Individual Project

2023-11-15

Data Set:

Obtained from the 2017 November update of the SEER (Surveillance, Epidemiology, and End Results) program of the National Cancer Institute: https://www.kaggle.com/datasets/reihanenamdari/breast-cancer/data. Patients whose survival months were less than 1 month were excluded; thus, 4024 patients were ultimately included.

Variables:

T- stage: is related with tumor size and if it has spread.

N- Stage: indicates the number of lymph nodes involved and how much cancer is found in them. The higher the N number, the greater the extent of the lymph node involvement.

A.stage: refer to M refers to whether the cancer is metastatic (it has spread to distant parts of the body).

Differentiate: indicates how similar is a cancer cell to a normal cell, if it is similar is "well differentiated"

Progesterone and Estrogen Status: indicates hormone receptors in cancer cells.

Tumor Size: Measured in millimeters.

6th Stage: Indicates tumor node and metastasis state.

```r
## Read the dataframe
df= read.csv('Breast_Cancer.csv', header=TRUE)
head(df)
```

```
##   Age  Race Marital.Status T.Stage N.Stage X6th.Stage             differentiate
## 1  68 White        Married      T1      N1        IIA     Poorly differentiated
## 2  50 White        Married      T2      N2       IIIA Moderately differentiated
## 3  58 White       Divorced      T3      N3       IIIC Moderately differentiated
## 4  58 White        Married      T1      N1        IIA     Poorly differentiated
## 5  47 White        Married      T2      N1        IIB     Poorly differentiated
## 6  51 White         Single      T1      N1        IIA Moderately differentiated
##   Grade  A.Stage Tumor.Size Estrogen.Status Progesterone.Status
## 1     3 Regional          4        Positive            Positive
## 2     2 Regional         35        Positive            Positive
## 3     2 Regional         63        Positive            Positive
## 4     3 Regional         18        Positive            Positive
## 5     3 Regional         41        Positive            Positive
## 6     2 Regional         20        Positive            Positive
##   Regional.Node.Examined Reginol.Node.Positive Survival.Months Status
## 1                     24                     1              60  Alive
## 2                     14                     5              62  Alive
## 3                     14                     7              75  Alive
## 4                      2                     1              84  Alive
## 5                      3                     1              50  Alive
## 6                     18                     2              89  Alive
```

```r
## There is no missing data
sum(is.na(df))
```

```
## [1] 0
```

```r
## Dimensions
dim(df)
```

```
## [1] 4024   16
```

Changing categorical and binary to numerical:

```r
library(dplyr)
```

```
## 
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
## 
##     filter, lag
```

```
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

```r
library(data.table)
```

```
## 
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
## 
##     between, first, last
```

```r
library(caret)
```

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

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

```r
#Categorical data conversion
df2<- copy(df)
lapply(df2,unique)
```

```
## $Age
##  [1] 68 50 58 47 51 40 69 46 65 48 62 61 56 43 60 57 55 63 66 53 59 54 49 64 42
## [26] 37 67 31 52 33 45 38 39 36 41 44 32 34 35 30
##
## $Race
## [1] "White" "Black" "Other"
##
## $Marital.Status
## [1] "Married"   "Divorced"  "Single "   "Widowed"   "Separated"
##
## $T.Stage
## [1] "T1" "T2" "T3" "T4"
##
## $N.Stage
## [1] "N1" "N2" "N3"
##
## $X6th.Stage
## [1] "IIA"  "IIIA" "IIIC" "IIB"  "IIIB"
##
## $differentiate
## [1] "Poorly differentiated"     "Moderately differentiated"
## [3] "Well differentiated"       "Undifferentiated"
##
## $Grade
## [1] "3"                     "2"                     "1"
## [4] " anaplastic; Grade IV"
##
## $A.Stage
## [1] "Regional" "Distant"
##
## $Tumor.Size
##   [1]   4  35  63  18  41  20   8  30 103  32  13  59  15  19  46  24  25  29
##  [19]  40  70  22  50  17  21  10  27  23   5  51   9  55 120  77   2  11  12
##  [37]  26  75 130  34  80   3  60  14  16  45  36  76  38  49   7  72 100  43
##  [55]  62  37  68  52  85  57  39  28  48 110  65   6 105 140  42  31  90 108
##  [73]  98  47  54  61  74  33   1  87  81  58 117  44 123 133  95 107  92  69
##  [91]  56  82  66  78  97  88  53  83 101  84 115  73 125 104  94  86  64  96
## [109]  79  67
##
## $Estrogen.Status
## [1] "Positive" "Negative"
##
## $Progesterone.Status
## [1] "Positive" "Negative"
##
## $Regional.Node.Examined
##  [1] 24 14  2  3 18 11  9 20 21 13 23 16  1 22 15  4 26 31 25 10  5  6 19 12  8
## [26] 17  7 49 33 30 34 28 32 27 42 29 41 39 46 40 51 44 38 47 54 36 61 37 35 43
## [51] 52 45 57 60
```

```
##
## $Reginol.Node.Positive
##  [1]  1  5  7  2 18 12  3 14 22 17 23  4 10  6  9  8 20 16 13 11 24 27 21 26 15
## [26] 28 19 29 31 46 33 37 30 35 25 32 41 34
##
## $Survival.Months
##   [1]  60  62  75  84  50  89  54  14  70  92  64  56  38  49 105 107  77  81
##  [19]  78 102  98  82  86  52  90  31  37 103  42  61  63  39  59  71  74  73
##  [37]  91 106  80  44  85  79 104  12  95  55 101  65  72  57  87  40  25   8
##  [55]  53  58  24  66  69  93  94 100  96  41  67  51  13  11  47  23  45  68
##  [73]  76  15  16  99   7  48  88  34  97  83  17   3  22  30   6  32   9   5
##  [91]  10  19  18  35  27  36   4  29  33  26  20  28  43   1  46  21   2
##
## $Status
## [1] "Alive" "Dead"
```

```
# T-stage, N-stage, X6th.Stage and Differentiated columns.
values <- c("T1"=1,"T2"=2,"T3"=3,"T4"=4,"N1"=1,"N2"=2,"N3"=3,"Poorly differentiated"=3,"Moderately diffe
colv <- c("T.Stage","N.Stage","differentiate","X6th.Stage")
df2 <- df2 %>%
mutate_at(vars(all_of(colv)), ~recode(., !!!values)) #re-coding the data
df2[colv] <- lapply(df2[colv], function(x) factor(x,ordered=TRUE)) #Changing it to ordered factor

##Levels
levels(df2$T.Stage) <- c( "T1","T2","T3","T4")
levels(df2$N.Stage) <- c( "N1","N2","N3")
levels(df2$differentiate) <- c( "Well differentiated","Moderately differentiated","Poorly differentiate
levels(df2$X6th.Stage) <- c("IIA","IIB","IIIA","IIIB","IIIC")

# Binary variables
df2$Estrogen.Status <- ifelse(df2$Estrogen.Status=="Positive",1,0) # Cancer cells have receptors for es
df2$Progesterone.Status <- ifelse(df2$Progesterone.Status =="Positive",1,0) # Cancer cells have recepto

df2$A.Stage <-ifelse(df2$A.Stage=="Distant",1,0)
df2$Status <- ifelse(df2$Status=="Dead",1,0)


# Eliminate Grade
table(df2$differentiate,df2$Grade) #Differentiate and Grade are the same, I am going to eliminate grade
```

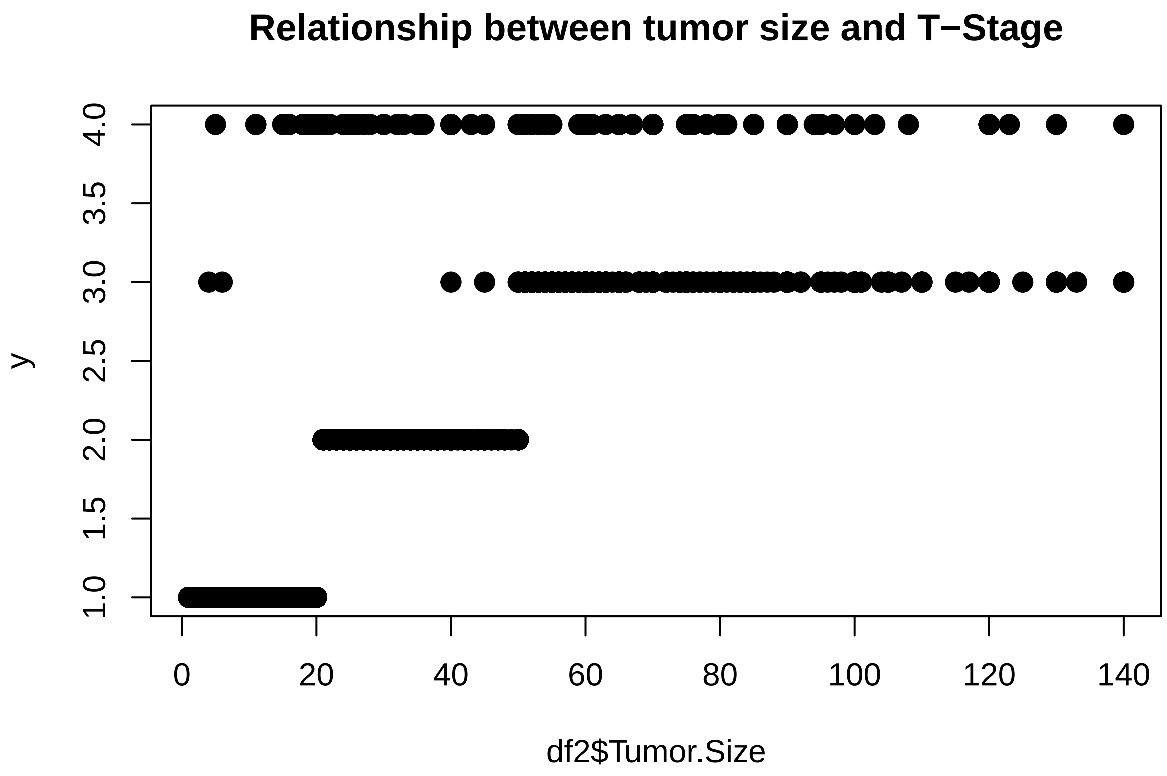
```
##
##                                 anaplastic; Grade IV    1    2    3
##   Well differentiated                              0  543    0    0
##   Moderately differentiated                        0    0 2351    0
##   Poorly differentiated                            0    0    0 1111
##   Undifferentiated                                19    0    0    0
```

```
df2$Grade <- NULL

#Categorical variables
df2$Race <- as.factor(df2$Race)
df2$Marital.Status <- as.factor(df2$Marital.Status)
str(df2)
```

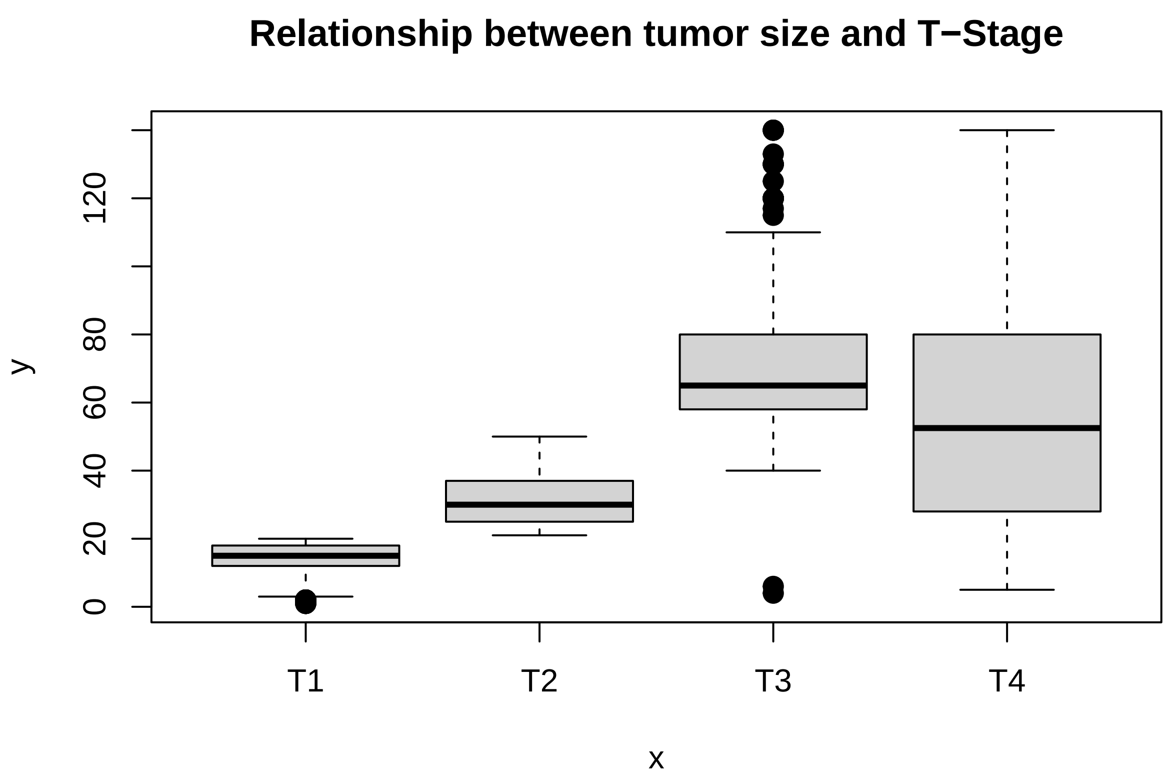
```
## ’data.frame’:    4024 obs. of  15 variables:
##  $ Age                   : int  68 50 58 58 47 51 51 40 40 69 ...
##  $ Race                  : Factor w/ 3 levels "Black","Other",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Marital.Status        : Factor w/ 5 levels "Divorced","Married",..: 2 2 1 2 2 4 2 2 1 2 ...
##  $ T.Stage               : Ord.factor w/ 4 levels "T1"<"T2"<"T3"<..: 1 2 3 1 2 1 1 2 4 4 ...
##  $ N.Stage               : Ord.factor w/ 3 levels "N1"<"N2"<"N3": 1 2 3 1 1 1 1 1 3 3 ...
##  $ X6th.Stage            : Ord.factor w/ 5 levels "IIA"<"IIB"<"IIIA"<..: 1 3 5 1 2 1 1 2 5 5 ...
##  $ differentiate         : Ord.factor w/ 4 levels "Well differentiated"<..: 3 2 2 3 3 2 1 2 3 1 ...
##  $ A.Stage               : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Tumor.Size            : int  4 35 63 18 41 20 8 30 103 32 ...
##  $ Estrogen.Status       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Progesterone.Status   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Regional.Node.Examined: int  24 14 14 2 3 18 11 9 20 21 ...
##  $ Reginol.Node.Positive : int  1 5 7 1 1 2 1 1 18 12 ...
##  $ Survival.Months       : int  60 62 75 84 50 89 54 14 70 92 ...
##  $ Status                : num  0 0 0 0 0 0 0 1 0 0 ...
```

Data visualizations.

```r
# Visualizations
y=as.numeric(df2$T.Stage)
plot(df2$Tumor.Size, y,pch=20, cex=2,main="Relationship between tumor size and T-Stage")
```



```r
plot(df2$T.Stage, df2$Tumor.Size,pch=20, cex=2,main="Relationship between tumor size and T-Stage")
```



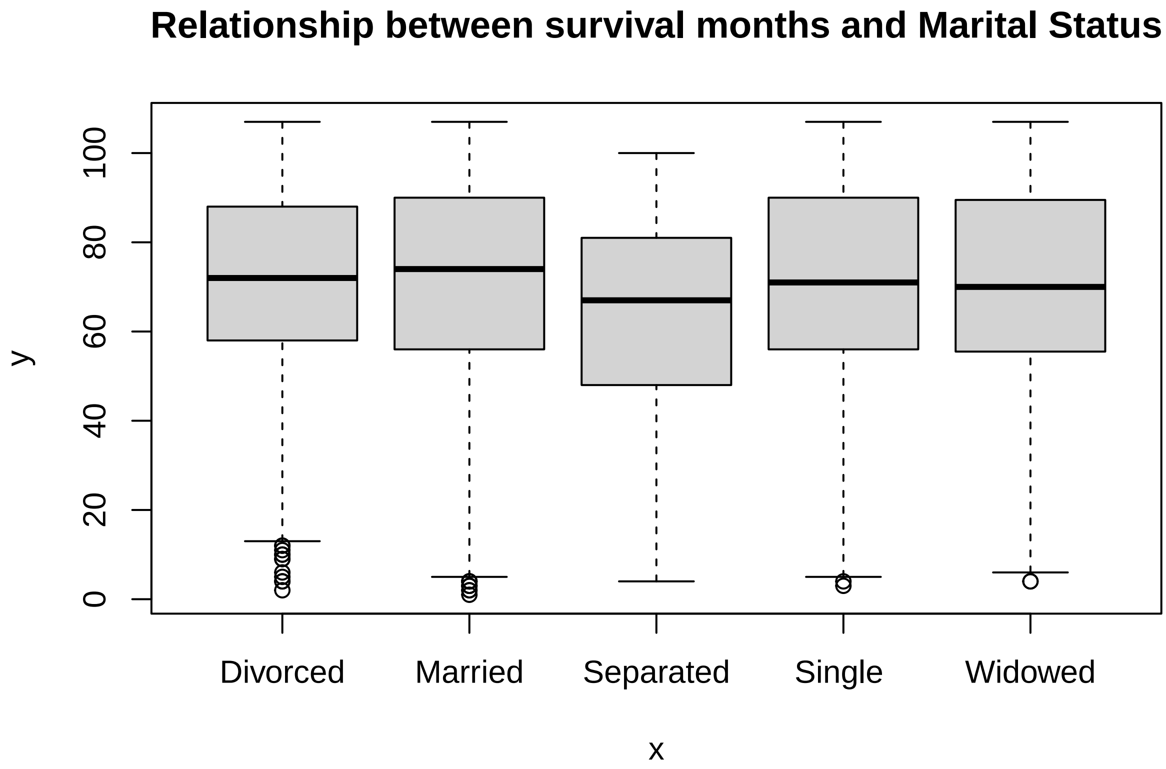
```r
table(df2$T.Stage, df2$A.Stage) #T-Stage and metastasis
```

```
##
##          0    1
##   T1  1594    9
##   T2  1756   30
##   T3   518   15
##   T4    64   38
```

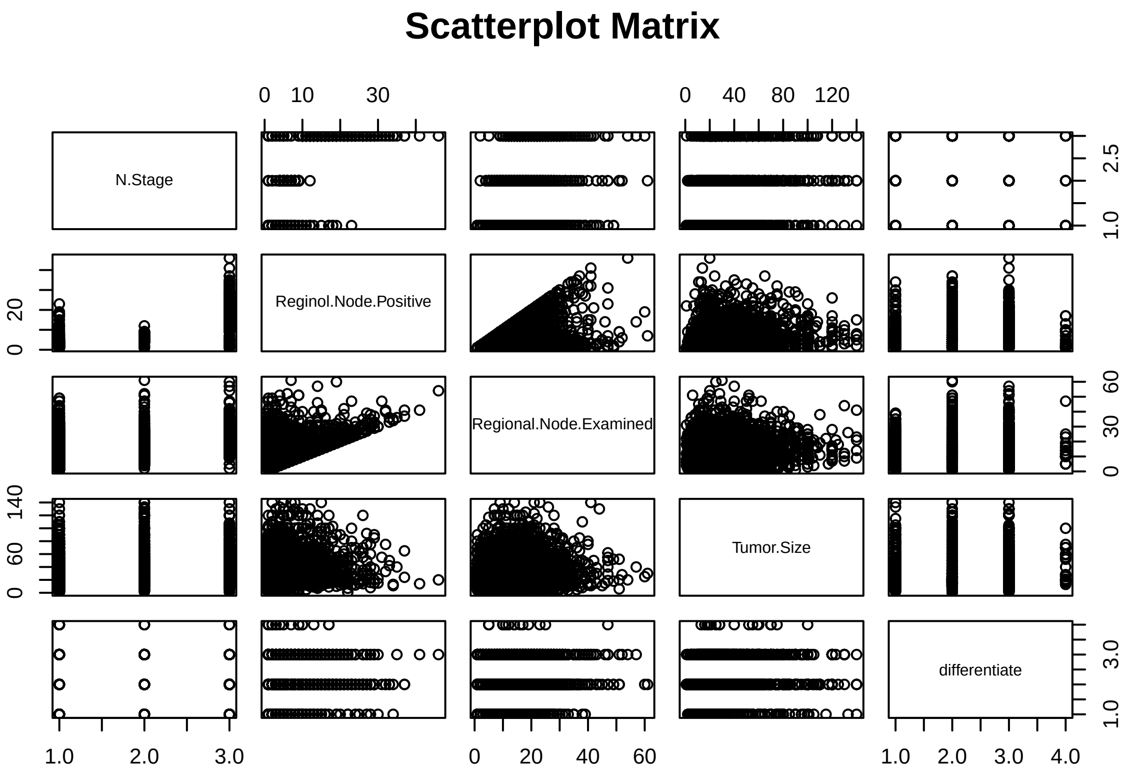
```r
names(df2)
```

```
##  [1] "Age"                    "Race"                   "Marital.Status"
##  [4] "T.Stage"                "N.Stage"                "X6th.Stage"
##  [7] "differentiate"          "A.Stage"                "Tumor.Size"
## [10] "Estrogen.Status"        "Progesterone.Status"    "Regional.Node.Examined"
## [13] "Reginol.Node.Positive"  "Survival.Months"        "Status"
```

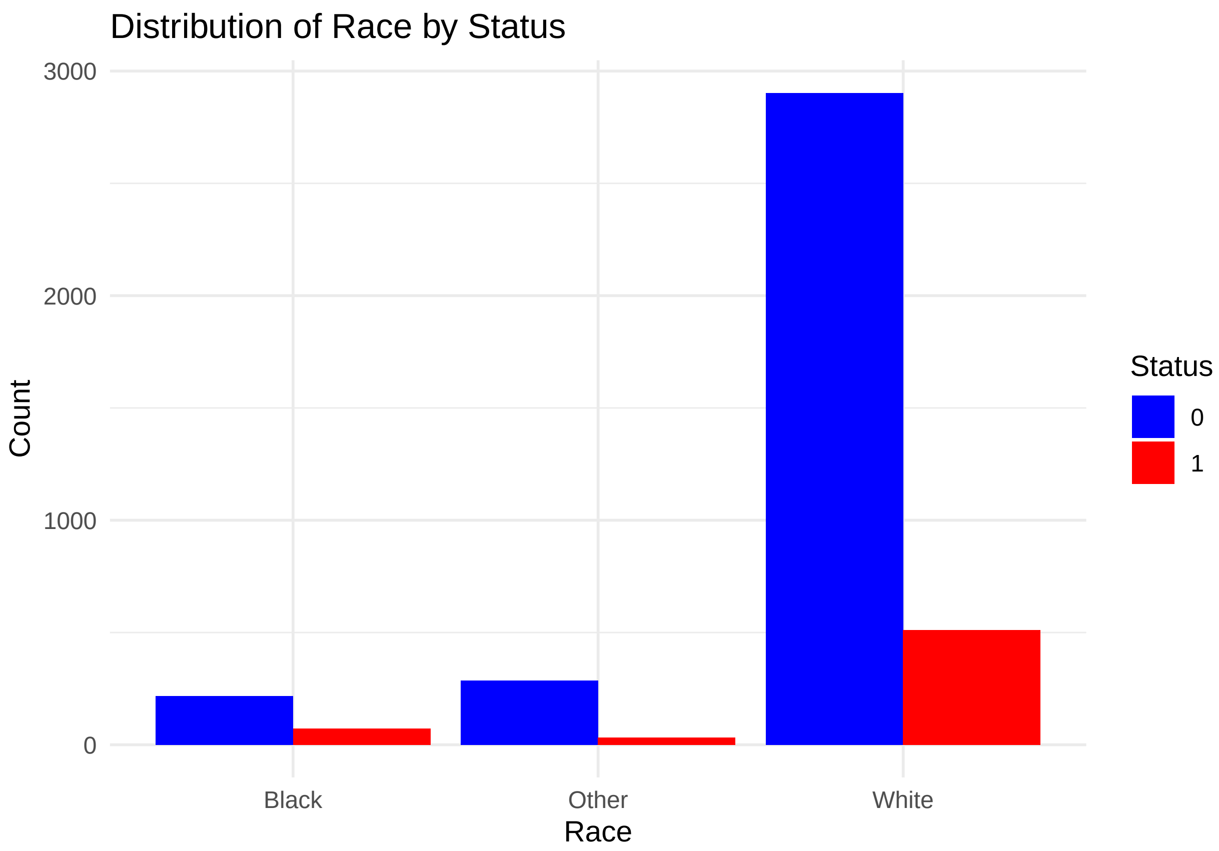
```r
plot(df2$Marital.Status,df2$Survival.Months,main="Relationship between survival months and Marital Statu
```



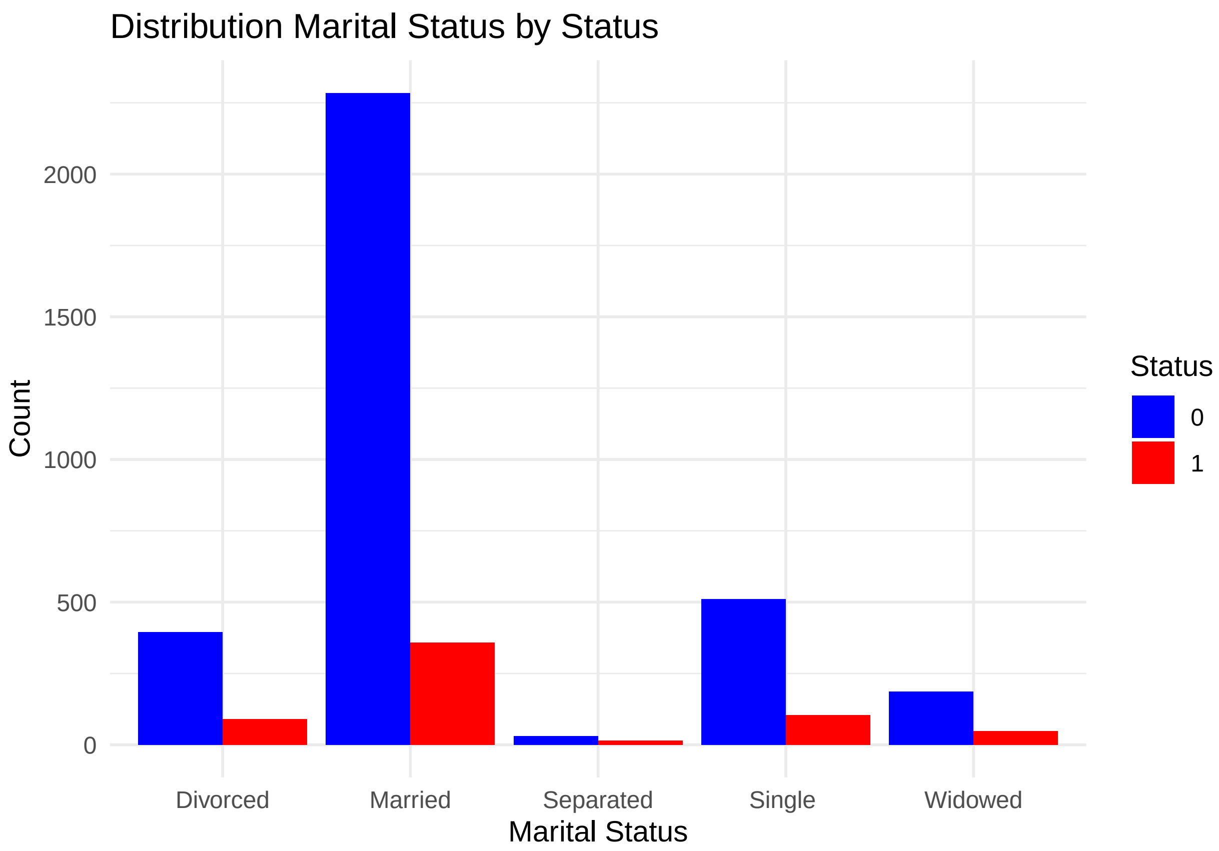
```
pairs(~N.Stage+Reginol.Node.Positive+Regional.Node.Examined+Tumor.Size+differentiate,data=df2,main="Sca
```



```r
library(ggplot2)

# Create a count plot
ggplot(df2, aes(x = Race, fill = factor(Status))) +
  geom_bar(position = "dodge", stat = "count") +
  labs(title = "Distribution of Race by Status",
       x = "Race",
       y = "Count",
       fill = "Status") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red")) +
  theme_minimal()
```



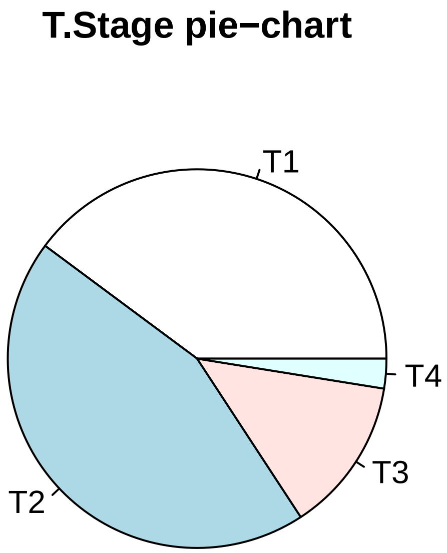
```r
ggplot(df2, aes(x = Marital.Status, fill = factor(Status))) +
  geom_bar(position = "dodge", stat = "count") +
  labs(title = "Distribution Marital Status by Status",
       x = "Marital Status",
       y = "Count",
       fill = "Status") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red")) +
  theme_minimal()
```



```r
table(df2$T.Stage)
```

```
##
##   T1   T2   T3   T4
## 1603 1786  533  102
```

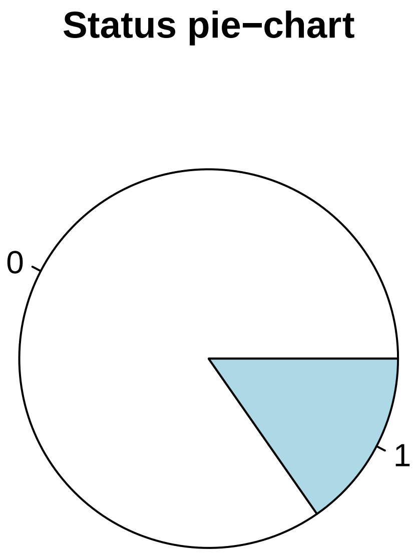
```r
pie(table(df2$T.Stage),main ="T.Stage pie-chart")
```



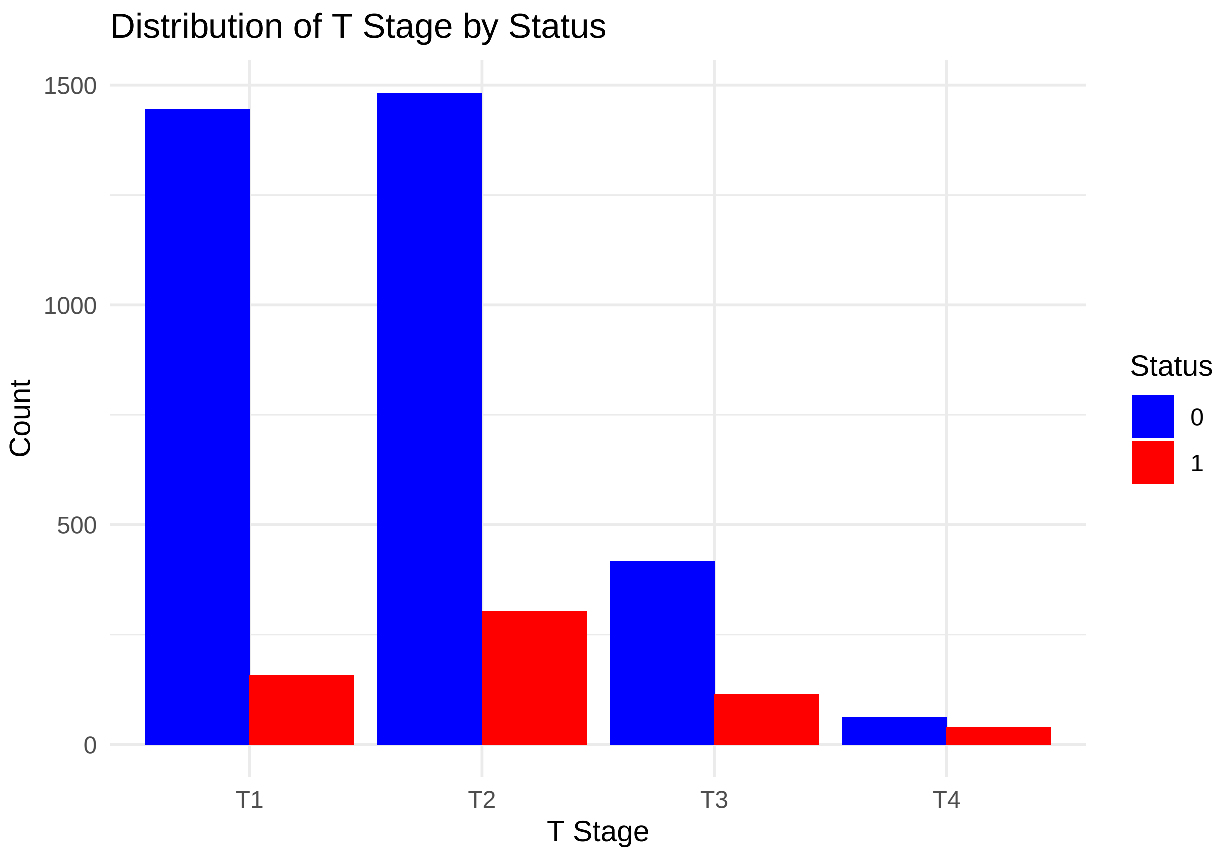
```
table(df2$Status)
```

```
##
##    0    1
## 3408  616
```

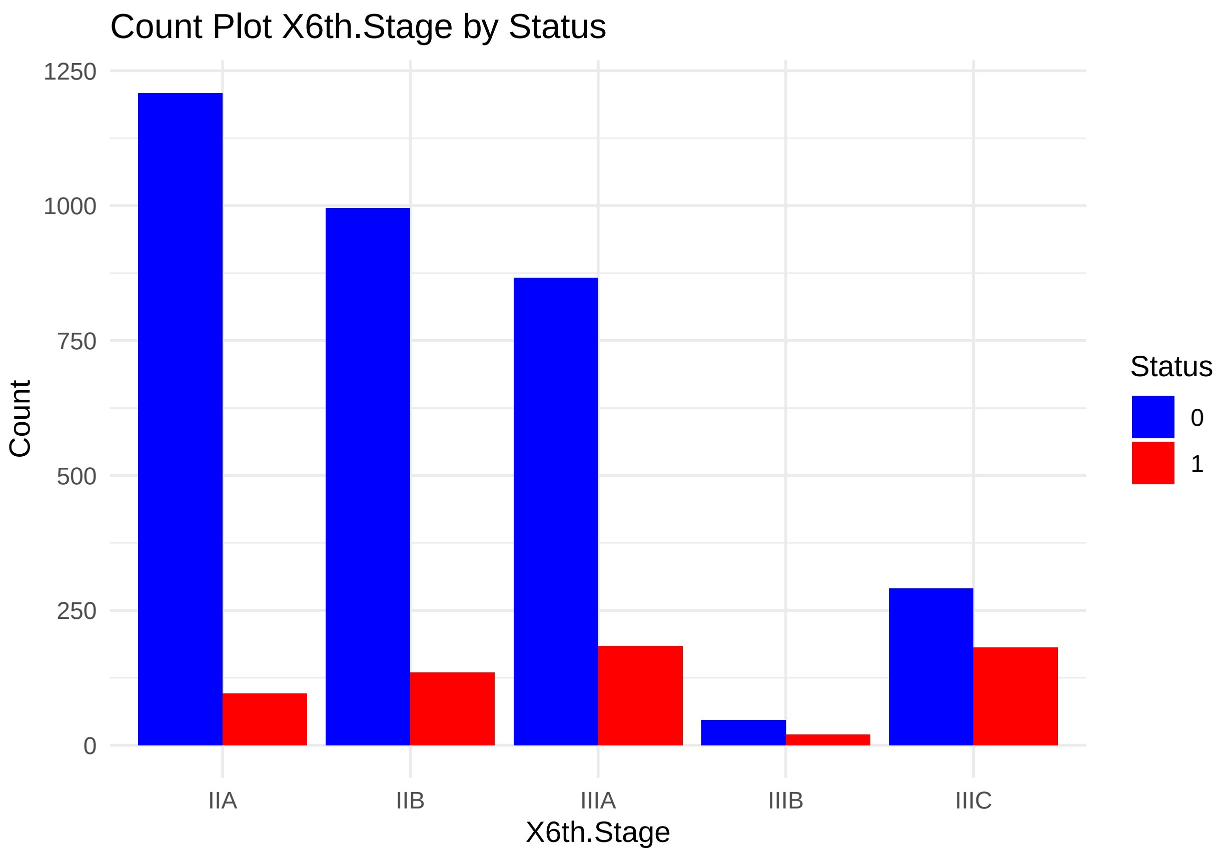
```
pie(table(df2$Status), main = "Status pie-chart")
```



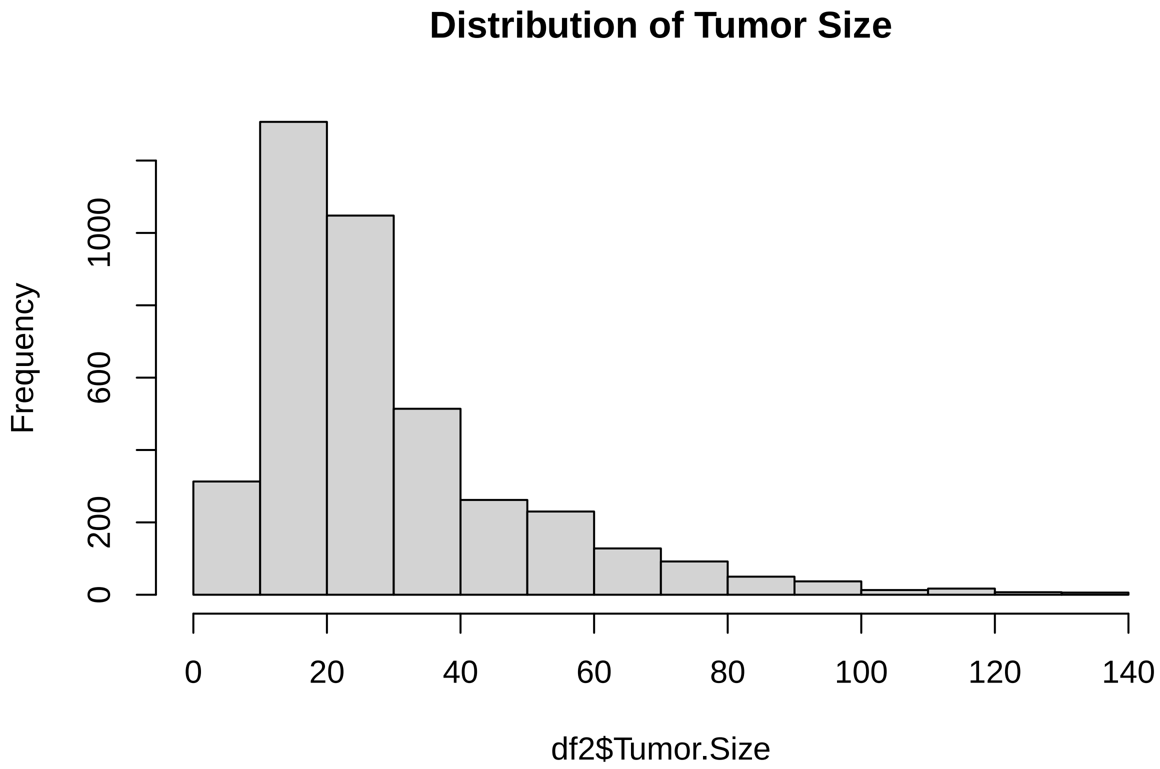
```r
ggplot(df2, aes(x = T.Stage, fill = factor(Status))) +
  geom_bar(position = "dodge", stat = "count") +
  labs(title = "Distribution of T Stage by Status",
       x = "T Stage",
       y = "Count",
       fill = "Status") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red")) +
  theme_minimal()
```



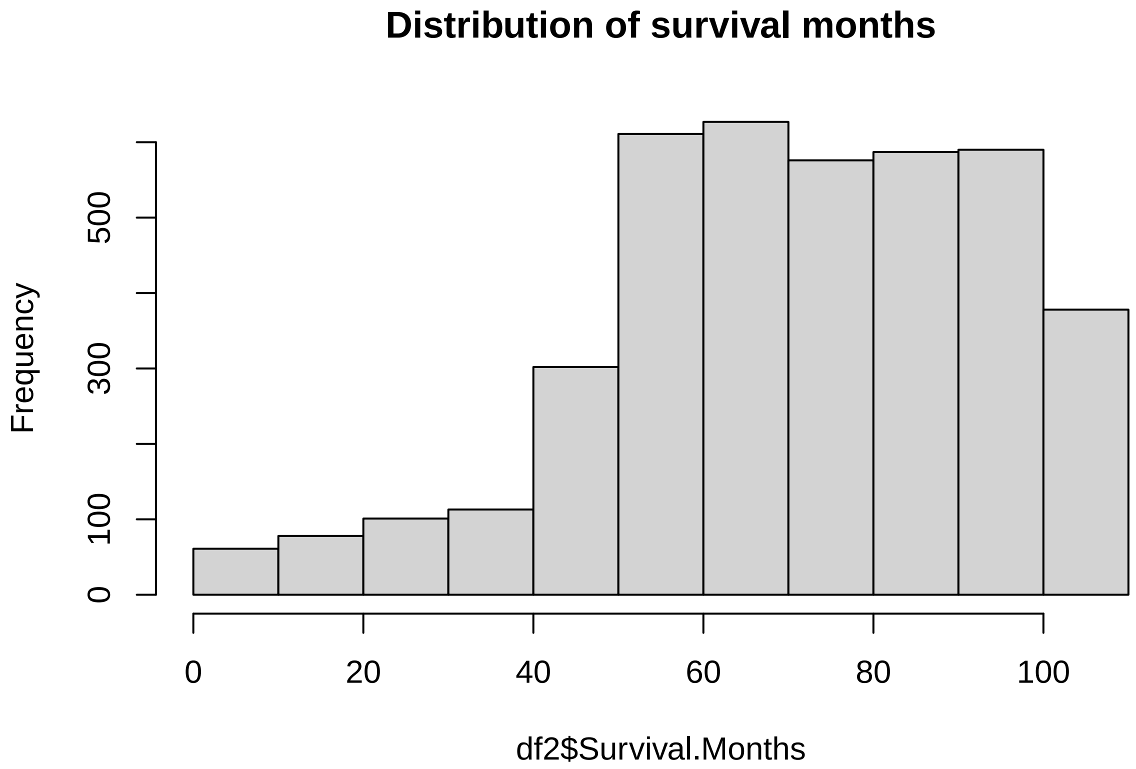
```r
ggplot(df2, aes(x = X6th.Stage, fill = factor(Status))) +
  geom_bar(position = "dodge", stat = "count") +
  labs(title = "Count Plot X6th.Stage by Status",
       x = "X6th.Stage",
       y = "Count",
       fill = "Status") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red")) +
  theme_minimal()
```



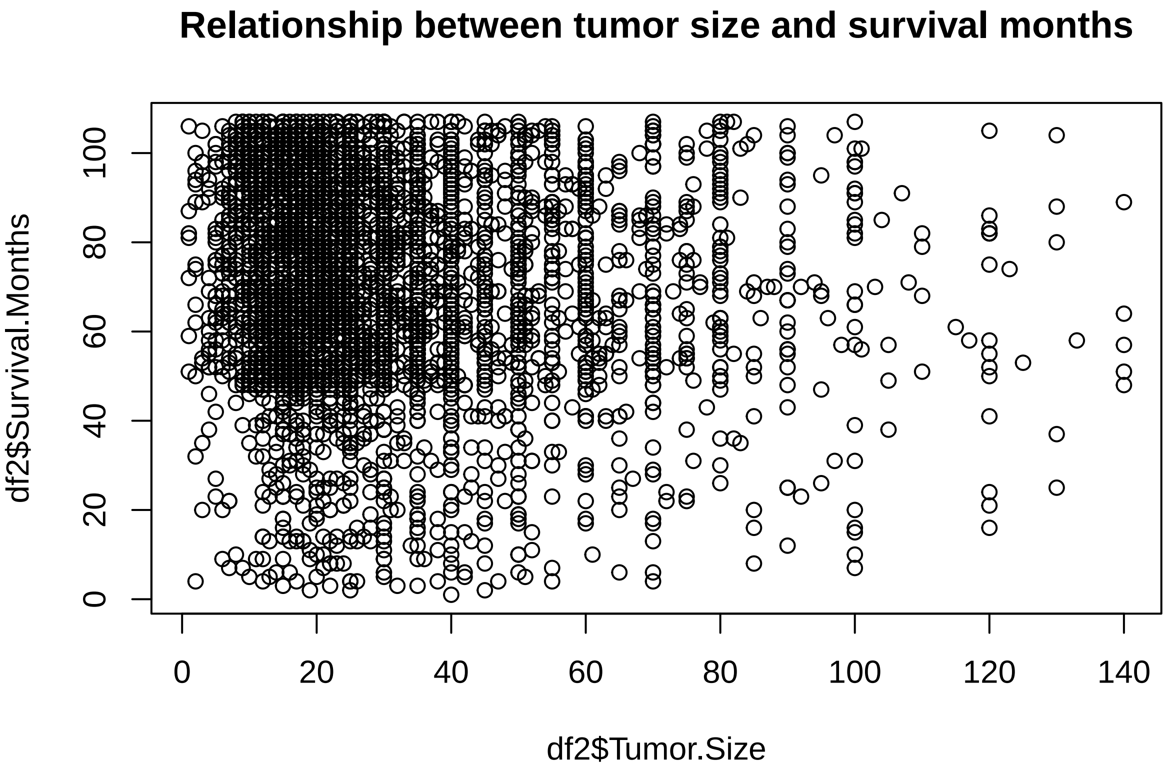
```r
hist(df2$Tumor.Size,main="Distribution of Tumor Size")
```



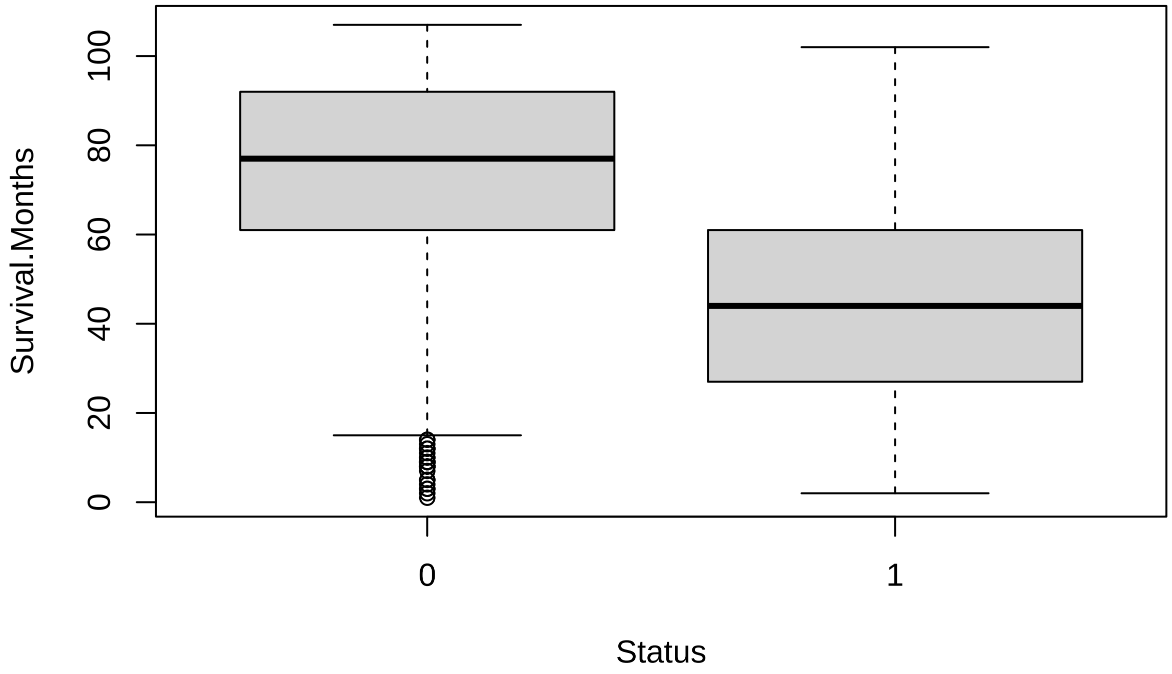
```r
hist(df2$Survival.Months, main = "Distribution of survival months")
```



```
plot(df2$Tumor.Size,df2$Survival.Months,main = "Relationship between tumor size and survival months")
```



```r
boxplot(Survival.Months~Status,data=df2)
```



Logistic Regresion

```
library(caret)

#Status as factor
df2$Status <- as.factor(df2$Status)

#Training and Testing
n=nrow(df2)
sample <- sample(c(TRUE, FALSE), n, replace=TRUE, prob=c(0.7,0.3))
train1  <- df2[sample, ]
test1   <- df2[!sample, ]

#FIRST MODEL: Logistic Regression with all the variables
model_logreg0 <- glm(Status ~ ., data = train1, family = binomial() )
summary(model_logreg0)
```

```
##
## Call:
## glm(formula = Status ~ ., family = binomial(), data = train1)
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               2.015238   0.802021    2.513  0.01198 *
## Age                       0.023184   0.007936    2.921  0.00349 **
## RaceOther                -0.469229   0.342022   -1.372  0.17009
## RaceWhite                -0.283744   0.232496   -1.220  0.22231
## Marital.StatusMarried    -0.064611   0.202296   -0.319  0.74943
## Marital.StatusSeparated   0.347707   0.715033    0.486  0.62677
## Marital.StatusSingle      0.107230   0.249680    0.429  0.66758
## Marital.StatusWidowed     0.098317   0.327243    0.300  0.76384
## T.Stage.L                 0.630579   0.471605    1.337  0.18119
## T.Stage.Q                 0.102867   0.336760    0.305  0.76002
## T.Stage.C                 0.097396   0.231946    0.420  0.67455
## N.Stage.L                -2.114059   2.270952   -0.931  0.35190
## N.Stage.Q                -1.767082   1.424108   -1.241  0.21467
## X6th.Stage.L              2.660123   2.234579    1.190  0.23388
## X6th.Stage.Q              1.841426   1.643749    1.120  0.26260
## X6th.Stage.C              0.994179   0.781566    1.272  0.20336
## X6th.Stage^4                    NA         NA       NA       NA
## differentiate.L           1.213666   0.665130    1.825  0.06805 .
## differentiate.Q           0.106422   0.497254    0.214  0.83053
## differentiate.C           0.125054   0.239968    0.521  0.60228
## A.Stage                  -0.385392   0.403262   -0.956  0.33923
## Tumor.Size                0.002051   0.005741    0.357  0.72094
## Estrogen.Status          -0.189163   0.282802   -0.669  0.50357
## Progesterone.Status      -0.513837   0.182988   -2.808  0.00498 **
## Regional.Node.Examined   -0.031757   0.009922   -3.201  0.00137 **
## Reginol.Node.Positive     0.061461   0.021662    2.837  0.00455 **
## Survival.Months          -0.063545   0.003439  -18.476  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2359.6  on 2808  degrees of freedom
## Residual deviance: 1528.5  on 2783  degrees of freedom
## AIC: 1580.5
##
## Number of Fisher Scoring iterations: 6
```

```
#First Model's Accuracy : 0.8897 ,  AIC: 1580.5, Specificity : 0.4573
predicted_probs0 <- predict(model_logreg0, newdata=test1, type="response")
predicted_class0 <- ifelse(predicted_probs0 > 0.5, 1, 0)
confusionMatrix(as.factor(predicted_class0), test1$Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 990 108
##          1  26  91
##
##                Accuracy : 0.8897
##                  95% CI : (0.8707, 0.9068)
##     No Information Rate : 0.8362
##     P-Value [Acc > NIR] : 7.724e-08
```

```
##
##                    Kappa : 0.5174
##
##  Mcnemar's Test P-Value : 2.609e-12
##
##             Sensitivity : 0.9744
##             Specificity : 0.4573
##          Pos Pred Value : 0.9016
##          Neg Pred Value : 0.7778
##              Prevalence : 0.8362
##          Detection Rate : 0.8148
##    Detection Prevalence : 0.9037
##       Balanced Accuracy : 0.7158
##
##        'Positive' Class : 0
##
```

```
#I am trying to predict if the person survives or not, then I have to eliminate survival months variabl
df3<-subset(df2,select=-Survival.Months)
## Training and Testing, no survival months
train  <- df3[sample, ]
test   <- df3[!sample, ]

#SECOND MODEL: Logistic Regression with all the variables, except survival months.
model_logreg1 <- glm(Status ~ ., data = train, family = binomial() )
summary(model_logreg1)
```

```
##
## Call:
## glm(formula = Status ~ ., family = binomial(), data = train)
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.499851   0.647369  -2.317 0.020512 *
## Age                      0.022615   0.006810   3.321 0.000897 ***
## RaceOther               -0.586594   0.295176  -1.987 0.046893 *
## RaceWhite               -0.306981   0.200408  -1.532 0.125578
## Marital.StatusMarried   -0.105791   0.171942  -0.615 0.538375
## Marital.StatusSeparated  0.827811   0.536733   1.542 0.122997
## Marital.StatusSingle    -0.019700   0.213014  -0.092 0.926315
## Marital.StatusWidowed   -0.046082   0.276335  -0.167 0.867559
## T.Stage.L                0.344176   0.382337   0.900 0.368020
## T.Stage.Q                0.036933   0.275038   0.134 0.893179
## T.Stage.C                0.159671   0.191798   0.832 0.405131
## N.Stage.L               -1.179964   1.825406  -0.646 0.518012
## N.Stage.Q               -1.155827   1.151495  -1.004 0.315494
## X6th.Stage.L             1.782936   1.793471   0.994 0.320162
## X6th.Stage.Q             1.087144   1.329216   0.818 0.413424
## X6th.Stage.C             0.556487   0.646879   0.860 0.389643
## X6th.Stage^4                   NA         NA      NA       NA
## differentiate.L          1.056840   0.491001   2.152 0.031364 *
## differentiate.Q         -0.026280   0.369567  -0.071 0.943309
## differentiate.C          0.030432   0.182182   0.167 0.867336
## A.Stage                 -0.015307   0.320528  -0.048 0.961912
## Tumor.Size               0.005335   0.004903   1.088 0.276557
## Estrogen.Status         -0.635529   0.218169  -2.913 0.003580 **
## Progesterone.Status     -0.533351   0.153754  -3.469 0.000523 ***
## Regional.Node.Examined  -0.036603   0.008929  -4.099 4.14e-05 ***
## Reginol.Node.Positive    0.067917   0.018480   3.675 0.000238 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2359.6  on 2808  degrees of freedom
## Residual deviance: 2036.3  on 2784  degrees of freedom
## AIC: 2086.3
##
## Number of Fisher Scoring iterations: 5
```

```
# Second Model's Accuracy: 0.851, AIC: 2086.3, Specificity : 0.1508
predicted_probs1 <- predict(model_logreg1, newdata=test, type="response")
predicted_class1 <- ifelse(predicted_probs1 > 0.5, 1, 0)
confusionMatrix(as.factor(predicted_class1), test$Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1004  169
##          1   12   30
##
##                Accuracy : 0.851
##                  95% CI : (0.8297, 0.8706)
##     No Information Rate : 0.8362
##     P-Value [Acc > NIR] : 0.08623
##
##                   Kappa : 0.2035
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.9882
##             Specificity : 0.1508
##          Pos Pred Value : 0.8559
##          Neg Pred Value : 0.7143
##              Prevalence : 0.8362
##          Detection Rate : 0.8263
##    Detection Prevalence : 0.9654
##       Balanced Accuracy : 0.5695
##
##        'Positive' Class : 0
##
```

```
#THIRD MODEL: Logistic Regression with selected variables
model_logreg2 <- glm(Status ~ X6th.Stage+Progesterone.Status+Estrogen.Status+Reginol.Node.Positive+Age+
summary(model_logreg2)
```

```
##
```

```
## Call:
## glm(formula = Status ~ X6th.Stage + Progesterone.Status + Estrogen.Status +
##     Reginol.Node.Positive + Age + Race, family = binomial(),
##     data = train)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.209697   0.414003  -2.922 0.003478 **
## X6th.Stage.L           1.250783   0.220830   5.664 1.48e-08 ***
## X6th.Stage.Q          -0.390576   0.163812  -2.384 0.017112 *
## X6th.Stage.C          -0.250954   0.230241  -1.090 0.275730
## X6th.Stage^4          -0.248645   0.185498  -1.340 0.180109
## Progesterone.Status   -0.616012   0.151040  -4.078 4.53e-05 ***
## Estrogen.Status       -0.754185   0.209848  -3.594 0.000326 ***
## Reginol.Node.Positive  0.045416   0.016011   2.837 0.004560 **
## Age                    0.018586   0.006516   2.853 0.004337 **
## RaceOther             -0.719717   0.286439  -2.513 0.011983 *
## RaceWhite             -0.422739   0.191585  -2.207 0.027347 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2359.6  on 2808  degrees of freedom
## Residual deviance: 2093.3  on 2798  degrees of freedom
## AIC: 2115.3
##
## Number of Fisher Scoring iterations: 5
```

```
#Third Model's Accuracy : 0.8477,AIC: 2115.3,  Specificity : 0.09548
predicted_probs2 <- predict(model_logreg2, newdata=test, type="response")
predicted_class2 <- ifelse(predicted_probs2 > 0.5, 1, 0)
confusionMatrix(as.factor(predicted_class2), test$Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1011  180
##          1    5   19
##
##                Accuracy : 0.8477
##                  95% CI : (0.8263, 0.8675)
##     No Information Rate : 0.8362
##     P-Value [Acc > NIR] : 0.1474
##
##                   Kappa : 0.1401
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.99508
##             Specificity : 0.09548
##          Pos Pred Value : 0.84887
##          Neg Pred Value : 0.79167
##               Prevalence : 0.83621
##           Detection Rate : 0.83210
##     Detection Prevalence : 0.98025
##        Balanced Accuracy : 0.54528
##
##         'Positive' Class : 0
##
```

Conclusions:

1. First Model: Logistic Regression was performed with all the variables and is the model with better results in accuracy (88%) and predicting death events (45%), but this model can't be used to reliably predict Y, because not possible to know the survived months variable, which is the most significant in this model.
2. Second Model: This model is 85% accurate, but this model predicts only 15% of death events.
3. Third Model: Considering only significant variables, this model is 84.7% accurate, similar to the second model but is not accurate predicting death events (only predicts 9%).

```
##Over Sampling death cases
library(ROSE)
```

```
## Warning: package 'ROSE' was built under R version 4.3.2
```

```
## Loaded ROSE 0.0-4
```

```
over <- ovun.sample(Status~.,data=train,method="over",p=0.3)
df.balanced <-over$data
head(df.balanced)
```

```
##   Age  Race Marital.Status T.Stage N.Stage X6th.Stage             differentiate
## 1  68 White        Married      T1      N1        IIA     Poorly differentiated
## 2  58 White       Divorced      T3      N3       IIIC Moderately differentiated
## 3  51 White         Single      T1      N1        IIA Moderately differentiated
## 4  51 White        Married      T1      N1        IIA       Well differentiated
## 5  40 White       Divorced      T4      N3       IIIC     Poorly differentiated
## 6  69 White        Married      T4      N3       IIIC       Well differentiated
##   A.Stage Tumor.Size Estrogen.Status Progesterone.Status Regional.Node.Examined
## 1       0          4               1                   1                     24
## 2       0         63               1                   1                     14
## 3       0         20               1                   1                     18
## 4       0          8               1                   1                     11
## 5       0        103               1                   1                     20
## 6       1         32               1                   1                     21
##   Reginol.Node.Positive Status
## 1                     1      0
## 2                     7      0
## 3                     2      0
## 4                     1      0
## 5                    18      0
## 6                    12      0
```

```r
table(df.balanced$Status)
```

```
## 
##    0    1 
## 2392  996
```

```r
##FOURTH MODEL: Logistic Regression with 30% over sampling
model_logreg3 <- glm(Status ~ ., data = df.balanced, family = binomial() )
summary(model_logreg3)
```

```
## 
## Call:
## glm(formula = Status ~ ., family = binomial(), data = df.balanced)
## 
## Coefficients: (1 not defined because of singularities)
##                           Estimate Std. Error z value Pr(>|z|)    
## (Intercept)              -0.938901   0.498389  -1.884  0.05958 .  
## Age                       0.030558   0.005029   6.077 1.23e-09 ***
## RaceOther                -0.478974   0.213820  -2.240  0.02509 *  
## RaceWhite                -0.390346   0.153137  -2.549  0.01080 *  
## Marital.StatusMarried    -0.110870   0.126427  -0.877  0.38051    
## Marital.StatusSeparated   0.827443   0.404542   2.045  0.04082 *  
## Marital.StatusSingle     -0.099041   0.159645  -0.620  0.53500    
## Marital.StatusWidowed    -0.104728   0.207181  -0.505  0.61322    
## T.Stage.L                 0.083661   0.318483   0.263  0.79279    
## T.Stage.Q                -0.009369   0.228720  -0.041  0.96732    
## T.Stage.C                 0.176959   0.152593   1.160  0.24618    
## N.Stage.L                -0.594604   1.501083  -0.396  0.69202    
## N.Stage.Q                -0.635066   0.932397  -0.681  0.49580    
## X6th.Stage.L              1.374412   1.482257   0.927  0.35380    
## X6th.Stage.Q              0.361895   1.075328   0.337  0.73646    
## X6th.Stage.C              0.153387   0.498266   0.308  0.75820    
## X6th.Stage^4                    NA         NA      NA       NA    
## differentiate.L           1.193321   0.392493   3.040  0.00236 ** 
## differentiate.Q           0.079668   0.295775   0.269  0.78766    
## differentiate.C           0.122114   0.144347   0.846  0.39756    
## A.Stage                   0.131266   0.248987   0.527  0.59806    
## Tumor.Size                0.006514   0.003774   1.726  0.08438 .  
## Estrogen.Status          -0.560963   0.172965  -3.243  0.00118 ** 
## Progesterone.Status      -0.569418   0.114381  -4.978 6.42e-07 ***
## Regional.Node.Examined   -0.044361   0.006592  -6.730 1.70e-11 ***
## Reginol.Node.Positive     0.075980   0.014319   5.306 1.12e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 4104.1  on 3387  degrees of freedom
## Residual deviance: 3468.8  on 3363  degrees of freedom
## AIC: 3518.8
## 
## Number of Fisher Scoring iterations: 4
```

```
#Accuracy, AIC: 3518.8,Accuracy : 0.828, Specificity : 0.3719
predicted_probs3 <- predict(model_logreg3, newdata=test, type="response")
predicted_class3 <- ifelse(predicted_probs3 > 0.5, 1, 0)
confusionMatrix(as.factor(predicted_class3), test$Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 932 125
##          1  84  74
##
##                Accuracy : 0.828
##                  95% CI : (0.8056, 0.8488)
##     No Information Rate : 0.8362
##     P-Value [Acc > NIR] : 0.79301
##
##                   Kappa : 0.3153
##
##  Mcnemar's Test P-Value : 0.00566
##
##             Sensitivity : 0.9173
##             Specificity : 0.3719
##          Pos Pred Value : 0.8817
##          Neg Pred Value : 0.4684
##              Prevalence : 0.8362
##          Detection Rate : 0.7671
##    Detection Prevalence : 0.8700
##       Balanced Accuracy : 0.6446
##
##        'Positive' Class : 0
##
```

4. This is an unbalanced dataset, the fourth model was trained with 30% over sampling. It is 82% accurate and predicts 37% of death events. In this model we observe a trade off between accuracy and specificity.

AUC-ROC

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

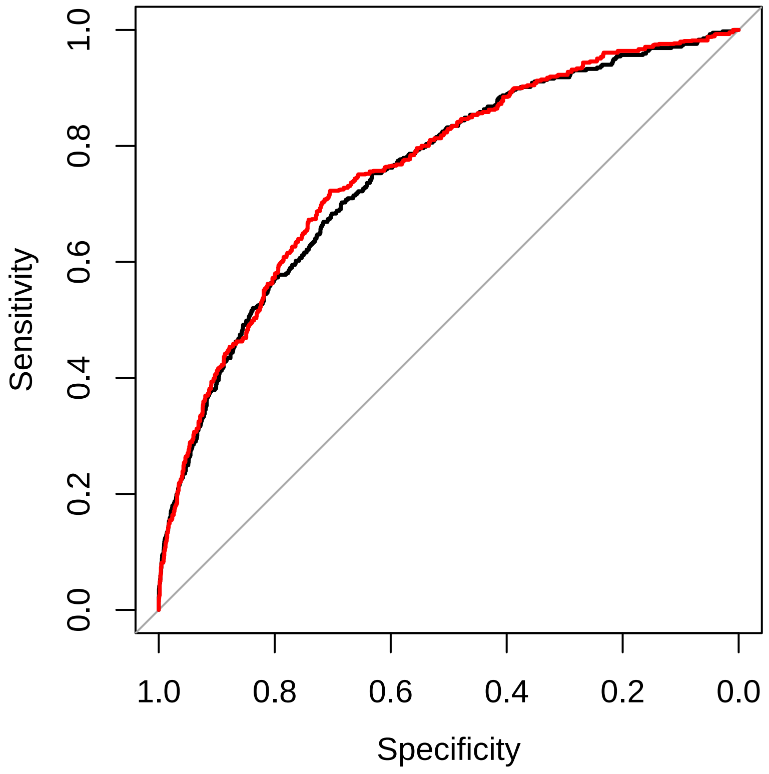
```r
State <- train[,14]
#1.Second model
predictions1=predict(model_logreg1)
roc1=roc(State ~ predictions1)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
#2.fourth model
State2 <- df.balanced[,14]
predictions2=predict(model_logreg3)
roc2=roc(State2 ~ predictions2)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
##Graph
par(pty="s")
plot(roc1)
plot(roc2, add=TRUE, col='red')
```



```r
names(roc2)
```

```
##  [1] "percent"            "sensitivities"      "specificities"     
##  [4] "thresholds"         "direction"          "cases"             
##  [7] "controls"           "fun.sesp"           "auc"               
## [10] "call"               "original.predictor" "original.response" 
## [13] "predictor"          "response"           "levels"            
## [16] "predictor.name"     "response.name"
```

```r
##Area under the curve
AUC_1=roc1$auc
AUC_2=roc2$auc
print(paste("The AUC for the model withoul survived months", AUC_1))
```

```
## [1] "The AUC for the model withoul survived months 0.75296802691626"
```

```r
print(paste("The AUC for the model with oversampling",AUC_2))
```

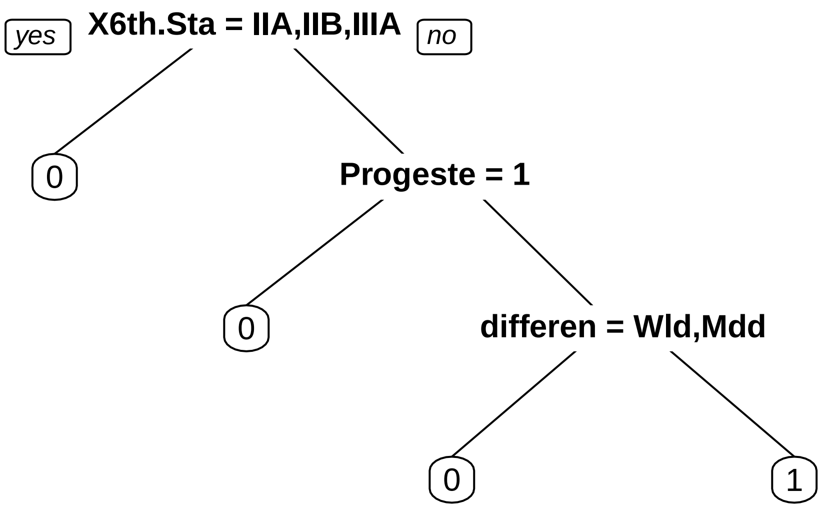
```
## [1] "The AUC for the model with oversampling 0.759951175941223"
```

Tree

```r
library(rpart)
library(rpart.plot)

## FIFTH MODEL
dt = rpart(formula = Status ~ ., data = train , method = "class")
summary(dt)
```

```
## Call:
## rpart(formula = Status ~ ., data = train, method = "class")
##   n= 2809 
## 
##           CP nsplit rel error   xerror       xstd
## 1 0.01598721      0 1.0000000 1.000000 0.04518941
## 2 0.01000000      3 0.9520384 1.014388 0.04545624
## 
## Variable importance
##            X6th.Stage               N.Stage Reginol.Node.Positive 
##                    28                    25                    23 
##   Progesterone.Status               T.Stage               A.Stage 
##                     6                     5                     5 
##         differentiate       Estrogen.Status                   Age 
##                     3                     3                     1 
##            Tumor.Size 
##                     1 
## 
## Node number 1: 2809 observations,    complexity param=0.01598721
##   predicted class=0  expected loss=0.1484514  P(node) =1
##     class counts:  2392   417
##    probabilities: 0.852 0.148
##   left son=2 (2430 obs) right son=3 (379 obs)
##   Primary splits:
##       X6th.Stage            splits as  LLLRR,     improve=41.75754, (0 missing)
##       N.Stage               splits as  LRR,       improve=39.12411, (0 missing)
##       Reginol.Node.Positive < 3.5   to the left,  improve=35.56691, (0 missing)
##       Estrogen.Status       < 0.5   to the right, improve=19.99980, (0 missing)
##       Progesterone.Status   < 0.5   to the right, improve=19.65865, (0 missing)
##   Surrogate splits:
##       N.Stage               splits as  LLR,       agree=0.983, adj=0.876, (0 split)
##       Reginol.Node.Positive < 9.5   to the left,  agree=0.974, adj=0.805, (0 split)
##       T.Stage               splits as  LLLR,      agree=0.890, adj=0.187, (0 split)
##       A.Stage               < 0.5   to the left,  agree=0.887, adj=0.161, (0 split)
##       Tumor.Size            < 118.5 to the left,  agree=0.866, adj=0.008, (0 split)
##
## Node number 2: 2430 observations
##   predicted class=0  expected loss=0.1144033  P(node) =0.8650765
##     class counts:  2152   278
##    probabilities: 0.886 0.114
##
## Node number 3: 379 observations,    complexity param=0.01598721
##   predicted class=0  expected loss=0.3667546  P(node) =0.1349235
##     class counts:   240   139
##    probabilities: 0.633 0.367
##   left son=6 (278 obs) right son=7 (101 obs)
##   Primary splits:
##       Progesterone.Status   < 0.5   to the right, improve=8.705796, (0 missing)
##       Estrogen.Status       < 0.5   to the right, improve=8.069071, (0 missing)
##       differentiate         splits as  LLRR,      improve=6.647339, (0 missing)
##       Reginol.Node.Positive < 14.5  to the left,  improve=4.804297, (0 missing)
##       Tumor.Size            < 104   to the left,  improve=3.855766, (0 missing)
##   Surrogate splits:
##       Estrogen.Status < 0.5   to the right, agree=0.842, adj=0.406, (0 split)
##
## Node number 6: 278 observations
##   predicted class=0  expected loss=0.3021583  P(node) =0.0989676
##     class counts:   194    84
##    probabilities: 0.698 0.302
##
## Node number 7: 101 observations,    complexity param=0.01598721
##   predicted class=1  expected loss=0.4554455  P(node) =0.03595586
##     class counts:    46    55
##    probabilities: 0.455 0.545
##   left son=14 (37 obs) right son=15 (64 obs)
##   Primary splits:
##       differentiate          splits as  LLRR,      improve=4.359145, (0 missing)
##       Estrogen.Status        < 0.5   to the right, improve=3.266071, (0 missing)
##       Regional.Node.Examined < 28.5  to the right, improve=3.248505, (0 missing)
##       A.Stage                < 0.5   to the left,  improve=3.176186, (0 missing)
##       Age                    < 51.5  to the right, improve=2.525634, (0 missing)
##   Surrogate splits:
##       Age                   < 55.5  to the right, agree=0.723, adj=0.243, (0 split)
##       Estrogen.Status       < 0.5   to the right, agree=0.693, adj=0.162, (0 split)
##       Reginol.Node.Positive < 22.5  to the right, agree=0.683, adj=0.135, (0 split)
##       Tumor.Size            < 13.5  to the left,  agree=0.673, adj=0.108, (0 split)
##       Race                  splits as  LLR,       agree=0.663, adj=0.081, (0 split)
##
## Node number 14: 37 observations
##   predicted class=0  expected loss=0.3513514  P(node) =0.01317195
##     class counts:    24    13
##    probabilities: 0.649 0.351
##
## Node number 15: 64 observations
##   predicted class=1  expected loss=0.34375  P(node) =0.02278391
##     class counts:    22    42
##    probabilities: 0.344 0.656
```

```
prp(dt)
```



```
# Accuracy:0.8378601, Specificity: 0.07035176
predicted_probs4 <- predict(dt, newdata=test, type="prob")
head(predicted_probs4)
```

```
##           0         1
## 2  0.8855967 0.1144033
## 4  0.8855967 0.1144033
## 5  0.8855967 0.1144033
## 8  0.8855967 0.1144033
## 11 0.8855967 0.1144033
## 16 0.8855967 0.1144033
```

```
pred_labels4 <- ifelse(predicted_probs4[, 2] > 0.5, 1, 0)

# Evaluate the model on the test set
conf_matrix <- table(pred_labels4, test$Status)
conf_matrix
```

```
##
## pred_labels4    0    1
##            0 1004  185
##            1   12   14
```

```
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy
```

```
## [1] 0.8378601
```

```
specificity <- conf_matrix[2,2]/sum(as.numeric(test$Status)-1)
specificity
```

```
## [1] 0.07035176
```

Conclusion: The decision tree is 83.7% accurate and 7% specific, this model is not useful for predicting death events.

Random Forest

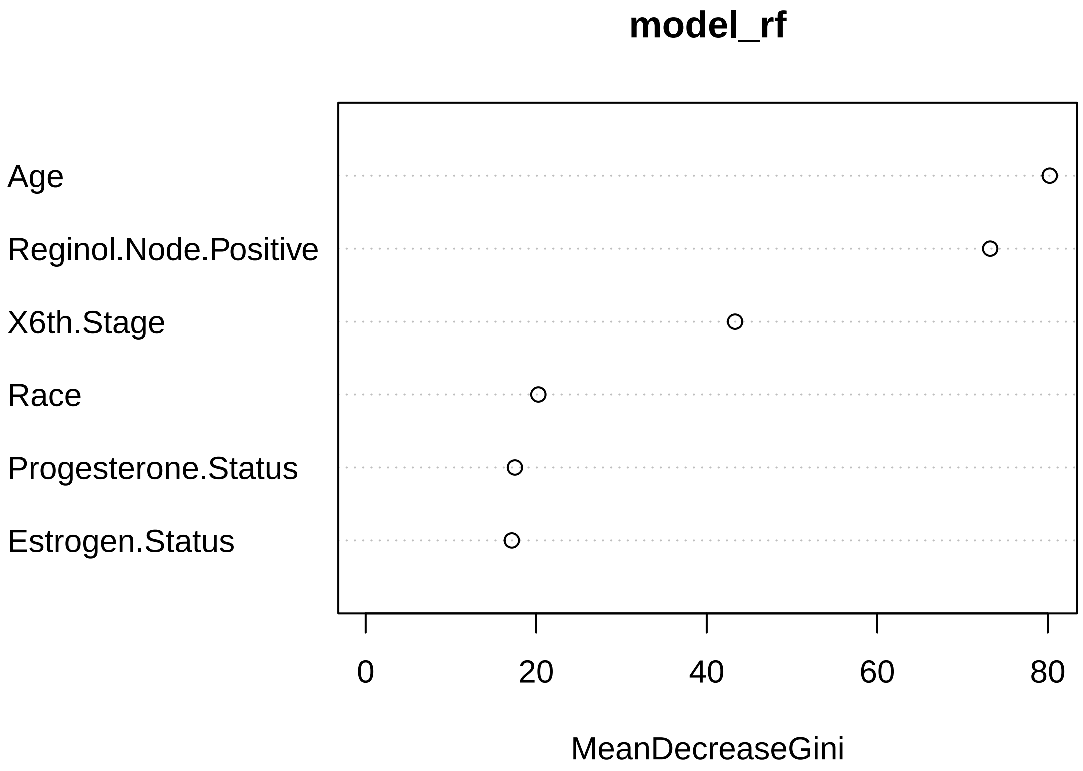
```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.2

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# SIXTH MODEL
model_rf = randomForest(Status ~  X6th.Stage+Progesterone.Status+Estrogen.Status+Reginol.Node.Positive+
summary(model_rf)
```

```
##                 Length Class  Mode
## call               4   -none- call
## type               1   -none- character
## predicted       2809   factor numeric
## err.rate         300   -none- numeric
## confusion          6   -none- numeric
## votes           5618   matrix numeric
## oob.times       2809   -none- numeric
## classes            2   -none- character
## importance         6   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y               2809   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
varImpPlot(model_rf)
```



```r
#accuracy: 0.8444444, specificity: 0.1005025
predicted6 <- predict(model_rf, newdata=test, type="response")
conf_matrix<-table(predicted6, test$Status)
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy
```

```
## [1] 0.8444444
```

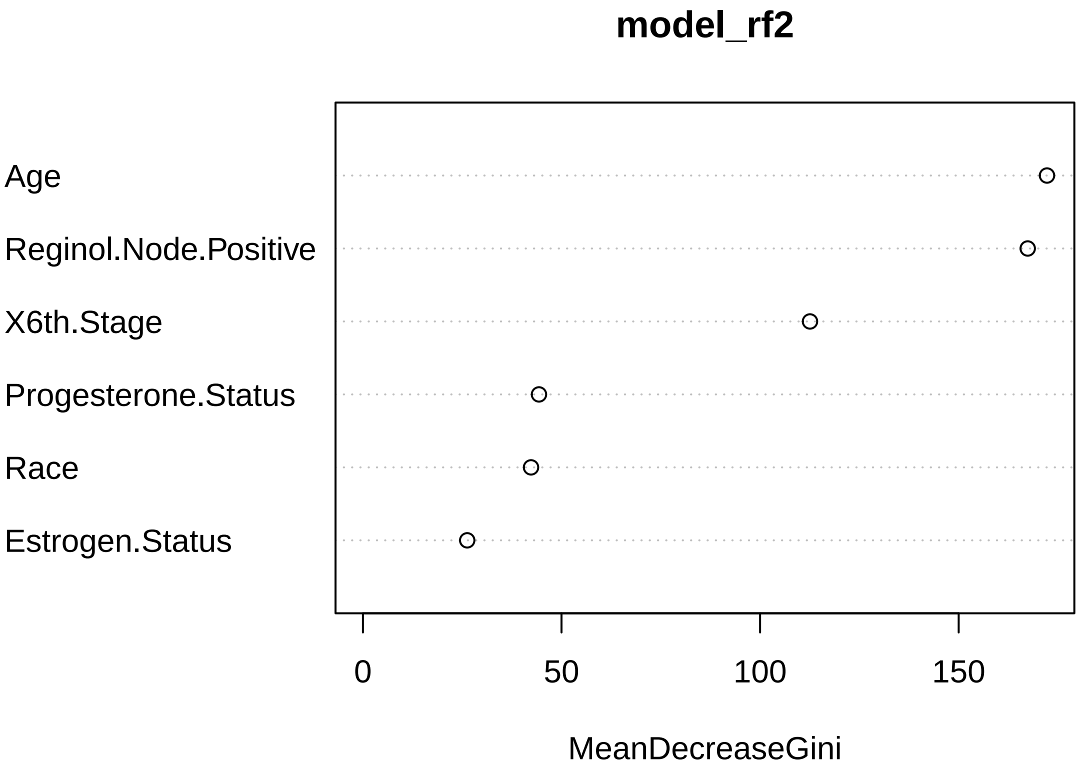
```r
specificity <- conf_matrix[2,2]/sum(as.numeric(test$Status)-1)
specificity
```

```
## [1] 0.1005025
```

```r
# SEVENTH MODEL
model_rf2 = randomForest(Status  ~  X6th.Stage+Progesterone.Status+Estrogen.Status+Reginol.Node.Positiv
summary(model_rf2)
```

```
##                 Length Class  Mode
## call               4   -none- call
## type               1   -none- character
## predicted       3388   factor numeric
## err.rate         300   -none- numeric
## confusion          6   -none- numeric
## votes           6776   matrix numeric
## oob.times       3388   -none- numeric
## classes            2   -none- character
## importance         6   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y               3388   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
varImpPlot(model_rf2)
```



```
#Accuracy: 0.8222222, specificity:  0.2763819
predicted7 <- predict(model_rf2, newdata=test, type="response")
conf_matrix<-table(predicted7, test$Status)
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy
```

```
## [1] 0.8222222
```

```r
specificity <- conf_matrix[2,2]/sum(as.numeric(test$Status)-1)
specificity
```

```
## [1] 0.2763819
```

XG Boost

```r
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.3.2
```

```
##
## Attaching package: 'xgboost'
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 4.3.2
```

```
## Thank you for using fastDummies!
```

```
## To acknowledge our work, please cite the package:
```

```
## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from
```

```r
library(DiagrammeR)
```

```
## Warning: package 'DiagrammeR' was built under R version 4.3.2
```

```r
# Calculate the scale_pos_weight value
Status <- as.numeric(df2$Status)-1
ndeath <- sum(Status)
nalive <- sum(Status==0)
scale_pos_weight <- nalive / ndeath
str(df2)
```

```
## 'data.frame':    4024 obs. of  15 variables:
##  $ Age                   : int  68 50 58 58 47 51 51 40 40 69 ...
##  $ Race                  : Factor w/ 3 levels "Black","Other",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Marital.Status        : Factor w/ 5 levels "Divorced","Married",..: 2 2 1 2 2 4 2 2 1 2 ...
##  $ T.Stage               : Ord.factor w/ 4 levels "T1"<"T2"<"T3"<..: 1 2 3 1 2 1 1 2 4 4 ...
##  $ N.Stage               : Ord.factor w/ 3 levels "N1"<"N2"<"N3": 1 2 3 1 1 1 1 1 3 3 ...
##  $ X6th.Stage            : Ord.factor w/ 5 levels "IIA"<"IIB"<"IIIA"<..: 1 3 5 1 2 1 1 2 5 5 ...
##  $ differentiate         : Ord.factor w/ 4 levels "Well differentiated"<..: 3 2 2 3 3 2 1 2 3 1 ...
##  $ A.Stage               : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Tumor.Size            : int  4 35 63 18 41 20 8 30 103 32 ...
##  $ Estrogen.Status       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Progesterone.Status   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Regional.Node.Examined: int  24 14 14 2 3 18 11 9 20 21 ...
##  $ Reginol.Node.Positive : int  1 5 7 1 1 2 1 1 18 12 ...
##  $ Survival.Months       : int  60 62 75 84 50 89 54 14 70 92 ...
##  $ Status                : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
```

```
##NUMERICAL DATA
numdata <- copy(df3)
numdata$Marital.Status <- as.numeric(numdata$Marital.Status)
unique(numdata$Race)
```

```
## [1] White Black Other
## Levels: Black Other White
```

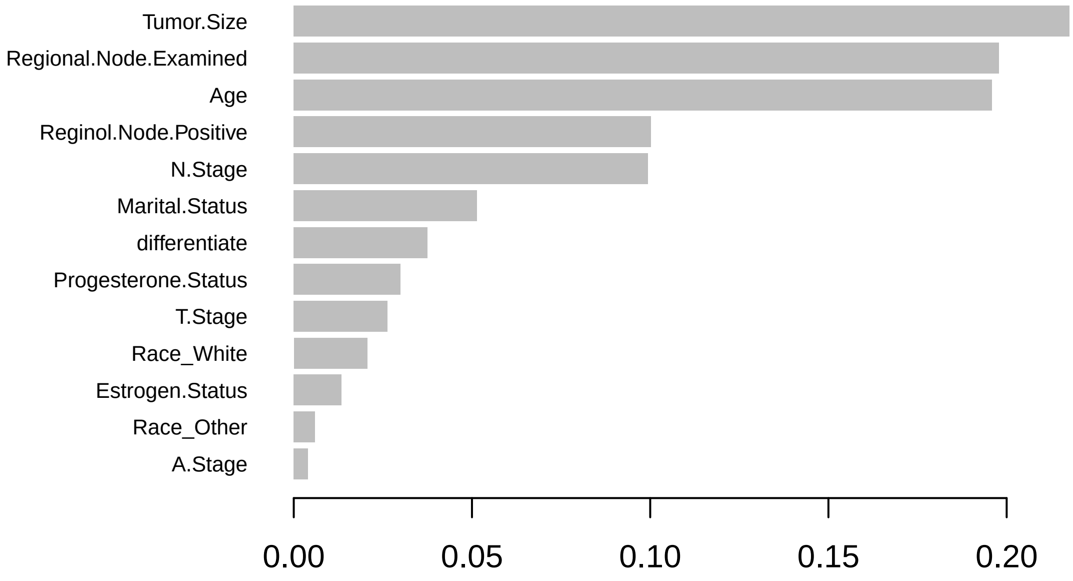
```
table(numdata$Race)
```

```
##
## Black Other White
##   291   320  3413
```

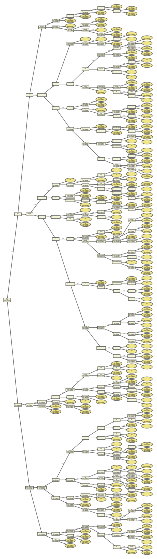
```
numdata$T.Stage <- as.numeric(numdata$T.Stage)
numdata$N.Stage <- as.numeric(numdata$N.Stage)
numdata$differentiate  <- as.numeric(numdata$differentiate)
numdata$Status <- as.numeric(numdata$Status)-1

##DUMMY DATA
dummydata <- dummy_cols(numdata, remove_first_dummy = TRUE)
dummydata$Race <- NULL
dummydata$X6th.Stage <- NULL

##Training and testing
traind  <- dummydata[sample, ]
testd  <- dummydata[!sample, ]

#Define XGBoost parameters
params <- list(
  objective = "binary:logistic",
  eval_metric = "logloss",
  scale_pos_weight = scale_pos_weight,
  max_depth = 10)

##EIGHTH MODEL - XGBOOST
model_xgb = xgboost(data = as.matrix(traind[, -which(names(traind) == "Status")]),
                     label = traind$Status,params=params,nthread = 2,nrounds=100)
```

```
## [1]  train-logloss:0.556657
## [2]  train-logloss:0.470337
## [3]  train-logloss:0.405138
## [4]  train-logloss:0.351633
## [5]  train-logloss:0.315596
## [6]  train-logloss:0.281047
## [7]  train-logloss:0.258978
## [8]  train-logloss:0.240224
## [9]  train-logloss:0.224973
## [10] train-logloss:0.214341
## [11] train-logloss:0.198095
## [12] train-logloss:0.185488
## [13] train-logloss:0.178579
## [14] train-logloss:0.170905
## [15] train-logloss:0.160531
## [16] train-logloss:0.151899
## [17] train-logloss:0.145143
## [18] train-logloss:0.142111
## [19] train-logloss:0.140453
## [20] train-logloss:0.135358
## [21] train-logloss:0.133982
## [22] train-logloss:0.129216
## [23] train-logloss:0.125820
## [24] train-logloss:0.116874
## [25] train-logloss:0.115249
## [26] train-logloss:0.112250
## [27] train-logloss:0.104343
## [28] train-logloss:0.101330
## [29] train-logloss:0.095843
## [30] train-logloss:0.092985
## [31] train-logloss:0.090218
## [32] train-logloss:0.085508
## [33] train-logloss:0.084467
## [34] train-logloss:0.082984
## [35] train-logloss:0.079314
## [36] train-logloss:0.076437
## [37] train-logloss:0.075327
## [38] train-logloss:0.073187
## [39] train-logloss:0.071783
## [40] train-logloss:0.069098
## [41] train-logloss:0.067090
## [42] train-logloss:0.063229
## [43] train-logloss:0.060799
## [44] train-logloss:0.058515
## [45] train-logloss:0.056634
## [46] train-logloss:0.055406
## [47] train-logloss:0.054513
## [48] train-logloss:0.053163
## [49] train-logloss:0.051653
## [50] train-logloss:0.050941
## [51] train-logloss:0.050703
## [52] train-logloss:0.049621
## [53] train-logloss:0.049192
## [54] train-logloss:0.048657
## [55] train-logloss:0.047453
## [56] train-logloss:0.047032
## [57] train-logloss:0.046678
## [58] train-logloss:0.045860
## [59] train-logloss:0.045278
## [60] train-logloss:0.044713
## [61] train-logloss:0.043925
## [62] train-logloss:0.043413
## [63] train-logloss:0.042905
## [64] train-logloss:0.041984
## [65] train-logloss:0.041862
## [66] train-logloss:0.040743
## [67] train-logloss:0.040205
## [68] train-logloss:0.039481
## [69] train-logloss:0.039066
## [70] train-logloss:0.038756
## [71] train-logloss:0.038510
## [72] train-logloss:0.037882
## [73] train-logloss:0.037598
## [74] train-logloss:0.037305
## [75] train-logloss:0.036281
## [76] train-logloss:0.036016
## [77] train-logloss:0.035825
## [78] train-logloss:0.035507
## [79] train-logloss:0.034653
## [80] train-logloss:0.033567
## [81] train-logloss:0.033146
## [82] train-logloss:0.032608
## [83] train-logloss:0.032139
## [84] train-logloss:0.031456
## [85] train-logloss:0.031244
## [86] train-logloss:0.030655
## [87] train-logloss:0.029964
## [88] train-logloss:0.029665
## [89] train-logloss:0.029447
## [90] train-logloss:0.029177
## [91] train-logloss:0.028864
## [92] train-logloss:0.028347
## [93] train-logloss:0.028214
## [94] train-logloss:0.028080
## [95] train-logloss:0.027696
## [96] train-logloss:0.027538
## [97] train-logloss:0.027222
## [98] train-logloss:0.027089
## [99] train-logloss:0.026856
## [100]    train-logloss:0.026515
```

```
importance_matrix = xgb.importance(feature_names = colnames(traind[, -which(names(traind) == "Status")]]
xgb.plot.importance(importance_matrix)
```



```
xgb.plot.tree(feature_names = colnames(traind[, -which(names(traind) == "Status")]), model = model_xgb,
```



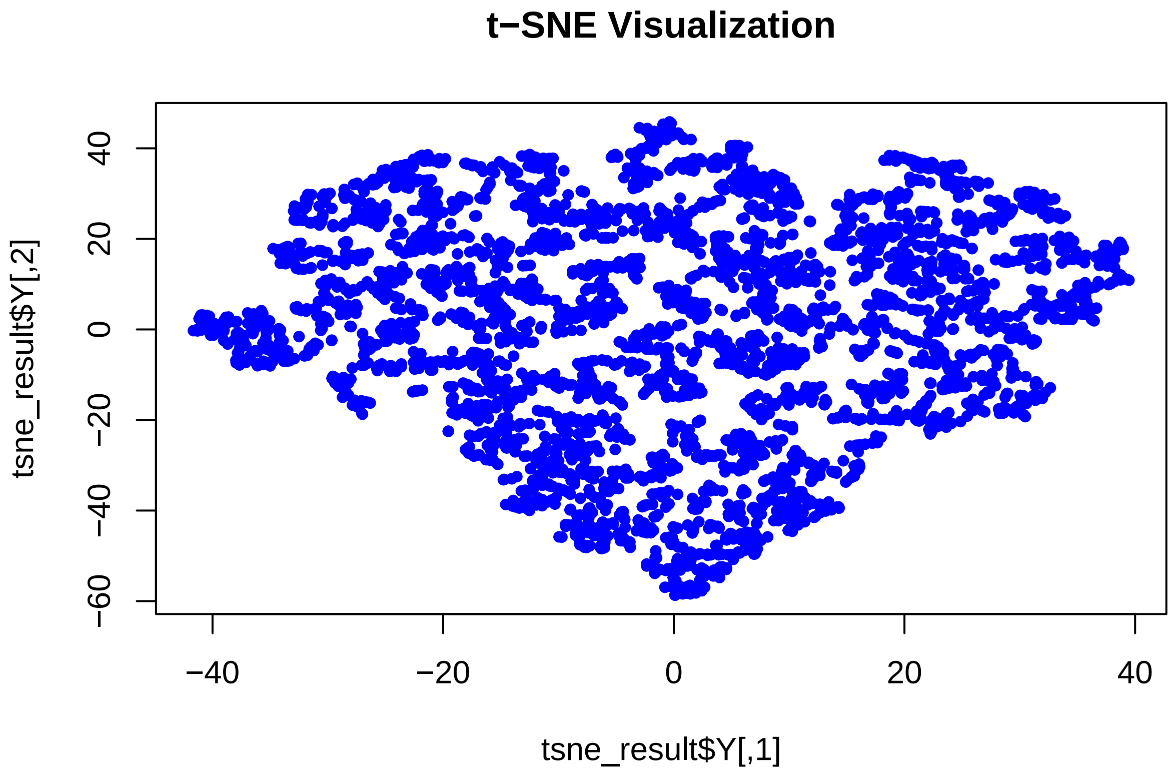
```r
testd2 <- as.matrix(testd[, -which(names(testd) == "Status")])
```

```r
#Accuracy :  0.7835, Specificity : 0.2161
pred_probs <- predict(model_xgb, testd2)
pred_labels <- ifelse(pred_probs > 0.5, 1, 0)
confusionMatrix(as.factor(pred_labels), as.factor(testd$Status))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 909 156
##          1 107  43
##
##                Accuracy : 0.7835
##                  95% CI : (0.7593, 0.8064)
##     No Information Rate : 0.8362
##     P-Value [Acc > NIR] : 0.999999
##
##                   Kappa : 0.1229
##
##  Mcnemar’s Test P-Value : 0.003078
##
##             Sensitivity : 0.8947
##             Specificity : 0.2161
##          Pos Pred Value : 0.8535
##          Neg Pred Value : 0.2867
##              Prevalence : 0.8362
##          Detection Rate : 0.7481
##    Detection Prevalence : 0.8765
##       Balanced Accuracy : 0.5554
##
##        ’Positive’ Class : 0
##
```

Dimensionality reduction

```r
##t-SNE
library(Rtsne)
df3_unique <- df3[!duplicated(df3), ]
# Run t-SNE
tsne_result <- Rtsne(df3_unique)
# Plot the result
plot(tsne_result$Y, col = "blue", pch = 20, main = "t-SNE Visualization")
```

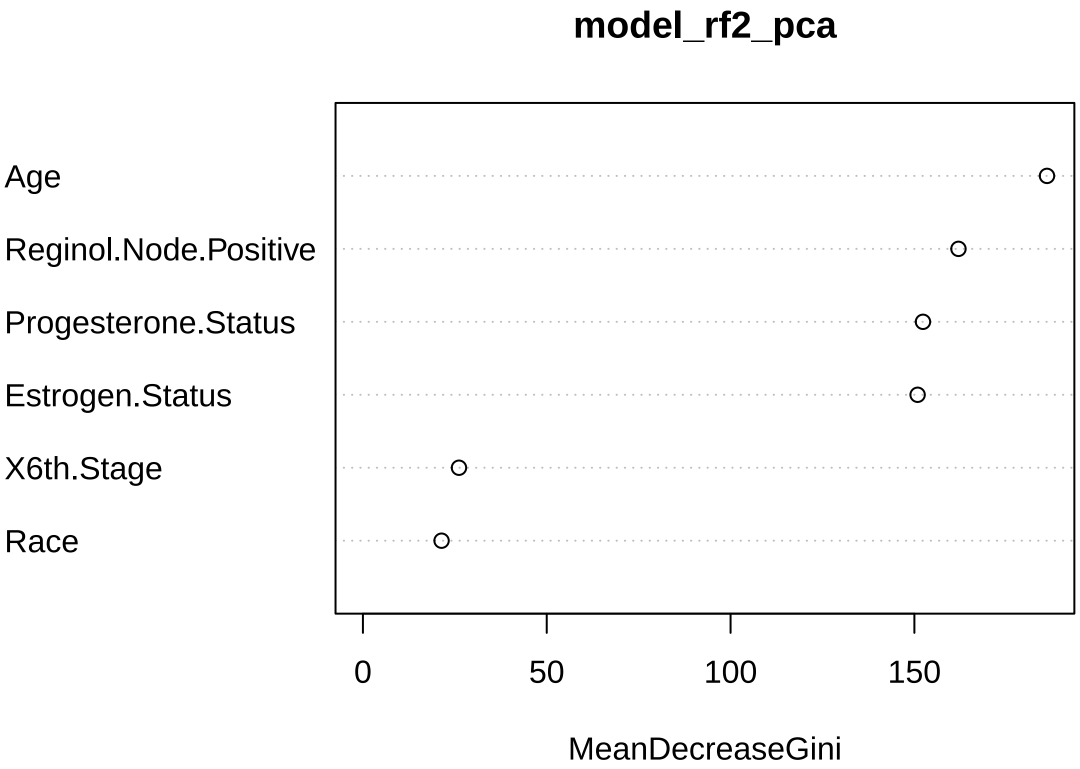


```r
## PCA
library(caret)
library(dplyr)

# Eliminate duplicates
numdata_unique <-numdata[!duplicated(numdata), ]
numdata_unique$Status <- as.factor(numdata_unique$Status)

#training and testing
nn <- nrow(numdata_unique)
sample2 <- sample(c(TRUE, FALSE), nn, replace=TRUE, prob=c(0.7,0.3))
trainpca  <- numdata_unique[sample2, ]
testpca   <- numdata_unique[!sample2, ]

#taking the numeric columns
numeric_columns <- sapply(trainpca, is.numeric)
ntrainpca  <- trainpca[,numeric_columns]
ntestpca <- testpca[,numeric_columns]

# Scale the numeric data (important for PCA)
scaled_train_data <- scale(ntrainpca)
scaled_test_data <- scale(ntestpca)

# Perform PCA on the training data
pca_model <- prcomp(scaled_train_data)
```

```r
# Transform the numeric features of the training data
train_pca <- predict(pca_model, scaled_train_data)

# Transform the numeric features of the testing data using the same PCA transformation
test_pca <- predict(pca_model, scaled_test_data)

# Replace the original numeric features with the PCA-transformed features in the datasets
trainpca[, numeric_columns] <- train_pca
testpca[, numeric_columns] <- test_pca

#NINETH MODEL: RANDOM FOREST WITH PCA
model_rf2_pca = randomForest(Status  ~  X6th.Stage+Progesterone.Status+Estrogen.Status+Reginol.Node.Posi
summary(model_rf2_pca)
```

```
##                 Length Class  Mode
## call               4   -none- call
## type               1   -none- character
## predicted       2820   factor numeric
## err.rate         300   -none- numeric
## confusion          6   -none- numeric
## votes           5640   matrix numeric
## oob.times       2820   -none- numeric
## classes            2   -none- character
## importance         6   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y               2820   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```r
varImpPlot(model_rf2_pca)
```



```
#Accuracy: 0.8544996, specificity:  0.1325301
predicted7 <- predict(model_rf2_pca, newdata=testpca, type="response")
conf_matrix<-table(predicted7, testpca$Status)
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy
```

```
## [1] 0.8354324
```

```
specificity <- conf_matrix[2,2]/sum(as.numeric(testpca$Status)-1)
specificity
```

```
## [1] 0.1755319
```

```
## TENTH MODEL: Logistic Regression with selected variables
model_logreg2 <- glm(Status ~ X6th.Stage+Progesterone.Status+Estrogen.Status+Reginol.Node.Positive+Age+R
summary(model_logreg2)
```

```
##
## Call:
## glm(formula = Status ~ X6th.Stage + Progesterone.Status + Estrogen.Status +
##     Reginol.Node.Positive + Age + Race, family = binomial(),
##     data = trainpca)
##
## Coefficients:
```

```
##                        Estimate Std. Error z value Pr(>|z|)    
## (Intercept)            -1.39873    0.19999  -6.994 2.67e-12 ***
## X6th.Stage.L           -0.21556    0.40061  -0.538 0.590527    
## X6th.Stage.Q           -0.27461    0.17578  -1.562 0.118230    
## X6th.Stage.C           -0.06037    0.24531  -0.246 0.805617    
## X6th.Stage^4           -0.48293    0.23249  -2.077 0.037783 *  
## Progesterone.Status     0.02921    0.02036   1.435 0.151244    
## Estrogen.Status         0.32519    0.04083   7.965 1.65e-15 ***
## Reginol.Node.Positive   0.16332    0.17974   0.909 0.363540    
## Age                     0.49504    0.05062   9.779  < 2e-16 ***
## RaceOther              -0.84381    0.27654  -3.051 0.002278 ** 
## RaceWhite              -0.61220    0.18500  -3.309 0.000936 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2401.4  on 2819  degrees of freedom
## Residual deviance: 2068.2  on 2809  degrees of freedom
## AIC: 2090.2
## 
## Number of Fisher Scoring iterations: 5
```

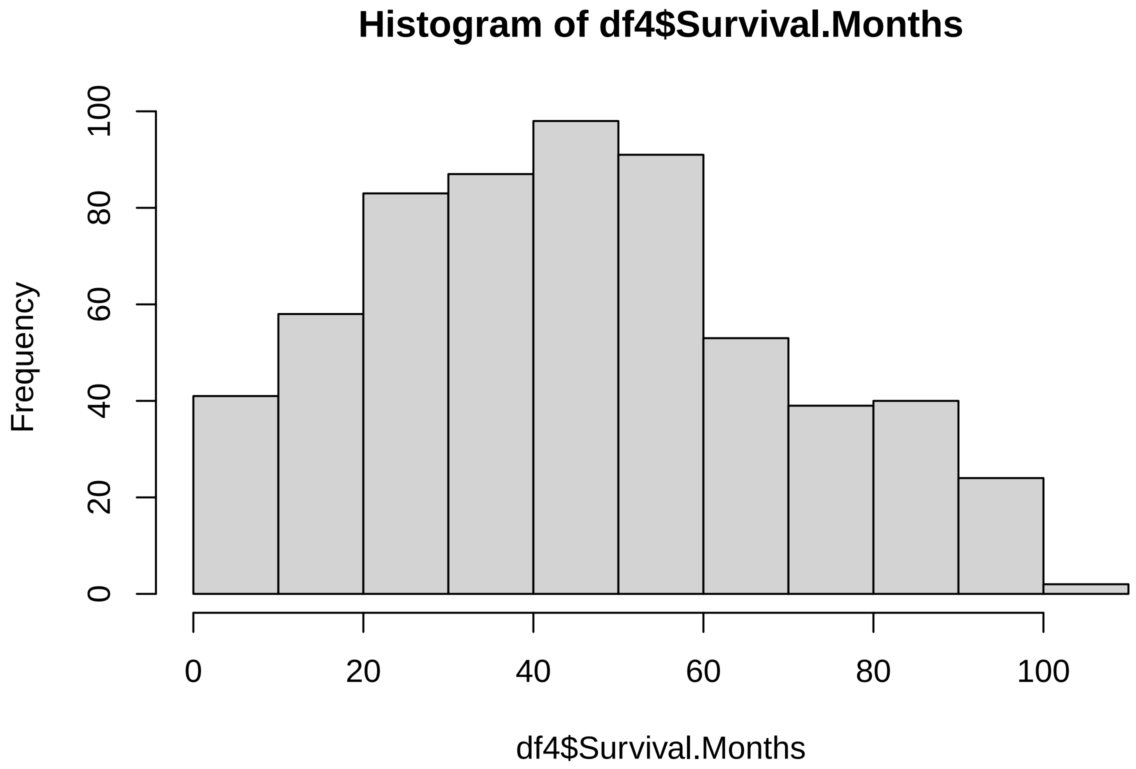
```
#Accuracy : 0.8638 ,   AIC: 2195.9, Specificity : 0.1325
predicted_probs2 <- predict(model_logreg2, newdata=testpca, type="response")
predicted_class2 <- ifelse(predicted_probs2 > 0.5, 1, 0)
confusionMatrix(as.factor(predicted_class2), testpca$Status)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 987 169
##          1  16  19
##                                           
##                Accuracy : 0.8447          
##                  95% CI : (0.8228, 0.8648)
##     No Information Rate : 0.8421          
##     P-Value [Acc > NIR] : 0.4247          
##                                           
##                   Kappa : 0.1272          
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 0.9840          
##             Specificity : 0.1011          
##          Pos Pred Value : 0.8538          
##          Neg Pred Value : 0.5429          
##              Prevalence : 0.8421          
##          Detection Rate : 0.8287          
##    Detection Prevalence : 0.9706          
##       Balanced Accuracy : 0.5426          
##                                           
##        'Positive' Class : 0               
```

```
##
```

Life Expectancy:

```
df4 <- df2[df2$Status == 1, ]
hist(df4$Survival.Months)
```



```
df4$Status <-NULL
#Training and testing
trainpo  <- df4[sample, ]
testpo   <- df4[!sample, ]

poisson_model <- glm(Survival.Months~.,data=trainpo,family = poisson)
summary(poisson_model)
```

```
##
## Call:
## glm(formula = Survival.Months ~ ., family = poisson, data = trainpo)
##
## Coefficients: (1 not defined because of singularities)
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               3.6095368  0.0766419  47.096  < 2e-16 ***
## Age                      -0.0038939  0.0008152  -4.777 1.78e-06 ***
## RaceOther                 0.1169409  0.0363434   3.218 0.001292 **
## RaceWhite                 0.1174575  0.0232135   5.060 4.20e-07 ***
## Marital.StatusMarried     0.0578694  0.0217089   2.666 0.007683 **
## Marital.StatusSeparated -0.0621916  0.0521824  -1.192 0.233335
## Marital.StatusSingle      0.0633746  0.0270351   2.344 0.019070 *
## Marital.StatusWidowed     0.0824576  0.0346316   2.381 0.017266 *
## T.Stage.L                 0.1044231  0.0392082   2.663 0.007738 **
## T.Stage.Q                 0.0630722  0.0298945   2.110 0.034873 *
## T.Stage.C                 0.0512002  0.0231131   2.215 0.026746 *
## N.Stage.L                 0.4433876  0.2100615   2.111 0.034794 *
## N.Stage.Q                 0.2760805  0.1355209   2.037 0.041632 *
## X6th.Stage.L             -0.5080491  0.2036845  -2.494 0.012621 *
## X6th.Stage.Q             -0.4018305  0.1565758  -2.566 0.010277 *
## X6th.Stage.C             -0.1079739  0.0805579  -1.340 0.180139
## X6th.Stage^4                     NA         NA      NA       NA
## differentiate.L           0.0024392  0.0524979   0.046 0.962941
## differentiate.Q          -0.0074019  0.0394582  -0.188 0.851200
## differentiate.C          -0.0040819  0.0200430  -0.204 0.838620
## A.Stage                  -0.3324143  0.0432144  -7.692 1.45e-14 ***
## Tumor.Size               -0.0007722  0.0005781  -1.336 0.181656
## Estrogen.Status           0.2332782  0.0272058   8.575  < 2e-16 ***
## Progesterone.Status       0.2421984  0.0201680  12.009  < 2e-16 ***
## Regional.Node.Examined   -0.0028105  0.0011490  -2.446 0.014445 *
## Reginol.Node.Positive     0.0083733  0.0021763   3.848 0.000119 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5598.1  on 435  degrees of freedom
## Residual deviance: 4814.2  on 411  degrees of freedom
##   (2373 observations deleted due to missingness)
## AIC: 7249.8
##
## Number of Fisher Scoring iterations: 5
```