Homework 4

Richard Albright
ISYE6414
Spring 2020

```
data = read.csv("breast_cancer_V2.csv", sep=',', header=TRUE)
data$Survival = data$Survived / data$Patients
```

Question 1: Fitting a Model (Links to an external site.)

Fit a logistic regression model using Survival as the response variable with NodeCount as the predictor and logit as the link function. Call it model1.

```
model1 <- glm(Survival ~ NodeCount, weights=Patients, data=data, family=binomial)
```

(a) Display the summary of model1. What are the model parameters and estimates?

```
print_output(summary(model1))
```

```
Call:
glm(formula = Survival ~ NodeCount, family = binomial, data = data,
    weights = Patients)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
−9.3939  −3.3772  −0.0395   3.5266   8.8649

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.509155   0.029618   17.19   <2e−16 ***
NodeCount   −0.045828   0.003501  −13.09   <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: 3091.4  on 151  degrees of freedom
Residual deviance: 2904.8  on 150  degrees of freedom
AIC: 3438.4

Number of Fisher Scoring iterations: 4
```

(b) Write down the equation for the Odds of Survival.

exp(Survival) = 0.509155 + (-0.045828 * NodeCount)

(c) Provide a meaningful interpretation for the coefficient for NodeCount with respect to the log-odds of survival and the odds of survival.

For every increase in NodeCount the log odds of survival decreases by -0.045828 and the odds of survival decreases by a factor of 0.9552062 (exp(-0.045828)).

Question 2: Inference (Links to an external site.)

(a) Using model1, find a 90% confidence interval for the coefficient for NodeCount.

```
logOddsNodeCt <- confint.default(model1, level=0.9)
logOddsNodeCt
```

```
                5 %         95 %
```

(Intercept) 0.46043671 0.55787278 NodeCount -0.05158601 -0.04006954

At a 90% confidence interval the log odds of the coefficient for NodeCount is between -0.05158601 and -0.04006954.

```
probNodeCt <-exp(logOddsNodeCt)
probNodeCt
```

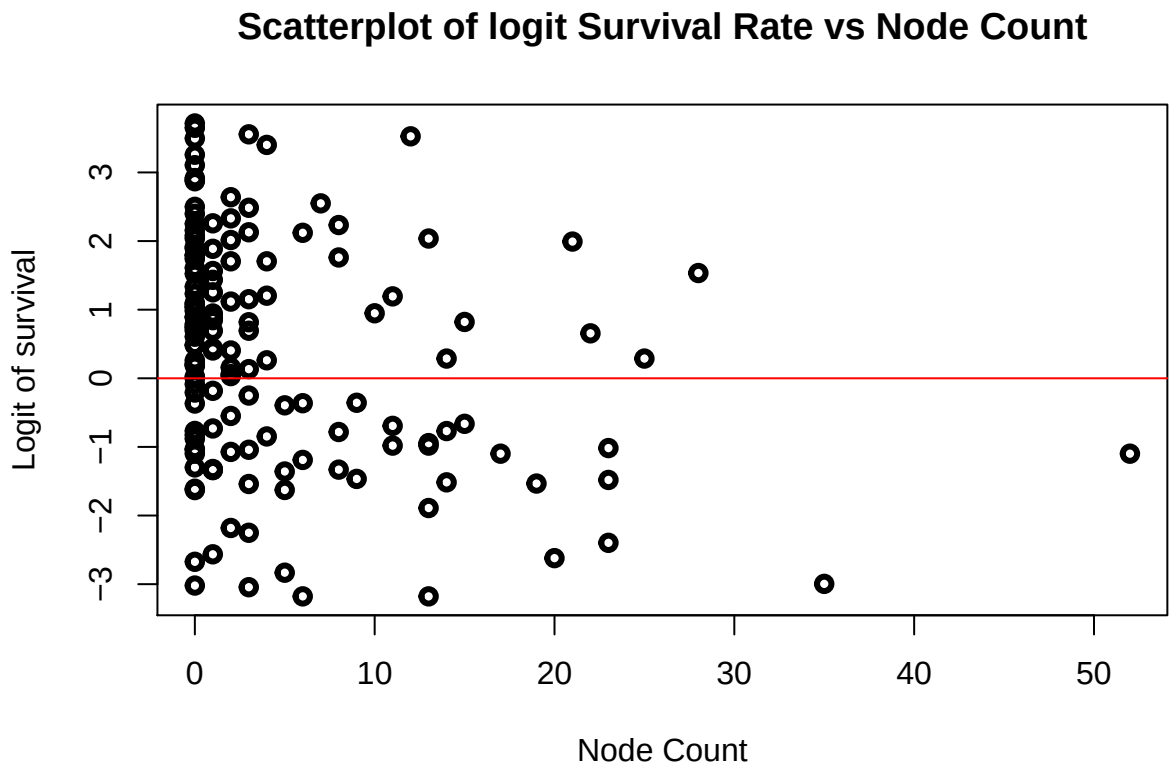
```
              5 %      95 %
```

(Intercept) 1.584766 1.7469524 NodeCount 0.949722 0.9607226

At a 90% confidence interval the probability factor of the coefficient for NodeCount is between 0.949722 and 0.9607226.

(b) Is model1 significant overall? How do you come to your conclusion?

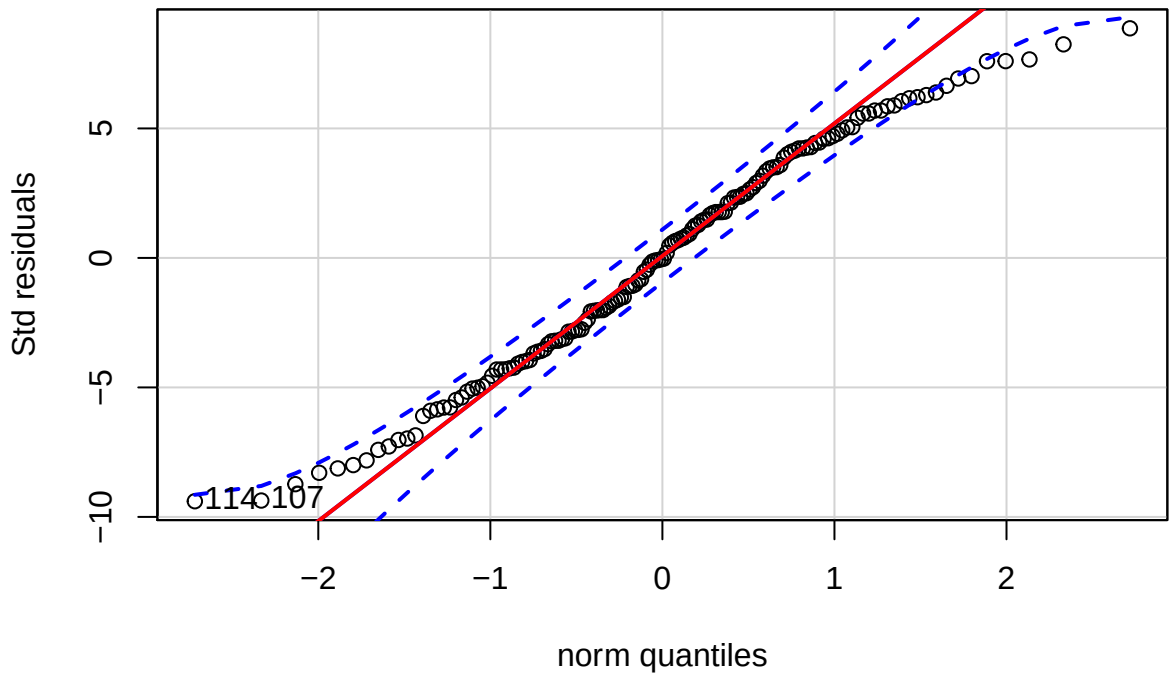
```
plot(
  data$NodeCount,
  log((data$Survived/data$Patients)/(1-data$Survived/data$Patients)),
  ylab="Logit of survival",
  xlab='Node Count',
  main="Scatterplot of logit Survival Rate vs Node Count",
  lwd=3)
abline(0,0, col='red')
```




```
car::qqPlot(residuals(model1), ylab="Std residuals")
```

[1] 114 107

```
qqline(residuals(model1, type='deviance'),col="red",lwd=2)
```




```
gstat = model1$null.deviance - deviance(model1)
pval = 1-pchisq(gstat,length(coef(model1))-1)
cbind(gstat, pval)
```

```
     gstat pval
```

[1,] 186.6847 0

The p-value of the model is approxiately 0. The model shows linearity and is normally distributed. The model is statistically significant.

(c) Which variables are significantly nonzero at the 0.01 significance level? Which are significantly negative? Why?

At a 0.01 significance level, the intercept is significantly nonzero at 0.509155 with a z-score of 17.19. The probability of the intecept being 0 is <2e-16 < 0.01. The node count is significantly negative at -0.045828 with a z-score of -13.09. The probability of the node count coefficient being 0 is <2e-16 < 0.01.

Question 3: Goodness of fit (Links to an external site.)

(a) Perform goodness of fit hypothesis tests using both deviance and Pearson residuals. What do you conclude? Explain the differences, if any, between these findings and what you found in Question 2b.

Deviance goodness of fit test.

Using the Chisq test in the anova function.

```
anova(model1, test="Chisq")
```

Analysis of Deviance Table

Model: binomial, link: logit

Response: Survival

Terms added sequentially (first to last)

```
      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
```

NULL 151 3091.4
NodeCount 1 186.69 150 2904.8 < 2.2e-16 *** — Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Using the phisq function.

```
c(deviance(model1), 1-pchisq(deviance(model1),150))
```

[1] 2904.764 0.000

Pearson residuals goodness of fit test using the phisq function.

```
1-pchisq(sum(residuals(model1, type = "pearson")^2), 150)
```

[1] 0

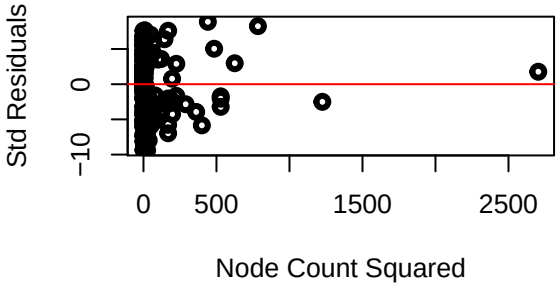
There are no differences between the 2. At a 0.01 significance level, both tests indicate the model is a good fit.

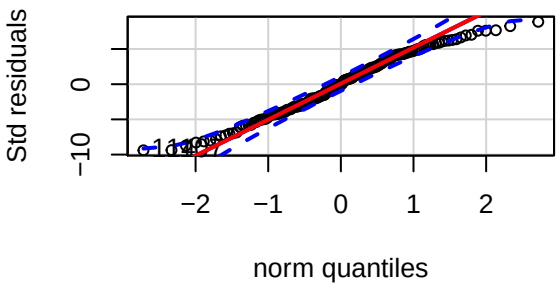
(b) Perform visual analytics for checking goodness of fit for this model and write your observations. Be sure to address the model assumptions. Only deviance residuals are required for this question.

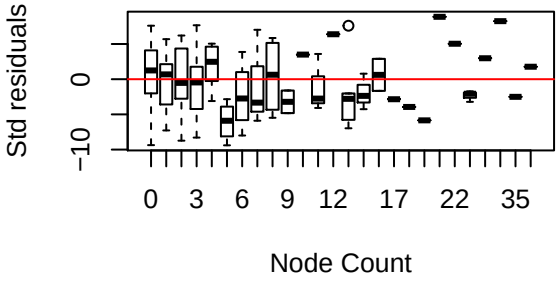
```
res = residuals(model1, type='deviance')
par(mfrow=c(2,2))
plot(data$NodeCount^2, res, ylab='Std Residuals', xlab='Node Count Squared', lwd=3)
abline(0,0, col='red')
car::qqPlot(res, ylab="Std residuals")
```

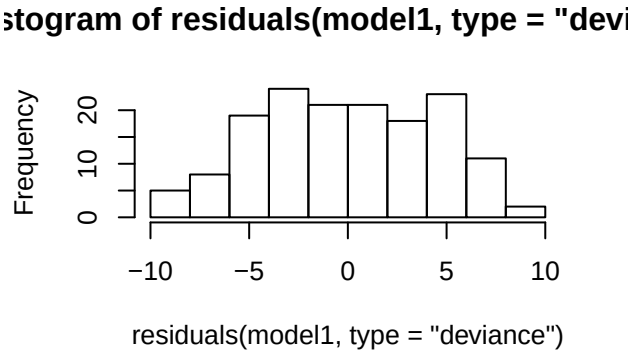
[1] 114 107

```
qqline(res,col="red",lwd=2)
boxplot(res~data$NodeCount,ylab = "Std residuals", xlab='Node Count')
abline(0,0, col='red')
hist(residuals(model1, type='deviance'))
```









The linearity assumption of Node Count vs the logit of the Survival Rate holds. The residuals are mostly normally distributed with a heavy tail on the right side of the distibution. The variance is not constant across node counts due to there being more data available in the sample for smaller node counts.

(c) Calculate the dispersion parameter for this model. Is this an overdispersed model?

Dispersion parameter using deviance.

```
sum(residuals(model1, type = "deviance")^2)/model1$df.residual
```

[1] 19.36509

Dispersion parameter using Pearson.

```
sum(residuals(model1, type = "pearson")^2)/model1$df.residual
```

[1] 17.17921

The dispersion parameter is using deviance is 19.36509, and using pearson is 17.17921. The model is overdispersed.

Question 4: Fitting the full model (Links to an external site.)

Fit a logistic regression model using Survival as the response variable with Age, OperationYear, and Node-Count as the predictors and logit as the link function. Call it model2.

```
model2 <- glm(Survival ~ Age + OperationYear + NodeCount,
              weights=Patients, data=data, family='binomial')
print_output(summary(model2))
```

```
Call:
glm(formula = Survival ~ Age + OperationYear + NodeCount, family = "binomial",
    data = data, weights = Patients)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
−10.2773   −2.8161    0.2896    2.7769   10.8433

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.181502   0.493448   4.421 9.83e−06 ***
Age           −0.045833   0.002333 −19.644  < 2e−16 ***
OperationYear  0.011898   0.007820   1.521    0.128
NodeCount     −0.053178   0.003685 −14.430  < 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: 3091.4  on 151  degrees of freedom
Residual deviance: 2491.3  on 148  degrees of freedom
AIC: 3028.9

Number of Fisher Scoring iterations: 4
```

(a) Write down the equation for the probability of Survival.

exp(Survival) = 2.181502 + (-0.045833 * Age) + (0.011898 * OperationYear) + (-0.053178 * NodeCount)

(b) Provide a meaningful interpretation for the coefficients of Age and OperationYear with respect the to the odds of survival.

(c) Is OperationYear significant given the other variables in model2?

No. Operation Year is not significant given the Probability of the Z-value is 0.128.

(d) Has your goodness of fit been affected? Repeat the tests, plots, and dispersion parameter calculation you performed in Question 3 with model2.

```
test <- deviance(model1) - deviance(model2)
cbind(test, 1-pchisq(test,2))
```

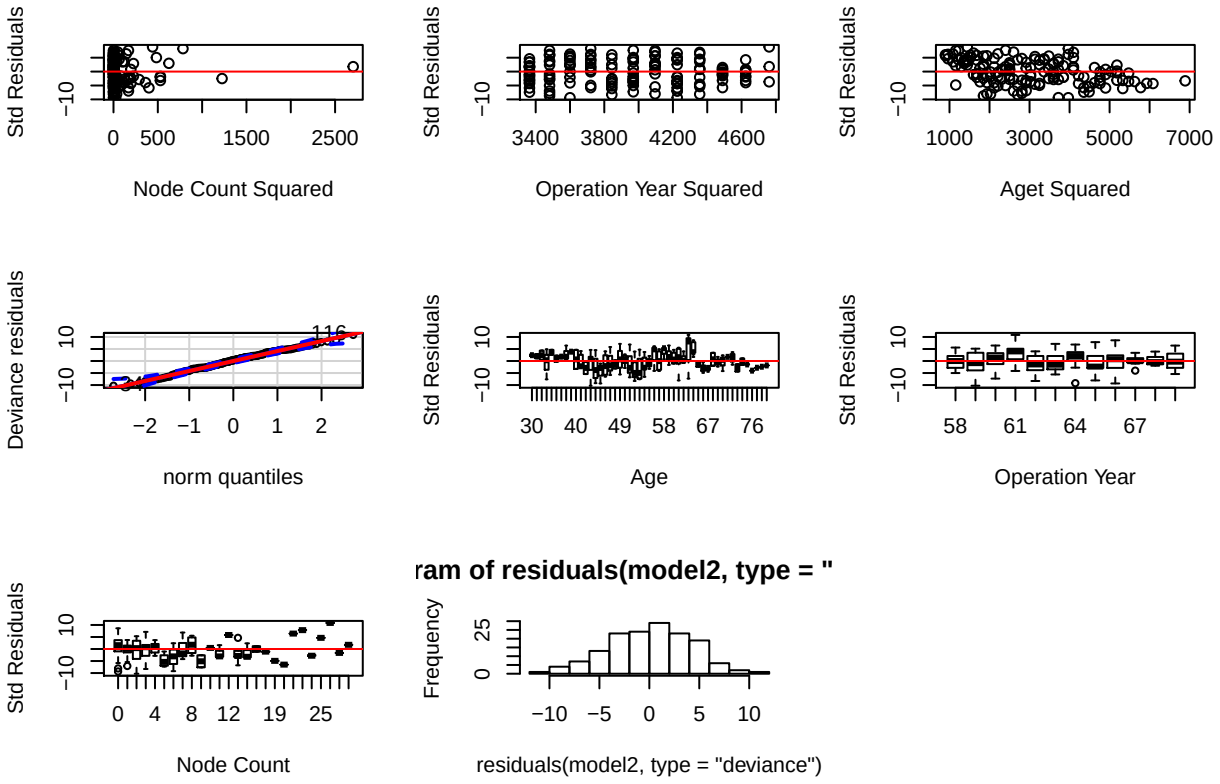
```
     test
```

[1,] 413.4662 0

```
par(mfrow=c(3,3))
plot(data$NodeCount^2 ,
     res,ylab="Std Residuals",xlab="Node Count Squared")
abline(0,0, col='red')
plot(data$OperationYear^2 ,
     res,ylab="Std Residuals",xlab="Operation Year Squared")
abline(0,0, col='red')
plot(data$Age^2,
     res,ylab="Std Residuals",xlab="Aget Squared")
abline(0,0, col='red')
car::qqPlot(residuals(model2, type='deviance'),
            ylab="Deviance residuals")
```

[1] 116 34

```
qqline(residuals(model2, type='deviance'),col="red",lwd=2)
boxplot(residuals(model2, type='deviance')~data$Age,
        ylab = "Std Residuals", xlab='Age')
abline(0,0, col='red')
boxplot(residuals(model2, type='deviance')~data$OperationYear,
        ylab = "Std Residuals", xlab='Operation Year')
abline(0,0, col='red')
boxplot(residuals(model2, type='deviance')~data$NodeCount,
        ylab = "Std Residuals", xlab='Node Count')
abline(0,0, col='red')
hist(residuals(model2, type='deviance'))
```



The deviance is reduced by 180.085 with 2 less degress of freedom ans is statistically significant. The goodness of fit has been impproved. The heavy tail in the distribution of residuals on the right side of model 2 dissapears in model 3, and is now normally distributed. The linearity assumption holds in model 3. Constant variance is still lacking for Node Count, but exists in Age and Operation Year.

(e) Overall, would you say model2 is a good-fitting model? If so, why? If not, what would you suggest to improve the fit and why? Note, we are not asking you to spend hours finding the best possible model but to offer plausible suggestions along with your reasoning.

Model 2 is a good fitting model, but age might work better as a factor variable, there may be a linear trend of in the middle of the boxplot chart that indicates we may be better off creating a dummy variable based on age ranges.

```
model3 <- glm(Survival ~ as.factor(Age) + OperationYear + NodeCount,
              weights=Patients, data=data, family=binomial)
print_output(summary(model3))
```

```
Call:
glm(formula = Survival ~ as.factor(Age) + OperationYear + NodeCount,
    family = binomial, data = data, weights = Patients)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
−11.1645   −1.7247   −0.0014    2.2862    7.9610

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        0.446885   0.705521   0.633 0.526465
as.factor(Age)31  −0.246506   0.397731  −0.620 0.535401
as.factor(Age)33  −0.351519   0.398128  −0.883 0.377274
as.factor(Age)34  −1.392166   0.350396  −3.973 7.09e−05 ***
as.factor(Age)35   0.399855   0.451798   0.885 0.376141
as.factor(Age)36  −0.750624   0.401888  −1.868 0.061797 .
as.factor(Age)37  −0.401098   0.378882  −1.059 0.289766
as.factor(Age)38  −0.276051   0.357478  −0.772 0.439985
as.factor(Age)39   0.616476   0.551027   1.119 0.263236
as.factor(Age)40  −1.312457   0.356990  −3.676 0.000236 ***
as.factor(Age)41  −1.292175   0.338009  −3.823 0.000132 ***
as.factor(Age)42  −2.372573   0.369846  −6.415 1.41e−10 ***
as.factor(Age)43  −2.036405   0.327642  −6.215 5.12e−10 ***
as.factor(Age)44  −2.314396   0.334966  −6.909 4.87e−12 ***
as.factor(Age)45  −3.237882   0.369193  −8.770  < 2e−16 ***
as.factor(Age)46  −2.093227   0.332454  −6.296 3.05e−10 ***
as.factor(Age)47  −1.955263   0.339207  −5.764 8.20e−09 ***
as.factor(Age)48  −2.153562   0.374437  −5.751 8.85e−09 ***
as.factor(Age)49  −1.557735   0.339267  −4.591 4.40e−06 ***
as.factor(Age)50  −1.722721   0.325054  −5.300 1.16e−07 ***
as.factor(Age)51  −1.944061   0.335773  −5.790 7.05e−09 ***
as.factor(Age)52  −2.815255   0.341725  −8.238  < 2e−16 ***
as.factor(Age)53  −3.022279   0.340343  −8.880  < 2e−16 ***
as.factor(Age)54  −2.421243   0.330043  −7.336 2.20e−13 ***
as.factor(Age)55  −2.129714   0.335742  −6.343 2.25e−10 ***
as.factor(Age)56  −1.442054   0.346510  −4.162 3.16e−05 ***
as.factor(Age)57  −1.990086   0.333955  −5.959 2.54e−09 ***
as.factor(Age)58  −0.807030   0.370002  −2.181 0.029172 *
as.factor(Age)59  −1.187657   0.384555  −3.088 0.002012 **
as.factor(Age)60  −1.887062   0.328376  −5.747 9.10e−09 ***
as.factor(Age)61  −2.246415   0.326531  −6.880 6.00e−12 ***
as.factor(Age)62  −1.914373   0.345290  −5.544 2.95e−08 ***
as.factor(Age)63  −1.344029   0.340201  −3.951 7.79e−05 ***
as.factor(Age)64  −0.863643   0.336276  −2.568 0.010221 *
as.factor(Age)65  −3.025672   0.402041  −7.526 5.24e−14 ***
as.factor(Age)66  −3.496575   0.365124  −9.576  < 2e−16 ***
as.factor(Age)67  −3.513133   0.342762 −10.249  < 2e−16 ***
as.factor(Age)68  −2.553681   0.379788  −6.724 1.77e−11 ***
as.factor(Age)69  −2.907321   0.355404  −8.180 2.83e−16 ***
as.factor(Age)70  −2.879073   0.424722  −6.779 1.21e−11 ***
as.factor(Age)71  −2.113041   0.549717  −3.844 0.000121 ***
as.factor(Age)72  −2.149167   0.402126  −5.345 9.07e−08 ***
as.factor(Age)73  −4.438770   0.567841  −7.817 5.41e−15 ***
as.factor(Age)74  −3.155313   0.505276  −6.245 4.25e−10 ***
as.factor(Age)75  −3.709649   0.638789  −5.807 6.35e−09 ***
as.factor(Age)76 −19.212903 591.935788  −0.032 0.974107
as.factor(Age)77 −18.476542 606.617136  −0.030 0.975702
as.factor(Age)78  −5.049828   1.081040  −4.671 2.99e−06 ***
as.factor(Age)83 −17.689094 621.650149  −0.028 0.977299
OperationYear      0.032328   0.010134   3.190 0.001422 **
NodeCount         −0.063358   0.004302 −14.727  < 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: 3091.4  on 151  degrees of freedom
Residual deviance: 1669.6  on 101  degrees of freedom
AIC: 2301.3

Number of Fisher Scoring iterations: 13
```

```
agetest = deviance(model2)-deviance(model3)
cbind(agetest, 1-pchisq(agetest,49))
```

```
  agetest
```

[1,] 821.6711 0

Testing for age goodness of fit, model 3 is an improvement over model 2. All ages between 40 and 75 are statistically significant at the 0.05 level. Since predictions would be less accurate for ages not included as factors, we may want to create an age factor variable where 1 = Age < 40, 2 = Age >= 40 and Age <= 75, and 3 = Age > 75.

```
data$AgeFactor <- cut(data$Age, c(0, 39, 74, 120))
model4 <- glm(Survival ~ AgeFactor + OperationYear + NodeCount,
              weights=Patients, data=data, family=binomial)
print_output(summary(model4))
```

```
Call:
glm(formula = Survival ~ AgeFactor + OperationYear + NodeCount,
    family = binomial, data = data, weights = Patients)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-9.6029  -2.4093  -0.1715   2.4141   9.5482

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       1.9458444  0.4998547   3.893 9.91e-05 ***
AgeFactor(39,74]  -1.6785824  0.0847517 -19.806  < 2e-16 ***
AgeFactor(74,120] -4.4319667  0.4722978  -9.384  < 2e-16 ***
OperationYear      0.0007726  0.0078832   0.098    0.922
NodeCount         -0.0511676  0.0037501 -13.644  < 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: 3091.4  on 151  degrees of freedom
Residual deviance: 2311.2  on 147  degrees of freedom
AIC: 2850.8

Number of Fisher Scoring iterations: 4
```

```
agetest = deviance(model2)-deviance(model4)
cbind(agetest, 1-pchisq(agetest,2))
```

```
 agetest
```

[1,] 180.085 0

Testing for age between model 4 and model 2. Model 4 is also an improvement over model 2, and would have better predictions for ages that were not included as factors in model 3. I would used model 4

Question 5: Prediction (Links to an external site.)

Suppose a 31-year-old individual with 3 nodes is operated on in 1970.

```
patient <- data.frame(Age=31, NodeCount=3, OperationYear=1970)
cost0.5 = function(y, pi){
  ypred=rep(0,length(y))
  ypred[pi>0.5] = 1
  err = mean(abs(y-ypred))
  return(err)}
cost0.6 = function(y, pi){
  ypred=rep(0,length(y))
  ypred[pi>0.6] = 1
  err = mean(abs(y-ypred))
  return(err)}
cost0.7 = function(y, pi){
  ypred=rep(0,length(y))
  ypred[pi>0.7] = 1
  err = mean(abs(y-ypred))
  return(err)}
cost0.8 = function(y, pi){
  ypred=rep(0,length(y))
  ypred[pi>0.8] = 1
  err = mean(abs(y-ypred))
  return(err)}
```

(a) Predict their probability of survival using model1.

```
model1.predict = predict.glm(model1,patient,type="response")
model1.err = cv.glm(data,model1,cost=cost0.6, K=10)$delta[1]
cbind(model1.predict, model1.err)
```

model1.predict model1.err 1 0.5918628 0.4015242

(b) Predict their probability of survival using model2.

```
model2.predict = predict.glm(model2,patient,type="response")
model2.err = cv.glm(data,model2,cost=cost0.5, K=10)$delta[1]
cbind(model2.predict, model2.err)
```

model2.predict model2.err 1 1 0.3483578

(c) Comment on how your predictions compare.

Based solely on node count, the patient has a 59% chance of surviving with a 41% chance of being misclassfied using a cost function of 0.6. But when age and operation year is considered as well, the patient has a 100% chance of surviving wuth a 36% chance of being misclassified using a cost function of 0.5.