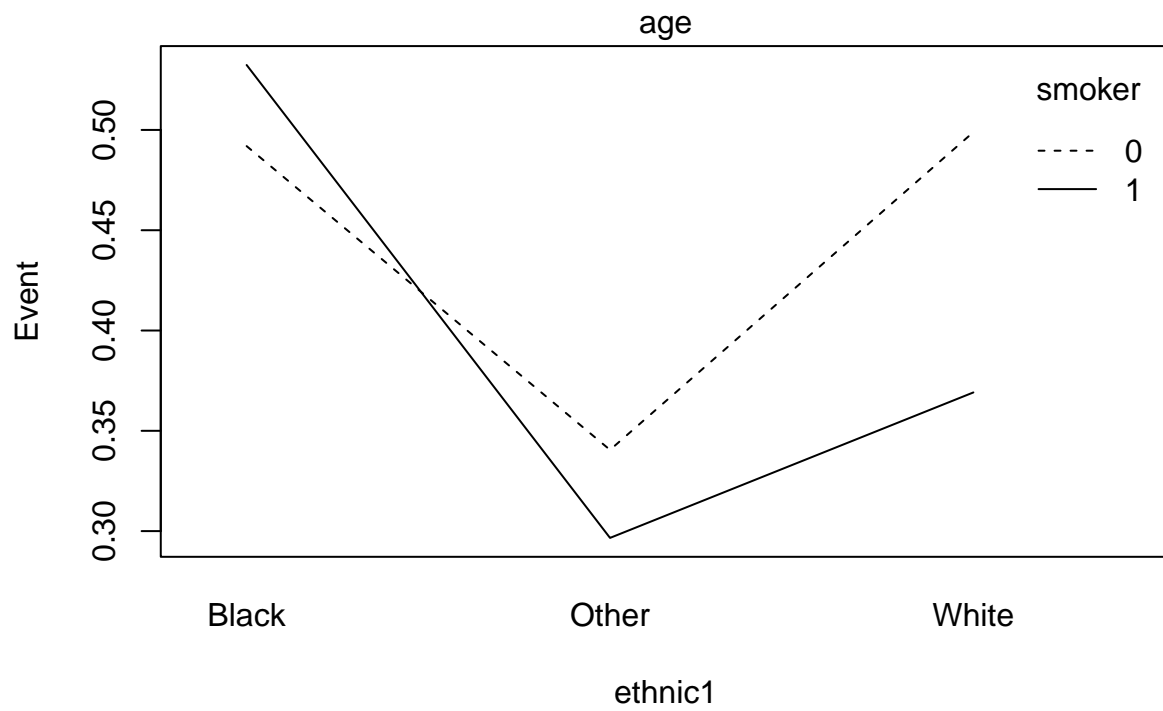
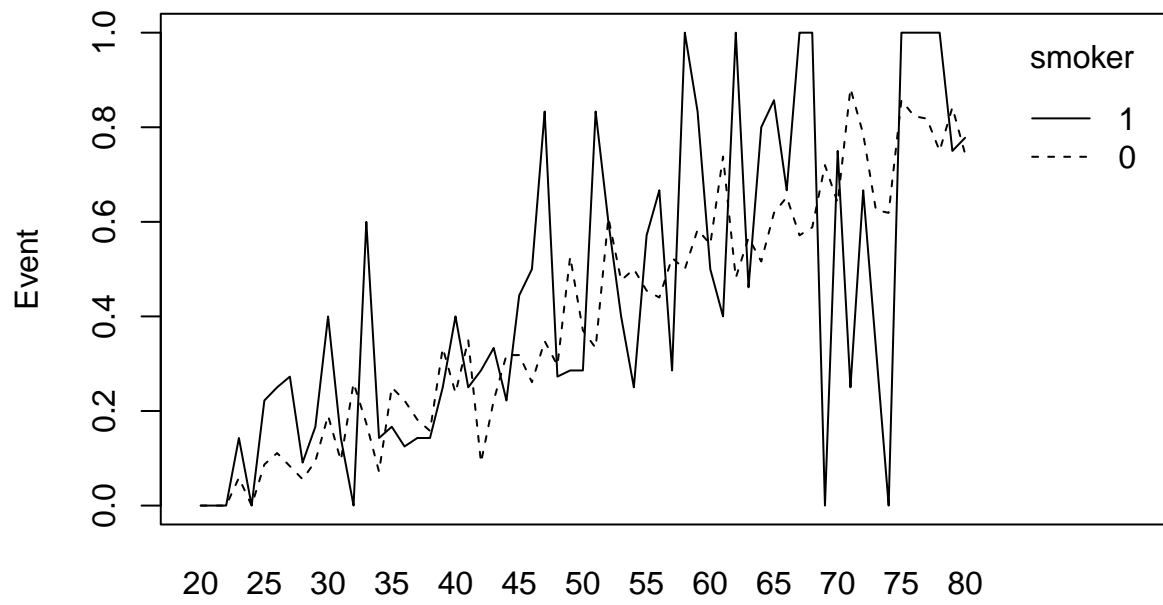


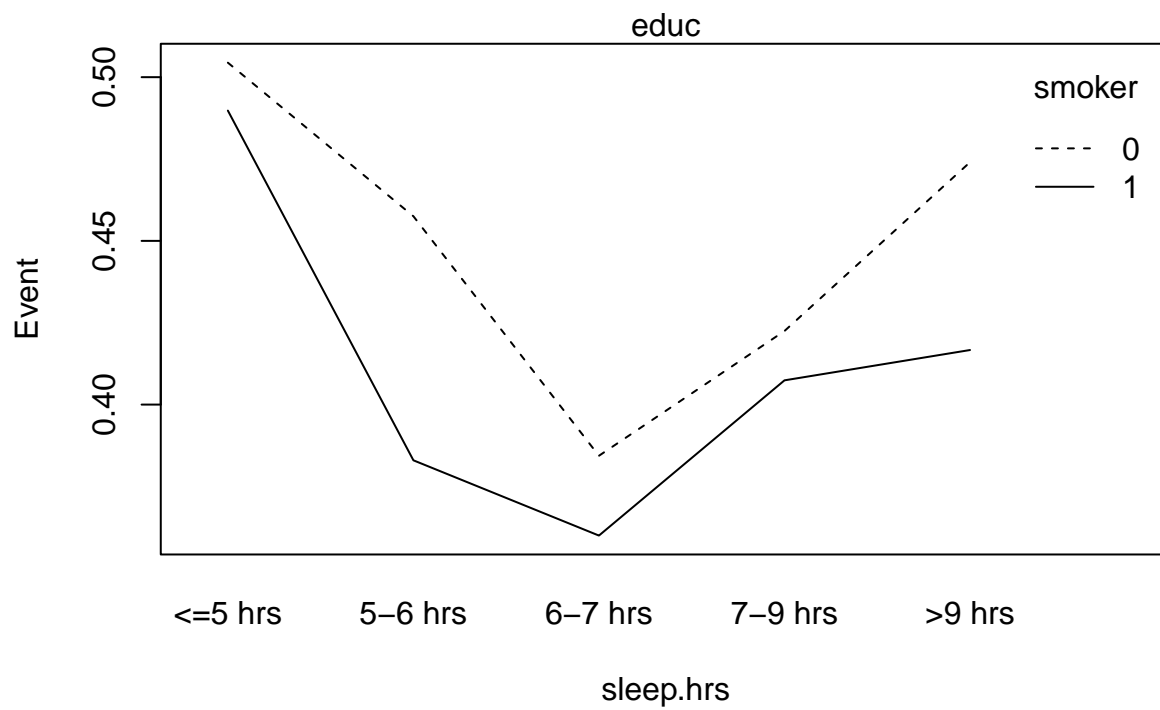
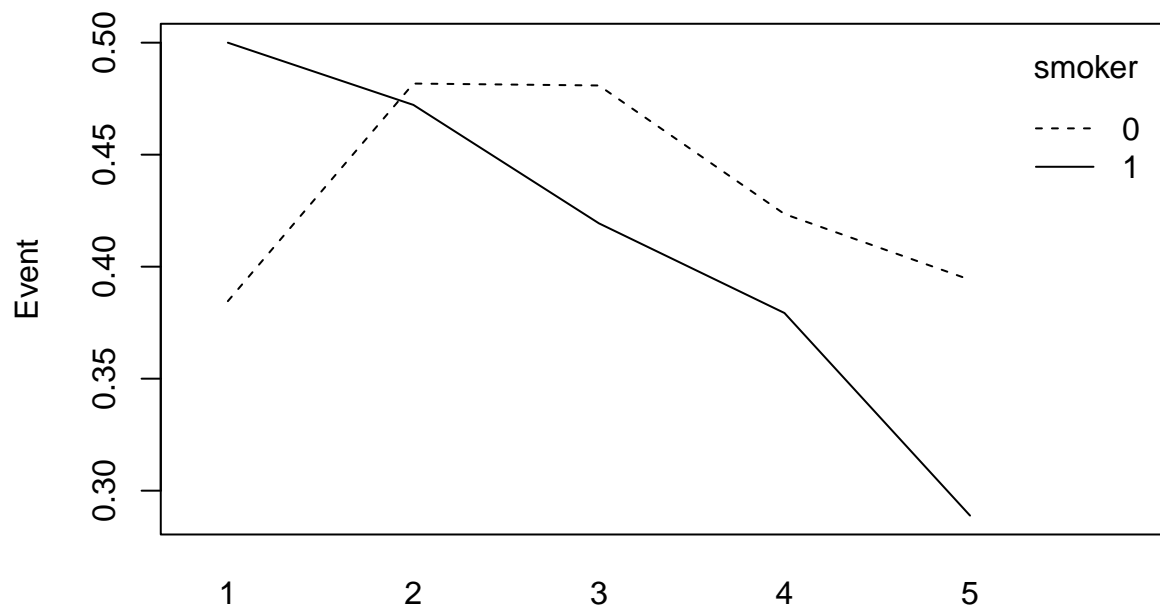


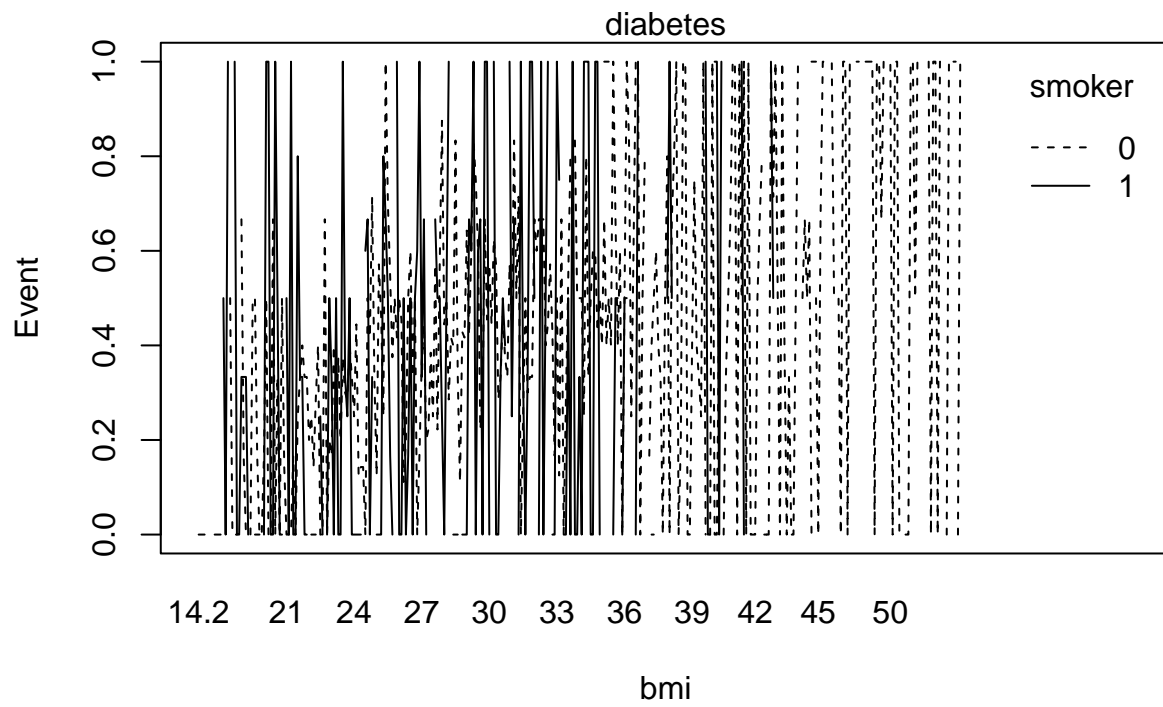
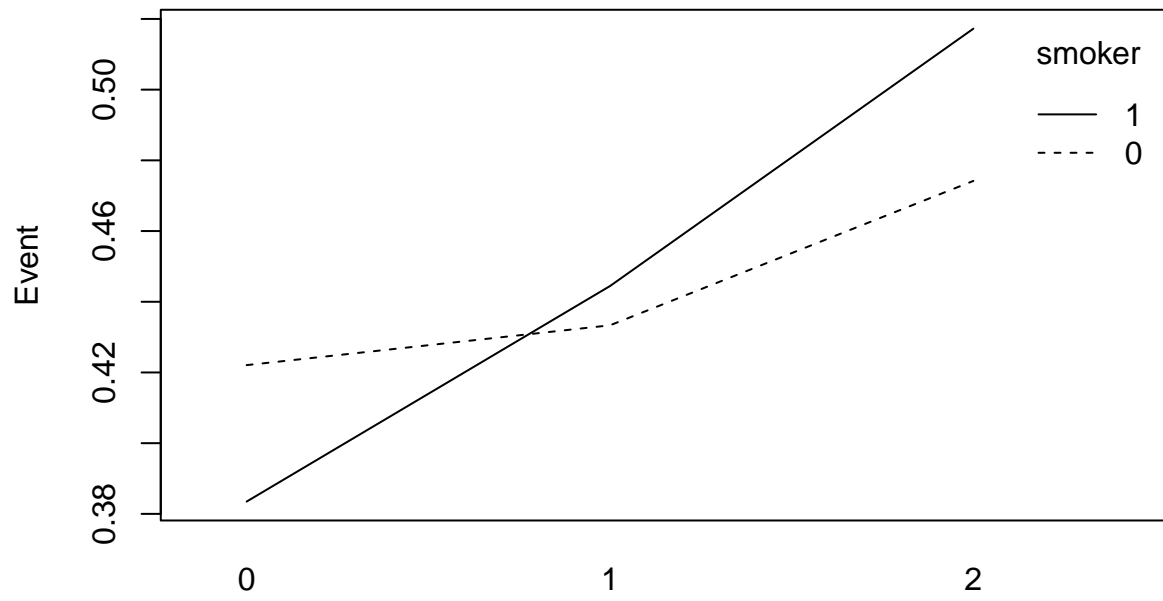


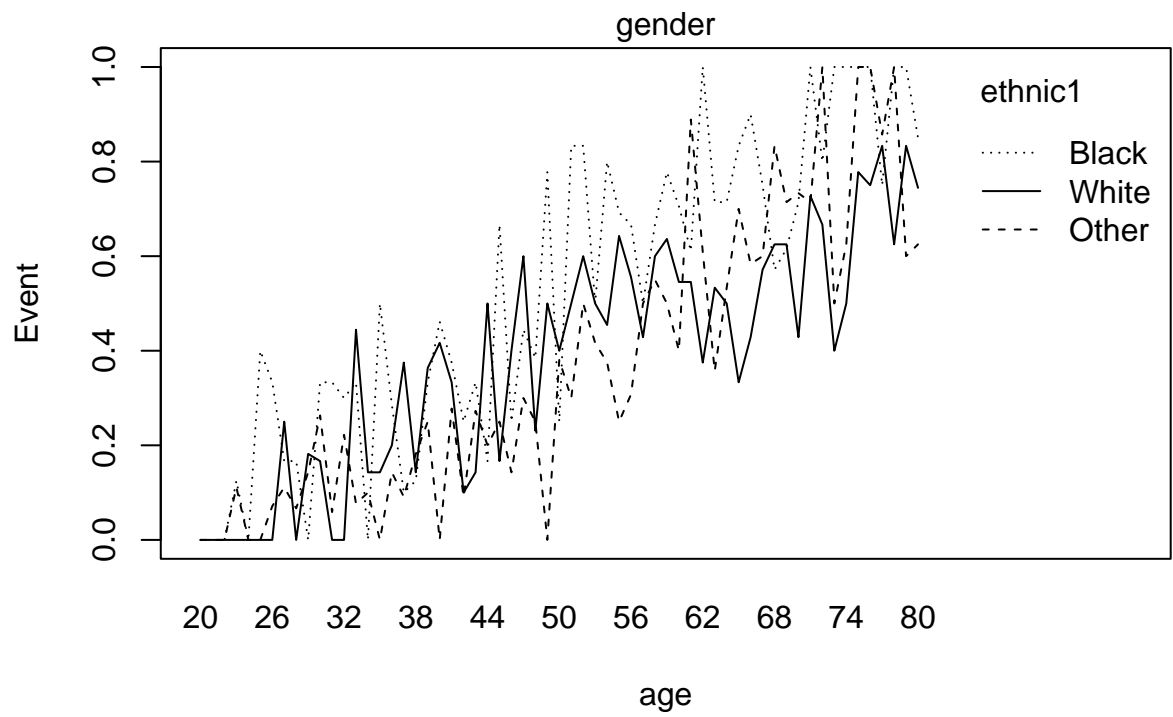
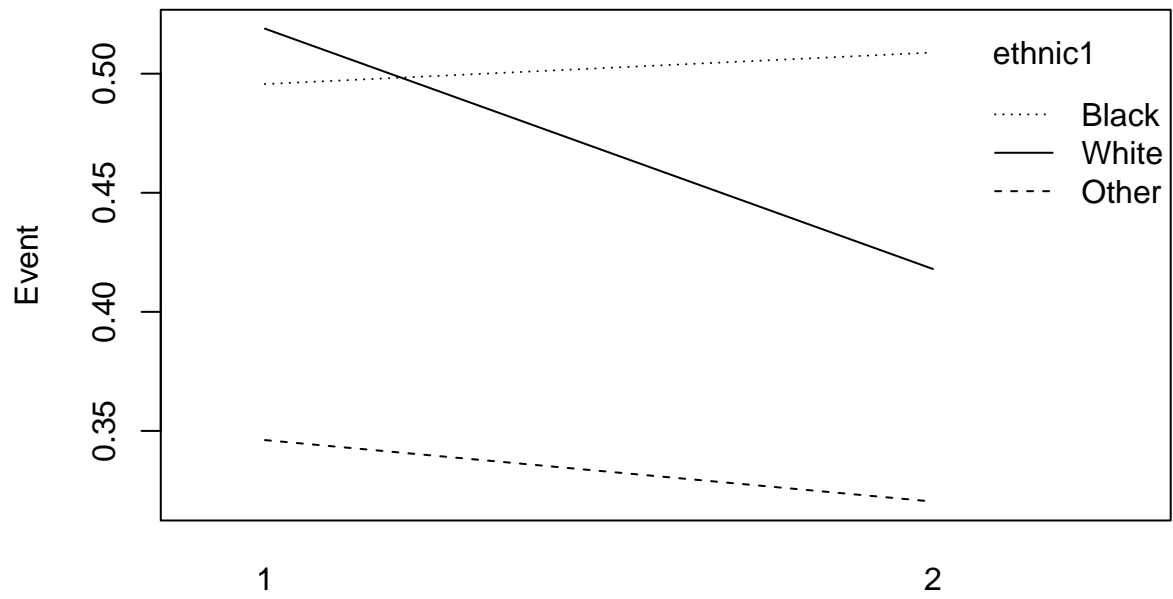


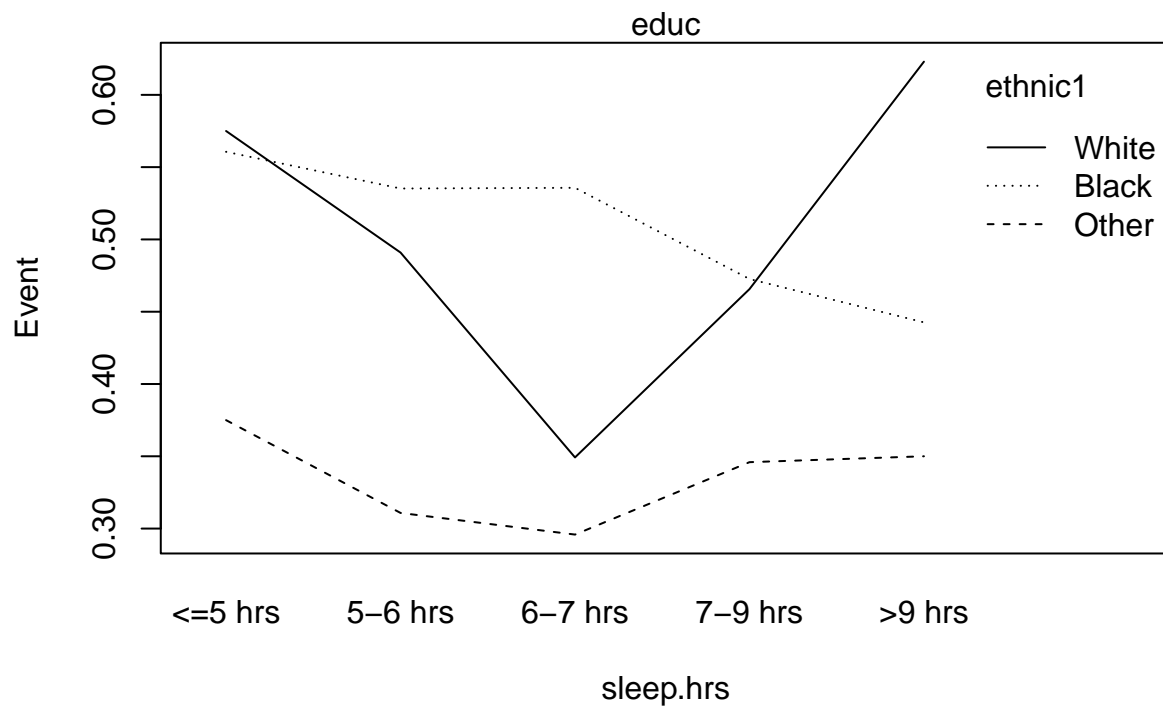
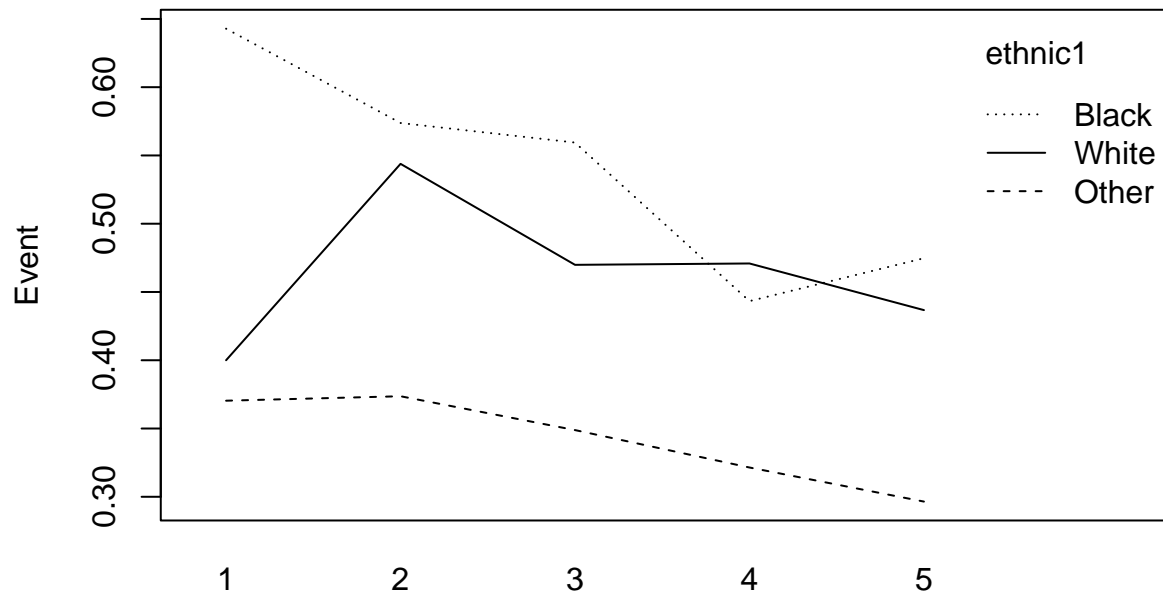


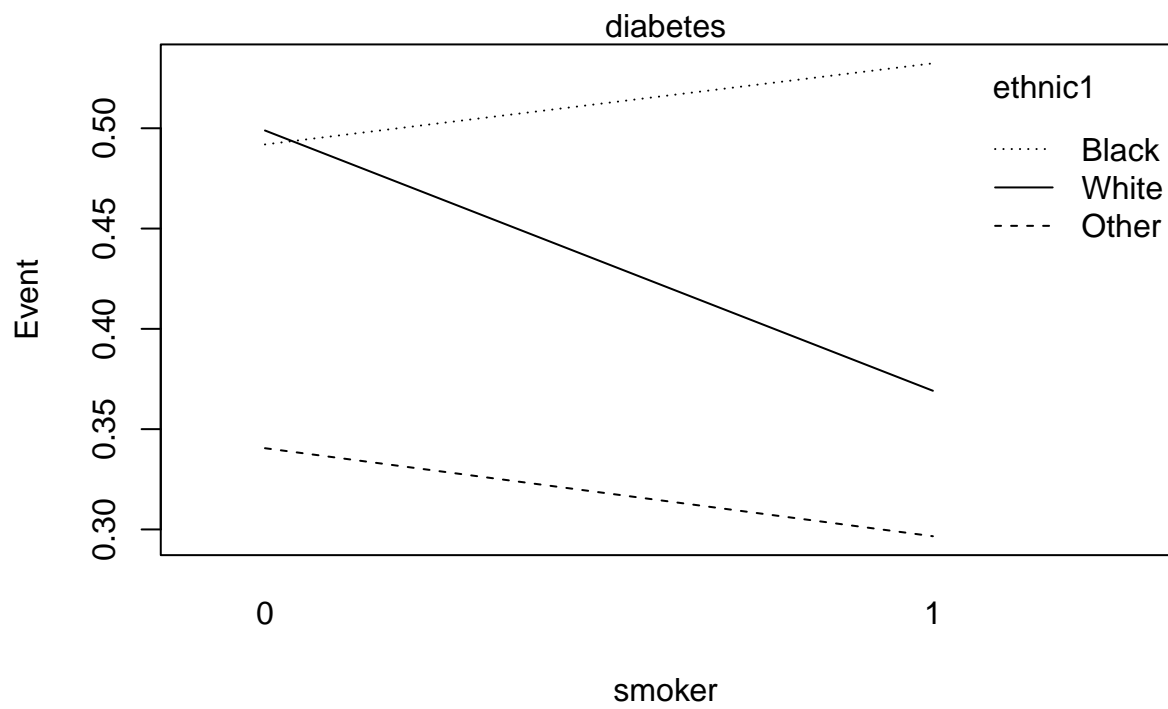
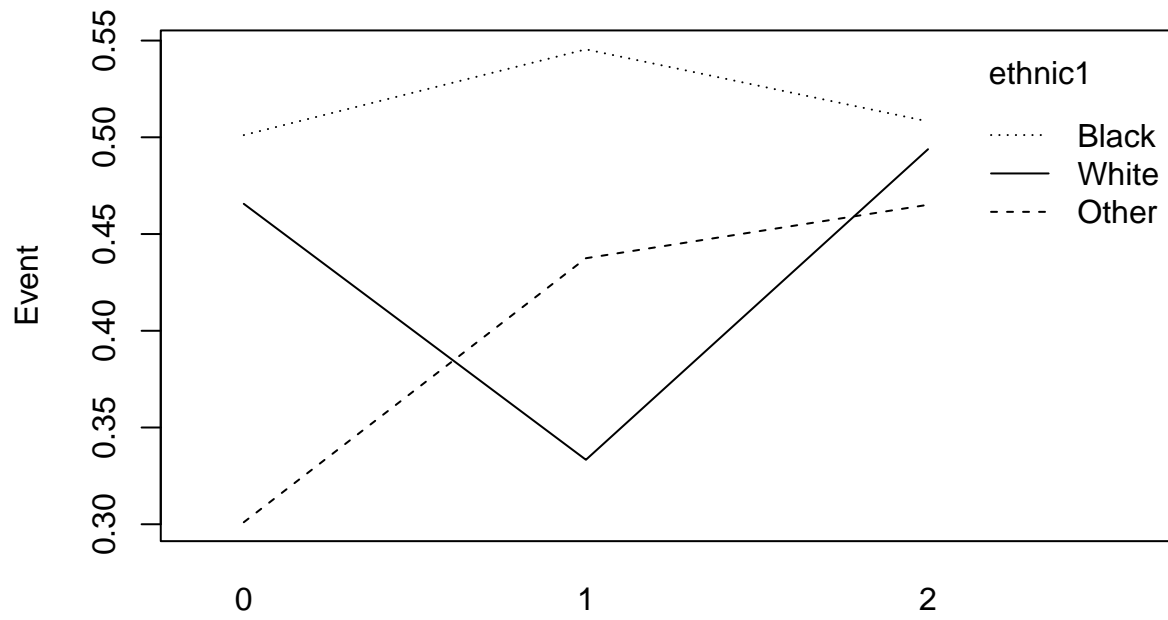


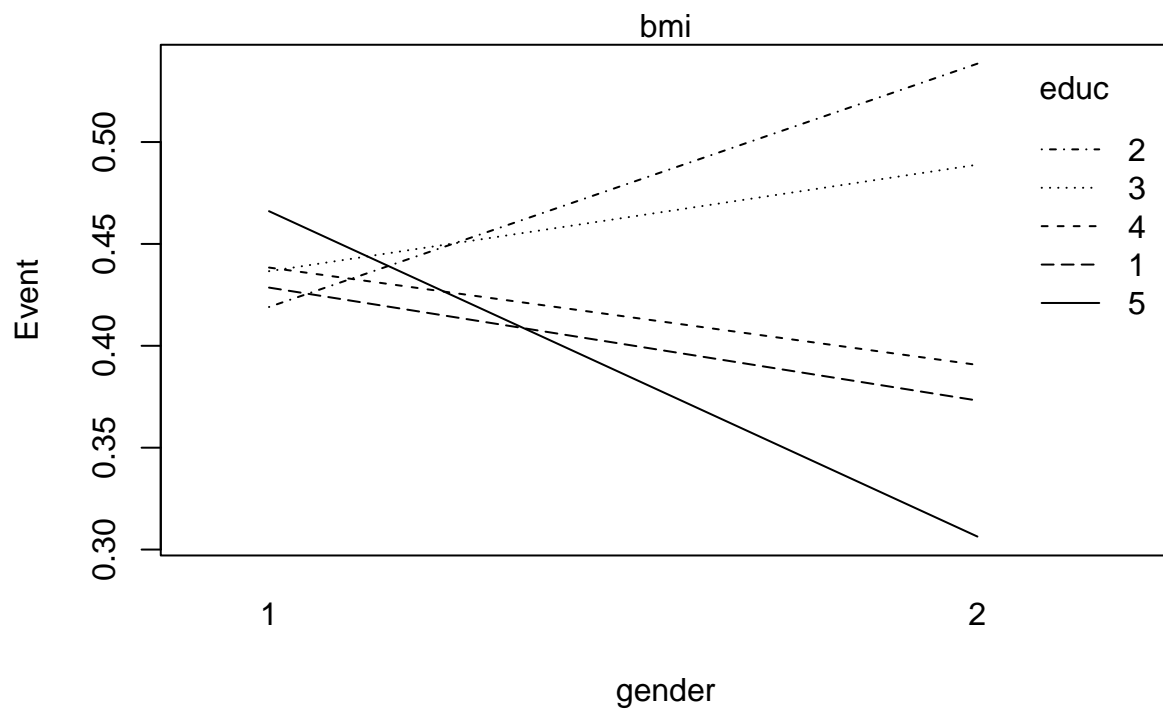
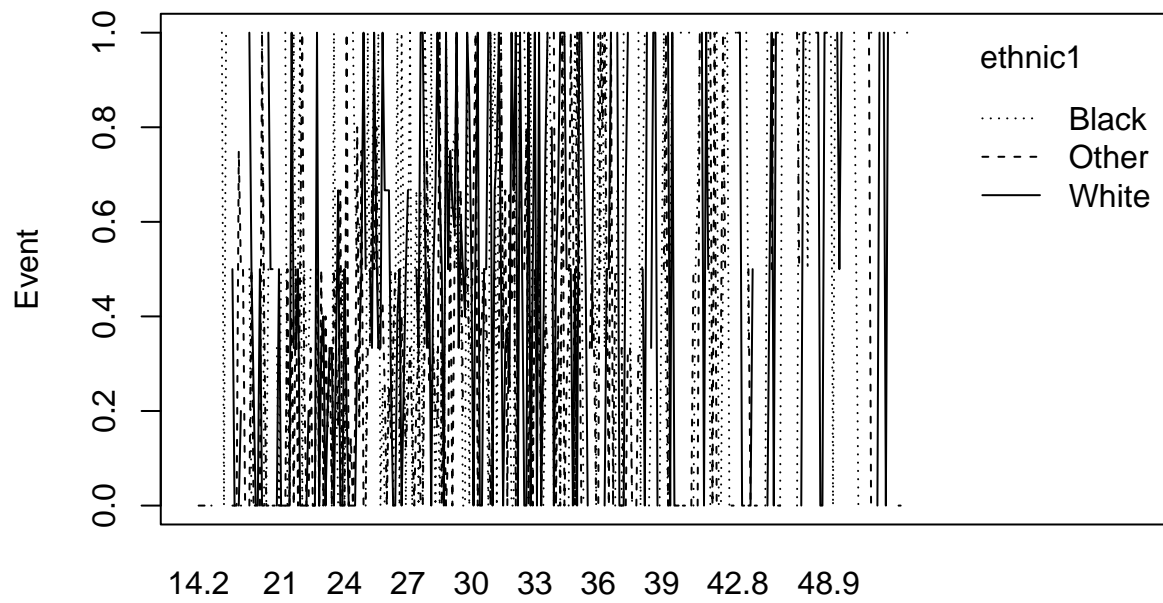


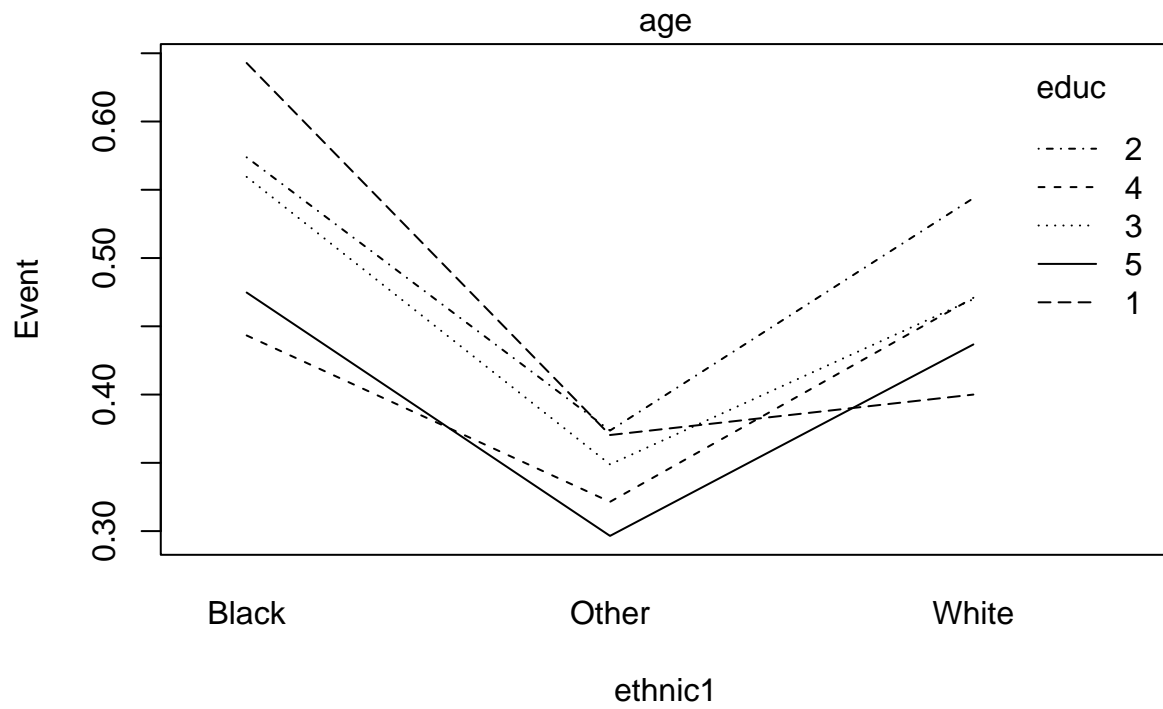
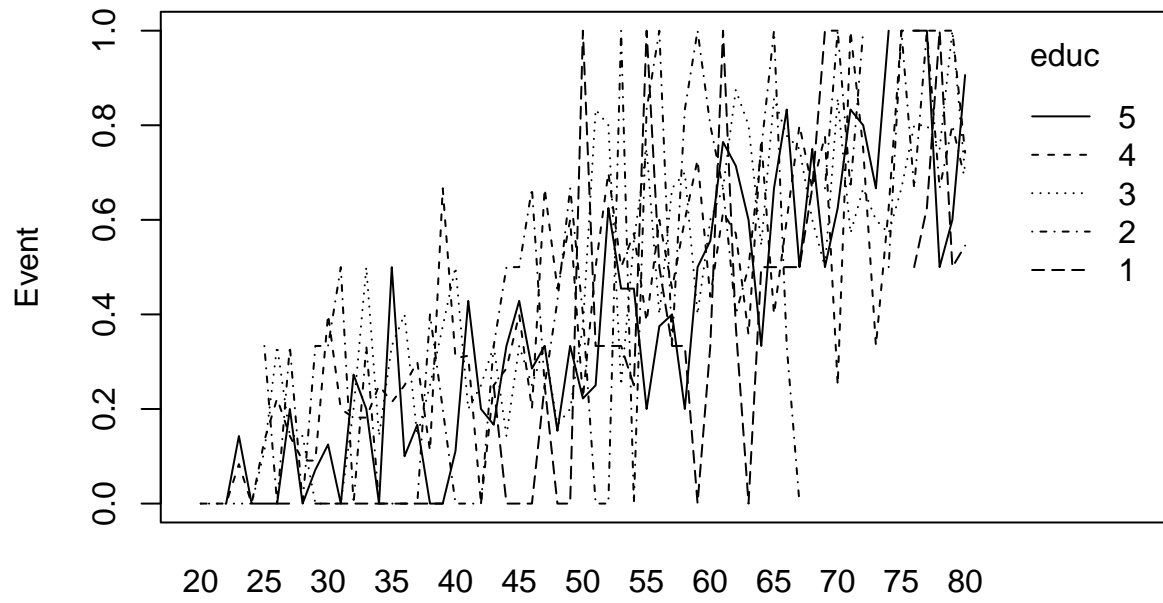


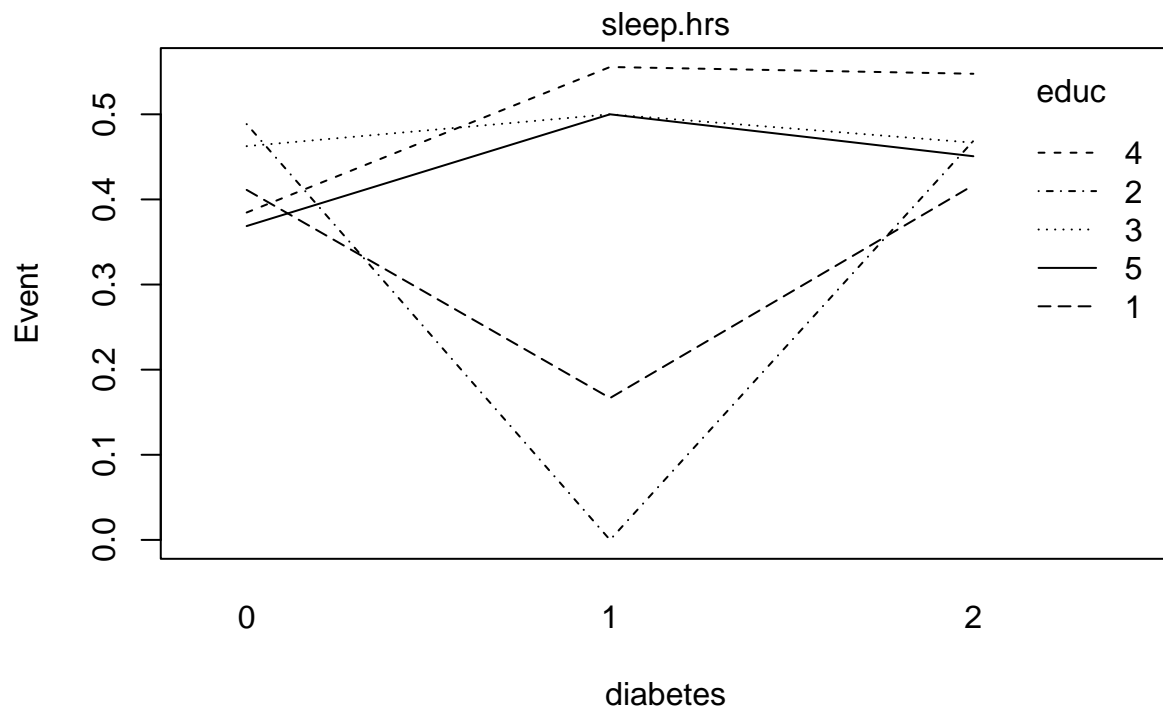
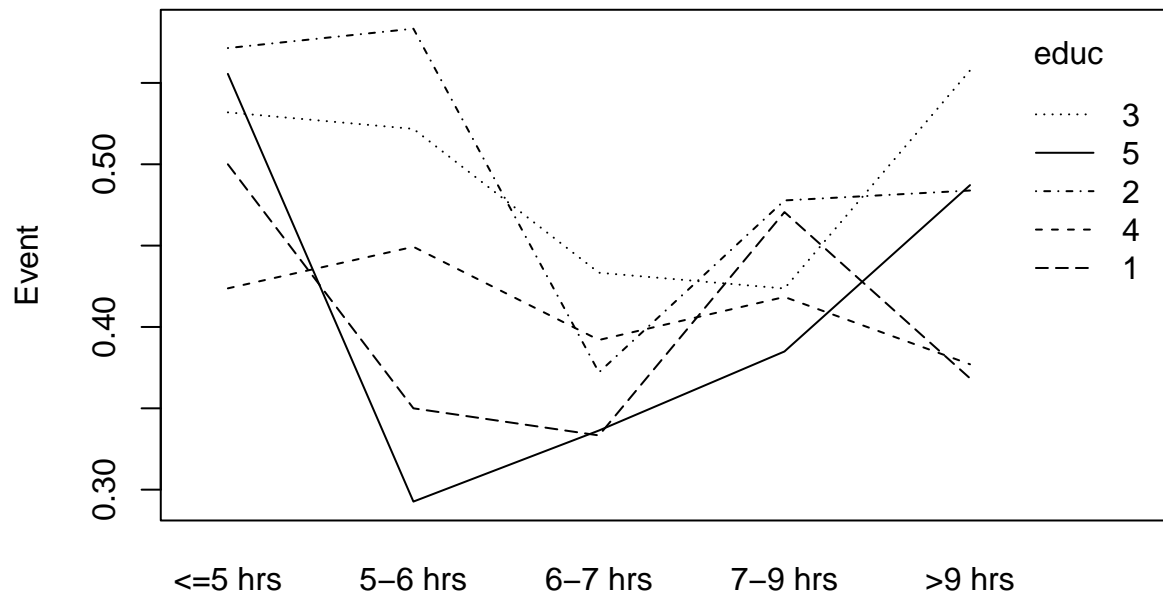


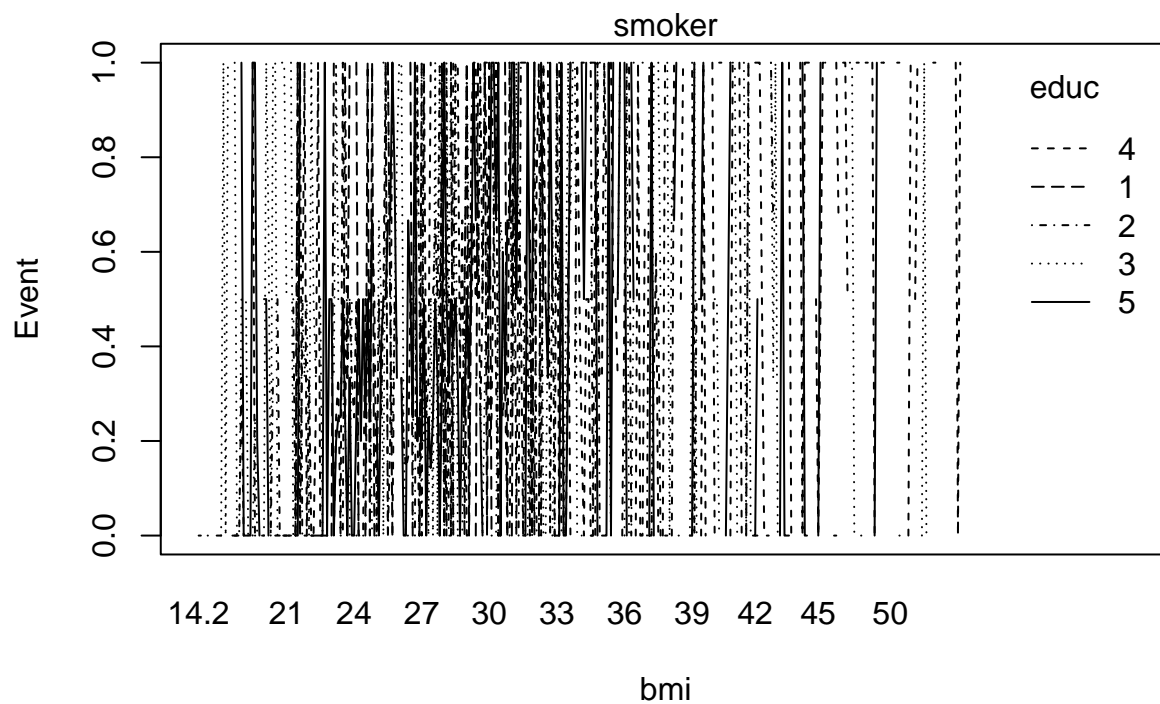
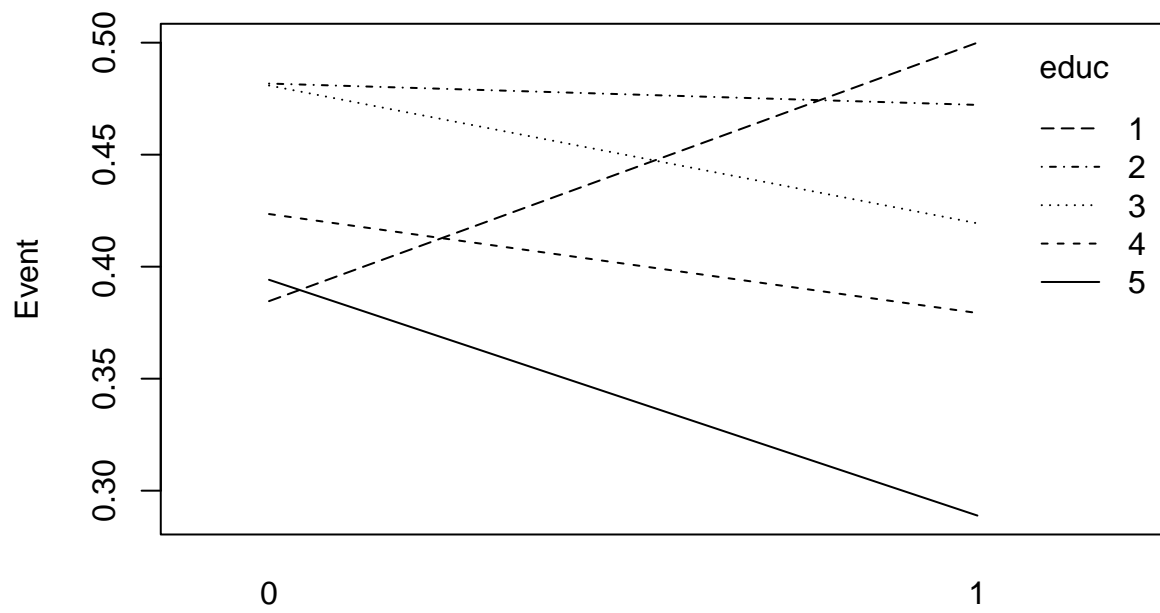












Exploring Interaction Effect Through LR Test

```
# Prettified table with signif. indicator:
predictors<-c("gender" ,"age","ethnic1","educ" ,"sleep.hrs", "diabetes" ,"smoker" ,"bmi")
lrfunction <- function(i, j, data) {
  formula_str <- paste("event ~", i, "+", j, "+", i, ":", j)
  model1 <- glm(formula_str, data = data, family = binomial)
  formula_str <- paste("event ~", i, "+", j)
  model2 <- glm(formula_str, data = data, family = binomial)
  lr_test <- lrtest(model1, model2)

  pvalue <- lr_test$Pr[2]
```

```

significance <- ifelse(pvalue < 0.01, "**", ifelse(pvalue < 0.05, "*", ""))

result <- data.frame(term1 = i, term2 = j, pvalue = pvalue, significance = paste0(round(pvalue, 3), si
return(result)
}

interaction_results <- data.frame(term1 = character(), term2 = character(), pvalue = numeric(), signifi

for (i in 1:(length(predictors))) {
  for (j in 1:(length(predictors))) {
    if (i == j) {
      pvalue <- NaN
      significance <- ""
    } else {
      result <- lrfunctor(predictors[[i]], predictors[[j]], data)
      pvalue <- result$pvalue
      significance <- result$significance
    }
    interaction_results <- rbind(interaction_results, data.frame(term1 = predictors[[i]], term2 = predi
  }
}

interaction_matrix <- reshape2::acast(interaction_results, term1 ~ term2, value.var = "significance")

interaction_matrix

```

```

##          age      bmi      diabetes educ      ethnic1 gender  sleep.hrs
## age      ""      "0.358" "0.554"  "0.154"  "0.033*" "0.834"  "0.972"
## bmi      "0.358" ""      "0.097"  "0.314"  "0.679"  "0.496"  "0.399"
## diabetes "0.554" "0.097" ""      "0.206"  "0.05*"  "0.38"   "0.74"
## educ     "0.154" "0.314" "0.206" ""      "0.849"  "0.003**" "0.729"
## ethnic1  "0.033*" "0.679" "0.05*" "0.849" ""      "0.152"  "0.102"
## gender   "0.834" "0.496" "0.38"  "0.003**" "0.152" ""      "0.184"
## sleep.hrs "0.972" "0.399" "0.74"  "0.729"  "0.102" "0.184" ""
## smoker   "0.468" "0.912" "0.293" "0.583"  "0.043*" "0.106"  "0.968"
##          smoker
## age      "0.468"
## bmi      "0.912"
## diabetes "0.293"
## educ     "0.583"
## ethnic1  "0.043*"
## gender   "0.106"
## sleep.hrs "0.968"
## smoker   ""

```

```
write.csv(interaction_matrix, ".\\interaction_result.csv", row.names=TRUE)
```

Select 4 significant interaction terms:

```

par(mfrow = c(1, 1))
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))
#options(repr.plot.width = 50, repr.plot.height = 20)

interaction.plot(

```

```

x.factor = as.factor(data$gender),
trace.factor = as.factor(data$educ),
response =as.numeric(as.character(data$event))
, type = "l", legend = TRUE,
xlab="gender",
ylab="event",
trace.label = "educ",
main="p-value: 0.002665085")

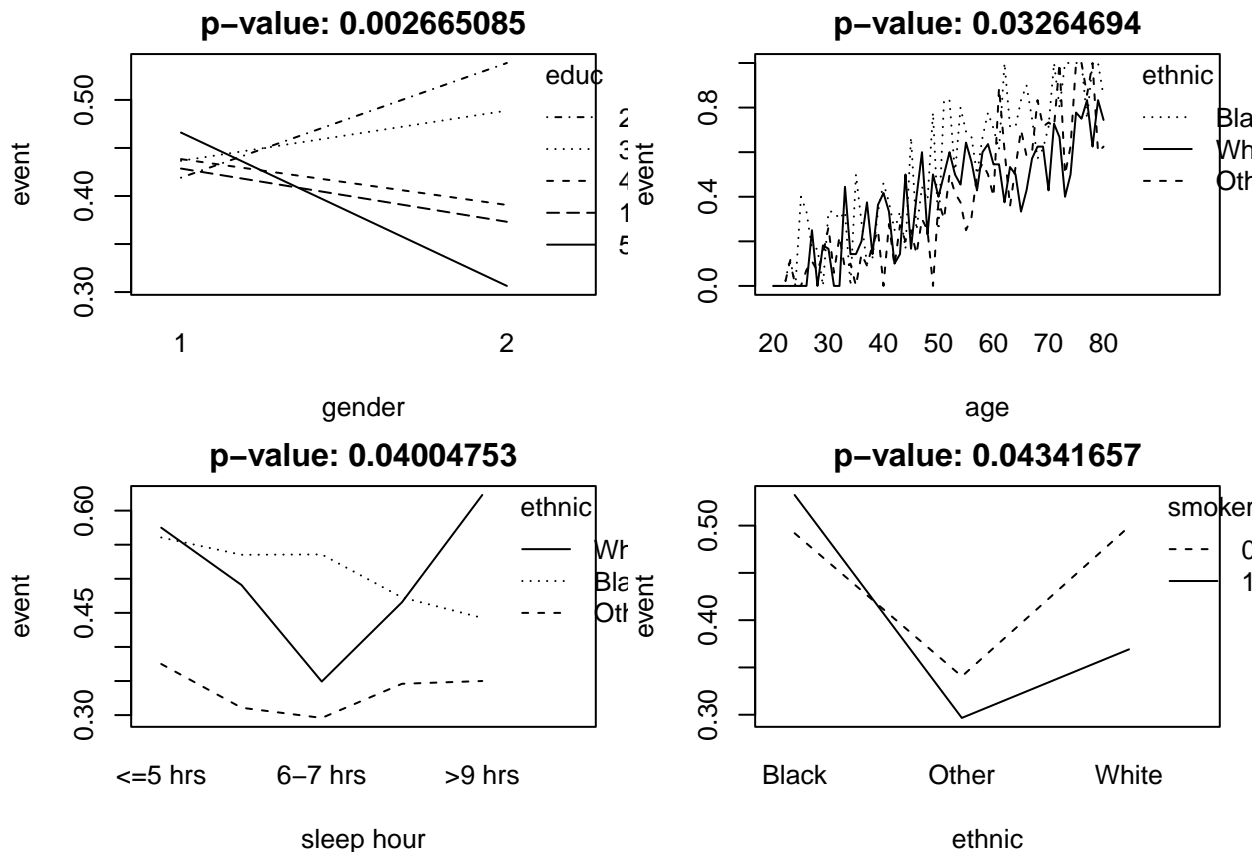
interaction.plot(
x.factor = as.factor(data$age),
trace.factor = as.factor(data$ethnic1),
response = as.numeric(as.character(data$event))
, type = "l", legend = TRUE,
xlab="age",
ylab="event",
trace.label = "ethnic",
main="p-value: 0.03264694")

interaction.plot(
x.factor = as.factor(data$sleep.hrs),
trace.factor = as.factor(data$ethnic1),
response =as.numeric(as.character(data$event))
, type = "l", legend = TRUE,
xlab="sleep hour",
ylab="event",
trace.label = "ethnic",
main="p-value: 0.04004753")

interaction.plot(
x.factor = as.factor(data$ethnic1),
trace.factor = as.factor(data$smoker),
response =as.numeric(as.character(data$event))

, type = "l", legend = TRUE,
xlab="ethnic",
ylab="event",
trace.label = "smoker",
main="p-value: 0.04341657")

```



Set a higher threshold (alpha = 0.01), so only the interaction term between gender and education is s

Step 5: Fit, Compare, Select Models

Evaluate Full Model

```
full.model <- glm(event ~. +educ*gender, data = data, family = binomial)
summary(full.model)
```

```
##
## Call:
## glm(formula = event ~ . + educ * gender, family = binomial, data = data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.978707   0.521941 -11.455 < 2e-16 ***
## gender2       -0.372400   0.402913  -0.924 0.355346
## age           0.075137   0.004129  18.197 < 2e-16 ***
## ethnic10ther  -0.495992   0.145565  -3.407 0.000656 ***
## ethnic1White  -0.638080   0.149327  -4.273 1.93e-05 ***
## educ2         0.291036   0.369643   0.787 0.431081
## educ3         0.361929   0.331783   1.091 0.275335
## educ4         0.331388   0.322932   1.026 0.304804
## educ5         0.581796   0.326390   1.783 0.074665 .
## sleep.hrs5-6 hrs -0.225793   0.249162  -0.906 0.364825
## sleep.hrs6-7 hrs -0.445424   0.220324  -2.022 0.043210 *
## sleep.hrs7-9 hrs -0.238886   0.203224  -1.175 0.239801
## sleep.hrs>9 hrs -0.285829   0.251464  -1.137 0.255681
## diabetes      -0.143095   0.075153  -1.904 0.056906 .
```

```
## smoker1          0.285674    0.142676    2.002 0.045257 *
## bmi              0.067162    0.007803    8.607 < 2e-16 ***
## gender2:educ2    0.709475    0.519313    1.366 0.171883
## gender2:educ3    0.422549    0.461811    0.915 0.360201
## gender2:educ4    0.362424    0.450508    0.804 0.421121
## gender2:educ5    -0.266648    0.463800   -0.575 0.565346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 2503.0 on 1835 degrees of freedom
## Residual deviance: 1907.2 on 1816 degrees of freedom
## AIC: 1947.2
##
## Number of Fisher Scoring iterations: 4
# AIC Backward Selection
step(full.model,direction="backward", k=2)
```

```
## Start: AIC=1947.17
## event ~ gender + age + ethnic1 + educ + sleep.hrs + diabetes +
## smoker + bmi + educ * gender
##
##           Df Deviance   AIC
## - sleep.hrs  4   1911.7 1943.7
## <none>                1907.2 1947.2
## - gender:educ  4   1915.5 1947.5
## - diabetes    1   1910.8 1948.8
## - smoker      1   1911.2 1949.2
## - ethnic1     2   1927.6 1963.6
## - bmi         1   1988.6 2026.6
## - age         1   2355.7 2393.7
##
## Step: AIC=1943.66
## event ~ gender + age + ethnic1 + educ + diabetes + smoker + bmi +
## gender:educ
##
##           Df Deviance   AIC
## - gender:educ  4   1919.7 1943.7
## <none>                1911.7 1943.7
## - diabetes    1   1915.5 1945.5
## - smoker      1   1915.8 1945.8
## - ethnic1     2   1933.1 1961.1
## - bmi         1   1994.4 2024.4
## - age         1   2363.6 2393.6
##
## Step: AIC=1943.66
## event ~ gender + age + ethnic1 + educ + diabetes + smoker + bmi
##
##           Df Deviance   AIC
## - gender     1   1921.0 1943.0
## - educ       4   1927.5 1943.5
## <none>                1919.7 1943.7
## - diabetes   1   1923.3 1945.3
```

```

## - smoker      1    1924.1 1946.1
## - ethnic1     2    1940.7 1960.7
## - bmi         1    2005.6 2027.6
## - age         1    2377.6 2399.6
##
## Step: AIC=1942.96
## event ~ age + ethnic1 + educ + diabetes + smoker + bmi
##
##           Df Deviance    AIC
## - educ      4    1928.8 1942.8
## <none>              1921.0 1943.0
## - diabetes  1    1924.5 1944.5
## - smoker    1    1926.0 1946.0
## - ethnic1   2    1941.7 1959.7
## - bmi       1    2005.8 2025.8
## - age       1    2383.6 2403.6
##
## Step: AIC=1942.79
## event ~ age + ethnic1 + diabetes + smoker + bmi
##
##           Df Deviance    AIC
## <none>              1928.8 1942.8
## - diabetes  1    1932.2 1944.2
## - smoker    1    1934.9 1946.9
## - ethnic1   2    1951.9 1961.9
## - bmi       1    2015.1 2027.1
## - age       1    2394.1 2406.1
##
## Call: glm(formula = event ~ age + ethnic1 + diabetes + smoker + bmi,
##           family = binomial, data = data)
##
## Coefficients:
## (Intercept)          age ethnic1Other ethnic1White    diabetes
##      -5.88095      0.07408     -0.56346     -0.63579     -0.13747
##      smoker1          bmi
##       0.34201      0.06741
##
## Degrees of Freedom: 1835 Total (i.e. Null); 1829 Residual
## Null Deviance:      2503
## Residual Deviance: 1929 AIC: 1943
##
## BIC Backward Selection
step(full.model,direction="backward", k=log(nrow(data)), k.out=log(nrow(data)))
##
## Start: AIC=2057.48
## event ~ gender + age + ethnic1 + educ + sleep.hrs + diabetes +
##       smoker + bmi + educ * gender
##
##           Df Deviance    AIC
## - sleep.hrs  4    1911.7 2031.9
## - gender:educ 4    1915.5 2035.8
## - diabetes   1    1910.8 2053.6
## - smoker     1    1911.2 2054.0
## <none>              1907.2 2057.5

```

```

## - ethnic1      2   1927.6 2062.9
## - bmi          1   1988.6 2131.4
## - age          1   2355.7 2498.5
##
## Step: AIC=2031.91
## event ~ gender + age + ethnic1 + educ + diabetes + smoker + bmi +
##       gender:educ
##
##           Df Deviance   AIC
## - gender:educ  4   1919.7 2009.8
## - diabetes     1   1915.5 2028.2
## - smoker       1   1915.8 2028.5
## <none>         1911.7 2031.9
## - ethnic1      2   1933.1 2038.3
## - bmi          1   1994.4 2107.1
## - age          1   2363.6 2476.3
##
## Step: AIC=2009.85
## event ~ gender + age + ethnic1 + educ + diabetes + smoker + bmi
##
##           Df Deviance   AIC
## - educ        4   1927.5 1987.7
## - gender       1   1921.0 2003.6
## - diabetes     1   1923.3 2006.0
## - smoker       1   1924.1 2006.7
## <none>         1919.7 2009.8
## - ethnic1     2   1940.7 2015.8
## - bmi          1   2005.6 2088.2
## - age          1   2377.6 2460.3
##
## Step: AIC=1987.67
## event ~ gender + age + ethnic1 + diabetes + smoker + bmi
##
##           Df Deviance   AIC
## - gender       1   1928.8 1981.4
## - diabetes     1   1931.1 1983.7
## - smoker       1   1933.0 1985.6
## <none>         1927.5 1987.7
## - ethnic1     2   1951.0 1996.0
## - bmi          1   2015.0 2067.6
## - age          1   2388.1 2440.7
##
## Step: AIC=1981.39
## event ~ age + ethnic1 + diabetes + smoker + bmi
##
##           Df Deviance   AIC
## - diabetes     1   1932.2 1977.3
## - smoker       1   1934.9 1980.0
## <none>         1928.8 1981.4
## - ethnic1     2   1951.9 1989.4
## - bmi          1   2015.1 2060.2
## - age          1   2394.1 2439.2
##
## Step: AIC=1977.31

```



```
## event ~ age + ethnic1 + smoker + bmi
##
##           Df Deviance    AIC
## - smoker   1   1938.0 1975.6
## <none>           1932.2 1977.3
## - ethnic1   2   1956.0 1986.1
## - bmi       1   2017.7 2055.3
## - age       1   2400.3 2437.8
##
## Step:  AIC=1975.6
## event ~ age + ethnic1 + bmi
##
##           Df Deviance    AIC
## <none>           1938.0 1975.6
## - ethnic1   2   1964.0 1986.5
## - bmi       1   2020.1 2050.2
## - age       1   2401.6 2431.6
##
## Call:  glm(formula = event ~ age + ethnic1 + bmi, family = binomial,
##           data = data)
##
## Coefficients:
## (Intercept)          age ethnic1Other ethnic1White          bmi
##      -5.60878       0.07119      -0.63061      -0.63284       0.06523
##
## Degrees of Freedom: 1835 Total (i.e. Null);  1831 Residual
## Null Deviance:      2503
## Residual Deviance: 1938  AIC: 1948
```

Final model(final.model):age+ethnic+smoker+bmi:

$$\begin{aligned}\text{logit}(\pi_E) &= \ln \left(\frac{\mathbb{P}(\text{event})}{1 - \mathbb{P}(\text{event})} \right) \\ &= \beta_0 + \beta_{\text{age}} \text{age} + \beta_{\text{Ethnic:Other}} \text{Ethnic:Other} + \beta_{\text{Ethnic:White}} \text{Ethnic:White} \\ &\quad + \beta_{\text{Ethnic:White}} \text{Ethnic:White} + \beta_{\text{Smoker}} \text{Smoker} + \beta_{\text{BMI}} \text{BMI}\end{aligned}$$

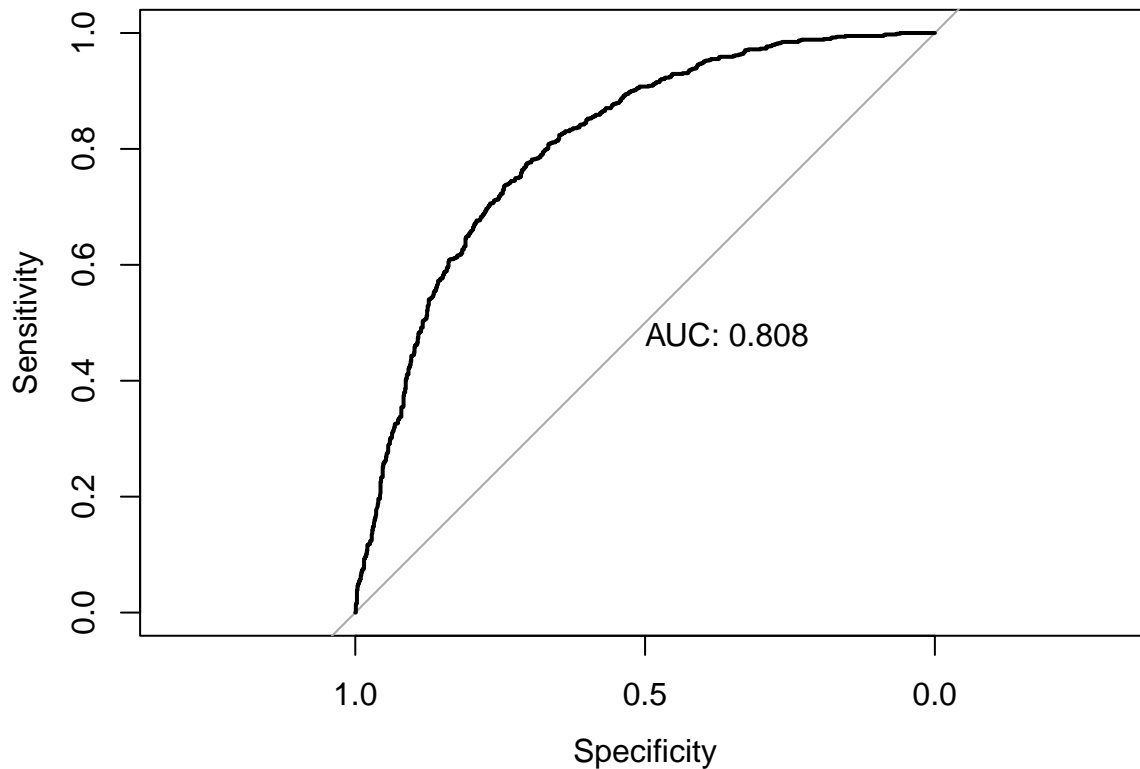
Step 6: Assess the model's overall fit

ROC Curve

```
## ROC curve:
final.model<-glm(event~age+ethnic1+smoker+bmi,data,family=binomial)
probs <- predict(final.model
                  , type = "response")
roc <- roc(data$event, probs)

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

```
plot(roc, print.auc = TRUE)
```



Confusion matrix

```
# install.packages(caret)
library(caret)
predicted_classes <- ifelse(probs > 0.5, 1, 0)
confusion_matrix <- caret::confusionMatrix(factor(data$event), factor(predicted_classes))

# Full results:
# print(confusion_matrix)

# Table only:
confusion_matrix$table
```

```
##           Reference
## Prediction    0    1
##           0 833 224
##           1 252 527
```

Probability plot:

```
## Age+smoker
mean_bmi <- mean(data$bmi)
mean_age <- mean(data$age)

plot_data <- expand.grid(
  age = seq(min(data$age), max(data$age), length.out = 100), # Adjust the length.out as needed
```

```

    smoker = as.factor(c(0, 1)),
    ethnic1= "Other",
    bmi = mean_bmi
  )

plot_data$predicted_prob <- predict(final.model, newdata = plot_data, type = "response")

plot.as<-ggplot(plot_data, aes(x = age, y = predicted_prob, color = factor(smoker))) +
  geom_line() +
  labs(title = "Probability Plot (age:smoker)") +
  scale_color_manual(values = c("blue", "red"))

## Age+ethnic
plot_data <- expand.grid(
  age = seq(min(data$age), max(data$age), length.out = 100),
  smoker = as.factor(0),
  ethnic1= as.factor(c("Black", "Other", "White" )),
  bmi = mean_bmi
)

plot_data$predicted_prob <- predict(final.model, newdata = plot_data, type = "response")
plot.ae<-ggplot(plot_data, aes(x = age, y = predicted_prob, color = factor(ethnic1))) +
  geom_line() +
  labs(title = "Probability Plot (age:ethnicity)") +
  scale_color_manual(values = c("black", "red", "blue"))

## bmi+smoker
plot_data <- expand.grid(
  bmi = seq(min(data$bmi), max(data$bmi), length.out = 100), # Adjust the length.out as needed
  smoker = as.factor(c(0, 1)),
  ethnic1= "Other",
  age = mean_age
)

plot_data$predicted_prob <- predict(final.model, newdata = plot_data, type = "response")

plot.bs<-ggplot(plot_data, aes(x = bmi, y = predicted_prob, color = factor(smoker))) +
  geom_line() +
  labs(title = "Probability Plot (bmi:smoker)") +
  scale_color_manual(values = c("blue", "red"))

## bmi+ethnic
plot_data <- expand.grid(
  bmi = seq(min(data$bmi), max(data$bmi), length.out = 100),
  smoker = as.factor(0),
  ethnic1= as.factor(c("Black", "Other", "White" )),
  age = mean_age
)

plot_data$predicted_prob <- predict(final.model, newdata = plot_data, type = "response")
plot.be<-ggplot(plot_data, aes(x = bmi, y = predicted_prob, color = factor(ethnic1))) +
  geom_line() +
  labs(title = "Probability Plot (bmi:ethnicity)") +

```

```
scale_color_manual(values = c("black", "red", "blue"))
```

```
combined_plot_h1 <- plot.as + plot.ae
```

```
combined_plot_h2 <- plot.bs + plot.be
```

```
combined_plot_h1/combined_plot_h2
```

