1. Orange Juice Data

(a) Apparenttly, Purchase and StoreID variables are qualitative and need to be transformed to factor.

```
> summary(ojdata[train,])
 Purchase WeekofPurchase  StoreID     PriceCH          PriceMM           DiscCH       
 CH:317   Min.   :227.0   1: 74   Min.   :1.690   Min.   :1.690   Min.   :0.00000  
 MM:218   1st Qu.:239.5   2:116   1st Qu.:1.790   1st Qu.:2.090   1st Qu.:0.00000  
          Median :256.0   3:108   Median :1.860   Median :2.090   Median :0.00000  
          Mean   :254.0   4: 68   Mean   :1.866   Mean   :2.084   Mean   :0.04723  
          3rd Qu.:267.0   7:169   3rd Qu.:1.990   3rd Qu.:2.180   3rd Qu.:0.00000  
          Max.   :278.0           Max.   :2.090   Max.   :2.290   Max.   :0.50000  
     DiscMM        SpecialCH SpecialMM    LoyalCH          SalePriceMM     SalePriceCH   
 Min.   :0.0000   0:460     0:459     Min.   :0.000017   Min.   :1.190   Min.   :1.390  
 1st Qu.:0.0000   1: 75     1: 76     1st Qu.:0.320000   1st Qu.:1.690   1st Qu.:1.750  
 Median :0.0000                       Median :0.600000   Median :2.090   Median :1.860  
 Mean   :0.1191                       Mean   :0.560588   Mean   :1.965   Mean   :1.819  
 3rd Qu.:0.2000                       3rd Qu.:0.854584   3rd Qu.:2.180   3rd Qu.:1.890  
 Max.   :0.8000                       Max.   :0.999947   Max.   :2.290   Max.   :2.090  
   PriceDiff      
 Min.   :-0.6700  
 1st Qu.: 0.0000  
 Median : 0.2400  
 Mean   : 0.1459  
 3rd Qu.: 0.3000  
 Max.   : 0.6400  
```

(b) The intercept is 1.7096, and coefficient for covariates are listed below. We can see that coefficient of WeekofPurchase, SpecialMM and StoreID2 are relatively close to 0, which means they have weak explanation to the target variable Purchase; while coefficient of PrichCH, PriceMM, DiscCH, DiscMM and LoyalCH are relatively far away from 0, which means they have strong explanation power.

And SalePriceMM, SalePriceCH and PriceDiff has na coefficients because of singularities

```
Call:
glm(formula = Purchase ~ ., family = binomial, data = ojdata[train, 
    ])

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5085  -0.5741  -0.2363   0.5150   2.7720  

Coefficients: (3 not defined because of singularities)
                   Estimate Std. Error z value Pr(>|z|)    
(Intercept)         1.70964    2.69235   0.635  0.52543    
WeekofPurchase      0.01310    0.01625   0.806  0.42034    
PriceCH             3.70726    2.66414   1.392  0.16406    
PriceMM            -4.31098    1.32790  -3.246  0.00117 ** 
DiscCH             -1.84703    1.55225  -1.190  0.23408    
DiscMM              2.46123    0.75908   3.242  0.00119 ** 
SpecialCH1         -0.75005    0.52243  -1.436  0.15109    
SpecialMM1          0.06780    0.41082   0.165  0.86892    
LoyalCH            -6.26312    0.58487 -10.709  < 2e-16 ***
SalePriceMM              NA         NA      NA       NA    
SalePriceCH              NA         NA      NA       NA    
PriceDiff                NA         NA      NA       NA    
`ojdata$StoreID2`  -0.05185    0.39117  -0.133  0.89455    
`ojdata$StoreID3`   0.34486    0.54369   0.634  0.52589    
`ojdata$StoreID4`  -0.70485    0.61150  -1.153  0.24906    
`ojdata$StoreID7`  -0.58866    0.41574  -1.416  0.15680    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 723.24  on 534  degrees of freedom
Residual deviance: 409.75  on 522  degrees of freedom
AIC: 435.75

Number of Fisher Scoring iterations: 5
```

(c) The best lambda is 0.01. The coefficients of the final model are listed as below.

```
> bestlam
[1] 0.01
> coef(lasso.mod)
17 x 1 sparse Matrix of class "dgCMatrix"
                               s0
(Intercept)          2.765636183
(Intercept)          .
WeekofPurchase       0.001376935
PriceCH              .
PriceMM              .
DiscCH               .
DiscMM               .
SpecialCH           -0.345163328
SpecialMM            0.024567878
LoyalCH             -5.721178509
SalePriceMM          .
SalePriceCH          .
PriceDiff           -2.232142779
`ojdata$StoreID2`    .
`ojdata$StoreID3`    0.193286456
`ojdata$StoreID4`   -0.231140496
`ojdata$StoreID7`   -0.482214096
```

(d) The classification error on the training data is about 0.1645.

```
> lda.pred=predict(lda.fit, ojdata[-1][train,])
> lda.class = lda.pred$class
> mean(lda.class!=y[train])
[1] 0.164486
```

(e) K=4 is the best, with the lowest classification error validation dataset.
While the classification error on training data is 0.1813

```
> predQuality
 [1] 0.2996255 0.3146067 0.2659176 0.2471910 0.2734082 0.3033708 0.2771536 0.2846442 0.2659176
[10] 0.2921348
>
> KNNpred = knn(x[train,], x[train,], y[train], k = 4)
> mean(KNNpred != y[train])
[1] 0.1813084
```

(f) I will choose Logistic Regression because of the lowest classification error on validation set.
The classification error on validation set are listed as below:

KNN: 0.2584

LDA: 0.1798

Logistic Regression (Lasso): 0.1573

```
> KNNpred = knn(x[train,], x[val,], y[train], k = 4)
> mean(KNNpred != y[val])
[1] 0.258427
>
> lda.pred=predict(lda.fit, ojdata[-1][val,])
> lda.class = lda.pred$class
> mean(lda.class!=y[val])
[1] 0.1797753
>
> pred = predict(lasso.mod, x[val,],type="class");
> mean(pred!=y[val])
[1] 0.1573034
```

(g) The final classification error on test data is 0.1567

```
> lasso.mod=glmnet(x[c(train,val),], y[c(train,val)], alpha=1, lambda=bestlam, family="binomial")
> pred = predict(lasso.mod, x[test,],type="class");
> mean(pred!=y[test])
[1] 0.1567164
```

(h) People who are predicted to buy CH should receive coupon. The best threshold is 0.01 according to the graph (because the profit from true positive beats the loss from false positive), the best payoff in the test data is 577.

```
#calculate costs
payoff =  c(c(3.5,0),c(-0.5,0))
sum(classficationTable * payoff)

# find the best threshold
predprob = predict(lasso.mod, x[test,],type="response")
payoffPerThreshold = vector("numeric",100)

for (i in 1:100)
{
  p = 0.01*i
  lgDecision = ifelse(predprob > p,'CH','MM')
  lgDecision
  classficationTable = table(truth=y[test], predict=lgDecision)
  payoffPerThreshold[i] = ifelse(i<=97,sum(classficationTable * payoff), 0)
}

plot(payoffPerThreshold,pch = 15, xlab = "Threshold")
```

```
> payoffPerThreshold[1]
[1] 577
```