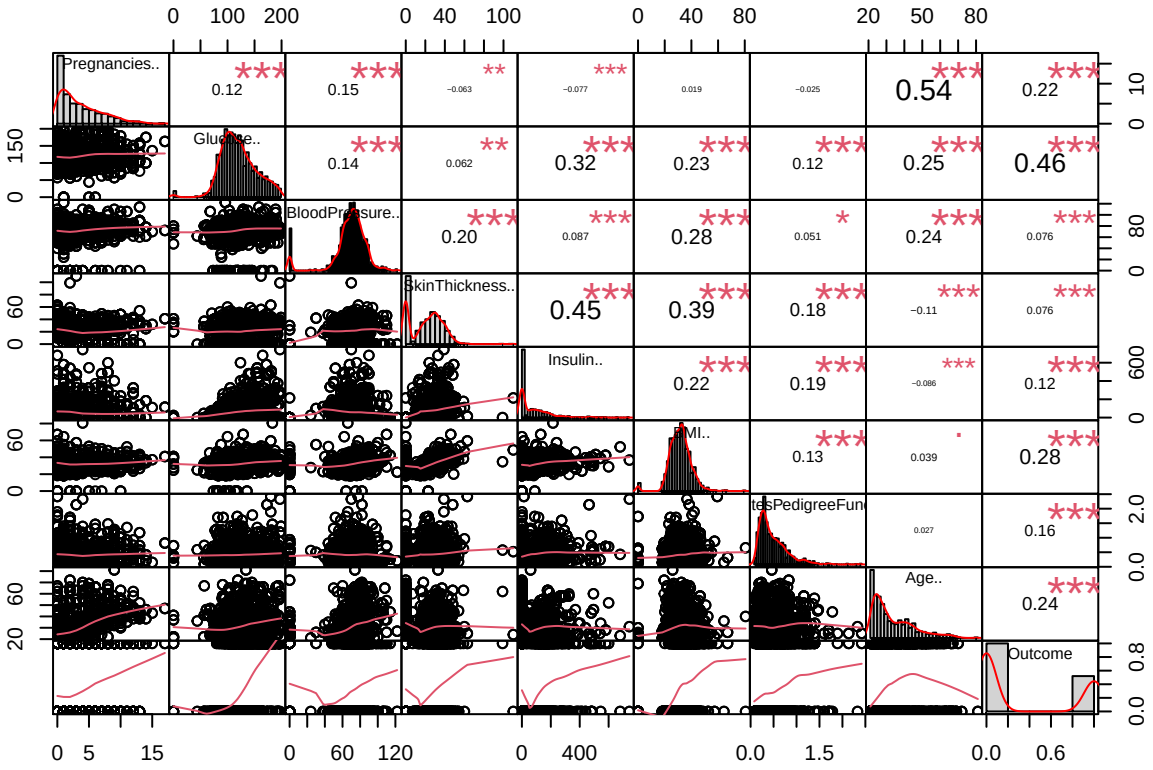
MiniProj_3

Wenxiong Lu

1a perform analysis by checking correlation graph, we can find that all of the predictors are related to the Outcome. Pairs like BMI and Pregnancies, DiabetesPedigreeFunction and Pregnancies, Age and DiabetesPedigreeFunction are not significantly related.



b

```
##
## Call:
## glm(formula = Outcome ~ ., family = binomial, data = db)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1942  -0.7256  -0.4473   0.7540   2.8979
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -8.0264511  0.4306345 -18.639  < 2e-16 ***
## Pregnancies..               0.1263845  0.0199997   6.319 2.63e-10 ***
## Glucose..                   0.0337202  0.0022258  15.150  < 2e-16 ***
## BloodPressure..            -0.0096446  0.0032441  -2.973  0.00295 **
```

```
## SkinThickness..             0.0005185  0.0042301   0.123  0.90244
## Insulin..                  -0.0012426  0.0005786  -2.148  0.03175 *
## BMI..                       0.0775549  0.0088819   8.732  < 2e-16 ***
## DiabetesPedigreeFunction..  0.8877583  0.1860275   4.772 1.82e-06 ***
## Age..                       0.0129414  0.0057020   2.270  0.02323 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2569.4  on 1999  degrees of freedom
## Residual deviance: 1914.3  on 1991  degrees of freedom
## AIC: 1932.3
##
## Number of Fisher Scoring iterations: 5
```

By checking z test, we can remove SkinThickness which is insiginificant and build the model using all other predictors.

c Summary of estimation of coefficients:

```
##
## Call:
## glm(formula = Outcome ~ Pregnancies.. + Glucose.. + BloodPressure.. +
##     Insulin.. + BMI.. + DiabetesPedigreeFunction.. + Age.., family = binomial,
##     data = db)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2028  -0.7253  -0.4454   0.7557   2.8980
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -8.0273146  0.4306244 -18.641  < 2e-16 ***
## Pregnancies..               0.1263707  0.0199944   6.320 2.61e-10 ***
## Glucose..                   0.0336810  0.0022020  15.296  < 2e-16 ***
## BloodPressure..            -0.0095806  0.0032013  -2.993  0.00276 **
## Insulin..                  -0.0012123  0.0005228  -2.319  0.02042 *
## BMI..                       0.0778743  0.0084946   9.167  < 2e-16 ***
## DiabetesPedigreeFunction..  0.8894946  0.1855205   4.795 1.63e-06 ***
## Age..                       0.0128944  0.0056879   2.267  0.02339 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2569.4  on 1999  degrees of freedom
## Residual deviance: 1914.3  on 1992  degrees of freedom
## AIC: 1930.3
##
## Number of Fisher Scoring iterations: 5
```

Final model: Outcome = x0+(Pregnancies)x1+(Glucose)x2+(BloodPressure)x3+(Insulin)x4+(BMI)5+(DiabetesPedigreeFunct

Confidence Intervals:

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

```
##                                   2.5 %        97.5 %
## (Intercept)                -8.889630784 -7.2009252668
## Pregnancies..               0.087447559  0.1658700222
## Glucose..                   0.029435255  0.0380709843
## BloodPressure..            -0.015885768 -0.0033221648
## Insulin..                  -0.002241105 -0.0001893038
## BMI..                       0.061474284  0.0947952879
## DiabetesPedigreeFunction..  0.527470753  1.2549028449
## Age..                       0.001711033  0.0240290378
```

Training Error rate

```
## [1] 0.216
```

Insulin.. -0.0012123
BMI.. 0.0778743 which means: (1)Holding all other variable the same, increase Insulin by 1-unit brings exp(-0.0012123)=0.9987884 change (or 0.0012116 decrease) in Outcome. (2)Holding all other variable the same, increase BMI by 1-unit brings exp(0.0778743)=1.080987 change (or 0.080987 increase) in Outcome.

#####2 a

Error Rate:

```
## [1] 0.216
```

Sensitivity

```
## [1] 0.740458
```

Specificity:

```
## [1] 0.799458
```

b my own code with test error rate

```
#(b)
n<-length(Outcome)
Q<-rep(0,n)
tfs<-glm.pred2!=Outcome
Q[tfs==TRUE]=1
cv1<-(sum(Q))/n
cv1
```

```
## [1] 0.216
```

c

```
## [1] 0.2195
```

we can find the results are very close.

d estimate the test error rate using LOOCV

```
## [1] 0.2185
```

(e,f) estimate results for LDA then QDA using LOOCV

```
## [1] "loocv for LDA: 0.2215"
```

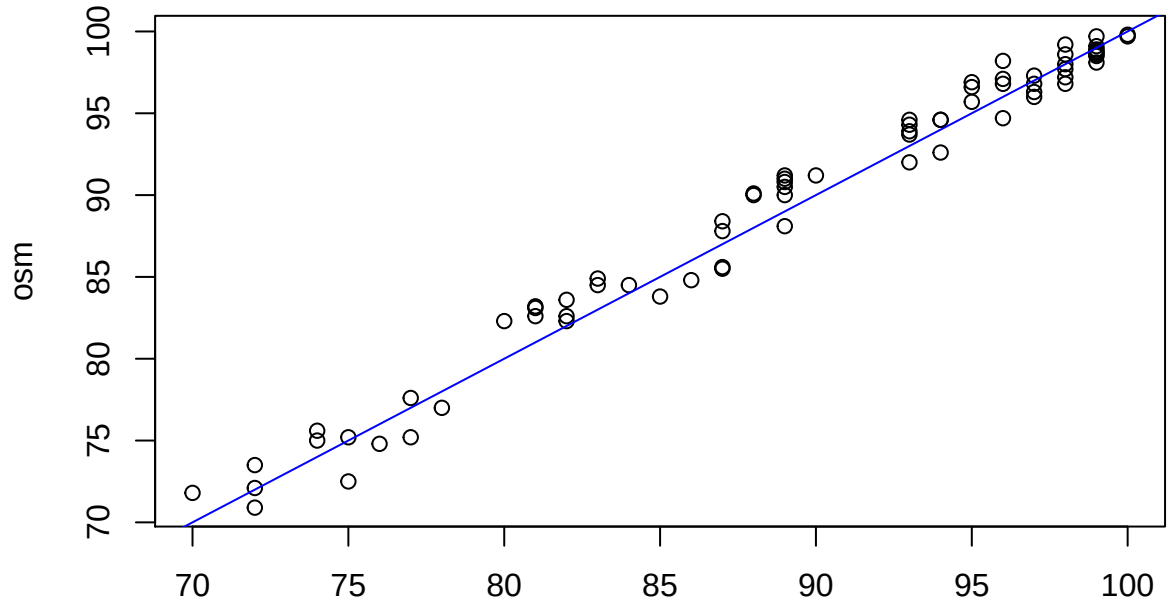
```
## [1] "loocv for QDA: 0.2445"
```

g for KNN, the following results are k = 3, 6, 9 respectively.

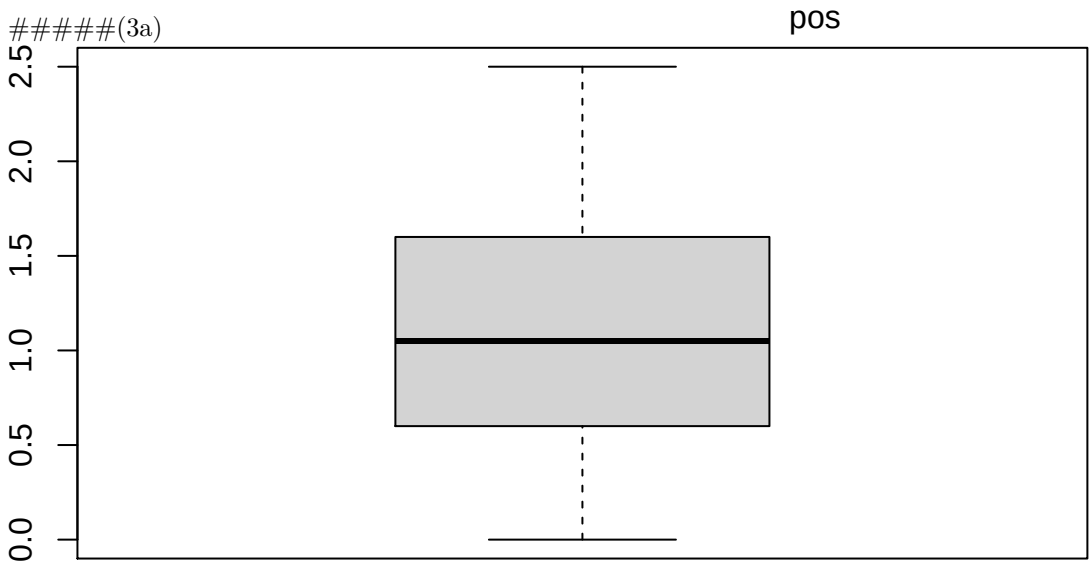
```
## [1] "0.2095 0.1735 0.2275"
```

k=6 has the lowest test error rate

(h) by compare the results, we can find the lowest test error rate is from knn with k=6. So Knn=6 is recommended.



#####(3a)



From the scatterplot, we find most of the data lies on the 45 degree line evenly and the boxplot shows that most of these data are between 0.5 to 1.5. So we can suggest presumably that there are agreement between the two datas.

b

D is the absolute value of differences. After sorting D increasingly, theta is the 90 percentile extreme difference. If theta is consider low, we can make sure at lease 90% of the difference are small. So the smaller this theta is, the closer the two data are in overall picture.

c theta estimate:

```
## [1] 65
```

```
## [1] 2
```

d my own code for bootstrap

estimated (90%) theta:

```
## [1] 2
```

bias:

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

standard error:

```
## [1] 1.124208
```

95%upper confidnece bound for theta

```
## [1] 4.230673
```

e using boot package

```
## [1] 2
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = sD, statistic = mb, R = 1000)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*        2  0.0056   0.1293266
```

Both outputs are very close.

(f):From my own code the upper confidence bound, we can find most of the differences are within the this bound. The bias is small with relatively low standard error. So there is agreement between the two sets of data which lead to the same conclusion by using statistics calculated from boot package. Two methods agree well enough for interchange.

Section 2:

setwd('/Users/wenxionglu/Documents/GraduateCourse/6340 Stats/proj3') db<-read.csv('diabetes.csv',header=TRUE)

db$Outcome < -as.factor(db$Outcome) attach(db) is.factor(Outcome) library(PerformanceAnalytics) library(MASS) library(lmtest) library(car) library(ISLR) library(boot) library(caret) library(e1071) set.seed(1)

1 (a) chart.Correlation(db) #matrix of scatterplot dev.off()

(b) mod.1 = glm(Outcome~., data = db,family = binomial) summary(mod.1) For each predictor, fit a logistic regression model to predict the response.

(c) mod.2<-update(mod.1,~.-SkinThickness..) summary(mod.2) confint(mod.2) contrasts(Outcome) prob<-predict(mod.2,type = 'response') glm.pred=rep(0,2000) glm.pred[prob>0.5]=1

sum(glm.pred!=Outcome)/2000 #the training error rate

2.(a) summary(mod.1) prob2<-predict(mod.1, type='response',data=db) glm.pred2<-rep(0,2000) glm.pred2[prob2>0.5]=1 table(glm.pred2,Outcome) sum(glm.pred2!=Outcome)/2000 # the trainig error rate

388/(388+136) #sensitivity 1180/(1180+296) #specificity (b) n<-length(Outcome) Q<-rep(0,n) tfs<-glm.pred2!=Outcome Q[tfs==TRUE]=1 cv1<-(sum(Q))/n cv1 (c)

cv2 = train(as.factor(Outcome) ~ ., db, method="glm", family='binomial',metric="Accuracy", trControl = trainControl(method = "LOOCV")) err2<-1-cv2$results$Accuracy

err2 (d) cv3 = train(Outcome ~ .-SkinThickness.., data=db, method="glm", family='binomial',metric='Accuracy', trControl = trainControl(method = "LOOCV")) err3<-1-cv3*results*Accuracy err3 (e,f) cv4 = train(as.factor(Outcome) ~ ., data=db, method="lda", trControl = trainControl(method = "loocv"),prior=c(0.658,0.342)) err4<-1-cv4*results*Accuracy #error rate of loocv err4 cv5 = train(as.factor(Outcome) ~ ., data=db, method="qda", trControl = trainControl(method = "loocv"),prior=c(0.658,0.342)) err5<-1-cv5*results*Accuracy #error rate of loocv err5

(g) cv6 = train(as.factor(Outcome) ~ ., data=db, method="knn", trControl = trainControl(method = "loocv"),prior=c(0.658,0.342)) err6<-1-cv6*results*Accuracy #error rate of loocv err6 #for k = 5, 6, 9

(h) recommend knn with k=6 in this case

3.(a) os<-read.table('oxygen_saturation.txt',header=TRUE) attach(os) D<-abs(pos-osm) x<-plot(pos,osm) abline(a=0,b=1,col='blue') boxplot(D)

(c) sD<-sort(D) #sorting and the absolute value of Difference round(length(sD)/(1/0.9)) #90% is at idx 65 theta<-sD[round(length(sD)*0.9)] theta

(d)

set.seed(1) my bootfunction for getting mean mbs<-function(sD,n){ x<-sD[n] return(sort(x)) }

mset<-mbs(sD,sample(72,1000,replace = TRUE)) x<-mset[round(length(mset)*0.9)] x bias<-2-x bias se<-sqrt(1/(1000-1)*sum((x-mset)^2)) se upp<-x+(qt(0.025,df=100-1,lower.tail = FALSE)*se) #upper confidence bound upp

(e) mb<-function(sD,n){ x<-sD[n] return(sort(x)[round(length(x)*0.9)]) } mb(sD,sample(72,1000,replace = TRUE))

boot(sD,mb,R=1000)