project 8

Binary logistic regression

2022 M06 2

mydata=read.csv(file.choose(),header=T,sep = ",")  
head(mydata)

## admit gre gpa rank  
## 1 0 380 3.61 3  
## 2 1 660 3.67 3  
## 3 1 800 4.00 1  
## 4 1 640 3.19 4  
## 5 0 520 2.93 4  
## 6 1 760 3.00 2

str(mydata)

## 'data.frame': 400 obs. of 4 variables:  
## $ admit: int 0 1 1 1 0 1 1 0 1 0 ...  
## $ gre : int 380 660 800 640 520 760 560 400 540 700 ...  
## $ gpa : num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...  
## $ rank : int 3 3 1 4 4 2 1 2 3 2 ...

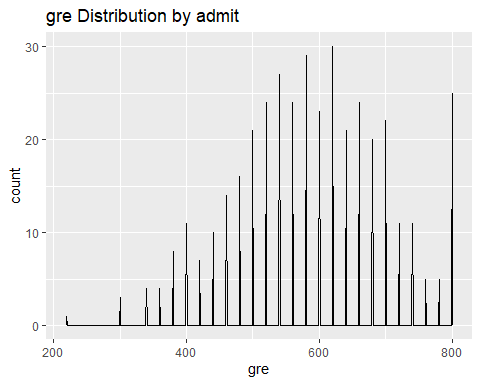
summary(mydata)

## admit gre gpa rank   
## Min. :0.0000 Min. :220.0 Min. :2.260 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.:520.0 1st Qu.:3.130 1st Qu.:2.000   
## Median :0.0000 Median :580.0 Median :3.395 Median :2.000   
## Mean :0.3175 Mean :587.7 Mean :3.390 Mean :2.485   
## 3rd Qu.:1.0000 3rd Qu.:660.0 3rd Qu.:3.670 3rd Qu.:3.000   
## Max. :1.0000 Max. :800.0 Max. :4.000 Max. :4.000

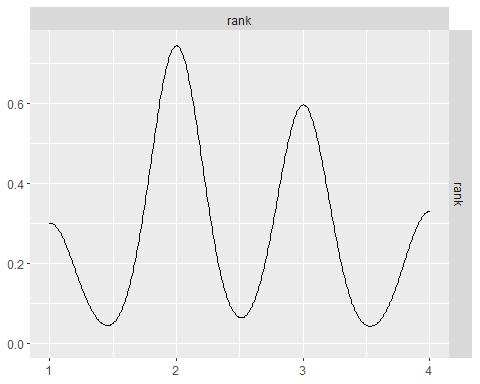
#checking for missing value  
sum(is.na(mydata))

## [1] 0

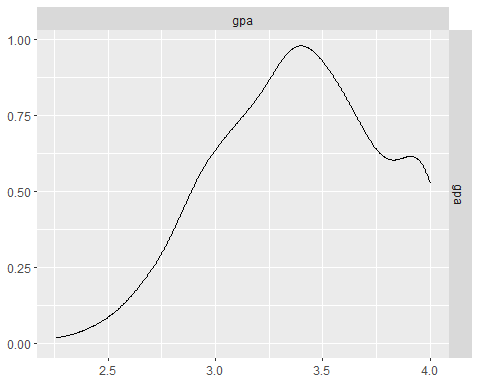
#result show, we have no missing value in our dataset  
#identifying the rows with NAs  
r=rownames(mydata)[apply(mydata, 2, anyNA)]  
#Similary result shows no row with missing value  
  
#Data Visualization  
ggplot(mydata, aes(gre, colour =gre)) +  
geom\_freqpoly(binwidth = 1) + labs(title="gre Distribution by admit")



#visual 2  
ggpairs(mydata,4)



ggpairs(mydata,3)



#Let's calculate the frequency of response variable under each rank.   
sapply(mydata, sd)

## admit gre gpa rank   
## 0.4660867 115.5165364 0.3805668 0.9444602

#Result shows the standard deviation value of each variable, for variable gre we have sd value of 115.52  
##two-way contingency table of categorical outcome and predictors we want to make sure there are not 0 cells  
xtabs(~ admit + rank, data = mydata)

## rank  
## admit 1 2 3 4  
## 0 28 97 93 55  
## 1 33 54 28 12

#Output shows the cross tab for admit and rank, rank 1, 2, 3, 4 under admit 0 is 28,97,93 and 55 respectively while rank 1, 2, 3, 4 under admit 1 is 33, 54, 28 and 12 respectively.  
mydata$rank = factor(mydata$rank)  
mylogit = glm(admit ~ gre + gpa + rank, data = mydata, family = "binomial") #binary logistic model  
summary(mylogit)

##   
## Call:  
## glm(formula = admit ~ gre + gpa + rank, family = "binomial",   
## data = mydata)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6268 -0.8662 -0.6388 1.1490 2.0790   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.989979 1.139951 -3.500 0.000465 \*\*\*  
## gre 0.002264 0.001094 2.070 0.038465 \*   
## gpa 0.804038 0.331819 2.423 0.015388 \*   
## rank2 -0.675443 0.316490 -2.134 0.032829 \*   
## rank3 -1.340204 0.345306 -3.881 0.000104 \*\*\*  
## rank4 -1.551464 0.417832 -3.713 0.000205 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 499.98 on 399 degrees of freedom  
## Residual deviance: 458.52 on 394 degrees of freedom  
## AIC: 470.52  
##   
## Number of Fisher Scoring iterations: 4

#We see the deviance residuals, which is a measure of model fit. This part of output shows the distribution of the deviance residuals for individual cases used in the model.  
#Variables whose P value less than 0.05 are considered to be statistically significant. Since the p-value is < 0.05 for gre,gpa,factor(rank)2,factor(rank)3 and factor(rank)4 variables are significant. which means there is relationship between the dependent variable admit and the independent variables at 5% confidence level. The positive coefficient of gre and gpa shows a positive relationship while the negative coefficient of factor(rank)2,factor(rank)3 and factor(rank)4 shows a negative relationship.  
#The logistic regression coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable.  
#For every one unit change in gre, the log odds of admission (versus not admission) increases by 0.002264  
#For every one unit change in gpa, the log odds of admission (versus not admission) increases by 0.804038  
#The rank means having attended an undergraduate institution with rank of 2, versus an institution with a rank of 1, changes the log odds of admission by negative 0.675443  
#Similarly, having attended an undergraduate institution with rank of 3, versus an institution with a rank of 1, changes the log odds of admission by negative 1.340204 and lastly, having attended an undergraduate institution with rank of 4, versus an institution with a rank of 1, changes the log odds of admission by negative 1.551464 and lastly  
  
#We can use the confint function to obtain confidence intervals for the coefficient estimates. Note that for logistic models, confidence intervals are based on the profiled log-likelihood function. We can also get CIs based on just the standard errors by using the default method  
## CIs using profiled log-likelihood confint(mylogit)   
confint(mylogit)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) -6.2716202334 -1.792547080  
## gre 0.0001375921 0.004435874  
## gpa 0.1602959439 1.464142727  
## rank2 -1.3008888002 -0.056745722  
## rank3 -2.0276713127 -0.670372346  
## rank4 -2.4000265384 -0.753542605

## CIs using standard errors confint.default(mylogit)   
confint.default(mylogit)

## 2.5 % 97.5 %  
## (Intercept) -6.2242418514 -1.755716295  
## gre 0.0001202298 0.004408622  
## gpa 0.1536836760 1.454391423  
## rank2 -1.2957512650 -0.055134591  
## rank3 -2.0169920597 -0.663415773  
## rank4 -2.3703986294 -0.732528724

wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 4:6)

## Wald test:  
## ----------  
##   
## Chi-squared test:  
## X2 = 20.9, df = 3, P(> X2) = 0.00011

#From the output we can see that the p-value of the test is 0.00011.  
#H0: overall effect of rank is not significant. that is all equal to zero  
#H1: overall effect of rank is significant. that is not all equal to zero  
#Since this p-value is less than 0.05, we reject the null hypothesis of the Wald test and conclude that the overall effect of rank is statistically significant  
l = cbind(0,0,0,1,-1,0)   
wald.test(b = coef(mylogit), Sigma = vcov(mylogit), L = l)

## Wald test:  
## ----------  
##   
## Chi-squared test:  
## X2 = 5.5, df = 1, P(> X2) = 0.019

#Since this p-value is less than 0.05, we also reject the null hypothesis of the Wald test. The chi-squared test statistic of 5.5 with 1 degree of freedom is associated with a p-value of 0.019, indicating that the difference between the coefficient for rank=2 and the coefficient for rank=3 is statistically significant.  
#Odd Ratio  
#The ODDS is the ratio of the probability of an event occurring to the event not occurring. When we take a ratio of two such odds, it called the Odds Ratio.   
#Mathematically, one can compute the odds ratio by taking exponent of the estimated coefficients.  
(exp(coef(mylogit))) #obtaining odds ratio

## (Intercept) gre gpa rank2 rank3 rank4   
## 0.0185001 1.0022670 2.2345448 0.5089310 0.2617923 0.2119375

exp(cbind(OR = coef(mylogit), confint(mylogit)))

## Waiting for profiling to be done...

## OR 2.5 % 97.5 %  
## (Intercept) 0.0185001 0.001889165 0.1665354  
## gre 1.0022670 1.000137602 1.0044457  
## gpa 2.2345448 1.173858216 4.3238349  
## rank2 0.5089310 0.272289674 0.9448343  
## rank3 0.2617923 0.131641717 0.5115181  
## rank4 0.2119375 0.090715546 0.4706961

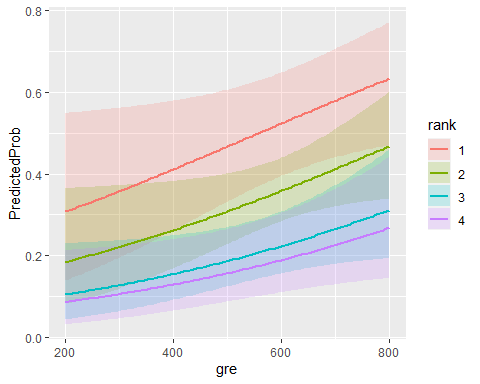
#Odds ratios that are greater than 1 indicate that the even is more likely to occur as the predictor increases. Odds ratios that are less than 1 indicate that the event is less likely to occur as the predictor increases.The above ODDS ratio table,we can observe that gre has an ODDS Ratio of 1.002, which indicates that for a one unit increase in gre the odds of being admitted to graduate school (versus not being admitted) increase by a factor of 1.002   
  
newdata1 = with(mydata, data.frame(gre = mean(gre), gpa = mean(gpa), rank = factor(1:4)))   
newdata1$rankP <- predict(mylogit, newdata = newdata1, type = "response")  
newdata1

## gre gpa rank rankP  
## 1 587.7 3.3899 1 0.5166016  
## 2 587.7 3.3899 2 0.3522846  
## 3 587.7 3.3899 3 0.2186120  
## 4 587.7 3.3899 4 0.1846684

#The output shows that the predicted probability of being accepted into a graduate program is 0.5166016 for students from the highest prestige undergraduate institutions (rank=1), and 0.3522846 for students from the lowest ranked institutions (rank=2), holding gre and gpa at their means.  
newdata2 = with(mydata, data.frame(gre = rep(seq(from = 200, to = 800, length.out = 100), 4), gpa = mean(gpa), rank = factor(rep(1:4, each = 100))))   
  
newdata3 = cbind(newdata2, predict(mylogit, newdata = newdata2, type="link", se=TRUE))   
newdata3 = within(newdata3, {  
 PredictedProb <- plogis(fit)  
 LL = plogis(fit - (1.96 \* se.fit))   
 UL = plogis(fit + (1.96 \* se.fit))  
})   
 ## view first few rows of final dataset  
head(newdata3)

## gre gpa rank fit se.fit residual.scale UL LL  
## 1 200.0000 3.3899 1 -0.8114870 0.5147714 1 0.5492064 0.1393812  
## 2 206.0606 3.3899 1 -0.7977632 0.5090986 1 0.5498513 0.1423880  
## 3 212.1212 3.3899 1 -0.7840394 0.5034491 1 0.5505074 0.1454429  
## 4 218.1818 3.3899 1 -0.7703156 0.4978239 1 0.5511750 0.1485460  
## 5 224.2424 3.3899 1 -0.7565919 0.4922237 1 0.5518545 0.1516973  
## 6 230.3030 3.3899 1 -0.7428681 0.4866494 1 0.5525464 0.1548966  
## PredictedProb  
## 1 0.3075737  
## 2 0.3105042  
## 3 0.3134499  
## 4 0.3164108  
## 5 0.3193867  
## 6 0.3223773

ggplot(newdata3, aes(x = gre, y = PredictedProb)) + geom\_ribbon(aes(ymin = LL, ymax = UL, fill = rank), alpha = .2) + geom\_line(aes(colour = rank), size=1)



with(mylogit, null.deviance - deviance)

## [1] 41.45903

#The degrees of freedom for the difference between the two models is equal to the number of predictor variables in the mode  
with(mylogit, df.null - df.residual)

## [1] 5

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

## [1] 7.578194e-08

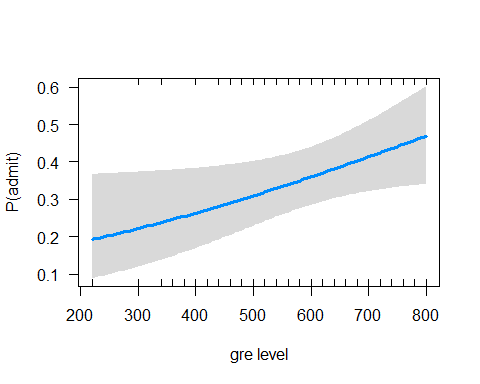
logLik(mylogit)

## 'log Lik.' -229.2587 (df=6)

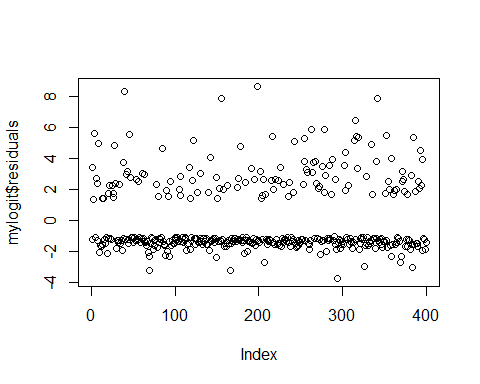
#The chi-square of 41.46 with 5 degrees of freedom and an associated p-value of less than 0.05 tells us that our model as a whole fits significantly better than an empty model  
  
# Pseudo R\_squared values and Likelyhood ratio test  
nagelkerke(mylogit)

## $Models  
##   
## Model: "glm, admit ~ gre + gpa + rank, binomial, mydata"  
## Null: "glm, admit ~ 1, binomial, mydata"   
##   
## $Pseudo.R.squared.for.model.vs.null  
## Pseudo.R.squared  
## McFadden 0.0829219  
## Cox and Snell (ML) 0.0984570  
## Nagelkerke (Cragg and Uhler) 0.1379960  
##   
## $Likelihood.ratio.test  
## Df.diff LogLik.diff Chisq p.value  
## -5 -20.73 41.459 7.5782e-08  
##   
## $Number.of.observations  
##   
## Model: 400  
## Null: 400  
##   
## $Messages  
## [1] "Note: For models fit with REML, these statistics are based on refitting with ML"  
##   
## $Warnings  
## [1] "None"

#The nagelkerke( ) function of rcompanion package provides three types of Pseudo R-squared value (McFadden, Cox and Snell, and Cragg and Uhler) and Likelihood ratio test results. The McFadden Pseudo R-squared value is the commonly reported metric for binary logistic regression model fit.The table result showed that the McFadden Pseudo R-squared value is 0.0829219, which indicates not a decent model fit.  
#probability plot  
# Probabilities of admit wrt gre  
visreg(mylogit, "gre", scale="response", rug=2, xlab="gre level",  
 ylab="P(admit)")

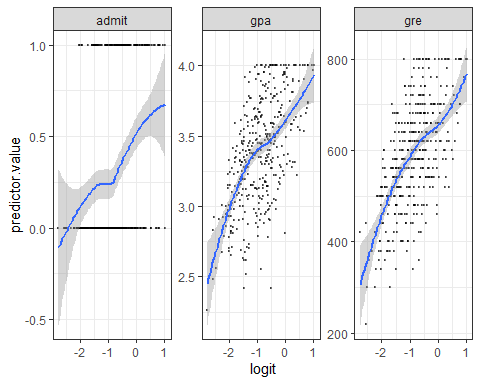


#In order to understand how the admit probabilities change with given values of independent variables, we generate the probability plots using visreg function. Here, we have plotted the gre in the x-axis and admit probabilities on the y-axis. The vertical rug lines indicate the density of observation along the x-axis. The dark band. along the blue line indicates a 95% confidence interval band.  
  
#Residual plot  
plot(mylogit$residuals)

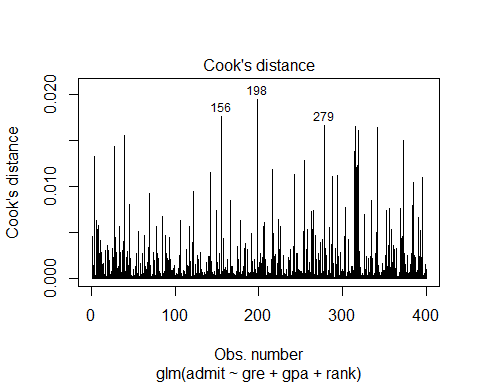


#diagnostics  
# Predict the probability (p) of admit positivity  
probabilities = predict(mylogit, type = "response")  
# Select only numeric predictors  
mydata2 = mydata %>%  
 dplyr::select\_if(is.numeric)   
predictors = colnames(mydata)  
# Bind the logit and tidying the data for plot  
mydata2 <- mydata2 %>%  
mutate(logit = log(probabilities/(1-probabilities))) %>%  
 gather(key = "predictors", value = "predictor.value", -logit)  
  
ggplot(mydata2, aes(logit, predictor.value))+  
 geom\_point(size = 0.5, alpha = 0.5) +  
 geom\_smooth(method = "loess") +   
 theme\_bw() +   
 facet\_wrap(~predictors, scales = "free\_y")

## `geom\_smooth()` using formula 'y ~ x'



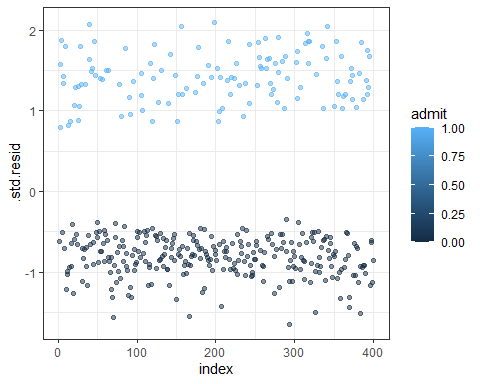
#The smoothed scatter plots show that variables gre, and gpa are all quite linearly associated with the admit outcome in logit scale.  
  
#Influential values are extreme individual data points that can alter the quality of the logistic regression model.  
  
#The most extreme values in the data can be examined by visualizing the Cook's distance values. Here we label the top 3 largest values:  
plot(mylogit, which = 4, id.n = 3)



#To check whether the data contains potential influential observations, the standardized residual error can be inspected. Data points with an absolute standardized residuals above 3 represent possible outliers and may deserve closer attention.  
# Extract model results  
mylogit.data = augment(mylogit) %>%   
 mutate(index = 1:n())   
  
mylogit.data %>% top\_n(3, .cooksd)

## # A tibble: 3 x 11  
## admit gre gpa rank .fitted .resid .std.resid .hat .sigma .cooksd index  
## <int> <int> <dbl> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int>  
## 1 1 580 2.86 4 -1.93 2.03 2.05 0.0149 1.08 0.0176 156  
## 2 1 400 3.23 4 -2.04 2.08 2.09 0.0147 1.07 0.0194 198  
## 3 1 680 3 4 -1.59 1.88 1.90 0.0195 1.08 0.0166 279

ggplot(mylogit.data, aes(index, .std.resid)) +   
 geom\_point(aes(color = admit), alpha = .5) +  
 theme\_bw()



mylogit.data %>%   
 filter(abs(.std.resid) > 3)

## # A tibble: 0 x 11  
## # ... with 11 variables: admit <int>, gre <int>, gpa <dbl>, rank <fct>,  
## # .fitted <dbl>, .resid <dbl>, .std.resid <dbl>, .hat <dbl>, .sigma <dbl>,  
## # .cooksd <dbl>, index <int>

#There is no influential observations in our data.  
#Multicollinearity  
#Multicollinearity corresponds to a situation where the data contain highly correlated predictor variables.  
#Multicollinearity is an important issue in regression analysis and should be fixed by removing the concerned variables. It can be assessed using the R function vif() [car package], which computes the variance inflation factors:  
car::vif(mylogit)

## GVIF Df GVIF^(1/(2\*Df))  
## gre 1.134377 1 1.065071  
## gpa 1.155902 1 1.075129  
## rank 1.025759 3 1.004248

#As a rule of thumb, a VIF value that exceeds 5 or 10 indicates a problematic amount of collinearity. In our example, there is no collinearity: all variables have a value of VIF well below 5.