Logistic Regression

```
churn_input = as.data.frame(  read.csv("churn.csv")   )
head(churn_input)

sum(churn_input$Churned)

Churn_logistic1 <- glm (Churned~Age + Married + Cust_years + Churned_contacts,
                  data=churn_input, family=binomial(link="logit"))
summary(Churn_logistic1)

Churn_logistic2 <- glm (Churned~Age + Married +  Churned_contacts,
                  data=churn_input, family=binomial(link="logit"))
summary(Churn_logistic2)

Churn_logistic3 <- glm (Churned~Age + Churned_contacts,
                  data=churn_input, family=binomial(link="logit"))
summary(Churn_logistic3)

# Deviance and the Log Likelihood Ratio Test

# Using the residual deviance from Churn_logistics2 and Churn_logistic3
# determine the significance of the computed test statistic
summary(Churn_logistic2)
pchisq(.9 , 1, lower=FALSE)

# Receiver Operating Characteristic (ROC) Curve

install.packages("ROCR")     #install, if necessary
library(ROCR)

pred = predict(Churn_logistic3, type="response")
predObj = prediction(pred, churn_input$Churned )

rocObj = performance(predObj, measure="tpr", x.measure="fpr")
aucObj = performance(predObj, measure="auc")

plot(rocObj, main = paste("Area under the curve:", round(aucObj@y.values[[1]] ,4)))

# extract the alpha(threshold), FPR, and TPR values from rocObj
alpha <- round(as.numeric(unlist(rocObj@alpha.values)),4)
fpr <- round(as.numeric(unlist(rocObj@x.values)),4)
tpr <- round(as.numeric(unlist(rocObj@y.values)),4)

# adjust margins and plot TPR and FPR
par(mar = c(5,5,2,5))
plot(alpha,tpr, xlab="Threshold", xlim=c(0,1), ylab="True positive rate", type="l")
par(new="True")
plot(alpha,fpr, xlab="", ylab="", axes=F, xlim=c(0,1), type="l" )
axis(side=4)
mtext(side=4, line=3, "False positive rate")
text(0.18,0.18,"FPR")
text(0.58,0.58,"TPR")

i <- which(round(alpha,2) == .5)
paste("Threshold=" , (alpha[i]) , " TPR=" , tpr[i] , " FPR=" , fpr[i])
```

```
i <- which(round(alpha,2) == .15)
paste("Threshold=" , (alpha[i]) , " TPR=" , tpr[i] , " FPR=" , fpr[i])
```

Decision Trees

```
install.packages("rpart.plot")

library("rpart")
library("rpart.plot")

# Read the data
setwd("~/DSA(R)")
banktrain <- read.table("bank-sample.csv",header=TRUE,sep=",")

## drop a few columns to simplify the tree

drops<-c("age", "balance", "day", "campaign", "pdays", "previous", "month")
banktrain <- banktrain [,!(names(banktrain) %in% drops)]
summary(banktrain)

# Make a simple decision tree by only keeping the categorical variables

fit <- rpart(subscribed ~ job + marital + education + default + housing + loan + contact + poutcome,
        method="class",
        data=banktrain,
        control=rpart.control(minsplit=1),
        parms=list(split='information'))
summary(fit)

# Plot the tree

rpart.plot(fit, type=4, extra=2, clip.right.labs=FALSE, varlen=0, faclen=3)

# section 7.1.2 The General Algorithm

# Entropy of coin flips
x <- sort(runif(1000))
y <- data.frame(x=x, y=-x*log2(x)-(1-x)*log2(1-x))
plot(y, type="l", xlab="P(X=1)", ylab=expression("H"["X"]))
grid()

# include a numeric variable "duration" into the model
fit <- rpart(subscribed ~ job + marital + education + default + housing + loan + contact + duration + poutcome,
        method="class",
        data=banktrain,
        control=rpart.control(minsplit=1),
        parms=list(split='information'))
summary(fit)

# Plot the tree
rpart.plot(fit, type=4, extra=2, clip.right.labs=FALSE, varlen=0, faclen=3)

# Predict
newdata <- data.frame(job="retired",
            marital="married",
            education="secondary",
            default="no",
            housing="yes",
            loan="no",
            contact = "cellular",
```

```
                    duration = 598,
                    poutcome="unknown")
newdata
predict(fit,newdata=newdata,type=c("class"))

# section 7.1.5 Decision Trees in R

library("rpart") # load libraries
library("rpart.plot")
```

K-Means

```
library(plyr)
library(ggplot2)
library(cluster)
library(lattice)
library(graphics)
library(grid)
library(gridExtra)
#import the student grades
grade_input=as.data.frame(read.csv("grades_km_input.csv"))
kmdata_orig=as.matrix(grade_input[,c("Student","English","Math","Science")])
kmdata<-kmdata_orig[,2:4]
kmdata[1:10,]
wss<-numeric(15)

for(k in 1:15) wss[k]<-sum(kmeans(kmdata,centers=k,nstart=25) $ withinss)
plot(1:15,wss,type="b",xlab="Number of clusters",ylab="Within Sum of Squares")
km=kmeans(kmdata,3,nstart=25)
km
c(wss[3],sum(km$withinss))
df=as.data.frame(kmdata_orig[,2:4])
df$cluster=factor(km$cluster)
km$cluster
centers=as.data.frame(km$centers)
g1= ggplot(data=df,aes(x=English,y=Math,color=cluster)) +
  geom_point() + theme(legend.position="right") +
  geom_point(data=centers,aes(x=English,y=Math,color=as.factor(c(1,2,3))),
           size=10,alpha=.3,show.legend = FALSE)
g2 =ggplot(data=df,aes(x=English,y=Science,color=cluster )) +
  geom_point() +
  geom_point(data=centers,aes(x=English,y=Science,color=as.factor(c(1,2,3))),
           size=10,alpha=.3,show.legend=FALSE)
g3 =ggplot(data=df,aes(x=Math,y=Science,color=cluster)) +
  geom_point() +
  geom_point(data=centers,aes(x=Math,y=Science, color=as.factor(c(1,2,3))),
           size=10,alpha=.3,show.legend = FALSE)
tmp = ggplot_gtable(ggplot_build(g1))
library(grid)
library(gridExtra)
grid.arrange(g1,g2,g3,ncol=1,top="High School Student Cluster Analysis")
```

Linear Regression

```
income_input=as.data.frame(read.csv("income.csv"))
income_input[1:10,]

summary(income_input)

library(lattice)

splom(~income_input[c(2:5)],  groups=NULL, data=income_input,
     axis.line.tck = 0,
     axis.text.alpha = 0)

results <- lm(Income~Age + Education + Gender, income_input)
summary(results)

results2 <- lm(Income ~ Age + Education, income_input)
summary(results2)

# this code from the text is for illustrative purposes only
# the income_input variable does not contain the U.S. states

results3 <- lm(Income~Age + Education,
            + Alabama,
            + Alaska,
            + Arizona,
            .
            .
            .
            + WestVirginia,
            + Wisconsin,
            income_input)

# compute confidence intevals for the model parameters

confint(results2, level = .95)

# compute a confidence interval on the expected income of a person
Age <- 41
Education <- 12
new_pt <- data.frame(Age, Education)

conf_int_pt <- predict(results2, new_pt, level=.95, interval="confidence")
conf_int_pt

# compute a prediction interval on the income of the same person
pred_int_pt <- predict(results2, new_pt, level=.95, interval="prediction")
pred_int_pt

with(results2, {
  plot(fitted.values, residuals,ylim=c(-40,40) )
  points(c(min(fitted.values),max(fitted.values)), c(0,0), type = "l")})

hist(results2$residuals, main="")

qqnorm(results2$residuals, ylab="Residuals", main="")
qqline(results2$residuals)
```