HW6 Solution CS6220-Data Mining

Problem 1 (a)

```r
#read data
mydata= read.csv("performance.csv", header = F)

#Check the data
head(mydata)
```

```
##   V1  V2
## 1  0 474
## 2  0 432
## 3  0 453
## 4  1 481
## 5  1 619
## 6  0 584
```

```r
# Assign column names to the data
names(mydata) <- c("Y", "X")

# Check for missing values
sum(is.na(mydata))
```

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

```r
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
```

```r
# Sort data based on emotional stability score
mydata =arrange(mydata, X)

# Calculate the median of emotional stability
my2groups = as.factor(mydata$X < median(mydata$X))
table(my2groups)
```

```
## my2groups
## FALSE  TRUE
##    14    13
```

```r
# Classify subjects by emotional stability and performance
mydata.table.2groups <- table(emStability=my2groups, perf=mydata$Y)
mydata.table.2groups
```

```
##            perf
## emStability  0  1
##       FALSE  4 10
##       TRUE   9  4
```

```r
# Compare proportions
# Not that prop.test compares columns, and our data compare rows. Therefore we transpose the matrix pri
prop.test(t(mydata.table.2groups))
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  t(mydata.table.2groups)
## X-squared = 2.9835, df = 1, p-value = 0.08412
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.82565184  0.01246502
## sample estimates:
##    prop 1    prop 2
## 0.3076923 0.7142857
```

```r
10/(4+10)
```

```
## [1] 0.7142857
```

```r
4/(9+4)
```

```
## [1] 0.3076923
```

```r
# Double-check the proportions
```

The group with the emotional stability higher than median, has more people who able to do the job; 71.4% vs 30.7% The 95% confidence interval for difference between proportions does not contain 0 (equivalently, the p-value exceeds 0.05), therefore there is no evidence against H_0 of the equality of the two proportions.

(b)

```r
#calculate the quartiles
my4groups = as.factor(1 * (mydata$X <= quantile(mydata$X, 0.25)) +
                      2 * (mydata$X > quantile(mydata$X, 0.25) &  mydata$X <=  quantile(mydata$X, 0.5))
                      3 * (mydata$X > quantile(mydata$X, 0.5) &  mydata$X <=  quantile(mydata$X, 0.75))
                      4 * (mydata$X > quantile(mydata$X, 0.75)))
table(my4groups)
```

```
## my4groups
## 1 2 3 4
## 7 7 6 7
```

```r
#show the data in the table
mydata.table.4groups = table(emStability=my4groups, perf=mydata$Y)
mydata.table.4groups
```

```
##            perf
## emStability 0 1
##           1 6 1
##           2 3 4
##           3 3 3
##           4 1 6
```

Pearson X^2 test

```r
summary(mydata.table.4groups)
```

```
## Number of cases in table: 27
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 7.259, df = 3, p-value = 0.0641
##  Chi-squared approximation may be incorrect
```

The p-value exceeds 0.05, therefore there is no evidence against H_0 of no association between emotional stability and performance. However note that the counts in the individual cells are very small (many are $<$ 5). The χ^2 approximation is very poor in this case, and the results are not reliable.

We conclude that partitioning the subjects into more groups gives us a more detailed view of the counts, but the individual counts are small and we are more likely to overfit.

(c)

```r
#logestic regression model
logistic.fit<- glm(Y ~ X, family=binomial, data=mydata)
```

The assumptions are: Y are Benoulli random variables, independent, and $P\{Y = 1\}$ is related to X according to the logistic function

$$P\{Y = 1|X\} = \frac{1}{1 + e^{-(\beta_0+\beta_1 X)}}$$

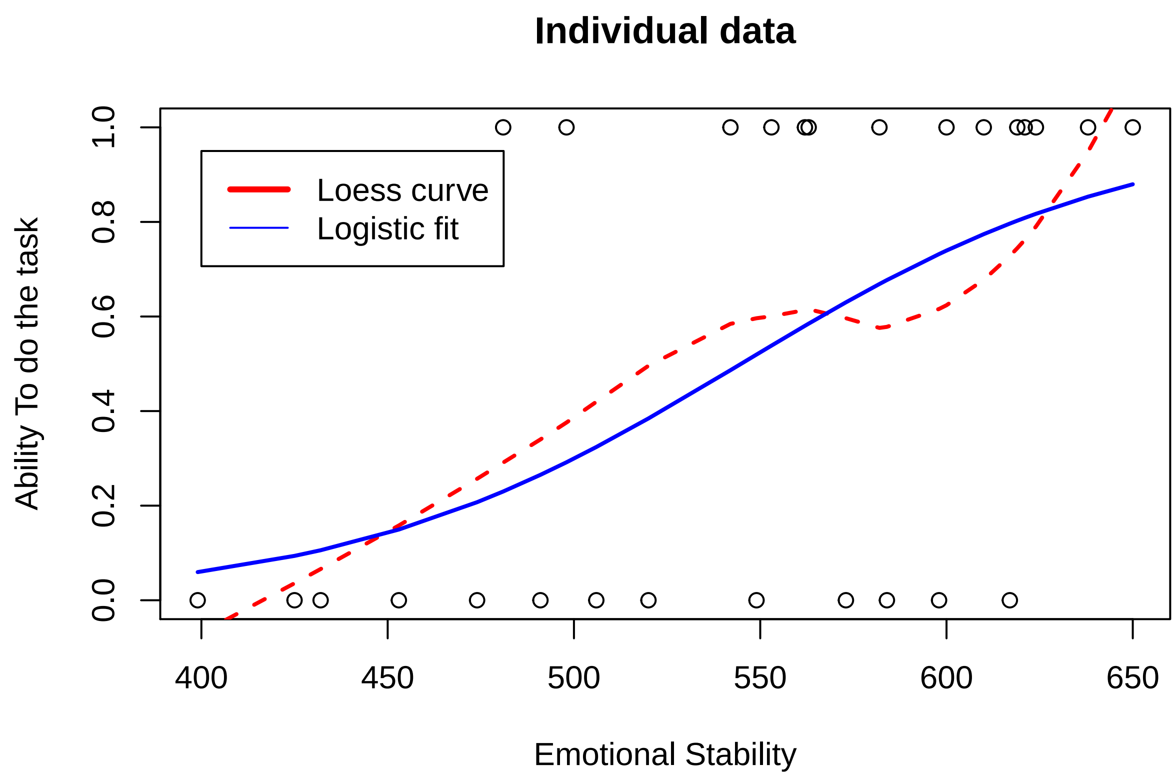
(d)

```r
#plot the individual data
plot(Y ~ X, mydata, xlab ="Emotional Stability", ylab = "Ability To do the task", main="Individual data

#plot the loess model
lines(mydata$X, predict(loess(Y ~ X, mydata),
data.frame(X=mydata$X)), lty = 2, lwd=2, col = 'red')

#plot the logestic model
lines(mydata$X, predict(logistic.fit,
data.frame(X = mydata$X), type = "resp"), lwd = 2, col= "blue")

#add the legends
legend(400, 0.95,c("Loess curve" ,"Logistic fit"),lty=c(1,1),lwd=c(3, 1),col=c("red", "blue"))
```

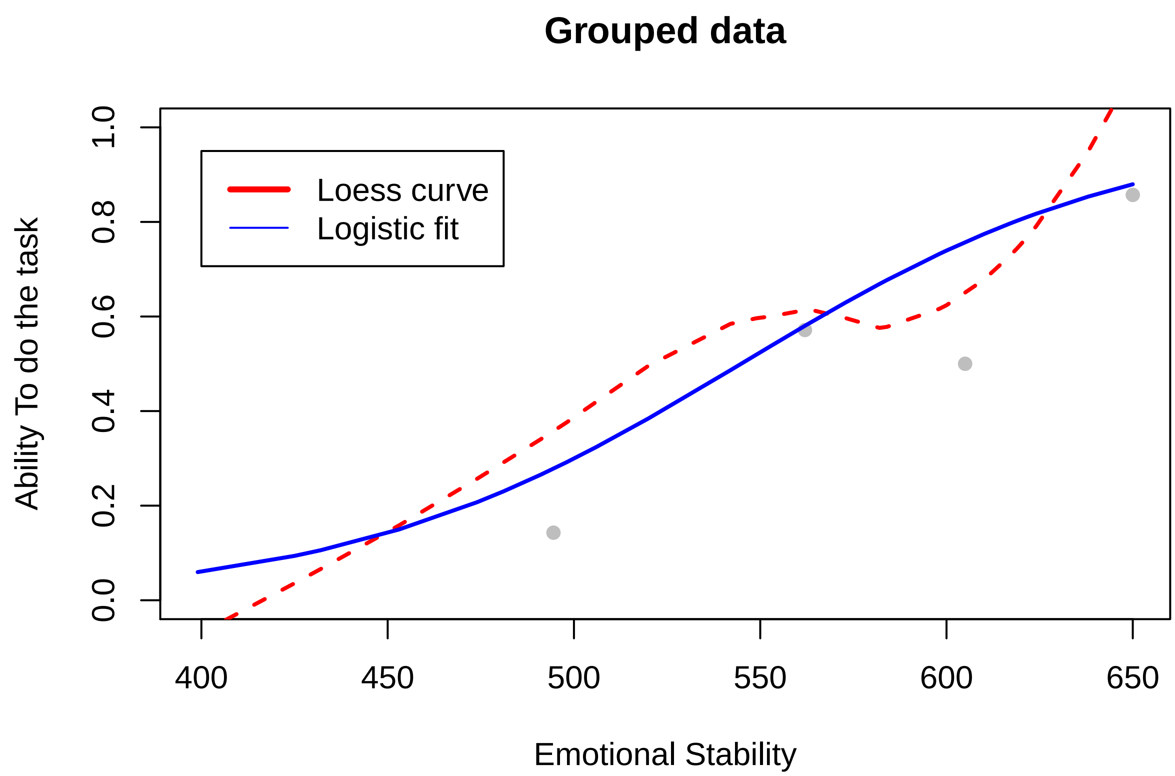



```
#plot the grouped data
plot(quantile(mydata$X, c(0.25, 0.5, 0.75, 1)), mydata.table.4groups[,2]/apply(mydata.table.4groups, 1,
xlab ="Emotional Stability", ylab = "Ability To do the task", main="Grouped data")

#plot the loess model
lines(mydata$X, predict(loess(Y ~ X, mydata),
data.frame(X=mydata$X)), lty = 2, lwd=2, col = 'red')

#plot the logestic model
lines(mydata$X, predict(logistic.fit,
data.frame(X = mydata$X), type = "resp"), lwd = 2, col= "blue")

#add the legends
legend(400, 0.95,c("Loess curve" ,"Logistic fit"),lty=c(1,1),lwd=c(3, 1),col=c("red", "blue"))
```



The logistic regression is generally plausible, however more adjustments may be needed.

(e)

```
summary(logistic.fit)
```

```
##
## Call:
## glm(formula = Y ~ X, family = binomial, data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7845  -0.8350   0.5065   0.8371   1.7145
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.308925   4.376997  -2.355   0.0185 *
## X             0.018920   0.007877   2.402   0.0163 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 37.393  on 26  degrees of freedom
## Residual deviance: 29.242  on 25  degrees of freedom
## AIC: 33.242
##
## Number of Fisher Scoring iterations: 4
```

The p-values are smaller for alfa = 0.05 and the odds of ability to do the job increases for every 0.018920 unit raise in emotional stability score.

The model is more consize than in the previous case and provides a stronger evidence for association. However the conclusion is also approximate.

(f)

```
beta1 = logistic.fit$coef[2]
beta1
```

```
##          X
## 0.01891983
```

```
exp.beta1 = exp(beta1)
exp.beta1
```

```
##      X
## 1.0191
```

Confidence Interval for $\beta 1$: $\beta 1 \pm z^{1-0.05/2} \cdot SE(\beta 1)$

So, confidence Interval for β_1

```
c(beta1 - qnorm(1-0.05/2) * 0.007877, beta1 + qnorm(1-0.05/2) * 0.007877)
```

```
##           X           X
## 0.003481193 0.034358466
```

$\hat{\beta}_1$ estimates the log (odds ratio) of performance, that follows a unit change in the emotional stability score. The confidence interval does not contain 0, indicating evidence against H_0 of no association. However the difference is very minor.

And confidence Interval for e^{β_1}:

```
CI = exp(c(beta1 - qnorm(1-0.05/2) * 0.007877, beta1 + qnorm(1-0.05/2) * 0.007877))
CI
```

```
##        X        X
## 1.003487 1.034956
```

$e^{\hat{\beta}_1}$ estimates the odds ratio of performance, that follows a unit change in the emotional stability score. The confidence interval does not contain 1, but again the strength of the evidence is quite week.

(g)

$P\{Y = 1|X = 550\} = 1/1 + e^{-}(\beta 0 + \beta 1 * x)$:

```
predict(logistic.fit, newdata=data.frame(X=550), type='response')
```

```
##         1
## 0.5242263
```

```r
#equivalently,
1/(1 + exp(10.308925 - 0.018920*550))
```

```
## [1] 0.5242497
```

(h)

```r
#calculate the probabilities
glm.probs=predict(logistic.fit, type="response")

#Confusion matrix
table(truth=mydata$Y, decision=glm.probs > 0.5)
```

```
##      decision
## truth FALSE TRUE
##     0     8    5
##     1     3   11
```

```r
#Sensitivity:
11/(11+3)
```

```
## [1] 0.7857143
```

```r
#pecificity:
8/(8+5)
```

```
## [1] 0.6153846
```

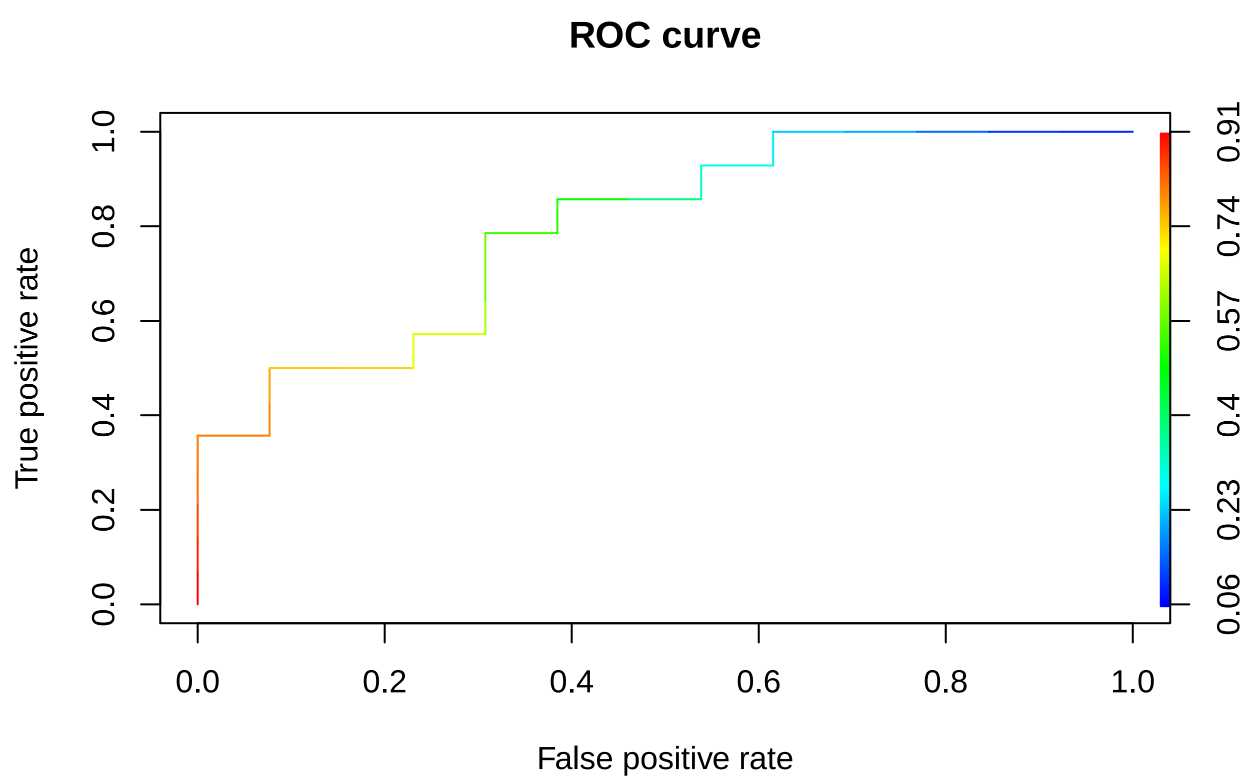
So the model could predict correctly the employees who *can't* make the job 61% of the times and the employees who *can* make the job 78% of the times.

(i)

```r
library(ROCR)
```

```
## Loading required package: gplots
##
## Attaching package: 'gplots'
##
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
pred <- prediction(glm.probs, labels=mydata$Y)
performance <- performance(pred, "tpr", "fpr")
# plotting the ROC curve
plot(performance, colorize=T, main="ROC curve")
```




```r
# Area under the curve
unlist(attributes(performance(pred, "auc"))$y.values)
```

```
## [1] 0.7967033
```

The ROC curve summarizes the sensitivity and the specificity of the prediction over all the probability cutoffs.

The area under curve is considerable so the model works well for this dataset. However, this performance is evaluated on the training set, and is likely optimistic. An independent evaluation on the validation set is needed for an unbiased characterization of the performance.

Problem 2

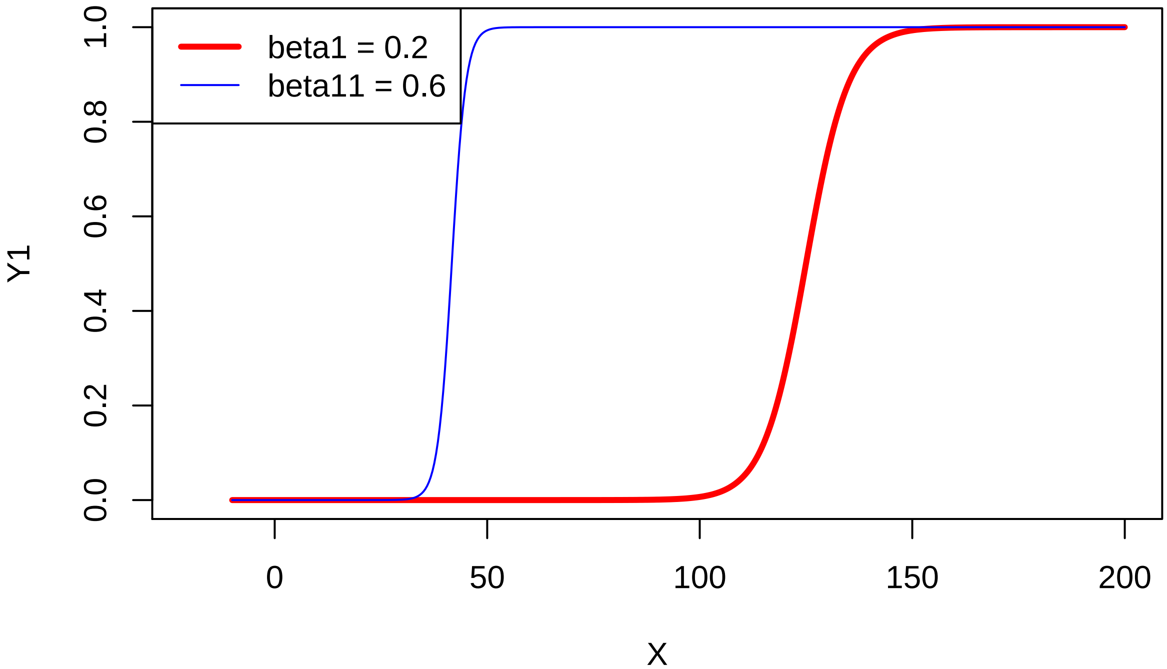
(a)

```r
X <- seq(from=-10, to=200, length=500)
```

$Y = 1/(1 + e^{-}\{\beta_0 + \beta_1 \cdot X\})$

```r
beta0 = -25
beta1 = 0.2
beta11 = 0.6
Y1 = 1/ ( 1+ exp(-1*(beta0 + beta1*X)) )
Y11 = 1/ ( 1+ exp(-1*(beta0 + beta11*X)) )
```

```r
plot(X,Y1, type ="l", col = "red", lwd = 3, xlim=c(-20, 200), ylim=c(0,1))
lines(X, Y11, col = "blue")
legend("topleft", c("beta1 = 0.2" ,"beta11 = 0.6"),
lty=c(1,1), lwd=c(3, 1), col=c("red", "blue"))
```



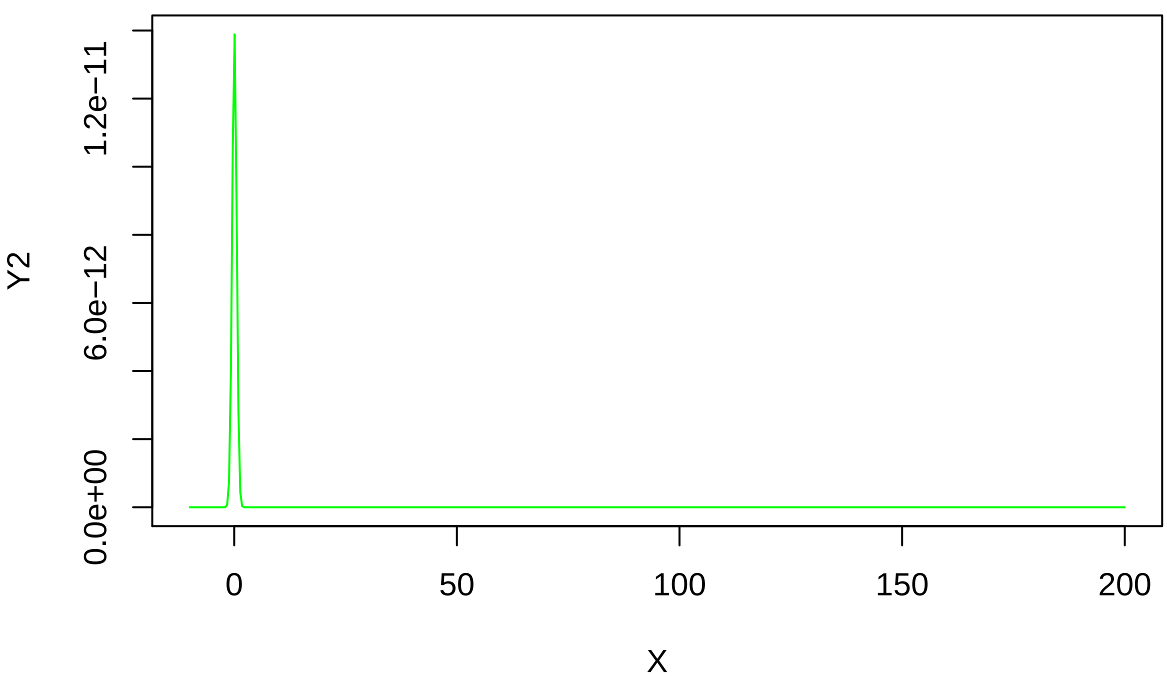
Stronger positive correlation means faster change in y as x change which could be seen in the graph.

(b)

$Y = 1/(1 + e^-\{\beta_0 + \beta_1 \cdot X + \beta_2 \cdot X^2\})$

```
beta0 = -25
beta1 = 0.2
beta2 = -2
Y2 = 1/ ( 1+ exp(-1*(beta0 + beta1*X + beta2*X^2)) )

plot(X, Y2, type ="l", col = "green")
```



Adding the X^2 term with the negative parameter expresses a non-monotoneous relationship between X and Y.