```r
suppressMessages(library(psych))
suppressMessages(library(dplyr))
suppressMessages(library(tidyverse))
suppressMessages(library(ResourceSelection))
suppressMessages(library(lmtest))
suppressMessages(library(VGAM))
```

####Case 1

###Data Preparation

##Data importing and cleaning

```r
df <- read.csv("MIDUS_III_Final_Exam_Fall2023_data.csv") %>% drop_na
df1 <- df[,c("heart", "bp", "smoke", "age", "male", "exercise")]
```

#Data Description

```r
describe(df1)
```

```
##          vars    n  mean    sd median trimmed   mad min max range  skew
## heart       1 1979  0.24  0.43      0    0.17  0.00   0   1     1  1.22
## bp          2 1979  0.51  0.50      1    0.51  0.00   0   1     1 -0.05
## smoke       3 1979  0.64  0.48      1    0.67  0.00   0   1     1 -0.57
## age         4 1979 64.09 11.01     64   63.86 11.86  39  92    53  0.16
## male        5 1979  0.49  0.50      0    0.49  0.00   0   1     1  0.03
## exercise    6 1979  0.19  0.39      0    0.12  0.00   0   1     1  1.55
##          kurtosis   se
## heart       -0.51 0.01
## bp          -2.00 0.01
## smoke       -1.67 0.01
## age         -0.68 0.25
## male        -2.00 0.01
## exercise     0.42 0.01
```

```r
lapply(df1[-4], table)
```

```
## $heart
## 
##    0    1 
## 1505  474 
## 
## $bp
## 
##    0    1 
##  966 1013 
## 
## $smoke
## 
##    0    1 
##  717 1262 
## 
## $male
## 
##    0    1 
## 1005  974 
##
```

```
## $exercise
##
##    0    1
## 1597  382
```

#Bivariable Analysis

```r
cor(df1,method = "spearman")[1, 2:6, drop = F]
```

```
##              bp      smoke      age      male    exercise
## heart 0.2116867 0.07816263 0.229714 0.1035217 -0.00748379
```

heart is more related with bp and age

###Step1

##Modeling

```r
#Since the outcome variable is binary, choose logistic regression (glm)
model1 <- glm(heart ~ ., family = binomial, data = df1)
summary(model1)
```

```
##
## Call:
## glm(formula = heart ~ ., family = binomial, data = df1)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.909779   0.370817 -13.240  < 2e-16 ***
## bp           0.851829   0.116975   7.282 3.29e-13 ***
## smoke        0.282466   0.119265   2.368   0.0179 *
## age          0.042696   0.005266   8.107 5.18e-16 ***
## male         0.444804   0.111906   3.975 7.04e-05 ***
## exercise     0.189172   0.143432   1.319   0.1872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2179.0  on 1978  degrees of freedom
## Residual deviance: 1993.1  on 1973  degrees of freedom
## AIC: 2005.1
##
## Number of Fisher Scoring iterations: 4
```

###Step2

##Modeling

```r
#Since the outcome variable is binary, choose logistic regression (glm)
model2 <- glm(heart ~ . + smoke*male, family = binomial, data = df1)
summary(model2)
```

```
##
## Call:
## glm(formula = heart ~ . + smoke * male, family = binomial, data = df1)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -4.634221   0.384024 -12.068  < 2e-16 ***
## bp           0.849921   0.117204   7.252 4.12e-13 ***
## smoke       -0.028744   0.171281  -0.168   0.8667
## age          0.041602   0.005281   7.878 3.33e-15 ***
## male         0.044772   0.196567   0.228   0.8198
## exercise     0.198499   0.143679   1.382   0.1671
## smoke:male   0.591985   0.239503   2.472   0.0134 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2179  on 1978  degrees of freedom
## Residual deviance: 1987  on 1972  degrees of freedom
## AIC: 2001
##
## Number of Fisher Scoring iterations: 4
```

###Step3

##Model fit

```
#test overall model fit
hoslem.test(model1$y, model1$fitted.values)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model1$y, model1$fitted.values
## X-squared = 13.53, df = 8, p-value = 0.09487
```

```
hoslem.test(model2$y, model2$fitted.values)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model2$y, model2$fitted.values
## X-squared = 13.677, df = 8, p-value = 0.09058
```

```
#comparing 2 models
lmtest::lrtest(model2, model1)
```

```
## Likelihood ratio test
##
## Model 1: heart ~ bp + smoke + age + male + exercise + smoke * male
## Model 2: heart ~ bp + smoke + age + male + exercise
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   7 -993.51
## 2   6 -996.57 -1 6.1138    0.01341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The Hosmer and Lemeshow Test result ($p = 0.095$ and 0.091) indicates that 2 models are good fit.

The likelihood ratio test has p-value $= 0.0134$, so we reject the null hypothesis and concludes that model2 (interaction effect model) has better fit.

##Check outliers

```r
#unique(which(abs(rstandard(model1)) > 2))
#unique(which(abs(rstandard(model2)) > 2))
# Calculate Pearson Residuals
pearson_res1 <- residuals(model1, type = "pearson")
pearson_res2 <- residuals(model2, type = "pearson")

outlier_threshold1 <- 2 * sd(pearson_res1)
outliers1 <- which(abs(pearson_res1) > outlier_threshold1)
df1[outliers1, ]
```

```
##     heart bp smoke age male exercise
## 4       1  0     1  51    1        1
## 5       1  0     0  66    0        1
## 7       1  1     0  53    0        0
## 25      1  1     0  58    0        0
## 40      1  0     0  76    0        0
## 75      1  0     1  65    0        0
## 86      1  0     1  74    0        0
## 99      1  0     0  60    0        1
## 104     1  1     1  54    0        0
## 124     1  1     0  50    1        0
## 149     1  0     0  51    0        0
## 157     1  1     1  44    1        0
## 166     1  0     0  62    1        0
## 170     1  1     1  54    0        0
## 175     1  0     0  64    1        0
## 176     1  0     0  49    0        0
## 189     1  0     0  47    1        0
## 259     1  0     0  59    0        1
## 267     1  0     0  54    1        0
## 305     1  0     1  74    0        0
## 337     1  0     1  69    0        1
## 352     1  0     1  52    0        0
## 422     1  1     1  56    0        0
## 423     1  0     0  62    1        0
## 458     1  0     1  45    1        0
## 466     1  0     1  70    0        0
## 469     1  0     1  67    0        1
## 500     1  0     1  62    1        0
## 528     1  0     1  65    0        1
## 530     1  0     1  63    1        0
## 552     1  0     1  49    1        0
## 566     1  0     0  46    1        0
## 573     1  0     1  64    0        0
## 582     1  0     1  47    1        0
## 610     1  0     1  73    0        0
## 617     1  0     1  51    1        0
## 634     1  0     1  74    0        0
## 657     1  0     0  53    1        0
## 678     1  0     1  62    0        0
## 692     1  1     0  63    0        0
## 693     1  0     0  62    1        0
## 714     1  0     0  64    1        0
## 721     1  0     1  58    1        0
```

```
## 735         1   0       1   64      1           0
## 749         1   1       1   54      0           0
## 755         1   0       1   61      1           1
## 762         1   0       0   56      0           1
## 784         1   0       1   65      0           0
## 801         1   1       1   53      0           0
## 809         1   1       1   52      0           0
## 872         1   0       1   46      1           0
## 877         1   1       1   45      1           0
## 941         1   0       1   50      0           1
## 947         1   0       1   76      0           0
## 954         1   0       0   51      1           0
## 971         1   0       1   60      1           0
## 981         1   1       1   53      0           0
## 991         1   0       0   76      0           0
## 1003        1   0       1   59      1           0
## 1071        1   0       0   66      1           1
## 1088        1   1       0   56      0           0
## 1114        1   0       0   43      1           0
## 1121        1   0       0   53      1           1
## 1128        1   0       1   59      1           0
## 1142        1   0       1   59      0           0
## 1149        1   1       0   56      0           0
## 1209        1   0       1   62      0           0
## 1221        1   1       0   52      1           0
## 1231        1   1       1   45      0           0
## 1271        1   1       0   52      1           0
## 1292        1   0       1   64      1           0
## 1300        1   0       0   48      0           0
## 1345        1   0       0   77      0           1
## 1353        1   0       1   59      1           0
## 1363        1   0       0   76      0           0
## 1383        1   1       0   55      0           0
## 1398        1   0       1   74      0           0
## 1413        1   0       1   52      1           0
## 1447        1   1       1   48      0           0
## 1461        1   0       1   73      0           0
## 1489        1   0       1   67      0           0
## 1504        1   0       0   50      0           1
## 1530        1   1       0   52      1           0
## 1535        1   0       1   70      0           0
## 1539        1   0       0   78      0           0
## 1569        1   1       0   58      0           0
## 1576        1   1       0   56      0           0
## 1582        1   1       0   46      0           0
## 1606        1   0       0   53      1           0
## 1611        1   0       1   59      0           0
## 1612        1   1       1   54      0           0
## 1628        1   0       0   63      1           1
## 1640        1   0       1   61      1           0
## 1650        1   0       0   64      1           0
## 1671        1   0       0   57      0           0
## 1705        1   1       1   53      0           0
## 1768        1   0       0   73      0           0
```

```
## 1804      1  0      0  76    0         0
## 1811      1  0      1  71    0         0
## 1838      1  0      1  76    0         0
## 1844      1  0      1  58    1         1
## 1849      1  1      1  50    0         0
## 1853      1  0      1  59    1         0
## 1859      1  1      0  51    0         0
## 1889      1  0      0  65    1         1
## 1891      1  0      1  72    0         0
## 1892      1  0      1  46    0         0
## 1906      1  0      0  63    0         1
## 1915      1  1      0  49    1         0
## 1920      1  0      1  58    0         0
## 1955      1  0      0  57    1         0
## 1959      1  1      1  52    0         0
## 1969      1  0      1  65    1         0
```

```r
outlier_threshold2 <- 2 * sd(pearson_res2)
outliers2 <- which(abs(pearson_res2) > outlier_threshold2)
df1[outliers2, ]
```

```
##       heart bp smoke age male exercise
## 4         1  0     1  51    1        1
## 5         1  0     0  66    0        1
## 7         1  1     0  53    0        0
## 25        1  1     0  58    0        0
## 40        1  0     0  76    0        0
## 75        1  0     1  65    0        0
## 86        1  0     1  74    0        0
## 99        1  0     0  60    0        1
## 104       1  1     1  54    0        0
## 124       1  1     0  50    1        0
## 149       1  0     0  51    0        0
## 166       1  0     0  62    1        0
## 170       1  1     1  54    0        0
## 175       1  0     0  64    1        0
## 176       1  0     0  49    0        0
## 189       1  0     0  47    1        0
## 210       1  0     0  70    1        1
## 229       1  1     1  57    0        0
## 248       1  0     0  76    1        0
## 259       1  0     0  59    0        1
## 267       1  0     0  54    1        0
## 305       1  0     1  74    0        0
## 337       1  0     1  69    0        1
## 352       1  0     1  52    0        0
## 422       1  1     1  56    0        0
## 423       1  0     0  62    1        0
## 458       1  0     1  45    1        0
## 466       1  0     1  70    0        0
## 469       1  0     1  67    0        1
## 500       1  0     1  62    1        0
## 511       1  1     1  58    0        0
## 528       1  0     1  65    0        1
## 530       1  0     1  63    1        0
```

```
## 552       1  0    1  49   1        0
## 566       1  0    0  46   1        0
## 573       1  0    1  64   0        0
## 582       1  0    1  47   1        0
## 586       1  1    1  58   0        0
## 610       1  0    1  73   0        0
## 617       1  0    1  51   1        0
## 634       1  0    1  74   0        0
## 657       1  0    0  53   1        0
## 678       1  0    1  62   0        0
## 693       1  0    0  62   1        0
## 714       1  0    0  64   1        0
## 721       1  0    1  58   1        0
## 749       1  1    1  54   0        0
## 751       1  1    1  58   0        0
## 762       1  0    0  56   0        1
## 784       1  0    1  65   0        0
## 801       1  1    1  53   0        0
## 809       1  1    1  52   0        0
## 872       1  0    1  46   1        0
## 899       1  1    1  58   0        0
## 941       1  0    1  50   0        1
## 947       1  0    1  76   0        0
## 954       1  0    0  51   1        0
## 971       1  0    1  60   1        0
## 981       1  1    1  53   0        0
## 991       1  0    0  76   0        0
## 1003      1  0    1  59   1        0
## 1008      1  1    0  57   1        0
## 1013      1  1    1  58   0        0
## 1071      1  0    0  66   1        1
## 1087      1  1    1  54   0        1
## 1088      1  1    0  56   0        0
## 1114      1  0    0  43   1        0
## 1121      1  0    0  53   1        1
## 1128      1  0    1  59   1        0
## 1131      1  0    0  76   1        0
## 1142      1  0    1  59   0        0
## 1149      1  1    0  56   0        0
## 1209      1  0    1  62   0        0
## 1221      1  1    0  52   1        0
## 1231      1  1    1  45   0        0
## 1271      1  1    0  52   1        0
## 1300      1  0    0  48   0        0
## 1350      1  0    0  76   1        0
## 1353      1  0    1  59   1        0
## 1363      1  0    0  76   0        0
## 1383      1  1    0  55   0        0
## 1398      1  0    1  74   0        0
## 1413      1  0    1  52   1        0
## 1447      1  1    1  48   0        0
## 1450      1  0    0  76   1        0
## 1461      1  0    1  73   0        0
## 1489      1  0    1  67   0        0
```

```
## 1504     1  0     0  50    0        1
## 1530     1  1     0  52    1        0
## 1535     1  0     1  70    0        0
## 1539     1  0     0  78    0        0
## 1569     1  1     0  58    0        0
## 1576     1  1     0  56    0        0
## 1579     1  0     0  75    1        0
## 1582     1  1     0  46    0        0
## 1606     1  0     0  53    1        0
## 1611     1  0     1  59    0        0
## 1612     1  1     1  54    0        0
## 1628     1  0     0  63    1        1
## 1640     1  0     1  61    1        0
## 1650     1  0     0  64    1        0
## 1671     1  0     0  57    0        0
## 1705     1  1     1  53    0        0
## 1768     1  0     0  73    0        0
## 1775     1  0     0  71    1        1
## 1802     1  1     1  52    0        1
## 1804     1  0     0  76    0        0
## 1811     1  0     1  71    0        0
## 1838     1  0     1  76    0        0
## 1844     1  0     1  58    1        1
## 1849     1  1     1  50    0        0
## 1853     1  0     1  59    1        0
## 1859     1  1     0  51    0        0
## 1889     1  0     0  65    1        1
## 1891     1  0     1  72    0        0
## 1892     1  0     1  46    0        0
## 1906     1  0     0  63    0        1
## 1915     1  1     0  49    1        0
## 1920     1  0     1  58    0        0
## 1955     1  0     0  57    1        0
## 1959     1  1     1  52    0        0
```

```r
# Calculate Deviance Residuals
deviance_res1 <- residuals(model1, type = "deviance")
deviance_res2 <- residuals(model2, type = "deviance")

outlier_threshold1 <- 2 * sd(deviance_res1)
outliers1 <- which(abs(deviance_res1) > outlier_threshold1)
df1[outliers1, ]
```

```
##      heart bp smoke age male exercise
## 5        1  0     0  66    0        1
## 75       1  0     1  65    0        0
## 99       1  0     0  60    0        1
## 149      1  0     0  51    0        0
## 176      1  0     0  49    0        0
## 189      1  0     0  47    1        0
## 259      1  0     0  59    0        1
## 267      1  0     0  54    1        0
## 352      1  0     1  52    0        0
## 458      1  0     1  45    1        0
## 552      1  0     1  49    1        0
```

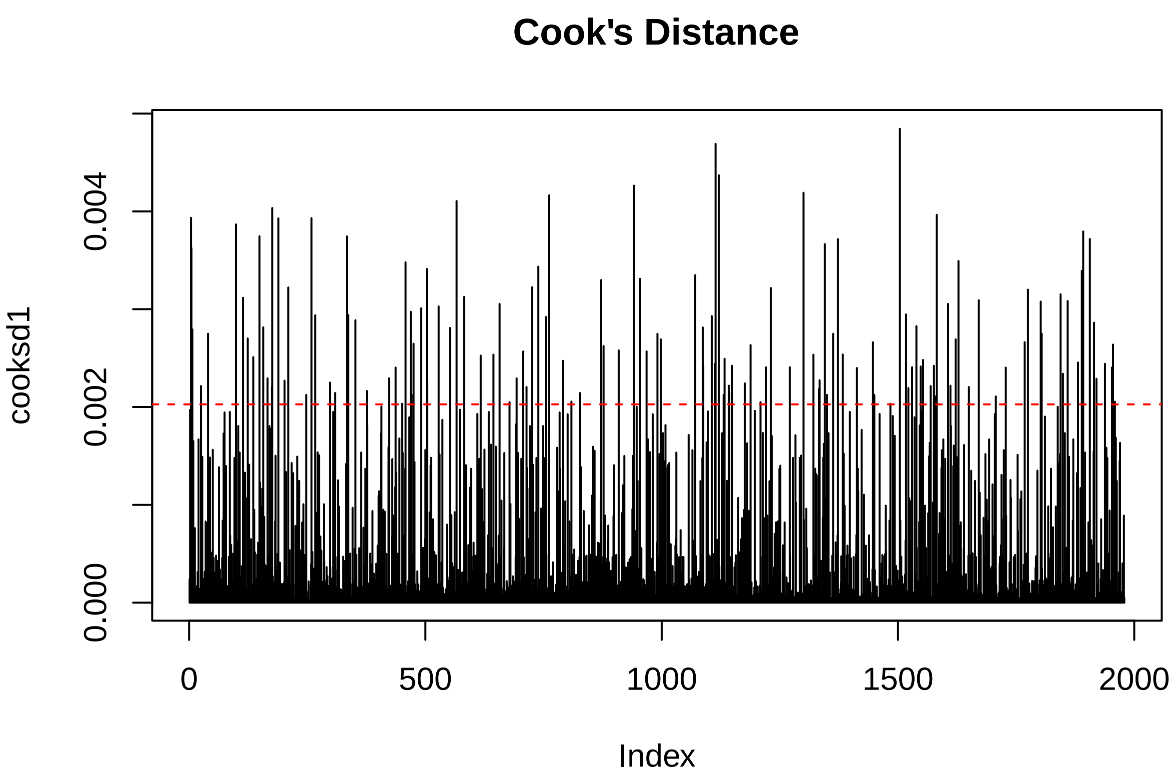
```
## 566      1  0     0  46    1        0
## 573      1  0     1  64    0        0
## 582      1  0     1  47    1        0
## 617      1  0     1  51    1        0
## 657      1  0     0  53    1        0
## 678      1  0     1  62    0        0
## 762      1  0     0  56    0        1
## 784      1  0     1  65    0        0
## 872      1  0     1  46    1        0
## 941      1  0     1  50    0        1
## 954      1  0     0  51    1        0
## 1114     1  0     0  43    1        0
## 1121     1  0     0  53    1        1
## 1142     1  0     1  59    0        0
## 1209     1  0     1  62    0        0
## 1231     1  1     1  45    0        0
## 1300     1  0     0  48    0        0
## 1413     1  0     1  52    1        0
## 1504     1  0     0  50    0        1
## 1582     1  1     0  46    0        0
## 1606     1  0     0  53    1        0
## 1611     1  0     1  59    0        0
## 1671     1  0     0  57    0        0
## 1859     1  1     0  51    0        0
## 1892     1  0     1  46    0        0
## 1906     1  0     0  63    0        1
## 1920     1  0     1  58    0        0
## 1955     1  0     0  57    1        0
```

```
outlier_threshold2 <- 2 * sd(deviance_res2)
outliers2 <- which(abs(deviance_res2) > outlier_threshold2)
df1[outliers2, ]
```

```
##      heart bp smoke age male exercise
## 75       1  0     1  65    0        0
## 99       1  0     0  60    0        1
## 149      1  0     0  51    0        0
## 166      1  0     0  62    1        0
## 175      1  0     0  64    1        0
## 176      1  0     0  49    0        0
## 189      1  0     0  47    1        0
## 259      1  0     0  59    0        1
## 267      1  0     0  54    1        0
## 352      1  0     1  52    0        0
## 423      1  0     0  62    1        0
## 458      1  0     1  45    1        0
## 552      1  0     1  49    1        0
## 566      1  0     0  46    1        0
## 573      1  0     1  64    0        0
## 582      1  0     1  47    1        0
## 617      1  0     1  51    1        0
## 657      1  0     0  53    1        0
## 678      1  0     1  62    0        0
## 693      1  0     0  62    1        0
## 714      1  0     0  64    1        0
```

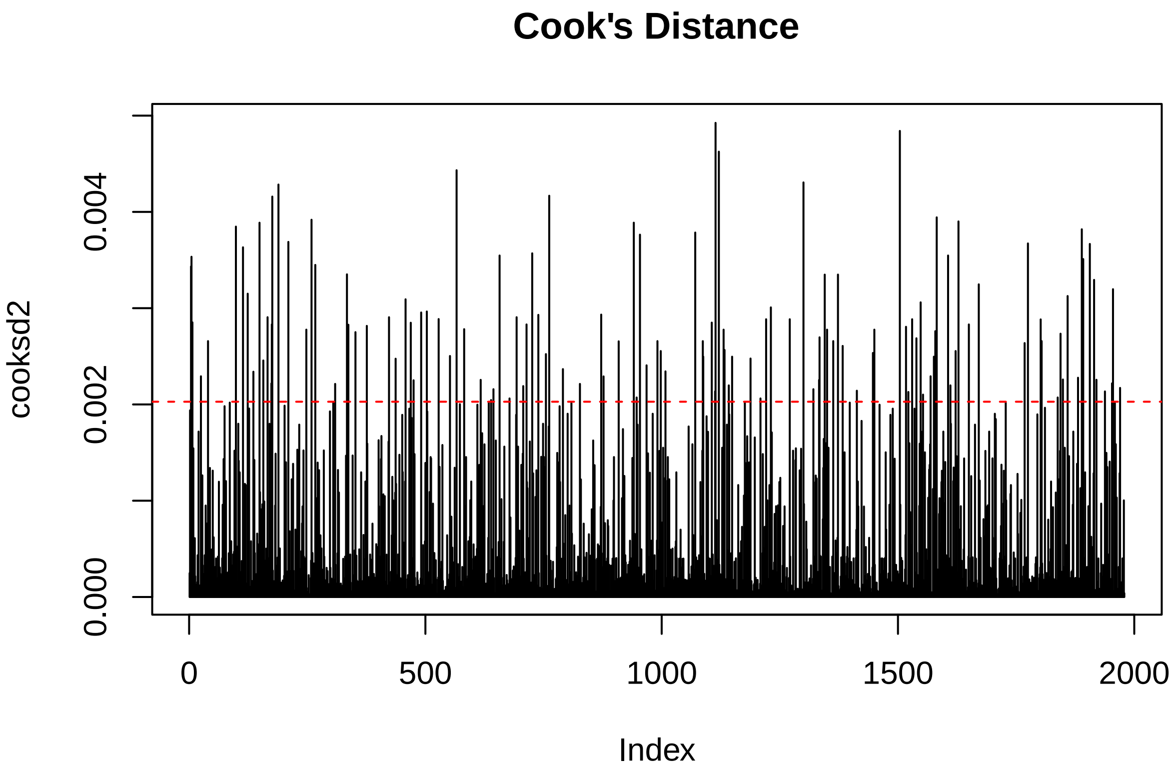
```
## 762         1  0      0  56     0         1
## 784         1  0      1  65     0         0
## 872         1  0      1  46     1         0
## 941         1  0      1  50     0         1
## 954         1  0      0  51     1         0
## 1114        1  0      0  43     1         0
## 1121        1  0      0  53     1         1
## 1142        1  0      1  59     0         0
## 1209        1  0      1  62     0         0
## 1231        1  1      1  45     0         0
## 1300        1  0      0  48     0         0
## 1413        1  0      1  52     1         0
## 1489        1  0      1  67     0         0
## 1504        1  0      0  50     0         1
## 1582        1  1      0  46     0         0
## 1606        1  0      0  53     1         0
## 1611        1  0      1  59     0         0
## 1650        1  0      0  64     1         0
## 1671        1  0      0  57     0         0
## 1892        1  0      1  46     0         0
## 1920        1  0      1  58     0         0
## 1955        1  0      0  57     1         0
```

```r
# Cook's distance
cooksd1 <- cooks.distance(model1)
plot(cooksd1, type="h", main="Cook's Distance") %>%
abline(h = 4/(nrow(df1)-length(coef(model1))), col = "red", lty = 2)
```




```r
cooksd2 <- cooks.distance(model2)
plot(cooksd2, type="h", main="Cook's Distance") %>%
```

```r
abline(h = 4/(nrow(df1)-length(coef(model2))), col = "red", lty = 2)
```




```r
summary(cooksd1)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 9.134e-06 6.223e-05 1.753e-04 5.140e-04 5.166e-04 4.844e-03
```

```r
summary(cooksd2)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 1.273e-05 6.175e-05 1.818e-04 5.138e-04 5.048e-04 4.925e-03
```

```r
# We find the cook's distance is small suggests the outliers do not affect the model that much.
```

####Case 2

###Data Preparation

##Data importing

```r
df2 <- df[,c("health", "depress", "alcage", "cigage", "age", "bp")]
```

#Data Description

```r
describe(df2)
```

```
##         vars    n  mean    sd median trimmed   mad min max range  skew kurtosis
## health     1 1979  1.74  0.87      2    1.62  1.48   1   4     3  0.98     0.14
## depress    2 1979  0.08  0.27      0    0.00  0.00   0   1     1  3.10     7.60
## alcage     3 1979 17.66  3.83     17   17.51  1.48   1  69    68  2.53    25.49
## cigage     4 1979 15.63  4.33     16   15.47  2.97   3  60    57  2.02    14.80
## age        5 1979 64.09 11.01     64   63.86 11.86  39  92    53  0.16    -0.68
## bp         6 1979  0.51  0.50      1    0.51  0.00   0   1     1 -0.05    -2.00
##           se
```

```
## health  0.02
## depress 0.01
## alcage  0.09
## cigage  0.10
## age     0.25
## bp      0.01
```

```r
lapply(df2[,c(1,2,6)], table)
```

```
## $health
##
##   1   2   3   4
## 963 659 257 100
##
## $depress
##
##    0    1
## 1821  158
##
## $bp
##
##    0    1
##  966 1013
```

#Bivariable Analysis

```r
cor(df2, method = "spearman")[1, 2:6, drop = F]
```

```
##          depress      alcage      cigage        age        bp
## health 0.1510899 -0.01427585 -0.01183471 0.06569963 0.2452973
```

health is more related with depress and bp

###Step1

##Modeling

```r
#Since the outcome variable is multicategorical and ordered, we choose cumulative logit model (vglm)
model3 <- vglm(health ~ ., family = cumulative(parallel = TRUE), data = df2)
summary(model3)
```

```
##
## Call:
## vglm(formula = health ~ ., family = cumulative(parallel = TRUE),
##     data = df2)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  0.870987   0.305714   2.849  0.00439 **
## (Intercept):2  2.547893   0.310849   8.197 2.47e-16 ***
## (Intercept):3  4.033825   0.323450  12.471  < 2e-16 ***
## depress       -1.201805   0.154374  -7.785 6.97e-15 ***
## alcage         0.003656   0.011935   0.306  0.75934
## cigage        -0.005030   0.010421  -0.483  0.62931
## age           -0.005494   0.004165  -1.319  0.18721
## bp            -0.945530   0.090151 -10.488  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2]),
## logitlink(P[Y<=3])
##
## Residual deviance: 4305.659 on 5929 degrees of freedom
##
## Log-likelihood: -2152.83 on 5929 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Exponentiated coefficients:
##   depress    alcage    cigage       age        bp
## 0.3006510 1.0036629 0.9949824 0.9945215 0.3884738
```

###Step2

```r
#
```

###Step3

##modeling

```r
model4 <- glm(bp ~ age + depress + factor(health), family = 'binomial', data = df2)
summary(model4)
```

```
##
## Call:
## glm(formula = bp ~ age + depress + factor(health), family = "binomial",
##     data = df2)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -3.156634   0.299377 -10.544  < 2e-16 ***
## age              0.043069   0.004533   9.501  < 2e-16 ***
## depress         -0.135462   0.182238  -0.743    0.457
## factor(health)2  0.691221   0.105434   6.556 5.53e-11 ***
## factor(health)3  1.098250   0.153704   7.145 8.98e-13 ***
## factor(health)4  1.968987   0.272189   7.234 4.69e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2742.4  on 1978  degrees of freedom
## Residual deviance: 2517.5  on 1973  degrees of freedom
## AIC: 2529.5
##
## Number of Fisher Scoring iterations: 4
```

exp(-3.156634+0.043069age-0.135462depress+0.691221f2+1.098250f3+1.968987f4) Excellent (health = 1) 1)age=64,depress=0 Prob=$1/(1 + exp(-(-3.156634 + 0.043069 * 64 - 0.135462 * 0))) = 40.1\%$ 2)age=64,depress=1 Prob=$1/(1 + exp(-(-3.156634 + 0.043069 * 64 - 0.135462 * 1))) = 36.9\%$ Diff=$40.1\% - 36.9\% = 3.2$

Good (health = 2) 1)age=64,depress=0 Prob=$1/(1 + exp(-(-3.156634 + 0.043069 * 64 - 0.135462 * 0 +$

$0.691221))) = 57.2\%$ 2)age=64,depress=1 Prob=$1/(1 + exp(-(-3.156634 + 0.043069 * 64 - 0.135462 * 1 + 0.691221))) = 53.9\%$ Diff=$57.2\% - 53.9\% = 3.3$

Fair (health = 3) 1)age=64,depress=0 Prob=$1/(1 + exp(-(-3.156634 + 0.043069 * 64 - 0.135462 * 0 + 1.098250))) = 66.8\%$ 2)age=64,depress=1 Prob=$1/(1 + exp(-(-3.156634 + 0.043069 * 64 - 0.135462 * 1 + 1.098250))) = 63.7\%$ Diff=$66.8\% - 63.7\% = 3.1$

Poor (health = 4) 1)age=64,depress=0 df2$bp < -factor(df$bp, labels = lev Prob=$1/(1 + exp(-(-3.156634 + 0.043069 * 64 - 0.135462 * 0 + 1.968987))) = 82.8\%$ 2)age=64,depress=1 Prob=$1/(1 + exp(-(-3.156634 + 0.043069 * 64 - 0.135462 * 1 + 1.968987))) = 80.7\%$ Diff=$82.8\% - 80.7\% = 2.1$