Assignment 1

Q1.

Please View annotated R code in Q1.R file (Tested on Boston dataset during development, however results not saved as I later saw it was used throughout the whole assignment

Q2.

Creating 'high' attribute:

```
Auto['high'] <- ifelse(Auto['mpg'] > 22 & Auto['mpg'] >= 23, 1, 0)[, 1]
Y <- Auto['high']
```

Each quantitative attribute was scaled:

```
horsepower <- scale(Auto['horsepower'])[, 1]
weight <- scale(Auto['weight'])[, 1]
year <- scale(Auto['year'])[, 1]
X <- data.frame(horsepower, weight, year)
```

Qualitative attributes were broken down into k-1 dummy variables using length(unique(attribute)) as K. (see for loop under # question 2)

```
for (i in unlist(unique(Auto['origin']))[-1]) {
 print(paste0('origin', i))
 X[paste0('origin', i)] <- ifelse(Auto['origin'] == i, 1, 0)[, 1]
}
```

Training & results:

```
logistic_regression(Y, X, 200, 0.05, 20)
```

```
[1] "MSE TEST: "
[1] 0.06062829
[1] "MSE TRAIN: "
[1] 0.0676882
```

Q3.

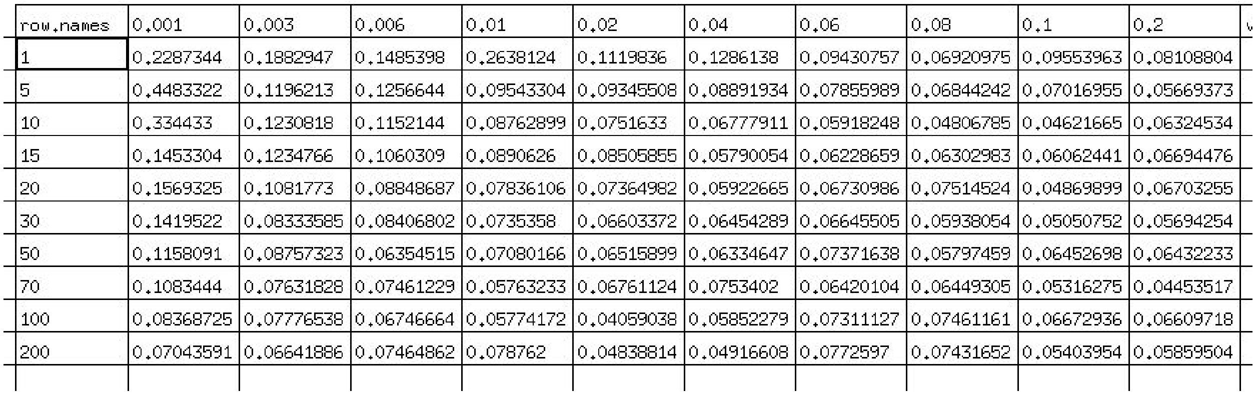
Dataset already declared, (equal split performed)
Training and test sets also declared and appropriately scaled
Global seed declared at top of file and used throughout questions.

Q4.

```
# The random numbers are already declared between -0.7 -> 0.7 for each weight (see Q1 file)
B <- runif(length(X.test), -0.7, 0.7)
```

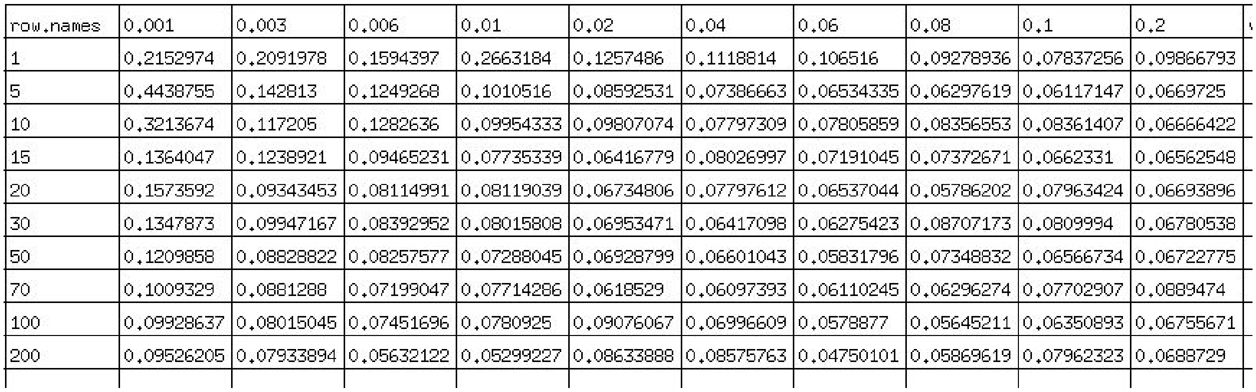
See Q4 section for code, table for test and training mse found bellow.
Column names are learning rates
Row names are number of epochs

Table of training mse:



row.names	0.001	0.003	0.006	0.01	0.02	0.04	0.06	0.08	0.1	0.2
1	0.2287344	0.1882947	0.1485398	0.2638124	0.1119836	0.1286138	0.09430757	0.06920975	0.09553963	0.08108804
5	0.4483322	0.1196213	0.1256644	0.09543304	0.09345508	0.08891934	0.07855989	0.06844242	0.07016955	0.05669373
10	0.334433	0.1230818	0.1152144	0.08762899	0.0751633	0.06777911	0.05918248	0.04806785	0.04621665	0.06324534
15	0.1453304	0.1234766	0.1060309	0.0890626	0.08505855	0.05790054	0.06228659	0.06302983	0.06062441	0.06694476
20	0.1569325	0.1081773	0.08848687	0.07836106	0.07364982	0.05922665	0.06730986	0.07514524	0.04869899	0.06703255
30	0.1419522	0.08333585	0.08406802	0.0735358	0.06603372	0.06454289	0.06645505	0.05938054	0.05050752	0.05694254
50	0.1158091	0.08757323	0.06354515	0.07080166	0.06515899	0.06334647	0.07371638	0.05797459	0.06452698	0.06432233
70	0.1083444	0.07631828	0.07461229	0.05763233	0.06761124	0.0753402	0.06420104	0.06449305	0.05316275	0.04453517
100	0.08368725	0.07776538	0.06746664	0.05774172	0.04059038	0.05852279	0.07311127	0.07461161	0.06672936	0.06609718
200	0.07043591	0.06641886	0.07464862	0.078762	0.04838814	0.04916608	0.0772597	0.07431652	0.05403954	0.05859504

Table of test mse:



row.names	0.001	0.003	0.006	0.01	0.02	0.04	0.06	0.08	0.1	0.2
1	0.2152974	0.2091978	0.1594397	0.2663184	0.1257486	0.1118814	0.106516	0.09278936	0.07837256	0.09866793
5	0.4438755	0.142813	0.1249268	0.1010516	0.08592531	0.07386663	0.06534335	0.06297619	0.06117147	0.0669725
10	0.3213674	0.117205	0.1282636	0.09954333	0.09807074	0.07797309	0.07805859	0.08356553	0.08361407	0.06666422
15	0.1364047	0.1238921	0.09465231	0.07735339	0.06416779	0.08026997	0.07191045	0.07372671	0.0662331	0.06562548
20	0.1573592	0.09343453	0.08114991	0.08119039	0.06734806	0.07797612	0.06537044	0.05786202	0.07963424	0.06693896
30	0.1347873	0.09947167	0.08392952	0.08015808	0.06953471	0.06417098	0.06275423	0.08707173	0.0809994	0.06780538
50	0.1209858	0.08828822	0.08257577	0.07288045	0.06928799	0.06601043	0.05831796	0.07348832	0.06566734	0.06722775
70	0.1009329	0.0881288	0.07199047	0.07714286	0.0618529	0.06097393	0.06110245	0.06296274	0.07702907	0.0889474
100	0.09928637	0.08015045	0.07451696	0.0780925	0.09076067	0.06996609	0.0578877	0.05645211	0.06350893	0.06755671
200	0.09526205	0.07933894	0.05632122	0.05299227	0.08633888	0.08575763	0.04750101	0.05869619	0.07962323	0.0688729

Observations:

We can observe that both the training and test mse tend to decrease as the number of epochs increases and then sometimes increase if the function has been overfit.
Whilst in the training mse the error tends to always decrease with an increase in epochs.
This is what is expected.
There are some anomalies in the data, however this can be attributed to error.

Q5:

Stopping when training mse change is less than 1% over past 10 iterations:

Training loop modified as follows:

```
while (steps_since_last_change < 10) {
 y_p <- sigmoid(X.train, B, b0)
 d <- calc_derivatives(y_p, X.train, Y.train)
 B <- B - lr * d[-1]
 b0 <- b0 - lr * d[1]
 r_train <- mean((sigmoid(X.train, B, b0) - Y.train)^2)
 r_train_top <- r_train * 1.01
 r_train_bottom <- r_train * 0.99
 if (training_mse_last < r_train_bottom || training_mse_last > r_train_top) {
   training_mse_last <- r_train
   #reset counter
   steps_since_last_change <- 0
 }
 steps_since_last_change <- steps_since_last_change + 1
}
```

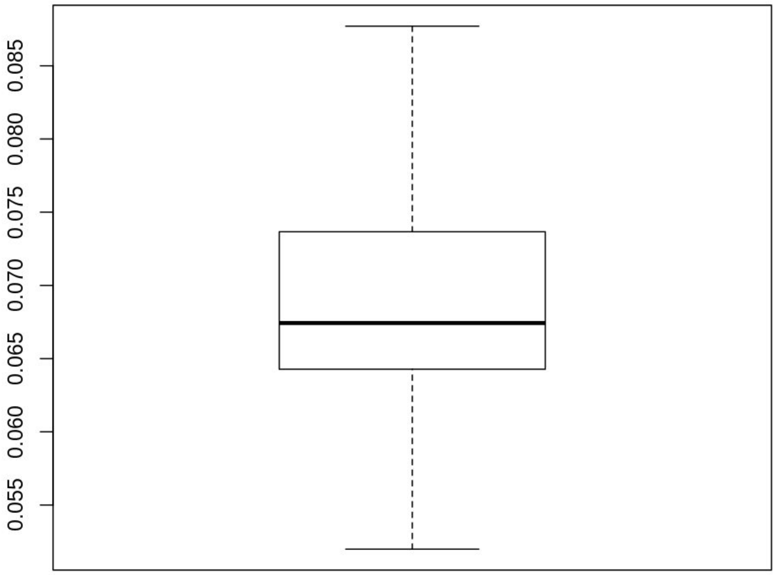
The counter is reset if more than a one percent change has been observed.

If the counter passes 10 points then the training loop breaks and it is hence complete.

Q6:

I chose values n=0.04 and epochs=30 in the interest of saving some time.

Boxplot of test mse over 100 samples. With above configuration of logistic regression.



Q7.

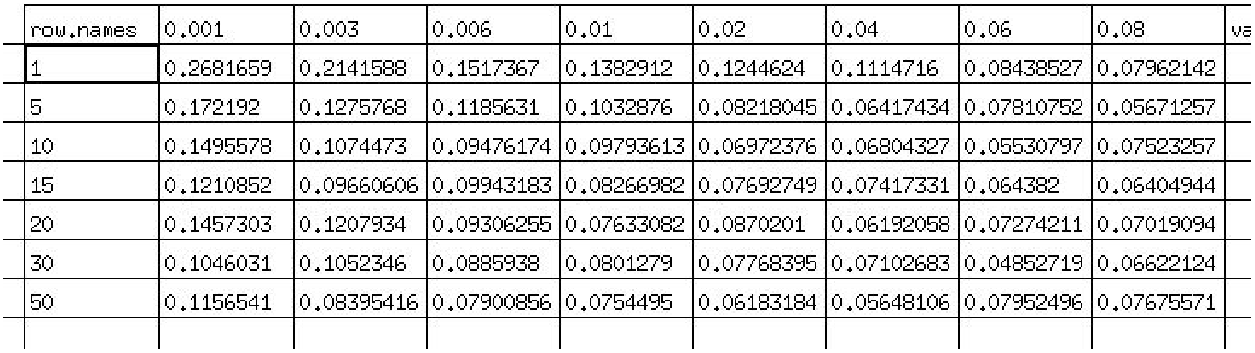
Modified training loop to this (4 runs with different weights, best prediction rule is picked):

```
min_mse <- 100
min_p_rule_b0 <- NULL
min_p_rule_B <- NULL
for (i in 1:4) {
 #  param vector (weights to be updated) -0.7 -> 0.7 (randomly)
 B <- runif(length(X.test), -0.7, 0.7)
 b0 <- 0
 for (e in 1:epochs) {
   y_p <- sigmoid(X.train, B, b0)
   d <- calc_derivatives(y_p, X.train, Y.train)
   B <- B - lr * d[-1]
   b0 <- b0 - lr * d[1]
 }
 r_train <- mean((sigmoid(X.train, B, b0) - Y.train)^2)
 if (r_train < min_mse) {
   min_mse <- r_train
   min_p_rule_b0 <- b0
   min_p_rule_B <- B
 }
}
```

Choosing Best of 4 prediction rules

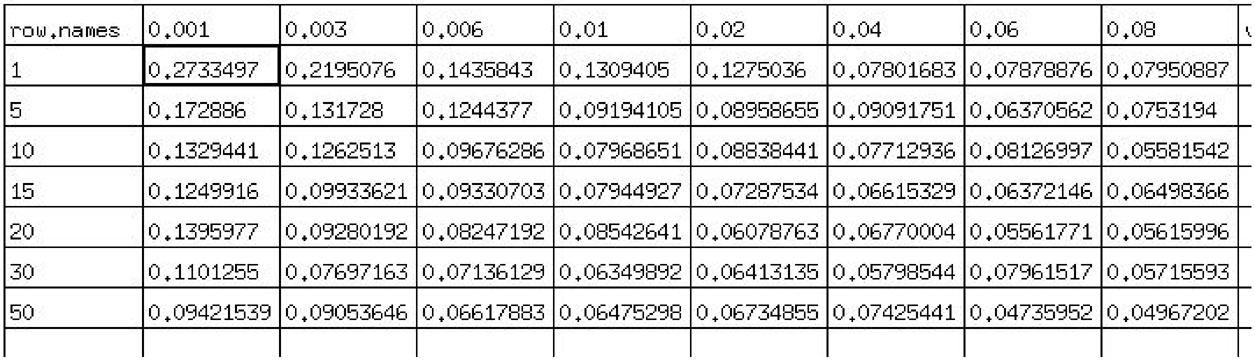
Again, column names are learning rates & row names are number of epochs. I did fewer epochs because it was taking to long.

Test mse table:



row.names	0.001	0.003	0.006	0.01	0.02	0.04	0.06	0.08	va
1	0.2681659	0.2141588	0.1517367	0.1382912	0.1244624	0.1114716	0.08438527	0.07962142	
5	0.172192	0.1275768	0.1185631	0.1032876	0.08218045	0.06417434	0.07810752	0.05671257	
10	0.1495578	0.1074473	0.09476174	0.09793613	0.06972376	0.06804327	0.05530797	0.07523257	
15	0.1210852	0.09660606	0.09943183	0.08266982	0.07692749	0.07417331	0.064382	0.06404944	
20	0.1457303	0.1207934	0.09306255	0.07633082	0.0870201	0.06192058	0.07274211	0.07019094	
30	0.1046031	0.1052346	0.0885938	0.0801279	0.07768395	0.07102683	0.04852719	0.06622124	
50	0.1156541	0.08395416	0.07900856	0.0754495	0.06183184	0.05648106	0.07952496	0.07675571	

Train mse table:



row.names	0.001	0.003	0.006	0.01	0.02	0.04	0.06	0.08	v
1	0.2733497	0.2195076	0.1435843	0.1309405	0.1275036	0.07801683	0.07878876	0.07950887	
5	0.172886	0.131728	0.1244377	0.09194105	0.08958655	0.09091751	0.06370562	0.0753194	
10	0.1329441	0.1262513	0.09676286	0.07968651	0.08838441	0.07712936	0.08126997	0.05581542	
15	0.1249916	0.09933621	0.09330703	0.07944927	0.07287534	0.06615329	0.06372146	0.06498366	
20	0.1395977	0.09280192	0.08247192	0.08542641	0.06078763	0.06770004	0.05561771	0.05615996	
30	0.1101255	0.07697163	0.07136129	0.06349892	0.06413135	0.05798544	0.07961517	0.05715593	
50	0.09421539	0.09053646	0.06617883	0.06475298	0.06734855	0.07425441	0.04735952	0.04967202	