Project1.Rmd

Dizhou Wu

2023-09-29

Project 1

```
# Load library
#library(MASS)
library(ROCR)
library(ggplot2)
#suppressMessages(library(tidyverse))
library(foreign)
library(statmod)
library(dplyr)
```

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

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

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

```
library(caret)
```

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

```
suppressMessages(library(car))
```

```
# Load data
source("http://www.openintro.org/stat/data/cdc.R")
```

```
# Save as .txt file
write.table(cdc, file = "cdc.txt", sep = "\t", row.names = FALSE)
```

```
#cdc
```

```r
cdc$wl <- cdc$weight - cdc$wtdesire
#cdc$wl <- sign(cdc$wl)*sqrt(abs(cdc$wl))
```

```r
# Fit a logistic regression model
m1 <- glm(exerany ~ wl + age + genhlth + hlthplan, data = cdc, family = binomial)
summary(m1)
```

```
##
## Call:
## glm(formula = exerany ~ wl + age + genhlth + hlthplan, family = binomial,
##     data = cdc)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.6218912  0.0663725  24.436  < 2e-16 ***
## wl                -0.0061747  0.0006754  -9.142  < 2e-16 ***
## age               -0.0082959  0.0010178  -8.151 3.62e-16 ***
## genhlthvery good -0.1609750  0.0501957  -3.207  0.00134 **
## genhlthgood       -0.6927566  0.0498977 -13.884  < 2e-16 ***
## genhlthfair       -1.1298026  0.0619783 -18.229  < 2e-16 ***
## genhlthpoor       -1.6744570  0.0896002 -18.688  < 2e-16 ***
## hlthplan           0.4683470  0.0489173   9.574  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22680  on 19999  degrees of freedom
## Residual deviance: 21466  on 19992  degrees of freedom
## AIC: 21482
##
## Number of Fisher Scoring iterations: 4
```

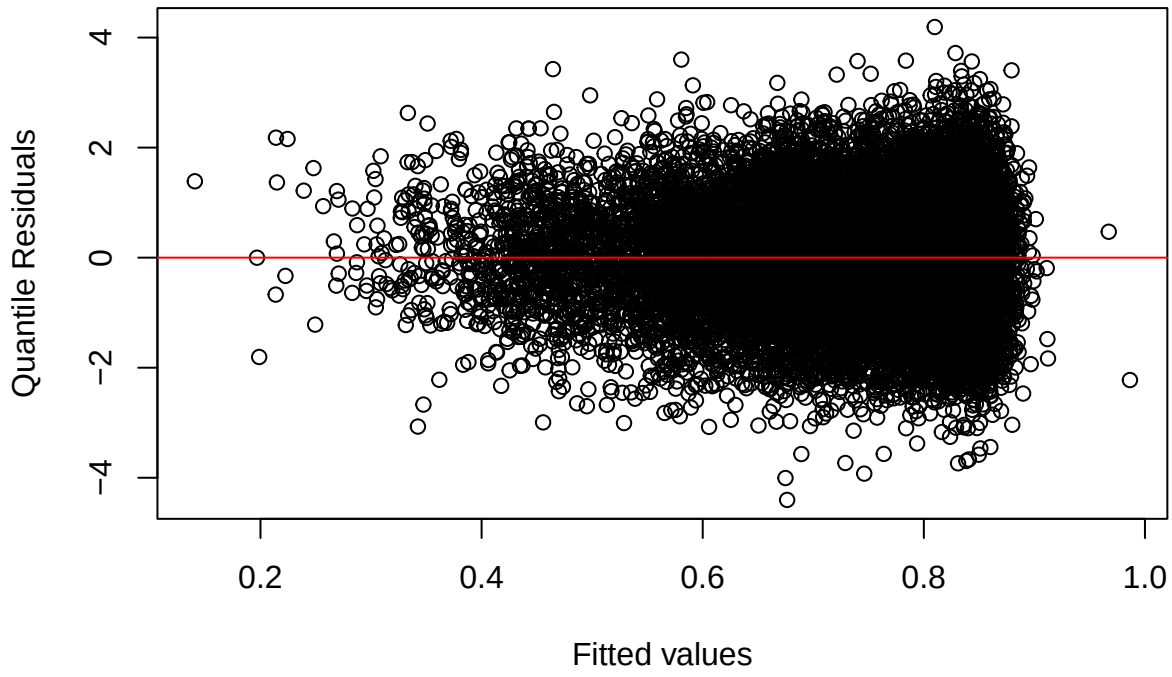
```r
# 95% Confidence Interval
conf_int <- confint(m1, level = 0.95)
```

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

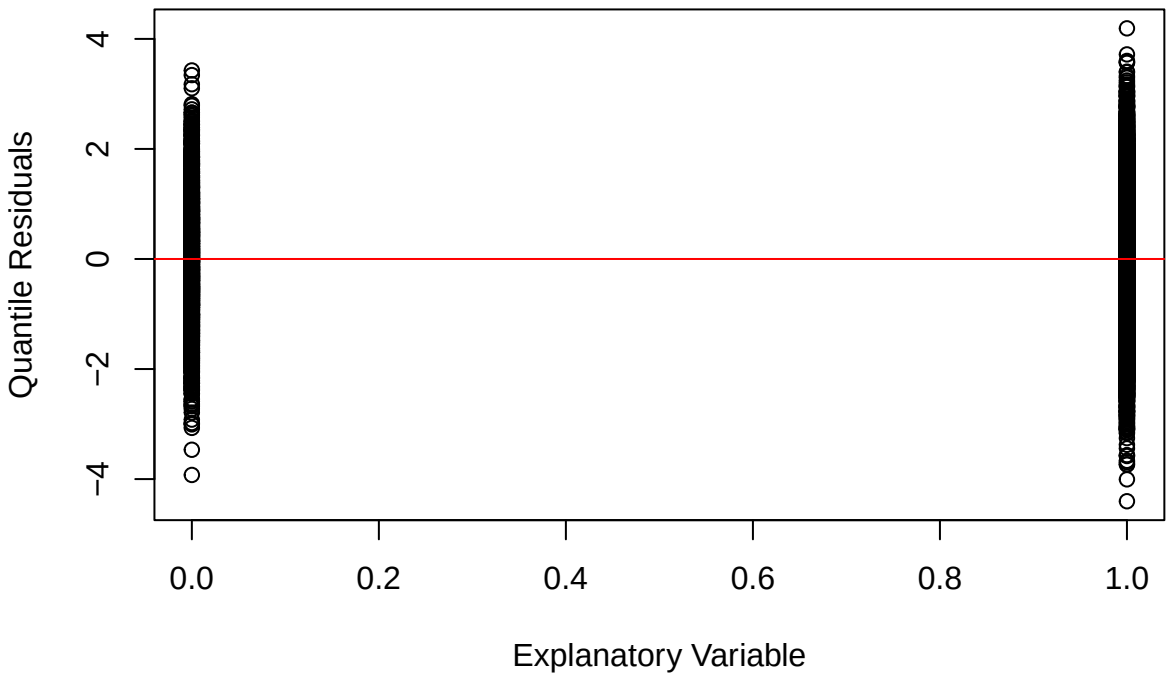
```r
print(conf_int)
```

```
##                         2.5 %       97.5 %
## (Intercept)        1.492288250  1.752481666
## wl                -0.007499645 -0.004851634
## age               -0.010290635 -0.006300603
## genhlthvery good -0.259653256 -0.062865912
## genhlthgood       -0.790905251 -0.595288642
## genhlthfair       -1.251430951 -1.008455748
## genhlthpoor       -1.850598553 -1.499270449
## hlthplan           0.372195287  0.563967309
```

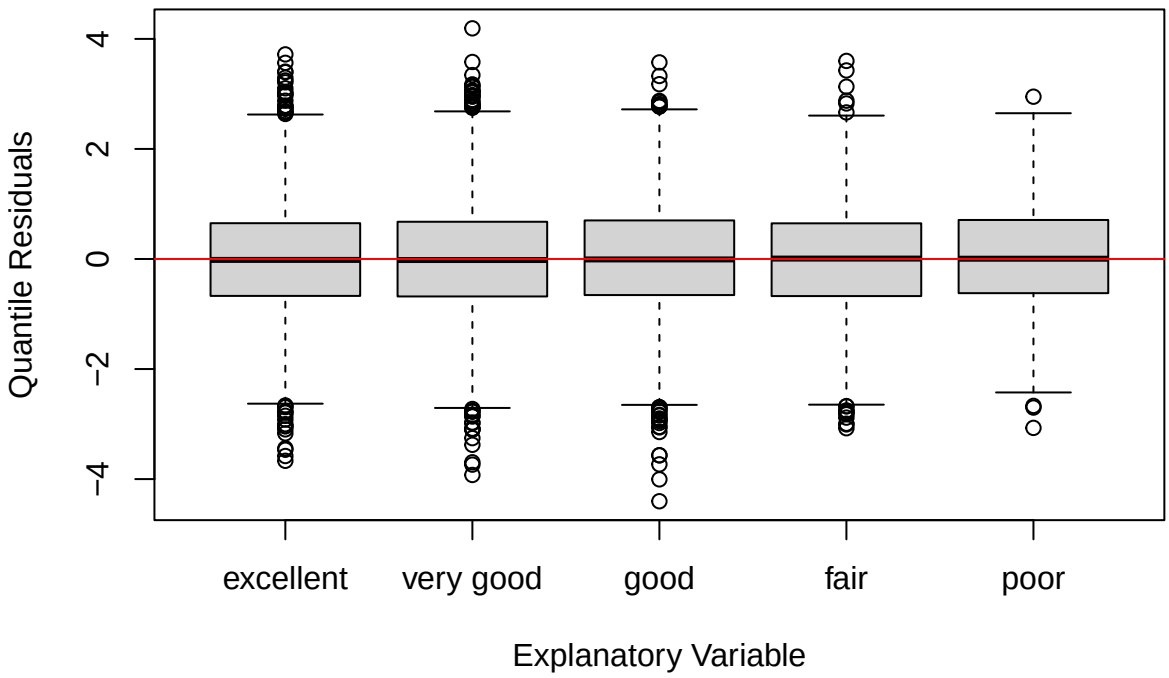
```r
# Quantile residual plot
qresiduals <- qresid(m1)
plot(m1$fitted.values, qresiduals, xlab="Fitted values", ylab="Quantile Residuals")
abline(h = 0, col = "red")
```



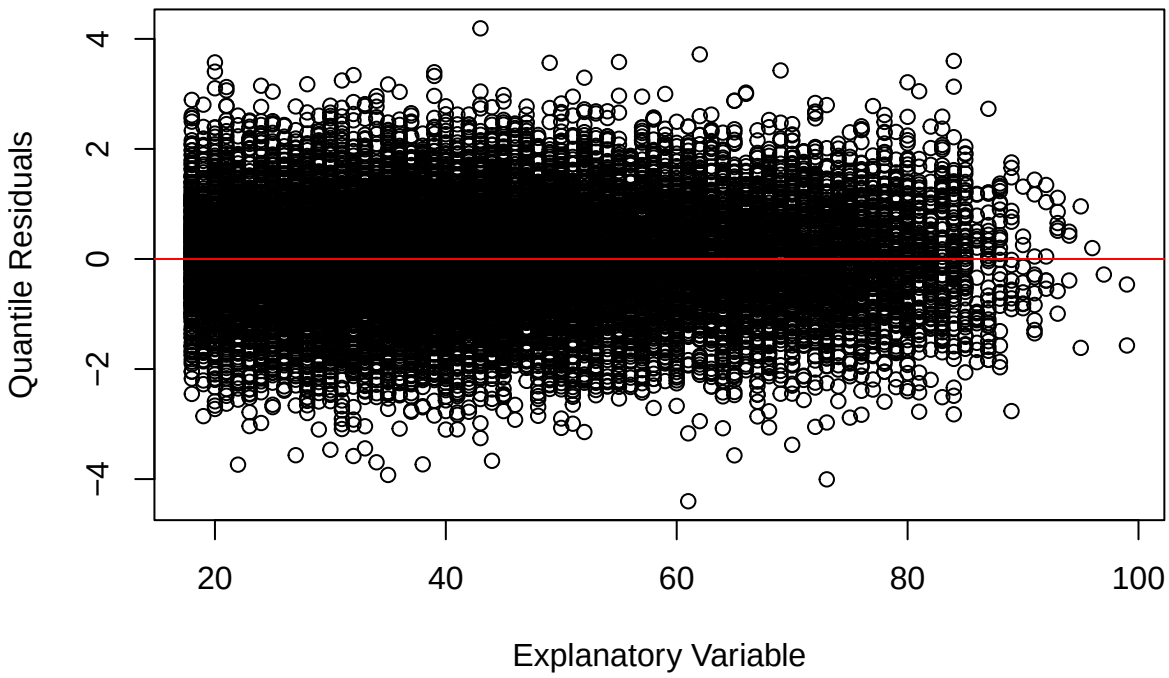

```r
# Assuming 'explanatory_var' is your explanatory variable of interest
plot(cdc$hlthplan, qresiduals, xlab="Explanatory Variable", ylab="Quantile Residuals")
abline(h = 0, col = "red")
```



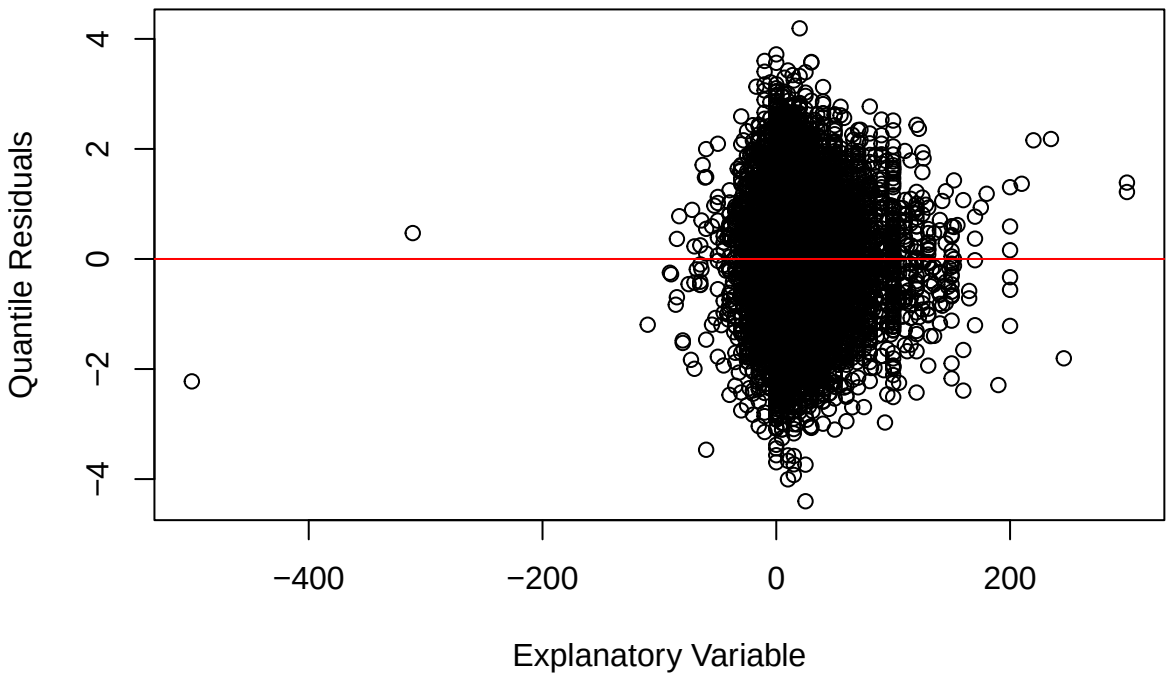

```
# Assuming 'explanatory_var' is your explanatory variable of interest
plot(cdc$genhlth, qresiduals, xlab="Explanatory Variable", ylab="Quantile Residuals")
abline(h = 0, col = "red")
```



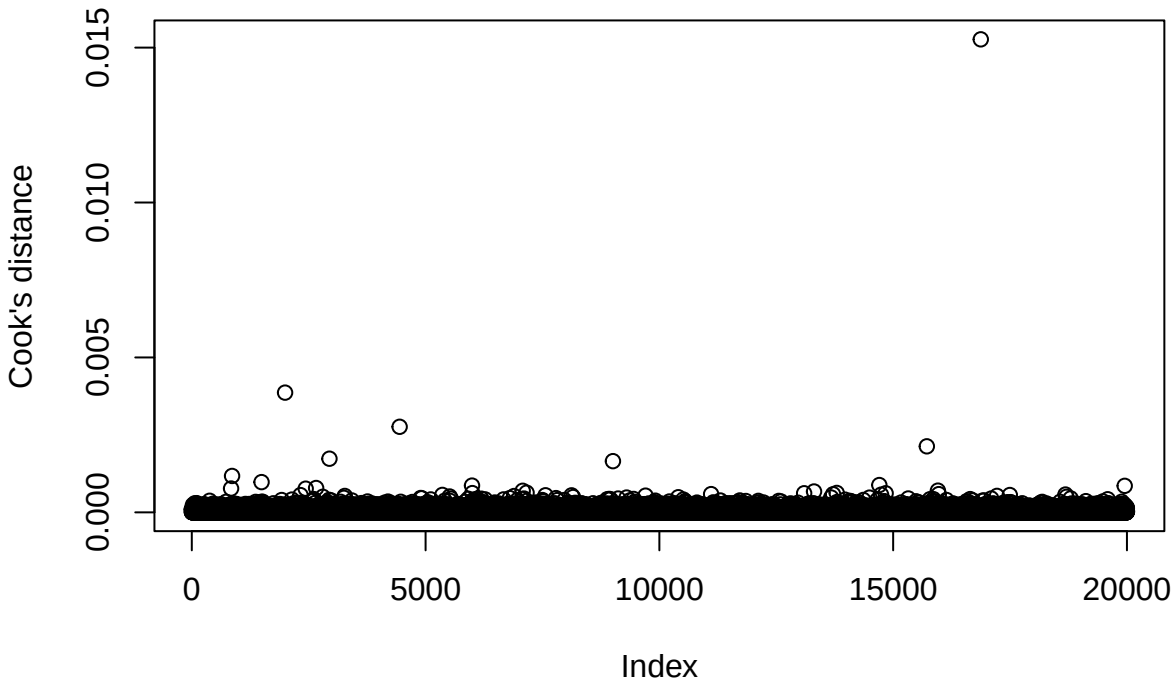

```
# Assuming 'explanatory_var' is your explanatory variable of interest
plot(cdc$age, qresiduals, xlab="Explanatory Variable", ylab="Quantile Residuals")
abline(h = 0, col = "red")
```




```
# Assuming 'explanatory_var' is your explanatory variable of interest
plot(cdc$wl, qresiduals, xlab="Explanatory Variable", ylab="Quantile Residuals")
abline(h = 0, col = "red")
```




```
# Cook's distance
cooks_d <- cooks.distance(m1)
plot(cooks_d, ylab="Cook's distance", xlab="Index")
```




```
#abline(h=c(0.5, 1), col="red")
```

```
# Variance inflation factor
vif(m1)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## wl       1.024285  1        1.012070
## age      1.126731  1        1.061476
## genhlth  1.095355  4        1.011450
## hlthplan 1.075021  1        1.036832
```

```
# Load library
library(MASS)
```

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

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

```
# Specify the starting model
m2 <- glm(exerany ~ wl + age + genhlth + hlthplan + smoke100 + gender + height, data = cdc, family = bi
summary(m2)
```

```
##
## Call:
## glm(formula = exerany ~ wl + age + genhlth + hlthplan + smoke100 +
##     gender + height, family = binomial, data = cdc)
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.2100482  0.4236023  -5.217 1.82e-07 ***
## wl               -0.0059376  0.0006876  -8.636  < 2e-16 ***
## age              -0.0068055  0.0010297  -6.609 3.86e-11 ***
## genhlthvery good -0.1555946  0.0504041  -3.087  0.00202 **
## genhlthgood      -0.6741612  0.0502245 -13.423  < 2e-16 ***
## genhlthfair      -1.0859501  0.0624742 -17.382  < 2e-16 ***
## genhlthpoor      -1.6336086  0.0903793 -18.075  < 2e-16 ***
## hlthplan          0.4369195  0.0492999   8.862  < 2e-16 ***
## smoke100         -0.0473941  0.0343428  -1.380  0.16758
## genderf           0.0840106  0.0486905   1.725  0.08446 .
## height            0.0560337  0.0059390   9.435  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22680  on 19999  degrees of freedom
## Residual deviance: 21327  on 19989  degrees of freedom
## AIC: 21349
##
## Number of Fisher Scoring iterations: 4
```

```
# 95% Confidence Interval
conf_int <- confint(m2, level = 0.95)
```

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

```
print(conf_int)
```

```
##                        2.5 %       97.5 %
## (Intercept)      -3.040829288 -1.380239461
## wl               -0.007285977 -0.004590464
## age              -0.008823302 -0.004786797
## genhlthvery good -0.254677552 -0.057073668
## genhlthgood      -0.772944283 -0.576047062
## genhlthfair      -1.208544966 -0.963626113
## genhlthpoor      -1.811270723 -1.456890335
## hlthplan          0.340013917  0.533285435
## smoke100         -0.114689293  0.019937429
## genderf          -0.011415662  0.179457563
## height            0.044406577  0.067688222
```

```
# Backward selection using BIC
backward_bic <- stepAIC(m2, scope = ~ .,
                        direction = "backward",
```

```
                      trace = 0, k = log(nrow(cdc)))

summary(backward_bic)
```

```
##
## Call:
## glm(formula = exerany ~ wl + age + genhlth + hlthplan + height,
##     family = binomial, data = cdc)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.6485683  0.2907484  -5.670 1.43e-08 ***
## wl               -0.0057545  0.0006798  -8.465  < 2e-16 ***
## age              -0.0070308  0.0010241  -6.865 6.63e-12 ***
## genhlthvery good -0.1597858  0.0503345  -3.174   0.0015 **
## genhlthgood      -0.6828178  0.0500515 -13.642  < 2e-16 ***
## genhlthfair      -1.0978963  0.0622297 -17.643  < 2e-16 ***
## genhlthpoor      -1.6491836  0.0899645 -18.331  < 2e-16 ***
## hlthplan          0.4444071  0.0491842   9.036  < 2e-16 ***
## height            0.0480878  0.0041773  11.512  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22680  on 19999  degrees of freedom
## Residual deviance: 21332  on 19991  degrees of freedom
## AIC: 21350
##
## Number of Fisher Scoring iterations: 4
```

```
# 95% Confidence Interval
conf_int <- confint(backward_bic, level = 0.95)
```

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

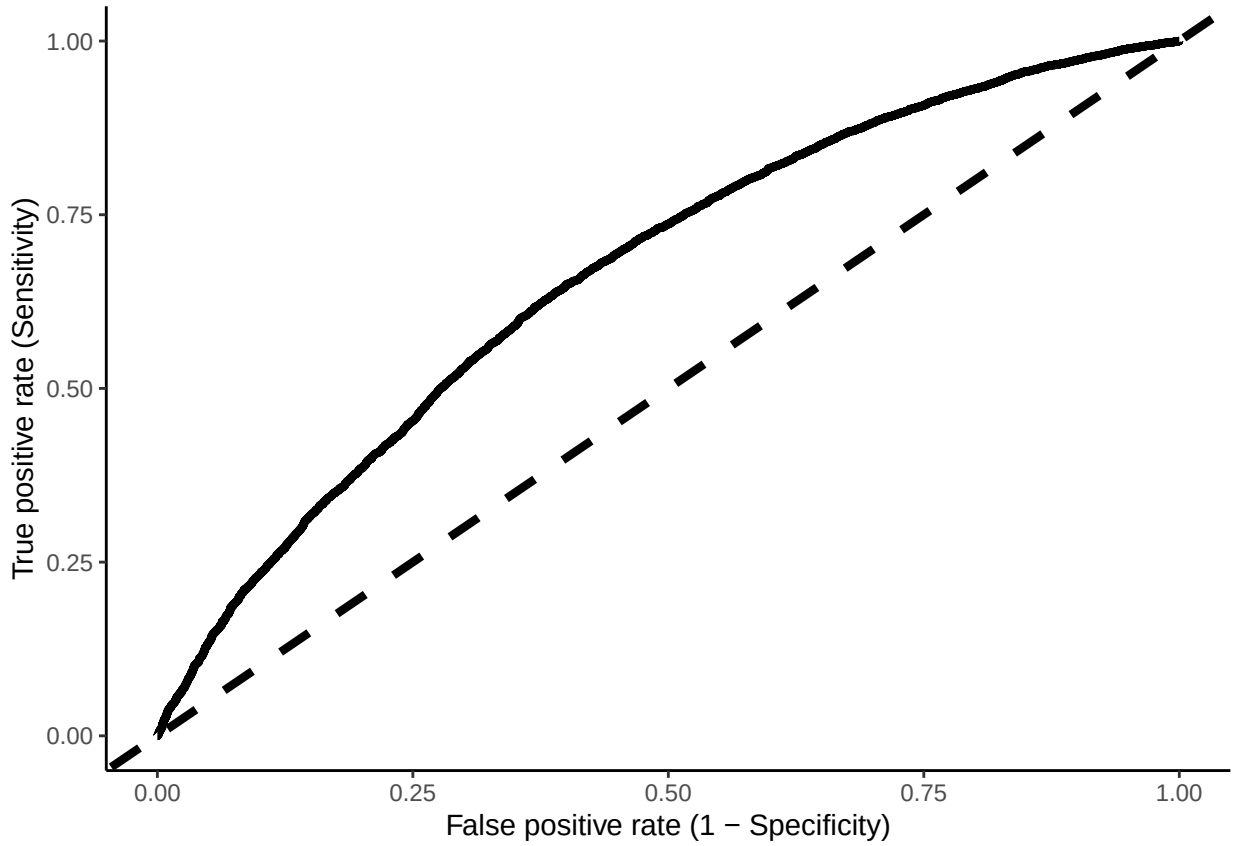
```
print(conf_int)
```

```
##                         2.5 %       97.5 %
## (Intercept)      -2.218863835 -1.079093202
## wl               -0.007087512 -0.004422554
## age              -0.009037747 -0.005023103
## genhlthvery good -0.258733979 -0.061402838
## genhlthgood      -0.781264585 -0.585045167
## genhlthfair      -1.220012517 -0.976051993
## genhlthpoor      -1.826037078 -1.473281593
## hlthplan          0.347728438  0.540546376
## height            0.039911470  0.056287123
```

```
# ROC curve for backward_bic
pred <- prediction(backward_bic$fitted.values, cdc$exerany)
perf <- performance(pred,"tpr","fpr")
performance(pred, "auc")@y.values
```

```
## [[1]]
## [1] 0.6658579
```

```
data.frame(fpr = perf@x.values[[1]],
tpr = perf@y.values[[1]]) |>
ggplot(aes(x = fpr, y = tpr)) +
geom_line(lwd=1.5) +
geom_abline(slope = 1, intercept = 0, lty = 2,
lwd = 1.5) +
labs(x = "False positive rate (1 - Specificity)",
y = "True positive rate (Sensitivity)") +
theme_classic()
```




```
# Create prediction and performance objects using your actual data
pred <- prediction(backward_bic$fitted.values, cdc$exerany)
perf <- performance(pred, "tpr", "fpr")

# Calculate distances to (0, 1)
distances <- sqrt((1 - perf@y.values[[1]])^2 + perf@x.values[[1]]^2)

# Identify the index of the closest point
min_index <- which.min(distances)

# Extract the best threshold
best_threshold <- perf@alpha.values[[1]][min_index]
```

```r
# Display the best threshold
if (!is.na(best_threshold)) {
  print(paste("Best threshold is:", best_threshold))
} else {
  print("Could not find the best threshold.")
}
```

```
## [1] "Best threshold is: 0.756789894121578"
```

```r
# Get the predicted probabilities
predicted_probs <- predict(backward_bic, type = "response")

# Create a new column with predicted classes based on the 0.76 threshold
cdc$predicted_exerany <- ifelse(predicted_probs > best_threshold, 1, 0)

# Convert the actual and predicted values to factors
cdc$predicted_exerany <- factor(cdc$predicted_exerany)
cdc$exerany <- factor(cdc$exerany)

# Create the confusion matrix
confusionMatrix(cdc$predicted_exerany, cdc$exerany)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3108 5408
##          1 1978 9506
##
##                Accuracy : 0.6307
##                  95% CI : (0.624, 0.6374)
##     No Information Rate : 0.7457
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2033
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6111
##             Specificity : 0.6374
##          Pos Pred Value : 0.3650
##          Neg Pred Value : 0.8278
##              Prevalence : 0.2543
##          Detection Rate : 0.1554
##    Detection Prevalence : 0.4258
##       Balanced Accuracy : 0.6242
##
##        'Positive' Class : 0
##
```