Tutorial 5 Solution

1. (MLR) Consider the horseshoe female crab data given in the csv file `crab.csv`. We would want to form a model for the weight of the female crabs (kg), which depends on its width (cm) and its spine condition (1 = both good, 2 = one worn or broken, 3 = both worn or broken).

   (a) Produce a scatter plot of variable weight against width for different condition of spine.

   (b) Fit a linear regression model for weight which has two explanatories, width and spine.

   (c) Is the fitted model significant?

   (d) Derive R^2 and adjusted R^2 of the fitted model.

   (e) Write down the fitted model.

   (f) Two female crabs of the same width, find the difference of their weight if one has spines are of good condition and another one with broken spines.

   (g) Predict the weight of a female crab that has width of 27 cm and has both spines worn or broken.

   Solutions:

   The first step is to import data set into R and declare that 'spine' is a factor (not quantitative).

```
> data<-read.csv('C:/Data/crab.csv')#, header=T)
> head(data)

  color spine width satell weight
1     3     3  28.3      8   3.05
2     4     3  22.5      0   1.55
3     2     1  26.0      9   2.30
4     4     3  24.8      0   2.10
5     4     3  26.0      4   2.60
6     3     3  23.8      0   2.10

> data$spine = as.factor(data$spine)
> table(data$spine)

  1   2   3
 37  15 121

> attach(data)
```

   (a)
```
> plot(width,weight, type = "n")
> points(width[which(spine==1)],weight[which(spine==1)],pch = 20, col = "black")
> points(width[which(spine==2)],weight[which(spine==2)],pch = 6, col = "red")
> points(width[which(spine==3)],weight[which(spine==3)],pch = 10, col= "blue")
> legend(30, 2, legend = c("Spine = 1", "Spine = 2", "Spine = 3"),
+        col = c("black", "red", "blue"), pch = c(20, 6, 10))
```

   The plot is given in Figure 2.

   (b) Fit a linear regression model for weight which has two explanatories, width and spine.

```
> M = lm(weight ~ width + spine, data = data)
> summary(M)

Call:
lm(formula = weight ~ width + spine, data = data)

Residuals:
```

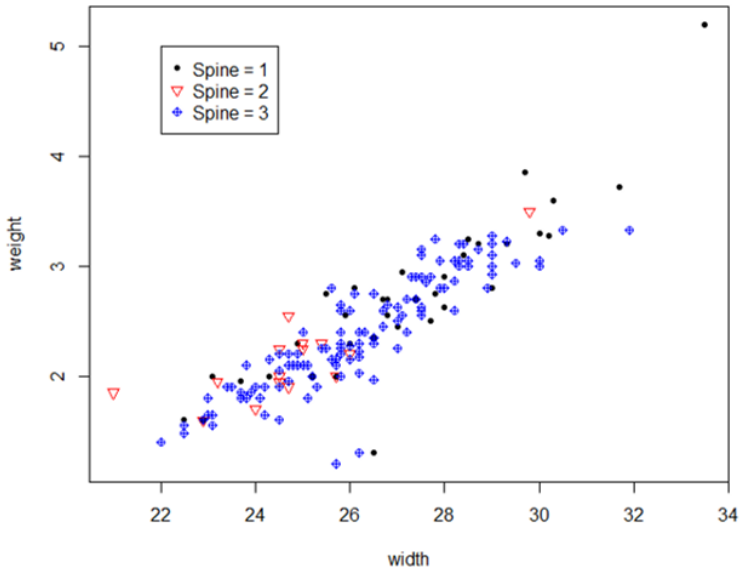

Figure 1: Weight against Width for different condition of Spine

```
     Min       1Q   Median       3Q      Max
-1.23016 -0.10828  0.01016  0.13356  0.96350

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.92955    0.27506 -14.286   <2e-16 ***
width        0.24376    0.01002  24.335   <2e-16 ***
spine2       0.05544    0.08475   0.654    0.514
spine3      -0.06969    0.05065  -1.376    0.171
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2656 on 169 degrees of freedom
Multiple R-squared:  0.7918,        Adjusted R-squared:  0.7881
F-statistic: 214.2 on 3 and 169 DF,  p-value: < 2.2e-16
```

From the output, variable spine is a categorical with 3 categories, hence it is represented by two indicator variables for {spine = 2} and {spine = 3}.

Category {spine = 1} is chosen (by R) as the reference.

(c) Is the fitted model significant?

The fitted model, M, has F-test for the overall significance of the model with extremely small p-value. Hence, model M is significant.

(d) Derive R^2 and adjusted R^2 of the fitted model.

$R^2 = 0.7918$ and $R^2_{adj} = 0.7881$.

(e) Write down the fitted model.

$$\widehat{\text{weight}} = -3.93 + 0.244\text{width} + 0.0554 \times I(\text{spine} = 2) - 0.0697 \times I(\text{spine} = 3).$$

(f) Two female crabs of the same width, one has spines are of good condition (means spine = 1) and another one with broken spines (means spine = 3).

The fitted equation for the crabs that have spines are of good condition is:

$$\widehat{\text{weight}} = -3.93 + 0.244\text{width},$$

while the fitted equation for the crabs that have broken spines is

$$\widehat{\text{weight}} = -3.93 + 0.244\text{width} - 0.0697.$$

When they have the same width, then on average the one that has spines of good condition is heavier than the one with broken spines by 0.0697kg.

(g) Predict the weight of a female crab that has width of 27 cm and has both spines worn or broken.

$$\widehat{\text{weight}} = -3.93 + 0.244 \times 27 - 0.0697 = 2.5883(kg).$$

*Note: this answer might be different from the answer that we derived by R below. That difference is due to rounding we have made when write down the fitted equation.

```
> new = data.frame(width = 27, spine = "3")
> predict(M,new)
       1
2.582352
```

2. The K-nearest neighbor classifier

The table below provides a training data set containing six observations, three predictors, and one qualitative response variable, Y.

Obs	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	-1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

(a) Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$.
Answer:

Obs	X_1	X_2	X_3	Y	Euclidean dist.
1	0	3	0	Red	3.00
2	2	0	0	Red	2.00
3	0	1	3	Red	3.16
4	0	1	2	Green	2.24
5	-1	0	1	Green	1.41
6	1	1	1	Red	1.73

(b) What is our prediction with $K = 1$? Why?

Answer: Green. When $K = 1$, the one nearest point is Green (5th point, with distance of 1.41). Hence, for the test point, we classify it to the category that is the same as the category of the nearest point.

(c) What is our prediction with $K = 3$? Why?

Answer: When $K = 3$, the three nearest points are the 5th (Green), 6th (Red) and 2nd (Red). Hence, the test point will be classified as Red.

(d) If the Bayes decision boundary (the gold standard decision boundary) in this problem is highly non-linear, then would we expect the best value for K to be large or small? Why?

Answer: A small value for K, since it translates to a more flexible classification method.

3. Measures of classifier performance

Suppose we have developed a K-nearest neighbors classifier for predicting diabetes status. The following table shows the actual response Y (1 =yes, 0 =no) and fitted value $\widehat{Y}$ using the classifier for 10 test data points. A test data point is predicted to be $\widehat{G} = 1$ if $\hat{Y} > \delta$, for a specified threshold value δ. (Recall that we use $\delta = 0.5$ in class, also known as the majority rule).

i	Y_i	$\hat{Y}_i$
1	1	0.9
2	1	0.5
3	0	0.7
4	1	0.4
5	1	0.5
6	0	0.2
7	0	0.7
8	1	0.9
9	0	0.1
10	0	0.1

(a) We define

$$TPR = \frac{TP}{TP + FN}; \quad FPR = \frac{FP}{FP + TN}.$$

For each of the thresholds $\delta = 0.3,\ 0.6$ and 0.8, derive TPR and FPR in making predictions with the K-nearest neighbors classifier for the 10 test data points. Plot TPR against FPR for the three thresholds.

		$\widehat{G}$		
		1	0	Total
Y	1	TP	FN	$TP + FN$
	0	FP	TN	$FP + TN$
	Total	$TP + FP$	$FN + TN$	$n = 10$

Answer: We first need to get the predicted outcome for each point for different thresholds (given in the last three columns in Table 1).

When $\sigma = 0.3$, we have:

		$\widehat{G}$		
		1	0	Total
Y	1	$TP = 5$	$FN = 0$	$TP + FN = 5$
	0	$FP = 2$	$TN = 3$	$FP + TN = 5$
	Total	$TP + FP = 7$	$FN + TN = 3$	$n = 10$

Hence, when $\sigma = 0.3$, $FPR = \frac{FP}{FP+TN} = \frac{2}{5} = 0.4$ and $TPR = \frac{TP}{TP+FN} = \frac{5}{5} = 1$.
Similarly, when $\sigma = 0.6$ we have $TP = 2$, $FN = 3$, $FP = 2$ and $TN = 3$. Hence, $FPR = 0.4$ and $TPR = 0.4$.
When $\sigma = 0.8$, $TP = 2$, $FN = 3$, $FP = 0$ and $TN = 5$. Hence, $FPR = 0$ and $TPR = 0.4$.

i	Y_i	$\hat{Y}_i$	Predicted $\hat{G}_i, \sigma = 0.3$	Predicted $\hat{G}_i, \sigma = 0.6$	Predicted $\hat{G}_i, \sigma = 0.8$
1	1	0.9	1	1	1
2	1	0.5	1	0	0
3	0	0.7	1	1	0
4	1	0.4	1	0	0
5	1	0.5	1	0	0
6	0	0.2	0	0	0
7	0	0.7	1	1	0
8	1	0.9	1	1	1
9	0	0.1	0	0	0
10	0	0.1	0	0	0

Table 1: Actual outcomes and the predicted outcomes for different thresholds

A plot where Y-axis is TPR and X-axis is FPR, the three points $(FPR, TPR) = \{(0.4, 1), (0.4, 0.4), (0, 0.4)\}$ is shown in the Figure 2.

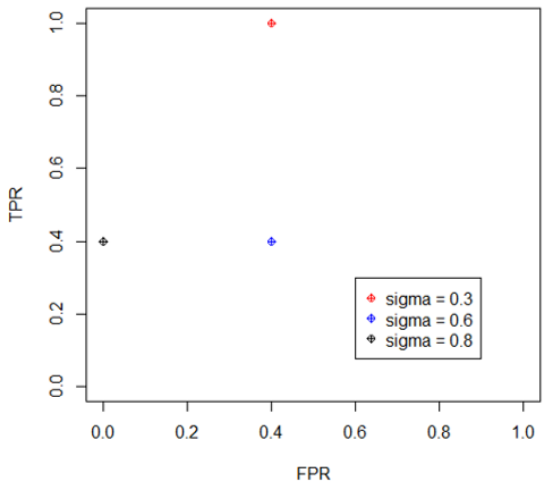

Figure 2: TPR against FPR for three different thresholds

(b) Can we add the two points $(0, 0)$ and $(1, 1)$ to the plot of TPR against FPR in part (a). Explain why or why not.

Answer:

If $\sigma > 0.9$ then all test points have predicted $\hat{G} = 0$ (predicted as negative), so $TPR = FPR = 0$.
If $\sigma < 0.1$, then all test points have predicted $\hat{G} = 1$ (predicted as positive) so $TPR = FPR = 1$.
Since there exist σ within the range from 0 to 1 for the two points to happen, these two points can be added to the plot.

4. The CSV file `Caravan.csv` contains data on 5822 real customer records on caravan insurance purchase. This data set is owned and supplied by the Dutch data mining company, Sentient Machine Research, and is based on real world business data. Each record consists of 86 variables, containing socio-demographic data (variables 1-43) and product ownership (variables 44-86). Variable 86 (**Purchase**) indicates whether the customer purchased a caravan insurance policy.

   For this business, assume that the overall error rate (equivalently, the *accuracy*) is not of interest. Instead, the company wants to use the classifier to predict who are the potential customers likely to purchase insurance. Then the metric *precision* will be important, since it relates the proportion of individuals who will actually purchase the insurance, among the group of individuals who are predicted to purchase insurance.

   (a) Without any classifier, if the company tries to sell insurance to a random selection of customers, what is the success rate?

   Answer:

```
> caravan = read.csv("C:/Data/Caravan.csv")
> dim(caravan) # 87 columns, the first column can be ignored
[1] 5822   87
> table(caravan$Purchase)
  No  Yes
5474  348
> table(caravan$Purchase)[2]/sum(table(caravan$Purchase))
       Yes
0.05977327
> # data set shows 6% of people purchased insurance
```

   (b) Standardize the input features. *Hint*: Use `scale()` command in R.

```
> caravan=caravan[,-1] # remove the first column since it provides no information
> standardized.X= scale(caravan[,-86]) # scaling all the data set, except the last column
```

   (c) Randomly select 2000 observations to form the test data, and the remaining observations will be the training data.

```
> n = dim(caravan)[1] # sample size = 5822
> test = sample(1:n, 2000) # sample a random set of 2000 indexes, from 1:n.
> train.X=standardized.X[-test ,] #training set
> test.X =standardized.X[test ,]  # test set
> train.Y=caravan$Purchase[-test] # response for training set
> test.Y =caravan$Purchase[test] # response for test set
```

   (d) Use 1-nearest neighbor classifier for the training data to predict if a customer will purchase insurance. Compute the precision of the classifier.

```
> set.seed (5)
> library(class)
> knn.pred = knn(train.X,test.X,train.Y,k=1) # KNN with k = 1
> confusion.matrix=table(test.Y,knn.pred)
> confusion.matrix
```

```
      knn.pred
test.Y   No  Yes
   No  1785  109
   Yes   97    9
```

```
> precision = confusion.matrix[2,2]/sum(confusion.matrix[,2])
> precision
```

```
[1] 0.07627119
```

(e) Repeat question 4d, for k-nearest neighbor classifier where $k = 3, 5$. Which value of k gives the best precision?

```
> # K = 3
>
> knn.pred = knn(train.X,test.X,train.Y,k=3) # KNN with k = 3
> confusion.matrix=table(test.Y,knn.pred)
> precision = confusion.matrix[2,2]/sum(confusion.matrix[,2])
> precision
```

```
[1] 0.125
```

```
> # K = 5
> knn.pred = knn(train.X,test.X,train.Y,k=5) # KNN with k = 5
> confusion.matrix=table(test.Y,knn.pred)
> precision = confusion.matrix[2,2]/sum(confusion.matrix[,2])
> precision
```

```
[1] 0.2727273
```

So far, $k = 5$ gives the best precision.

However, one might use N-fold cross validation to have the average precision for each k. With that, the value of k that gives largest average precision is chosen.