Breast cancer. Logistic regression analysis.

Maryna Shut

2023-19-01

Overview, goal and methods

In this notebook I'm going to do a logistic regression analysis of a dataset to classify a tumour as malignant or benign. In the analysis I will also be using backward elimination, I will explore the step() function and check the effectiveness of the model by calculating the accuracy, sensitivity, precision and F1 score. The variable I'm going to be predicting can have one of the 2 values: 0 or 1, therefore this is a binary classification project.

Terminology

Before we start with the analysis it is important to understand what exactly we are trying to predict and what the information provided, our variables of the dataset, mean. "Benign" refers to a type of medical condition or growth that is not cancerous or dangerous as opposed to "malignant". The dataset contains 9 independent variables, each of them is a feature that is typically used in breast cancer analysis. Let's break them down and understand what they mean.

- Clump thickness is a measure of how thick the cells are within a tumor. Benign cells tend to be grouped in mono-layers, while cancerous - in multi-layer.(Sarkar et al. 2017, p. 1)
- Uniformity of cell size and uniformity of cell shape are two characteristics that can be used to describe the appearance of cells under a microscope. Here we are checking the degree to which the cells in a sample are similar in size and shape.
- Marginal adhesion is the degree to which cells in a tissue sample adhere, or stick, to one another at the edges of the sample. Loss of adhesion might be a sign of malignancy.
- Single epithelial cell size is the size of individual cells in an epithelial tissue sample. Epithelial tissue is a type of tissue that covers the surface of the body and lines internal organs and structures. It is made up of cells that are tightly packed together and held in place by specialized junctions.
- Bare nuclei refers to cells in a tissue sample that are missing their cell membranes and cytoplasm, leaving only the nucleus visible.
- Bland chromatin is the appearance of the genetic material (chromatin) in the nucleus of a cell under a microscope. Chromatin is made up of DNA and proteins, and it contains the genetic information that controls the cell's functions. When the chromatin in a cell's nucleus is compact and uniform in appearance, it is said to be "bland."
- Normal nucleoli are small, spherical structures found within the nucleus of a cell. They are composed of DNA, RNA, and proteins and are responsible for synthesizing ribosomes, which are the cellular structures that produce proteins. Nucleoli are usually visible under a microscope and can vary in size and appearance depending on the stage of the cell cycle and the cell's function. In normal, healthy cells, nucleoli are usually small and have a distinct, well-defined border.
- Mitosis is the process of cell division that occurs in all living organisms. During mitosis, a single cell divides into two daughter cells, each of which contains a copy of the parent cell's DNA. The process of mitosis is essential for the growth and repair of tissues and the production of new cells.

```
 # importing dataset

df<- read.csv("breast_cancer.csv")
```

```
#studying the dataset
head(df)
```

```
##   Clump.Thickness Uniformity.of.Cell.Size Uniformity.of.Cell.Shape
## 1               5                       1                        1
## 2               5                       4                        4
## 3               3                       1                        1
## 4               6                       8                        8
## 5               4                       1                        1
## 6               8                      10                       10
##   Marginal.Adhesion Single.Epithelial.Cell.Size Bare.Nuclei Bland.Chromatin
## 1                 1                           2           1               3
## 2                 5                           7          10               3
## 3                 1                           2           2               3
## 4                 1                           3           4               3
## 5                 3                           2           1               3
## 6                 8                           7          10               9
##   Normal.Nucleoli Mitoses Class
## 1               1       1     2
## 2               2       1     2
## 3               1       1     2
## 4               7       1     2
## 5               1       1     2
## 6               7       1     4
```

Checking unique values in the column "Class".

```
unique(df$Class)
```

```
## [1] 2 4
```

```
# replacing the values with 0 and 1 for the purpose of building logistic regression model

df$Class <- ifelse(df$Class == 2, 0, 1)
```

This is important information. These two values refer to 'malignant' = 4 or 'benign' = 2. However, for the purpose of building logistic regression model I replaced these values with 0 for benign and 1 for malignant. Now I'm going to check if there are any missing values in this dataset.

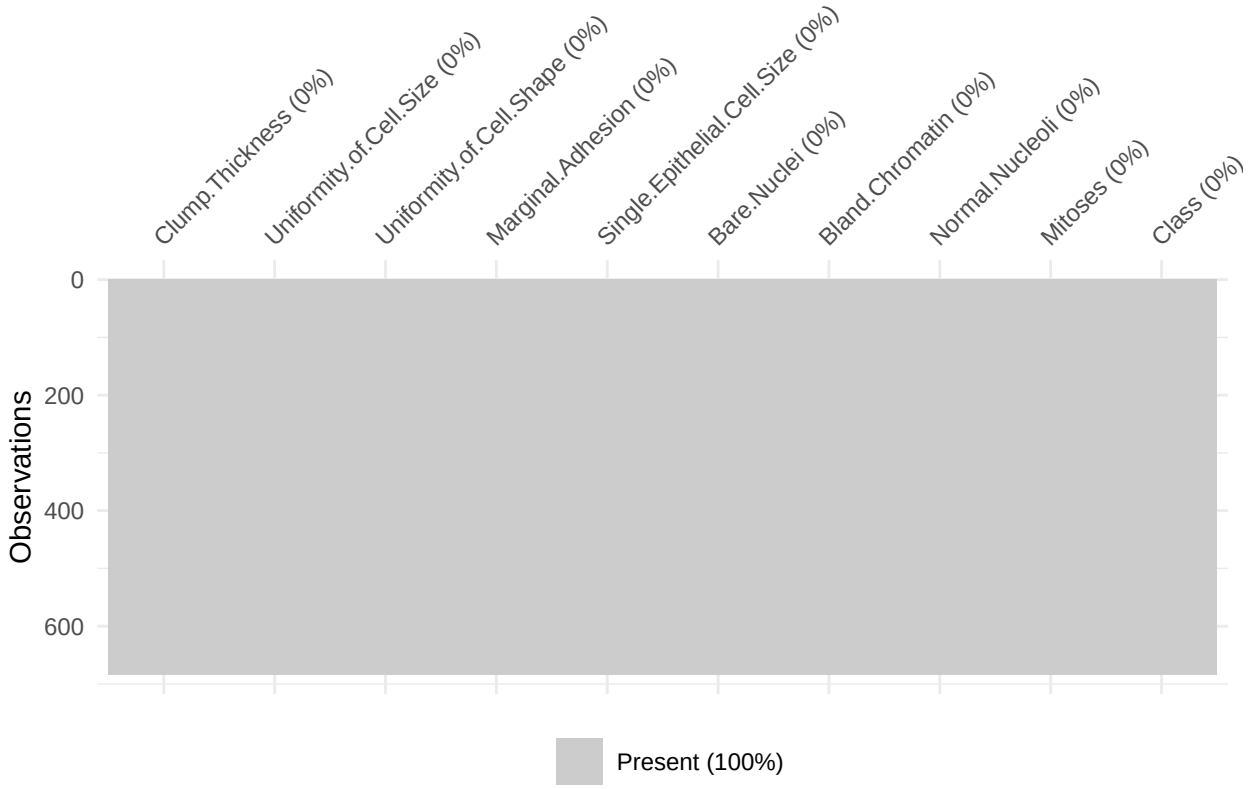
```
# checking if any columns have missing values

colSums(is.na(df))
```

```
##             Clump.Thickness     Uniformity.of.Cell.Size
##                           0                           0
##    Uniformity.of.Cell.Shape           Marginal.Adhesion
##                           0                           0
## Single.Epithelial.Cell.Size                Bare.Nuclei
##                           0                           0
##             Bland.Chromatin             Normal.Nucleoli
##                           0                           0
##                     Mitoses                       Class
##                           0                           0
```

There is also a visual way to check on the missing values.

```
# using vis_miss function to visually identify missing values
vis_miss(df)
```

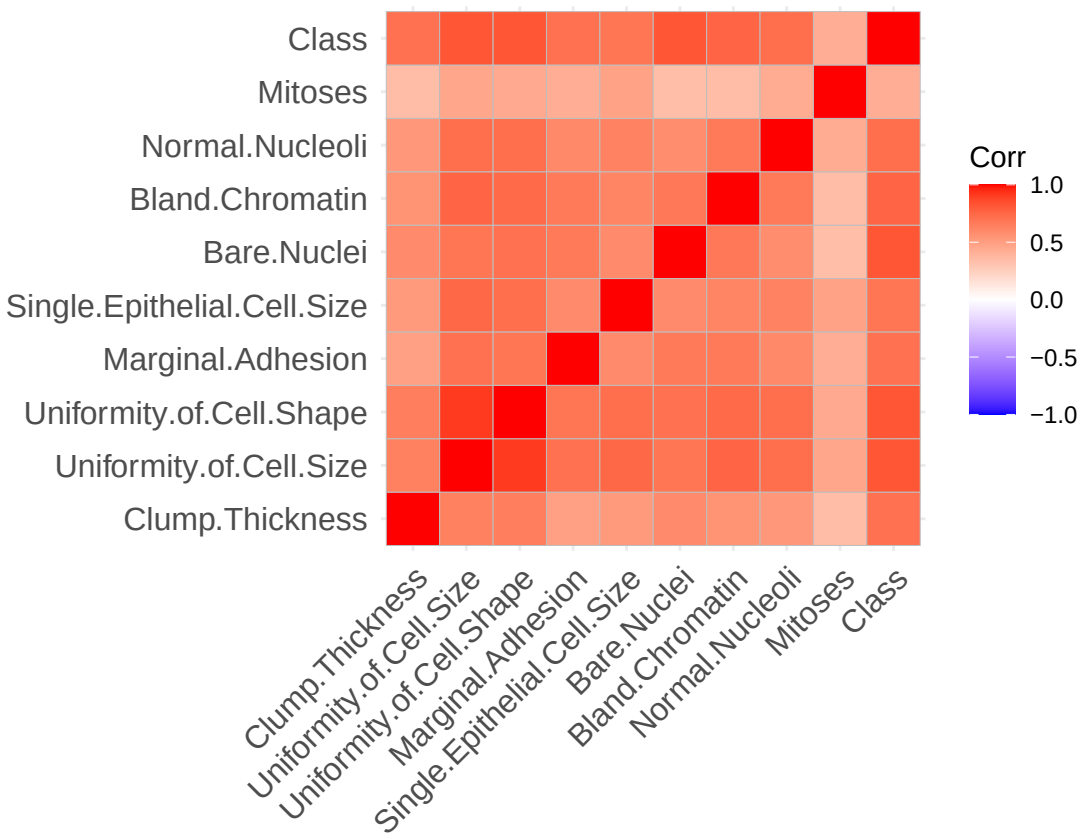


Correlation

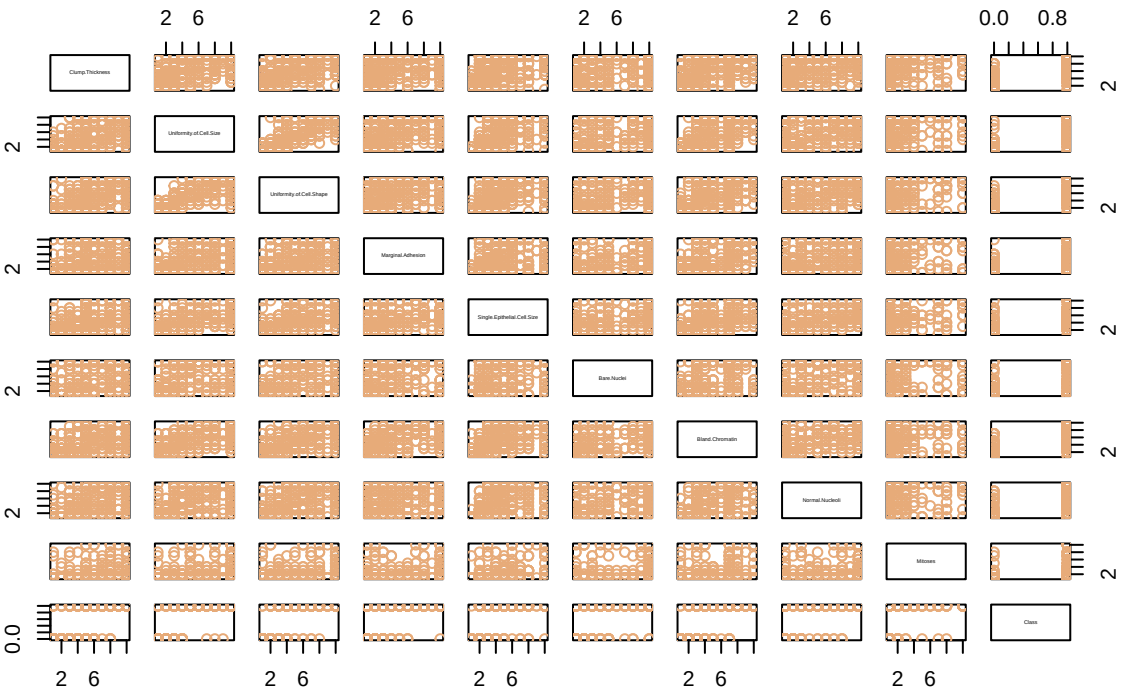
```
# finding correlations between the variables

correlation <- cor(df[,1:10])

ggcorrplot(correlation)
```




```r
# pairwise correlation
pairs(df,
      cex.labels = 0.3,
       col = c("#E7AB79"),
       pch = 21,
       main = "Pairwise correlation",
       col.labels = "black")
```



In order to train and test the model I need to split the data first.

```
# creating the index for the split
set.seed(123)
index <- createDataPartition(df$Class, p=0.8, times = 1, list=FALSE)

# splitting the data into train and test sets

train_set <- df%>% slice(index)
test_set <- df%>% slice(-index)
```

```
#checking that the split worked well
length(train_set$Class)
```

```
## [1] 547
```

```
length(test_set$Class)
```

```
## [1] 136
```

After the data has been split, I can start training my model.

Training logistic regression model

```
classifier <- glm(Class ~.,
                  family = binomial,  # specification of logistic regression
                  data = train_set)
```

This is a great feature of analysing data with R: the summary function that lets you see so much information from deviance to z and p-value (Pr(>|z|)). It is easy to read as this report puts 'asterisks' signs and explains the meaning. For example, 'asterisks' tell us that there's a strong correlation. It is important to keep in

mind though, that we could perform a backward elimination and by excluding some of the variables, the correlation shown by the summary() function may change.

```
summary(classifier)
```

```
##
## Call:
## glm(formula = Class ~ ., family = binomial, data = train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4023  -0.1108  -0.0687   0.0192   2.4740
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -9.84423    1.27072  -7.747 9.41e-15 ***
## Clump.Thickness               0.39364    0.15837   2.486 0.012932 *
## Uniformity.of.Cell.Size       0.16231    0.24179   0.671 0.502018
## Uniformity.of.Cell.Shape      0.26939    0.25095   1.073 0.283064
## Marginal.Adhesion             0.31655    0.14117   2.242 0.024943 *
## Single.Epithelial.Cell.Size   0.04137    0.20017   0.207 0.836265
## Bare.Nuclei                   0.41245    0.10605   3.889 0.000101 ***
## Bland.Chromatin               0.55779    0.21058   2.649 0.008078 **
## Normal.Nucleoli               0.14992    0.12754   1.175 0.239817
## Mitoses                       0.50687    0.34756   1.458 0.144747
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 708.985  on 546  degrees of freedom
## Residual deviance:  80.026  on 537  degrees of freedom
## AIC: 100.03
##
## Number of Fisher Scoring iterations: 8
```

We don't calculate R2 for logistic regression, as logistic regression doesn't have the same concept of residuals as a linear regression. Instead, logistic regression has "maximum likelihood".

```
# removing Uniformity of cell size to see if there are any differences
classifier_2 <- glm(Class ~.,
                    family = binomial,  # specification of logistic regression
                    data = train_set[-2])
summary(classifier_2)
```

```
##
## Call:
## glm(formula = Class ~ ., family = binomial, data = train_set[-2])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4349  -0.1102  -0.0665   0.0192   2.4872
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -9.95284    1.26065  -7.895 2.90e-15 ***
```

```
## Clump.Thickness              0.41228    0.15495   2.661  0.00780 **
## Uniformity.of.Cell.Shape     0.37467    0.18600   2.014  0.04398 *
## Marginal.Adhesion            0.31315    0.13981   2.240  0.02510 *
## Single.Epithelial.Cell.Size  0.06616    0.19412   0.341  0.73323
## Bare.Nuclei                  0.41096    0.10539   3.899 9.64e-05 ***
## Bland.Chromatin              0.56141    0.20586   2.727  0.00639 **
## Normal.Nucleoli              0.16950    0.12438   1.363  0.17297
## Mitoses                      0.52970    0.35060   1.511  0.13083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 708.985  on 546  degrees of freedom
## Residual deviance:  80.505  on 538  degrees of freedom
## AIC: 98.505
##
## Number of Fisher Scoring iterations: 8
```

Another way to optimize a model is with the usage of step() function. This function uses the AIC (Akaike information criterion) that helps exclude the variables that don't add to the accuracy of the model or bring the accuracy down. Let's run the function and see if the summary shows any differences.

```
# inputting original classifier
aic_model <- step(classifier)
```

```
## Start:  AIC=100.03
## Class ~ Clump.Thickness + Uniformity.of.Cell.Size + Uniformity.of.Cell.Shape +
##     Marginal.Adhesion + Single.Epithelial.Cell.Size + Bare.Nuclei +
##     Bland.Chromatin + Normal.Nucleoli + Mitoses
##
##                               Df Deviance     AIC
## - Single.Epithelial.Cell.Size  1   80.069  98.069
## - Uniformity.of.Cell.Size      1   80.505  98.505
## - Uniformity.of.Cell.Shape     1   81.108  99.108
## - Normal.Nucleoli              1   81.458  99.458
## <none>                             80.026 100.026
## - Mitoses                      1   83.439 101.439
## - Marginal.Adhesion            1   85.178 103.178
## - Clump.Thickness              1   87.253 105.253
## - Bland.Chromatin              1   87.829 105.829
## - Bare.Nuclei                  1   97.436 115.436
##
## Step:  AIC=98.07
## Class ~ Clump.Thickness + Uniformity.of.Cell.Size + Uniformity.of.Cell.Shape +
##     Marginal.Adhesion + Bare.Nuclei + Bland.Chromatin + Normal.Nucleoli +
##     Mitoses
##
##                             Df Deviance     AIC
## - Uniformity.of.Cell.Size    1   80.622  96.622
## - Uniformity.of.Cell.Shape   1   81.198  97.198
## - Normal.Nucleoli            1   81.568  97.568
## <none>                           80.069  98.069
## - Mitoses                    1   83.462  99.462
## - Marginal.Adhesion          1   85.840 101.840
```

```
## - Clump.Thickness           1   87.295 103.295
## - Bland.Chromatin           1   88.460 104.460
## - Bare.Nuclei               1   99.519 115.519
##
## Step:  AIC=96.62
## Class ~ Clump.Thickness + Uniformity.of.Cell.Shape + Marginal.Adhesion +
##     Bare.Nuclei + Bland.Chromatin + Normal.Nucleoli + Mitoses
##
##                            Df Deviance     AIC
## <none>                          80.622  96.622
## - Normal.Nucleoli           1   82.885  96.885
## - Mitoses                   1   84.596  98.596
## - Uniformity.of.Cell.Shape  1   85.883  99.883
## - Marginal.Adhesion         1   86.917 100.917
## - Clump.Thickness           1   89.259 103.259
## - Bland.Chromatin           1   90.412 104.412
## - Bare.Nuclei               1  100.237 114.237
```

```
# checking the output for the best model found by the step() function

summary(aic_model)
```

```
##
## Call:
## glm(formula = Class ~ Clump.Thickness + Uniformity.of.Cell.Shape +
##     Marginal.Adhesion + Bare.Nuclei + Bland.Chromatin + Normal.Nucleoli +
##     Mitoses, family = binomial, data = train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4743  -0.1099  -0.0661   0.0187   2.4462
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -9.9190     1.2572  -7.890 3.02e-15 ***
## Clump.Thickness            0.4156     0.1547   2.687  0.00722 **
## Uniformity.of.Cell.Shape   0.3879     0.1834   2.115  0.03441 *
## Marginal.Adhesion          0.3258     0.1341   2.429  0.01515 *
## Bare.Nuclei                0.4188     0.1027   4.077 4.56e-05 ***
## Bland.Chromatin            0.5767     0.2004   2.878  0.00400 **
## Normal.Nucleoli            0.1785     0.1220   1.464  0.14330
## Mitoses                    0.5253     0.3481   1.509  0.13120
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 708.985  on 546  degrees of freedom
## Residual deviance:  80.622  on 539  degrees of freedom
## AIC: 96.622
##
## Number of Fisher Scoring iterations: 8
```

The step() has found the most efficient model with the lowest AIC.

The function excluded 2 variables: Uniformity.of.Cell.Size and Single.Epithelial.Cell.Size.

Now the model can be tested on our test set.

```
predicted_probability <- predict(aic_model,
                                 type = "response",  # gives result listed in a single vector
                                 newdata = test_set[-10])  # removing dep. var. column

y_hat <- ifelse(predicted_probability > 0.5, 1, 0)
```

Confusion Matrix

```
cm <- table(test_set[,10], y_hat)
cm
```

```
##    y_hat
##      0  1
##   0 87  2
##   1  4 43
```

```
# the accuracy of the model

accuracy <- ((cm[1,1] + cm[2,2]) / (cm[1,1] + cm[2,2] + cm[2,1] + cm[1,2]))*100
cat(paste("Accuracy: ", round(accuracy,2), "%"))
```

```
## Accuracy:  95.59 %
```

```
# when a model predicts a positive value, how often is it right?

precision <- (cm[1,1] / (cm[1,1] + cm[1,2]))*100
cat(paste("Precision: ", round(precision,2), "%"))
```

```
## Precision:  97.75 %
```

```
# recall - the model's ability to predict positive values
# (how often does a model predict the correct positive values)

recall <- (cm[1,1] / (cm[1,1] + cm[2,1]))*100
cat(paste("Recall: ", round(recall,2), "%"))
```

```
## Recall:  95.6 %
```

```
# harmonic average of the precision and recall
F_1 <- (2*precision * recall) / (precision + recall)
cat(paste("F_1 score: ", round(F_1,2), "%"))
```

```
## F_1 score:  96.67 %
```

Cross-validation of the model

```
# monte carlo simulation

# deleting the variables that the function step() excluded
df <- subset(df, select = -c(Uniformity.of.Cell.Size,Single.Epithelial.Cell.Size))

lg <- replicate(100, {
```

```
 # splitting the data
 index <- createDataPartition(df$Class, p=0.8, times = 1, list=FALSE)
 train_set <- df%>% slice(index)
 test_set <- df%>% slice(-index)


 # building the model
 classifier <- glm(Class ~.,
                   family = binomial,
                   data = train_set)

 predicted_probability <- predict(classifier,
                                  type = "response",
                                  newdata = test_set[-10])

 y_hat <- ifelse(predicted_probability > 0.5, 1, 0)

 correct_predictions <- sum(y_hat == test_set$Class)

 total_predictions <- length(y_hat)

 accuracy <- correct_predictions / total_predictions

 return(accuracy)

})

paste("Cross-validated accuracy of the model: ",mean(lg)*100)
```

```
## [1] "Cross-validated accuracy of the model:  97.0808823529412"
```

A we can see, the model returned a pretty good result.

Visualization

Now we can visualize the data. I will first create separate plots for the variables and then I will combine them together for an easier way to compare the logistic regression lines.

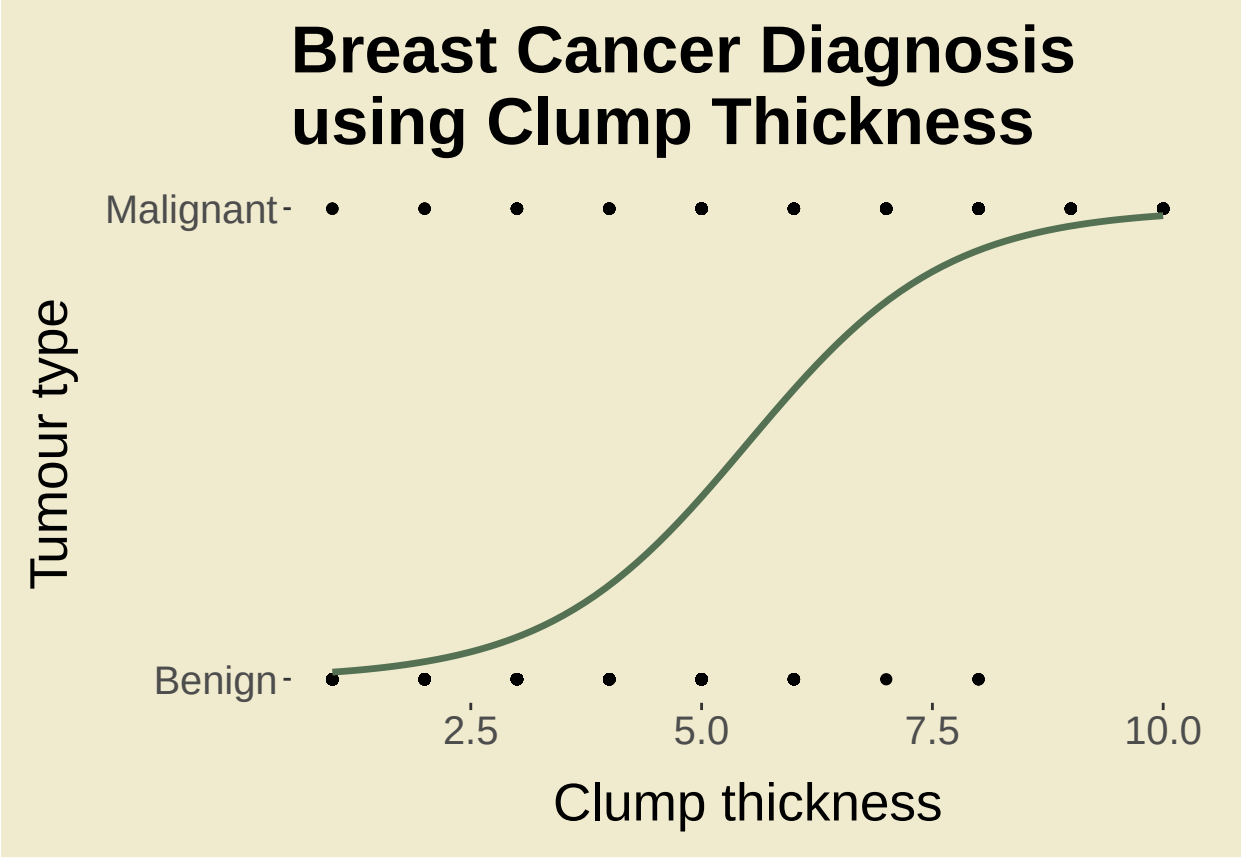
I will plot each feature one by one first and then create a combined plot.

I could use library(gridExtra) arranging existing plots together, but unfortunately it doesn't create visually the result I need, so I will create 2 slightly different plots per feature and then combine the result.

```
# clump thickness plot
ggplot() +
  geom_point(aes(df$Clump.Thickness, df$Class)) +
  geom_smooth(aes(df$Clump.Thickness, df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
  ggtitle("Breast Cancer Diagnosis \nusing Clump Thickness") +
  ylab("Tumour type") +
  xlab("Clump thickness") +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.title = element_text(size = 25, face="bold",
                                  margin = margin(10,0,10,0)),
```

```
      plot.background = element_rect(fill = "#F0EBCE"),
      panel.background = element_rect(fill = "#F0EBCE"),
      axis.text.x = element_text(size = 15),
      axis.title.x = element_text(size = 20, margin = margin(11,0,10,0)),
      axis.text.y = element_text(size = 15),
      axis.title.y = element_text(size = 20, margin=margin(0,10,0,11)),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      plot.margin = margin(0,0.5,0,0, "cm"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

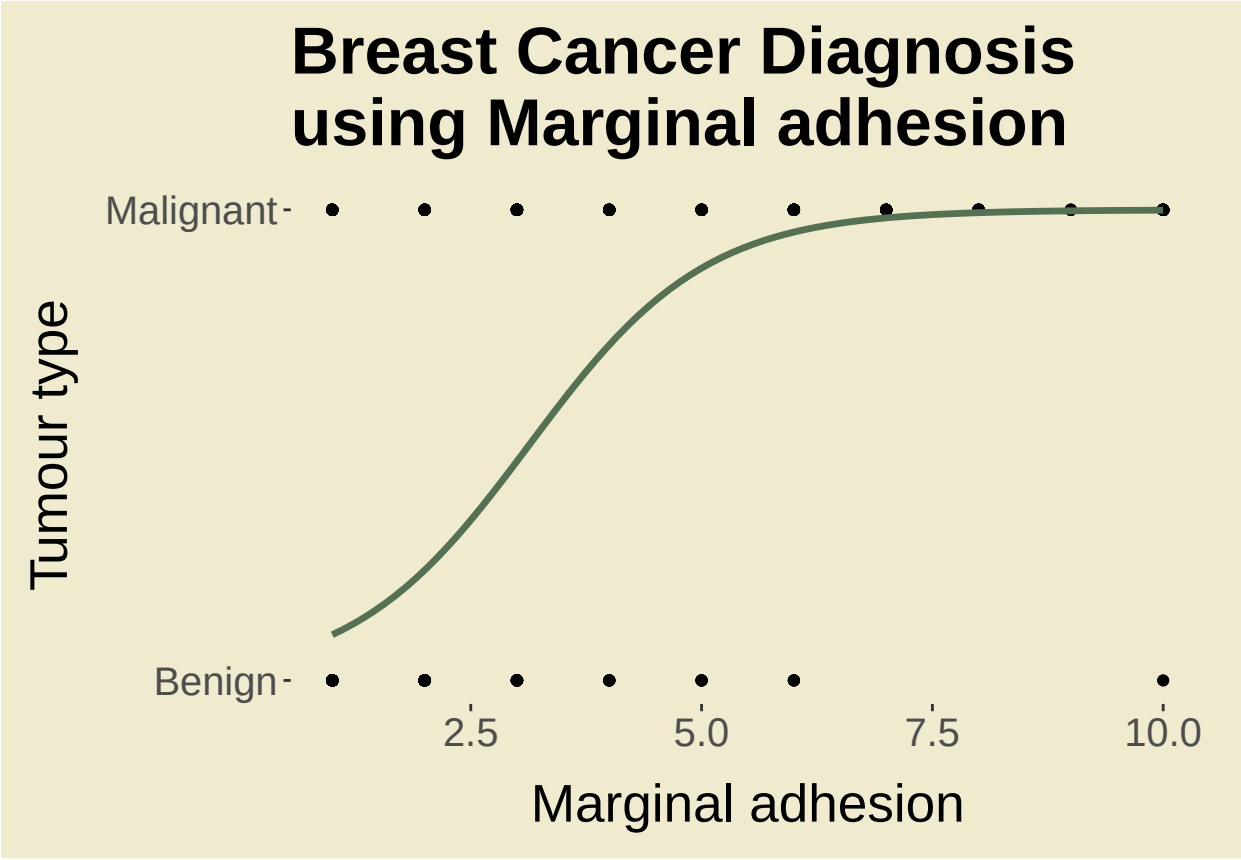



```
# additional plot for combination
ct <- ggplot() +
  geom_point(aes(df$Clump.Thickness,df$Class)) +
  geom_smooth(aes(df$Clump.Thickness,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.background = element_rect(fill = "#F0EBCE"),
      panel.background = element_rect(fill = "#F0EBCE"),
      axis.text.x = element_text(size = 10),
      axis.title.x = element_blank(),
      axis.text.y = element_text(size = 10),
      axis.title.y = element_blank(),
      panel.grid.major = element_blank(),
```

```
      panel.grid.minor = element_blank(),
      plot.margin = margin(0,0.5, 0, 0, "cm")) +
      annotate("text",x=7.0, y=0.2, label="Clump thickness", size = 6)
```

```
# marginal adhesion plot
ggplot() +
  geom_point(aes(df$Marginal.Adhesion,df$Class)) +
  geom_smooth(aes(df$Marginal.Adhesion,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
  ggtitle("Breast Cancer Diagnosis \nusing Marginal adhesion") +
  ylab("Tumour type") +
  xlab("Marginal adhesion") +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.title = element_text(size = 25, face="bold", margin = margin(10,0,10,0)),
      plot.background = element_rect(fill = "#F0EBCE"),
      panel.background = element_rect(fill = "#F0EBCE"),
      axis.text.x = element_text(size = 15),
      axis.title.x = element_text(size = 20, margin = margin(11,0,10,0)),
      axis.text.y = element_text(size = 15),
      axis.title.y = element_text(size = 20, margin=margin(0,10,0,11)),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      plot.margin = margin(0,0.5,0,0, "cm"))
```

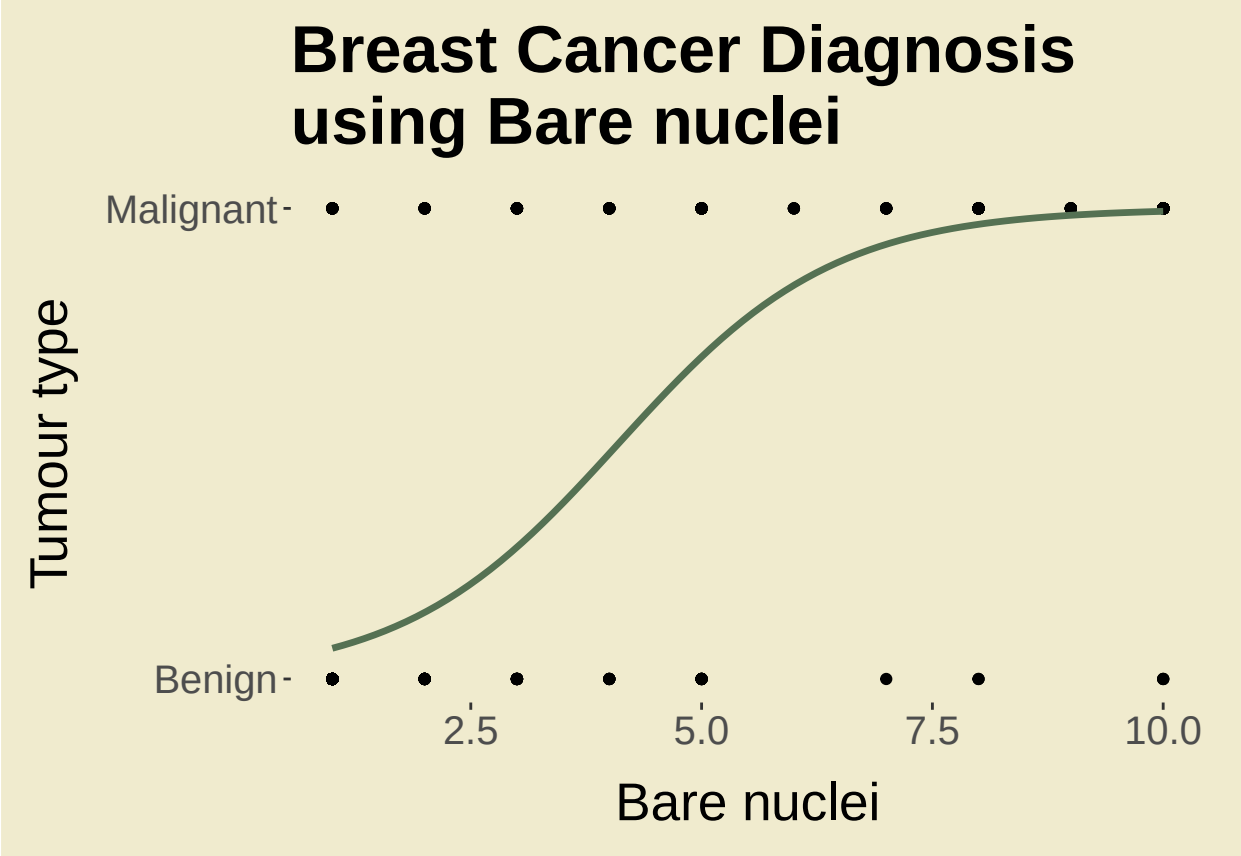
```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
# additional plot for combination
ma <- ggplot() +
  geom_point(aes(df$Marginal.Adhesion,df$Class)) +
  geom_smooth(aes(df$Marginal.Adhesion,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.background = element_rect(fill = "#F0EBCE"),
        panel.background = element_rect(fill = "#F0EBCE"),
        axis.text.x = element_text(size = 10),
        axis.title.x = element_blank(),
        axis.text.y = element_text(size = 10),
        axis.title.y = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.margin = margin(0,0.5, 0, 0, "cm")) +
        annotate("text",x=6.5, y=0.2, label="Marginal adhesion", size = 6)
```

```r
# Bare nuclei plot
ggplot() +
  geom_point(aes(df$Bare.Nuclei,df$Class)) +
  geom_smooth(aes(df$Bare.Nuclei,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
  ggtitle("Breast Cancer Diagnosis \nusing Bare nuclei") +
  ylab("Tumour type") +
  xlab("Bare nuclei") +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.title = element_text(size = 25, face="bold", margin = margin(10,0,10,0)),
        plot.background = element_rect(fill = "#F0EBCE"),
        panel.background = element_rect(fill = "#F0EBCE"),
        axis.text.x = element_text(size = 15),
        axis.title.x = element_text(size = 20, margin = margin(11,0,10,0)),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 20, margin=margin(0,10,0,11)),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.margin = margin(0,0.5,0,0, "cm"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

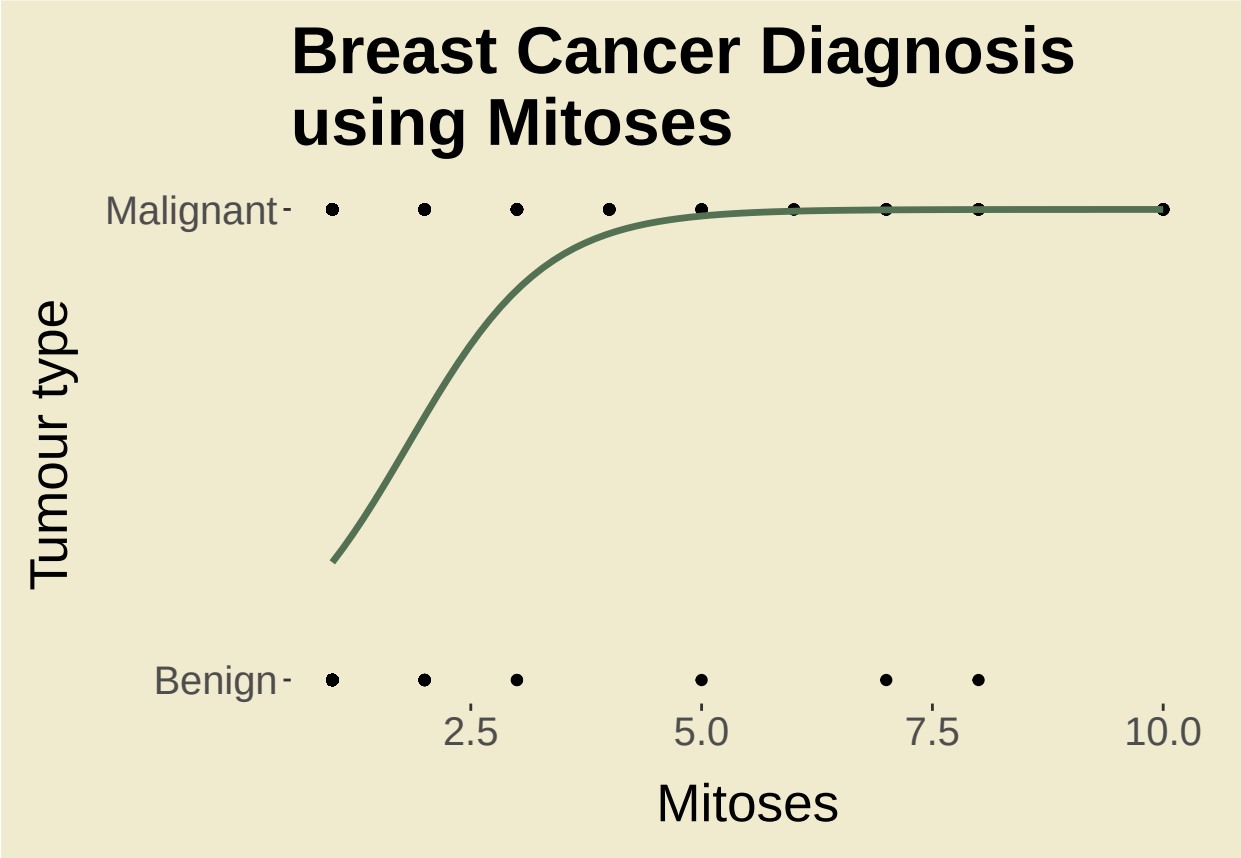



```
# additional plot for combination
bn <- ggplot() +
  geom_point(aes(df$Bare.Nuclei,df$Class)) +
  geom_smooth(aes(df$Bare.Nuclei,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.background = element_rect(fill = "#F0EBCE"),
        panel.background = element_rect(fill = "#F0EBCE"),
        axis.text.x = element_text(size = 10),
        axis.title.x = element_blank(),
        axis.text.y = element_text(size = 10),
        axis.title.y = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.margin = margin(0,0.5, 0, 0, "cm")) +
        annotate("text",x=6.5, y=0.2, label="Bare nuclei", size = 6)
```

```
# Mitoses plot
ggplot() +
  geom_point(aes(df$Mitoses,df$Class)) +
  geom_smooth(aes(df$Mitoses,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
  ggtitle("Breast Cancer Diagnosis \nusing Mitoses") +
  ylab("Tumour type") +
  xlab("Mitoses") +
```

```
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.title = element_text(size = 25, face="bold",margin = margin(10,0,10,0)),
        plot.background = element_rect(fill = "#F0EBCE"),
        panel.background = element_rect(fill = "#F0EBCE"),
        axis.text.x = element_text(size = 15),
        axis.title.x = element_text(size = 20, margin = margin(11,0,10,0)),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 20, margin=margin(0,10,0,11)),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.margin = margin(0,0.5,0,0, "cm"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

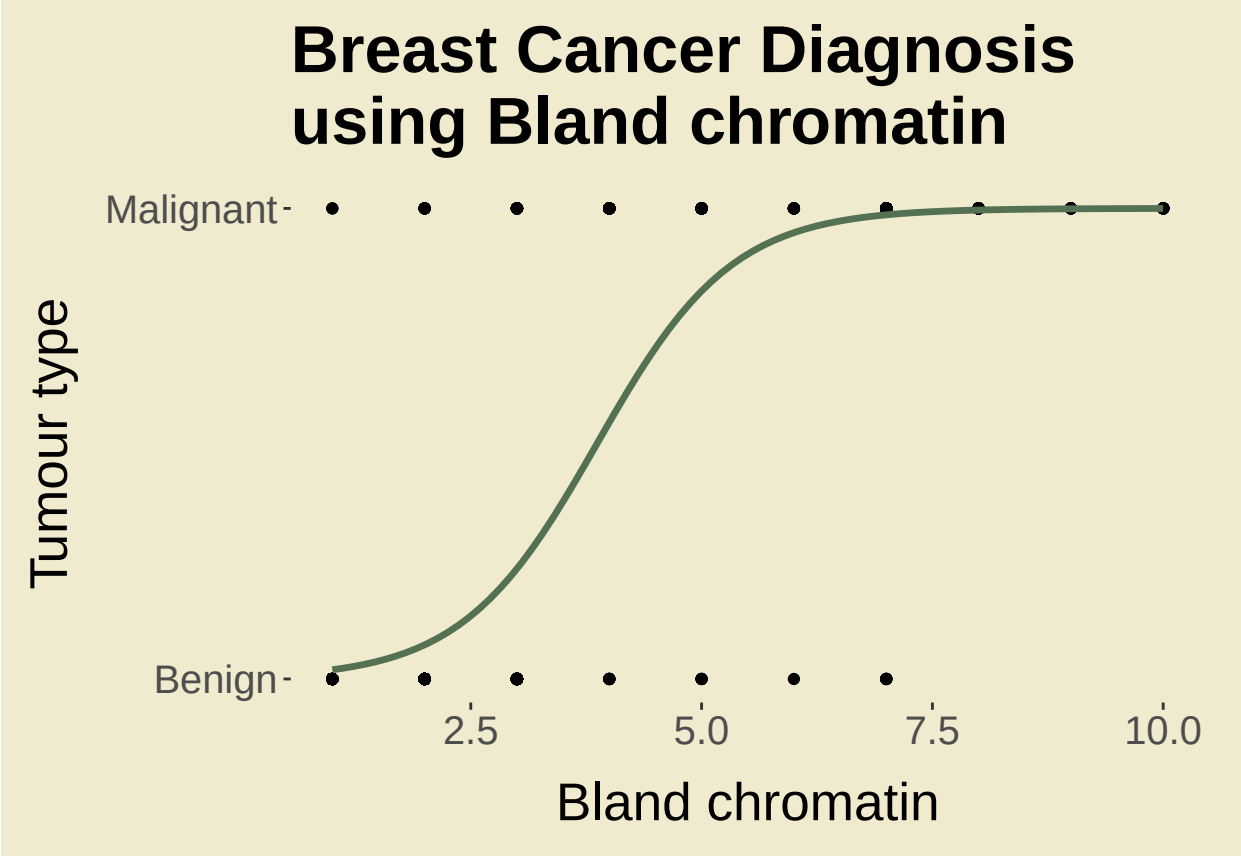



```
# additional plot for combination
um <- ggplot() +
  geom_point(aes(df$Mitoses,df$Class)) +
  geom_smooth(aes(df$Mitoses,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.background = element_rect(fill = "#F0EBCE"),
        panel.background = element_rect(fill = "#F0EBCE"),
        axis.text.x = element_text(size = 10),
        axis.title.x = element_blank(),
        axis.text.y = element_text(size = 10),
```

```
      axis.title.y = element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      plot.margin = margin(0,0.5, 0, 0, "cm")) +
      annotate("text",x=6.5, y=0.2, label="Mitoses", size = 6)
```

```
# bland chromatin plot
ggplot() +
  geom_point(aes(df$Bland.Chromatin,df$Class)) +
  geom_smooth(aes(df$Bland.Chromatin,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
  ggtitle("Breast Cancer Diagnosis \nusing Bland chromatin") +
  ylab("Tumour type") +
  xlab("Bland chromatin") +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.title = element_text(size = 25, face="bold", margin = margin(10,0,10,0)),
      plot.background = element_rect(fill = "#F0EBCE"),
      panel.background = element_rect(fill = "#F0EBCE"),
      axis.text.x = element_text(size = 15),
      axis.title.x = element_text(size = 20, margin = margin(11,0,10,0)),
      axis.text.y = element_text(size = 15),
      axis.title.y = element_text(size = 20, margin=margin(0,10,0,11)),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      plot.margin = margin(0,0.5,0,0, "cm"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

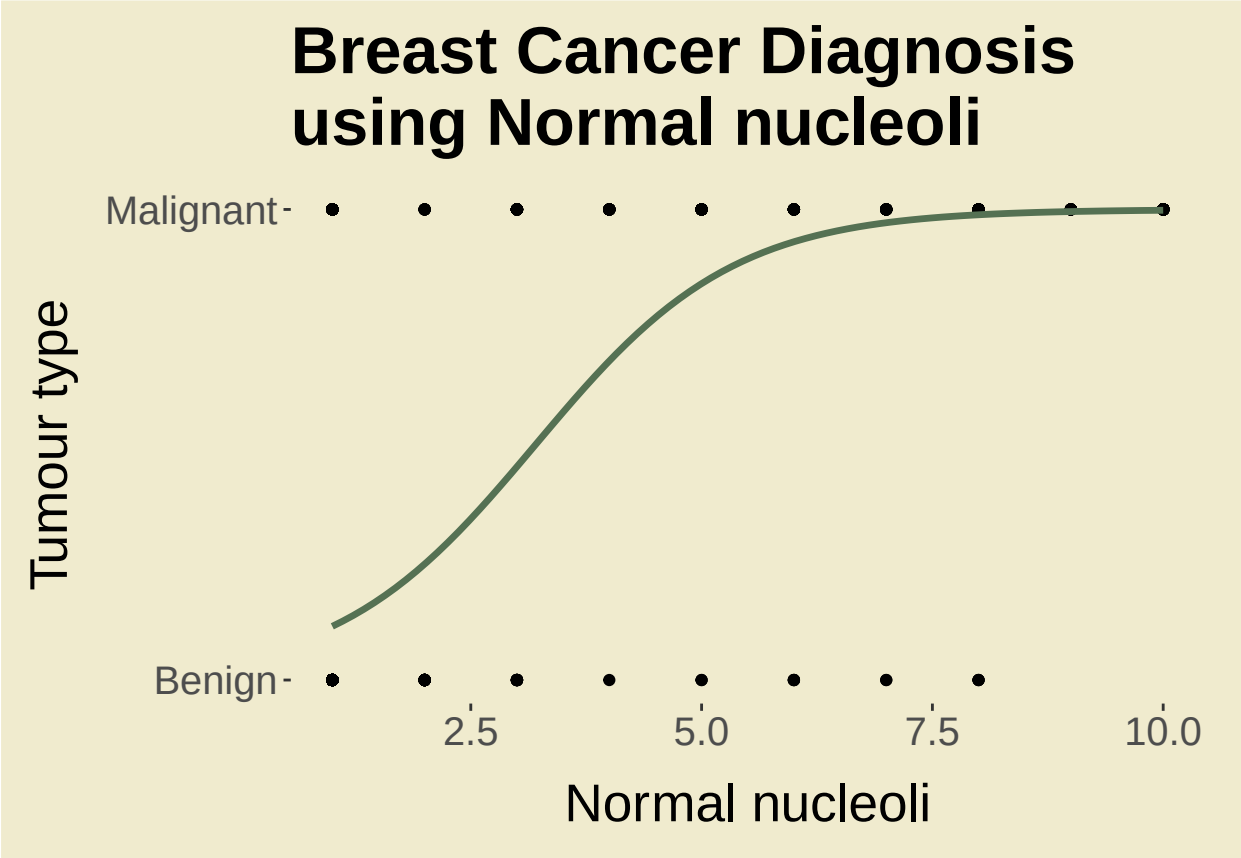



```
# additional plot for combination
bc <- ggplot() +
  geom_point(aes(df$Bland.Chromatin,df$Class)) +
  geom_smooth(aes(df$Bland.Chromatin,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.background = element_rect(fill = "#F0EBCE"),
        panel.background = element_rect(fill = "#F0EBCE"),
        axis.text.x = element_text(size = 10),
        axis.title.x = element_blank(),
        axis.text.y = element_text(size = 10),
        axis.title.y = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.margin = margin(0,0.5, 0, 0, "cm")) +
        annotate("text",x=6.5, y=0.2, label="Bland chromatin", size = 6)
```

```
# normal nucleoli plot
 ggplot() +
  geom_point(aes(df$Normal.Nucleoli,df$Class)) +
  geom_smooth(aes(df$Normal.Nucleoli,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
  ggtitle("Breast Cancer Diagnosis \nusing Normal nucleoli") +
  ylab("Tumour type") +
  xlab("Normal nucleoli") +
```

```
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.title = element_text(size = 25, face="bold",margin = margin(10,0,10,0)),
        plot.background = element_rect(fill = "#F0EBCE"),
        panel.background = element_rect(fill = "#F0EBCE"),
        axis.text.x = element_text(size = 15),
        axis.title.x = element_text(size = 20, margin = margin(11,0,10,0)),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 20, margin=margin(0,10,0,11)),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.margin = margin(0,0.5,0,0, "cm"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# additional plot for combination
nn <- ggplot() +
  geom_point(aes(df$Normal.Nucleoli,df$Class)) +
  geom_smooth(aes(df$Normal.Nucleoli,df$Class),
              method = "glm", se = FALSE, method.args = list(family = "binomial"),
              color = "#557153", size = 1.2) +
scale_y_continuous(breaks = c(0, 1), labels = c("Benign", "Malignant")) +
  theme(plot.background = element_rect(fill = "#F0EBCE"),
        panel.background = element_rect(fill = "#F0EBCE"),
        axis.text.x = element_text(size = 10),
        axis.title.x = element_blank(),
        axis.text.y = element_text(size = 10),
```

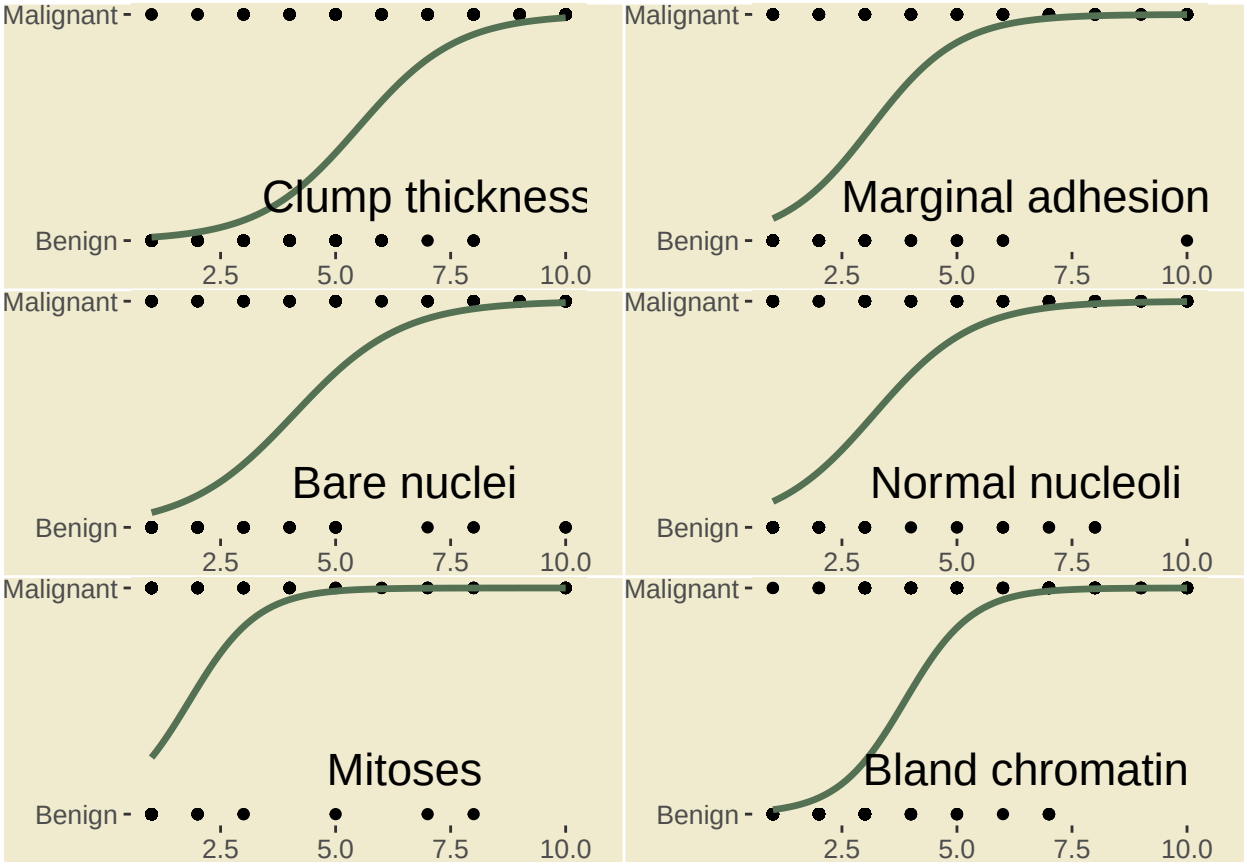
```
        axis.title.y = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.margin = margin(0,0.5, 0, 0, "cm")) +
        annotate("text",x=6.5, y=0.2, label="Normal nucleoli", size = 6)
```

Combining the plots

```
# gathering all the plots together so they are easier to compare

ggarrange(ct, ma, bn, nn, um, bc,
          ncol = 2, nrow = 3)
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



Conclusion

We have studied different measures typically used to test a patient for breast cancer. We have built a logistic regression model, using these measures, and seen that it works well classifying values. We made sure we have built the most efficient model with the help of backwards elimination and AIC score through step() function.

References

Dataset source, UCI Machine Learning Repository.

Irizarry, R A 2019, 'Introduction to Data Science', CRC Press, Boca Raton.

Johns Hopkins University, 2023, *Glossary of Breast Cancer Terms*, <https://pathology.jhu.edu/breast/glossary>.

Sarkar, S K, Nag, A, 2017, *Identifying Patients at Risk of Breast Cancer through Decision Trees*, viewed 15 January, 2023, <https://www.researchgate.net/publication/325868350_Identifying_Patients_at_Risk_of_Breast_Cancer_through_Decision_Trees>