Homework 4

4375 Machine Learning with Dr. Mazidi

Anthony Martinez | amm180005

9/19

This script will run Logistic Regression and Naive Bayes on the BreastCancer data set which is part of package mlbench.

Step 1: Data exploration

- Load package mlbench, installing it at the console if necessary
- Load data(BreastCancer)
- Run str() and head() to look at the data
- Run summary() on the Class column
- Use R code to calculate and output the percentage in each class, with a label using paste()

Comment on the types of predictors available in terms of their data types:

```
# your code here
if (!require("mlbench")){
  install.packages("mlbench")
}
```

```
## Loading required package: mlbench
```

```
data("BreastCancer")
as.data.frame(BreastCancer)
```

	Id <chr>	Cl.thickness <ord>	Cell.size <ord>	Cell.shape <ord>	Marg.adhesion <ord>	Epith.c.size <ord>	Bare.nuclei <fct>	I
1	1000025	5	1	1	1	2	1	3
2	1002945	5	4	4	5	7	10	3
3	1015425	3	1	1	1	2	2	3
4	1016277	6	8	8	1	3	4	3
5	1017023	4	1	1	3	2	1	3
6	1017122	8	10	10	8	7	10	9
7	1018099	1	1	1	1	2	10	3
8	1018561	2	1	2	1	2	1	3
9	1033078	2	1	1	1	2	1	1

	Id	Cl.thickness	Cell.size	Cell.shape	Marg.adhesion	Epith.c.size	Bare.nuclei	
	<chr>	<ord>	<ord>	<ord>	<ord>	<ord>	<fct>	
10	1033078	4	2	1	1	2	1	2

1-10 of 699 rows | 1-9 of 12 columns Previous **1** 2 3 4 5 6 … 70 Next

```
str(BreastCancer)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ Id             : chr  "1000025" "1002945" "1015425" "1016277" ...
##  $ Cl.thickness   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 5 5 3 6 4 8 1 2 2 4 ...
##  $ Cell.size      : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 1 1 2 ...
##  $ Cell.shape     : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 2 1 1 ...
##  $ Marg.adhesion  : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 5 1 1 3 8 1 1 1 1 ...
##  $ Epith.c.size   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare.nuclei    : Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10 10 1 1 1 ...
##  $ Bl.cromatin    : Factor w/ 10 levels "1","2","3","4",..: 3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal.nucleoli: Factor w/ 10 levels "1","2","3","4",..: 1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses        : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 5 1 ...
##  $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
```

```
head(BreastCancer)
```

	Id	Cl.thickness	Cell.size	Cell.shape	Marg.adhesion	Epith.c.size	Bare.nuclei	Bl.
	<chr>	<ord>	<ord>	<ord>	<ord>	<ord>	<fct>	<fc
1	1000025	5	1	1	1	2	1	3
2	1002945	5	4	4	5	7	10	3
3	1015425	3	1	1	1	2	2	3
4	1016277	6	8	8	1	3	4	3
5	1017023	4	1	1	3	2	1	3
6	1017122	8	10	10	8	7	10	9

6 rows | 1-9 of 12 columns

```
percent <- summary(BreastCancer$Class)/699*100

print(paste('The percentages of the benign class is: ', percent[1]))
```

```
## [1] "The percentages of the benign class is:  65.5221745350501"
```

```
print(paste('The percentages of the malignant class is: ', percent[2]))
```

```
## [1] "The percentages of the malignant class is:  34.4778254649499"
```

Step 2: First logistic regression model

- Cell.size and Cell.shape are in one of 10 levels
- Build a logistic regression model called glm0, where Class is predicted by Cell.size and Cell.shape
- Do you get any error or warning messages? Google the message and try to decide what happened
- Run summary on glm0 to confirm that it did build a model
- Write about why you think you got this warning message and what you could possibly do about it. List the source of your information in a simple markdown link.

Your commentary here: I get a warning message that says "glm.fit: fitted proabilities numerically 0 or 1 occured. After researching the warning I've learned that his occurs when a you fit a logistice regression model that predicts a probability of observation(s) in the data that are indistinguishable from 0 or 1. Usually, this is due to outliers What I could potentially do is to remove any outlines from the data. Source (https://www.statology.org/glm-fit-fitted-probabilities-numerically-0-or-1-occurred/)

```
# your code here

glm0 <- glm(BreastCancer$Class~BreastCancer$Cell.size+BreastCancer$Cell.shape, data=BreastCance
r, family = binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm0)
```

```
##
## Call:
## glm(formula = BreastCancer$Class ~ BreastCancer$Cell.size + BreastCancer$Cell.shape,
##     family = binomial, data = BreastCancer)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6380  -0.0844  -0.0844   0.0000   3.3583
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  7.77977  757.06727   0.010    0.992
## BreastCancer$Cell.size.L    10.45177  950.68968   0.011    0.991
## BreastCancer$Cell.size.Q     0.04063 1479.65504   0.000    1.000
## BreastCancer$Cell.size.C    10.70546  948.84001   0.011    0.991
## BreastCancer$Cell.size^4    12.06582 1241.92612   0.010    0.992
## BreastCancer$Cell.size^5     0.74199  792.70275   0.001    0.999
## BreastCancer$Cell.size^6    -3.08210 1011.79270  -0.003    0.998
## BreastCancer$Cell.size^7     7.47104 1044.50458   0.007    0.994
## BreastCancer$Cell.size^8     5.60143  830.93455   0.007    0.995
## BreastCancer$Cell.size^9   -10.22144 1812.16582  -0.006    0.995
## BreastCancer$Cell.shape.L   18.15803 2619.03235   0.007    0.994
## BreastCancer$Cell.shape.Q    9.14381 1500.17053   0.006    0.995
## BreastCancer$Cell.shape.C    5.50082 1302.51283   0.004    0.997
## BreastCancer$Cell.shape^4   -2.23752 2679.86462  -0.001    0.999
## BreastCancer$Cell.shape^5   -5.76978 3193.32564  -0.002    0.999
## BreastCancer$Cell.shape^6   -5.58415 2713.54558  -0.002    0.998
## BreastCancer$Cell.shape^7   -3.94569 1740.80748  -0.002    0.998
## BreastCancer$Cell.shape^8   -1.82009  827.39666  -0.002    0.998
## BreastCancer$Cell.shape^9   -0.77209  257.90960  -0.003    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 900.53  on 698  degrees of freedom
## Residual deviance: 198.66  on 680  degrees of freedom
## AIC: 236.66
##
## Number of Fisher Scoring iterations: 19
```

Step 3: Data Wrangling

Notice in the summary() of glm0 that most of the levels of Cell.size and Cell.shape became predictors and that they had very high p-values, that is, they are not good predictors. We would need a lot more data to build a good logistic regression model this way. Many examples per factor level are generally required for model building. A better approach might be to just have 2 levels for each variable.

In this step:

- Add two new columns to BreastCancer as listed below:
    a. Cell.small which is a binary factor that is 1 if Cell.size==1 and 0 otherwise
    b. Cell.regular which is a binary factor that is 1 if Cell.shape==1 and 0 otherwise
- Run summary() on Cell.size and Cell.shape as well as the new columns

- Comment on the distribution of the new columns
- Do you think what we did is a good idea? Why or why not?

Your commentary here: Yes this was a good idea. From the summary method we can see that more than half of the cell sizes are 1, so it makes more sense to categorize the cell sizes as 1 or not 1. The same can be said for the cell shape.

```
# BreastCancer$Cell.small column
BreastCancer$Cell.small <- 0
BreastCancer$Cell.small[BreastCancer$Cell.size == 1] <- 1
BreastCancer$Cell.small <- factor(BreastCancer$Cell.small)
```

```
# BreastCancer$Cell.regular column
BreastCancer$Cell.regular <- 0
BreastCancer$Cell.regular[BreastCancer$Cell.shape == 1] <- 1
BreastCancer$Cell.regular <- factor(BreastCancer$Cell.regular)

summary(BreastCancer$Cell.size)
```

```
##   1   2   3   4   5   6   7   8   9  10
## 384  45  52  40  30  27  19  29   6  67
```

```
summary(BreastCancer$Cell.shape)
```

```
##   1   2   3   4   5   6   7   8   9  10
## 353  59  56  44  34  30  30  28   7  58
```

```
summary(BreastCancer$Cell.small)
```

```
##   0   1
## 315 384
```

```
summary(BreastCancer$Cell.regular)
```

```
##   0   1
## 346 353
```

Step 4: Examine the relationship of malignancy to Cell.size and Cell.shape

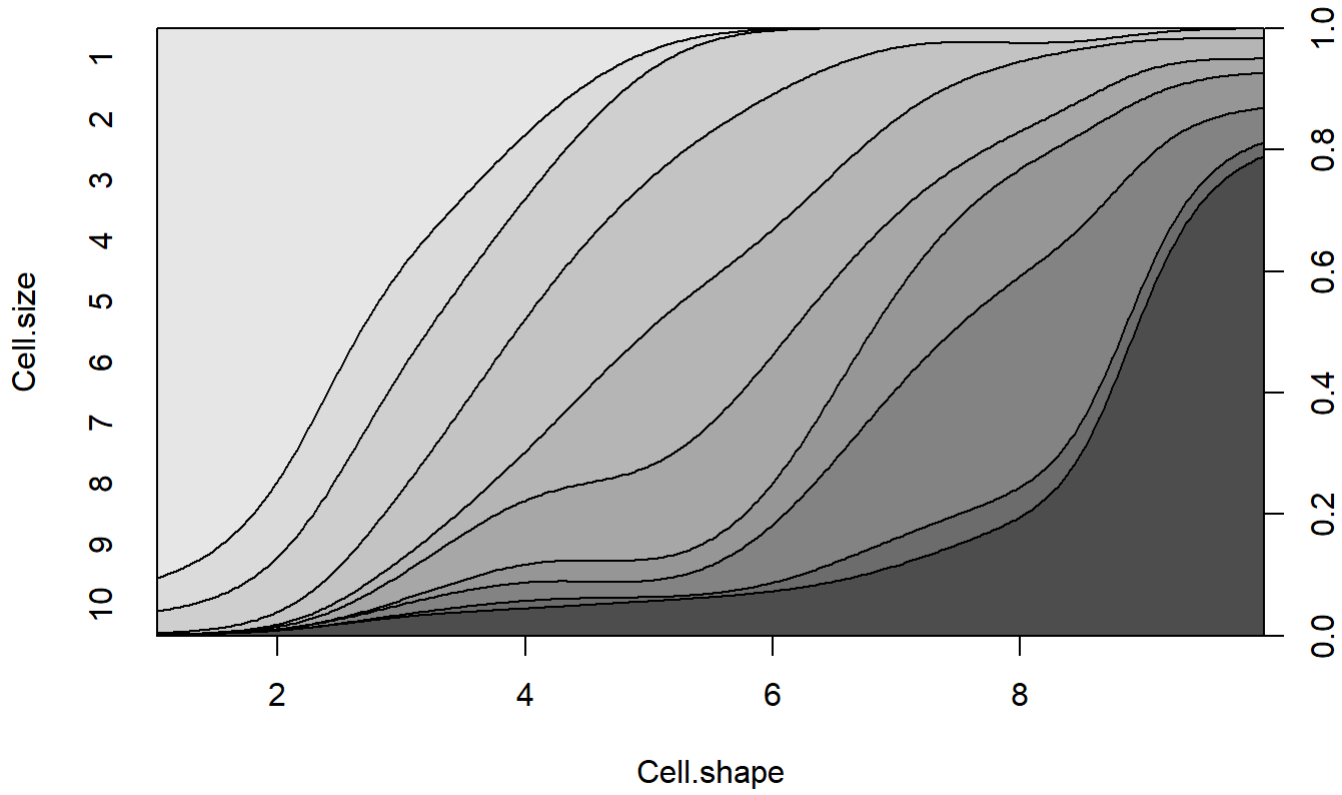
- Create conditional density plots using the original Cell.size and Cell.shape, but first, attach() the data to reduce typing
- Then use par(mfrow=c(1,2)) to set up a 1x2 grid for two cdplot() graphs with Class~Cell.size and Class~Cell.shape
- Observing the plots, write a sentence or two comparing size and malignant, and shape and malignant

- Do you think our cutoff points for size==1 and shape==1 were justified now that you see this graph? Why or why not?

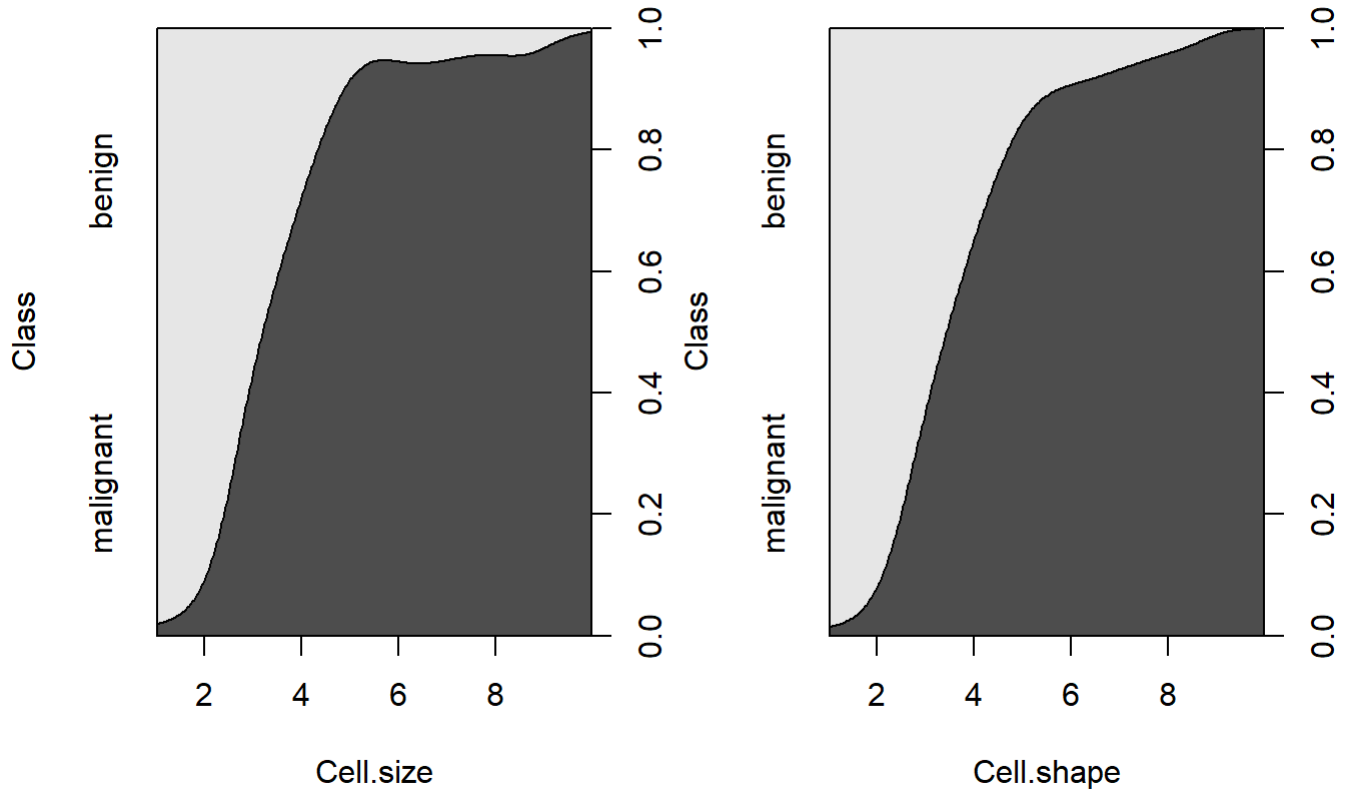
Your commentary here: From the density plots we can see that for cell size==1 there are a lot of malignant cells. As the Cell.size increases there is a sharp increase in the benign cells. The same thing occurs for Cell.shape The cutoff points for size==1 and shape==1 are justified.

```
# your code

attach(BreastCancer)
cdplot(Cell.size~Cell.shape)
```




```
par(mfrow=c(1,2))
cdplot(Class~Cell.size)
cdplot(Class~Cell.shape)
```




```
detach(BreastCancer)
```

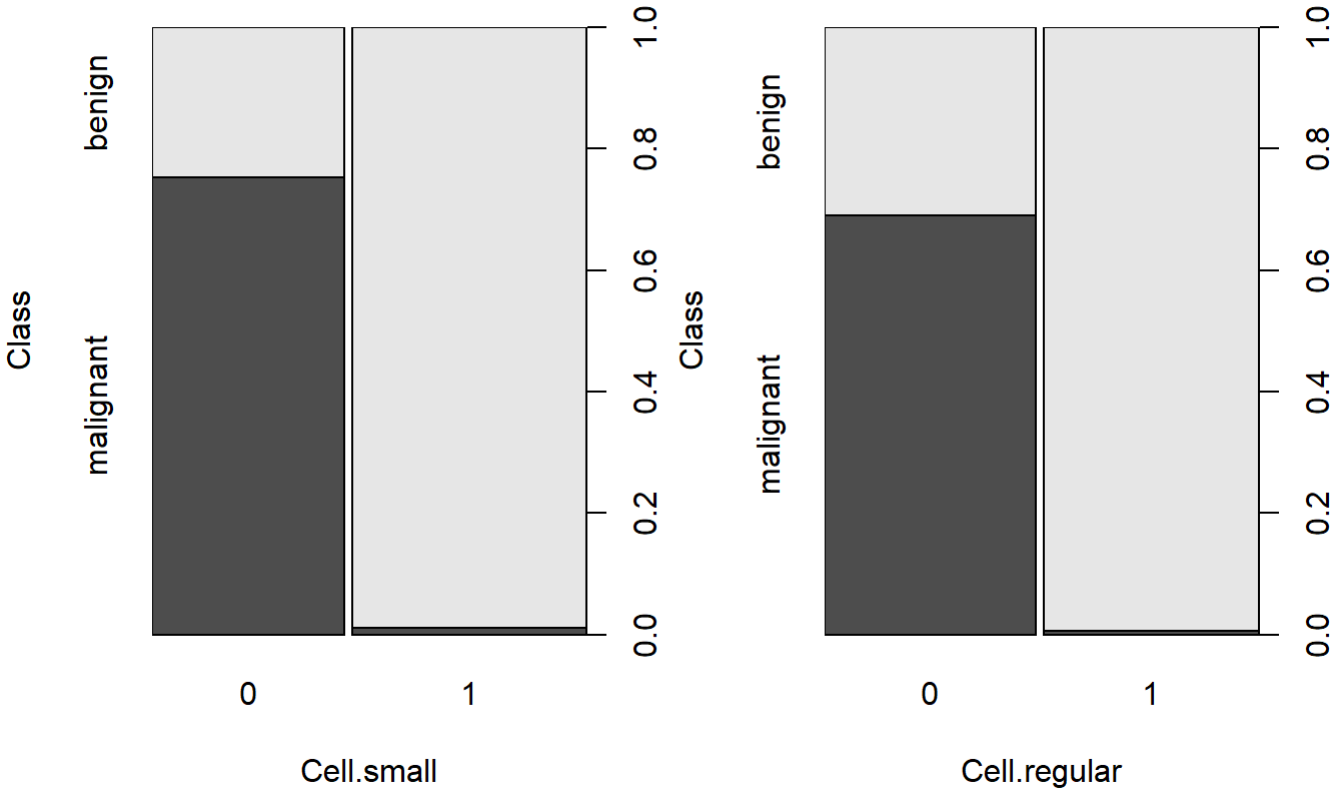
Step 5: Explore the new columns

- Create plots (not cdplots) with the two new columns
- Again, use par(mfrow=c(1,2)) to set up a 1x2 grid for two plot() graphs with Class~Cell.small and Class~Cell.regular
- Now create two cdplot() graphs for the new columns
- Compute and output with labels the following: ((Examples on p. 142 may help)
    a. calculate the percentage of malignant observations that are small
    b. calculate the percentage of malignant observations that are not small
    c. calculate the percentage of malignant observations that are regular
    d. calculate the percentage of malignant observations that are not regular
- Write whether you think small and regular will be good predictors

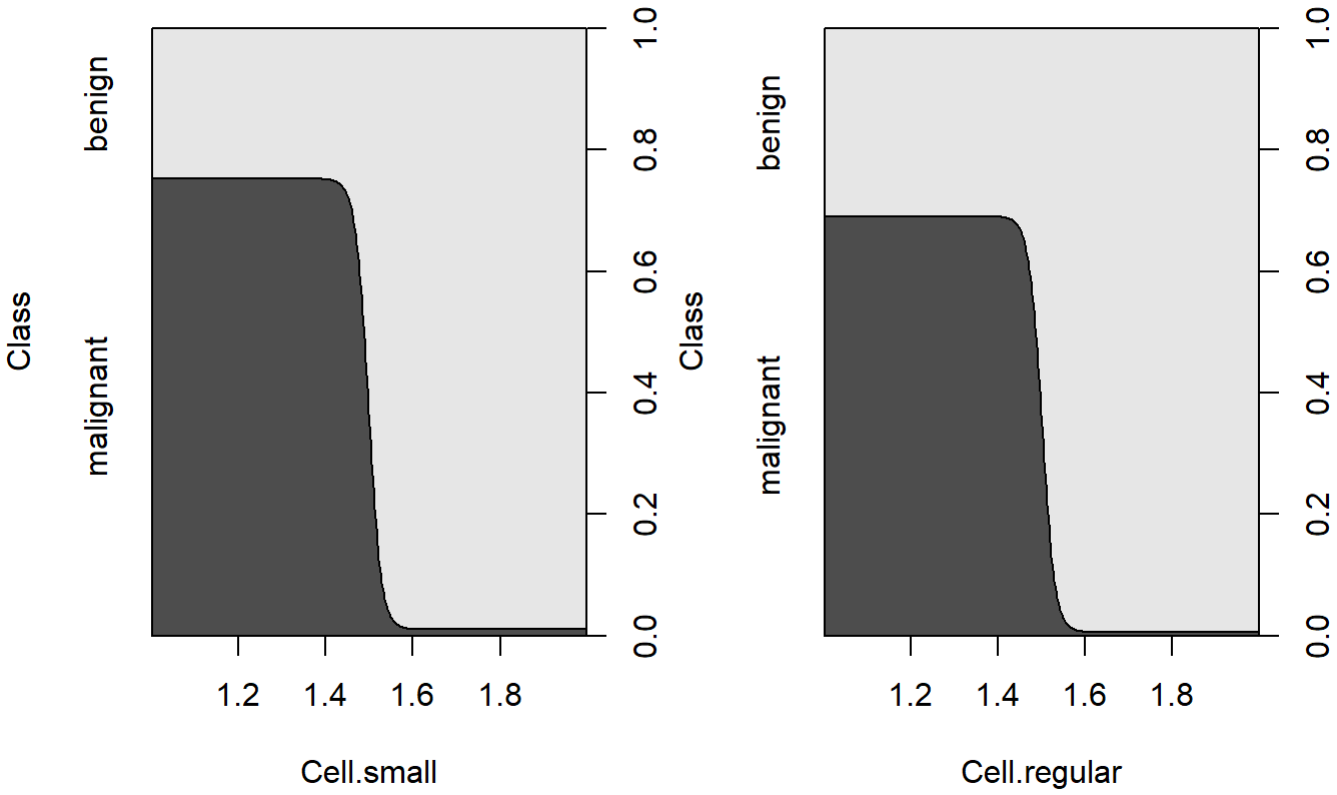
Your commentary here: I would say that small and regular will be good predictors. This is because the majority of malignant are one type of size and shape.

```
# plots here

attach(BreastCancer)
par(mfrow=c(1,2))
plot(Class~Cell.small, xlab = "Cell.small", ylab = "Class")
plot(Class~Cell.regular, xlab = "Cell.regular", ylab = "Class")
```




```
par(mfrow=c(1,2))
cdplot(Class~Cell.small)
cdplot(Class~Cell.regular)
```




```
# calculate the percentage of malignant observations that are small

small <- c(
nrow(BreastCancer[BreastCancer$Cell.small=="0",])/nrow(BreastCancer),
nrow(BreastCancer[BreastCancer$Cell.small=="1",])/nrow(BreastCancer)
)


regular <- c(
nrow(BreastCancer[BreastCancer$Cell.regular=="0",])/nrow(BreastCancer),
nrow(BreastCancer[BreastCancer$Cell.regular=="1",])/nrow(BreastCancer)
)


detach(BreastCancer)
```

```
# calculations and output here
print("Prior probability, small=no, small=yes:")
```

```
## [1] "Prior probability, small=no, small=yes:"
```

```
small
```

```
## [1] 0.4506438 0.5493562
```

```
print("Prior probability, regular=no, regular=yes:")
```

```
## [1] "Prior probability, regular=no, regular=yes:"
```

```
regular
```

```
## [1] 0.4949928 0.5050072
```

Step 6: Train/test split

- Divide the data into 80/20 train/test sets, using seed 1234

```
# your code here
set.seed(1234)
i <- sample(1:nrow(BreastCancer), .80*nrow(BreastCancer), replace=FALSE)
train <- BreastCancer[i,]
test <- BreastCancer[-i,]
```

Step 7: Build a logistic regression model

- Build a logistic regression model predicting malignant with two preditors: Cell.small and Cell. regular
- Run summary() on the model
- Which if any of the predictors are good predictors?
- Comment on the model null variance versus residual variance and what it means
- Comment on the AIC score

Your commentary here: Cell.small and Cell.regular are good predictors. They both have p-values close to 0 The null variance measures the lack of fit of the model only considering the intercepts. The residual deviance measures the lack of fir of the entire model. We want to see the residual deviance much lower than Null deviance. Which is the case here.

AIC stands for Akaike Information Criterion, it is based on the deviance The AIC score is useful for comparing models. The lower AIC is better. Notice that the AIC score is higher for this model than glm0 which means glm1 might be a better model for this data.

```
# your code here
glm1 <- glm(BreastCancer$Class~BreastCancer$Cell.small+BreastCancer$Cell.regular, data=train, fa
mily=binomial)
summary(glm1)
```

```
##
## Call:
## glm(formula = BreastCancer$Class ~ BreastCancer$Cell.small +
##     BreastCancer$Cell.regular, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8041  -0.0517  -0.0517   0.6614   3.6378
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   1.4087     0.1466   9.612  < 2e-16 ***
## BreastCancer$Cell.small1     -4.0405     0.5396  -7.489 6.96e-14 ***
## BreastCancer$Cell.regular1   -3.9835     0.7467  -5.334 9.59e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 900.53  on 698  degrees of freedom
## Residual deviance: 334.40  on 696  degrees of freedom
## AIC: 340.4
##
## Number of Fisher Scoring iterations: 8
```

Step 8: Evaluate on the test data

- Test the model on the test data
- Compute and output accuracy
- Output the confusion matrix and related stats using the confusionMatrix() function in the caret package
- Were the mis-classifications more false positives or false negatives?

Your commentary here:

```
# your code here
probs <- predict(glm1, newdata=test, type="response")
```

```
## Warning: 'newdata' had 140 rows but variables found have 699 rows
```

```
pred <- ifelse(probs>0.5, 2, 1)

# calculate and ouput accuracy
acc <- mean(pred==as.integer(test$Class))
```

```
## Warning in pred == as.integer(test$Class): longer object length is not a
## multiple of shorter object length
```

```
print(paste("glm1 accuracy = ", acc))
```

```
## [1] "glm1 accuracy =  0.520743919885551"
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
#confusionMatrix(as.factor(pred),reference=test$Class)
```

Step 9: Model coefficients

- The coefficients from the model are in units of logits. Extract and output the coefficient of Cell.small with glm1$coefficients[]
- Find the estimated probability of malignancy if Cell.small is true using exp(). See the example on p. 107 of the pdf.
- Find the probability of malignancy if Cell.small is true over the whole BreastCancer data set and compare results. Are they close? Why or why not?

Your commentary here:

```
# your code here
```

Step 10: More logistic regression models

- Build two more models, glm_small using only Cell.small, and glm_regular using Cell.regular as the predictor
- Use anova(glm_small, glm_regular, glm1) to compare all 3 models, using whatever names you used for your models. Analyze the results of the anova().
- Also, compare the 3 AIC scores of the models. Feel free to use the internet to help you interpret AIC scores.

Your commentary here:

```
# your code here
```

Step 11: A Naive Bayes model

- Build a Naive Bayes Model Class ~ Cell.small + Cell.regular on the training data using library e1071
- Output the model parameters
- Aand nswer the following questions:
    a. What percentage of the training data is benign?
    b. What is the likelihood that a malignant sample is not small?
    c. What is the likelihood that a malignant sample is not regular?

Your commentary here:

```
# your code here
```

Step 12: Evaluate the model

- Predict on the test data with Naive Bayes model
- Output the confusion matrix
- Are the results the same or different? Why do you think that is the case?

Your commentary here:

```
# your code here
```