Project One

Jacob Hansén (960512-1533),
Kevin Olsson (960406-0336), and
Liam Persson (970502-0858)

October 2019

Contents

1 Part A **3**
- 1.1 Method and resources . . . 3
- 1.2 Assignment a) . . . 3
- 1.3 Assignment b) . . . 3
- 1.4 Assignment c) . . . 6
- 1.5 Assignment d) . . . 7

2 Part B **8**
- 2.1 Method and resources . . . 8
- 2.2 Assignment a) . . . 9
- 2.3 Assignment b) . . . 11
- 2.4 Assignment c) . . . 12
- 2.5 Part B conclusions) . . . 15

3 Part C **15**
- 3.1 Assignment a) . . . 15
- 3.2 Assignment b) . . . 16

1 Part A

1.1 Method and resources

This part of the project was carried out using the program R, version 3.5.2. The packages used were the following:

Package	Purpose
ISLR	Obtain data.
ggplot2	Plotting relevant plots.
dplyr	Matrix and vector operations.
gridExtra	Plotting several plots on grid.
caTools	Splitting tools for training and test data sets.

In this assignment, the basic R functions were used to fit the models. For example, for logistic regression: "model ¡- glm(formula, family = gaussian, data,...)". Moreover, no noteworthy functions were used to conduct any special operations to the data, other than using certain commands of used packages to plot the data in a desired way, etc.

1.2 Assignment a)

The package *ISLR* was used to access the data set "Auto". The data set was briefly examined and it was concluded that it contained 392 observations and 9 variables. Following the assignment instructions, the median was computed for the variable "mpg" which gave the value 22.75. Subsequently, each observation was then classified with a new variable "mpg_bin" with the following specifications:

$$mpg\_bin = 1, if mpg \geq 22.75$$
$$mpg\_bin = 0, if mpg < 22.75$$

Moreover, the variable "mpg" was removed as instructed in the assignment. If this variable were included in the predictive models trained on this data with "mpg_bin" as the dependent variable, then the variable "mpg" would be the most deciding variable since "mpg_bin" depends entirely on "mpg".

1.3 Assignment b)

For this assignment, some liberty was taken in the interpretation of how the data should be explored graphically. Firstly, note that the variable "name" does not entail any relevant information since it contains 301 levels with 392 observations. Secondly, simply plotting all "mpg" values for each number of cylinders (e.g.) would plot numerous ones and zeros upon each other for each number of cylinders. To acquire some kind of knowledge about the relationship between "mpg" and the predictors, the average value of "mpg" per level of each predictor was predicted.

For example, if 50% of the cars with 4 cylinders were classified as 1 (high mpg) and the remaining 50% as 0 (low mpg), then the average would be 0.5. Thus, the plot below presents the average of "mpg_bin" per predictor.

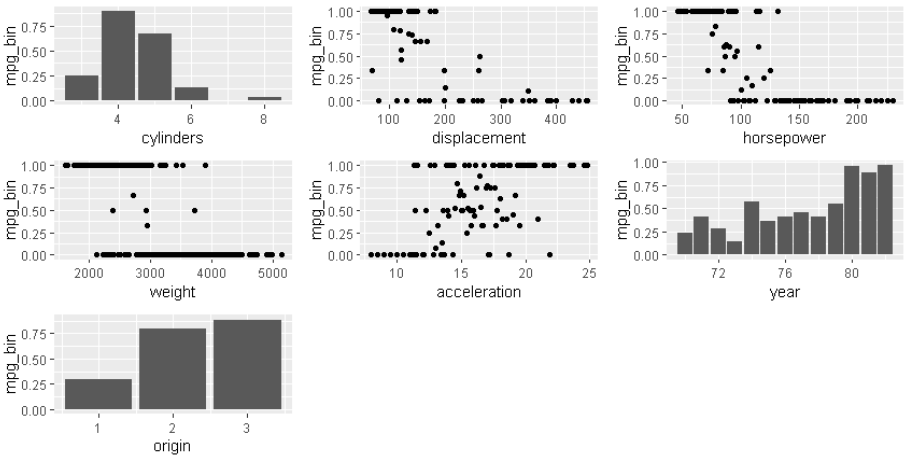


Figure 1: Mean plots

The relationship between "mpg_bin" and "cylinders" poses some questions, since it seems as if cars with 4 or 5 cylinders have high mpg and cars with 3, 6, or 8 cylinders have lower mpg to greater extent. A suspicion that was confirmed was that there was probably a low number of observations in some of these categories.

Cylinders	*3*	*4*	*5*	*6*	*8*
# Observations	4	199	3	83	103

One could conclude that a higher number of cylinders yields a lower mpg, but with such few levels it was decided to remove the variable "cylinders".

Furthermore, note that the high number of levels in relation to data points for "displacement", "horsepower", "weight", and "acceleration", results in many levels which have an average of 1 or 0. The interpretability is further improved by grouping these variables. Each variable was grouped in continuous intervals where each group had the same number of observations in them. "Displacement", "horsepower" and "weight" were grouped into 8 groups, and "acceleration" was grouped into 6 groups. The grouped variables gave the following mean plots:

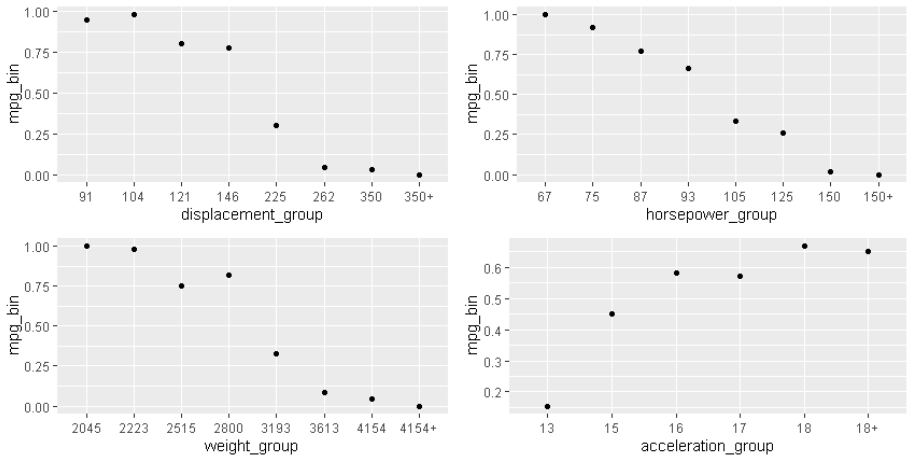


Figure 2: Mean plots with groups

There seems to be linear relationships between "mpg_bin" and all 4 grouped variables.

The assignment suggested to analyze the data with boxplots, but with the specifications of "mpg_bin", the boxplots provide little to no insight into the relationship between the predictors and "mpg_bin". An example is shown below for "cylinders"

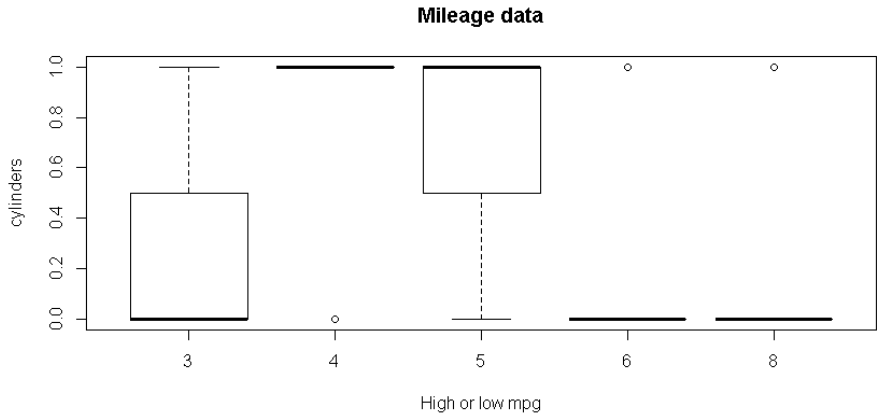


Figure 3: Example of boxplot

Finally, the predictors were plotted against each other to reveal any correlations.

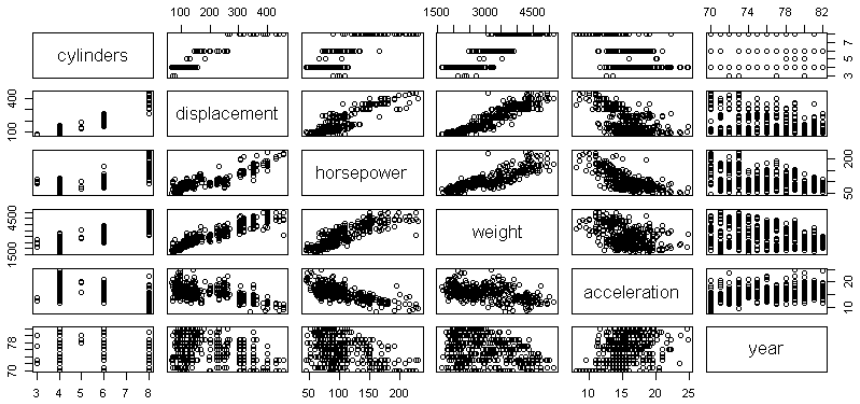


Figure 4: Correlation plot

It seems as if "displacement", "weight", and "horsepower" are all positively linearly correlated to each other. In such a scenario, a reasonable precaution would be to remove up to 2 variables in order to reduce multicollinearity, which affects the performance of logistic regression, LDA, QDA, and kNN. Looking at figure 2, the plot of "displacement" and "weight" are strikingly similar, more so than any of those plots compared to that of "horsepower". In an effort to not remove too much information from the data set, only "displacement" will be removed as a variable amongst these.

Three variables have been removed in this step: "name", "cylinders", and "displacement". Thus, the model fitted in subsequent assignments is:

$$mpg\_bin \sim horsepower, weight, acceleration, year, origin$$

1.4 Assignment c)

The training set was split into a training data set and a test data set. For the training data, 75% of the original data set was sampled and the remaining 25% became the test data set.

Thereafter, the training data was used to fit models with the methods logistic regression, LDA, QDA, and k-nearest-neighbours (kNN). The training errors and the test errors were recorded for each model and they will be presented below. Firstly, the method for fitting the kNN method should be explained further.

The models were fitted using kNN for $k = 1, ..., 100$ in order to find the k which minimizes the test error. This was done and plotting the value of k against the test error gave the following graph:

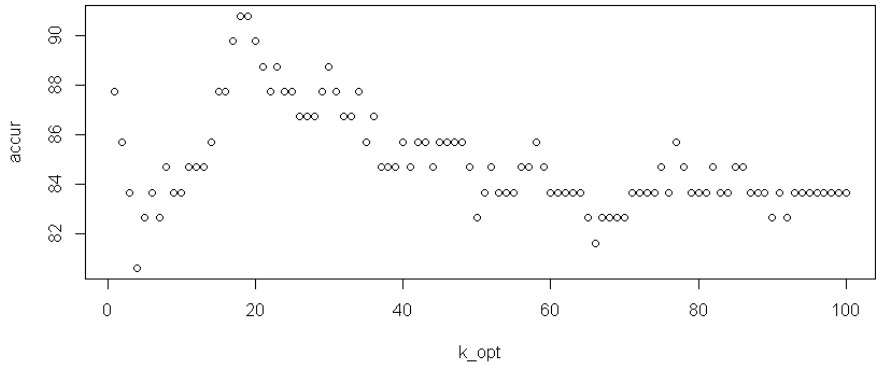


Figure 5: kNN graph

The optimal $k's$ were, as shown in the graph above, $k = 18, k = 19$ which both yielded a test error of: 9.18%. The model with the smaller of the two $k = 18$ was chosen to be compared to the models of the other methods and are shown below.

Model	*Test error*	*Training error*
Logistic regression	8.16%	10.54%
LDA	10.20%	9.52%
QDA	9.18%	11.22%
kNN	9.18%	10.20%

One might note from the table below that logistic regression gave the best test error while the best training error was obtained with LDA. However, the test error rate for LDA was the greatest, and 2.04% greater than that of logistic regression. Therefore, it is difficult to discern a single model as "the best".

1.5 Assignment d)

The last part of this assignment was conducted by comparing the test and training error rates for 4 different sizes of the training and test sets. The proportions were 80%, 60%, 40%, and 20% where if the training data set was 60% of the total data, clearly the test data was 40% of the total data. For each method, for each data set size, the training and test error rates were recorded. This was done with twice and then the averages of these values were computed, in order to obtain more reliable figures.

Package	Purpose
ISLR	Obtain data.
ggplot2	Plotting relevant plots.

(all in %)	Test error				Training error			
Proportion of data	**logistic**	LDA	QDA	kNN	**logistic**	LDA	QDA	kNN
0.8	7.69	10.90	8.97	10.13	10.86	10.39	11.02	12.32
0.6	9.29	10.26	10.90	11.46	10.01	10.11	10.75	11.73
0.4	8.47	12.08	9.96	10.59	10.00	10.96	10.10	11.41
0.2	10.35	11.62	10.35	12.10	10.54	10.06	7.82	10.08

There are some patterns one might note. Firstly, the test error of logistic regression seems to increase as the proportion of test data set size increases. A similar, but reversed, relationship might be observed for the training error of QDA and KNN where the training error decreases with the size of the training set size. Other than that, there are no real patterns that can be observed as the data set varies.

2 Part B

2.1 Method and resources

This part of the project was carried out using the program R, version 3.5.2. The packages used were the following:

In this assignment, the basic R functions were used to fit the models. For example, for logistic regression: "model ¡- glm(formula, family = gaussian, data,...)". Moreover, no noteworthy functions were used to conduct any special operations to the data, other than using certain commands of used packages to plot the data in a desired way, etc.

Throughout Part B, the number of data points, $N = 3000$, was chosen as a large value but the data dividing fraction, $M = 1500$, was chosen deliberately as half the value of N for the sake of giving both sets the same amount of data points, making the key insights of the different combinations of mean and variance easier to understand with visualisations. Let the first M points belong to class 1 (i.e. those with y=1) and the remaining $N - M$ points belong to class 2 (i.e. those with y=-1).

It was interpreted that when changing the distribution parameters in the three assignments, this was for purpose of exploring how this affected the distributions of the two classes of points relative to each other and thus the performance of the models logistic regression, LDA, and QDA. Moreover, the test error was chosen as a tool to assess the performance of the models.

2.2 Assignment a)

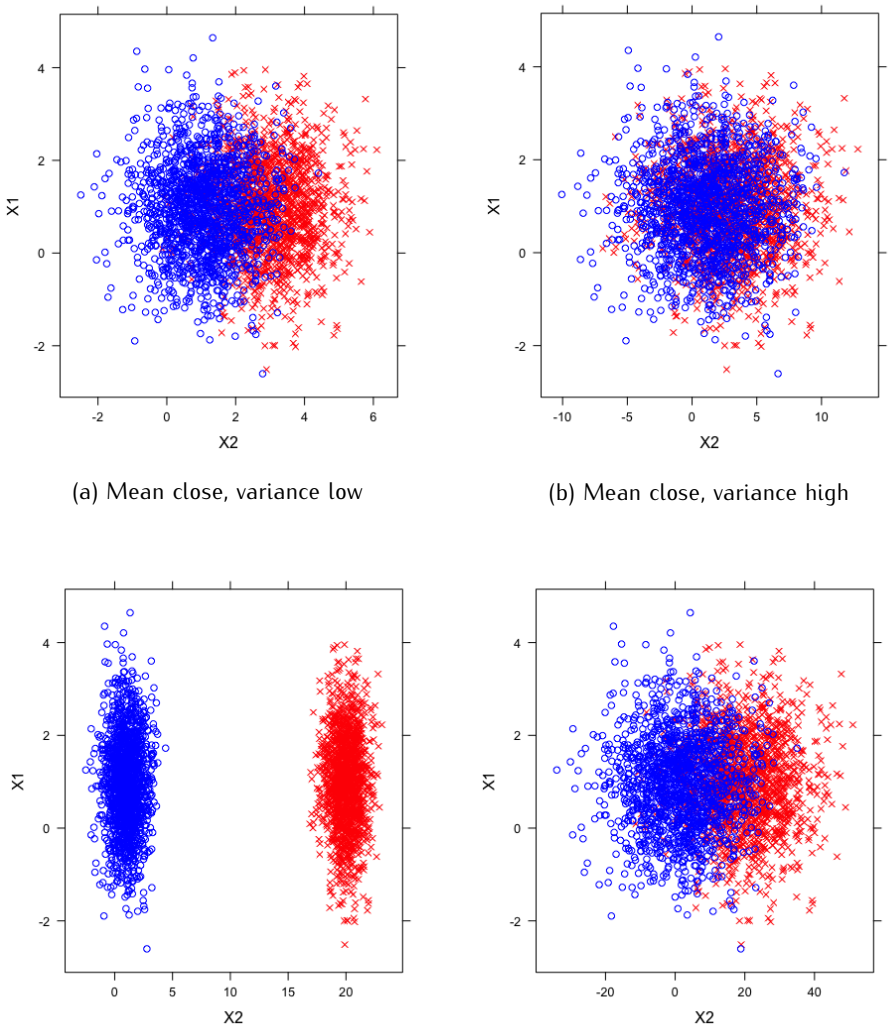
In this assignment, only one component of the gaussian bivariates was varied in order to explore how this affected the accuracy of the different models. There were 4 different cases (a, b, c, d, not to be confused with assignments a, b, c or parts A, B, C) assessed, and the bivariate distributions are presented below:

(a) Class 1: $N((1,1), \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$ Class 2: $N((1,3), \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$

(b) Class 1: $N((1,1), \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$ Class 2: $N((1,3), \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix})$

(c) Class 1: $N((1,1), \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$ Class 2: $N((1,20), \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$

(d) Class 1: $N((1,1), \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$ Class 2: $N((1,20), \begin{pmatrix} 1 & 0 \\ 0 & 100 \end{pmatrix})$



(a) Mean close, variance low

(b) Mean close, variance high

(c) Mean not close, variance low

(d) Mean not close, variance high

Figure 6: Visualisations of distinct cases of parameter values

The hypothesis here was of course, that for case b, the test error would be the highest, since the distributions make the data points overlap to a high degree, and that the test error would be the lowest for case c, where the data points are most separated.

The test errors obtained are presented below:

Case	*log. reg.*	*LDA*	*QDA*
a	16.4%	16.6%	15.5%
b	39.3%	39.3%	39.5%
c	0%	0%	0%
d	17.9%	17.9%	17.9%

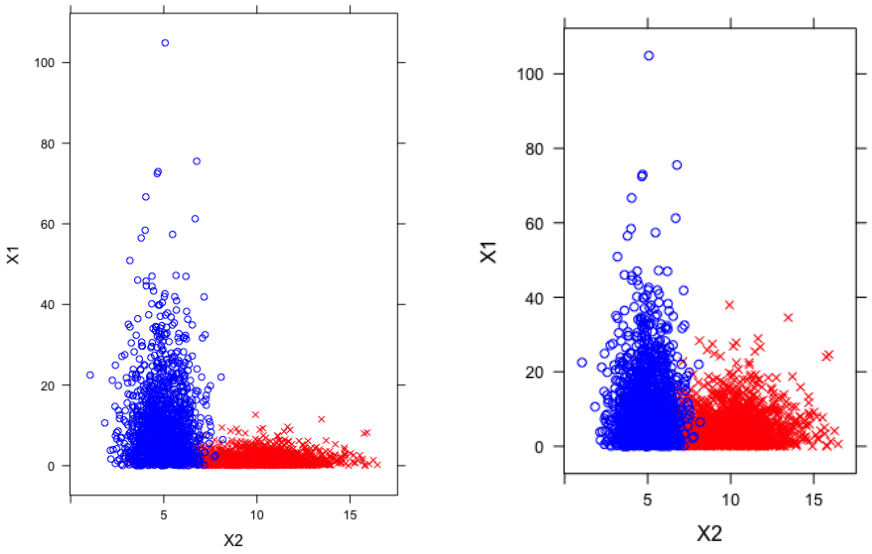
The error rates for the different methods are similar for all 4 cases. The hypothesis proved, unsurprisingly, to be correct. Indeed, case c allows the methods to classify the classes with 0% error in the test data set. For case b, where the two classes nearly completely overlap, the test error rate nears 50%, which in this case would a useless model since classifying between two classes with an equal number of observations is a 50-50 guess.

2.3 Assignment b)

For this assignment, the first component was sampled from an exponential distribution. Moreover, only the exponentially distributed component was varied, since the Gaussian component had already been explored in assignment a). Because the exponential parameter only has one parameter, there were only two cases: one where the parameters were close in value and one where the parameters were not close in value. Thus the two cases a and b were:

(a) Class 1: $x_1 \sim exp(0.1), x_2 \sim N(5,1)$, Class 2: $x_1 \sim exp(1), x_2 \sim N(10,2)$

(b) Class 1: $x_1 \sim exp(0.1), x_2 \sim N(5,1)$, Class 2: $x_1 \sim exp(0.2), x_2 \sim N(10,2)$



(a) Exponential parameter not close (b) Exponential parameter close

Figure 7: Visualisation of distinct cases of parameter values

The hypothesis here was that the test error of case a would be lower than that of case b, since the exponential parameters were close for the two classes. The test errors are presented below:

Case	*log. reg.*	*LDA*	*QDA*
a	5%	5%	5%
b	4.9%	5%	5%

The hypothesis was rejected by the results. Most likely, the models differentiate the two classes mainly by the x_2 axis, meaning that changing the distribution along the x_1 axis does not alter the classification algorithms much.

2.4 Assignment c)

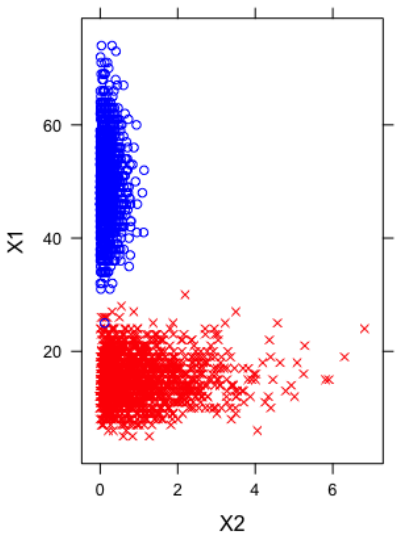
For this assignment, x_1 was sampled from a Poisson distribution and x_2 from an exponential distribution. There were 4 cases investigated:

(a) Class 1: $x_1 \sim Po(50), x_2 \sim exp(10)$, Class 2: $x_1 \sim Po(15), x_2 \sim exp(1)$

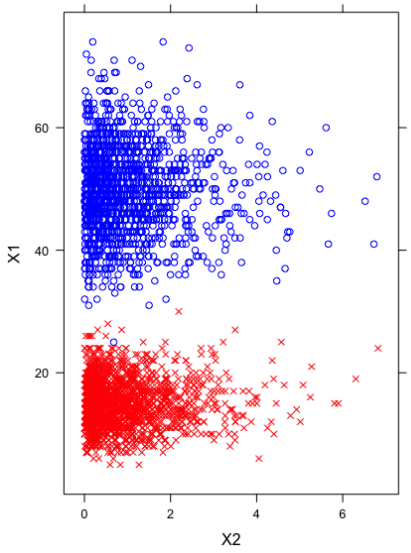
(b) Class 1: $x_1 \sim Po(50), x_2 \sim exp(1)$, Class 2: $x_1 \sim Po(15), x_2 \sim exp(1)$

(c) Class 1: $x_1 \sim Po(30), x_2 \sim exp(1)$, Class 2: $x_1 \sim Po(15), x_2 \sim exp(0.5)$

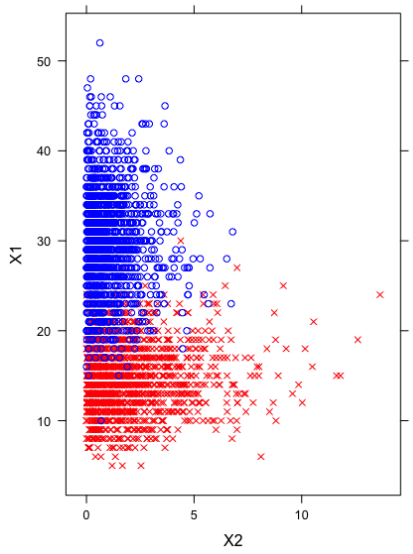
(d) Class 1: $x_1 \sim Po(20), x_2 \sim exp(1)$, Class 2: $x_1 \sim Po(15), x_2 \sim exp(1)$



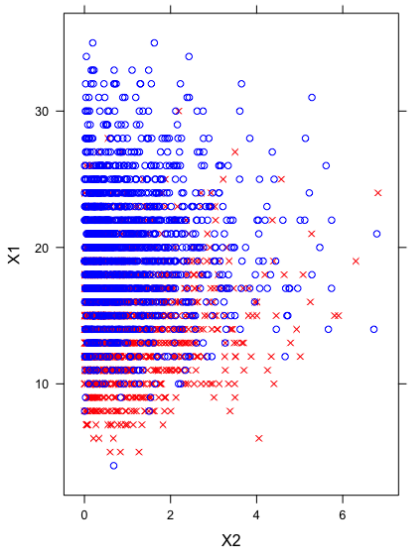
(a) Exponential parameter not close, Poisson parameter far



(b) Exponential parameter same, Poisson parameter far



(c) Exponential parameter close, Poisson parameter close



(d) Exponential parameter same, Poisson parameter closer

Figure 8: Visualisation of distinct cases of different parameter values

The hypothesis here was that cases a and b would have similar test errors, case c would have slightly higher test errors, and case d the highest. The test errors are presented below:

Case	*log. reg.*	*LDA*	*QDA*
a	0%	0%	0%
b	0%	0%	0%
c	1.8%	1.8%	1.8%
d	2.86%	2.86%	2.86%

The hypothesis proved to be correct, as suspected. The errors for the different models are, however, equal for each case.

2.5 Part B conclusions)

Regarding the test errors for the different methods used (logistic regression, LDA, and QDA), then one could hypothesize that the test error for logistic regression would be lower in the cases in assignment b) and c) compared to assignment a). This is because LDA and QDA assumes that the observations come from a normal distribution (while logistic regression does not make any such assumptions), which in assignment a) both components of the observation data points do. In assignment b) and c), one and two components respectively of the data points are sampled from non-Gaussian distributions. However, such an expected pattern is not observed in the results provided above. Logistic regression performs ever so slightly better than LDA and QDA in case b in assignment b), but at the same time worse than QDA in case a in assignment a). This could be an anomaly where these specific configurations lead to this result.

3 Part C

3.1 Assignment a)

To calculate the fraction of the available observations for $p = 1$ we know that 10% of the observations will be used if $0.05 \leq x \leq 0.95$. When $0 \leq x < 0.05$ the fraction of used points will be $0.05 + x$ with implies that the average value for $0 < x < 0.05$ is that 7.5% of the observation will be used. The same fraction of the observations will be used for $0.95 < x \leq 1$. In total, $0.9 * 0.1 + 0.05 * 0.075 + 0.05 * 0.075 = 0.0975$ of the available observations will be used. When using the same calculations for two dimensions we will get $0.0975^2 = 0.0095$. Indeed, if we use the same calculations in p-dimensions we will get the formula 0.0975^p. This implies that the fraction of observations used for $p = 100$ is $0.0975^{100} \approx 8 * 10^{-102}$.

When we spread the observations over a large number of dimensions we see that the observations will spread out more spatially, resulting in a smaller number of

close neighbors. This phenomena is called "curse of dimensionality"[1, p. 108]. Thus, a high p leads to a small number of close neighbours. In effect, in order to obtain a certain number of neighbours that satisfy the proximity requirements of this assignment, the training set needs to have a certain size, statistically. For example, if we want to use the ten closest neighbours for $p = 100$ we need a training data set with approximately $8 * 10^{103}$ (since the proportion $8 * 10^{-102}$ of this number is 10), which is an unrealistic size of a data set. Therefore KNN is a good method for small p[1, p. 108] when the nearest neighbours are close but a poor method if for higher dimensions.

Moreover, KNN is a non-parametric method since it doesn't make any assumptions about the distribution of $f(X)$. In contrast, LDA and QDA are parametric methods since they make assumptions about the distribution of $f(X)$, namely that it is Gaussian. Of course, this leads to a poor fit if the observations are actually from some other distribution which is not close to Gaussian distribution. A general rule is that "parametric methods will tend to outperform non-parametric approaches when there is a small amount of observations per predictor" which is the case for high p. [1, p. 109]

3.2 Assignment b)

LDA assumes that the observations are from a distribution where all the classes have the same covariance matrix. It projects the observations down to a hyperplane such that the means of the observations are minimized, while taking into account a normalization factor that measures within-class variance. [1, p. 142]. QDA instead assumes that each class has its own covariance matrix [1, p. 149], which makes QDA more flexible.

If the Bayes decision boundary is linear LDA will perform better than QDA on the test set because it better describes the Bayes decision boundary. Therefore LDA will perform better when the Bayes decision boundary is linear. The randomness of the data will make QDA detect patterns in the training set that do not exist in the test set, resulting in an overfit for QDA. However, if we just look at the training set, QDA will perform better since QDA overfits.

Since LDA assumes that all of the classes are from a distribution with the same covariance matrix, the predictions on a non-linear Bayes decision boundary will be poor. QDA instead assumes that each class has its own covariance which is the case when the Bayes decision boundary is non-linear. Therefore QDA perform better for the test set when Bayes decision boundary is non-linear.

The performance on the training set when the Bayes decision boundary is linear will be better for QDA. A more flexible method will always result in better performance on the training set. The bias will be lower for the more flexible model. In this case QDA, which is more flexible, will have lower bias and therefore perform better. The bias arise when a complicated problem is approximated by a simpler one, in this case trying to approximate a non-linear Bayes decision boundary with LDA which is linear.

In general the prediction accuracy will improve when you have a bigger sample size for both QDA and LDA. The more data you have the less impact do noise have on the output. The prediction accuracy will increase more for QDA than LDA which implies that it is better to use QDA for big sample sizes but on the other hand LDA tend to be better to use if the sample size is small. This because QDA is much more flexible than LDA and therefore needs more data and is more effected by noise. Because QDA doesn't need the same covariance matrix for all classes it is much more flexible and sensitive than LDA. A extreme point is much more likely to change the outcome dramatically when using QDA than LDA. Therefore it is more crucial for QDA with many training observations to get a good estimate of the decision boundary.

References

[1] Gareth James, Daniela Witten, Trevor Hastie Robert Tibshirani. *An Introduction to Statistical Learning*. Springer Science, New York 2013